STATA
LONGITUDINAL-DATA/PANEL-DATA
REFERENCE MANUAL
RELEASE 19

A Stata Press Publication
StataCorp LLC
College Station, Texas

Published by Stata Press, 4905 Lakeway Drive, College Station, Texas 77845

ISBN-10: 1-59718-450-0
ISBN-13: 978-1-59718-450-2

The suggested citation for this software is

StataCorp. 2025. *Stata 19*. Statistical software. StataCorp LLC.

The suggested citation for this manual is

StataCorp. 2025. *Stata 19 Longitudinal-Data/Panel-Data Reference Manual*. College Station, TX: Stata Press.

Contents

Cross-referencing the documentation

When reading this manual, you will find references to other Stata manuals, for example, **[U] 27 Overview of Stata estimation commands**; **[R] regress**; and **[D] reshape**. The first example is a reference to chapter 27, *Overview of Stata estimation commands*, in the *User's Guide*; the second is a reference to the `regress` entry in the *Base Reference Manual*; and the third is a reference to the `reshape` entry in the *Data Management Reference Manual*.

All the manuals in the Stata Documentation have a shorthand notation:

[GSM]	*Getting Started with Stata for Mac*
[GSU]	*Getting Started with Stata for Unix*
[GSW]	*Getting Started with Stata for Windows*
[U]	*Stata User's Guide*
[R]	*Stata Base Reference Manual*
[ADAPT]	*Stata Adaptive Designs: Group Sequential Trials Reference Manual*
[BAYES]	*Stata Bayesian Analysis Reference Manual*
[BMA]	*Stata Bayesian Model Averaging Reference Manual*
[CAUSAL]	*Stata Causal Inference and Treatment-Effects Estimation Reference Manual*
[CM]	*Stata Choice Models Reference Manual*
[D]	*Stata Data Management Reference Manual*
[DSGE]	*Stata Dynamic Stochastic General Equilibrium Models Reference Manual*
[ERM]	*Stata Extended Regression Models Reference Manual*
[FMM]	*Stata Finite Mixture Models Reference Manual*
[FN]	*Stata Functions Reference Manual*
[G]	*Stata Graphics Reference Manual*
[H2OML]	*Machine Learning in Stata Using H2O: Ensemble Decision Trees Reference Manual*
[IRT]	*Stata Item Response Theory Reference Manual*
[LASSO]	*Stata Lasso Reference Manual*
[XT]	*Stata Longitudinal-Data/Panel-Data Reference Manual*
[META]	*Stata Meta-Analysis Reference Manual*
[ME]	*Stata Multilevel Mixed-Effects Reference Manual*
[MI]	*Stata Multiple-Imputation Reference Manual*
[MV]	*Stata Multivariate Statistics Reference Manual*
[PSS]	*Stata Power, Precision, and Sample-Size Reference Manual*
[P]	*Stata Programming Reference Manual*
[RPT]	*Stata Reporting Reference Manual*
[SP]	*Stata Spatial Autoregressive Models Reference Manual*
[SEM]	*Stata Structural Equation Modeling Reference Manual*
[SVY]	*Stata Survey Data Reference Manual*
[ST]	*Stata Survival Analysis Reference Manual*
[TABLES]	*Stata Customizable Tables and Collected Results Reference Manual*
[TS]	*Stata Time-Series Reference Manual*
[I]	*Stata Index*
[M]	*Mata Reference Manual*

Intro — Introduction to longitudinal-data/panel-data manual

Description

This manual documents the xt commands and is referred to as [XT] in cross-references.

Following this entry, [XT] **xt** provides an overview of the xt commands. The other parts of this manual are arranged alphabetically. If you are new to Stata's xt commands, we recommend that you read the following sections first:

[XT] **xt**	Introduction to xt commands
[XT] **xtset**	Declare a dataset to be panel data
[XT] **xtreg**	Linear models for panel data

Stata is continually being updated, and Stata users are always writing new commands. To find out about the latest cross-sectional time-series features, type `search panel data` after installing the latest official updates; see [R] **update**.

Also see

[U] **1.3 What's new**

[R] **Intro** — Introduction to base reference manual

Description   Remarks and examples   References   Also see

Description

The xt series of commands provides tools for analyzing panel data (also known as longitudinal data or, in some disciplines, as cross-sectional time series when there is an explicit time component). Panel datasets have the form $\mathbf{x}_{it}$, where $\mathbf{x}_{it}$ is a vector of observations for unit i and time t. The particular commands (such as `xtdescribe`, `xtsum`, and `xtreg`) are documented in alphabetical order in the entries that follow this entry. If you do not know the name of the command you need, try browsing the second part of this description section, which organizes the xt commands by topic. The next section, *Remarks and examples*, describes concepts that are common across commands.

The `xtset` command sets the panel variable and the time variable; see [XT] **xtset**. Most xt commands require that the panel variable be specified, and some require that the time variable also be specified. Once you `xtset` your data, you need not do it again. The `xtset` information is stored with your data.

If you have previously `tsset` your data by using both a panel and a time variable, these settings will be recognized by `xtset`, and you need not `xtset` your data.

If your interest is in general time-series analysis, see **[U] 27.14 Time-series models** and the *Time-Series Reference Manual*. If your interest is in multilevel mixed-effects models, see **[U] 27.16 Multilevel mixed-effects models** and the *Multilevel Mixed-Effects Reference Manual*. If you are interested in SAR (spatial autoregressive or simultaneously autoregressive) models for panel data, see [SP] **spxtregress**. If you are interested in extended panel-data regression models that address endogenous covariates, nonrandom treatment assignment, and endogenous sample selection, see the *Extended Regression Models Reference Manual*. If you are interested in the mixed logit choice model for panel data, see [CM] **cmxtmixlogit**.

Setup

`xtset`	Declare data to be panel data

Data management and exploration tools

`xtdescribe`	Describe pattern of xt data
`xtsum`	Summarize xt data
`xttab`	Tabulate xt data
`xtdata`	Faster specification searches with xt data
`xtline`	Panel-data line plots

Linear regression estimators

xtreg	Linear models for panel data
xtregar	Fixed- and random-effects linear models with an AR(1) disturbance
xtgls	GLS linear model with heteroskedastic and correlated errors
xtpcse	Linear regression with panel-corrected standard errors
xthtaylor	Hausman–Taylor estimator for error-components model
xtfrontier	Stochastic frontier models for panel data
xtrc	Random-coefficients model
xtivreg	Instrumental variables and two-stage least squares for panel-data models
xtheckman	Random-effects regression with sample selection
xtdidregress	Fixed-effects difference in differences
xthdidregress	Heterogeneous difference in differences for panel data
xteregress	Random-effects models with endogenous covariates, treatment, and sample selection

Unit-root tests

xtunitroot	Panel-data unit-root tests

Cointegration tests

xtcointtest	Panel-data cointegration tests

Dynamic panel-data estimators

xtabond	Arellano–Bond linear dynamic panel-data estimation
xtdpd	Linear dynamic panel-data estimation
xtdpdsys	Arellano–Bover/Blundell–Bond linear dynamic panel-data estimation
xtvar	Panel-data vector autoregressive models

Censored-outcome estimators

xttobit	Random-effects tobit model
xtintreg	Random-effects interval-data regression model
xteintreg	Random-effects interval-data regression models with endogenous covariates, treatment, and sample selection

Binary-outcome estimators

xtlogit	Fixed-effects, random-effects, and population-averaged logit models
xtprobit	Random-effects and population-averaged probit models
xtcloglog	Random-effects and population-averaged cloglog models
xteprobit	Random-effects probit models with endogenous covariates, treatment, and sample selection

Ordinal-outcome estimators

xtologit	Random-effects ordered logistic model
xtoprobit	Random-effects ordered probit model
xteoprobit	Random-effects ordered probit models with endogenous covariates, treatment, and sample selection

Categorical-outcome estimators

xtmlogit	Fixed-effects and random-effects multinomial logit models
cmxtmixlogit	Panel-data mixed logit choice model

Count-data estimators

xtpoisson	Fixed-effects, random-effects, and population-averaged Poisson models
xtnbreg	Fixed-effects, random-effects, & population-averaged negative binomial models

Survival-time estimators

xtstreg	Random-effects parametric survival models

Generalized estimating equations estimator

xtgee	GEE population-averaged panel-data models

Spatial autoregressive or simultaneously autoregressive estimator

spxtregress	Spatial autoregressive models for panel data

Utility

quadchk	Check sensitivity of quadrature approximation

Remarks and examples

Consider having data on n units—individuals, firms, countries, or whatever—over T periods. The data might be income and other characteristics of n persons surveyed each of T years, the output and costs of n firms collected over T months, or the health and behavioral characteristics of n patients collected over T years. In panel datasets, we write x_{it} for the value of x for unit i at time t. The xt commands assume that such datasets are stored as a sequence of observations on (i, t, x).

For a discussion of panel-data models, see Baltagi (2013), Greene (2018, chap. 11), Hsiao (2014), and Wooldridge (2010). Cameron and Trivedi (2022) illustrate many of Stata's panel-data estimators.

For an introduction to linear, nonlinear, and dynamic panel-data analysis in Stata, we offer NetCourse 471, *Introduction to Panel Data Using Stata*; see https://www.stata.com/netcourse/panel-data-intro-nc471/.

▷ Example 1

If we had data on pulmonary function (measured by forced expiratory volume, or FEV) along with smoking behavior, age, sex, and height, a piece of the data might be

```
. list in 1/6, separator(0) divider
```

	pid	yr_visit	fev	age	sex	height	smokes
1.	1071	1991	1.21	25	1	69	0
2.	1071	1992	1.52	26	1	69	0
3.	1071	1993	1.32	28	1	68	0
4.	1072	1991	1.33	18	1	71	1
5.	1072	1992	1.18	20	1	71	1
6.	1072	1993	1.19	21	1	71	0

The xt commands need to know the identity of the variable identifying patient, and some of the xt commands also need to know the identity of the variable identifying time. With these data, we would type

```
. xtset pid yr_visit
```

If we resaved the data, we need not respecify `xtset`.

◁

❑ Technical note

Panel data stored as shown above are said to be in the long form. Perhaps the data are in the wide form with 1 observation per unit and multiple variables for the value in each year. For instance, a piece of the pulmonary function data might be

```
pid    sex  fev91  fev92  fev93  age91  age92   age93
      1071      1   1.21   1.52   1.32     25     26      28
      1072      1   1.33   1.18   1.19     18     20      21
```

Data in this form can be converted to the long form by using `reshape`; see [D] **reshape**.

❑

▷ Example 2

Data for some of the periods might be missing. That is, we have panel data on $i = 1, \dots, n$ and $t = 1, \dots, T$, but only T_i of those observations are defined. With such missing periods—called unbalanced data—a piece of our pulmonary function data might be

```
. list in 1/6, separator(0) divider
```

	pid	yr_visit	fev	age	sex	height	smokes
1.	1071	1991	1.21	25	1	69	0
2.	1071	1992	1.52	26	1	69	0
3.	1071	1993	1.32	28	1	68	0
4.	1072	1991	1.33	18	1	71	1
5.	1072	1993	1.19	21	1	71	0
6.	1073	1991	1.47	24	0	64	0

Patient ID 1072 is not observed in 1992. The xt commands are robust to this problem.

◁

❑ Technical note

In many of the entries in [XT], we will use data from a subsample of the NLSY data (Center for Human Resource Research 1989) on young women aged 14–24 years in 1968. Women were surveyed in each of the 21 years 1968–1988, except for the six years 1974, 1976, 1979, 1981, 1984, and 1986. We use two different subsets: `nlswork.dta` and `union.dta`.

For `nlswork.dta`, our subsample is of 4,711 women in years when employed, not enrolled in school and evidently having completed their education, and with wages in excess of $1/hour but less than $700/hour.

```
. use https://www.stata-press.com/data/r19/nlswork, clear
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)

. describe

Contains data from https://www.stata-press.com/data/r19/nlswork.dta
 Observations:        28,534              National Longitudinal Survey of
                                            Young Women, 14-24 years old in
                                            1968
    Variables:            21              27 Nov 2024 08:14
                                          (_dta has notes)
-------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
-------------------------------------------------------------------------------
idcode          int     %8.0g                 NLS ID
year            byte    %8.0g                 Interview year
birth_yr        byte    %8.0g                 Birth year
age             byte    %8.0g                 Age in current year
race            byte    %8.0g      racelbl    Race
msp             byte    %8.0g                 1 if married, spouse present
nev_mar         byte    %8.0g                 1 if never married
grade           byte    %8.0g                 Current grade completed
collgrad        byte    %8.0g                 1 if college graduate
not_smsa        byte    %8.0g                 1 if not SMSA
c_city          byte    %8.0g                 1 if central city
south           byte    %8.0g                 1 if south
ind_code        byte    %8.0g                 Industry of employment
occ_code        byte    %8.0g                 Occupation
union           byte    %8.0g                 1 if union
wks_ue          byte    %8.0g                 Weeks unemployed last year
ttl_exp         float   %9.0g                 Total work experience
tenure          float   %9.0g                 Job tenure, in years
hours           int     %8.0g                 Usual hours worked
wks_work        int     %8.0g                 Weeks worked last year
ln_wage         float   %9.0g                 ln(wage/GNP deflator)
-------------------------------------------------------------------------------
Sorted by: idcode  year
```

```
. summarize
```

Variable	Obs	Mean	Std. dev.	Min	Max
idcode	28,534	2601.284	1487.359	1	5159
year	28,534	77.95865	6.383879	68	88
birth_yr	28,534	48.08509	3.012837	41	54
age	28,510	29.04511	6.700584	14	46
race	28,534	1.303392	.4822773	1	3
msp	28,518	.6029175	.4893019	0	1
nev_mar	28,518	.2296795	.4206341	0	1
grade	28,532	12.53259	2.323905	0	18
collgrad	28,534	.1680451	.3739129	0	1
not_smsa	28,526	.2824441	.4501961	0	1
c_city	28,526	.357218	.4791882	0	1
south	28,526	.4095562	.4917605	0	1
ind_code	28,193	7.692973	2.994025	1	12
occ_code	28,413	4.777672	3.065435	1	13
union	19,238	.2344319	.4236542	0	1
wks_ue	22,830	2.548095	7.294463	0	76
ttl_exp	28,534	6.215316	4.652117	0	28.88461
tenure	28,101	3.123836	3.751409	0	25.91667
hours	28,467	36.55956	9.869623	1	168
wks_work	27,831	53.98933	29.03232	0	104
ln_wage	28,534	1.674907	.4780935	0	5.263916

Many of the variables in the `nlswork` dataset are indicator variables, so we have used factor variables (see [U] **11.4.3 Factor variables**) in many of the examples in this manual. You will see terms like `c.age#c.age` or `2.race` in estimation commands. `c.age#c.age` is just age interacted with age, or age-squared, and `2.race` is just an indicator variable for black (`race` = 2).

Instead of using factor variables, you could type

```
. generate age2 = age*age
. generate black = (race==2)
```

and substitute `age2` and `black` in your estimation command for `c.age#c.age` and `2.race`, respectively.

There are advantages, however, to using factor variables. First, you do not actually have to create new variables, so the number of variables in your dataset is less.

Second, by using factor variables, we are able to take better advantage of postestimation commands. For example, if we specify the simple model

```
. xtreg ln_wage age age2, fe
```

then `age` and `age2` are completely separate variables. Stata has no idea that they are related—that one is the square of the other. Consequently, if we compute the average marginal effect of age on the log of wages,

```
. margins, dydx(age)
```

then the reported marginal effect is with respect to the `age` variable alone and not with respect to the true effect of age, which involves the coefficients on both `age` and `age2`.

If instead we fit our model using an interaction of age with itself for the square of age,

```
. xtreg ln_wage age c.age#c.age, fe
```

then Stata has a deep understanding that the coefficients age and c.age#c.age are related. After fitting this model, the marginal effect reported by margins includes the full effect of age on the log of income, including the contribution of both coefficients.

```
. margins, dydx(age)
```

There are other reasons for preferring factor variables; see [R] **margins** for examples.

For union.dta, our subset was sampled only from those with union membership information from 1970 to 1988. Our subsample is of 4,434 women. The important variables are age (16–46), grade (years of schooling completed, ranging from 0 to 18), not_smsa (28% of the person-time was spent living outside a standard metropolitan statistical area (SMSA), and south (41% of the person-time was in the South). The dataset also has variable union. Overall, 22% of the person-time is marked as time under union membership, and 44% of these women have belonged to a union.

```
. use https://www.stata-press.com/data/r19/union
(NLS Women 14-24 in 1968)

. describe

Contains data from https://www.stata-press.com/data/r19/union.dta
 Observations:        26,200                  NLS Women 14-24 in 1968
    Variables:             8                  4 May 2024 13:54
                                              (_dta has notes)
-------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
-------------------------------------------------------------------------------
idcode          int     %8.0g                 NLS ID
year            byte    %8.0g                 Interview year
age             byte    %8.0g                 Age in current year
grade           byte    %8.0g                 Current grade completed
not_smsa        byte    %8.0g                 1 if not SMSA
south           byte    %8.0g                 1 if south
union           byte    %8.0g                 1 if union
black           byte    %8.0g                 Race black
-------------------------------------------------------------------------------
Sorted by: idcode  year

. summarize

    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
      idcode |     26,200    2611.582    1484.994          1       5159
        year |     26,200    79.47137    5.965499         70         88
         age |     26,200    30.43221    6.489056         16         46
       grade |     26,200    12.76145    2.411715          0         18
    not_smsa |     26,200    .2837023    .4508027          0          1
-------------+---------------------------------------------------------
       south |     26,200    .4130153    .4923849          0          1
       union |     26,200    .2217939    .4154611          0          1
       black |     26,200     .274542    .4462917          0          1
```

In many of the examples where the union dataset is used, we also include an interaction between the year variable and the south variable—south#c.year. This interaction is created using factor-variables notation; see [U] **11.4.3 Factor variables**.

With both datasets, we have typed

```
. xtset idcode year
```

❑

❑ Technical note

The `xtset` command sets the t and i index for xt data by declaring them as characteristics of the data; see [P] **char**. The panel variable is stored in `_dta[iis]` and the time variable is stored in `_dta[tis]`.

❑

❑ Technical note

Throughout the entries in [XT], when random-effects models are fit, a likelihood-ratio test that the variance of the random effects is zero is included. These tests occur on the boundary of the parameter space, invalidating the usual theory associated with such tests. However, these likelihood-ratio tests have been modified to be valid on the boundary. In particular, the null distribution of the likelihood-ratio test statistic is not the usual χ_1^2 but is rather a 50:50 mixture of a χ_0^2 (point mass at zero) and a χ_1^2, denoted as $\overline{\chi}_{01}^2$. See Gutierrez, Carter, and Drukker (2001) for a full discussion.

❑

References

Andreß, H.-J., K. Golsch, and A. W. Schmidt. 2013. *Applied Panel Data Analysis for Economic and Social Surveys*. Berlin: Springer.

Baltagi, B. H. 2013. *Econometric Analysis of Panel Data*. 5th ed. Chichester, UK: Wiley.

Cameron, A. C., and P. K. Trivedi. 2022. *Microeconometrics Using Stata*. 2nd ed. College Station, TX: Stata Press.

Center for Human Resource Research. 1989. *National Longitudinal Survey of Labor Market Experience, Young Women 14–24 years of age in 1968*. Columbus, OH: Ohio State University Press.

Frees, E. W. 2004. *Longitudinal and Panel Data: Analysis and Applications in the Social Sciences*. Cambridge: Cambridge University Press.

Greene, W. H. 2018. *Econometric Analysis*. 8th ed. New York: Pearson.

Gutierrez, R. G., S. L. Carter, and D. M. Drukker. 2001. sg160: On boundary-value likelihood-ratio tests. *Stata Technical Bulletin* 60: 15–18. Reprinted in *Stata Technical Bulletin Reprints*, vol. 10, pp. 269–273. College Station, TX: Stata Press.

Hsiao, C. 2014. *Analysis of Panel Data*. 3rd ed. New York: Cambridge University Press.

Sun, Y., and G. Dhaene. 2019. xtspj: A command for split-panel jackknife estimation. *Stata Journal* 19: 335–374.

Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

Also see

[XT] **xtset** — Declare data to be panel data

quadchk — Check sensitivity of quadrature approximation

Description Quick start Menu Syntax
Options Remarks and examples

Description

quadchk checks the quadrature approximation used in the random-effects estimators of the following commands:

xtcloglog
xtintreg
xtlogit
xtmlogit
xtologit
xtoprobit
xtpoisson, re with the normal option
xtprobit
xtstreg
xttobit

quadchk refits the model for different numbers of quadrature points and then compares the different solutions. quadchk respects all options supplied to the original model except or, vce(), and the *maximize_options*.

Quick start

Check quadrature approximation using the default range of quadrature points

```
quadchk
```

Same as above, but use 8 and 16 quadrature points

```
quadchk 8 16
```

Same as above, but suppress the iteration log and output of the refitted models

```
quadchk 8 16, nooutput
```

Refit the model instead of using original estimates

```
quadchk 8 16, nooutput nofrom
```

Menu

Statistics > Longitudinal/panel data > Setup and utilities > Check sensitivity of quadrature approximation

Syntax

quadchk [$\#_1$ $\#_2$] [, nooutput nofrom]

$\#_1$ and $\#_2$ specify the number of quadrature points to use in the comparison runs of the previous model. The default is to use approximately $2n_q/3$ and $4n_q/3$ points, where n_q is the number of quadrature points used in the original estimation.

Options

nooutput suppresses the iteration log and output of the refitted models.

nofrom forces the refitted models to start from scratch rather than starting from the previous estimation results. Specifying the nofrom option can level the playing field in testing estimation results.

Remarks and examples

Remarks are presented under the following headings:

What makes a good random-effects model fit?
How do I know whether I have a good quadrature approximation?
What can I do to improve my results?

What makes a good random-effects model fit?

Some random-effects estimators in Stata use adaptive or nonadaptive Gauss–Hermite quadrature to compute the log likelihood and its derivatives. As a rule, adaptive quadrature, which is the default integration method, is much more accurate. The quadchk command provides a means to look at the numerical accuracy of either quadrature approximation. A good random-effects model fit depends on both the goodness of the quadrature approximation and the goodness of the data.

The accuracy of the quadrature approximation depends on three factors. The first and second are how many quadrature points are used and where the quadrature points fall. These two factors directly influence the accuracy of the quadrature approximation. The number of quadrature points may be specified with the intpoints() option. However, once the number of points is specified, their abscissas (locations) and corresponding weights are completely determined. Increasing the number of points expands the range of the abscissas and, to a lesser extent, increases the density of the abscissas. For this reason, a function that undulates between the abscissas can be difficult to approximate.

Third, the smoothness of the function being approximated influences the accuracy of the quadrature approximation. Gauss–Hermite quadrature estimates integrals of the type

$$\int_{-\infty}^{\infty} e^{-x^2} f(x)dx$$

and the approximation is exact if $f(x)$ is a polynomial of degree less than the number of integration points. Therefore, $f(x)$ that are well approximated by polynomials of a given degree have integrals that are well approximated by Gauss–Hermite quadrature with that given number of integration points. Both large panel sizes and high ρ can reduce the accuracy of the quadrature approximation.

A final factor affects the goodness of the random-effects model: the data themselves. For high ρ, for example, there is high intrapanel correlation, and panels look like observations. The model becomes unidentified. Here, even with exact quadrature, fitting the model would be difficult.

How do I know whether I have a good quadrature approximation?

quadchk is intended as a tool to help you know whether you have a good quadrature approximation. As a rule of thumb, if the coefficients do not change by more than a relative difference of 10^{-4} (0.01%), the choice of quadrature points does not significantly affect the outcome, and the results may be confidently interpreted. However, if the results do change appreciably—greater than a relative difference of 10^{-2} (1%)—then quadrature is not reliably approximating the likelihood.

What can I do to improve my results?

If the quadchk command indicates that the estimation results are sensitive to the number of quadrature points, there are several things you can do. First, if you are not using adaptive quadrature, switch to adaptive quadrature.

Adaptive quadrature can improve the approximation by transforming the integrand so that the abscissas and weights sample the function on a more suitable range. Details of this transformation are in *Methods and formulas* for the given commands; for example, see [XT] **xtprobit**.

If the model still shows sensitivity to the number of quadrature points, increase the number of quadrature points with the intpoints() option. This option will increase the range and density of the sampling used for the quadrature approximation.

If neither of these works, you may then want to consider an alternative model, such as a fixed-effects, pooled, or population-averaged model. Alternatively, a different random-effects model whose likelihood is not approximated via quadrature (for example, xtpoisson, re) may be a better choice.

▷ Example 1

Here we synthesize data according to the model

$$E(y) = 0.05\,x_1 + 0.08\,x_2 + 0.08\,x_3 + 0.1\,x_4 + 0.1\,x_5 + 0.1\,x_6 + 0.1\epsilon$$

$$z = \begin{cases} 1 & \text{if } y \geq 0 \\ 0 & \text{if } y < 0 \end{cases}$$

where the intrapanel correlation is 0.5 and the x1 variable is constant within panels. We first fit a random-effects probit model, and then we check the stability of the quadrature calculation:

```
. use https://www.stata-press.com/data/r19/quad1

. xtset id

Panel variable: id (balanced)

. xtprobit z x1-x6
  (output omitted)
Random-effects probit regression                     Number of obs    =  6,000
Group variable: id                                   Number of groups =    300

Random effects u_i ~ Gaussian                        Obs per group:
                                                                  min =     20
                                                                  avg =   20.0
                                                                  max =     20

Integration method: mvaghermite                      Integration pts. =     12

                                                     Wald chi2(6)     =  29.24
Log likelihood = -3347.1097                          Prob > chi2      = 0.0001

------------------------------------------------------------------------------
           z | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
          x1 |   .0043068   .0607058     0.07   0.943    -.1146743    .1232879
          x2 |   .1000742    .066331     1.51   0.131    -.0299323    .2300806
          x3 |   .1503539   .0662503     2.27   0.023     .0205057    .2802021
          x4 |    .123015   .0377089     3.26   0.001     .0491069     .196923
          x5 |   .1342988   .0657222     2.04   0.041     .0054856     .263112
          x6 |   .0879933   .0455753     1.93   0.054    -.0013325    .1773192
       _cons |   .0757067    .060359     1.25   0.210    -.0425948    .1940083
-------------+----------------------------------------------------------------
    /lnsig2u |  -.0329916   .1026847                      -.23425    .1682667
-------------+----------------------------------------------------------------
     sigma_u |   .9836395   .0505024                      .889474    1.087774
         rho |   .4917528   .0256642                     .4417038    .5419677
------------------------------------------------------------------------------
LR test of rho=0: chibar2(01) = 1582.67                Prob >= chibar2 = 0.000
```

```
. quadchk
Refitting model intpoints() =  8
  (output omitted)
Refitting model intpoints() = 16
  (output omitted)
                                Quadrature check

                     Fitted        Comparison     Comparison
                   quadrature      quadrature     quadrature
                   12 points        8 points      16 points

Log                -3347.1097      -3347.1153     -3347.1099
likelihood                         -.00561484     -.00014288   Difference
                                    1.678e-06      4.269e-08   Relative difference

z:                  .0043068        .0043068       .00430541
      x1                            2.300e-12     -1.388e-06   Difference
                                    5.340e-10     -.00032222   Relative difference

z:                 .10007418       .10007418      .10007431
      x2                            6.513e-13      1.362e-07   Difference
                                    6.508e-12      1.361e-06   Relative difference

z:                 .15035391       .15035391      .15035406
      x3                            1.625e-12      1.520e-07   Difference
                                    1.080e-11      1.011e-06   Relative difference

z:                 .12301495       .12301495      .12301506
      x4                            1.059e-12      1.099e-07   Difference
                                    8.611e-12      8.931e-07   Relative difference

z:                 .13429881       .13429881      .13429896
      x5                            1.257e-12      1.471e-07   Difference
                                    9.361e-12      1.096e-06   Relative difference

z:                 .08799332       .08799332      .08799346
      x6                            8.576e-13      1.363e-07   Difference
                                    9.746e-12      1.549e-06   Relative difference

z:                 .07570675       .07570675      .07570423
   _cons                            5.024e-12     -2.516e-06   Difference
                                    6.636e-11     -.00003323   Relative difference

/:                -.03299164      -.03299164     -.03298184
 lnsig2u                            1.861e-11      9.798e-06   Difference
                                   -5.640e-10     -.00029699   Relative difference
```

We see that the largest difference is in the x1 variable with a relative difference of 0.03% between the model with 12 integration points and 16. This example is somewhat rare in that the differences between eight quadrature points and 12 are smaller than those between 12 and 16. Usually the opposite occurs: the model results converge as you add quadrature points. Here we have an indication that perhaps some minor feature of the model was missed with eight points and 12 but seen with 16. Because all differences are very small, we could accept this model as is. We would like to have a largest relative difference of about 0.01%, and this is close. The differences and relative differences are small, indicating that refitting the random-effects probit model with a few more integration points will yield a satisfactory result. Indeed, refitting the model with the `intpoints(20)` option yields completely satisfactory results when checked with `quadchk`.

Nonadaptive Gauss–Hermite quadrature does not yield such robust results.

```
. xtprobit z x1-x6, intmethod(ghermite) nolog

Random-effects probit regression                    Number of obs     =  6,000
Group variable: id                                  Number of groups =    300

Random effects u_i ~ Gaussian                       Obs per group:
                                                                 min =     20
                                                                 avg =   20.0
                                                                 max =     20

Integration method: ghermite                        Integration pts. =     12

                                                    Wald chi2(6)      =  36.15
Log likelihood = -3349.6926                         Prob > chi2       = 0.0000

------------------------------------------------------------------------------
           z | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
          x1 |   .1156763   .0554925     2.08   0.037     .0069131    .2244396
          x2 |   .1005555    .066227     1.52   0.129    -.0292469     .230358
          x3 |   .1542187   .0660852     2.33   0.020     .0246941    .2837433
          x4 |   .1257616   .0375776     3.35   0.001     .0521108    .1994123
          x5 |   .1366003   .0654696     2.09   0.037     .0082823    .2649182
          x6 |   .0870325   .0453489     1.92   0.055    -.0018497    .1759147
       _cons |   .1098393   .0500514     2.19   0.028     .0117404    .2079382
-------------+----------------------------------------------------------------
    /lnsig2u |  -.0791821   .0971063                     -.2695071    .1111428
-------------+----------------------------------------------------------------
     sigma_u |   .9611824   .0466685                      .8739313    1.057145
         rho |   .4802148   .0242386                      .4330281    .5277571
------------------------------------------------------------------------------
LR test of rho=0: chibar2(01) = 1577.50               Prob >= chibar2 = 0.000
```

```
. quadchk, nooutput
Refitting model intpoints() =  8
Refitting model intpoints() = 16
                              Quadrature check

                   Fitted      Comparison     Comparison
                 quadrature    quadrature     quadrature
                 12 points     8 points       16 points
─────────────────────────────────────────────────────────
Log              -3349.6926    -3354.6372     -3348.3881
likelihood                     -4.9446636      1.3045063    Difference
                                .00147615     -.00038944    Relative difference
─────────────────────────────────────────────────────────
z:                .11567633     .16153998      .07007833
      x1                        .04586365      -.045598     Difference
                                .39648262     -.39418608    Relative difference
─────────────────────────────────────────────────────────
z:                .10055552     .10317831      .09937417
      x2                        .00262279     -.00118135    Difference
                                .02608297     -.01174825    Relative difference
─────────────────────────────────────────────────────────
z:                 .1542187     .15465369      .15150516
      x3                        .00043499     -.00271354    Difference
                                .00282062      -.0175954    Relative difference
─────────────────────────────────────────────────────────
z:                .12576159     .12880254       .1243974
      x4                        .00304096     -.00136418    Difference
                                .02418032     -.01084739    Relative difference
─────────────────────────────────────────────────────────
z:                .13660028     .13475211      .13707075
      x5                       -.00184817      .00047047    Difference
                               -.01352978      .00344411    Relative difference
─────────────────────────────────────────────────────────
z:                .08703252     .08568342      .08738135
      x6                        -.0013491      .00034883    Difference
                                -.0155011      .00400809    Relative difference
─────────────────────────────────────────────────────────
z:                .10983928     .11031299      .09654975
   _cons                        .00047371     -.01328953    Difference
                                .00431274     -.12099067    Relative difference
─────────────────────────────────────────────────────────
/:               -.07918212    -.18133821     -.05815644
 lnsig2u                       -.10215609      .02102568    Difference
                                1.2901408     -.26553572    Relative difference
─────────────────────────────────────────────────────────
```

Here we see that the x1 variable (the one that was constant within panel) changed with a relative difference of nearly 40%! This example clearly demonstrates the benefit of adaptive quadrature methods.

◁

▷ Example 2

Here we rerun the previous nonadaptive quadrature model, but using the `intpoints(120)` option to increase the number of integration points to 120. We get results close to those from adaptive quadrature and an acceptable `quadchk`. This example demonstrates the efficacy of increasing the number of integration points to improve the quadrature approximation.

```
. xtprobit z x1-x6, intmethod(ghermite) intpoints(120) nolog
Random-effects probit regression                    Number of obs     =  6,000
Group variable: id                                  Number of groups =    300

Random effects u_i ~ Gaussian                       Obs per group:
                                                                 min =     20
                                                                 avg =   20.0
                                                                 max =     20

Integration method: ghermite                        Integration pts.  =    120

                                                    Wald chi2(6)      =  29.24
Log likelihood = -3347.1099                         Prob > chi2       = 0.0001

------------------------------------------------------------------------------
           z | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
          x1 |   .0043059   .0607087     0.07   0.943     -.114681    .1232929
          x2 |   .1000743   .0663311     1.51   0.131    -.0299322    .2300808
          x3 |   .1503541   .0662503     2.27   0.023     .0205058    .2802023
          x4 |   .1230151   .0377089     3.26   0.001      .049107    .1969232
          x5 |    .134299   .0657223     2.04   0.041     .0054856    .2631123
          x6 |   .0879935   .0455753     1.93   0.054    -.0013325    .1773194
       _cons |   .0757054   .0603621     1.25   0.210    -.0426021    .1940128
-------------+----------------------------------------------------------------
    /lnsig2u |  -.0329832   .1026863                     -.2342446    .1682783
-------------+----------------------------------------------------------------
     sigma_u |   .9836437   .0505034                      .8894764     1.08778
         rho |    .491755   .0256646                      .4417052    .5419706
------------------------------------------------------------------------------
LR test of rho=0: chibar2(01) = 1582.67                Prob >= chibar2 = 0.000
```

```
. quadchk, nooutput
Refitting model intpoints() = 80
Refitting model intpoints() = 160
                          Quadrature check

                  Fitted      Comparison    Comparison
              quadrature      quadrature    quadrature
              120 points       80 points    160 points

Log           -3347.1099      -3347.1099    -3347.1099
likelihood                    -.00007138     2.440e-07   Difference
                               2.133e-08    -7.289e-11   Relative difference

z:             .00430592       .00431318     .00430553
      x1                       7.259e-06    -3.871e-07   Difference
                               .00168592    -.00008991   Relative difference

z:             .10007431       .10007415     .10007431
      x2                      -1.519e-07     5.585e-09   Difference
                              -1.517e-06     5.580e-08   Relative difference

z:             .15035406       .15035407     .15035406
      x3                       1.699e-08     7.636e-09   Difference
                               1.130e-07     5.078e-08   Relative difference

z:             .12301506       .12301512     .12301506
      x4                       6.036e-08     5.353e-09   Difference
                               4.907e-07     4.352e-08   Relative difference

z:             .13429895       .13429962     .13429896
      x5                       6.646e-07     4.785e-09   Difference
                               4.949e-06     3.563e-08   Relative difference

z:             .08799345       .08799334     .08799346
      x6                      -1.123e-07     3.049e-09   Difference
                              -1.276e-06     3.465e-08   Relative difference

z:             .07570536       .07570205     .07570442
   _cons                      -3.305e-06    -9.405e-07   Difference
                              -.00004365    -.00001242   Relative difference

/:            -.03298317      -.03298909    -.03298186
 lnsig2u                      -5.919e-06     1.304e-06   Difference
                               .00017945    -.00003952   Relative difference
```

◁

▷ Example 3

Here we synthesize data the same way as in the previous example, but we make the intrapanel correlation equal to 0.1 instead of 0.5. We again fit a random-effects probit model and check the quadrature:

```
. use https://www.stata-press.com/data/r19/quad2

. xtset id

Panel variable: id (balanced)

. xtprobit z x1-x6

Fitting comparison model:

Iteration 0:  Log likelihood = -4142.2915
Iteration 1:  Log likelihood = -4120.4109
Iteration 2:  Log likelihood = -4120.4099
Iteration 3:  Log likelihood = -4120.4099

Fitting full model:

rho =  0.0    Log likelihood = -4120.4099
rho =  0.1    Log likelihood = -4065.7986
rho =  0.2    Log likelihood = -4087.7703

Iteration 0:  Log likelihood = -4065.7986
Iteration 1:  Log likelihood = -4065.3157
Iteration 2:  Log likelihood = -4065.3144
Iteration 3:  Log likelihood = -4065.3144

Random-effects probit regression                    Number of obs     =  6,000
Group variable: id                                  Number of groups =    300

Random effects u_i ~ Gaussian                       Obs per group:
                                                                 min =     20
                                                                 avg =   20.0
                                                                 max =     20

Integration method: mvaghermite                     Integration pts. =     12

                                                    Wald chi2(6)      =  39.43
Log likelihood = -4065.3144                         Prob > chi2       = 0.0000

------------------------------------------------------------------------------
           z | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
          x1 |   .0246943    .025112     0.98   0.325    -.0245243    .0739129
          x2 |   .1300123   .0587906     2.21   0.027     .0147847    .2452398
          x3 |   .1190409   .0579539     2.05   0.040     .0054533    .2326284
          x4 |    .139197   .0331817     4.19   0.000     .0741621    .2042319
          x5 |    .077364   .0578454     1.34   0.181     -.036011    .1907389
          x6 |   .0862028   .0401185     2.15   0.032      .007572    .1648336
       _cons |   .0922653   .0244392     3.78   0.000     .0443653    .1401652
-------------+----------------------------------------------------------------
    /lnsig2u |  -2.343939    .1575275                     -2.652687   -2.035191
-------------+----------------------------------------------------------------
     sigma_u |   .3097563   .0243976                      .2654461    .3614631
         rho |   .0875487   .0125839                      .0658236    .1155574
------------------------------------------------------------------------------
LR test of rho=0: chibar2(01) = 110.19                Prob >= chibar2 = 0.000
```

```
. quadchk, nooutput
Refitting model intpoints() =  8
Refitting model intpoints() = 16
                                 Quadrature check

                   Fitted        Comparison     Comparison
                 quadrature      quadrature     quadrature
                 12 points        8 points      16 points
─────────────────────────────────────────────────────────────
Log             -4065.3144      -4065.3144     -4065.3144
likelihood                      -2.268e-08      6.366e-12   Difference
                                 5.578e-12     -1.566e-15   Relative difference
─────────────────────────────────────────────────────────────
z:               .02469427       .02469427      .02469427
      x1                        -7.290e-12     -8.007e-12   Difference
                                -2.952e-10     -3.242e-10   Relative difference
─────────────────────────────────────────────────────────────
z:               .13001229       .13001229      .13001229
      x2                        -3.131e-11     -6.880e-13   Difference
                                -2.408e-10     -5.292e-12   Relative difference
─────────────────────────────────────────────────────────────
z:               .11904089       .11904089      .11904089
      x3                        -1.291e-11     -3.030e-13   Difference
                                -1.085e-10     -2.546e-12   Relative difference
─────────────────────────────────────────────────────────────
z:               .13919697       .13919697      .13919697
      x4                         2.885e-12      1.693e-13   Difference
                                 2.072e-11      1.216e-12   Relative difference
─────────────────────────────────────────────────────────────
z:               .07736398       .07736398      .07736398
      x5                        -1.160e-11     -4.557e-13   Difference
                                -1.500e-10     -5.891e-12   Relative difference
─────────────────────────────────────────────────────────────
z:               .08620282       .08620282      .08620282
      x6                         1.181e-11      3.191e-13   Difference
                                 1.370e-10      3.702e-12   Relative difference
─────────────────────────────────────────────────────────────
z:               .09226527       .09226527      .09226527
   _cons                        -5.700e-12     -1.837e-11   Difference
                                -6.177e-11     -1.991e-10   Relative difference
─────────────────────────────────────────────────────────────
/:              -2.3439389      -2.3439389     -2.3439389
 lnsig2u                        -5.892e-09     -2.172e-10   Difference
                                 2.514e-09      9.267e-11   Relative difference
─────────────────────────────────────────────────────────────
```

Here we see that the quadrature approximation is stable. With this result, we can confidently interpret the results. Satisfactory results are also obtained in this case with nonadaptive quadrature.

◁

Description | Syntax | Options | Remarks and examples
Methods and formulas | Reference | Also see

Description

This entry describes the *vce_options*, which are common to most xt estimation commands. Not all the options documented below work with all xt estimation commands; see the documentation for the particular estimation command. If an option is listed there, it is applicable.

The `vce()` option specifies how to estimate the variance–covariance matrix (VCE) corresponding to the parameter estimates. The standard errors reported in the table of parameter estimates are the square root of the variances (diagonal elements) of the VCE.

Syntax

estimation_cmd ... [, *vce_options* ...]

vce_options	Description
`vce(oim)`	observed information matrix (OIM)
`vce(opg)`	outer product of the gradient (OPG) vectors
`vce(robust)`	Huber/White/sandwich estimator
`vce(cluster` *clustvar*`)`	clustered sandwich estimator
`vce(bootstrap` [, *bootstrap_options*]`)`	bootstrap estimation
`vce(jackknife` [, *jackknife_options*]`)`	jackknife estimation
`nmp`	use divisor $N - P$ instead of the default N
`scale(x2` | `dev` | `phi` | #`)`	override the default scale parameter; available only with population-averaged models

Options

SE/Robust

`vce(oim)` is usually the default for models fit using maximum likelihood. `vce(oim)` uses the observed information matrix (OIM); see [R] **ml**.

`vce(opg)` uses the sum of the outer product of the gradient (OPG) vectors; see [R] **ml**. This is the default VCE when the `technique(bhhh)` option is specified; see [R] **Maximize**.

`vce(robust)` uses the robust or sandwich estimator of variance. This estimator is robust to some types of misspecification so long as the observations are independent; see [U] **20.22 Obtaining robust variance estimates**.

If the command allows `pweights` and you specify them, `vce(robust)` is implied; see [U] **20.24.3 Sampling weights**.

`vce(cluster` *clustvar*`)` specifies that the standard errors allow for intragroup correlation, relaxing the usual requirement that the observations be independent. That is to say, the observations are independent across groups (clusters) but not necessarily within groups. *clustvar* specifies to which group each observation belongs, for example, `vce(cluster personid)` in data with repeated observations on individuals. `vce(cluster` *clustvar*`)` affects the standard errors and variance–covariance matrix of the estimators but not the estimated coefficients; see **[U] 20.22 Obtaining robust variance estimates**.

`vce(bootstrap` [, *bootstrap_options*]`)` uses a nonparametric bootstrap; see [R] **bootstrap**. After estimation with `vce(bootstrap)`, see [R] **bootstrap postestimation** to obtain percentile-based or bias-corrected confidence intervals.

`vce(jackknife` [, *jackknife_options*]`)` uses the delete-one jackknife; see [R] **jackknife**.

`nmp` specifies that the divisor $N - P$ be used instead of the default N, where N is the total number of observations and P is the number of coefficients estimated.

`scale(x2` | `dev` | `phi` | #`)` overrides the default scale parameter. By default, `scale(1)` is assumed for the discrete distributions (binomial, negative binomial, and Poisson), and `scale(x2)` is assumed for the continuous distributions (gamma, Gaussian, and inverse Gaussian).

`scale(x2)` specifies that the scale parameter be set to the Pearson χ^2 (or generalized χ^2) statistic divided by the residual degrees of freedom, which is recommended by McCullagh and Nelder (1989) as a good general choice for continuous distributions.

`scale(dev)` sets the scale parameter to the deviance divided by the residual degrees of freedom. This option provides an alternative to `scale(x2)` for continuous distributions and for over- or underdispersed discrete distributions.

`scale(phi)` specifies that the scale parameter be estimated from the data. `xtgee`'s default scaling makes results agree with other estimators and has been recommended by McCullagh and Nelder (1989) in the context of GLM. When comparing results with calculations made by other software, you may find that the other packages do not offer this feature. In such cases, specifying `scale(phi)` should match their results.

`scale(#)` sets the scale parameter to #. For example, using `scale(1)` in `family(gamma)` models results in exponential-errors regression (if you assume independent correlation structure).

Remarks and examples

When you are working with panel-data models, we strongly encourage you to use the `vce(bootstrap)` or `vce(jackknife)` option instead of the corresponding prefix command. For example, to obtain jackknife standard errors with `xtlogit`, type

```
. use https://www.stata-press.com/data/r19/clogitid
. xtlogit y x1 x2, fe vce(jackknife)
(running xtlogit on estimation sample)
Jackknife replications (66): .........10.........20.........30.........40.......
> ..50.........60...... done
Conditional fixed-effects logistic regression        Number of obs     =    369
                                                     Replications      =     66
Group variable: id                                   Number of groups =     66
                                                     Obs per group:
                                                                  min =      2
                                                                  avg =    5.6
                                                                  max =     10
                                                     F(2, 65)          =   4.58
Log likelihood = -123.41386                          Prob > F          = 0.0137
                                     (Replications based on 66 clusters in id)

             |              Jackknife
           y | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
          x1 |     .653363   .3010608    2.17   0.034      .052103    1.254623
          x2 |    .0659169   .0487858    1.35   0.181    -.0315151    .1633489
```

If you wish to specify more options to the bootstrap or jackknife estimation, you can include them within the `vce()` option. Below we refit our model requesting bootstrap standard errors based on 300 replications, we set the random-number seed so that our results can be reproduced, and we suppress the display of the replication dots.

```
. xtlogit y x1 x2, fe vce(bootstrap, reps(300) seed(123) nodots)
Conditional fixed-effects logistic regression        Number of obs     =    369
                                                     Replications      =    300
Group variable: id                                   Number of groups =     66
                                                     Obs per group:
                                                                  min =      2
                                                                  avg =    5.6
                                                                  max =     10
                                                     Wald chi2(2)      =   9.26
Log likelihood = -123.41386                          Prob > chi2       = 0.0097
                                     (Replications based on 66 clusters in id)

             |   Observed   Bootstrap                         Normal-based
           y | coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
          x1 |     .653363    .307093    2.13   0.033     .0514717    1.255254
          x2 |    .0659169   .0477384    1.38   0.167    -.0276486    .1594824
```

❑ Technical note

To perform jackknife estimation on panel data, you must omit entire panels rather than individual observations. To replicate the output above using the `jackknife` prefix command, you would type

```
. jackknife, cluster(id): xtlogit y x1 x2, fe
```

Similarly, bootstrap estimation on panel data requires you to resample entire panels rather than individual observations. The `vce(bootstrap)` and `vce(jackknife)` options handle this for you automatically.

❑

Methods and formulas

By default, Stata's maximum likelihood estimators display standard errors based on variance estimates given by the inverse of the negative Hessian (second derivative) matrix. If `vce(robust)`, `vce(cluster` *clustvar*`)`, or `pweights` are specified, standard errors are based on the robust variance estimator (see **[U] 20.22 Obtaining robust variance estimates**); likelihood-ratio tests are not appropriate here (see **[SVY] Survey**), and the model χ^2 is from a Wald test. If `vce(opg)` is specified, the standard errors are based on the outer product of the gradients; this option has no effect on likelihood-ratio tests, though it does affect Wald tests.

If `vce(bootstrap)` or `vce(jackknife)` is specified, the standard errors are based on the chosen replication method; here the model χ^2 or F statistic is from a Wald test using the respective replication-based covariance matrix. The t distribution is used in the coefficient table when the `vce(jackknife)` option is specified. `vce(bootstrap)` and `vce(jackknife)` are also available with some commands that are not maximum likelihood estimators.

Reference

McCullagh, P., and J. A. Nelder. 1989. *Generalized Linear Models*. 2nd ed. London: Chapman and Hall/CRC.

Also see

[R] **bootstrap** — Bootstrap sampling and estimation

[R] **jackknife** — Jackknife estimation

[R] **ml** — Maximum likelihood estimation

[U] 20 Estimation and postestimation commands

xtabond — Arellano–Bond linear dynamic panel-data estimation

Description

xtabond fits a linear dynamic panel-data model where the unobserved panel-level effects are correlated with the lags of the dependent variable, known as the Arellano–Bond estimator. This estimator is designed for datasets with many panels and few periods, and it requires that there be no autocorrelation in the idiosyncratic errors.

Quick start

Arellano–Bond estimation of y on x1 and x2 using xtset data

```
xtabond y x1 x2
```

One-step estimator with robust standard errors

```
xtabond y x1 x2, vce(robust)
```

Two-step estimator with bias-corrected robust standard errors

```
xtabond y x1 x2, vce(robust) twostep
```

Arellano–Bond estimation also including 2 lagged values of y

```
xtabond y x1 x2, lags(2)
```

Menu

Statistics > Longitudinal/panel data > Dynamic panel data (DPD) > Arellano–Bond estimation

Syntax

`xtabond` *depvar* [*indepvars*] [*if*] [*in*] [, *options*]

options	Description
Model	
`noconstant`	suppress constant term
`inst(`*varlist*`)`	additional instrument variables
`lags(#)`	use # lags of dependent variable as covariates; default is `lags(1)`
`maxldep(#)`	maximum lags of dependent variable for use as instruments
`maxlags(#)`	maximum lags of predetermined and endogenous variables for use as instruments
`twostep`	compute the two-step estimator instead of the one-step estimator
Predetermined	
`pre(`*varlist*[...]`)`	predetermined variables; can be specified more than once
Endogenous	
`endogenous(`*varlist*[...]`)`	endogenous variables; can be specified more than once
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `gmm` or `robust`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`artests(#)`	use # as maximum order for AR tests; default is `artests(2)`
display_options	control spacing and line width
`coeflegend`	display legend instead of statistics

A panel variable and a time variable must be specified; use `xtset`; see [XT] **xtset**.

indepvars and all *varlists*, except `pre(`*varlist*[...]`)` and `endogenous(`*varlist*[...]`)`, may contain time-series operators; see [U] **11.4.4 Time-series varlists**. The specification of *depvar* may not contain time-series operators.

`by`, `collect`, `statsby`, and `xi` are allowed; see [U] **11.1.10 Prefix commands**.

`coeflegend` does not appear in the dialog box.

See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

`noconstant`; see [R] **Estimation options**.

`inst(`*varlist*`)` specifies a set of variables to be used as additional instruments. These instruments are not differenced by `xtabond` before including them in the instrument matrix.

`lags(#)` sets p, the number of lags of the dependent variable to be included in the model. The default is $p = 1$.

`maxldep(#)` sets the maximum number of lags of the dependent variable that can be used as instruments. The default is to use all $T_i - p - 2$ lags.

`maxlags(#)` sets the maximum number of lags of the predetermined and endogenous variables that can be used as instruments. For predetermined variables, the default is to use all $T_i - p - 1$ lags. For endogenous variables, the default is to use all $T_i - p - 2$ lags.

twostep specifies that the two-step estimator be calculated.

Predetermined

pre(*varlist* [, lagstruct(*prelags*, *premaxlags*)]) specifies that a set of predetermined variables be included in the model. Optionally, you may specify that *prelags* lags of the specified variables also be included. The default for *prelags* is 0. Specifying *premaxlags* sets the maximum number of further lags of the predetermined variables that can be used as instruments. The default is to include $T_i - p - 1$ lagged levels as instruments for predetermined variables. You may specify as many sets of predetermined variables as you need within the standard Stata limits on matrix size. Each set of predetermined variables may have its own number of *prelags* and *premaxlags*.

Endogenous

endogenous(*varlist* [, lagstruct(*endlags*, *endmaxlags*)]) specifies that a set of endogenous variables be included in the model. Optionally, you may specify that *endlags* lags of the specified variables also be included. The default for *endlags* is 0. Specifying *endmaxlags* sets the maximum number of further lags of the endogenous variables that can be used as instruments. The default is to include $T_i - p - 2$ lagged levels as instruments for endogenous variables. You may specify as many sets of endogenous variables as you need within the standard Stata limits on matrix size. Each set of endogenous variables may have its own number of *endlags* and *endmaxlags*.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory and that are robust to some kinds of misspecification; see *Remarks and examples* below.

vce(gmm), the default, uses the conventionally derived variance estimator for generalized method of moments estimation.

vce(robust) uses the robust estimator. After one-step estimation, this is the Arellano–Bond robust VCE estimator. After two-step estimation, this is the Windmeijer (2005) WC-robust estimator.

Reporting

level(#); see [R] **Estimation options**.

artests(#) specifies the maximum order of the autocorrelation test to be calculated. The tests are reported by estat abond; see [XT] **xtabond postestimation**. Specifying the order of the highest test at estimation time is more efficient than specifying it to estat abond, because estat abond must refit the model to obtain the test statistics. The maximum order must be less than or equal to the number of periods in the longest panel. The default is artests(2).

display_options: vsquish and nolstretch; see [R] **Estimation options**.

The following option is available with xtabond but is not shown in the dialog box:

coeflegend; see [R] **Estimation options**.

Remarks and examples

Linear dynamic panel-data models include p lags of the dependent variable as covariates and contain unobserved panel-level effects, fixed or random. By construction, the unobserved panel-level effects are correlated with the lagged dependent variables, making standard estimators inconsistent. Arellano and Bond (1991) derived a consistent generalized method of moments (GMM) estimator for the parameters of this model; `xtabond` implements this estimator.

Anderson and Hsiao (1981, 1982) propose using further lags of the level or the difference of the dependent variable to instrument the lagged dependent variables that are included in a dynamic panel-data model after the panel-level effects have been removed by first-differencing. A version of this estimator can be obtained from `xtivreg` (see [XT] **xtivreg**). Arellano and Bond (1991) build upon this idea by noting that, in general, there are many more instruments available. Building on Holtz-Eakin, Newey, and Rosen (1988) and using the GMM framework developed by Hansen (1982), they identify how many lags of the dependent variable, the predetermined variables, and the endogenous variables are valid instruments and how to combine these lagged levels with first differences of the strictly exogenous variables into a potentially large instrument matrix. Using this instrument matrix, Arellano and Bond (1991) derive the corresponding one-step and two-step GMM estimators, as well as the robust VCE estimator for the one-step model. They also found that the robust two-step VCE was seriously biased. Windmeijer (2005) worked out a bias-corrected (WC) robust estimator for VCEs of two-step GMM estimators, which is implemented in `xtabond`. The test of autocorrelation of order m and the Sargan test of overidentifying restrictions derived by Arellano and Bond (1991) can be obtained with `estat abond` and `estat sargan`, respectively; see [XT] **xtabond postestimation**.

The Arellano–Bond estimator is designed for datasets with many panels and few periods, and it requires that there be no autocorrelation in the idiosyncratic errors. For a related estimator that uses additional moment conditions, but still requires no autocorrelation in the idiosyncratic errors, see [XT] **xtdpdsys**. For estimators that allow for some autocorrelation in the idiosyncratic errors, at the cost of a more complicated syntax, see [XT] **xtdpd**.

▷ Example 1: One-step estimator

Arellano and Bond (1991) apply their new estimators and test statistics to a model of dynamic labor demand that had previously been considered by Layard and Nickell (1986) using data from an unbalanced panel of firms from the United Kingdom. All variables are indexed over the firm i and time t. In this dataset, $\texttt{n}_{it}$ is the log of employment in firm i at time t, $\texttt{w}_{it}$ is the natural log of the real product wage, $\texttt{k}_{it}$ is the natural log of the gross capital stock, and $\texttt{ys}_{it}$ is the natural log of industry output. The model also includes time dummies `yr1980`, `yr1981`, `yr1982`, `yr1983`, and `yr1984`. In table 4 of Arellano and Bond (1991), the authors present the results they obtained from several specifications.

In column a1 of table 4, Arellano and Bond report the coefficients and their standard errors from the robust one-step estimators of a dynamic model of labor demand in which $\texttt{n}_{it}$ is the dependent variable and its first two lags are included as regressors. To clarify some important issues, we will begin with the homoskedastic one-step version of this model and then consider the robust case. Here is the command using `xtabond` and the subsequent output for the homoskedastic case:

```
. use https://www.stata-press.com/data/r19/abdata
. xtabond n l(0/1).w l(0/2).(k ys) yr1980-yr1984 year, lags(2) noconstant
Arellano-Bond dynamic panel-data estimation     Number of obs     =        611
Group variable: id                              Number of groups  =        140
Time variable: year
                                                Obs per group:
                                                              min =          4
                                                              avg =   4.364286
                                                              max =          6
Number of instruments =     41                  Wald chi2(16)     =    1757.07
                                                Prob > chi2       =     0.0000
One-step results
------------------------------------------------------------------------------
           n | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
           n |
         L1. |   .6862261   .1486163     4.62   0.000     .3949435    .9775088
         L2. |  -.0853582   .0444365    -1.92   0.055    -.1724523    .0017358
             |
           w |
         --. |  -.6078208   .0657694    -9.24   0.000    -.7367265   -.4789151
         L1. |   .3926237   .1092374     3.59   0.000     .1785222    .6067251
             |
           k |
         --. |   .3568456   .0370314     9.64   0.000     .2842653    .4294259
         L1. |  -.0580012   .0583051    -0.99   0.320     -.172277    .0562747
         L2. |  -.0199475   .0416274    -0.48   0.632    -.1015357    .0616408
             |
          ys |
         --. |   .6085073   .1345412     4.52   0.000     .3448115    .8722031
         L1. |  -.7111651   .1844599    -3.86   0.000      -1.0727   -.3496304
         L2. |   .1057969   .1428568     0.74   0.459    -.1741974    .3857912
             |
      yr1980 |   .0029062   .0212705     0.14   0.891    -.0387832    .0445957
      yr1981 |  -.0404378   .0354707    -1.14   0.254    -.1099591    .0290836
      yr1982 |  -.0652767    .048209    -1.35   0.176    -.1597646    .0292111
      yr1983 |  -.0690928   .0627354    -1.10   0.271    -.1920521    .0538664
      yr1984 |  -.0650302   .0781322    -0.83   0.405    -.2181665    .0881061
        year |   .0095545   .0142073     0.67   0.501    -.0182912    .0374002
------------------------------------------------------------------------------
Instruments for differenced equation
        GMM-type: L(2/.).n
        Standard: D.w LD.w D.k LD.k L2D.k D.ys LD.ys L2D.ys D.yr1980
                  D.yr1981 D.yr1982 D.yr1983 D.yr1984 D.year
```

The coefficients are identical to those reported in column a1 of table 4, as they should be. Of course, the standard errors are different because we are considering the homoskedastic case. Although the moment conditions use first-differenced errors, `xtabond` estimates the coefficients of the level model and reports them accordingly.

The footer in the output reports the instruments used. The first line indicates that `xtabond` used lags from 2 on back to create the GMM-type instruments described in Arellano and Bond (1991) and Holtz-Eakin, Newey, and Rosen (1988); also see *Methods and formulas* in [XT] **xtdpd**. The second and third lines indicate that the first difference of all the exogenous variables were used as standard instruments. GMM-type instruments use the lags of a variable to contribute multiple columns to the instrument matrix, whereas each standard instrument contributes one column to the instrument matrix. The notation `L(2/.).n` indicates that GMM-type instruments were created using lag 2 of `n` from on back. (`L(2/4).n` would indicate that GMM-type instruments were created using only lags 2, 3, and 4 of `n`.)

After `xtabond`, `estat sargan` reports the Sargan test of overidentifying restrictions.

```
. estat sargan
Sargan test of overidentifying restrictions
H0: Overidentifying restrictions are valid

        chi2(25)     =  65.81806
        Prob > chi2  =    0.0000
```

Only for a homoskedastic error term does the Sargan test have an asymptotic χ^2 distribution. In fact, Arellano and Bond (1991) show that the one-step Sargan test overrejects in the presence of heteroskedasticity. Because its asymptotic distribution is not known under the assumptions of the `vce(robust)` model, `xtabond` does not compute it when `vce(robust)` is specified. The Sargan test, reported by Arellano and Bond (1991, table 4, column a1), comes from the one-step homoskedastic estimator and is the same as the one reported here. The output above presents strong evidence against the null hypothesis that the overidentifying restrictions are valid. Rejecting this null hypothesis implies that we need to reconsider our model or our instruments, unless we attribute the rejection to heteroskedasticity in the data-generating process. Although performing the Sargan test after the two-step estimator is an alternative, Arellano and Bond (1991) found a tendency for this test to underreject in the presence of heteroskedasticity. (See [XT] **xtdpd** for an example indicating that this rejection may be due to misspecification.)

By default, `xtabond` calculates the Arellano–Bond test for first- and second-order autocorrelation in the first-differenced errors. (Use `artests()` to compute tests for higher orders.) There are versions of this test for both the homoskedastic and the robust cases, although their values are different. Use `estat abond` to report the test results.

```
. estat abond
Arellano-Bond test for zero autocorrelation in first-differenced errors
H0: No autocorrelation

Order         z    Prob > z
----------------------------
    1   -3.9394      0.0001
    2   -.54239      0.5876
----------------------------
```

When the idiosyncratic errors are independent and identically distributed (i.i.d.), the first-differenced errors are first-order serially correlated. So, as expected, the output above presents strong evidence against the null hypothesis of zero autocorrelation in the first-differenced errors at order 1. Serial correlation in the first-differenced errors at an order higher than 1 implies that the moment conditions used by `xtabond` are not valid; see [XT] **xtdpd** for an example of an alternative estimation method. The output above presents no significant evidence of serial correlation in the first-differenced errors at order 2.

◁

▷ Example 2: A one-step estimator with robust VCE

Consider the output from the one-step robust estimator of the same model:

```
. xtabond n l(0/1).w l(0/2).(k ys) yr1980-yr1984 year, lags(2) vce(robust)
> noconstant

Arellano-Bond dynamic panel-data estimation     Number of obs     =        611
Group variable: id                              Number of groups  =        140
Time variable: year
                                                Obs per group:
                                                              min =          4
                                                              avg =   4.364286
                                                              max =          6

Number of instruments =     41                  Wald chi2(16)     =    1727.45
                                                Prob > chi2       =     0.0000

One-step results
                                     (Std. err. adjusted for clustering on id)
------------------------------------------------------------------------------
             |               Robust
           n | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
           n |
         L1. |   .6862261   .1445943     4.75   0.000     .4028266    .9696257
         L2. |  -.0853582   .0560155    -1.52   0.128    -.1951467    .0244302
             |
           w |
         --. |  -.6078208   .1782055    -3.41   0.001    -.9570972   -.2585445
         L1. |   .3926237   .1679931     2.34   0.019     .0633632    .7218842
             |
           k |
         --. |   .3568456   .0590203     6.05   0.000      .241168    .4725233
         L1. |  -.0580012   .0731797    -0.79   0.428    -.2014308    .0854284
         L2. |  -.0199475   .0327126    -0.61   0.542    -.0840631    .0441681
             |
          ys |
         --. |   .6085073   .1725313     3.53   0.000     .2703522    .9466624
         L1. |  -.7111651   .2317163    -3.07   0.002    -1.165321   -.2570095
         L2. |   .1057969   .1412021     0.75   0.454    -.1709542     .382548
             |
      yr1980 |   .0029062   .0158028     0.18   0.854    -.0280667    .0338791
      yr1981 |  -.0404378   .0280582    -1.44   0.150    -.0954307    .0145552
      yr1982 |  -.0652767   .0365451    -1.79   0.074    -.1369038    .0063503
      yr1983 |  -.0690928    .047413    -1.46   0.145    -.1620205    .0238348
      yr1984 |  -.0650302   .0576305    -1.13   0.259    -.1779839    .0479235
        year |   .0095545   .0102896     0.93   0.353    -.0106127    .0297217
------------------------------------------------------------------------------
Instruments for differenced equation
        GMM-type: L(2/.).n
        Standard: D.w LD.w D.k LD.k L2D.k D.ys LD.ys L2D.ys D.yr1980
                  D.yr1981 D.yr1982 D.yr1983 D.yr1984 D.year
```

The coefficients are the same, but now the standard errors match that reported in Arellano and Bond (1991, table 4, column a1). Most of the robust standard errors are higher than those that assume a homoskedastic error term.

The Sargan statistic cannot be calculated after requesting a robust VCE, but robust tests for serial correlation are available.

```
. estat abond

Arellano-Bond test for zero autocorrelation in first-differenced errors
H0: No autocorrelation

Order          z    Prob > z
----------------------------
    1    -3.5996     0.0003
    2    -.51603     0.6058
----------------------------
```

The value of the test for second-order autocorrelation matches those reported in Arellano and Bond (1991, table 4, column a1) and presents no evidence of model misspecification.

◁

▷ Example 3: The Wald model test

xtabond reports the Wald statistic of the null hypothesis that all the coefficients except the constant are zero. Here the null hypothesis is that all the coefficients are zero, because there is no constant in the model. In our previous example, the null hypothesis is soundly rejected. In column a1 of table 4, Arellano and Bond report a χ^2 test of the null hypothesis that all the coefficients are zero, except the time trend and the time dummies. Here is this test in Stata:

```
. test l.n l2.n w l.w k l.k l2.k ys l.ys l2.ys

 ( 1)  L.n = 0
 ( 2)  L2.n = 0
 ( 3)  w = 0
 ( 4)  L.w = 0
 ( 5)  k = 0
 ( 6)  L.k = 0
 ( 7)  L2.k = 0
 ( 8)  ys = 0
 ( 9)  L.ys = 0
 (10)  L2.ys = 0

           chi2( 10) =  408.29
         Prob > chi2 =    0.0000
```

◁

▷ Example 4: A two-step estimator with Windmeijer bias-corrected robust VCE

The two-step estimator with the Windmeijer bias-corrected robust VCE of the same model produces the following output:

```
. xtabond n l(0/1).w l(0/2).(k ys) yr1980-yr1984 year, lags(2) twostep
> vce(robust) noconstant

Arellano-Bond dynamic panel-data estimation     Number of obs     =        611
Group variable: id                              Number of groups  =        140
Time variable: year
                                                Obs per group:
                                                              min =          4
                                                              avg =   4.364286
                                                              max =          6

Number of instruments =     41                  Wald chi2(16)     =    1104.72
                                                Prob > chi2       =     0.0000
Two-step results
                                     (Std. err. adjusted for clustering on id)
------------------------------------------------------------------------------
             |              WC-robust
           n | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
           n |
         L1. |   .6287089   .1934138     3.25   0.001     .2496248    1.007793
         L2. |  -.0651882   .0450501    -1.45   0.148    -.1534847    .0231084
             |
           w |
         --. |  -.5257597   .1546107    -3.40   0.001     -.828791   -.2227284
         L1. |   .3112899   .2030006     1.53   0.125     -.086584    .7091638
             |
           k |
         --. |   .2783619   .0728019     3.82   0.000     .1356728    .4210511
         L1. |   .0140994   .0924575     0.15   0.879     -.167114    .1953129
         L2. |  -.0402484   .0432745    -0.93   0.352    -.1250649    .0445681
             |
          ys |
         --. |   .5919243   .1730916     3.42   0.001      .252671    .9311776
         L1. |  -.5659863   .2611008    -2.17   0.030    -1.077734   -.0542381
         L2. |   .1005433   .1610987     0.62   0.533    -.2152043    .4162908
             |
      yr1980 |   .0006378   .0168042     0.04   0.970    -.0322978    .0335734
      yr1981 |  -.0550044   .0313389    -1.76   0.079    -.1164275    .0064187
      yr1982 |   -.075978   .0419276    -1.81   0.070    -.1581545    .0061986
      yr1983 |  -.0740708   .0528381    -1.40   0.161    -.1776315      .02949
      yr1984 |  -.0906606   .0642615    -1.41   0.158    -.2166108    .0352896
        year |   .0112155   .0116783     0.96   0.337    -.0116735    .0341045
------------------------------------------------------------------------------
Instruments for differenced equation
        GMM-type: L(2/.).n
        Standard: D.w LD.w D.k LD.k L2D.k D.ys LD.ys L2D.ys D.yr1980
                  D.yr1981 D.yr1982 D.yr1983 D.yr1984 D.year
```

Arellano and Bond recommend against using the two-step nonrobust results for inference on the coefficients because the standard errors tend to be biased downward (see Arellano and Bond 1991 for details). The output above uses the Windmeijer bias-corrected (WC) robust VCE, which Windmeijer (2005) showed to work well. The magnitudes of several of the coefficient estimates have changed, and one even switched its sign.

The test for autocorrelation presents no evidence of model misspecification:

```
. estat abond

Arellano-Bond test for zero autocorrelation in first-differenced errors
H0: No autocorrelation

Order          z    Prob > z
─────────────────────────────
    1    -2.1255     0.0335
    2    -.35166     0.7251
─────────────────────────────
```

◁

Manuel Arellano (1957–) was born in Elda in Alicante, Spain. He earned degrees in economics from the University of Barcelona and the London School of Economics. After various posts in Oxford and London, he returned to Spain as professor of econometrics at Madrid in 1991. He is a leading expert on panel-data econometrics.

Stephen Roy Bond (1963–) earned degrees in economics from Cambridge and Oxford. Following various posts at Oxford, he now works mainly at the Institute for Fiscal Studies in London. His research interests include company taxation, dividends, and the links between financial markets, corporate control, and investment.

▷ Example 5: Including an estimator for the constant

Thus far we have been specifying the `noconstant` option to keep to the standard Arellano–Bond estimator, which uses instruments only for the difference equation. The constant estimated by `xtabond` is a constant in the level equation, and it is estimated from the level errors. The output below illustrates that including a constant in the model does not affect the other parameter estimates.

```
. xtabond n l(0/1).w l(0/2).(k ys) yr1980-yr1984 year, lags(2) twostep vce(robust)

Arellano-Bond dynamic panel-data estimation     Number of obs     =        611
Group variable: id                              Number of groups  =        140
Time variable: year
                                                Obs per group:
                                                              min =          4
                                                              avg =   4.364286
                                                              max =          6

Number of instruments =     42                  Wald chi2(16)     =    1104.72
                                                Prob > chi2       =     0.0000
Two-step results
                                        (Std. err. adjusted for clustering on id)
------------------------------------------------------------------------------
             |              WC-robust
           n | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
           n |
         L1. |   .6287089   .1934138     3.25   0.001     .2496248    1.007793
         L2. |  -.0651882   .0450501    -1.45   0.148    -.1534847    .0231084
             |
           w |
         --. |  -.5257597   .1546107    -3.40   0.001     -.828791   -.2227284
         L1. |   .3112899   .2030006     1.53   0.125     -.086584    .7091638
             |
           k |
         --. |   .2783619   .0728019     3.82   0.000     .1356728    .4210511
         L1. |   .0140994   .0924575     0.15   0.879     -.167114    .1953129
         L2. |  -.0402484   .0432745    -0.93   0.352    -.1250649    .0445681
             |
          ys |
         --. |   .5919243   .1730916     3.42   0.001      .252671    .9311776
         L1. |  -.5659863   .2611008    -2.17   0.030    -1.077734   -.0542381
         L2. |   .1005433   .1610987     0.62   0.533    -.2152043    .4162908
             |
      yr1980 |   .0006378   .0168042     0.04   0.970    -.0322978    .0335734
      yr1981 |  -.0550044   .0313389    -1.76   0.079    -.1164275    .0064187
      yr1982 |   -.075978   .0419276    -1.81   0.070    -.1581545    .0061986
      yr1983 |  -.0740708   .0528381    -1.40   0.161    -.1776315      .02949
      yr1984 |  -.0906606   .0642615    -1.41   0.158    -.2166108    .0352896
        year |   .0112155   .0116783     0.96   0.337    -.0116735    .0341045
       _cons |  -21.53725   23.23138    -0.93   0.354    -67.06992    23.99542
------------------------------------------------------------------------------
Instruments for differenced equation
        GMM-type: L(2/.).n
        Standard: D.w LD.w D.k LD.k L2D.k D.ys LD.ys L2D.ys D.yr1980
                  D.yr1981 D.yr1982 D.yr1983 D.yr1984 D.year
Instruments for level equation
        Standard: _cons
```

Including the constant does not affect the other parameter estimates because it is identified only by the level errors; see [XT] **xtdpd** for details.

◁

▷ Example 6: Including predetermined covariates

Sometimes we cannot assume strict exogeneity. Recall that a variable, x_{it}, is said to be strictly exogenous if $E[x_{it}\epsilon_{is}] = 0$ for all t and s. If $E[x_{it}\epsilon_{is}] \neq 0$ for $s < t$ but $E[x_{it}\epsilon_{is}] = 0$ for all $s \geq t$, the variable is said to be predetermined. Intuitively, if the error term at time t has some feedback on the subsequent realizations of x_{it}, x_{it} is a predetermined variable. Because unforecastable errors today might affect future changes in the real wage and in the capital stock, we might suspect that the log of the real product wage and the log of the gross capital stock are predetermined instead of strictly exogenous. Here we treat `w` and `k` as predetermined and use lagged levels as instruments.

```
. xtabond n l(0/1).ys yr1980-yr1984 year, lags(2) twostep pre(w, lag(1,.))
> pre(k, lag(2,.)) noconstant vce(robust)

Arellano-Bond dynamic panel-data estimation     Number of obs     =        611
Group variable: id                              Number of groups  =        140
Time variable: year
                                                Obs per group:
                                                              min =          4
                                                              avg =   4.364286
                                                              max =          6

Number of instruments =     83                  Wald chi2(15)     =     958.30
                                                Prob > chi2       =     0.0000
Two-step results
                                     (Std. err. adjusted for clustering on id)
```

n	Coefficient	WC-robust std. err.	z	P>|z|	[95% conf.	interval]
n						
L1.	.8580958	.1265515	6.78	0.000	.6100594	1.106132
L2.	-.081207	.0760703	-1.07	0.286	-.2303022	.0678881
w						
--.	-.6910855	.1387684	-4.98	0.000	-.9630666	-.4191044
L1.	.5961712	.1497338	3.98	0.000	.3026982	.8896441
k						
--.	.4140654	.1382788	2.99	0.003	.1430439	.6850868
L1.	-.1537048	.1220244	-1.26	0.208	-.3928681	.0854586
L2.	-.1025833	.0710886	-1.44	0.149	-.2419143	.0367477
ys						
--.	.6936392	.1728623	4.01	0.000	.3548354	1.032443
L1.	-.8773678	.2183085	-4.02	0.000	-1.305245	-.449491
yr1980	-.0072451	.017163	-0.42	0.673	-.0408839	.0263938
yr1981	-.0609608	.030207	-2.02	0.044	-.1201655	-.0017561
yr1982	-.1130369	.0454826	-2.49	0.013	-.2021812	-.0238926
yr1983	-.1335249	.0600213	-2.22	0.026	-.2511645	-.0158853
yr1984	-.1623177	.0725434	-2.24	0.025	-.3045001	-.0201352
year	.0264501	.0119329	2.22	0.027	.003062	.0498381

```
Instruments for differenced equation
        GMM-type: L(2/.).n L(1/.).L.w L(1/.).L2.k
        Standard: D.ys LD.ys D.yr1980 D.yr1981 D.yr1982 D.yr1983 D.yr1984
                  D.year
```

The footer informs us that we are now including GMM-type instruments from the first lag of `L.w` on back and from the first lag of `L2.k` on back.

◁

❏ Technical note

The above example illustrates that `xtabond` understands `pre(w, lag(1, .))` to mean that `L.w` is a predetermined variable and `pre(k, lag(2, .))` to mean that `L2.k` is a predetermined variable. This is a stricter definition than the alternative that `pre(w, lag(1, .))` means only that `w` is predetermined but includes a lag of `w` in the model and that `pre(k, lag(2, .))` means only that `k` is predetermined but includes first and second lags of `k` in the model. If you prefer the weaker definition, `xtabond` still gives you consistent estimates, but it is not using all possible instruments; see [XT] **xtdpd** for an example of how to include all possible instruments.

❏

▷ Example 7: Including endogenous covariates

We might instead suspect that `w` and `k` are endogenous in that $E[x_{it}\epsilon_{is}] \neq 0$ for $s \leq t$ but $E[x_{it}\epsilon_{is}] = 0$ for all $s > t$. By this definition, endogenous variables differ from predetermined variables only in that the former allow for correlation between the x_{it} and the ϵ_{it} at time t, whereas the latter do not. Endogenous variables are treated similarly to the lagged dependent variable. Levels of the endogenous variables lagged two or more periods can serve as instruments. In this example, we treat `w` and `k` as endogenous variables.

```
. xtabond n l(0/1).ys yr1980-yr1984 year, lags(2) twostep
> endogenous(w, lag(1,.)) endogenous(k, lag(2,.)) noconstant vce(robust)
Arellano-Bond dynamic panel-data estimation     Number of obs     =        611
Group variable: id                              Number of groups  =        140
Time variable: year
                                                Obs per group:
                                                              min =          4
                                                              avg =   4.364286
                                                              max =          6

Number of instruments =     71                  Wald chi2(15)     =     967.61
                                                Prob > chi2       =     0.0000
Two-step results
                                     (Std. err. adjusted for clustering on id)

                          WC-robust
           n  Coefficient  std. err.      z    P>|z|     [95% conf. interval]

           n
         L1.    .6640937   .1278908     5.19   0.000     .4134323     .914755
         L2.    -.041283    .081801    -0.50   0.614    -.2016101    .1190441

           w
         --.   -.7143942     .13083    -5.46   0.000    -.9708162   -.4579721
         L1.    .3644198    .184758     1.97   0.049     .0023008    .7265388

           k
         --.    .5028874   .1205419     4.17   0.000     .2666296    .7391452
         L1.   -.2160842   .0972855    -2.22   0.026    -.4067603    -.025408
         L2.   -.0549654   .0793673    -0.69   0.489    -.2105225    .1005917

          ys
         --.    .5989356   .1779731     3.37   0.001     .2501148    .9477564
         L1.   -.6770367   .1961166    -3.45   0.001    -1.061418   -.2926553

      yr1980   -.0061122   .0155287    -0.39   0.694    -.0365478    .0243235
      yr1981     -.04715   .0298348    -1.58   0.114    -.1056252    .0113251
      yr1982   -.0817646   .0486049    -1.68   0.093    -.1770285    .0134993
      yr1983   -.0939251   .0675804    -1.39   0.165    -.2263802    .0385299
      yr1984    -.117228   .0804716    -1.46   0.145    -.2749493    .0404934
        year    .0208857   .0103485     2.02   0.044     .0006031    .0411684

Instruments for differenced equation
        GMM-type: L(2/.).n L(2/.).L.w L(2/.).L2.k
        Standard: D.ys LD.ys D.yr1980 D.yr1981 D.yr1982 D.yr1983 D.yr1984
                  D.year
```

Although some estimated coefficients changed in magnitude, none changed in sign, and these results are similar to those obtained by treating w and k as predetermined.

◁

The Arellano–Bond estimator is for datasets with many panels and few periods. (Technically, the large-sample properties are derived with the number of panels going to infinity and the number of periods held fixed.) The number of instruments increases quadratically in the number of periods. If your dataset is better described by a framework in which both the number of panels and the number of periods is large, then you should consider other estimators such as those in [XT] **xtivreg** or xtreg, fe in [XT] **xtreg**; see Alvarez and Arellano (2003) for a discussion of this case.

▷ Example 8: Restricting the number of instruments

Treating variables as predetermined or endogenous quickly increases the size of the instrument matrix. (See *Methods and formulas* in [XT] **xtdpd** for a discussion of how this matrix is created and what determines its size.) GMM estimators with too many overidentifying restrictions may perform poorly in small samples. (See Kiviet 1995 for a discussion of the dynamic panel-data case.)

To handle these problems, you can set a maximum number of lagged levels to be included as instruments for lagged-dependent or the predetermined variables. Here is an example in which a maximum of three lagged levels of the predetermined variables are included as instruments:

```
. xtabond n l(0/1).ys yr1980-yr1984 year, lags(2) twostep
> pre(w, lag(1,3)) pre(k, lag(2,3)) noconstant vce(robust)

Arellano-Bond dynamic panel-data estimation     Number of obs     =        611
Group variable: id                              Number of groups  =        140
Time variable: year
                                                Obs per group:
                                                              min =          4
                                                              avg =   4.364286
                                                              max =          6

Number of instruments =     67                  Wald chi2(15)     =    1116.89
                                                Prob > chi2       =     0.0000
Two-step results
                                      (Std. err. adjusted for clustering on id)
```

n	Coefficient	WC-robust std. err.	z	P>|z|	[95% conf.	interval]
n						
L1.	.931121	.1456964	6.39	0.000	.6455612	1.216681
L2.	-.0759918	.0854356	-0.89	0.374	-.2434425	.0914589
w						
--.	-.6475372	.1687931	-3.84	0.000	-.9783656	-.3167089
L1.	.6906238	.1789698	3.86	0.000	.3398493	1.041398
k						
--.	.3788106	.1848137	2.05	0.040	.0165824	.7410389
L1.	-.2158533	.1446198	-1.49	0.136	-.4993028	.0675962
L2.	-.0914584	.0852267	-1.07	0.283	-.2584997	.0755829
ys						
--.	.7324964	.176748	4.14	0.000	.3860766	1.078916
L1.	-.9428141	.2735472	-3.45	0.001	-1.478957	-.4066715
yr1980	-.0102389	.0172473	-0.59	0.553	-.0440431	.0235652
yr1981	-.0763495	.0296992	-2.57	0.010	-.1345589	-.0181402
yr1982	-.1373829	.0441833	-3.11	0.002	-.2239806	-.0507853
yr1983	-.1825149	.0613674	-2.97	0.003	-.3027928	-.0622369
yr1984	-.2314023	.0753669	-3.07	0.002	-.3791186	-.083686
year	.0310012	.0119167	2.60	0.009	.0076448	.0543576

```
Instruments for differenced equation
        GMM-type: L(2/.).n L(1/3).L.w L(1/3).L2.k
        Standard: D.ys LD.ys D.yr1980 D.yr1981 D.yr1982 D.yr1983 D.yr1984
                  D.year
```

◁

▷ Example 9: Missing observations in the middle of panels

xtabond handles data in which there are missing observations in the middle of the panels. In this example, we deliberately set the dependent variable to missing in the year 1980:

```
. replace n=. if year==1980
(140 real changes made, 140 to missing)

. xtabond n l(0/1).w l(0/2).(k ys) yr1980-yr1984 year, lags(2) noconstant
>  vce(robust)
note: yr1980 omitted from div() because of collinearity.
note: yr1981 omitted from div() because of collinearity.
note: yr1982 omitted from div() because of collinearity.
note: yr1980 omitted because of collinearity.
note: yr1981 omitted because of collinearity.
note: yr1982 omitted because of collinearity.

Arellano-Bond dynamic panel-data estimation     Number of obs     =        115
Group variable: id                              Number of groups  =        101
Time variable: year
                                                Obs per group:
                                                              min =          1
                                                              avg =   1.138614
                                                              max =          2

Number of instruments =     18                  Wald chi2(12)     =      44.48
                                                Prob > chi2       =     0.0000

One-step results
                                    (Std. err. adjusted for clustering on id)
------------------------------------------------------------------------------
             |               Robust
           n | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
           n |
         L1. |   .1790577   .2204682     0.81   0.417     -.253052    .6111674
         L2. |   .0214253   .0488476     0.44   0.661    -.0743143    .1171649
             |
           w |
         --. |  -.2513405   .1402114    -1.79   0.073    -.5261498    .0234689
         L1. |   .1983952   .1445875     1.37   0.170    -.0849912    .4817815
             |
           k |
         --. |   .3983149   .0883352     4.51   0.000     .2251811    .5714488
         L1. |   -.025125   .0909236    -0.28   0.782     -.203332    .1530821
         L2. |  -.0359338   .0623382    -0.58   0.564    -.1581144    .0862468
             |
          ys |
         --. |   .3663201   .3824893     0.96   0.338    -.3833451    1.115985
         L1. |  -.6319976   .4823958    -1.31   0.190    -1.577476    .3134807
         L2. |   .5318404   .4105269     1.30   0.195    -.2727775    1.336458
             |
      yr1983 |  -.0047543    .024855    -0.19   0.848    -.0534692    .0439606
      yr1984 |          0  (omitted)
        year |   .0014465    .010355     0.14   0.889    -.0188489    .0217419
------------------------------------------------------------------------------
Instruments for differenced equation
        GMM-type: L(2/.).n
        Standard: D.w LD.w D.k LD.k L2D.k D.ys LD.ys L2D.ys D.yr1983
                  D.yr1984 D.year
```

There are two important aspects to this example. First, xtabond reports that variables have been omitted from the model and from the div() instrument list. For xtabond, the div() instrument list is the list of instruments created from the strictly exogenous variables; see [XT] **xtdpd** for more about the div()

instrument list. Second, because `xtabond` uses time-series operators in its computations, `if` statements and missing values are not equivalent. An `if` statement causes the false observations to be excluded from the sample, but it computes the time-series operators wherever possible. In contrast, missing data prevent evaluation of the time-series operators that involve missing observations. Thus the example above is not equivalent to the following one:

```
. use https://www.stata-press.com/data/r19/abdata, clear
. xtabond n l(0/1).w l(0/2).(k ys) yr1980-yr1984 year if year!=1980,
> lags(2) noconstant vce(robust)
note: yr1980 omitted from div() because of collinearity.
note: yr1980 omitted because of collinearity.

Arellano-Bond dynamic panel-data estimation     Number of obs     =        473
Group variable: id                              Number of groups  =        140
Time variable: year
                                                Obs per group:
                                                              min =          3
                                                              avg =   3.378571
                                                              max =          5

Number of instruments =     37                  Wald chi2(15)     =    1041.61
                                                Prob > chi2       =     0.0000
One-step results
                                     (Std. err. adjusted for clustering on id)
------------------------------------------------------------------------------
             |               Robust
           n | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
           n |
         L1. |   .7210062   .1321214     5.46   0.000     .4620531    .9799593
         L2. |  -.0960646   .0570547    -1.68   0.092    -.2078898    .0157606
             |
           w |
         --. |  -.6684175   .1739484    -3.84   0.000     -1.00935   -.3274849
         L1. |    .482322   .1647185     2.93   0.003     .1594797    .8051642
             |
           k |
         --. |   .3802777   .0728546     5.22   0.000     .2374853    .5230701
         L1. |   -.104598    .088597    -1.18   0.238    -.278245     .069049
         L2. |  -.0272055   .0379994    -0.72   0.474    -.101683     .0472721
             |
          ys |
         --. |   .4655989   .1864368     2.50   0.013     .1001895    .8310082
         L1. |  -.8562492   .2187886    -3.91   0.000    -1.285067   -.4274315
         L2. |   .0896556   .1440035     0.62   0.534    -.192586     .3718972
             |
      yr1981 |  -.0711626   .0205299    -3.47   0.001    -.1114005   -.0309247
      yr1982 |  -.1212749   .0334659    -3.62   0.000    -.1868669   -.0556829
      yr1983 |  -.1470248   .0461714    -3.18   0.001    -.2375191   -.0565305
      yr1984 |  -.1519021   .0543904    -2.79   0.005    -.2585054   -.0452988
        year |   .0203277   .0108732     1.87   0.062    -.0009833    .0416387
------------------------------------------------------------------------------
Instruments for differenced equation
        GMM-type: L(2/.).n
        Standard: D.w LD.w D.k LD.k L2D.k D.ys LD.ys L2D.ys D.yr1981
                  D.yr1982 D.yr1983 D.yr1984 D.year
```

The year 1980 is omitted from the sample, but when the value of a variable from 1980 is required because a lag or difference is required, the 1980 value is used.

◁

Stored results

`xtabond` stores the following in `e()`:

Scalars	
`e(N)`	number of observations
`e(N_g)`	number of groups
`e(df_m)`	model degrees of freedom
`e(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(g_max)`	largest group size
`e(t_min)`	minimum time in sample
`e(t_max)`	maximum time in sample
`e(chi2)`	χ^2
`e(arm#)`	test for autocorrelation of order #
`e(artests)`	number of AR tests computed
`e(sig2)`	estimate of σ_ϵ^2
`e(rss)`	sum of squared differenced residuals
`e(sargan)`	Sargan test statistic
`e(rank)`	rank of `e(V)`
`e(zrank)`	rank of instrument matrix
Macros	
`e(cmd)`	`xtabond`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(twostep)`	`twostep`, if specified
`e(ivar)`	variable denoting groups
`e(tvar)`	variable denoting time within groups
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. err.
`e(system)`	`system`, if system estimator
`e(transform)`	specified transform
`e(datasignature)`	checksum from `datasignature`
`e(datasignaturevars)`	variables used in calculation of checksum
`e(properties)`	`b V`
`e(estat_cmd)`	program used to implement `estat`
`e(predict)`	program used to implement `predict`
`e(marginsok)`	predictions allowed by `margins`
Matrices	
`e(b)`	coefficient vector
`e(V)`	variance–covariance matrix of the estimators
Functions	
`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices	
`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

A dynamic panel-data model has the form

$$y_{it} = \sum_{j=1}^{p} \alpha_j y_{i,t-j} + \mathbf{x}_{it}\boldsymbol{\beta}_1 + \mathbf{w}_{it}\boldsymbol{\beta}_2 + \nu_i + \epsilon_{it} \quad i = 1, \ldots, N \quad t = 1, \ldots, T_i \tag{1}$$

where

the α_j are p parameters to be estimated,
$\mathbf{x}_{it}$ is a $1 \times k_1$ vector of strictly exogenous covariates,
$\boldsymbol{\beta}_1$ is a $k_1 \times 1$ vector of parameters to be estimated,
$\mathbf{w}_{it}$ is a $1 \times k_2$ vector of predetermined and endogenous covariates,
$\boldsymbol{\beta}_2$ is a $k_2 \times 1$ vector of parameters to be estimated,
ν_i are the panel-level effects (which may be correlated with the covariates), and
ϵ_{it} are i.i.d. over the whole sample with variance σ_ϵ^2.

The ν_i and the ϵ_{it} are assumed to be independent for each i over all t.

By construction, the lagged dependent variables are correlated with the unobserved panel-level effects, making standard estimators inconsistent. With many panels and few periods, estimators are constructed by first-differencing to remove the panel-level effects and using instruments to form moment conditions.

`xtabond` uses a GMM estimator to estimate $\alpha_1, \ldots, \alpha_p$, $\boldsymbol{\beta}_1$, and β_2. The moment conditions are formed from the first-differenced errors from (1) and instruments. Lagged levels of the dependent variable, the predetermined variables, and the endogenous variables are used to form GMM-type instruments. See Arellano and Bond (1991) and Holtz-Eakin, Newey, and Rosen (1988) for discussions of GMM-type instruments. First differences of the strictly exogenous variables are used as standard instruments.

`xtabond` uses `xtdpd` to perform its computations, so the formulas are given in *Methods and formulas* of [XT] **xtdpd**.

References

Alvarez, J., and M. Arellano. 2003. The time series and cross-section asymptotics of dynamic panel data estimators. *Econometrica* 71: 1121–1159. https://doi.org/10.1111/1468-0262.00441.

Anderson, T. W., and C. Hsiao. 1981. Estimation of dynamic models with error components. *Journal of the American Statistical Association* 76: 598–606. https://doi.org/10.2307/2287517.

———. 1982. Formulation and estimation of dynamic models using panel data. *Journal of Econometrics* 18: 47–82. https://doi.org/10.1016/0304-4076(82)90095-1.

Arellano, M., and S. Bond. 1991. Some tests of specification for panel data: Monte Carlo evidence and an application to employment equations. *Review of Economic Studies* 58: 277–297. https://doi.org/10.2307/2297968.

Baltagi, B. H. 2013. *Econometric Analysis of Panel Data*. 5th ed. Chichester, UK: Wiley.

Blackburne, E. F., III, and M. W. Frank. 2007. Estimation of nonstationary heterogeneous panels. *Stata Journal* 7: 197–208.

Bruno, G. S. F. 2005. Estimation and inference in dynamic unbalanced panel-data models with a small number of individuals. *Stata Journal* 5: 473–500.

De Vos, I., G. Everaert, and I. Ruyssen. 2015. Bootstrap-based bias correction and inference for dynamic panels with fixed effects. *Stata Journal* 15: 986–1018.

Grotti, R., and G. Cutuli. 2018. xtpdyn: A community-contributed command for fitting dynamic random-effects probit models with unobserved heterogeneity. *Stata Journal* 18: 844–862.

Hansen, L. P. 1982. Large sample properties of generalized method of moments estimators. *Econometrica* 50: 1029–1054. https://doi.org/10.2307/1912775.

Holtz-Eakin, D., W. K. Newey, and H. S. Rosen. 1988. Estimating vector autoregressions with panel data. *Econometrica* 56: 1371–1395. https://doi.org/10.2307/1913103.

Kiviet, J. F. 1995. On bias, inconsistency, and efficiency of various estimators in dynamic panel data models. *Journal of Econometrics* 68: 53–78. https://doi.org/10.1016/0304-4076(94)01643-E.

Kripfganz, S. 2016. Quasi–maximum likelihood estimation of linear dynamic short-T panel-data models. *Stata Journal* 16: 1013–1038.

Layard, R., and S. J. Nickell. 1986. Unemployment in Britain. *Economica* 53: S121–S169. https://doi.org/10.2307/2554377.

Neal, T. 2014. Panel cointegration analysis with xtpedroni. *Stata Journal* 14: 684–692.

Pinzon, E. 2015. xtabond cheat sheet. *The Stata Blog: Not Elsewhere Classified.* https://blog.stata.com/2015/11/12/xtabond-cheat-sheet/.

Seo, M. H., S. Kim, and Y.-J. Kim. 2019. Estimation of dynamic panel threshold model using Stata. *Stata Journal* 19: 685–697.

Williams, R., P. D. Allison, and E. Moral-Benito. 2018. Linear dynamic panel-data estimation using maximum likelihood and structural equation modeling. *Stata Journal* 18: 293–326.

Windmeijer, F. 2005. A finite sample correction for the variance of linear efficient two-step GMM estimators. *Journal of Econometrics* 126: 25–51. https://doi.org/10.1016/j.jeconom.2004.02.005.

Also see

[XT] **xtabond postestimation** — Postestimation tools for xtabond

[XT] **xtdpd** — Linear dynamic panel-data estimation

[XT] **xtdpdsys** — Arellano–Bover/Blundell–Bond linear dynamic panel-data estimation

[XT] **xtivreg** — Instrumental variables and two-stage least squares for panel-data models

[XT] **xtreg** — Linear models for panel data

[XT] **xtregar** — Fixed- and random-effects linear models with an AR(1) disturbance

[XT] **xtset** — Declare data to be panel data

[U] 20 Estimation and postestimation commands

xtabond postestimation — Postestimation tools for xtabond

Postestimation commands   predict   margins   estat
Remarks and examples   Methods and formulas   Reference   Also see

Postestimation commands

The following postestimation commands are of special interest after `xtabond`:

Command	Description
`estat abond`	test for autocorrelation
`estat sargan`	Sargan test of overidentifying restrictions

The following standard postestimation commands are also available:

Command	Description
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
`forecast`	dynamic forecasts and simulations
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	linear predictions and their SEs, residual errors
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

predict

Description for predict

`predict` creates a new variable containing predictions such as linear predictions.

Menu for predict

Statistics > Postestimation

Syntax for predict

`predict` [*type*] *newvar* [*if*] [*in*] [, `xb` `e` `stdp` `difference`]

Options for predict

Main

`xb`, the default, calculates the linear prediction.

`e` calculates the residual error.

`stdp` calculates the standard error of the prediction, which can be thought of as the standard error of the predicted expected value or mean for the observation's covariate pattern. The standard error of the prediction is also referred to as the standard error of the fitted value. `stdp` may not be combined with `difference`.

`difference` specifies that the statistic be calculated for the first differences instead of the levels, the default.

margins

Description for margins

margins estimates margins of responses for linear predictions.

Menu for margins

Statistics > Postestimation

Syntax for margins

margins [*marginlist*] [, *options*]

margins [*marginlist*] , predict(*statistic* ...) [*options*]

statistic	Description
xb	linear prediction; the default
e	not allowed with margins
stdp	not allowed with margins

Statistics not allowed with margins are functions of stochastic quantities other than e(b).

For the full syntax, see [R] **margins**.

estat

Description for estat

estat abond reports the Arellano–Bond test for serial correlation in the first-differenced residuals.

estat sargan reports the Sargan test of the overidentifying restrictions.

Menu for estat

Statistics > Postestimation

Syntax for estat

Test for autocorrelation

estat abond [, artests(#)]

Sargan test of overidentifying restrictions

estat sargan

Option for estat abond

artests(#) specifies the highest order of serial correlation to be tested. By default, the tests computed during estimation are reported. The model will be refit when artests(#) specifies a higher order than that computed during the original estimation. The model can only be refit if the data have not changed.

Remarks and examples

Remarks are presented under the following headings:

estat abond

`estat abond` reports the Arellano–Bond test for serial correlation in the first-differenced errors at order m. Rejecting the null hypothesis of no serial correlation in the first-differenced errors at order zero does not imply model misspecification because the first-differenced errors are serially correlated if the idiosyncratic errors are independent and identically distributed. Rejecting the null hypothesis of no serial correlation in the first-differenced errors at an order greater than one implies model misspecification; see example 5 in [XT] **xtdpd** for an alternative estimator that allows for idiosyncratic errors that follow a first-order moving average process.

After the one-step system estimator, the test can be computed only when `vce(robust)` has been specified. (The system estimator is used to estimate the constant in `xtabond`.)

See *Remarks and examples* in [XT] **xtabond** for more remarks about `estat abond` that are made in the context of the examples analyzed therein.

estat sargan

The distribution of the Sargan test is known only when the errors are independent and identically distributed. For this reason, `estat sargan` does not produce a test statistic when `vce(robust)` was specified in the call to `xtabond`.

See *Remarks and examples* in [XT] **xtabond** for more remarks about `estat sargan` that are made in the context of the examples analyzed therein.

Methods and formulas

See [XT] **xtdpd postestimation** for the formulas.

Reference

Pinzon, E. 2015. xtabond cheat sheet. *The Stata Blog: Not Elsewhere Classified.* https://blog.stata.com/2015/11/12/xtabond-cheat-sheet/.

Also see

[XT] **xtabond** — Arellano–Bond linear dynamic panel-data estimation

[U] 20 Estimation and postestimation commands

xtcloglog — Random-effects and population-averaged cloglog models

Description

`xtcloglog` fits population-averaged and random-effects complementary log–log (cloglog) models for a binary dependent variable. Complementary log–log models are typically used when one of the outcomes is rare relative to the other.

Quick start

Random-effects complementary log–log regression of `y` on `x1` and `x2` using `xtset` data

```
xtcloglog y x1 x2
```

Add indicators for levels of categorical variable `a` and interact `x1` with `x2`

```
xtcloglog y x1 x2 c.x1#c.x2 i.a
```

Same as above, but suppress iteration log

```
xtcloglog y x1 x2 c.x1#c.x2 i.a, nolog
```

Population-averaged model with an exchangeable correlation structure

```
xtcloglog y x1 x2 c.x1#c.x2 i.a, pa
```

Random-effects model with bootstrap standard errors

```
xtcloglog y x1 x2 c.x1#c.x2 i.a, vce(bootstrap)
```

Population-averaged model with an autoregressive correlation structure of order 1

```
xtcloglog y x1 x2 c.x1#c.x2 i.a, pa corr(ar 1)
```

Menu

Statistics > Longitudinal/panel data > Binary outcomes > Complementary log–log regression (RE, PA)

Syntax

Random-effects (RE) model

`xtcloglog` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, `re` *RE_options*]

Population-averaged (PA) model

`xtcloglog` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] , `pa` [*PA_options*]

RE_options	Description
Model	
`noconstant`	suppress constant term
`re`	use random-effects estimator; the default
`offset(`*varname*`)`	include *varname* in model with coefficient constrained to 1
`constraints(`*constraints*`)`	apply specified linear constraints
`asis`	retain perfect predictor variables
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `robust`, `cluster` *clustvar*, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`lrmodel`	perform the likelihood-ratio model test instead of the default Wald test
`eform`	report exponentiated coefficients
`nocnsreport`	do not display constraints
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Integration	
`intmethod(`*intmethod*`)`	integration method; *intmethod* may be `mvaghermite` (the default) or `ghermite`
`intpoints(#)`	use # quadrature points; default is `intpoints(12)`
Maximization	
maximize_options	control the maximization process; seldom used
`collinear`	keep collinear variables
`coeflegend`	display legend instead of statistics

PA_options	Description
Model	
noconstant	suppress constant term
pa	use population-averaged estimator
offset(*varname*)	include *varname* in model with coefficient constrained to 1
asis	retain perfect predictor variables
Correlation	
corr(*correlation*)	within-panel correlation structure
force	estimate even if observations unequally spaced in time
SE/Robust	
vce(*vcetype*)	*vcetype* may be conventional, robust, bootstrap, or jackknife
nmp	use divisor $N - P$ instead of the default N
scale(*parm*)	overrides the default scale parameter; *parm* may be x2, dev, phi, or #
Reporting	
level(#)	set confidence level; default is level(95)
eform	report exponentiated coefficients
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Optimization	
optimize_options	control the optimization process; seldom used
coeflegend	display legend instead of statistics

correlation	Description
exchangeable	exchangeable; the default
independent	independent
unstructured	unstructured
fixed *matname*	user-specified
ar #	autoregressive of order #
stationary #	stationary of order #
nonstationary #	nonstationary of order #

A panel variable must be specified. For xtcloglog, pa, correlation structures other than exchangeable and independent require that a time variable also be specified. Use xtset; see [XT] **xtset**.

indepvars may contain factor variables; see [U] **11.4.3 Factor variables**.

by, collect, mi estimate, and statsby are allowed; see [U] **11.1.10 Prefix commands**. fp is allowed for the random-effects model.

vce(bootstrap) and vce(jackknife) are not allowed with the mi estimate prefix; see [MI] **mi estimate**.

iweights, fweights, and pweights are allowed for the population-averaged model, and iweights are allowed for the random-effects model; see [U] **11.1.6 weight**. Weights must be constant within panel.

collinear and coeflegend do not appear in the dialog box.

See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options for RE model

Model

noconstant; see [R] **Estimation options**.

re requests the random-effects estimator, which is the default.

offset(*varname*), constraints(*constraints*); see [R] **Estimation options**.

asis forces retention of perfect predictor variables and their associated, perfectly predicted observations and may produce instabilities in maximization; see [R] **probit**.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (oim), that are robust to some kinds of misspecification (robust), that allow for intragroup correlation (cluster *clustvar*), and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] ***vce_options***.

Specifying vce(robust) is equivalent to specifying vce(cluster *panelvar*); see *xtcloglog, re and the robust VCE estimator* in *Methods and formulas*.

Reporting

level(#), lrmodel; see [R] **Estimation options**.

eform displays the exponentiated coefficients and corresponding standard errors and confidence intervals.

nocnsreport; see [R] **Estimation options**.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(*%fmt*), pformat(*%fmt*), sformat(*%fmt*), and nolstretch; see [R] **Estimation options**.

Integration

intmethod(*intmethod*), intpoints(#); see [R] **Estimation options**.

Maximization

maximize_options: difficult, technique(*algorithm_spec*), iterate(#), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), nonrtolerance, and from(*init_specs*); see [R] **Maximize**. These options are seldom used.

The following options are available with xtcloglog but are not shown in the dialog box:

collinear, coeflegend; see [R] **Estimation options**.

Options for PA model

Model

noconstant; see [R] **Estimation options**.

pa requests the population-averaged estimator.

`offset(`*varname*`)`; see [R] **Estimation options**

`asis` forces retention of perfect predictor variables and their associated, perfectly predicted observations and may produce instabilities in maximization; see [R] **probit**.

Correlation

`corr(`*correlation*`)` specifies the within-panel correlation structure; the default corresponds to the equal-correlation model, `corr(exchangeable)`.

When you specify a correlation structure that requires a lag, you indicate the lag after the structure's name with or without a blank; for example, `corr(ar 1)` or `corr(ar1)`.

If you specify the fixed correlation structure, you specify the name of the matrix containing the assumed correlations following the word `fixed`, for example, `corr(fixed myr)`.

`force` specifies that estimation be forced even though the time variable is not equally spaced. This is relevant only for correlation structures that require knowledge of the time variable. These correlation structures require that observations be equally spaced so that calculations based on lags correspond to a constant time change. If you specify a time variable indicating that observations are not equally spaced, the (time dependent) model will not be fit. If you also specify `force`, the model will be fit, and it will be assumed that the lags based on the data ordered by the time variable are appropriate.

SE/Robust

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`conventional`), that are robust to some kinds of misspecification (`robust`), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [XT] ***vce_options***.

`vce(conventional)`, the default, uses the conventionally derived variance estimator for generalized least-squares regression.

`nmp`, `scale(x2 | dev | phi | #)`; see [XT] ***vce_options***.

Reporting

`level(#)`; see [R] **Estimation options**.

`eform` displays the exponentiated coefficients and corresponding standard errors and confidence intervals.

display_options: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(`*style*`)`, `cformat(%`*fmt*`)`, `pformat(%`*fmt*`)`, `sformat(%`*fmt*`)`, and `nolstretch`; see [R] **Estimation options**.

Optimization

optimize_options control the iterative optimization process. These options are seldom used.

`iterate(#)` specifies the maximum number of iterations. When the number of iterations equals #, the optimization stops and presents the current results, even if convergence has not been reached. The default is `iterate(100)`.

`tolerance(#)` specifies the tolerance for the coefficient vector. When the relative change in the coefficient vector from one iteration to the next is less than or equal to #, the optimization process is stopped. `tolerance(1e-6)` is the default.

`log` and `nolog` specify whether to display the iteration log. The iteration log is displayed by default unless you used `set iterlog off` to suppress it; see `set iterlog` in [R] ***set iter***.

`trace` specifies that the current estimates be printed at each iteration.

The following option is available with `xtcloglog` but is not shown in the dialog box:

`coeflegend`; see [R] **Estimation options**.

Remarks and examples

`xtcloglog` may be used to fit a population-averaged model or a random-effects complementary log–log (cloglog) model. There is no command for a conditional fixed-effects model, as there does not exist a sufficient statistic allowing the fixed effects to be conditioned out of the likelihood. Unconditional fixed-effects cloglog models may be fit with `cloglog` with indicator variables for the panels. However, unconditional fixed-effects estimates are biased. We do not discuss fixed-effects further in this entry.

By default, the population-averaged model is an equal-correlation model; that is, `xtcloglog, pa` assumes `corr(exchangeable)`. Thus, `xtcloglog, pa` is a shortcut command for fitting the population-averaged model using `xtgee`; see [XT] **xtgee**. Typing

```
. xtcloglog ..., pa ...
```

is equivalent to typing

```
. xtgee ..., ... family(binomial) link(cloglog) corr(exchangeable)
```

Also see [XT] **xtgee** for information about `xtcloglog`.

By default or when `re` is specified, `xtcloglog` fits, via maximum likelihood, the random-effects model

$$\Pr(y_{it} \neq 0|\mathbf{x}_{it}) = P(\mathbf{x}_{it}\boldsymbol{\beta} + \nu_i)$$

for $i = 1, \dots, n$ panels, where $t = 1, \dots, n_i$, ν_i are i.i.d., $N(0, \sigma_\nu^2)$, and $P(z) = 1 - \exp\{-\exp(z)\}$.

Underlying this model is the variance-components model

$$y_{it} \neq 0 \iff \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i + \epsilon_{it} > 0$$

where ϵ_{it} are i.i.d. extreme-value (Gumbel) distributed with the mean equal to Euler's constant and variance $\sigma_\epsilon^2 = \pi^2/6$, independently of ν_i. The nonsymmetric error distribution is an alternative to logit and probit analysis and is typically used when the positive (or negative) outcome is rare.

▷ Example 1

Suppose that we are studying unionization of women in the United States and are using the `union` dataset; see [XT] **xt**. We wish to fit a random-effects model of union membership:

```
. use https://www.stata-press.com/data/r19/union
(NLS Women 14-24 in 1968)
. xtcloglog union age grade not_smsa south##c.year
  (output omitted)
Random-effects complementary log-log model          Number of obs    = 26,200
Group variable: idcode                              Number of groups =  4,434
Random effects u_i ~ Gaussian                       Obs per group:
                                                                 min =      1
                                                                 avg =    5.9
                                                                 max =     12
Integration method: mvaghermite                     Integration pts. =     12
                                                    Wald chi2(6)     = 248.58
Log likelihood = -10535.928                         Prob > chi2      = 0.0000
```

union	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
age	.0128659	.0119004	1.08	0.280	-.0104586	.0361903
grade	.06985	.0138135	5.06	0.000	.042776	.096924
not_smsa	-.198416	.0647943	-3.06	0.002	-.3254104	-.0714215
1.south	-2.047645	.488965	-4.19	0.000	-3.005999	-1.089291
year	-.0006432	.0123569	-0.05	0.958	-.0248623	.0235759
south#c.year						
1	.0164259	.006065	2.71	0.007	.0045387	.0283132
_cons	-3.269158	.659029	-4.96	0.000	-4.560831	-1.977485
/lnsig2u	1.24128	.0461705			1.150787	1.331772
sigma_u	1.860118	.0429413			1.77783	1.946214
rho	.677778	.0100834			.6577057	.6972152

```
LR test of rho=0: chibar2(01) = 6009.36                 Prob >= chibar2 = 0.000
```

The output includes the additional panel-level variance component, which is parameterized as the log of the standard deviation, $\ln\sigma_\nu$ (labeled `lnsig2u` in the output). The standard deviation σ_ν is also included in the output, labeled `sigma_u`, together with ρ (labeled `rho`),

$$\rho = \frac{\sigma_\nu^2}{\sigma_\nu^2 + \sigma_\epsilon^2}$$

which is the proportion of the total variance contributed by the panel-level variance component.

When `rho` is zero, the panel-level variance component is not important, and the panel estimator is no different from the pooled estimator (`cloglog`). A likelihood-ratio test of this is included at the bottom of the output, which formally compares the pooled estimator with the panel estimator.

As an alternative to the random-effects specification, you might want to fit an equal-correlation population-averaged cloglog model by typing

```
. xtcloglog union age grade not_smsa south##c.year, pa
Iteration 1:   Tolerance = .11878399
Iteration 2:   Tolerance = .01424628
Iteration 3:   Tolerance = .00075278
Iteration 4:   Tolerance = .00003195
Iteration 5:   Tolerance = 1.661e-06
Iteration 6:   Tolerance = 8.308e-08

GEE population-averaged model                       Number of obs     = 26,200
Group variable: idcode                              Number of groups  =  4,434
Family: Binomial                                    Obs per group:
Link:   Complementary log-log                                    min =      1
Correlation: exchangeable                                        avg =    5.9
                                                                 max =     12
                                                    Wald chi2(6)      = 234.66
Scale parameter = 1                                 Prob > chi2       = 0.0000

------------------------------------------------------------------------------
       union | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
         age |   .0153737   .0081156     1.89   0.058    -.0005326      .03128
       grade |   .0549518   .0095093     5.78   0.000     .0363139    .0735897
    not_smsa |  -.1045232   .0431082    -2.42   0.015    -.1890138   -.0200326
     1.south |  -1.714868   .3384558    -5.07   0.000    -2.378229   -1.051507
        year |  -.0115881   .0084125    -1.38   0.168    -.0280763    .0049001
             |
south#c.year |
          1  |   .0149796   .0041687     3.59   0.000     .0068091    .0231501
             |
       _cons |  -1.488278   .4468005    -3.33   0.001    -2.363991   -.6125652
------------------------------------------------------------------------------
```

◁

▷ Example 2

In [R] **cloglog**, we showed these results and compared them with `cloglog, vce(cluster id)`. `xtcloglog` with the `pa` option allows a `vce(robust)` option so we can obtain the population-averaged cloglog estimator with the robust variance calculation by typing

```
. xtcloglog union age grade not_smsa south##c.year, pa vce(robust)
  (output omitted)
GEE population-averaged model                     Number of obs    = 26,200
Group variable: idcode                            Number of groups =  4,434
Family: Binomial                                  Obs per group:
Link:   Complementary log-log                                  min =      1
Correlation: exchangeable                                      avg =    5.9
                                                               max =     12
                                                  Wald chi2(6)     = 157.24
Scale parameter = 1                               Prob > chi2      = 0.0000

                                 (Std. err. adjusted for clustering on idcode)
```

union	Coefficient	Semirobust std. err.	z	P>|z|	[95% conf.	interval]
age	.0153737	.0079446	1.94	0.053	-.0001974	.0309448
grade	.0549518	.0117258	4.69	0.000	.0319697	.077934
not_smsa	-.1045232	.0548598	-1.91	0.057	-.2120465	.0030001
1.south	-1.714868	.4864999	-3.52	0.000	-2.66839	-.7613455
year	-.0115881	.0085742	-1.35	0.177	-.0283932	.005217
south#c.year						
1	.0149796	.0060548	2.47	0.013	.0031124	.0268468
_cons	-1.488278	.4924738	-3.02	0.003	-2.453509	-.5230472

These standard errors are similar to those shown for `cloglog, vce(cluster id)` in [R] **cloglog**.

◁

❑ Technical note

The random-effects model is calculated using quadrature, which is an approximation whose accuracy depends partially on the number of integration points used. We can use the `quadchk` command to see if changing the number of integration points affects the results. If the results change, the quadrature approximation is not accurate given the number of integration points. Try increasing the number of integration points using the `intpoints()` option and run `quadchk` again. Do not attempt to interpret the results of estimates when the coefficients reported by `quadchk` differ substantially. See [XT] **quadchk** for details and [XT] **xtprobit** for an example.

Because the `xtcloglog` likelihood function is calculated by Gauss–Hermite quadrature, on large problems the computations can be slow. Computation time is roughly proportional to the number of points used for the quadrature.

❑

Stored results

`xtcloglog, re` stores the following in `e()`:

Scalars

`e(N)`	number of observations
`e(N_g)`	number of groups
`e(k)`	number of parameters
`e(k_aux)`	number of auxiliary parameters
`e(k_eq)`	number of equations in `e(b)`
`e(k_eq_model)`	number of equations in overall model test
`e(k_dv)`	number of dependent variables
`e(df_m)`	model degrees of freedom
`e(ll)`	log likelihood
`e(ll_0)`	log likelihood, constant-only model
`e(ll_c)`	log likelihood, comparison model
`e(chi2)`	χ^2
`e(chi2_c)`	χ^2 for comparison test
`e(N_clust)`	number of clusters
`e(rho)`	ρ
`e(sigma_u)`	panel-level standard deviation
`e(n_quad)`	number of quadrature points
`e(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(g_max)`	largest group size
`e(p)`	p-value for model test
`e(rank)`	rank of `e(V)`
`e(rank0)`	rank of `e(V)` for constant-only model
`e(ic)`	number of iterations
`e(rc)`	return code
`e(converged)`	`1` if converged, `0` otherwise

Macros

`e(cmd)`	`xtcloglog`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(ivar)`	variable denoting groups
`e(model)`	`re`
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(title)`	title in estimation output
`e(clustvar)`	name of cluster variable
`e(offset)`	linear offset variable
`e(chi2type)`	`Wald` or `LR`; type of model χ^2 test
`e(chi2_ct)`	`Wald` or `LR`; type of model χ^2 test corresponding to `e(chi2_c)`
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. err.
`e(intmethod)`	integration method
`e(distrib)`	`Gaussian`; the distribution of the random effect
`e(opt)`	type of optimization
`e(which)`	`max` or `min`; whether optimizer is to perform maximization or minimization
`e(ml_method)`	type of `ml` method
`e(user)`	name of likelihood-evaluator program
`e(technique)`	maximization technique
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(marginsdefault)`	default `predict()` specification for `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`

Matrices

`e(b)`	coefficient vector
`e(Cns)`	constraints matrix
`e(ilog)`	iteration log
`e(gradient)`	gradient vector
`e(V)`	variance–covariance matrix of the estimators
`e(V_modelbased)`	model-based variance

Functions

`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

`xtcloglog, pa` stores the following in `e()`:

Scalars

`e(N)`	number of observations
`e(N_g)`	number of groups
`e(df_m)`	model degrees of freedom
`e(chi2)`	χ^2
`e(p)`	p-value for model test
`e(df_pear)`	degrees of freedom for Pearson χ^2
`e(chi2_dev)`	χ^2 test of deviance
`e(chi2_dis)`	χ^2 test of deviance dispersion
`e(deviance)`	deviance
`e(dispers)`	deviance dispersion
`e(phi)`	scale parameter
`e(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(g_max)`	largest group size
`e(rank)`	rank of `e(V)`
`e(tol)`	target tolerance
`e(dif)`	achieved tolerance
`e(rc)`	return code

Macros

`e(cmd)`	`xtgee`
`e(cmd2)`	`xtcloglog`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(ivar)`	variable denoting groups
`e(tvar)`	variable denoting time within groups
`e(model)`	`pa`
`e(family)`	`binomial`
`e(link)`	`cloglog`; link function
`e(corr)`	correlation structure
`e(scale)`	`x2`, `dev`, `phi`, or #; scale parameter
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(offset)`	linear offset variable
`e(chi2type)`	`Wald`; type of model χ^2 test
`e(vce)`	*vcetype* specified in `vce()`

`e(vcetype)`	title used to label Std. err.
`e(nmp)`	`nmp`, if specified
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(marginsnotok)`	predictions disallowed by `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`
Matrices	
`e(b)`	coefficient vector
`e(R)`	estimated working correlation matrix
`e(V)`	variance–covariance matrix of the estimators
`e(V_modelbased)`	model-based variance
Functions	
`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices	
`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

`xtcloglog, pa` reports the population-averaged results obtained using `xtgee, family(binomial) link(cloglog)` to obtain estimates.

For the random-effects model, assume a normal distribution, $N(0, \sigma_\nu^2)$, for the random effects ν_i,

$$\Pr(y_{i1}, \dots, y_{in_i} | \mathbf{x}_{i1}, \dots, \mathbf{x}_{in_i}) = \int_{-\infty}^{\infty} \frac{e^{-\nu_i^2/2\sigma_\nu^2}}{\sqrt{2\pi}\sigma_\nu} \left\{ \prod_{t=1}^{n_i} F(y_{it}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i) \right\} d\nu_i$$

where

$$F(y, z) = \begin{cases} 1 - \exp\{-\exp(z)\} & \text{if } y \neq 0 \\ \exp\{-\exp(z)\} & \text{otherwise} \end{cases}$$

The panel-level likelihood l_i is given by

$$\begin{aligned} l_i &= \int_{-\infty}^{\infty} \frac{e^{-\nu_i^2/2\sigma_\nu^2}}{\sqrt{2\pi}\sigma_\nu} \left\{ \prod_{t=1}^{n_i} F(y_{it}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i) \right\} d\nu_i \\ &\equiv \int_{-\infty}^{\infty} g(y_{it}, x_{it}, \nu_i) d\nu_i \end{aligned}$$

This integral can be approximated with M-point Gauss–Hermite quadrature

$$\int_{-\infty}^{\infty} e^{-x^2} h(x) dx \approx \sum_{m=1}^{M} w_m^* h(a_m^*)$$

This is equivalent to

$$\int_{-\infty}^{\infty} f(x) dx \approx \sum_{m=1}^{M} w_m^* \exp\left\{(a_m^*)^2\right\} f(a_m^*)$$

where the w_m^* denote the quadrature weights and the a_m^* denote the quadrature abscissas. The log likelihood, L, is the sum of the logs of the panel-level likelihoods l_i.

The default approximation of the log likelihood is by adaptive Gauss–Hermite quadrature, which approximates the panel-level likelihood with

$$l_i \approx \sqrt{2}\widehat{\sigma}_i \sum_{m=1}^{M} w_m^* \exp\left\{(a_m^*)^2\right\} g(y_{it}, x_{it}, \sqrt{2}\widehat{\sigma}_i a_m^* + \widehat{\mu}_i)$$

where $\widehat{\sigma}_i$ and $\widehat{\mu}_i$ are the adaptive parameters for panel i. Therefore, with the definition of $g(y_{it}, x_{it}, \nu_i)$, the total log likelihood is approximated by

$$L \approx \sum_{i=1}^{n} w_i \log\left[\sqrt{2}\widehat{\sigma}_i \sum_{m=1}^{M} w_m^* \exp\{(a_m^*)^2\} \frac{\exp\{-(\sqrt{2}\widehat{\sigma}_i a_m^* + \widehat{\mu}_i)^2/2\sigma_\nu^2\}}{\sqrt{2\pi}\sigma_\nu}\right.$$
$$\left.\prod_{t=1}^{n_i} F(y_{it}, x_{it}\boldsymbol{\beta} + \sqrt{2}\widehat{\sigma}_i a_m^* + \widehat{\mu}_i)\right]$$

where w_i is the user-specified weight for panel i; if no weights are specified, $w_i = 1$.

The default method of adaptive Gauss–Hermite quadrature is to calculate the posterior mean and variance and use those parameters for $\widehat{\mu}_i$ and $\widehat{\sigma}_i$ by following the method of Naylor and Smith (1982), further discussed in Skrondal and Rabe-Hesketh (2004). We start with $\widehat{\sigma}_{i,0} = 1$ and $\widehat{\mu}_{i,0} = 0$, and the posterior means and variances are updated in the kth iteration. That is, at the kth iteration of the optimization for l_i, we use

$$l_{i,k} \approx \sum_{m=1}^{M} \sqrt{2}\widehat{\sigma}_{i,k-1} w_m^* \exp\{a_m^*)^2\} g(y_{it}, x_{it}, \sqrt{2}\widehat{\sigma}_{i,k-1} a_m^* + \widehat{\mu}_{i,k-1})$$

Letting

$$\tau_{i,m,k-1} = \sqrt{2}\widehat{\sigma}_{i,k-1} a_m^* + \widehat{\mu}_{i,k-1}$$

$$\widehat{\mu}_{i,k} = \sum_{m=1}^{M} (\tau_{i,m,k-1}) \frac{\sqrt{2}\widehat{\sigma}_{i,k-1} w_m^* \exp\{(a_m^*)^2\} g(y_{it}, x_{it}, \tau_{i,m,k-1})}{l_{i,k}}$$

and

$$\widehat{\sigma}_{i,k} = \sum_{m=1}^{M} (\tau_{i,m,k-1})^2 \frac{\sqrt{2}\widehat{\sigma}_{i,k-1} w_m^* \exp\{(a_m^*)^2\} g(y_{it}, x_{it}, \tau_{i,m,k-1})}{l_{i,k}} - (\widehat{\mu}_{i,k})^2$$

and this is repeated until $\widehat{\mu}_{i,k}$ and $\widehat{\sigma}_{i,k}$ have converged for this iteration of the maximization algorithm. This adaptation is applied on every iteration until the log-likelihood change from the preceding iteration is less than a relative difference of 1e–6; after this, the quadrature parameters are fixed.

The log likelihood can also be calculated by nonadaptive Gauss–Hermite quadrature, the `intmethod(ghermite)` option, where $\rho = \sigma_\nu^2/(\sigma_\nu^2 + 1)$:

$$\begin{aligned} L &= \sum_{i=1}^{n} w_i \log\left\{ \Pr(y_{i1}, \dots, y_{in_i} | \mathbf{x}_{i1}, \dots, \mathbf{x}_{in_i}) \right\} \\ &\approx \sum_{i=1}^{n} w_i \log\left[\frac{1}{\sqrt{\pi}} \sum_{m=1}^{M} w_m^* \prod_{t=1}^{n_i} F\left\{ y_{it}, \mathbf{x}_{it}\boldsymbol{\beta} + a_m^* \left(\frac{2\rho}{1-\rho}\right)^{1/2} \right\} \right] \end{aligned}$$

Both quadrature formulas require that the integrated function be well approximated by a polynomial of degree equal to the number of quadrature points. The number of periods (panel size) can affect whether

$$\prod_{t=1}^{n_i} F(y_{it}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i)$$

is well approximated by a polynomial. As panel size and ρ increase, the quadrature approximation can become less accurate. For large ρ, the random-effects model can also become unidentified. Adaptive quadrature gives better results for correlated data and large panels than nonadaptive quadrature; however, we recommend that you use the `quadchk` command (see [XT] **quadchk**) to verify the quadrature approximation used in this command, whichever approximation you choose.

xtcloglog, re and the robust VCE estimator

Specifying `vce(robust)` or `vce(cluster` *clustvar*`)` causes the Huber/White/sandwich VCE estimator to be calculated for the coefficients estimated in this regression. See [P] **_robust**, particularly *Introduction* and *Methods and formulas*. Wooldridge (2020) and Arellano (2003) discuss this application of the Huber/White/sandwich VCE estimator. As discussed by Wooldridge (2020), Stock and Watson (2008), and Arellano (2003), specifying `vce(robust)` is equivalent to specifying `vce(cluster` *panelvar*`)`, where *panelvar* is the variable that identifies the panels.

Clustering on the panel variable produces a consistent VCE estimator when the disturbances are not identically distributed over the panels or there is serial correlation in ϵ_{it}.

The cluster–robust VCE estimator requires that there are many clusters and the disturbances are uncorrelated across the clusters. The panel variable must be nested within the cluster variable because of the within-panel correlation that is generally induced by the random-effects transform when there is heteroskedasticity or within-panel serial correlation in the idiosyncratic errors.

References

Arellano, M. 2003. *Panel Data Econometrics*. Oxford: Oxford University Press. https://doi.org/10.1093/0199245282.001.0001.

Liang, K.-Y., and S. L. Zeger. 1986. Longitudinal data analysis using generalized linear models. *Biometrika* 73: 13–22. https://doi.org/10.1093/biomet/73.1.13.

Naylor, J. C., and A. F. M. Smith. 1982. Applications of a method for the efficient computation of posterior distributions. *Journal of the Royal Statistical Society,* C ser., 31: 214–225. https://doi.org/10.2307/2347995.

Neuhaus, J. M. 1992. Statistical methods for longitudinal and clustered designs with binary responses. *Statistical Methods in Medical Research* 1: 249–273. https://doi.org/10.1177/096228029200100303.

Neuhaus, J. M., J. D. Kalbfleisch, and W. W. Hauck. 1991. A comparison of cluster-specific and population-averaged approaches for analyzing correlated binary data. *International Statistical Review* 59: 25–35. https://doi.org/10.2307/1403572.

Pendergast, J. F., S. J. Gange, M. A. Newton, M. J. Lindstrom, M. Palta, and M. R. Fisher. 1996. A survey of methods for analyzing clustered binary response data. *International Statistical Review* 64: 89–118. https://doi.org/10.2307/1403425.

Skrondal, A., and S. Rabe-Hesketh. 2004. *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models*. Boca Raton, FL: Chapman and Hall/CRC.

Stock, J. H., and M. W. Watson. 2008. Heteroskedasticity-robust standard errors for fixed effects panel data regression. *Econometrica* 76: 155–174. https://doi.org/10.1111/j.0012-9682.2008.00821.x.

Wooldridge, J. M. 2020. *Introductory Econometrics: A Modern Approach*. 7th ed. Boston: Cengage.

Also see

[XT] **xtcloglog postestimation** — Postestimation tools for xtcloglog

[XT] **quadchk** — Check sensitivity of quadrature approximation

[XT] **xtgee** — GEE population-averaged panel-data models

[XT] **xtlogit** — Fixed-effects, random-effects, and population-averaged logit models

[XT] **xtprobit** — Random-effects and population-averaged probit models

[XT] **xtset** — Declare data to be panel data

[ME] **mecloglog** — Multilevel mixed-effects complementary log–log regression

[MI] **Estimation** — Estimation commands for use with mi estimate

[R] **cloglog** — Complementary log–log regression

[U] 20 Estimation and postestimation commands

Postestimation commands predict margins Remarks and examples Also see

Postestimation commands

The following postestimation commands are available after `xtcloglog`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of parameters
* `estat ic`	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC, respectively)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
† `forecast`	dynamic forecasts and simulations
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
`lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	linear predictions and their SEs, probabilities
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of parameters
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

*`estat ic` is not appropriate after `xtcloglog, pa`.

†`forecast` is not appropriate with `mi` estimation results.

predict

Description for predict

`predict` creates a new variable containing predictions such as probabilities, linear predictions, standard errors, and the equation-level score.

Menu for predict

Statistics > Postestimation

Syntax for predict

Random-effects (RE) model

`predict` [*type*] *newvar* [*if*] [*in*] [, *RE_statistic* `nooffset`]

Population-averaged (PA) model

`predict` [*type*] *newvar* [*if*] [*in*] [, *PA_statistic* `nooffset`]

RE_statistic	Description
Main	
`xb`	linear prediction; the default
`pr`	marginal probability of a positive outcome
`pu0`	probability of a positive outcome
`stdp`	standard error of the linear prediction

PA_statistic	Description
Main	
`mu`	predicted probability of *depvar*; considers the `offset()`; the default
`rate`	predicted probability of *depvar*
`xb`	linear prediction
`stdp`	standard error of the linear prediction
`score`	first derivative of the log likelihood with respect to $\mathbf{x}_{it}\boldsymbol{\beta}$

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

Options for predict

Main

`xb` calculates the linear prediction, which is $\mathbf{x}_{it}\boldsymbol{\beta}$ if `offset()` was not specified when the model was fit and $\mathbf{x}_{it}\boldsymbol{\beta} + \text{offset}_{it}$ if `offset()` was specified. This is the default for the random-effects model.

`pr` calculates the probability of a positive outcome that is marginal with respect to the random effect, which means that the probability is calculated by integrating the prediction function with respect to the random effect over its entire support.

`pu0` calculates the probability of a positive outcome, assuming that the random effect for that observation's panel is zero ($\nu_i = 0$). This may not be similar to the proportion of observed outcomes in the group.

`stdp` calculates the standard error of the linear prediction.

`mu` and `rate` both calculate the predicted probability of *depvar*. `mu` takes into account the `offset()`. `rate` ignores those adjustments. `mu` and `rate` are equivalent if you did not specify `offset()`. `mu` is the default for the population-averaged model.

`score` calculates the equation-level score, $u_{it} = \partial \ln L(\mathbf{x}_{it}\boldsymbol{\beta})/\partial(\mathbf{x}_{it}\boldsymbol{\beta})$.

`nooffset` is relevant only if you specified `offset(`*varname*`)` for `xtcloglog`. It modifies the calculations made by `predict` so that they ignore the offset variable; the linear prediction is treated as $\mathbf{x}_{it}\boldsymbol{\beta}$ rather than $\mathbf{x}_{it}\boldsymbol{\beta} + \text{offset}_{it}$.

margins

Description for margins

`margins` estimates margins of responses for probabilities and linear predictions.

Menu for margins

Statistics > Postestimation

Syntax for margins

`margins` [*marginlist*] [, *options*]

`margins` [*marginlist*] , `predict(`*statistic* ...`)` [`predict(`*statistic* ...`)` ...] [*options*]

Random-effects (RE) model

statistic	Description
`pr`	marginal probability of a positive outcome; the default
`pu0`	probability of a positive outcome
`xb`	linear prediction
`stdp`	not allowed with `margins`

Population-averaged (PA) model

statistic	Description
`mu`	predicted probability of *depvar*; considers the `offset()`; the default
`rate`	predicted probability of *depvar*
`xb`	linear prediction
`stdp`	not allowed with `margins`
`score`	not allowed with `margins`

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.

For the full syntax, see [R] **margins**.

Remarks and examples

▷ Example 1: Average marginal effects

In example 1 of [XT] **xtcloglog**, we fit the model

```
. use https://www.stata-press.com/data/r19/union
(NLS Women 14-24 in 1968)
. xtcloglog union age grade not_smsa south##c.year, pa
  (output omitted)
```

Here we use `margins` to determine the average effect each regressor has on the probability of a positive response in the sample.

```
. margins, dydx(*)
Average marginal effects                                Number of obs = 26,200
Model VCE: Conventional
Expression: Pr(union != 0), predict()
dy/dx wrt:  age grade not_smsa 1.south year
```

	dy/dx	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
age	.0028297	.0014952	1.89	0.058	-.000101	.0057603
grade	.0101144	.0017498	5.78	0.000	.0066848	.013544
not_smsa	-.0192384	.0079304	-2.43	0.015	-.0347818	-.0036951
1.south	-.0913197	.0073101	-12.49	0.000	-.1056473	-.0769921
year	-.0012694	.001534	-0.83	0.408	-.004276	.0017371

```
Note: dy/dx for factor levels is the discrete change from the base level.
```

We see that an additional year of schooling (covariate `grade`) increases the probability that a woman belongs to a union by an average of about one percentage point.

◁

Also see

[XT] **xtcloglog** — Random-effects and population-averaged cloglog models

[U] 20 Estimation and postestimation commands

xtcointtest — Panel-data cointegration tests

Description

`xtcointtest` performs the Kao (1999), Pedroni (1999, 2004), and Westerlund (2005) tests of cointegration on a panel dataset. Panel-specific means (fixed effects) and panel-specific time trends may be included in the cointegrating regression model.

All tests have a common null hypothesis of no cointegration. The alternative hypothesis of the Kao tests and the Pedroni tests is that the variables are cointegrated in all panels. In one version of the Westerlund test, the alternative hypothesis is that the variables are cointegrated in some of the panels. In another version of the Westerlund test, the alternative hypothesis is that the variables are cointegrated in all the panels.

Quick start

Kao test of no cointegration between `y` and `x` with the alternative hypothesis that they are cointegrated in all panels using `xtset` data

```
xtcointtest kao y x
```

Pedroni test of no cointegration using a panel-specific autoregressive (AR) term and panel-specific time trends with the alternative hypothesis of cointegration in all panels

```
xtcointtest pedroni y x, trend
```

Same as above, but use the same AR term in all panels

```
xtcointtest pedroni y x, trend ar(same)
```

Westerlund test of no cointegration with the alternative hypothesis that the variables are cointegrated in some of the panels

```
xtcointtest westerlund y x
```

Westerlund test of no cointegration with the alternative hypothesis of cointegration in all panels

```
xtcointtest westerlund y x, allpanels
```

Menu

Statistics > Longitudinal/panel data > Cointegrated data > Tests for cointegration

Syntax

Kao test

xtcointtest kao *depvar varlist* [*if*] [*in*] [, *kao_options*]

Pedroni test

xtcointtest pedroni *depvar varlist* [*if*] [*in*] [, *pedroni_options*]

Westerlund test

xtcointtest westerlund *depvar varlist* [*if*] [*in*] [, *westerlund_options*]

kao_options	Description
Main	
lags(*lspec*)	specify lag structure for augmented Dickey–Fuller regressions
kernel(*kspec*)	specify method to estimate long-run variance
demean	subtract cross-sectional means

pedroni_options	Description
Main	
ar(panelspecific | same)	specify autoregressive parameter as panel specific or as the same for all panels; ar(panelspecific) is the default
trend	include panel-specific time trends
noconstant	suppress panel-specific means
lags(*lspec*)	specify lag structure for augmented Dickey–Fuller regressions
kernel(*kspec*)	specify method to estimate long-run variance
demean	subtract cross-sectional means

westerlund_options	Description
Main	
somepanels	use alternative hypothesis of cointegration in some panels; the default
allpanels	use alternative hypothesis of cointegration in all panels
trend	include panel-specific time trends
demean	subtract cross-sectional means

collect is allowed with all xtcointtest tests; see **[U] 11.1.10 Prefix commands**.

lspec is

#	number of lags of series; 1 is the default
aic #	Akaike information criterion (AIC) with up to # lags
bic #	Bayesian information criterion (BIC) with up to # lags
hqic #	Hannan–Quinn information criterion (HQIC) with up to # lags

kspec is

bartlett nwest	Bartlett kernel with Newey–West lags; the default
bartlett #	Bartlett kernel with up to # lags
parzen nwest	Parzen kernel with Newey–West lags
parzen #	Parzen kernel with up to # lags
quadraticspectral nwest	quadratic spectral kernel with Newey–West lags
quadraticspectral #	quadratic spectral kernel with up to # lags

Options

Options are presented under the following headings:

Options for xtcointtest kao
Options for xtcointtest pedroni
Options for xtcointtest westerlund

Options for xtcointtest kao

Main

`lags(`*lspec*`)` specifies the lag structure to use for the augmented Dickey–Fuller (ADF) regressions performed in computing the test statistic.

`lags(`#`)` specifies that # lags of the series be used in the ADF regressions. # must be a nonnegative integer. The default is `lags(1)`.

`lags(aic | bic | hqic` #`)` specifies that `xtcointtest` fit ADF regressions with 1 to # lags and choose the number of lags for which the AIC, BIC, or HQIC is minimized.

`kernel(`*kspec*`)` specifies the method used to estimate the long-run variance of each panel's series. You may specify the kernel type and either #, the maximum number of lags as a positive integer, or `nwest`, the maximum number of lags selected by the bandwidth-selection algorithm given in Newey and West (1994). The kernel type may be `bartlett`, `parzen`, or `quadraticspectral`. The default is `kernel(bartlett nwest)`.

`demean` specifies that `xtcointtest` first subtract the cross-sectional averages from the series. When specified, for each time period `xtcointtest` computes the mean of the series across panels and subtracts this mean from the series. Levin, Lin, and Chu (2002) suggest this procedure to mitigate the impact of cross-sectional dependence.

Options for xtcointtest pedroni

Main

`ar(panelspecific | same)` specifies whether the AR parameter for ADF or Phillips–Perron (PP) regressions is panel specific or the same across panels.

`ar(panelspecific)` specifies that the AR parameter be panel specific in the ADF or PP regressions. The test statistics obtained from using this option are also known as group-mean statistics or between-dimension statistics. This is the default.

`ar(same)` specifies that the AR parameter be the same for all panels in the ADF or PP regressions. The test statistics obtained from using this option are also known as panel cointegration statistics or within-dimension statistics.

`trend` includes panel-specific linear time trends in the model for the dependent variable on the covariates.

`noconstant` suppresses the panel-specific means in the model for the dependent variable on the covariates. Specifying `noconstant` imposes the assumption that the series has a mean of zero for all panels. This option may not be specified with `trend`.

`lags(`*lspec*`)` specifies the lag structure to use for the ADF regressions performed in computing the test statistic. See the description of `lags()` under *Options for xtcointtest kao* for additional details.

`kernel(`*kspec*`)` specifies the method used to estimate the long-run variance of each panel's series. See the description of `kernel()` under *Options for xtcointtest kao* for additional details.

`demean` specifies that `xtcointtest` first subtract the cross-sectional averages from the series. See the description of `demean` under *Options for xtcointtest kao* for additional details.

Options for xtcointtest westerlund

Main

`somepanels` specifies that the test statistic for panel cointegration be computed using the alternative hypothesis that some of the panels are cointegrated. This statistic is also known as the group-mean variance-ratio (VR) statistic. This option uses a regression in which the AR parameter for Dickey–Fuller (DF) regressions is panel specific. This is the default.

`allpanels` specifies that the test statistic for panel cointegration be computed using the alternative hypothesis that all the panels are cointegrated, also known as the panel VR statistic. This option also implies that the AR parameter for DF regressions is the same for all panels.

`trend` includes panel-specific linear time trends in the model for dependent variable on the covariates.

`demean` specifies that `xtcointtest` first subtract the cross-sectional averages from the series. See the description of `demean` under *Options for xtcointtest kao* for additional details.

Remarks and examples

Remarks are presented under the following headings:

Overview

A stationary process has a time-invariant mean and a time-invariant variance. By contrast, a nonstationary process has a time-varying mean, a time-varying variance, or both. A nonstationary process may wander arbitrarily over time because its first two moments vary over time.

When the first difference of a nonstationary process is stationary, the process is said to be integrated of order one, denoted $I(1)$. When a linear combination of several $I(1)$ series is stationary, the series are said to be cointegrated (Engle and Granger 1987). We test for cointegration because cointegration implies that the $I(1)$ series are in a long-run equilibrium; they move together, although the group of them can wander arbitrarily.

For example, income and consumption are $I(1)$ series that wander over time. According to economic theory, income determines consumption in the long run. In practice, time-series data on income and consumption typically have periods where the series seem to wander in isolation, which is contrary to the theory. However, when we look at the overall trend, the two series are close to one another, implying a long-run relation. A test of cointegration provides evidence that indeed there is (or is not) a long-run relation between these series even if they tend to deviate temporarily.

`xtcointtest` implements tests of cointegration in panel data, which have many observations on each of many individual units. This type of sample is known as large-N-large-T-panel data. The popular Engle–Granger residual-based test for cointegration has low power when applied to a single time series but has good power when statistics from many individual panels are combined. The Kao tests, the Pedroni tests, and the Westerlund tests implemented in `xtcointtest` combine statistics computed for each individual in the panel, thereby producing a test with higher power. Furthermore, the limiting distribution of the combined test converges to a standard normal distribution after appropriate standardization, whereas tests for cointegration based on a single time series have nonstandard distributions.

All the tests in `xtcointtest` are based on the following panel-data model for the $I(1)$ dependent variable y_{it}, where $i = 1, \ldots, N$ denotes the panel (individual) and $t = 1, \ldots, T_i$ denotes time:

$$y_{it} = \mathbf{x}'_{it}\boldsymbol{\beta}_i + \mathbf{z}'_{it}\boldsymbol{\gamma}_i + e_{it} \tag{1}$$

For each panel i, each of the covariates in $\mathbf{x}_{it}$ is an $I(1)$ series. All the tests require that the covariates are not cointegrated among themselves. The Pedroni and Westerlund tests allow a maximum of seven covariates in $\mathbf{x}_{it}$. $\boldsymbol{\beta}_i$ denotes the cointegrating vector, which may vary across panels. $\boldsymbol{\gamma}_i$ is a vector of coefficients on $\mathbf{z}_{it}$, the deterministic terms that control for panel-specific effects and linear time trends. e_{it} is the error term.

Depending on the options specified with `xtcointtest`, the vector $\mathbf{z}_{it}$ allows for panel-specific means, panel-specific means and panel-specific time trends, or nothing. By default, $\mathbf{z}_{it} = 1$, so the term $\mathbf{z}'_{it}\boldsymbol{\gamma}_i$ represents panel-specific means (fixed effects). If `trend` is specified, $\mathbf{z}'_{it} = (1, t)$ so $\mathbf{z}'_{it}\boldsymbol{\gamma}_i$ represents panel-specific means and panel-specific linear time trends. For tests that allow it, specifying `noconstant` omits the $\mathbf{z}'_{it}\boldsymbol{\gamma}_i$ term.

The tests share a common null hypothesis that y_{it} and $\mathbf{x}_{it}$ are not cointegrated. `xtcointtest` tests for no cointegration by testing that e_t is nonstationary. Rejection of the null hypothesis implies that e_{it} is stationary and that the series y_{it} and $\mathbf{x}_{it}$ are cointegrated. The alternative hypothesis of the Kao tests, the Pedroni tests, and the `allpanels` version of the Westerlund test is that the variables are cointegrated in all panels. The alternative hypothesis of the `somepanels` version of the Westerlund test is that the variables are cointegrated in some of the panels.

All tests allow unbalanced panels and require that N is large enough that the distribution of a sample average of panel-level statistics converges to its population distribution. They also require that each T_i is large enough to run time-series regressions using observations only from that panel. These tests have nominal coverage only when both T and N are large. The smallest combinations of T and N for which the tests have close to nominal coverage and decent power differs by test and varies with the degree of serial correlation in the residuals. See *Test details* for more information. All the tests require that there be no gaps in any panel's series.

Test details

The Kao, Pedroni, and Westerlund tests implement different types of tests for whether e_{it} is nonstationary. The DF t tests, ADF t tests, PP t tests, and their variants that are reported by `xtcointtest kao` and `xtcointtest pedroni` use different regression frameworks to handle serial correlation in e_{it}. The VR tests that are reported by `xtcointtest westerlund` and `xtcointtest pedroni` do not require modeling or accommodating for serial correlation; see Westerlund (2005).

All variants of the DF t test statistics are constructed by fitting the model in (1) using ordinary least squares, obtaining the predicted residuals ($\hat{e}_{it}$), and then fitting the DF regression model

$$\hat{e}_{it} = \rho \hat{e}_{i,t-1} + \nu_{it} \tag{2}$$

where ρ is the AR parameter and ν_{it} is a stationary error term. The DF t and the unadjusted DF t test whether the coefficient ρ is 1. By contrast, the modified DF t and the unadjusted modified DF t test whether $\rho - 1 = 0$. Nonstationarity under the null hypothesis causes a test of whether $\rho = 1$ to differ from a test of whether $\rho - 1 = 0$; see Dickey and Fuller (1979) and Kao (1999).

The variants of the PP t test statistics are also constructed by fitting the model in (1) using ordinary least squares and obtaining the predicted residuals ($\hat{e}_{it}$). For the PP t tests, we then fit the DF regression model

$$\hat{e}_{it} = \rho_i \hat{e}_{i,t-1} + \nu_{it} \tag{3}$$

In this case, we have a panel-specific AR parameter ρ_i. The PP t tests whether the ρ_is are 1, whereas the modified PP t tests whether $\rho_i - 1 = 0$. The PP t test statistic is nonparametrically adjusted for serial correlation in the residuals using the Newey and West (1987) heteroskedasticity- and autocorrelation-consistent (HAC) covariance matrix estimator.

The DF t, the modified DF t, the PP t, the modified PP t, and the modified VR tests are derived by specifying a data-generating process for the dependent variable and the regressors. This specification allows the regressors to be endogenous as well as serially correlated. Therefore, constructing the test statistics requires estimating the contemporaneous and dynamic covariances between the regressors and the dependent variable. The unadjusted DF t and the unadjusted modified DF t assume absence of serial correlation and strictly exogenous covariates and do not require any adjustments in the residuals.

Like the DF and PP tests, the ADF t tests that $\rho = 1$. However, the ADF test uses additional lags of the residuals to control for serial correlation instead of the Newey–West nonparametric adjustments. The ADF regression is

$$\hat{e}_{it} = \rho_i \hat{e}_{i,t-1} + \sum_{j=1}^{p} \rho_{ij} \Delta \hat{e}_{i,t-j} + \nu^*_{it}$$

where $\Delta \hat{e}_{i,t-j}$ is the jth lag of the first difference of $\hat{e}_{it}$ and $j = 1, \ldots, p$ is where p is the number of lag differences.

The VR tests are based on Phillips and Ouliaris (1990) and Breitung (2002), where the test statistic is constructed as a ratio of variances. These tests do not require modeling or accommodating serial correlation; see Westerlund (2005). VR tests also test for no cointegration by testing for the presence of a unit root in the residuals. However, they do so using the ratio of variances of the predicted residuals. The modified VR test removes estimated conditional variances prior to computing the VR. For further details, see *Methods and formulas*.

These tests get good coverage and power properties by combining panel-level statistics computed from a time-series regression using only the observations in that panel. Kao (1999) finds that his tests have nearly nominal size when $T = 100$ and $N = 300$. Pedroni (2004) finds that his tests have nearly nominal size when $T = 250$ and $N = 60$. Westerlund (2005) limited his simulations to datasets with $T = 150$, and he did not find a combination of T and N in which his tests had nearly nominal size. He said that $T > 150$ should produce better coverage. Each author used a different data-generating process; see Kao (1999), Pedroni (2004), and Westerlund (2005) for details.

❑ Technical note

The asymptotic distribution of all the test statistics are obtained using sequential limit theory, denoted as $(T, N) \rightarrow_{\text{seq}} \infty$, in which the time dimension goes to infinity followed by the number of panels going to infinity. See Phillips and Moon (2000) for an introduction to asymptotic theory that depend on both N and T and their relation to nonstationary panels. Phillips and Moon (1999) contains a more technical discussion of "multi-indexed" asymptotic theory.

❑

Kao tests

The tests derived in Kao (1999) assume a cointegrating vector that is the same across all panels, which restricts $\boldsymbol{\beta}_i = \boldsymbol{\beta}$ in (1). Kao tests estimate panel-specific means and do not allow a time trend, so $\mathbf{z}$ from (1) is always a vector of 1s for Kao tests. This yields the cointegrating relationship

$$y_{it} = \gamma_i + \mathbf{x}'_{it}\boldsymbol{\beta} + e_{it}$$

where γ_i denotes panel-specific means (fixed effects). The null hypothesis of the Kao test is that there is no cointegration among the series. The alternative hypothesis is that the series in all panels are cointegrated with the same cointegrating vector.

`xtcointtest kao` reports the modified DF t, DF t, ADF t, unadjusted modified DF t, and unadjusted DF t statistics. They are constructed using the estimated ρ from DF and ADF regressions; see *Test details*. The test statistics differ in how they formulate the hypothesis and in how they control for serial correlation in e_{it}. See *Test details* for an overview of the differences in the test statistics and see *Kao tests* in *Methods and formulas* for further discussion.

▷ Example 1: Kao tests assuming a constant cointegrating vector

We are interested in the long-run effect of domestic research and development (R&D) and foreign R&D on an economy's productivity. The fictitious dataset, `xtcoint.dta`, is a balanced panel on 100 countries observed from 1973q3 to 2010q4. It contains quarterly data on the log of productivity (`productivity`), log of domestic R&D capital stock (`rddomestic`), and log of foreign R&D (`rdforeign`).

The cointegrating relationship is specified as

$$\texttt{productivity}_{it} = \gamma_i + \beta_1 \texttt{rddomestic}_{it} + \beta_2 \texttt{rdforeign}_{it} + e_{it}$$

where γ_i is the panel-specific mean and the cointegrating parameters, β_1 and β_2, are the same across panels. We assume that each series is $I(1)$. A formal test for the presence of a unit root in panel data may be performed using `xtunitroot`. We perform the Kao test of cointegration by typing

```
. use https://www.stata-press.com/data/r19/xtcoint
(Fictitious cointegration data)

. xtcointtest kao productivity rddomestic rdforeign

Kao test for cointegration
--------------------------

H0: No cointegration                        Number of panels       =    100
Ha: All panels are cointegrated             Number of periods      =    148

Cointegrating vector: Same
Panel means:          Included              Kernel:            Bartlett
Time trend:           Not included          Lags:              3.60 (Newey-West)
AR parameter:         Same                  Augmented lags:    1
------------------------------------------------------------------------------
                                            Statistic          p-value
------------------------------------------------------------------------------
 Modified Dickey-Fuller t                   -23.6733           0.0000
 Dickey-Fuller t                            -15.1293           0.0000
 Augmented Dickey-Fuller t                   -3.6909           0.0001
 Unadjusted modified Dickey-Fuller t        -46.7561           0.0000
 Unadjusted Dickey-Fuller t                 -20.2521           0.0000
------------------------------------------------------------------------------
```

We used a model with panel-specific means and no time trend, as reported in the header. The AR parameter that determines the presence or lack of cointegration is assumed to be the same for all panels and is thus labeled as `Same` in the header.

By default, `xtcointtest kao` uses a Bartlett kernel with Newey and West (1994) automatic lag selection algorithm. In this example, the algorithm chose an average of 3.6 lags across all panels to correct for serial correlation. To choose different kernels and the number of lags, specify the `kernel()` option. The ADF t statistic also includes lagged differences of the dependent variable to control for serial correlation. The number of lags is reported in `Augmented lags`. By default, `xtcointtest kao` uses the first lag. To include more lags, specify the `lags()` option.

The output reports the values of all test statistics with their respective p-values. All test statistics reject the null hypothesis of no cointegration in favor of the alternative hypothesis of the existence of a cointegrating relation among `productivity`, `rddomestic`, and `rdforeign`. The modified DF t, the DF t, and the ADF t test statistics are adjusted for serial correlation using the HAC estimator; see *Methods and formulas*.

◁

Pedroni tests

The tests derived by Pedroni (1999, 2004) allow for panel-specific cointegrating vectors. This heterogeneity distinguishes Pedroni tests from those derived by Kao. Another difference is that the Pedroni tests allow the AR coefficient (ρ_i) to vary over panels as in (3), while the Kao tests assumed the same AR coefficient. These panel-specific AR coefficients are the default in the Pedroni tests, but the `ar(same)` option restricts the AR coefficients ($\rho_i = \rho$) to be the same over panels.

Pedroni (1999, 2004) refers to the tests based on panel-specific AR parameters as "between-dimension tests" and refers to the tests based on the same AR parameters as "within-dimension tests".

See *Test details* and *Methods and formulas* for further discussion of the specific tests.

▷ Example 2: Pedroni cointegration test with panel-specific AR parameter

Continuing with example 1, we perform the Pedroni test of cointegration between `productivity`, `rddomestic`, and `rdforeign`, assuming panel-specific cointegrating vectors and autoregressive parameters. The cointegrating relationship is specified as

$$\texttt{productivity}_{it} = \gamma_i + \beta_{1i}\texttt{rddomestic}_{it} + \beta_{2i}\texttt{rdforeign}_{it} + e_{it}$$

where β_{1i} and β_{2i} represent panel-specific cointegration parameters.

```
. xtcointtest pedroni productivity rddomestic rdforeign

Pedroni test for cointegration
------------------------------
H0: No cointegration                           Number of panels       =    100
Ha: All panels are cointegrated                Number of periods      =    149

Cointegrating vector: Panel specific
Panel means:          Included                 Kernel:            Bartlett
Time trend:           Not included             Lags:              4.00 (Newey-West)
AR parameter:         Panel specific           Augmented lags:    1
------------------------------------------------------------------------------
                                               Statistic          p-value
------------------------------------------------------------------------------
 Modified Phillips-Perron t                    -26.1145           0.0000
 Phillips-Perron t                             -21.2436           0.0000
 Augmented Dickey-Fuller t                     -25.3701           0.0000
------------------------------------------------------------------------------
```

All the test statistics reject the null hypothesis of no cointegration in favor of the alternative hypothesis that `productivity`, `rddomestic`, and `rdforeign` are cointegrated in all panels with a panel-specific cointegrating vector.

The model underlying the reported statistics includes panel-specific means and panel-specific AR parameters and does not include a time trend. All three statistics used a Bartlett kernel with four lags, as selected by the Newey–West methods, to adjust for serial correlation. The ADF test used a regression with only one additional lag.

◁

▷ Example 3: Pedroni cointegration test with a common AR parameter

The alternative hypothesis in example 2 allows for panel-specific AR parameters. In this example, we use the ar(same) option to specify an alternative hypothesis that assumes the same AR parameter across all panels.

```
. xtcointtest pedroni productivity rddomestic rdforeign, ar(same)
Pedroni test for cointegration
──────────────────────────────
H0: No cointegration                       Number of panels       =    100
Ha: All panels are cointegrated            Number of periods      =    149

Cointegrating vector: Panel specific
Panel means:          Included             Kernel:          Bartlett
Time trend:           Not included         Lags:            4.00 (Newey-West)
AR parameter:         Same                 Augmented lags:  1
──────────────────────────────────────────────────────────────────────────────
                                            Statistic         p-value
──────────────────────────────────────────────────────────────────────────────
 Modified variance ratio                     14.8852          0.0000
 Modified Phillips-Perron t                 -23.0042          0.0000
 Phillips-Perron t                          -17.6735          0.0000
 Augmented Dickey-Fuller t                  -21.1119          0.0000
──────────────────────────────────────────────────────────────────────────────
```

All test statistics reject the null hypothesis of no cointegration in favor of the alternative hypothesis of cointegration between productivity, rddomestic, and rdforeign.

The header reports Same for the AR parameter, reminding us that we are now using an alternative hypothesis that assumes a constant ρ for all panels.

◁

Westerlund tests

Westerlund (2005) derived a pair of VR test statistics for the null hypothesis of no cointegration. The default test uses a model in which the AR parameter is panel specific and for which the alternative hypothesis is that the series in some of the panels are cointegrated. Specifying the allpanels option produces the results for a test in which the alternative hypothesis is that the series in all the panels are cointegrated, and this test uses a model in which the AR parameter is the same over the panels. More specifically, the alternative hypothesis using the allpanels option restricts $\rho_i = \rho$ in (3).

See *Test details* and *Methods and formulas* for further discussion of the specific tests.

▷ Example 4: Westerlund test with some panels cointegrated under the alternative

Continuing with example 1, we perform the Westerlund test of cointegration between productivity, rddomestic, and rdforeign. The cointegrating relationship is specified as

$$\texttt{productivity}_{it} = \gamma_i + \beta_{1i}\texttt{rddomestic}_{it} + \beta_{2i}\texttt{rdforeign}_{it} + e_{it}$$

where β_{1i} and β_{2i} are panel-specific cointegration parameters. We now test the null hypothesis of no cointegration under the alternative that some of the β_{1i} and β_{2i} produce cointegrated series:

```
. xtcointtest westerlund productivity rddomestic rdforeign
Westerlund test for cointegration
─────────────────────────────────
H0: No cointegration                        Number of panels      =     100
Ha: Some panels are cointegrated            Number of periods     =     150

Cointegrating vector: Panel specific
Panel means:          Included
Time trend:           Not included
AR parameter:         Panel specific
──────────────────────────────────────────────────────────────────────────────
                                            Statistic         p-value
──────────────────────────────────────────────────────────────────────────────
 Variance ratio                              -8.0237          0.0000
──────────────────────────────────────────────────────────────────────────────
```

The VR test statistic rejects the null hypothesis of no cointegration between `productivity`, `rddomestic`, and `rdforeign` in favor of the alternative that at least some panels are cointegrated.

The header tells us that the cointegrating vectors vary by panel, that panel-specific means were included in the model, that no time trend was included in the model, and that the AR parameter varies by panel.

◁

▷ Example 5: Westerlund test with all panels cointegrated under the alternative

In this example, we use the `allpanels` option to test the null hypothesis of no cointegration under the alternative hypothesis that all panels are cointegrated. This test is based on a model in which the AR parameter is the same over the panels.

```
. xtcointtest westerlund productivity rddomestic rdforeign, allpanels
Westerlund test for cointegration
─────────────────────────────────
H0: No cointegration                        Number of panels      =     100
Ha: All panels are cointegrated             Number of periods     =     150

Cointegrating vector: Panel specific
Panel means:          Included
Time trend:           Not included
AR parameter:         Same
──────────────────────────────────────────────────────────────────────────────
                                            Statistic         p-value
──────────────────────────────────────────────────────────────────────────────
 Variance ratio                              -5.9709          0.0000
──────────────────────────────────────────────────────────────────────────────
```

The VR statistic rejects the null hypothesis of no cointegration. This implies all panels are cointegrated.

◁

Stored results

`xtcointtest kao` stores the following in `r()`:

Scalars

r(N)	number of observations
r(N_g)	number of groups
r(N_t)	number of time periods
r(hac_lagm)	average lags used in HAC variance estimator
r(adf_lags)	lags used in ADF regressions

Macros

r(test)	kao
r(hac_kernel)	kernel used in HAC variance estimator
r(hac_method)	HAC lag-selection algorithm
r(adf_method)	ADF regression lag-selection criterion
r(demean)	demean, if the data were demeaned
r(deterministics)	constant

Matrices

r(stats)	Kao test statistics
r(p)	p-values

`xtcointtest pedroni` stores the following in `r()`:

Scalars

r(N)	number of observations
r(N_g)	number of groups
r(N_t)	number of time periods
r(hac_lagm)	average lags used in HAC variance estimator
r(adf_lags)	lags used in ADF regressions

Macros

r(test)	pedroni
r(hac_kernel)	kernel used in HAC variance estimator
r(hac_method)	HAC lag-selection algorithm
r(adf_method)	ADF regression lag-selection criterion
r(demean)	demean, if the data were demeaned
r(deterministics)	noconstant, constant, or trend

Matrices

r(stats)	Pedroni test statistics
r(p)	p-values

`xtcointtest westerlund` stores the following in `r()`:

Scalars

r(N)	number of observations
r(N_g)	number of groups
r(N_t)	number of time periods
r(stat)	Westerlund test statistic
r(p)	p-value

Macros

r(test)	westerlund
r(demean)	demean, if the data were demeaned
r(deterministics)	constant or trend

Methods and formulas

Methods and formulas are presented under the following headings:

Overview

Consider the panel-data model

$$y_{it} = \mathbf{x}'_{it}\boldsymbol{\beta}_i + \mathbf{z}'_{it}\boldsymbol{\gamma}_i + e_{it} \tag{4}$$

where $i = 1, \ldots, N$ denotes the panel and $t = 1, \ldots, T_i$ denotes time. For each i, y_{it} is a nonstationary dependent variable for which the first difference is stationary, which is to say that y_{it} is integrated of order 1—denoted $I(1)$—for each panel. Similarly, $\mathbf{x}_{it}$ is a $k \times 1$ vector of $I(1)$ variables. $\boldsymbol{\beta}_i$ denotes the cointegrating vector that may vary across panels. $\mathbf{z}_{it}$ contains terms to control for panel-specific effects and or panel-specific time trends. $\boldsymbol{\gamma}_i$ denotes the coefficients on the deterministic terms such as panel-specific means and panel-specific linear time trends. e_{it} is an error term.

The vector $\mathbf{z}_{it}$ allows for panel-specific means, panel-specific means and panel-specific time trends, or nothing, depending on the options specified to `xtcointtest`. By default, $\mathbf{z}_{it} = 1$, so the term $\mathbf{z}'_{it}\boldsymbol{\gamma}_i$ represents panel-specific means (fixed effects). If `trend` is specified, then $\mathbf{z}'_{it} = (1, t)$, so $\mathbf{z}'_{it}\boldsymbol{\gamma}_i$ represents panel-specific means and panel-specific linear time trends. For tests that allow it, specifying `noconstant` omits the $\mathbf{z}'_{it}\boldsymbol{\gamma}_i$ term.

The data-generating process for y_{it} and $\mathbf{x}_{it}$ is given by

$$\begin{aligned} y_{it} &= y_{i,t-1} + u_{it} \\ \mathbf{x}_{it} &= \mathbf{x}_{i,t-1} + \boldsymbol{\epsilon}_{it} \end{aligned}$$

Let $\mathbf{w}_{it} = (u_{it}, \boldsymbol{\epsilon}_{it})'$ denote a $(k+1) \times 1$ vector process with zero mean and long-run covariance matrix $\boldsymbol{\Omega}_i$. (A long-run covariance matrix is a covariance matrix that accounts for the serial correlation in the process; see Hall (2005, sec. 3.5) for an introduction.) The long-run matrix can be decomposed as $\boldsymbol{\Omega}_i = \boldsymbol{\Sigma}_i + \boldsymbol{\Gamma}'_i + \boldsymbol{\Gamma}_i$, where $\boldsymbol{\Sigma}_i$ and $\boldsymbol{\Gamma}_i$ denote the contemporaneous and autocovariance matrices for a given panel i. The elements of long-run and contemporaneous matrices $\boldsymbol{\Omega}_i$ and $\boldsymbol{\Sigma}_i$ are given by

$$\boldsymbol{\Omega}_i = \begin{bmatrix} \omega^2_{u,i} & \boldsymbol{\Omega}_{u\boldsymbol{\epsilon},i} \\ \boldsymbol{\Omega}'_{u\boldsymbol{\epsilon},i} & \boldsymbol{\Omega}_{\boldsymbol{\epsilon},i} \end{bmatrix}$$

$$\boldsymbol{\Sigma}_i = \begin{bmatrix} \sigma^2_{u,i} & \boldsymbol{\Sigma}_{u\boldsymbol{\epsilon},i} \\ \boldsymbol{\Sigma}'_{u\boldsymbol{\epsilon},i} & \boldsymbol{\Sigma}_{\boldsymbol{\epsilon},i} \end{bmatrix}$$

We obtain consistent estimators $\widehat{\boldsymbol{\Omega}}_i$ and $\widehat{\boldsymbol{\Sigma}}_i$ using Newey and West (1987).

Kao tests

Kao (1999) assumes the same cointegrating vector $\boldsymbol{\beta}_i = \boldsymbol{\beta}$ in (4) so that all panels share a common slope coefficient. This implies a common long-run covariance matrix given by $\boldsymbol{\Omega} = \boldsymbol{\Sigma} + \boldsymbol{\Gamma}' + \boldsymbol{\Gamma}$. The regression model is

$$y_{it} = \gamma_i + \mathbf{x}'_{it}\boldsymbol{\beta} + e_{it}$$

where γ_i denotes panel-specific fixed effects and $\boldsymbol{\beta}$ is the same cointegrating vector.

Kao (1999) proposes five test statistics. The DF t, the modified DF t, the unadjusted DF t, and the unadjusted modified DF t are based on the DF regression

$$\widehat{e}_{it} = \rho\widehat{e}_{i,t-1} + \nu_{it}$$

where ρ is the common AR parameter of the estimated residuals.

The test statistics based on DF regressions are

$$\text{DF } t = \frac{t_\rho + \frac{\sqrt{6N}\,\widehat{\sigma}_\nu}{2\widehat{\omega}_\nu}}{\sqrt{\frac{\widehat{\omega}_\nu^2}{2\widehat{\sigma}_\nu^2} + \frac{3\widehat{\sigma}_\nu^2}{10\widehat{\omega}_\nu^2}}}$$

$$\text{Modified DF } t = \frac{\sqrt{N}T(\widehat{\rho}-1) + \frac{3\sqrt{N}\,\widehat{\sigma}_\nu^2}{\widehat{\omega}_\nu^2}}{\sqrt{3 + \frac{36\widehat{\sigma}_\nu^4}{5\widehat{\omega}_\nu^4}}}$$

where $\widehat{\rho}$ is the estimated value of ρ. $\widehat{\sigma}_\nu^2$ and $\widehat{\omega}_\nu^2$ are scalar terms that are consistent estimates of $\sigma_\nu^2 = \sigma_u^2 - \boldsymbol{\Sigma}'_{u\boldsymbol{\epsilon}}\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\boldsymbol{\Sigma}_{u\boldsymbol{\epsilon}}$ and $\omega_\nu^2 = \omega_u^2 - \boldsymbol{\Omega}'_{u\boldsymbol{\epsilon}}\boldsymbol{\Omega}_{\boldsymbol{\epsilon}}\boldsymbol{\Omega}_{u\boldsymbol{\epsilon}}$. t_ρ is the t statistic for testing the null hypothesis $H_0\colon \rho = 1$.

The DF test statistics that assume strict exogeneity and absence of serial correlation are given by

$$\text{Unadjusted DF } t = \sqrt{\frac{5t_\rho}{4}} + \sqrt{\frac{15N}{8}}$$

$$\text{Unadjusted modified DF } t = \frac{\sqrt{N}T(\widehat{\rho}-1) + 3\sqrt{N}}{\sqrt{51/5}}$$

The ADF regression is given by

$$\widehat{e}_{it} = \rho\widehat{e}_{i,t-1} + \sum_{j=1}^{p}\rho_j\Delta\widehat{e}_{i,t-j} + \nu_{it}^* \tag{5}$$

where p is the number of lagged difference terms.

The test statistic based on ADF regression is

$$\text{ADF } t = \frac{t_{\text{ADF}} + \frac{\sqrt{6N}\widehat{\sigma}_\nu}{2\widehat{\omega}_\nu}}{\sqrt{\frac{\widehat{\omega}_\nu^2}{2\widehat{\sigma}_\nu^2} + \frac{3\widehat{\sigma}_\nu^2}{10\widehat{\omega}_\nu^2}}}$$

where

$$t_{\text{ADF}} = \frac{\widehat{\rho}}{\widehat{\text{SE}}(\widehat{\rho})}$$

is computed from the ADF regression.

The asymptotic distribution of all test statistics converge to $N(0,1)$.

Pedroni tests

Pedroni (1999) assumes a panel-specific cointegrating vector as in (4), where all panels have individual slope coefficients. The panel cointegration tests are obtained by testing for a unit root in the estimated residuals using the ADF regression in (5) but allowing panel-specific ρ_i instead of ρ or using the PP regressions given in Pedroni (1999).

Pedroni (1999, 2004) derives test statistics based on a model in which the AR parameter either is panel-specific or is the same over the panels. Pedroni (1999, 2004) calls the panel-specific-AR test statistics "group-mean statistics" and the same-AR test statistics "panel cointegration statistics".

The panel-specific-AR test statistics are

$$\text{Modified PP } t = TN^{-1/2}\sum_{i=1}^{N}\left(\sum_{t=1}^{T}\widehat{e}_{i,t-1}^2\right)^{-1}\sum_{t=1}^{T}(\widehat{e}_{i,t-1}\Delta\widehat{e}_{i,t} - \widehat{\lambda}_i)$$

$$\text{PP } t = N^{-1/2}\sum_{i=1}^{N}\left(\widehat{\sigma}_i^2\sum_{t=1}^{T}\widehat{e}_{i,t-1}^2\right)^{-1/2}\sum_{t=1}^{T}(\widehat{e}_{i,t-1}\Delta\widehat{e}_{i,t} - \widehat{\lambda}_i)$$

$$\text{ADF } t = N^{-1/2}\sum_{i=1}^{N}\left(\sum_{t=1}^{T}\widehat{s}_i^{*2}\widehat{e}_{i,t-1}^2\right)^{-1/2}\sum_{t=1}^{T}\widehat{e}_{i,t-1}\Delta\widehat{e}_{i,t}$$

where $\widehat{e}_{it}$ are the residuals from the panel-data regression model in (4). We calculate

$$\widehat{\lambda}_i = \frac{1}{2}(\widehat{\sigma}_i^2 - \widehat{s}_i^2)$$

where $\widehat{s}_i^2$ and $\widehat{\sigma}_i^2$ are the individual contemporaneous and long-run variances of the residuals from the DF regression in (3). $\widehat{s}_i^{*2}$ is the individual contemporaneous variance of the residuals from the ADF regression in (5) but with panel-specific ρ_i instead of ρ.

The same-AR test statistics are

$$\text{Modified VR} = T^2 N^{3/2} \left(\sum_{i=1}^{N} \sum_{t=1}^{T} \widehat{L}_{11i}^{-2} \widehat{e}_{i,t-1}^2 \right)^{-1}$$

$$\text{Modified PP } t = T\sqrt{N} \left(\sum_{i=1}^{N} \sum_{t=1}^{T} \widehat{L}_{11i}^{-2} \widehat{e}_{i,t-1}^2 \right)^{-1} \sum_{i=1}^{N} \sum_{t=1}^{T} \widehat{L}_{11i}^{-2} (\widehat{e}_{i,t-1} \Delta \widehat{e}_{i,t} - \widehat{\lambda}_i)$$

$$\text{PP } t = \left(\tilde{\sigma}_{N,T}^2 \sum_{i=1}^{N} \sum_{t=1}^{T} \widehat{L}_{11i}^{-2} \widehat{e}_{i,t-1}^2 \right)^{-1/2} \sum_{i=1}^{N} \sum_{t=1}^{T} \widehat{L}_{11i}^{-2} (\widehat{e}_{i,t-1} \Delta \widehat{e}_{i,t} - \widehat{\lambda}_i)$$

$$\text{ADF } t = \left(\tilde{s}_{N,T}^{*2} \sum_{i=1}^{N} \sum_{t=1}^{T} \widehat{L}_{11i}^{-2} \widehat{e}_{i,t-1}^2 \right)^{-1/2} \sum_{i=1}^{N} \sum_{t=1}^{T} \widehat{L}_{11i}^{-2} \widehat{e}_{i,t-1} \Delta \widehat{e}_{i,t}$$

where the residuals are as defined above and where

$$\tilde{\sigma}_{N,T}^2 = \frac{1}{N} \sum_{i=1}^{N} \widehat{L}_{11i}^{-2} \widehat{\sigma}_i^2$$

$$\widehat{L}_{11i} = \widehat{\omega}_{u,i}^2 - \widehat{\mathbf{\Omega}}_{u\boldsymbol{\epsilon},i} \widehat{\mathbf{\Omega}}_{\boldsymbol{\epsilon},i} \widehat{\mathbf{\Omega}}_{u\boldsymbol{\epsilon},i}'$$

and

$$\tilde{s}_{N,T}^{*2} = \frac{1}{N} \sum_{i=1}^{N} \widehat{s}_i^{*2}$$

The asymptotic distribution of all test statistics, after appropriate standardization, converges to $N(0,1)$. The adjustment is given by

$$\frac{\chi - \mu\sqrt{N}}{\sqrt{\nu}}$$

where χ is any of the test statistics given above, and the parameters μ and ν are the mean and variance of the test statistic obtained through simulation. Refer to Pedroni (1999) for details and an algorithm to obtain the predicted residuals. The adjusted statistics are reported in the output.

Westerlund tests

Westerlund (2005) assumes panel-specific cointegrating vectors as in (4), where all panels have individual slope coefficients. The VR test statistics are obtained by testing for a unit root in the predicted residuals using the DF regression in (3).

Westerlund (2005) derives test statistics based on a model in which the AR parameter either is panel-specific or is the same over the panels.

The panel-specific-AR test statistic is used to test the null hypothesis of no cointegration against the alternative hypothesis that some panels are cointegrated. The same-AR test statistic is used to test the null hypothesis of no cointegration against the alternative hypothesis that all the panels are cointegrated.

The panel-specific-AR test statistic is given by

$$\text{VR} = \sum_{i=1}^{N}\sum_{t=1}^{T} \widehat{E}_{it}^2 \widehat{R}_i^{-1}$$

The same-AR test statistic is given by

$$\text{VR} = \sum_{i=1}^{N}\sum_{t=1}^{T} \widehat{E}_{it}^2 \left(\sum_{i=1}^{N} \widehat{R}_i\right)^{-1}$$

where $\widehat{E}_{it} = \sum_{j=1}^{t} \widehat{e}_{ij}$, $\widehat{R}_i = \sum_{t=1}^{T} \widehat{e}_{it}^2$, and $\widehat{e}_{it}$ are the residuals from the panel-data regression model in (4). The asymptotic distribution of all test statistics, after appropriate standardization, converges to $N(0,1)$.

Long-run variance

We use the Newey and West (1987) estimator to consistently estimate the long-run variance matrix $\boldsymbol{\Omega}_i$, given by

$$\widehat{\boldsymbol{\Omega}}_i = \frac{1}{T}\sum_{t=1}^{T} \widehat{\mathbf{w}}_{it}\widehat{\mathbf{w}}_{it}' + \frac{1}{T}\sum_{j=1}^{m} K(j,m) \sum_{t=j+1}^{T} \left(\widehat{\mathbf{w}}_{it}\widehat{\mathbf{w}}_{it}' + \widehat{\mathbf{w}}_{i,t-j}\widehat{\mathbf{w}}_{it}'\right)$$

where m is the maximum number of lags and $K(j,m)$ is the kernel weight function. Define $z = j/(m+1)$. If *kernel* is `bartlett`, then

$$K(j,m) = \begin{cases} 1-z & 0 \le z \le 1 \\ 0 & \text{otherwise} \end{cases}$$

If *kspec* is `parzen`, then

$$K(j,m) = \begin{cases} 1-6z^2+6z^3 & 0 \le z \le 0.5 \\ 2(1-z)^3 & 0.5 < z \le 1 \\ 0 & \text{otherwise} \end{cases}$$

If *kernel* is `quadraticspectral`, then

$$K(j,m) = \begin{cases} 1 & z = 0 \\ 3\{\sin(\theta)/\theta - \cos(\theta)\}/\theta^2 & \text{otherwise} \end{cases}$$

where $\theta = 6\pi z/5$. If we request automatic bandwidth (lag) selection using the Newey–West algorithm, then the method documented in *Methods and formulas* of [R] **ivregress** with $\mathbf{z}_i = \mathbf{h} = 1$ is used.

References

Breitung, J. 2002. Nonparametric tests for unit roots and cointegration. *Journal of Econometrics* 108: 343–363. https://doi.org/10.1016/S0304-4076(01)00139-7.

Dickey, D. A., and W. A. Fuller. 1979. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association* 74: 427–431. https://doi.org/10.2307/2286348.

Ditzen, J. 2018. Estimating dynamic common-correlated effects in Stata. *Stata Journal* 18: 585–617.

Engle, R. F., and C. W. J. Granger. 1987. Co-integration and error correction: Representation, estimation, and testing. *Econometrica* 55: 251–276. https://doi.org/10.2307/1913236.

Hall, A. R. 2005. *Generalized Method of Moments*. Oxford: Oxford University Press.

Herwartz, H., S. Maxand, F. H. C. Raters, and Y. M. Walle. 2018. Panel unit-root tests for heteroskedastic panels. *Stata Journal* 18: 184–196.

Kao, C. 1999. Spurious regression and residual-based tests for cointegration in panel data. *Journal of Econometrics* 90: 1–44. https://doi.org/10.1016/S0304-4076(98)00023-2.

Levin, A., C.-F. Lin, and C.-S. J. Chu. 2002. Unit root tests in panel data: Asymptotic and finite-sample properties. *Journal of Econometrics* 108: 1–24. https://doi.org/10.1016/S0304-4076(01)00098-7.

Newey, W. K., and K. D. West. 1987. A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix. *Econometrica* 55: 703–708. https://doi.org/10.2307/1913610.

———. 1994. Automatic lag selection in covariance matrix estimation. *Review of Economic Studies* 61: 631–653. https://doi.org/10.2307/2297912.

Pedroni, P. 1999. Critical values for cointegration tests in heterogeneous panels with multiple regressors. *Oxford Bulletin of Economics and Statistics* 61: 653–670.

———. 2004. Panel cointegration: Asymptotic and finite sample properties of pooled time series tests with an application to the PPP hypothesis. *Econometric Theory* 20: 597–625. https://doi.org/10.1017/S0266466604203073.

Phillips, P. C. B., and H. R. Moon. 1999. Linear regression limit theory for nonstationary panel data. *Econometrica* 67: 1057–1111. https://doi.org/10.1111/1468-0262.00070.

———. 2000. Nonstationary panel data analysis: An overview of some recent developments. *Econometric Reviews* 19: 263–286. https://doi.org/10.1080/07474930008800473.

Phillips, P. C. B., and S. Ouliaris. 1990. Asymptotic properties of residual based tests for cointegration. *Econometrica* 58: 165–193. https://doi.org/10.2307/2938339.

Westerlund, J. 2005. New simple tests for panel cointegration. *Econometric Reviews* 24: 297–316. https://doi.org/10.1080/07474930500243019.

Wursten, J. 2018. Testing for serial correlation in fixed-effects panel models. *Stata Journal* 18: 76–100.

Xiao, J., A. Juodis, Y. Karavias, V. Sarafidis, and J. Ditzen. 2023. Improved tests for Granger noncausality in panel data. *Stata Journal* 23: 230–242.

Also see

[XT] **xtunitroot** — Panel-data unit-root tests

[TS] **dfgls** — DF-GLS unit-root test

[TS] **dfuller** — Augmented Dickey–Fuller unit-root test

[TS] **pperron** — Phillips–Perron unit-root test

xtdata — Faster specification searches with xt data

Description

xtdata produces a transformed dataset of the variables specified in *varlist* or of all the variables in the data. Once the data are transformed, Stata's regress command may be used to perform specification searches more quickly than xtreg; see [R] **regress** and [XT] **xtreg**. Using xtdata, re also creates a variable named constant. When using regress after xtdata, re, specify noconstant and include constant in the regression. After xtdata, be and xtdata, fe, you need not include constant or specify regress's noconstant option.

Quick start

Convert data to a form suitable for random-effects estimation using xtset data

```
xtdata, re
```

Same as above, but convert only variables v1, v2 and v3

```
xtdata v1 v2 v3, re
```

Convert all variables beginning with prefix to a form suitable for fixed-effects estimation

```
xtdata prefix*, fe
```

Convert data for between estimation if the dataset has changed since last save

```
xtdata, be clear
```

Menu

Statistics > Longitudinal/panel data > Setup and utilities > Faster specification searches with xt data

Syntax

`xtdata` [*varlist*] [*if*] [*in*] [, *options*]

options	Description
Main	
`re`	convert data to a form suitable for random-effects estimation
`ratio(#)`	ratio of random effect to pure residual (standard deviations)
`be`	convert data to a form suitable for between estimation
`fe`	convert data to a form suitable for fixed-effects (within) estimation
`nodouble`	keep original variable type; default is to recast type as `double`
`clear`	overwrite current data in memory

A panel variable must be specified; use `xtset`; see [XT] **xtset**.

Options

Main

`re` specifies that the data be converted into a form suitable for random-effects estimation. `re` is the default if `be`, `fe`, or `re` is not specified. `ratio()` must also be specified.

`ratio(#)` (use with `xtdata, re` only) specifies the ratio $\sigma_\nu/\sigma_\epsilon$, which is the ratio of the random effect to the pure residual. This is the ratio of the standard deviations, not the variances.

`be` specifies that the data be converted into a form suitable for between estimation.

`fe` specifies that the data be converted into a form suitable for fixed-effects (within) estimation.

`nodouble` specifies that transformed variables keep their original types, if possible. The default is to recast variables to `double`.

Remember that `xtdata` transforms variables to be differences from group means, pseudodifferences from group means, or group means. Specifying `nodouble` will decrease the size of the resulting dataset but may introduce roundoff errors in these calculations.

`clear` specifies that the data may be converted even though the dataset has changed since it was last saved on disk.

Remarks and examples

If you have not read [XT] **xt** and [XT] **xtreg**, please do so.

The formal estimation commands of `xtreg`—see [XT] **xtreg**—do not produce results instantaneously, especially with large datasets. Equations (2), (3), and (4) of [XT] **xtreg** describe the data necessary to fit each of the models with OLS. The idea here is to transform the data once to the appropriate form and then use `regress` to fit such models more quickly.

▷ Example 1

We will use the example in [XT] **xtreg** demonstrating between-effects regression. Another way to estimate the between equation is to convert the data in memory to the between data:

```
. use https://www.stata-press.com/data/r19/nlswork
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)

. generate age2=age^2
(24 missing values generated)

. generate ttl_exp2 = ttl_exp^2

. generate tenure2=tenure^2
(433 missing values generated)

. generate byte black = race==2

. xtdata ln_w grade age* ttl_exp* tenure* black not_smsa south, be clear

. regress ln_w grade age* ttl_exp* tenure* black not_smsa south

      Source |       SS           df       MS      Number of obs   =     4,697
-------------+----------------------------------   F(10, 4686)     =    450.23
       Model |  415.021613        10  41.5021613   Prob > F        =    0.0000
    Residual |  431.954995     4,686  .092179896   R-squared       =    0.4900
-------------+----------------------------------   Adj R-squared   =    0.4889
       Total |  846.976608     4,696  .180361288   Root MSE        =    .30361

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
       grade |   .0607602   .0020006    30.37   0.000     .0568382    .0646822
         age |   .0323158   .0087251     3.70   0.000     .0152105    .0494211
        age2 |  -.0005997   .0001429    -4.20   0.000    -.0008799   -.0003194
  (output omitted)
       south |  -.0993378    .010136    -9.80   0.000    -.1192091   -.0794665
       _cons |   .3339113   .1210434     2.76   0.006     .0966093    .5712133
------------------------------------------------------------------------------
```

The output is the same as that produced by `xtreg, be`; the reported R^2 is the R^2 between. Using `xtdata` followed by just one `regress` does not save time. Using `xtdata` is justified when you intend to explore the specification of the model by running many alternative regressions.

◁

❑ Technical note

When using `xtdata`, you must eliminate any variables that you do not intend to use and that have missing values. `xtdata` follows a casewise-deletion rule, which means that an observation is excluded from the conversion if it is missing on any of the variables. In the example above, we specified that the variables be converted on the command line. We could also drop the variables first, and it might even be useful to preserve our estimation sample:

```
. use https://www.stata-press.com/data/r19/nlswork, clear
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)

. generate age2=age^2
(24 missing values generated)

. generate ttl_exp2 = ttl_exp^2

. generate tenure2=tenure^2
(433 missing values generated)

. generate byte black = race==2

. keep id year ln_w grade age* ttl_exp* tenure* black not_smsa south

. save xtdatasmpl
file xtdatasmpl.dta saved
```

❑

▷ Example 2

xtdata with the fe option converts the data so that results are equivalent to those from estimating by using xtreg with the fe option.

```
. xtdata, fe
. regress ln_w grade age* ttl_exp* tenure* black not_smsa south
note: grade omitted because of collinearity.
note: black omitted because of collinearity.
      Source |       SS           df       MS      Number of obs   =    28,091
-------------+----------------------------------   F(8, 28082)     =    732.64
       Model |  412.443881         8  51.5554852   Prob > F        =    0.0000
    Residual |  1976.12232    28,082  .070369714   R-squared       =    0.1727
-------------+----------------------------------   Adj R-squared   =    0.1724
       Total |   2388.5662    28,090  .085032617   Root MSE        =    .26527

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
       grade |          0  (omitted)
         age |   .0359987   .0030903    11.65   0.000     .0299415    .0420558
        age2 |   -.000723   .0000486   -14.88   0.000    -.0008183   -.0006277
     ttl_exp |   .0334668   .0027061    12.37   0.000     .0281627    .0387708
    ttl_exp2 |   .0002163   .0001166     1.86   0.064    -.0000122    .0004447
      tenure |   .0357539   .0016871    21.19   0.000     .0324472    .0390606
     tenure2 |  -.0019701   .0001141   -17.27   0.000    -.0021937   -.0017465
       black |          0  (omitted)
    not_smsa |  -.0890108   .0086982   -10.23   0.000    -.1060597   -.0719619
       south |  -.0606309   .0099761    -6.08   0.000    -.0801845   -.0410772
       _cons |    1.03732   .0443093    23.41   0.000     .9504716    1.124168
------------------------------------------------------------------------------
```

The coefficients reported by regress after xtdata, fe are the same as those reported by xtreg, fe, but the standard errors are slightly smaller. This is because no adjustment has been made to the estimated covariance matrix for the estimation of the person means. The difference is small, however, and results are adequate for a specification search.

◁

▷ Example 3

To use xtdata, re, you must specify the ratio $\sigma_\nu/\sigma_\epsilon$, which is the ratio of the standard deviations of the random effect and pure residual. Merely to show the relationship of regress after xtdata, re to xtreg, re, we will specify this ratio as $0.25790526/0.29068923 = 0.88721987$, which is the number xtreg reports when the model is fit from the outset; see the random-effects example in [XT] **xtreg**. For specification searches, however, it is adequate to specify this number more crudely, and, when performing the specification search for this manual entry, we used ratio(1).

```
. use https://www.stata-press.com/data/r19/xtdatasmpl, clear
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)
. xtdata, clear re ratio(.88721987)
------------------- theta --------------------
  min      5%       median        95%      max
0.2520   0.2520     0.5499     0.7016   0.7206
```

xtdata reports the distribution of θ based on the specified ratio. If these were balanced data, θ would have been constant.

When running regressions with these data, you must specify the noconstant option and include the variable constant:

```
. regress ln_w grade age* ttl_exp* tenure* black not_smsa south constant,
> noconstant

      Source |       SS           df       MS      Number of obs   =    28,091
-------------+----------------------------------   F(11, 28080)    =  14302.56
       Model |  13271.7208        11  1206.52007   Prob > F        =    0.0000
    Residual |  2368.74223    28,080  .084356917   R-squared       =    0.8486
-------------+----------------------------------   Adj R-squared   =    0.8485
       Total |   15640.463    28,091  .556778435   Root MSE        =    .29044

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
       grade |   .0646499   .0017812    36.30   0.000     .0611587    .0681411
         age |   .0368059   .0031195    11.80   0.000     .0306915    .0429203
        age2 |  -.0007133     .00005   -14.27   0.000    -.0008113   -.0006153
  (output omitted)
       south |  -.0868922   .0073032   -11.90   0.000    -.1012068   -.0725775
    constant |   .2387206    .049469     4.83   0.000      .141759    .3356822
------------------------------------------------------------------------------
```

Results are the same coefficients and standard errors that xtreg, re estimated in example 4 of [XT] **xtreg**. The summaries at the top, however, should be ignored, as they are expressed in terms of (4) of [XT] **xtreg**, and, moreover, for a model without a constant.

◁

❑ Technical note

Using xtdata requires some caution. The following guidelines may help:

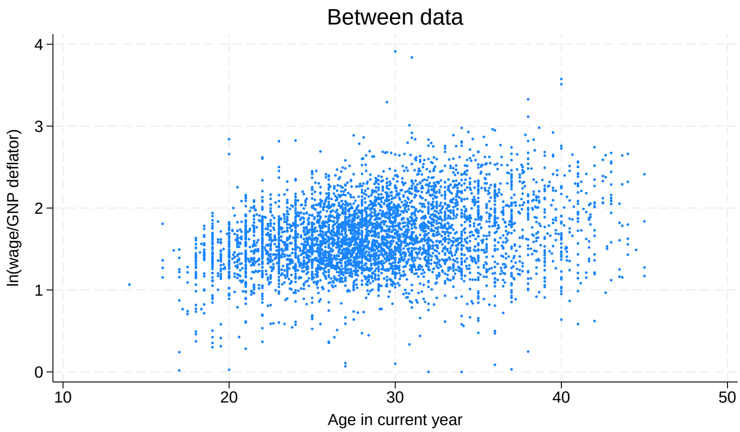
1. xtdata is intended for use only during the specification search phase of analysis. Results should be estimated with xtreg on unconverted data.
2. After converting the data, you may use regress to obtain estimates of the coefficients and their standard errors. For regress after xtdata, fe, the standard errors are too small, but only slightly.
3. You may loosely interpret the coefficient's significance tests and confidence intervals. However, for results after xtdata, fe and re, an incorrect (but close to correct) distribution is assumed.
4. You should ignore the summary statistics reported at the top of regress's output.
5. After converting the data, you may form linear, but not nonlinear, combinations of regressors; that is, if your data contained age, it would not be correct to convert the data and then form age squared. All nonlinear transformations should be done before conversion. (For xtdata, be, you can get away with forming nonlinear combinations ex post, but the results will not be exact.)

❑

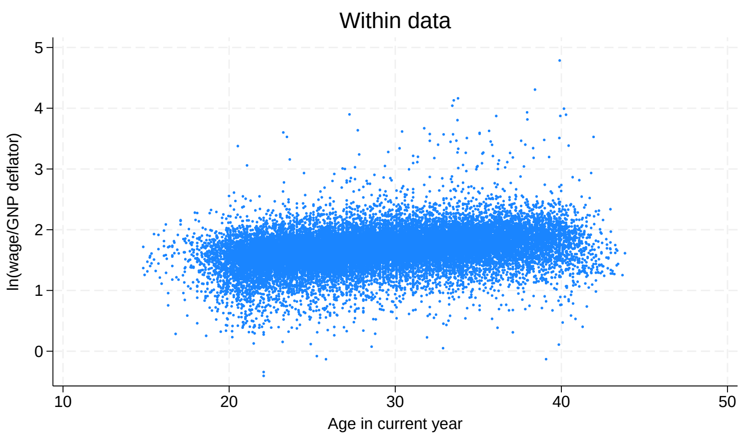
❑ Technical note

The `xtdata` command can be used to help you examine data, especially with `scatter`.

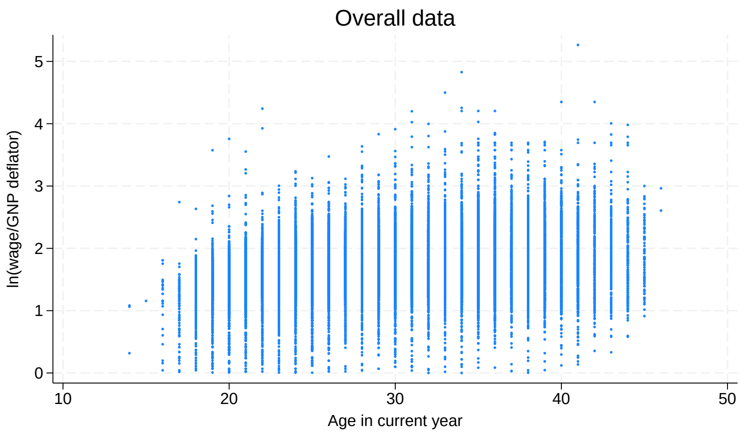
```
. use https://www.stata-press.com/data/r19/xtdatasmpl, clear
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)
. xtdata, be
. scatter ln_wage age, title(Between data) msymbol(o) msize(tiny)
```




```
. use https://www.stata-press.com/data/r19/xtdatasmpl, clear
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)
. xtdata, fe
. scatter ln_wage age, title(Within data) msymbol(o) msize(tiny)
```



```
. use https://www.stata-press.com/data/r19/xtdatasmpl, clear
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)
. scatter ln_wage age, title(Overall data) msymbol(o) msize(tiny)
```



❑

Methods and formulas

(This section is a continuation of the *Methods and formulas* of [XT] **xtreg**.)

`xtdata, be`, `fe`, and `re` transform the data according to (2), (3), and (4), respectively, of [XT] **xtreg**, except that `xtdata, fe` adds back in the overall mean, thus forming the transformation

$$\mathbf{x}_{it} - \overline{x}_i + \overline{\overline{x}}$$

`xtdata, re` requires the user to specify r as an estimate of $\sigma_\nu/\sigma_\epsilon$. θ_i is calculated from

$$\theta_i = 1 - \frac{1}{\sqrt{T_i r^2 + 1}}$$

Also see

[XT] **xtsum** — Summarize xt data

xtdescribe — Describe pattern of xt data

Description

`xtdescribe` describes the participation pattern of cross-sectional time-series (xt) data.

Quick start

Describe the 9 most common participation patterns of `xtset` data

```
xtdescribe
```

Describe up to 15 of the most common participation patterns

```
xtdescribe, patterns(15)
```

Same as above, but list all participation patterns

```
xtdescribe, patterns(1000)
```

Describe patterns only for study subjects, denoted by binary variable `insample` = 1

```
xtdescribe if insample
```

Menu

Statistics > Longitudinal/panel data > Setup and utilities > Describe pattern of xt data

Syntax

`xtdescribe` [*if*] [*in*] [, *options*]

options	Description
Main	
`patterns(#)`	maximum participation patterns; default is `patterns(9)`
`width(#)`	display # width of participation patterns; default is `width(100)`

A panel variable and a time variable must be specified; use `xtset`; see [XT] **xtset**.
`by` is allowed; see [D] **by**.

Options

Main

`patterns(#)` specifies the maximum number of participation patterns to be reported; `patterns(9)` is the default. Specifying `patterns(50)` would list up to 50 patterns. Specifying `patterns(1000)` is taken to mean `patterns(∞)`: all the patterns will be listed.

`width(#)` specifies the desired width of the participation patterns to be displayed; `width(100)` is the default. If the number of times is greater than `width()`, then each column in the participation pattern represents multiple periods as indicated in a footnote at the bottom of the table. The actual width may differ slightly from the requested width depending on the span of the time variable and the number of periods.

Remarks and examples

If you have not read [XT] **xt**, please do so.

`xtdescribe` describes the cross-sectional and time-series aspects of the data in memory.

▷ Example 1

In [XT] **xt**, we introduced data based on a subsample of the NLSY data on young women aged 14–24 years in 1968. Here is a description of the data used in many of the [XT] **xt** examples:

```
. use https://www.stata-press.com/data/r19/nlswork
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)

. xtdescribe

  idcode:  1, 2, ..., 5159                                  n =       4711
    year:  68, 69, ..., 88                                  T =         15
           Delta(year) = 1 unit
           Span(year)  = 21 periods
           (idcode*year uniquely identifies each observation)

Distribution of T_i:   min      5%     25%       50%       75%     95%     max
                         1       1       3         5         9      13      15

     Freq.  Percent    Cum. |  Pattern
 ---------------------------+-----------------------
      136     2.89    2.89  |  1....................
      114     2.42    5.31  |  ....................1
       89     1.89    7.20  |  .................1.11
       87     1.85    9.04  |  ...................11
       86     1.83   10.87  |  111111.1.11.1.11.1.11
       61     1.29   12.16  |  ..............11.1.11
       56     1.19   13.35  |  11...................
       54     1.15   14.50  |  ...............1.1.11
       54     1.15   15.64  |  .......1.11.1.11.1.11
     3974    84.36  100.00  | (other patterns)
 ---------------------------+-----------------------
     4711   100.00          |  XXXXXX.X.XX.X.XX.X.XX
```

`xtdescribe` tells us that we have 4,711 women in our data and that the `idcode` that identifies each ranges from 1 to 5,159. We are also told that the maximum number of individual years over which we observe any woman is 15, though the `year` variable spans 21 years. The delta or periodicity of `year` is one unit, meaning that in principle we could observe each woman yearly. We are reassured that `idcode` and `year`, taken together, uniquely identify each observation in our data. We are also shown the distribution of T_i; 50% of our women are observed 5 years or less. Only 5% of our women are observed for 13 years or more.

Finally, we are shown the participation pattern. A 1 in the pattern means one observation that year; a dot means no observation. The largest fraction of our women (still only 2.89%) was observed in the single year 1968 and not thereafter; the next largest fraction was observed in 1988 but not before; and the next largest fraction was observed in 1985, 1987, and 1988.

At the bottom is the sum of the participation patterns, including the patterns that were not shown. We can see that none of the women were observed in six of the years (there are six dots). (The survey was not administered in those six years.)

We could see more of the patterns by specifying the `patterns()` option, or we could see all the patterns by specifying `patterns(1000)`.

◁

▷ Example 2

The strange participation patterns shown above have to do with our subsampling of the data, not with the administrators of the survey. Here are the data from which we drew the sample used in [XT] **xt**:

```
. xtdescribe

 idcode:  1, 2, ..., 5159                                    n =       5159
   year:  68, 69, ..., 88                                    T =         15
          Delta(year) = 1; (88-68)+1 = 21
          (idcode*year does not uniquely identify observations)

Distribution of T_i:   min      5%      25%       50%       75%     95%     max
                         1       2       11        15        16      19      30

     Freq.  Percent    Cum. |  Pattern
 ---------------------------+-----------------------
     1034     20.04   20.04 |  111111.1.11.1.11.1.11
      153      2.97   23.01 |  1....................
      147      2.85   25.86 |  112111.1.11.1.11.1.11
      130      2.52   28.38 |  111112.1.11.1.11.1.11
      122      2.36   30.74 |  111211.1.11.1.11.1.11
      113      2.19   32.93 |  11...................
       84      1.63   34.56 |  111111.1.11.1.11.1.12
       79      1.53   36.09 |  111111.1.12.1.11.1.11
       67      1.30   37.39 |  111111.1.11.1.11.1.1.
     3230     62.61  100.00 | (other patterns)
 ---------------------------+-----------------------
     5159    100.00         |  XXXXXX.X.XX.X.XX.X.XX
```

We have multiple observations per year. In the pattern, 2 indicates that a woman appears twice in the year, 3 indicates 3 times, and so on—`X` indicates 10 or more, should that be necessary.

In fact, this is a dataset that was itself extracted from the NLSY, in which t is not time but job number. To simplify exposition, we made a simpler dataset by selecting the last job in each year.

◁

▷ Example 3

When the number of periods is greater than the width of the participation pattern, each column will represent more than one period.

```
. use https://www.stata-press.com/data/r19/xtdesxmpl

. xtdescribe, width(16)

 patient:  1, 2, ..., 30                                  n =          30
    time:  09mar2007 16:00:00, 09mar2007 17:00:00, ...,   T =          32
           10mar2007 23:00:00
           Delta(time) = 1 hour
           Span(time)  = 32 periods
           (patient*time uniquely identifies each observation)

Distribution of T_i:   min      5%     25%       50%       75%     95%     max
                        30      30      31        32        32      32      32

     Freq.  Percent    Cum. |  Pattern*
 ---------------------------+------------------
       21     70.00   70.00 |  2222222222222222
        4     13.33   83.33 |  1222222222222222
        3     10.00   93.33 |  222222222222222.
        2      6.67  100.00 |  .222222222222222
 ---------------------------+------------------
       30    100.00         |  XXXXXXXXXXXXXXXX
 ----------------------------------------------
 *Each column represents 2 periods.
```

We have data for 30 patients who were observed hourly between 4:00 PM on March 9, 2007, and 11:00 PM on March 10, a span of 32 hours. We have complete records for 21 of the patients. The footnote indicates that each column in the pattern represents two periods, so for four patients we have an observation taken at either 4:00 PM or 5:00 PM on March 9, but we do not have observations for both times. There are three patients for whom we are missing both the 10:00 PM and 11:00 PM observations on March 10, and there are two patients for whom we are missing the 4:00 PM and 5:00 PM observations for March 9.

◁

Reference

Cox, N. J. 2007. Speaking Stata: Counting groups, especially panels. *Stata Journal* 7: 571–581.

Also see

[XT] **xtsum** — Summarize xt data

[XT] **xttab** — Tabulate xt data

xtdidregress — Fixed-effects difference-in-differences estimation

Description

xtdidregress estimates the average treatment effect on the treated (ATET) from observational data by difference in differences (DID) or difference in difference in differences (DDD) for panel data. The ATET of a binary or continuous treatment on a continuous outcome is estimated by fitting a linear model with time and individual (panel) fixed effects.

Quick start

DID estimate of the ATET of treat1 on outcome y1 using xtset data; y1 modeled using covariates x1 and x2, and individual (panel) and tvar fixed effects, with the treatment occurring at the grpvar1 and tvar levels

```
xtdidregress (y1 x1 x2) (treat1), group(grpvar1) time(tvar)
```

Same as above, but compute wild cluster–bootstrap p-values and confidence intervals with grpvar1 as the clustering variable

```
xtdidregress (y1 x1 x2) (treat1), group(grpvar1) time(tvar) ///
    wildbootstrap
```

Same as above, but aggregate data at the grpvar1 and tvar levels to use the Donald and Lang (2007) method to compute the ATET and standard errors

```
xtdidregress (y1 x1 x2) (treat1), group(grpvar1) time(tvar) ///
    aggregate(dlang)
```

Aggregate data at the grpvar1 and tvar1 levels to estimate the ATET

```
xtdidregress 1(y x1 x2) (grpvar1), group(state) time(tvar1) ///
    aggregate(standard)
```

Menu

Statistics > Longitudinal/panel data > Difference in differences (DID) > DID (FE)

Syntax

For syntax, methods, and all other information on xtdidregress, see [CAUSAL] **didregress**.

Reference

Donald, S. G., and K. Lang. 2007. Inference with difference-in-differences and other panel data. *Review of Economics and Statistics* 89: 221–233. https://doi.org/10.1162/rest.89.2.221.

xtdpd — Linear dynamic panel-data estimation

Description

xtdpd fits a linear dynamic panel-data model where the unobserved panel-level effects are correlated with the lags of the dependent variable. The command can fit Arellano–Bond and Arellano–Bover/Blundell–Bond models like those fit by xtabond and xtdpdsys. However, it also allows models with low-order moving-average correlation in the idiosyncratic errors or predetermined variables with a more complicated structure than allowed for xtabond or xtdpdsys.

Quick start

Arellano–Bond model of y on L.y and x with the first difference of x as an instrument for the difference equation using xtset data

```
xtdpd y L.y x, div(x) dgmmiv(y)
```

Add the first difference of the lag of x as an instrument for the level equation

```
xtdpd y L.y x, div(x) dgmmiv(y) lgmmiv(x)
```

Use lags 3 to 5 of x as instruments for the difference equation

```
xtdpd y L.y x, div(x) dgmmiv(y, lagrange(3 5))
```

Menu

Statistics > Longitudinal/panel data > Dynamic panel data (DPD) > Linear DPD estimation

Syntax

xtdpd *depvar* [*indepvars*] [*if*] [*in*], dgmmiv(*varlist* [...]) [*options*]

options	Description
Model	
* dgmmiv(*varlist*[...])	GMM-type instruments for the difference equation; can be specified more than once
lgmmiv(*varlist*[...])	GMM-type instruments for the level equation; can be specified more than once
iv(*varlist*[...])	standard instruments for the difference and level equations; can be specified more than once
div(*varlist*[...])	standard instruments for the difference equation only; can be specified more than once
liv(*varlist*)	standard instruments for the level equation only; can be specified more than once
noconstant	suppress constant term
twostep	compute the two-step estimator instead of the one-step estimator
hascons	check for collinearity only among levels of independent variables; by default checks occur among levels and differences
fodeviation	use forward-orthogonal deviations instead of first differences
SE/Robust	
vce(*vcetype*)	*vcetype* may be gmm or robust
Reporting	
level(#)	set confidence level; default is level(95)
artests(#)	use # as maximum order for AR tests; default is artests(2)
display_options	control spacing and line width
coeflegend	display legend instead of statistics

*dgmmiv() is required.
A panel variable and a time variable must be specified; use xtset; see [XT] **xtset**.
depvar, *indepvars*, and all *varlists* may contain time-series operators; see [U] **11.4.4 Time-series varlists**.
by, collect, statsby, and xi are allowed; see [U] **11.1.10 Prefix commands**.
coeflegend does not appear in the dialog box.
See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

dgmmiv(*varlist* [, lagrange(*flag* [*llag*])]) specifies GMM-type instruments for the difference equation. Levels of the variables are used to form GMM-type instruments for the difference equation. All possible lags are used, unless lagrange(*flag llag*) restricts the lags to begin with *flag* and end with *llag*. You may specify as many sets of GMM-type instruments for the difference equation as you need within the standard Stata limits on matrix size. Each set may have its own *flag* and *llag*. dgmmiv() is required.

`lgmmiv(`*varlist* [, `lag(#)`]`)` specifies GMM-type instruments for the level equation. Differences of the variables are used to form GMM-type instruments for the level equation. The first lag of the differences is used unless `lag(#)` is specified, indicating that #th lag of the differences be used. You may specify as many sets of GMM-type instruments for the level equation as you need within the standard Stata limits on matrix size. Each set may have its own *lag*.

`iv(`*varlist* [, `nodifference`]`)` specifies standard instruments for both the differenced and level equations. Differences of the variables are used as instruments for the differenced equations, unless `nodifference` is specified, which requests that levels be used. Levels of the variables are used as instruments for the level equations. You may specify as many sets of standard instruments for both the differenced and level equations as you need within the standard Stata limits on matrix size.

`div(`*varlist* [, `nodifference`]`)` specifies additional standard instruments for the difference equation. Specified variables may not be included in `iv()` or in `liv()`. Differences of the variables are used, unless `nodifference` is specified, which requests that levels of the variables be used as instruments for the difference equation. You may specify as many additional sets of standard instruments for the difference equation as you need within the standard Stata limits on matrix size.

`liv(`*varlist*`)` specifies additional standard instruments for the level equation. Specified variables may not be included in `iv()` or in `div()`. Levels of the variables are used as instruments for the level equation. You may specify as many additional sets of standard instruments for the level equation as you need within the standard Stata limits on matrix size.

`noconstant`; see [R] **Estimation options**.

`twostep` specifies that the two-step estimator be calculated.

`hascons` specifies that `xtdpd` check for collinearity only among levels of independent variables; by default checks occur among levels and differences. This option should typically be specified when time-invariant covariates or time-indicator variables are included in the level equation.

`fodeviation` specifies that forward-orthogonal deviations be used instead of first differences. `fodeviation` is not allowed when there are gaps in the data or when `lgmmiv()` is specified.

SE/Robust

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory and that are robust to some kinds of misspecification; see *Methods and formulas*.

`vce(gmm)`, the default, uses the conventionally derived variance estimator for generalized method of moments estimation.

`vce(robust)` uses the robust estimator. For the one-step estimator, this is the Arellano–Bond robust VCE estimator. For the two-step estimator, this is the Windmeijer (2005) WC-robust estimator.

Reporting

`level(#)`; see [R] **Estimation options**.

`artests(#)` specifies the maximum order of the autocorrelation test to be calculated. The tests are reported by `estat abond`; see [XT] **xtdpd postestimation**. Specifying the order of the highest test at estimation time is more efficient than specifying it to `estat abond`, because `estat abond` must refit the model to obtain the test statistics. The maximum order must be less than or equal to the number of periods in the longest panel. The default is `artests(2)`.

display_options: `vsquish` and `nolstretch`; see [R] **Estimation options**.

The following option is available with xtdpd but is not shown in the dialog box:

coeflegend; see [R] **Estimation options**.

Remarks and examples

If you have not read [XT] **xtabond** and [XT] **xtdpdsys**, you should do so before continuing.

Linear dynamic panel-data models include p lags of the dependent variable as covariates and contain unobserved panel-level effects, fixed or random. By construction, the unobserved panel-level effects are correlated with the lagged dependent variables, making standard estimators inconsistent. xtdpd fits a dynamic panel-data model by using the Arellano–Bond (1991) or the Arellano–Bover/Blundell–Bond (1995, 1998) estimator.

Consider the dynamic panel-data model

$$y_{it} = \sum_{j=1}^{p} \alpha_j y_{i,t-j} + \mathbf{x}_{it}\boldsymbol{\beta}_1 + \mathbf{w}_{it}\boldsymbol{\beta}_2 + \nu_i + \epsilon_{it} \qquad i = \{1,\ldots,N\};\;\; t = \{1,\ldots,T_i\} \tag{1}$$

where

the $\alpha_1, \ldots, \alpha_p$ are p parameters to be estimated,

$\mathbf{x}_{it}$ is a $1 \times k_1$ vector of strictly exogenous covariates,

$\boldsymbol{\beta}_1$ is a $k_1 \times 1$ vector of parameters to be estimated,

$\mathbf{w}_{it}$ is a $1 \times k_2$ vector of predetermined covariates,

$\boldsymbol{\beta}_2$ is a $k_2 \times 1$ vector of parameters to be estimated,

ν_i are the panel-level effects (which may be correlated with x_{it} or w_{it}), and

and ϵ_{it} are i.i.d. or come from a low-order moving-average process, with variance σ^2_ϵ.

Building on the work of Anderson and Hsiao (1981, 1982) and Holtz-Eakin, Newey, and Rosen (1988), Arellano and Bond (1991) derived one-step and two-step GMM estimators using moment conditions in which lagged levels of the dependent and predetermined variables were instruments for the difference equation. Blundell and Bond (1998) show that the lagged-level instruments in the Arellano–Bond estimator become weak as the autoregressive process becomes too persistent or the ratio of the variance of the panel-level effect ν_i to the variance of the idiosyncratic error ϵ_{it} becomes too large. Building on the work of Arellano and Bover (1995), Blundell and Bond (1998) proposed a system estimator that uses moment conditions in which lagged differences are used as instruments for the level equation in addition to the moment conditions of lagged levels as instruments for the difference equation. The additional moment conditions are valid only if the initial condition $E[\nu_i \Delta y_{i2}] = 0$ holds for all i; see Blundell and Bond (1998) and Blundell, Bond, and Windmeijer (2000).

xtdpd fits dynamic panel-data models by using the Arellano–Bond or the Arellano–Bover/Blundell–Bond system estimator. The parameters of many standard models can be more easily estimated using the Arellano–Bond estimator implemented in xtabond or using the Arellano–Bover/Blundell–Bond system estimator implemented in xtdpdsys; see [XT] **xtabond** and [XT] **xtdpdsys**. xtdpd can fit more complex models at the cost of a more complicated syntax. That the idiosyncratic errors follow a low-order MA process and that the predetermined variables have a more complicated structure than accommodated by xtabond and xtdpdsys are two common reasons for using xtdpd instead of xtabond or xtdpdsys.

The standard GMM robust two-step estimator of the VCE is known to be seriously biased. Windmeijer (2005) derived a bias-corrected robust estimator for two-step VCEs from GMM estimators known as the WC-robust estimator, which is implemented in `xtdpd`.

The Arellano–Bond test of autocorrelation of order m and the Sargan test of overidentifying restrictions derived by Arellano and Bond (1991) are computed by `xtdpd` but reported by `estat abond` and `estat sargan`, respectively; see [XT] **xtdpd postestimation**.

Because `xtdpd` extends `xtabond` and `xtdpdsys`, [XT] **xtabond** and [XT] **xtdpdsys** provide useful background.

▷ Example 1: An Arellano–Bond estimator

Arellano and Bond (1991) apply their new estimators and test statistics to a model of dynamic labor demand that had previously been considered by Layard and Nickell (1986), using data from an unbalanced panel of firms from the United Kingdom. All variables are indexed over the firm i and time t. In this dataset, $\mathtt{n}_{it}$ is the log of employment in firm i inside the United Kingdom at time t, $\mathtt{w}_{it}$ is the natural log of the real product wage, $\mathtt{k}_{it}$ is the natural log of the gross capital stock, and $\mathtt{ys}_{it}$ is the natural log of industry output. The model also includes time dummies `yr1980`, `yr1981`, `yr1982`, `yr1983`, and `yr1984`. To gain some insight into the syntax for `xtdpd`, we reproduce the first example from [XT] **xtabond** using `xtdpd`:

```
. use https://www.stata-press.com/data/r19/abdata
. xtdpd L(0/2).n L(0/1).w L(0/2).(k ys) yr1980-yr1984 year, noconstant
> div(L(0/1).w L(0/2).(k ys) yr1980-yr1984 year) dgmmiv(n)
Dynamic panel-data estimation                   Number of obs      =       611
Group variable: id                              Number of groups   =       140
Time variable: year
                                                Obs per group:
                                                              min =          4
                                                              avg =   4.364286
                                                              max =          6
Number of instruments =     41                  Wald chi2(16)      =   1757.07
                                                Prob > chi2        =    0.0000
One-step results
------------------------------------------------------------------------------
           n | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
           n |
         L1. |   .6862261   .1486163     4.62   0.000     .3949435    .9775088
         L2. |  -.0853582   .0444365    -1.92   0.055    -.1724523    .0017358
             |
           w |
         --. |  -.6078208   .0657694    -9.24   0.000    -.7367265   -.4789151
         L1. |   .3926237   .1092374     3.59   0.000     .1785222    .6067251
             |
           k |
         --. |   .3568456   .0370314     9.64   0.000     .2842653    .4294259
         L1. |  -.0580012   .0583051    -0.99   0.320     -.172277    .0562747
         L2. |  -.0199475   .0416274    -0.48   0.632    -.1015357    .0616408
             |
          ys |
         --. |   .6085073   .1345412     4.52   0.000     .3448115    .8722031
         L1. |  -.7111651   .1844599    -3.86   0.000      -1.0727   -.3496304
         L2. |   .1057969   .1428568     0.74   0.459    -.1741974    .3857912
             |
      yr1980 |   .0029062   .0212705     0.14   0.891    -.0387832    .0445957
      yr1981 |  -.0404378   .0354707    -1.14   0.254    -.1099591    .0290836
      yr1982 |  -.0652767    .048209    -1.35   0.176    -.1597646    .0292111
      yr1983 |  -.0690928   .0627354    -1.10   0.271    -.1920521    .0538664
      yr1984 |  -.0650302   .0781322    -0.83   0.405    -.2181665    .0881061
        year |   .0095545   .0142073     0.67   0.501    -.0182912    .0374002
------------------------------------------------------------------------------
Instruments for differenced equation
        GMM-type: L(2/.).n
        Standard: D.w LD.w D.k LD.k L2D.k D.ys LD.ys L2D.ys D.yr1980
                  D.yr1981 D.yr1982 D.yr1983 D.yr1984 D.year
```

Unlike most instrumental-variables estimation commands, the independent variables in the varlist are not automatically used as instruments. In this example, all the independent variables are strictly exogenous, so we include them in `div()`, a list of variables whose first differences will be instruments for the difference equation. We include the dependent variable in `dgmmiv()`, a list of variables whose lagged levels will be used to create GMM-type instruments for the difference equation. (GMM-type instruments are discussed in a technical note below.)

The footer in the output reports the instruments used. The first line indicates that `xtdpd` used lags from 2 on back to create the GMM-type instruments described in Arellano and Bond (1991) and Holtz-Eakin, Newey, and Rosen (1988). The second line says that the first difference of all the variables included in the `div()` varlist were used as standard instruments for the difference equation.

◁

❑ Technical note

GMM-type instruments are built from lags of one variable. Ignoring the strictly exogenous variables for simplicity, our model is

$$n_{it} = \alpha_1 n_{it-1} + \alpha_2 n_{it-2} + \nu_i + \epsilon_{it} \tag{2}$$

After differencing we have

$$\Delta n_{it} = \Delta\alpha_1 n_{it-1} + \Delta\alpha_2 n_{it-2} + \Delta\epsilon_{it} \tag{3}$$

Equation (3) implies that we need instruments that are not correlated with either ϵ_{it} or ϵ_{it-1}. Equation (2) shows that `L2.n` is the first lag of `n` that is not correlated with ϵ_{it} or ϵ_{it-1}, so it is the first lag of `n` that can be used to instrument the difference equation.

Consider the following data from one of the complete panels in the previous example:

```
. list id year n L2.n dl2.n if id==140
```

	id	year	n	L2. n	L2D. n
1023.	140	1976	.4324315	.	.
1024.	140	1977	.3694925	.	.
1025.	140	1978	.3541718	.4324315	.
1026.	140	1979	.3632532	.3694925	-.0629391
1027.	140	1980	.3371863	.3541718	-.0153207
1028.	140	1981	.285179	.3632532	.0090815
1029.	140	1982	.1756326	.3371863	-.026067
1030.	140	1983	.1275133	.285179	-.0520073
1031.	140	1984	.0889263	.1756326	-.1095464

The missing values in `L2D.n` show that we lose 3 observations because of lags and the difference that removes the panel-level effects. The first nonmissing observation occurs in 1979 and observations on `n` from 1976 and 1977 are available to instrument the 1979 difference equation. The table below gives the observations available to instrument the differenced equation for the data above.

Year of difference errors	Years of instruments	Number of instruments
1979	1976–1977	2
1980	1976–1978	3
1981	1976–1979	4
1982	1976–1980	5
1983	1976–1981	6
1984	1976–1982	7

The table shows that there are a total of 27 GMM-type instruments.

The output in the example above informs us that there were a total of 41 instruments applied to the difference equation. Because there are 14 standard instruments, there must have been 27 GMM-type instruments, which matches our above calculation.

❑

▷ Example 2: An Arellano–Bond estimator with predetermined variables

Sometimes we cannot assume strict exogeneity. Recall that a variable x_{it} is said to be strictly exogenous if $E[x_{it}\epsilon_{is}] = 0$ for all t and s. If $E[x_{it}\epsilon_{is}] \neq 0$ for $s < t$ but $E[x_{it}\epsilon_{is}] = 0$ for all $s \geq t$, the variable is said to be predetermined. Intuitively, if the error term at time t has some feedback on the subsequent realizations of x_{it}, x_{it} is a predetermined variable. In the output below, we use `xtdpd` to reproduce example 6 in [XT] **xtabond**.

```
. xtdpd L(0/2).n L(0/1).(w ys) L(0/2).k yr1980-yr1984 year,
> div(L(0/1).(ys) yr1980-yr1984 year) dgmmiv(n) dgmmiv(L.w L2.k, lag(1 .))
> twostep noconstant vce(robust)
Dynamic panel-data estimation                   Number of obs      =        611
Group variable: id                              Number of groups   =        140
Time variable: year
                                                Obs per group:
                                                             min =          4
                                                             avg =   4.364286
                                                             max =          6
Number of instruments =     83                  Wald chi2(15)      =     958.30
                                                Prob > chi2        =     0.0000
Two-step results
                                     (Std. err. adjusted for clustering on id)
------------------------------------------------------------------------------
             |              WC-robust
           n | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
           n |
         L1. |   .8580958   .1265515     6.78   0.000     .6100594    1.106132
         L2. |   -.081207   .0760703    -1.07   0.286    -.2303022    .0678881
             |
           w |
         --. |  -.6910855   .1387684    -4.98   0.000    -.9630666   -.4191044
         L1. |   .5961712   .1497338     3.98   0.000     .3026982    .8896441
             |
          ys |
         --. |   .6936392   .1728623     4.01   0.000     .3548354    1.032443
         L1. |  -.8773678   .2183085    -4.02   0.000    -1.305245    -.449491
             |
           k |
         --. |   .4140654   .1382788     2.99   0.003     .1430439    .6850868
         L1. |  -.1537048   .1220244    -1.26   0.208    -.3928681    .0854586
         L2. |  -.1025833   .0710886    -1.44   0.149    -.2419143    .0367477
             |
      yr1980 |  -.0072451    .017163    -0.42   0.673    -.0408839    .0263938
      yr1981 |  -.0609608    .030207    -2.02   0.044    -.1201655   -.0017561
      yr1982 |  -.1130369   .0454826    -2.49   0.013    -.2021812   -.0238926
      yr1983 |  -.1335249   .0600213    -2.22   0.026    -.2511645   -.0158853
      yr1984 |  -.1623177   .0725434    -2.24   0.025    -.3045001   -.0201352
        year |   .0264501   .0119329     2.22   0.027      .003062    .0498381
------------------------------------------------------------------------------
Instruments for differenced equation
        GMM-type: L(2/.).n L(1/.).L.w L(1/.).L2.k
        Standard: D.ys LD.ys D.yr1980 D.yr1981 D.yr1982 D.yr1983 D.yr1984
                  D.year
```

The footer informs us that we are now including GMM-type instruments from the first lag of `L.w` on back and from the first lag of `L2.k` on back.

◁

▷ Example 3: A weaker definition of predetermined variables

As discussed in [XT] **xtabond** and [XT] **xtdpdsys**, `xtabond` and `xtdpdsys` both use a strict definition of predetermined variables with lags. In the strict definition, the most recent lag of the variable in `pre()` is considered predetermined. (Here specifying `pre(w, lag(1, .))` to `xtabond` means that `L.w` is a predetermined variable and `pre(k, lag(2, .))` means that `L2.k` is a predetermined variable.) In a weaker definition, the current observation is considered predetermined, but subsequent lags are included in the model. Here `w` and `k` would be predetermined instead of `L.w` and `L2.w`. The output below implements this weaker definition for the previous example.

```
. xtdpd L(0/2).n L(0/1).(w ys) L(0/2).k yr1980-yr1984 year,
> div(L(0/1).(ys) yr1980-yr1984 year) dgmmiv(n) dgmmiv(w k, lag(1 .))
> twostep noconstant vce(robust)

Dynamic panel-data estimation                   Number of obs     =        611
Group variable: id                              Number of groups  =        140
Time variable: year
                                                Obs per group:
                                                              min =          4
                                                              avg =   4.364286
                                                              max =          6

Number of instruments =    101                  Wald chi2(15)     =     879.53
                                                Prob > chi2       =     0.0000

Two-step results
                                     (Std. err. adjusted for clustering on id)
```

n	Coefficient	WC-robust std. err.	z	P>|z|	[95% conf.	interval]
n						
L1.	.6343155	.1221058	5.19	0.000	.3949925	.8736384
L2.	-.0871247	.0704816	-1.24	0.216	-.2252661	.0510168
w						
--.	-.720063	.1133359	-6.35	0.000	-.9421973	-.4979287
L1.	.238069	.1223186	1.95	0.052	-.0016712	.4778091
ys						
--.	.5999718	.1653036	3.63	0.000	.2759827	.923961
L1.	-.5674808	.1656411	-3.43	0.001	-.8921314	-.2428303
k						
--.	.3931997	.0986673	3.99	0.000	.1998153	.5865842
L1.	-.0019641	.0772814	-0.03	0.980	-.1534329	.1495047
L2.	-.0231165	.0487317	-0.47	0.635	-.1186288	.0723958
yr1980	-.006209	.0162138	-0.38	0.702	-.0379875	.0255694
yr1981	-.0398491	.0313794	-1.27	0.204	-.1013516	.0216535
yr1982	-.0525715	.0397346	-1.32	0.186	-.1304498	.0253068
yr1983	-.0451175	.051418	-0.88	0.380	-.145895	.05566
yr1984	-.0437772	.0614391	-0.71	0.476	-.1641955	.0766412
year	.0173374	.0108665	1.60	0.111	-.0039605	.0386352

```
Instruments for differenced equation
        GMM-type: L(2/.).n L(1/.).w L(1/.).k
        Standard: D.ys LD.ys D.yr1980 D.yr1981 D.yr1982 D.yr1983 D.yr1984
                  D.year
```

As expected, the output shows that the additional 18 instruments available under the weaker definition can affect the magnitudes of the estimates. Applying the stricter definition when the true model was generated by the weaker definition yielded consistent but inefficient results; there were some additional moment conditions that could have been included but were not. In contrast, applying the weaker definition when the true model was generated by the stricter definition yields inconsistent estimates.

◁

▷ Example 4: A system estimator of a dynamic panel-data model

Here we use xtdpd to reproduce example 2 from [XT] **xtdpdsys** in which we used the system estimator to fit a model with predetermined variables.

```
. xtdpd L(0/1).n L(0/2).(w k) yr1980-yr1984 year, div(yr1980-yr1984 year)
> dgmmiv(n) dgmmiv(L2.(w k), lag(1 .)) lgmmiv(n L1.(w k)) vce(robust) hascons
Dynamic panel-data estimation                   Number of obs     =        751
Group variable: id                              Number of groups  =        140
Time variable: year
                                                Obs per group:
                                                             min =          5
                                                             avg =   5.364286
                                                             max =          7

Number of instruments =     95                  Wald chi2(13)     =    7562.80
                                                Prob > chi2       =     0.0000
One-step results
                                     (Std. err. adjusted for clustering on id)
------------------------------------------------------------------------------
             |               Robust
           n | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
           n |
         L1. |    .913278   .0460602    19.83   0.000     .8230017    1.003554
             |
           w |
         --. |   -.728159   .1019044    -7.15   0.000     -.927888   -.5284301
         L1. |   .5602737   .1939617     2.89   0.004     .1801156    .9404317
         L2. |  -.0523028   .1487653    -0.35   0.725    -.3438774    .2392718
             |
           k |
         --. |   .4820097   .0760787     6.34   0.000     .3328983    .6311212
         L1. |  -.2846944   .0831902    -3.42   0.001    -.4477442   -.1216446
         L2. |  -.1394181   .0405709    -3.44   0.001    -.2189356   -.0599006
             |
      yr1980 |  -.0325146   .0216371    -1.50   0.133    -.0749226    .0098935
      yr1981 |  -.0726116   .0346482    -2.10   0.036    -.1405207   -.0047024
      yr1982 |  -.0477038   .0451914    -1.06   0.291    -.1362772    .0408696
      yr1983 |  -.0396264   .0558734    -0.71   0.478    -.1491362    .0698835
      yr1984 |  -.0810383   .0736648    -1.10   0.271    -.2254186     .063342
        year |   .0192741   .0145326     1.33   0.185    -.0092092    .0477574
       _cons |  -37.34972   28.77747    -1.30   0.194    -93.75252    19.05308
------------------------------------------------------------------------------
Instruments for differenced equation
        GMM-type: L(2/.).n L(1/.).L2.w L(1/.).L2.k
        Standard: D.yr1980 D.yr1981 D.yr1982 D.yr1983 D.yr1984 D.year
Instruments for level equation
        GMM-type: LD.n L2D.w L2D.k
        Standard: _cons
```

The first lags of the variables included in `lgmmiv()` are used to create GMM-type instruments for the level equation. Only the first lags of the variables in `lgmmiv()` are used because the moment conditions using higher lags are redundant; see Blundell and Bond (1998) and Blundell, Bond, and Windmeijer (2000).

◁

▷ Example 5: Allowing for MA(1) errors

All the previous examples have used moment conditions that are valid only if the idiosyncratic errors are i.i.d. This example shows how to use `xtdpd` to estimate the parameters of a model with first-order moving-average [MA(1)] errors using the Arellano–Bond estimator, the Arellano–Bover/Blundell–Bond system estimator, or any other consistent GMM estimator you want to specify. For simplicity, we assume that the independent variables are strictly exogenous. Also, to highlight the fact that we can specify the instrument list flexibly, we only include the levels and first lags of the exogenous variables in the instrument list. An Arellano–Bond estimator, for instance, would have included levels and first and second lags of the exogenous variables.

We begin by noting that the Sargan test rejects the null hypothesis that the overidentifying restrictions are valid in the model with i.i.d. errors.

```
. xtdpd L(0/1).n L(0/2).(w k) yr1980-yr1984 year,
> div(L(0/1).(w k) yr1980-yr1984 year) dgmmiv(n) hascons
  (output omitted)
. estat sargan
Sargan test of overidentifying restrictions
H0: Overidentifying restrictions are valid

        chi2(24)     =  49.70094
        Prob > chi2  =    0.0015
```

Assuming that the idiosyncratic errors are MA(1) implies that only lags three or higher are valid instruments for the difference equation. (See the technical note below.)

```
. xtdpd L(0/1).n L(0/2).(w k) yr1980-yr1984 year,
> div(L(0/1).(w k) yr1980-yr1984 year) dgmmiv(n, lag(3 .)) hascons
Dynamic panel-data estimation                  Number of obs     =        751
Group variable: id                             Number of groups  =        140
Time variable: year
                                               Obs per group:
                                                             min =          5
                                                             avg =   5.364286
                                                             max =          7
Number of instruments =     32                 Wald chi2(13)     =    1195.04
                                               Prob > chi2       =     0.0000
One-step results

------------------------------------------------------------------------------
           n | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
           n |
         L1. |   .8696303   .2014473     4.32   0.000     .4748008     1.26446
             |
           w |
         --. |  -.5802971   .0762659    -7.61   0.000    -.7297756   -.4308187
         L1. |   .2918658   .1543883     1.89   0.059    -.0107296    .5944613
         L2. |  -.5903459   .2995123    -1.97   0.049    -1.177379   -.0033126
             |
           k |
         --. |   .3428139   .0447916     7.65   0.000     .2550239    .4306039
         L1. |  -.1383918   .0825823    -1.68   0.094    -.3002502    .0234665
         L2. |  -.0260956   .1535855    -0.17   0.865    -.3271177    .2749265
             |
      yr1980 |  -.0036873   .0301587    -0.12   0.903    -.0627973    .0554226
      yr1981 |     .00218   .0592014     0.04   0.971    -.1138526    .1182125
      yr1982 |   .0782939   .0897622     0.87   0.383    -.0976367    .2542246
      yr1983 |   .1734231   .1308914     1.32   0.185    -.0831193    .4299655
      yr1984 |   .2400685   .1734456     1.38   0.166    -.0998787    .5800157
        year |  -.0354681   .0309963    -1.14   0.253    -.0962198    .0252836
       _cons |   73.13706   62.61443     1.17   0.243    -49.58496    195.8591
------------------------------------------------------------------------------
Instruments for differenced equation
        GMM-type: L(3/.).n
        Standard: D.w LD.w D.k LD.k D.yr1980 D.yr1981 D.yr1982 D.yr1983
                  D.yr1984 D.year
Instruments for level equation
        Standard: _cons
```

The results from `estat sargan` no longer reject the null hypothesis that the overidentifying restrictions are valid.

```
. estat sargan
Sargan test of overidentifying restrictions
H0: Overidentifying restrictions are valid
        chi2(18)     =  20.80081
        Prob > chi2  =    0.2896
```

Moving on to the system estimator, we note that the Sargan test rejects the null hypothesis after fitting the model with i.i.d. errors.

```
. xtdpd L(0/1).n L(0/2).(w k) yr1980-yr1984 year,
> div(L(0/1).(w k) yr1980-yr1984 year) dgmmiv(n) lgmmiv(n) hascons
  (output omitted)
. estat sargan
Sargan test of overidentifying restrictions
H0: Overidentifying restrictions are valid

        chi2(31)     =  59.22907
        Prob > chi2  =    0.0017
```

Now we fit the model using the additional moment conditions constructed from the second lag of n as an instrument for the level equation.

```
. xtdpd L(0/1).n L(0/2).(w k) yr1980-yr1984 year,
> div(L(0/1).(w k) yr1980-yr1984 year) dgmmiv(n, lag(3 .)) lgmmiv(n, lag(2))
> hascons

Dynamic panel-data estimation                  Number of obs     =        751
Group variable: id                             Number of groups  =        140
Time variable: year
                                               Obs per group:
                                                             min =          5
                                                             avg =   5.364286
                                                             max =          7

Number of instruments =     38                 Wald chi2(13)     =    3680.01
                                               Prob > chi2       =     0.0000
One-step results
```

n	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
n						
L1.	.9603675	.095608	10.04	0.000	.7729794	1.147756
w						
--.	-.5433987	.068835	-7.89	0.000	-.6783128	-.4084845
L1.	.4356183	.0881727	4.94	0.000	.262803	.6084336
L2.	-.2785721	.1115061	-2.50	0.012	-.4971201	-.0600241
k						
--.	.3139331	.0419054	7.49	0.000	.2317999	.3960662
L1.	-.160103	.0546915	-2.93	0.003	-.2672963	-.0529096
L2.	-.1295766	.0507752	-2.55	0.011	-.2290943	-.030059
yr1980	-.0200704	.0248954	-0.81	0.420	-.0688644	.0287236
yr1981	-.0425838	.0422155	-1.01	0.313	-.1253246	.040157
yr1982	.0048723	.0600938	0.08	0.935	-.1129093	.122654
yr1983	.0458978	.0785687	0.58	0.559	-.1080941	.1998897
yr1984	.0633219	.1026188	0.62	0.537	-.1378074	.2644511
year	-.0075599	.019059	-0.40	0.692	-.0449148	.029795
_cons	16.20856	38.00619	0.43	0.670	-58.28221	90.69932

```
Instruments for differenced equation
        GMM-type: L(3/.).n
        Standard: D.w LD.w D.k LD.k D.yr1980 D.yr1981 D.yr1982 D.yr1983
                  D.yr1984 D.year
Instruments for level equation
        GMM-type: L2D.n
        Standard: _cons
```

The estimate of the coefficient on `L.n` is now 0.96. Blundell, Bond, and Windmeijer (2000, 63–65) show that the moment conditions in the system estimator remain informative as the true coefficient on `L.n` approaches unity. Holtz-Eakin, Newey, and Rosen (1988) show that because the large-sample distribution of the estimator is derived for fixed number of periods and a growing number of individuals there is no "unit-root" problem.

The results from `estat sargan` no longer reject the null hypothesis that the overidentifying restrictions are valid.

```
. estat sargan
Sargan test of overidentifying restrictions
H0: Overidentifying restrictions are valid
        chi2(24)     =  27.22585
        Prob > chi2  =    0.2940
```

◁

❑ Technical note

To find the valid moment conditions for the model with MA(1) errors, we begin by writing the model

$$n_{it} = \alpha n_{it-1} + \boldsymbol{\beta} x_{it} + \nu_i + \epsilon_{it} + \gamma \epsilon_{it-1}$$

where the ϵ_{it} are assumed to be i.i.d.

Because the composite error, $\epsilon_{it} + \gamma\epsilon_{it-1}$, is MA(1), only lags two or higher are valid instruments for the level equation, assuming the initial condition that $E[\nu_i \Delta n_{i2}] = 0$. The key to this point is that lagging the above equation two periods shows that ϵ_{it-2} and ϵ_{it-3} appear in the equation for n_{it-2}. Because the ϵ_{it} are i.i.d., n_{it-2} is a valid instrument for the level equation with errors $\nu_i + \epsilon_{it} + \gamma\epsilon_{it-1}$. ($n_{it-2}$ will be correlated with n_{it-1} but uncorrelated with the errors $\nu_i + \epsilon_{it} + \gamma\epsilon_{it-1}$.) An analogous argument works for higher lags.

First-differencing the above equation yields

$$\Delta n_{it} = \alpha \Delta n_{it-1} + \boldsymbol{\beta} \Delta x_{it} + \Delta\epsilon_{it} + \gamma \Delta\epsilon_{it-1}$$

Because ϵ_{it-2} is the farthest lag of ϵ_{it} that appears in the difference equation, lags three or higher are valid instruments for the differenced composite errors. (Lagging the level equation three periods shows that only ϵ_{it-3} and ϵ_{it-4} appear in the equation for n_{it-3}, which implies that n_{it-3} is a valid instrument for the current difference equation. An analogous argument works for higher lags.)

❑

Stored results

xtdpd stores the following in e():

Scalars

e(N)	number of observations
e(N_g)	number of groups
e(df_m)	model degrees of freedom
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(t_min)	minimum time in sample
e(t_max)	maximum time in sample
e(chi2)	χ^2
e(arm#)	test for autocorrelation of order #
e(artests)	number of AR tests computed
e(sig2)	estimate of σ_ϵ^2
e(rss)	sum of squared differenced residuals
e(sargan)	Sargan test statistic
e(rank)	rank of e(V)
e(zrank)	rank of instrument matrix

Macros

e(cmd)	xtdpd
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(twostep)	twostep, if specified
e(ivar)	variable denoting groups
e(tvar)	variable denoting time within groups
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(system)	system, if system estimator
e(transform)	specified transform
e(datasignature)	checksum from datasignature
e(datasignaturevars)	variables used in calculation of checksum
e(properties)	b V
e(estat_cmd)	program used to implement estat
e(predict)	program used to implement predict
e(marginsok)	predictions allowed by margins

Matrices

e(b)	coefficient vector
e(V)	variance–covariance matrix of the estimators

Functions

e(sample)	marks estimation sample

In addition to the above, the following is stored in r():

Matrices

r(table)	matrix containing the coefficients with their standard errors, test statistics, *p*-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

Consider dynamic panel-data models of the form

$$y_{it} = \sum_{j=1}^{p} \alpha_j y_{i,t-j} + \mathbf{x}_{it}\boldsymbol{\beta}_1 + \mathbf{w}_{it}\boldsymbol{\beta}_2 + \nu_i + \epsilon_{it}$$

where the variables are as defined as in (1).

x and **w** may contain lagged independent variables and time dummies.

Let $\mathbf{X}_{it}^L = (y_{i,t-1}, y_{i,t-2}, \ldots, y_{i,t-p}, \mathbf{x}_{it}, \mathbf{w}_{it})$ be the $1 \times K$ vector of covariates for i at time t, where $K = p + k_1 + k_2$, p is the number of included lags, k_1 is the number of strictly exogenous variables in x_{it}, and k_2 is the number of predetermined variables in w_{it}. (The superscript L stands for levels.)

Now rewrite this relationship as a set of T_i equations for each individual,

$$\mathbf{y}_i^L = \mathbf{X}_i^L \boldsymbol{\delta} + \nu_i \boldsymbol{\iota}_i + \boldsymbol{\epsilon}_i$$

where T_i is the number of observations available for individual i; $\mathbf{y}_i$, $\boldsymbol{\iota}_i$, and $\boldsymbol{\epsilon}_i$ are $T_i \times 1$, whereas $\mathbf{X}_i$ is $T_i \times K$.

The estimators use both the levels and a transform of the variables in the above equation. Denote the transformed variables by an *, so that $\mathbf{y}_i^*$ is the transformed $\mathbf{y}_i^L$ and $\mathbf{X}_i^*$ is the transformed $\mathbf{X}_i^L$. The transform may be either the first difference or the forward-orthogonal deviations (FOD) transform. The (i,t)th observation of the FOD transform of a variable x is given by

$$x_{it}^* = c_t \left\{ x_{it} - \frac{1}{T-t}(x_{it+1} + x_{it+2} + \cdots + x_{iT}) \right\}$$

where $c_t^2 = (T-t)/(T-t+1)$ and T is the number of observations on x; see Arellano and Bover (1995) and Arellano (2003).

Here we present the formulas for the Arellano–Bover/Blundell–Bond system estimator. The formulas for the Arellano–Bond estimator are obtained by setting the additional level matrices in the system estimator to null matrices.

Stacking the transformed and untransformed vectors of the dependent variable for a given i yields

$$\mathbf{y}_i = \begin{pmatrix} \mathbf{y}_i^* \\ \mathbf{y}_i^L \end{pmatrix}$$

Similarly, stacking the transformed and untransformed matrices of the covariates for a given i yields

$$\mathbf{X}_i = \begin{pmatrix} \mathbf{X}_i^* \\ \mathbf{X}_i^L \end{pmatrix}$$

$\mathbf{Z}_i$ is a matrix of instruments,

$$\mathbf{Z}_i = \begin{pmatrix} \mathbf{Z}_{di} & \mathbf{0} & \mathbf{D}_i & \mathbf{0} & \mathbf{I}_i^d \\ \mathbf{0} & \mathbf{Z}_{Li} & \mathbf{0} & \mathbf{L}_i & \mathbf{I}_i^L \end{pmatrix}$$

where $\mathbf{Z}_{di}$ is the matrix of GMM-type instruments created from the `dgmmiv()` options, $\mathbf{Z}_{Li}$ is the matrix of GMM-type instruments created from the `lgmmiv()` options, $\mathbf{D}_i$ is the matrix of standard instruments created from the `div()` options, $\mathbf{L}_i$ is the matrix of standard instruments created from the `liv()` options, $\mathbf{I}_i^d$ is the matrix of standard instruments created from the `iv()` options for the differenced errors, and $\mathbf{I}_i^L$ is the matrix of standard instruments created from the `iv()` options for the level errors.

`div()`, `liv()`, and `iv()` simply add columns to instrument matrix. The GMM-type instruments are more involved. Begin by considering a simple balanced-panel example in which our model is

$$y_{it} = \alpha_1 y_{i,t-1} + \alpha_2 y_{i,t-2} + \nu_i + \epsilon_{it}$$

We do not need to consider covariates because strictly exogenous variables are handled using `div()`, `iv()`, or `liv()`, and predetermined or endogenous variables are handled analogous to the dependent variable.

Assume that the data come from a balanced panel in which there are no missing values. After first-differencing the equation, we have

$$\Delta y_{it} = \alpha_1 \Delta y_{i,t-1} + \alpha_2 \Delta y_{i,t-2} + \Delta\epsilon_{it}$$

The first 3 observations are lost to lags and differencing. If we assume that the ϵ_{it} are not autocorrelated, for each i at $t = 4$, y_{i1} and y_{i2} are valid instruments for the differenced equation. Similarly, at $t = 5$, y_{i1}, y_{i2}, and y_{i3} are valid instruments. We specify `dgmmiv(y)` to obtain an instrument matrix with one row for each period that we are instrumenting:

$$\mathbf{Z}_{di} = \begin{pmatrix} y_{i1} & y_{i2} & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & y_{i1} & y_{i2} & y_{i3} & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 & y_{i1} & \dots & y_{i,T-2} \end{pmatrix}$$

Because $p = 2$, $\mathbf{Z}_{di}$ has $T - p - 1$ rows and $\sum_{m=p}^{T-2} m$ columns.

Specifying `lgmmiv(y)` creates the instrument matrix

$$\mathbf{Z}_{Li} = \begin{pmatrix} \Delta.y_{i2} & 0 & 0 & \dots & 0 \\ 0 & \Delta.y_{i3} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \Delta.y_{i(T_i-1)} \end{pmatrix}$$

This extends to other lag structures with complete data. Unbalanced data and missing observations are handled by dropping the rows for which there are no data and filling in zeros in columns where missing data are required.

After defining

$$\mathbf{Q}_{xz} = \sum_i \mathbf{X}_i'\mathbf{Z}_i$$

$$\mathbf{Q}_{zy} = \sum_i \mathbf{Z}_i'\mathbf{y}_i$$

$$\mathbf{W}_1 = \mathbf{Q}_{xz}\mathbf{A}_1\mathbf{Q}_{xz}'$$

$$\mathbf{A}_1 = \left(\sum_i \mathbf{Z}_i'\mathbf{H}_{1i}\mathbf{Z}_i\right)^{-1}$$

and

$$\mathbf{H}_{1i} = \begin{pmatrix} \mathbf{H}_{di} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{Li} \end{pmatrix}$$

the one-step estimates are given by

$$\widehat{\boldsymbol{\beta}}_1 = \mathbf{W}_1^{-1}\mathbf{Q}_{xz}\mathbf{A}_1\mathbf{Q}_{zy}$$

When using the first-difference transform $\mathbf{H}_{di}$, is given by

$$\mathbf{H}_{di} = \begin{pmatrix} 1 & -.5 & 0 & \dots & 0 & 0 \\ -.5 & 1 & -.5 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & -.5 \\ 0 & 0 & 0 & \dots & -.5 & 1 \end{pmatrix}$$

and $\mathbf{H}_{Li}$ is given by 0.5 times the identity matrix. When using the FOD transform, both $\mathbf{H}_{di}$ and $\mathbf{H}_{Li}$ are equal to the identity matrix.

The transformed one-step residuals are given by

$$\widehat{\boldsymbol{\epsilon}}^*_{1i} = \mathbf{y}^*_i - \widehat{\boldsymbol{\beta}}_1 \mathbf{X}^*_i$$

which are used to compute

$$\widehat{\sigma}^2_1 = \{1/(N-K)\} \sum_i^N \widehat{\boldsymbol{\epsilon}}^{*\prime}_{1i} \widehat{\boldsymbol{\epsilon}}^*_{1i}$$

The GMM one-step VCE is then given by

$$\widehat{V}_{\text{GMM}}[\widehat{\boldsymbol{\beta}}_1] = \widehat{\sigma}^2_1 \mathbf{W}_1^{-1}$$

The one-step level residuals are given by

$$\widehat{\boldsymbol{\epsilon}}^L_{1i} = \mathbf{y}^L_i - \widehat{\boldsymbol{\beta}}_1 \mathbf{X}^L_i$$

Stacking the residual vectors yields

$$\widehat{\boldsymbol{\epsilon}}_{1i} = \begin{pmatrix} \widehat{\boldsymbol{\epsilon}}^*_{1i} \\ \widehat{\boldsymbol{\epsilon}}^L_{1i} \end{pmatrix}$$

which is used to compute $\mathbf{H}_{2i} = \widehat{\boldsymbol{\epsilon}}'_{1i} \widehat{\boldsymbol{\epsilon}}_{1i}$, which is used in

$$\mathbf{A}_2 = \left(\sum_i \mathbf{Z}'_i \mathbf{H}_{2i} \mathbf{Z}_i \right)^{-1}$$

and the robust one-step VCE is given by

$$\widehat{V}_{\text{robust}}[\widehat{\boldsymbol{\beta}}_1] = \mathbf{W}_1^{-1} \mathbf{Q}_{xz} \mathbf{A}_1 \mathbf{A}_2^{-1} \mathbf{A}_1 \mathbf{Q}'_{xz} \mathbf{W}_1^{-1}$$

$\widehat{V}_{\text{robust}}[\widehat{\boldsymbol{\beta}}_1]$ is robust to heteroskedasticity in the errors.

After defining

$$\mathbf{W}_2 = \mathbf{Q}_{xz} \mathbf{A}_2 \mathbf{Q}'_{xz}$$

the two-step estimates are given by

$$\widehat{\boldsymbol{\beta}}_2 = \mathbf{W}_2^{-1} \mathbf{Q}_{xz} \mathbf{A}_2 \mathbf{Q}_{zy}$$

The GMM two-step VCE is then given by

$$\widehat{V}_{\text{GMM}}[\widehat{\boldsymbol{\beta}}_2] = \mathbf{W}_2^{-1}$$

The GMM two-step VCE is known to be severely biased. Windmeijer (2005) derived the Windmeijer bias-corrected (WC) estimator for the robust VCE of two-step GMM estimators. `xtdpd` implements this WC-robust estimator of the VCE. The formulas for this method are involved; see Windmeijer (2005). The WC-robust estimator of the VCE is robust to heteroskedasticity in the errors.

Acknowledgment

We thank David Roodman of the Open Philanthropy Project, who wrote `xtabond2`.

References

Anderson, T. W., and C. Hsiao. 1981. Estimation of dynamic models with error components. *Journal of the American Statistical Association* 76: 598–606. https://doi.org/10.2307/2287517.

———. 1982. Formulation and estimation of dynamic models using panel data. *Journal of Econometrics* 18: 47–82. https://doi.org/10.1016/0304-4076(82)90095-1.

Arellano, M. 2003. *Panel Data Econometrics*. Oxford: Oxford University Press. https://doi.org/10.1093/0199245282.001.0001.

Arellano, M., and S. Bond. 1991. Some tests of specification for panel data: Monte Carlo evidence and an application to employment equations. *Review of Economic Studies* 58: 277–297. https://doi.org/10.2307/2297968.

Arellano, M., and O. Bover. 1995. Another look at the instrumental variable estimation of error-components models. *Journal of Econometrics* 68: 29–51. https://doi.org/10.1016/0304-4076(94)01642-D.

Baltagi, B. H. 2013. *Econometric Analysis of Panel Data*. 5th ed. Chichester, UK: Wiley.

Blackburne, E. F., III, and M. W. Frank. 2007. Estimation of nonstationary heterogeneous panels. *Stata Journal* 7: 197–208.

Blundell, R. W., and S. Bond. 1998. Initial conditions and moment restrictions in dynamic panel data models. *Journal of Econometrics* 87: 115–143. https://doi.org/10.1016/S0304-4076(98)00009-8.

Blundell, R. W., S. Bond, and F. Windmeijer. 2000. “Estimation in dynamic panel data models: Improving on the performance of the standard GMM estimator”. In *Nonstationary Panels, Cointegrating Panels and Dynamic Panels*, edited by B. H. Baltagi, 53–92. New York: Elsevier. https://doi.org/10.1016/S0731-9053(00)15003-0.

Bruno, G. S. F. 2005. Estimation and inference in dynamic unbalanced panel-data models with a small number of individuals. *Stata Journal* 5: 473–500.

De Vos, I., G. Everaert, and I. Ruyssen. 2015. Bootstrap-based bias correction and inference for dynamic panels with fixed effects. *Stata Journal* 15: 986–1018.

Grotti, R., and G. Cutuli. 2018. xtpdyn: A community-contributed command for fitting dynamic random-effects probit models with unobserved heterogeneity. *Stata Journal* 18: 844–862.

Hansen, L. P. 1982. Large sample properties of generalized method of moments estimators. *Econometrica* 50: 1029–1054. https://doi.org/10.2307/1912775.

Holtz-Eakin, D., W. K. Newey, and H. S. Rosen. 1988. Estimating vector autoregressions with panel data. *Econometrica* 56: 1371–1395. https://doi.org/10.2307/1913103.

Keane, M. P., and T. Neal. 2016. The Keane and Runkle estimator for panel-data models with serial correlation and instruments that are not strictly exogenous. *Stata Journal* 16: 523–549.

Kripfganz, S. 2016. Quasi–maximum likelihood estimation of linear dynamic short-T panel-data models. *Stata Journal* 16: 1013–1038.

Layard, R., and S. J. Nickell. 1986. Unemployment in Britain. *Economica* 53: S121–S169. https://doi.org/10.2307/2554377.

Li, J., M. J. Zyphur, G. Sugihara, and P. J. Laub. 2021. Beyond linearity, stability, and equilibrium: The edm package for empirical dynamic modeling and convergent cross-mapping in Stata. *Stata Journal* 21: 220–258.

Neal, T. 2014. Panel cointegration analysis with xtpedroni. *Stata Journal* 14: 684–692.

Sasaki, Y., and Y. Xin. 2022. xtusreg: Software for dynamic panel regression under irregular time spacing. *Stata Journal* 22: 713–724.

Seo, M. H., S. Kim, and Y.-J. Kim. 2019. Estimation of dynamic panel threshold model using Stata. *Stata Journal* 19: 685–697.

Williams, R., P. D. Allison, and E. Moral-Benito. 2018. Linear dynamic panel-data estimation using maximum likelihood and structural equation modeling. *Stata Journal* 18: 293–326.

Windmeijer, F. 2005. A finite sample correction for the variance of linear efficient two-step GMM estimators. *Journal of Econometrics* 126: 25–51. https://doi.org/10.1016/j.jeconom.2004.02.005.

Also see

[XT] **xtdpd postestimation** — Postestimation tools for xtdpd

[XT] **xtabond** — Arellano–Bond linear dynamic panel-data estimation

[XT] **xtdpdsys** — Arellano–Bover/Blundell–Bond linear dynamic panel-data estimation

[XT] **xtivreg** — Instrumental variables and two-stage least squares for panel-data models

[XT] **xtreg** — Linear models for panel data

[XT] **xtregar** — Fixed- and random-effects linear models with an AR(1) disturbance

[XT] **xtset** — Declare data to be panel data

[R] **gmm** — Generalized method of moments estimation

[U] 20 Estimation and postestimation commands

xtdpd postestimation — Postestimation tools for xtdpd

Postestimation commands

The following postestimation commands are of special interest after `xtdpd`:

Command	Description
`estat abond`	test for autocorrelation
`estat sargan`	Sargan test of overidentifying restrictions

The following standard postestimation commands are also available:

Command	Description
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
`forecast`	dynamic forecasts and simulations
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	linear predictions and their SEs, residual errors
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

predict

Description for predict

predict creates a new variable containing predictions such as linear predictions.

Menu for predict

Statistics > Postestimation

Syntax for predict

predict [*type*] *newvar* [*if*] [*in*] [, xb e stdp difference]

Options for predict

Main

xb, the default, calculates the linear prediction.

e calculates the residual error.

stdp calculates the standard error of the prediction, which can be thought of as the standard error of the predicted expected value or mean for the observation's covariate pattern. The standard error of the prediction is also referred to as the standard error of the fitted value. stdp may not be combined with difference.

difference specifies that the statistic be calculated for the first differences instead of the levels, the default.

margins

Description for margins

`margins` estimates margins of responses for linear predictions.

Menu for margins

Statistics > Postestimation

Syntax for margins

`margins` [*marginlist*] [, *options*]

`margins` [*marginlist*] , `predict(`*statistic* ...`)` [*options*]

statistic	Description
`xb`	linear prediction; the default
`e`	not allowed with `margins`
`stdp`	not allowed with `margins`

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.

For the full syntax, see [R] **margins**.

estat

Description for estat

`estat abond` reports the Arellano–Bond test for serial correlation in the first-differenced residuals.

`estat sargan` reports the Sargan test of the overidentifying restrictions.

Menu for estat

Statistics > Postestimation

Syntax for estat

Test for autocorrelation

`estat abond [ , artests(#) ]`

Sargan test of overidentifying restrictions

`estat sargan`

Option for estat abond

`artests(#)` specifies the highest order of serial correlation to be tested. By default, the tests computed during estimation are reported. The model will be refit when `artests(#)` specifies a higher order than that computed during the original estimation. The model can only be refit if the data have not changed.

Remarks and examples

Remarks are presented under the following headings:

estat abond
estat sargan

estat abond

The moment conditions used by `xtdpd` are valid only if there is no serial correlation in the idiosyncratic errors. Testing for serial correlation in dynamic panel-data models is tricky because one needs to apply a transform to remove the panel-level effects, but the transformed errors have a more complicated error structure than the idiosyncratic errors. The Arellano–Bond test for serial correlation reported by `estat abond` tests for serial correlation in the first-differenced errors.

Because the first difference of independent and identically distributed idiosyncratic errors will be autocorrelated, rejecting the null hypothesis of no serial correlation at order one in the first-differenced errors does not imply that the model is misspecified. Rejecting the null hypothesis at higher orders implies that the moment conditions are not valid. See example 5 in [XT] **xtdpd** for an alternative estimator that allows for idiosyncratic errors that follow a first-order moving average process.

After the one-step system estimator, the test can be computed only when `vce(robust)` has been specified.

estat sargan

Like all GMM estimators, the estimator in `xtdpd` can produce consistent estimates only if the moment conditions used are valid. Although there is no method to test if the moment conditions from an exactly identified model are valid, one can test whether the overidentifying moment conditions are valid. `estat sargan` implements the Sargan test of overidentifying conditions discussed in Arellano and Bond (1991).

Only for a homoskedastic error term does the Sargan test have an asymptotic χ^2 distribution. In fact, Arellano and Bond (1991) show that the one-step Sargan test overrejects in the presence of heteroskedasticity. Because its asymptotic distribution is not known under the assumptions of the `vce(robust)` model, `xtdpd` does not compute it when `vce(robust)` is specified.

Methods and formulas

The notation for $\widehat{\boldsymbol{\epsilon}}^*_{1i}$, $\widehat{\boldsymbol{\epsilon}}_{1i}$, $\mathbf{H}_{1i}$, $\mathbf{H}_{2i}$, $\mathbf{X}_i$, $\mathbf{Z}_i$, $\mathbf{W}_1$, $\mathbf{W}_2$, $\widehat{\mathbf{V}}_*[\widehat{\boldsymbol{\beta}}_*]$, $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{Q}_{xz}$, and $\widehat{\sigma}^2_1$ has been defined in *Methods and formulas* of [XT] **xtdpd**.

The Arellano–Bond test for zero mth-order autocorrelation in the first-differenced errors is given by

$$A(m) = \frac{s_0}{\sqrt{s_1 + s_2 + s_3}}$$

where the definitions of s_0, s_1, s_2, and s_3 vary over the estimators and transforms.

We begin by defining $\widehat{\mathbf{u}}^*_{1i} = Lm.\widehat{\boldsymbol{\epsilon}}^*_{1i}$, with the missing values filled in with zeros. Letting $j = 1$ for the one-step estimator, $j = 2$ for the two-step estimator, $c =$ GMM for the GMM VCE estimator, and $c =$ robust for the robust VCE estimator, we can now define s_0, s_1, s_2, and s_3:

$$s_0 = \sum_i \widehat{\mathbf{u}}^{*\prime}_{ji} \widehat{\boldsymbol{\epsilon}}^*_{ji}$$

$$s_1 = \sum_i \widehat{\mathbf{u}}^{*\prime}_{ji} \mathbf{H}_{ji} \widehat{\mathbf{u}}^*_{ji}$$

$$s_2 = -2\mathbf{q}_{ji} \mathbf{W}_j^{-1} \mathbf{Q}_{xz} \mathbf{A}_j \mathbf{Q}_{zu}$$

$$s_3 = \mathbf{q}_{jx} \widehat{\mathbf{V}}_c \left[\widehat{\boldsymbol{\beta}}_j\right] \mathbf{q}'_{jx}$$

where

$$\mathbf{q}_{jx} = \left(\sum_i \widehat{\mathbf{u}}^{*\prime}_{ji} \mathbf{X}_i \right)$$

and $\mathbf{Q}_{zu}$ varies over estimator and transform.

For the Arellano–Bond estimator with the first-differenced transform,

$$\mathbf{Q}_{zu} = \left(\sum_i \mathbf{Z}'_i \mathbf{H}_{ji} \widehat{\mathbf{u}}^*_{ji} \right)$$

For the Arellano–Bond estimator with the FOD transform,

$$\mathbf{Q}_{zu} = \left(\sum_i \mathbf{Z}_i' \mathbf{Q}_{\text{fod}} \right)$$

where

$$\mathbf{Q}_{\text{fod}} = \begin{pmatrix} -\sqrt{\frac{T_i+1}{T_i}} & 0 & \cdots & 0 \\ \sqrt{\frac{T_i-1}{T_i}} & \sqrt{\frac{T_i}{T_i-1}} & \cdots & 0 \\ 0 & . & . & \vdots \\ 0 & \cdots & \sqrt{\frac{1}{2}} & -\sqrt{\frac{2}{1}} \end{pmatrix} \widehat{\mathbf{u}}_{ji}^*$$

and * implies the first-differenced transform instead of the FOD transform.

For the Arellano–Bover/Blundell–Bond system estimator with the first-differenced transform,

$$\mathbf{Q}_{zu} = \left(\sum_i \mathbf{Z}_i' \widehat{\boldsymbol{\epsilon}}_{ji} \widehat{\boldsymbol{\epsilon}}_{ji}^{*\prime} \widehat{\mathbf{u}}_{ji}^* \right)$$

After a one-step estimator, the Sargan test is

$$S_1 = \frac{1}{\widehat{\sigma}_1^2} \left(\sum_i \widehat{\boldsymbol{\epsilon}}_{1i}' \mathbf{Z}_i \right) \mathbf{A}_1 \left(\sum_i \mathbf{Z}_i' \widehat{\boldsymbol{\epsilon}}_{1i} \right)$$

The transformed two-step residuals are given by

$$\widehat{\boldsymbol{\epsilon}}_{2i}^* = \mathbf{y}_i^* - \widehat{\boldsymbol{\beta}}_2 \mathbf{X}_i^*$$

and the level two-step residuals are given by

$$\widehat{\boldsymbol{\epsilon}}_{2i}^L = \mathbf{y}_i^L - \widehat{\boldsymbol{\beta}}_2 \mathbf{X}_i^L$$

Stacking the residual vectors yields

$$\widehat{\boldsymbol{\epsilon}}_{2i} = \begin{pmatrix} \widehat{\boldsymbol{\epsilon}}_{2i}^* \\ \widehat{\boldsymbol{\epsilon}}_{2i}^L \end{pmatrix}$$

After a two-step estimator, the Sargan test is

$$S_2 = \left(\sum_i \widehat{\boldsymbol{\epsilon}}_{2i}' \mathbf{Z}_i \right) \mathbf{A}_2 \left(\sum_i \mathbf{Z}_i' \widehat{\boldsymbol{\epsilon}}_{2i} \right)$$

Reference

Arellano, M., and S. Bond. 1991. Some tests of specification for panel data: Monte Carlo evidence and an application to employment equations. *Review of Economic Studies* 58: 277–297. https://doi.org/10.2307/2297968.

Also see

[XT] **xtdpd** — Linear dynamic panel-data estimation

[U] 20 Estimation and postestimation commands

xtdpdsys — Arellano–Bover/Blundell–Bond linear dynamic panel-data estimation

Description

`xtdpdsys` fits a linear dynamic panel-data model where the unobserved panel-level effects are correlated with the lags of the dependent variable. This model is an extension of the Arellano–Bond estimator that accommodates large autoregressive parameters and a large ratio of the variance of the panel-level effect to the variance of idiosyncratic error. This is known as the Arellano–Bover/Blundell–Bond system estimator. This estimator is designed for datasets with many panels and few periods. This method assumes that there is no autocorrelation in the idiosyncratic errors and requires that the panel-level effects be uncorrelated with the first difference of the first observation of the dependent variable.

Quick start

Dynamic panel-data regression of `y` on `x` with default Arellano–Bond instruments and lagged difference of `y`

```
xtdpdsys y x
```

Add the lagged difference of `x` as an instrument

```
xtdpdsys y x, pre(x)
```

Set the maximum number of lags of the dependent variable used as instruments to 2

```
xtdpdsys y x, maxldep(2)
```

Menu

Statistics > Longitudinal/panel data > Dynamic panel data (DPD) > Arellano–Bover/Blundell–Bond estimation

Syntax

`xtdpdsys` *depvar* [*indepvars*] [*if*] [*in*] [, *options*]

options	Description
Model	
`noconstant`	suppress constant term
`lags(#)`	use # lags of dependent variable as covariates; default is `lags(1)`
`maxldep(#)`	maximum lags of dependent variable for use as instruments
`maxlags(#)`	maximum lags of predetermined and endogenous variables for use as instruments
`twostep`	compute the two-step estimator instead of the one-step estimator
Predetermined	
`pre(`*varlist*[...]`)`	predetermined variables; can be specified more than once
Endogenous	
`endogenous(`*varlist*[...]`)`	endogenous variables; can be specified more than once
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `gmm` or `robust`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`artests(#)`	use # as maximum order for AR tests; default is `artests(2)`
display_options	control spacing and line width
`coeflegend`	display legend instead of statistics

A panel variable and a time variable must be specified; use [XT] **xtset**.

indepvars and all *varlists*, except `pre(`*varlist*[...]`)` and `endogenous(`*varlist*[...]`)`, may contain time-series operators; see **[U] 11.4.4 Time-series varlists**. The specification of *depvar* may not contain time-series operators.

`by`, `collect`, `statsby`, and `xi` are allowed; see **[U] 11.1.10 Prefix commands**.

`coeflegend` does not appear in the dialog box.

See **[U] 20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

`noconstant`; see [R] **Estimation options**.

`lags(#)` sets p, the number of lags of the dependent variable to be included in the model. The default is $p = 1$.

`maxldep(#)` sets the maximum number of lags of the dependent variable that can be used as instruments. The default is to use all $T_i - p - 2$ lags.

`maxlags(#)` sets the maximum number of lags of the predetermined and endogenous variables that can be used as instruments. For predetermined variables, the default is to use all $T_i - p - 1$ lags. For endogenous variables, the default is to use all $T_i - p - 2$ lags.

`twostep` specifies that the two-step estimator be calculated.

Predetermined

pre(*varlist* [, lagstruct(*prelags*, *premaxlags*)]) specifies that a set of predetermined variables be included in the model. Optionally, you may specify that *prelags* lags of the specified variables also be included. The default for *prelags* is 0. Specifying *premaxlags* sets the maximum number of further lags of the predetermined variables that can be used as instruments. The default is to include $T_i - p - 1$ lagged levels as instruments for predetermined variables. You may specify as many sets of predetermined variables as you need within the standard Stata limits on matrix size. Each set of predetermined variables may have its own number of *prelags* and *premaxlags*.

Endogenous

endogenous(*varlist* [, lagstruct(*endlags*, *endmaxlags*)]) specifies that a set of endogenous variables be included in the model. Optionally, you may specify that *endlags* lags of the specified variables also be included. The default for *endlags* is 0. Specifying *endmaxlags* sets the maximum number of further lags of the endogenous variables that can be used as instruments. The default is to include $T_i - p - 2$ lagged levels as instruments for endogenous variables. You may specify as many sets of endogenous variables as you need within the standard Stata limits on matrix size. Each set of endogenous variables may have its own number of *endlags* and *endmaxlags*.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory and that are robust to some kinds of misspecification; see *Methods and formulas* in [XT] **xtdpd**.

vce(gmm), the default, uses the conventionally derived variance estimator for generalized method of moments estimation.

vce(robust) uses the robust estimator. For the one-step estimator, this is the Arellano–Bond robust VCE estimator. For the two-step estimator, this is the Windmeijer (2005) WC-robust estimator.

Reporting

level(#); see [R] **Estimation options**.

artests(#) specifies the maximum order of the autocorrelation test to be calculated. The tests are reported by estat abond; see [XT] **xtdpdsys postestimation**. Specifying the order of the highest test at estimation time is more efficient than specifying it to estat abond, because estat abond must refit the model to obtain the test statistics. The maximum order must be less than or equal the number of periods in the longest panel. The default is artests(2).

display_options: vsquish and nolstretch; see [R] **Estimation options**.

The following option is available with xtdpdsys but is not shown in the dialog box:

coeflegend; see [R] **Estimation options**.

Remarks and examples

If you have not read [XT] **xtabond**, you may want to do so before continuing.

Linear dynamic panel-data models include p lags of the dependent variable as covariates and contain unobserved panel-level effects, fixed or random

Consider the dynamic panel-data model

$$y_{it} = \sum_{j=1}^{p} \alpha_j y_{i,t-j} + \mathbf{x}_{it}\boldsymbol{\beta}_1 + \mathbf{w}_{it}\boldsymbol{\beta}_2 + \nu_i + \epsilon_{it} \quad i = 1, \ldots, N \quad t = 1, \ldots, T_i \tag{1}$$

where

the α_j are p parameters to be estimated,
$\mathbf{x}_{it}$ is a $1 \times k_1$ vector of strictly exogenous covariates,
$\boldsymbol{\beta}_1$ is a $k_1 \times 1$ vector of parameters to be estimated,
$\mathbf{w}_{it}$ is a $1 \times k_2$ vector of predetermined or endogenous covariates,
$\boldsymbol{\beta}_2$ is a $k_2 \times 1$ vector of parameters to be estimated,
ν_i are the panel-level effects (which may be correlated with the covariates), and
ϵ_{it} are i.i.d. over the whole sample with variance σ^2_ϵ.

The ν_i and the ϵ_{it} are assumed to be independent for each i over all t.

By construction, the lagged dependent variables are correlated with the unobserved panel-level effects, making standard estimators inconsistent. Arellano and Bond (1991) derived a consistent generalized method of moments (GMM) estimator for this model. With many panels and few periods, the Arellano–Bond estimator is constructed by first-differencing to remove the panel-level effects and using instruments to form moment conditions.

Blundell and Bond (1998) show that the lagged-level instruments in the Arellano–Bond estimator become weak as the autoregressive process becomes too persistent or the ratio of the variance of the panel-level effects ν_i to the variance of the idiosyncratic error ϵ_{it} becomes too large. Building on the work of Arellano and Bover (1995), Blundell and Bond (1998) proposed a system estimator that uses moment conditions in which lagged differences are used as instruments for the level equation in addition to the moment conditions of lagged levels as instruments for the difference equation. The additional moment conditions are valid only if the initial condition $E[\nu_i \Delta y_{i2}] = 0$ holds for all i; see Blundell and Bond (1998) and Blundell, Bond, and Windmeijer (2000).

`xtdpdsys` fits dynamic panel-data estimators with the Arellano–Bover/Blundell–Bond system estimator. This estimator is designed for datasets with many panels and few periods. This method assumes that there is no autocorrelation in the idiosyncratic errors and requires the initial condition that the panel-level effects be uncorrelated with the first difference of the first observation of the dependent variable. Because `xtdpdsys` extends `xtabond`, [XT] **xtabond** provides useful background.

▷ Example 1: A dynamic panel model

In their article, Arellano and Bond (1991) apply their estimators and test statistics to a model of dynamic labor demand that had previously been considered by Layard and Nickell (1986), using data from an unbalanced panel of firms from the United Kingdom. All variables are indexed over the firm i and time t. In this dataset, $\texttt{n}_{it}$ is the log of employment in firm i at time t, $\texttt{w}_{it}$ is the natural log of the real product wage, $\texttt{k}_{it}$ is the natural log of the gross capital stock, and $\texttt{ys}_{it}$ is the natural log of industry output. The model also includes time dummies `yr1980`, `yr1981`, `yr1982`, `yr1983`, and `yr1984`.

For comparison, we begin by using `xtabond` to fit a model to these data.

```
. use https://www.stata-press.com/data/r19/abdata

. xtabond n L(0/2).(w k) yr1980-yr1984 year, vce(robust)

Arellano-Bond dynamic panel-data estimation     Number of obs     =        611
Group variable: id                              Number of groups  =        140
Time variable: year
                                                Obs per group:
                                                              min =          4
                                                              avg =   4.364286
                                                              max =          6

Number of instruments =     40                  Wald chi2(13)     =    1318.68
                                                Prob > chi2       =     0.0000

One-step results
                                     (Std. err. adjusted for clustering on id)
------------------------------------------------------------------------------
             |               Robust
           n | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
           n |
         L1. |   .6286618   .1161942     5.41   0.000     .4009254    .8563983
             |
           w |
         --. |  -.5104249   .1904292    -2.68   0.007    -.8836592   -.1371906
         L1. |   .2891446    .140946     2.05   0.040     .0128954    .5653937
         L2. |  -.0443653   .0768135    -0.58   0.564     -.194917    .1061865
             |
           k |
         --. |   .3556923   .0603274     5.90   0.000     .2374528    .4739318
         L1. |  -.0457102   .0699732    -0.65   0.514    -.1828552    .0914348
         L2. |  -.0619721   .0328589    -1.89   0.059    -.1263743    .0024301
             |
      yr1980 |  -.0282422   .0166363    -1.70   0.090    -.0608488    .0043643
      yr1981 |  -.0694052    .028961    -2.40   0.017    -.1261677   -.0126426
      yr1982 |  -.0523678   .0423433    -1.24   0.216    -.1353591    .0306235
      yr1983 |  -.0256599   .0533747    -0.48   0.631    -.1302723    .0789525
      yr1984 |  -.0093229   .0696241    -0.13   0.893    -.1457837    .1271379
        year |   .0019575   .0119481     0.16   0.870    -.0214604    .0253754
       _cons |  -2.543221   23.97919    -0.11   0.916    -49.54158    44.45514
------------------------------------------------------------------------------
Instruments for differenced equation
        GMM-type: L(2/.).n
        Standard: D.w LD.w L2D.w D.k LD.k L2D.k D.yr1980 D.yr1981 D.yr1982
                  D.yr1983 D.yr1984 D.year
Instruments for level equation
        Standard: _cons
```

Now we fit the same model by using xtdpdsys:

```
. xtdpdsys n L(0/2).(w k) yr1980-yr1984 year, vce(robust)
System dynamic panel-data estimation            Number of obs      =        751
Group variable: id                              Number of groups   =        140
Time variable: year
                                                Obs per group:
                                                              min =          5
                                                              avg =   5.364286
                                                              max =          7

Number of instruments =     47                  Wald chi2(13)      =    2579.96
                                                Prob > chi2        =     0.0000
One-step results
```

n	Coefficient	Robust std. err.	z	P>|z|	[95% conf.	interval]
n						
L1.	.8221535	.093387	8.80	0.000	.6391184	1.005189
w						
--.	-.5427935	.1881721	-2.88	0.004	-.911604	-.1739831
L1.	.3703602	.1656364	2.24	0.025	.0457189	.6950015
L2.	-.0726314	.0907148	-0.80	0.423	-.2504292	.1051664
k						
--.	.3638069	.0657524	5.53	0.000	.2349346	.4926792
L1.	-.1222996	.0701521	-1.74	0.081	-.2597951	.015196
L2.	-.0901355	.0344142	-2.62	0.009	-.1575862	-.0226849
yr1980	-.0308622	.016946	-1.82	0.069	-.0640757	.0023512
yr1981	-.0718417	.0293223	-2.45	0.014	-.1293123	-.014371
yr1982	-.0384806	.0373631	-1.03	0.303	-.1117111	.0347498
yr1983	-.0121768	.0498519	-0.24	0.807	-.1098847	.0855311
yr1984	-.0050903	.0655011	-0.08	0.938	-.1334701	.1232895
year	.0058631	.0119867	0.49	0.625	-.0176304	.0293566
_cons	-10.59198	23.92087	-0.44	0.658	-57.47602	36.29207

```
Instruments for differenced equation
        GMM-type: L(2/.).n
        Standard: D.w LD.w L2D.w D.k LD.k L2D.k D.yr1980 D.yr1981 D.yr1982
                  D.yr1983 D.yr1984 D.year
Instruments for level equation
        GMM-type: LD.n
        Standard: _cons
```

If you are unfamiliar with the L().() notation, see **[U] 13.10 Time-series operators**. That the system estimator produces a much higher estimate of the coefficient on lagged employment agrees with the results in Blundell and Bond (1998), who show that the system estimator does not have the downward bias that the Arellano–Bond estimator has when the true value is high.

Comparing the footers illustrates the difference between the two estimators; xtdpdsys includes lagged differences of n as instruments for the level equation, whereas xtabond does not. Comparing the headers shows that xtdpdsys has seven more instruments than xtabond. (As it should; there are 7 observations on LD.n available in the complete panels that run from 1976–1984, after accounting for the first two years that are lost because the model has two lags.) Only the first lags of the variables are used because the moment conditions using higher lags are redundant; see Blundell and Bond (1998) and Blundell, Bond, and Windmeijer (2000).

`estat abond` reports the Arellano–Bond test for serial correlation in the first-differenced errors. The moment conditions are valid only if there is no serial correlation in the idiosyncratic errors. Because the first difference of independent and identically distributed idiosyncratic errors will be autocorrelated, rejecting the null hypothesis of no serial correlation at order one in the first-differenced errors does not imply that the model is misspecified. Rejecting the null hypothesis at higher orders implies that the moment conditions are not valid. See [XT] **xtdpd** for an alternative estimator in this case.

```
. estat abond
Arellano-Bond test for zero autocorrelation in first-differenced errors
H0: No autocorrelation

Order         z    Prob > z
---------------------------
    1   -4.6414      0.0000
    2   -1.0572      0.2904
---------------------------
```

The above output does not present evidence that the model is misspecified.

◁

▷ Example 2: Including predetermined covariates

Sometimes we cannot assume strict exogeneity. Recall that a variable x_{it} is said to be strictly exogenous if $E[x_{it}\epsilon_{is}] = 0$ for all t and s. If $E[x_{it}\epsilon_{is}] \neq 0$ for $s < t$ but $E[x_{it}\epsilon_{is}] = 0$ for all $s \geq t$, the variable is said to be predetermined. Intuitively, if the error term at time t has some feedback on the subsequent realizations of x_{it}, x_{it} is a predetermined variable. Because unforecastable errors today might affect future changes in the real wage and in the capital stock, we might suspect that the log of the real product wage and the log of the gross capital stock are predetermined instead of strictly exogenous.

```
. xtdpdsys n yr1980-yr1984 year, pre(w k, lag(2, .)) vce(robust)
System dynamic panel-data estimation            Number of obs      =       751
Group variable: id                              Number of groups   =       140
Time variable: year
                                                Obs per group:
                                                              min =          5
                                                              avg =   5.364286
                                                              max =          7
Number of instruments =     95                  Wald chi2(13)      =    7562.80
                                                Prob > chi2        =     0.0000
One-step results
------------------------------------------------------------------------------
             |               Robust
           n | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
           n |
         L1. |    .913278   .0460602    19.83   0.000     .8230017    1.003554
             |
           w |
         --. |   -.728159   .1019044    -7.15   0.000     -.927888   -.5284301
         L1. |   .5602737   .1939617     2.89   0.004     .1801156    .9404317
         L2. |  -.0523028   .1487653    -0.35   0.725    -.3438774    .2392718
             |
           k |
         --. |   .4820097   .0760787     6.34   0.000     .3328983    .6311212
         L1. |  -.2846944   .0831902    -3.42   0.001    -.4477442   -.1216446
         L2. |  -.1394181   .0405709    -3.44   0.001    -.2189356   -.0599006
             |
      yr1980 |  -.0325146   .0216371    -1.50   0.133    -.0749226    .0098935
      yr1981 |  -.0726116   .0346482    -2.10   0.036    -.1405207   -.0047024
      yr1982 |  -.0477038   .0451914    -1.06   0.291    -.1362772    .0408696
      yr1983 |  -.0396264   .0558734    -0.71   0.478    -.1491362    .0698835
      yr1984 |  -.0810383   .0736648    -1.10   0.271    -.2254186     .063342
        year |   .0192741   .0145326     1.33   0.185    -.0092092    .0477574
       _cons |  -37.34972   28.77747    -1.30   0.194    -93.75252    19.05308
------------------------------------------------------------------------------
Instruments for differenced equation
        GMM-type: L(2/.).n L(1/.).L2.w L(1/.).L2.k
        Standard: D.yr1980 D.yr1981 D.yr1982 D.yr1983 D.yr1984 D.year
Instruments for level equation
        GMM-type: LD.n L2D.w L2D.k
        Standard: _cons
```

The footer informs us that we are now including GMM-type instruments from the first lag of `L.w` on back and from the first lag of `L2.k` on back for the differenced errors and the second lags of the differences of `w` and `k` as instruments for the level errors.

◁

❑ Technical note

The above example illustrates that `xtdpdsys` understands `pre(w k, lag(2, .))` to mean that `L2.w` and `L2.k` are predetermined variables. This is a stricter definition than the alternative that `pre(w k, lag(2, .))` means only that `w k` are predetermined but to include two lags of `w` and two lags of `k` in the model. If you prefer the weaker definition, `xtdpdsys` still gives you consistent estimates, but it is not using all possible instruments; see [XT] **xtdpd** for an example of how to include all possible instruments.

❑

Stored results

xtdpdsys stores the following in e():

Scalars	
e(N)	number of observations
e(N_g)	number of groups
e(df_m)	model degrees of freedom
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(t_min)	minimum time in sample
e(t_max)	maximum time in sample
e(chi2)	χ^2
e(arm#)	test for autocorrelation of order #
e(artests)	number of AR tests computed
e(sig2)	estimate of σ_ϵ^2
e(rss)	sum of squared differenced residuals
e(sargan)	Sargan test statistic
e(rank)	rank of e(V)
e(zrank)	rank of instrument matrix
Macros	
e(cmd)	xtdpdsys
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(twostep)	twostep, if specified
e(ivar)	variable denoting groups
e(tvar)	variable denoting time within groups
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(system)	system, if system estimator
e(transform)	specified transform
e(datasignature)	checksum from datasignature
e(datasignaturevars)	variables used in calculation of checksum
e(properties)	b V
e(estat_cmd)	program used to implement estat
e(predict)	program used to implement predict
e(marginsok)	predictions allowed by margins
Matrices	
e(b)	coefficient vector
e(V)	variance–covariance matrix of the estimators
Functions	
e(sample)	marks estimation sample

In addition to the above, the following is stored in r():

Matrices	
r(table)	matrix containing the coefficients with their standard errors, test statistics, *p*-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

xtdpdsys uses xtdpd to perform its computations, so the formulas are given in *Methods and formulas* of [XT] **xtdpd**.

Acknowledgment

We thank David Roodman of the Open Philanthropy Project, who wrote xtabond2.

References

Anderson, T. W., and C. Hsiao. 1981. Estimation of dynamic models with error components. *Journal of the American Statistical Association* 76: 598–606. https://doi.org/10.2307/2287517.

———. 1982. Formulation and estimation of dynamic models using panel data. *Journal of Econometrics* 18: 47–82. https://doi.org/10.1016/0304-4076(82)90095-1.

Arellano, M., and S. Bond. 1991. Some tests of specification for panel data: Monte Carlo evidence and an application to employment equations. *Review of Economic Studies* 58: 277–297. https://doi.org/10.2307/2297968.

Arellano, M., and O. Bover. 1995. Another look at the instrumental variable estimation of error-components models. *Journal of Econometrics* 68: 29–51. https://doi.org/10.1016/0304-4076(94)01642-D.

Baltagi, B. H. 2013. *Econometric Analysis of Panel Data*. 5th ed. Chichester, UK: Wiley.

Blackburne, E. F., III, and M. W. Frank. 2007. Estimation of nonstationary heterogeneous panels. *Stata Journal* 7: 197–208.

Blundell, R. W., and S. Bond. 1998. Initial conditions and moment restrictions in dynamic panel data models. *Journal of Econometrics* 87: 115–143. https://doi.org/10.1016/S0304-4076(98)00009-8.

Blundell, R. W., S. Bond, and F. Windmeijer. 2000. "Estimation in dynamic panel data models: Improving on the performance of the standard GMM estimator". In *Nonstationary Panels, Cointegrating Panels and Dynamic Panels*, edited by B. H. Baltagi, 53–92. New York: Elsevier. https://doi.org/10.1016/S0731-9053(00)15003-0.

Bruno, G. S. F. 2005. Estimation and inference in dynamic unbalanced panel-data models with a small number of individuals. *Stata Journal* 5: 473–500.

De Vos, I., G. Everaert, and I. Ruyssen. 2015. Bootstrap-based bias correction and inference for dynamic panels with fixed effects. *Stata Journal* 15: 986–1018.

Grotti, R., and G. Cutuli. 2018. xtpdyn: A community-contributed command for fitting dynamic random-effects probit models with unobserved heterogeneity. *Stata Journal* 18: 844–862.

Hansen, L. P. 1982. Large sample properties of generalized method of moments estimators. *Econometrica* 50: 1029–1054. https://doi.org/10.2307/1912775.

Holtz-Eakin, D., W. K. Newey, and H. S. Rosen. 1988. Estimating vector autoregressions with panel data. *Econometrica* 56: 1371–1395. https://doi.org/10.2307/1913103.

Keane, M. P., and T. Neal. 2016. The Keane and Runkle estimator for panel-data models with serial correlation and instruments that are not strictly exogenous. *Stata Journal* 16: 523–549.

Kripfganz, S. 2016. Quasi–maximum likelihood estimation of linear dynamic short-T panel-data models. *Stata Journal* 16: 1013–1038.

Layard, R., and S. J. Nickell. 1986. Unemployment in Britain. *Economica* 53: S121–S169. https://doi.org/10.2307/2554377.

Magazzini, L., and G. Calzolari. 2023. A Lagrange multiplier test for the mean stationarity assumption in dynamic panel-data models. *Stata Journal* 23: 418–437.

Neal, T. 2014. Panel cointegration analysis with xtpedroni. *Stata Journal* 14: 684–692.

Sasaki, Y., and Y. Xin. 2022. xtusreg: Software for dynamic panel regression under irregular time spacing. *Stata Journal* 22: 713–724.

Seo, M. H., S. Kim, and Y.-J. Kim. 2019. Estimation of dynamic panel threshold model using Stata. *Stata Journal* 19: 685–697.

Williams, R., P. D. Allison, and E. Moral-Benito. 2018. Linear dynamic panel-data estimation using maximum likelihood and structural equation modeling. *Stata Journal* 18: 293–326.

Windmeijer, F. 2005. A finite sample correction for the variance of linear efficient two-step GMM estimators. *Journal of Econometrics* 126: 25–51. https://doi.org/10.1016/j.jeconom.2004.02.005.

Also see

[XT] **xtdpdsys postestimation** — Postestimation tools for xtdpdsys

[XT] **xtabond** — Arellano–Bond linear dynamic panel-data estimation

[XT] **xtdpd** — Linear dynamic panel-data estimation

[XT] **xtivreg** — Instrumental variables and two-stage least squares for panel-data models

[XT] **xtreg** — Linear models for panel data

[XT] **xtregar** — Fixed- and random-effects linear models with an AR(1) disturbance

[XT] **xtset** — Declare data to be panel data

[U] 20 Estimation and postestimation commands

xtdpdsys postestimation — Postestimation tools for xtdpdsys

Postestimation commands

The following postestimation commands are of special interest after `xtdpdsys`:

Command	Description
`estat abond`	test for autocorrelation
`estat sargan`	Sargan test of overidentifying restrictions

The following standard postestimation commands are also available:

Command	Description
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
`forecast`	dynamic forecasts and simulations
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	linear predictions and their SEs, residual errors
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

predict

Description for predict

`predict` creates a new variable containing predictions such as linear predictions.

Menu for predict

Statistics > Postestimation

Syntax for predict

`predict` [*type*] *newvar* [*if*] [*in*] [`, xb e stdp difference`]

Options for predict

Main

`xb`, the default, calculates the linear prediction.

`e` calculates the residual error.

`stdp` calculates the standard error of the prediction, which can be thought of as the standard error of the predicted expected value or mean for the observation's covariate pattern. The standard error of the prediction is also referred to as the standard error of the fitted value. `stdp` may not be combined with `difference`.

`difference` specifies that the statistic be calculated for the first differences instead of the levels, the default.

margins

Description for margins

`margins` estimates margins of responses for linear predictions.

Menu for margins

Statistics > Postestimation

Syntax for margins

`margins` [*marginlist*] [, *options*]

`margins` [*marginlist*] , `predict(`*statistic* ...`)` [*options*]

statistic	Description
`xb`	linear prediction; the default
`e`	not allowed with `margins`
`stdp`	not allowed with `margins`

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.

For the full syntax, see [R] **margins**.

estat

Description for estat

`estat abond` reports the Arellano–Bond test for serial correlation in the first-differenced residuals.

`estat sargan` reports the Sargan test of the overidentifying restrictions.

Menu for estat

Statistics > Postestimation

Syntax for estat

Test for autocorrelation

`estat abond [ , artests(#) ]`

Sargan test of overidentifying restrictions

`estat sargan`

Option for estat abond

`artests(#)` specifies the highest order of serial correlation to be tested. By default, the tests computed during estimation are reported. The model will be refit when `artests(#)` specifies a higher order than that computed during the original estimation. The model can only be refit if the data have not changed.

Remarks and examples

Remarks are presented under the following headings:

estat abond

The moment conditions used by `xtdpdsys` are valid only if there is no serial correlation in the idiosyncratic errors. Testing for serial correlation in dynamic panel-data models is tricky because one needs to apply a transform to remove the panel-level effects, but the transformed errors have a more complicated error structure than the idiosyncratic errors. The Arellano–Bond test for serial correlation reported by `estat abond` tests for serial correlation in the first-differenced errors.

Because the first difference of independent and identically distributed idiosyncratic errors will be serially correlated, rejecting the null hypothesis of no serial correlation in the first-differenced errors at order one does not imply that the model is misspecified. Rejecting the null hypothesis at higher orders implies that the moment conditions are not valid. See example 5 in [XT] **xtdpd** for an alternative estimator that allows for idiosyncratic errors that follow a first-order moving average process.

After the one-step system estimator, the test can be computed only when `vce(robust)` has been specified.

estat sargan

Like all GMM estimators, the estimator in xtdpdsys can produce consistent estimates only if the moment conditions used are valid. Although there is no method to test if the moment conditions from an exactly identified model are valid, one can test whether the overidentifying moment conditions are valid. estat sargan implements the Sargan test of overidentifying conditions discussed in Arellano and Bond (1991).

Only for a homoskedastic error term does the Sargan test have an asymptotic χ^2 distribution. In fact, Arellano and Bond (1991) show that the one-step Sargan test overrejects in the presence of heteroskedasticity. Because its asymptotic distribution is not known under the assumptions of the vce(robust) model, xtdpdsys does not compute it when vce(robust) is specified. See [XT] **xtdpd** for an example in which the null hypothesis of the Sargan test is not rejected.

```
. use https://www.stata-press.com/data/r19/abdata
. xtdpdsys n L(0/2).(w k) yr1980-yr1984 year
  (output omitted)
. estat sargan
Sargan test of overidentifying restrictions
H0: Overidentifying restrictions are valid

        chi2(33)     =  63.63911
        Prob > chi2  =    0.0011
```

The output above presents strong evidence against the null hypothesis that the overidentifying restrictions are valid. Rejecting this null hypothesis implies that we need to reconsider our model or our instruments, unless we attribute the rejection to heteroskedasticity in the data-generating process. Although performing the Sargan test after the two-step estimator is an alternative, Arellano and Bond (1991) found a tendency for this test to underreject in the presence of heteroskedasticity.

Methods and formulas

The formulas are given in *Methods and formulas* of [XT] **xtdpd postestimation**.

Reference

Arellano, M., and S. Bond. 1991. Some tests of specification for panel data: Monte Carlo evidence and an application to employment equations. *Review of Economic Studies* 58: 277–297. https://doi.org/10.2307/2297968.

Also see

[XT] **xtdpdsys** — Arellano–Bover/Blundell–Bond linear dynamic panel-data estimation

[U] 20 Estimation and postestimation commands

xteintreg — Extended random-effects interval regression

Description

xteintreg fits a random-effects interval-data regression model that accommodates any combination of endogenous covariates, nonrandom treatment assignment, and endogenous sample selection and also accounts for correlation of observations within panels or within groups.

The dependent variable may be measured as point data, interval data, left-censored data, or right-censored data. Continuous, binary, and ordinal endogenous covariates are allowed. Treatment assignment may be endogenous or exogenous. A probit or tobit model may be used to account for endogenous sample selection.

xteintreg fits extended regression models for panel data in the same way that eintreg does for cross-sectional data. See [ERM] **eintreg** to learn about these models and how to fit them using xteintreg.

Quick start

All *Quick start* examples use an interval-measured dependent variable with the interval's lower bound recorded in variable y_l and its upper bound recorded in y_u.

Random-effects regression of [y_l, y_u] on x with continuous endogenous covariate y2 modeled by x and z using xtset data

```
xteintreg y_l y_u x, endogenous(y2 = x z)
```

Same as above, but with binary endogenous covariate d modeled by x and z

```
xteintreg y_l y_u x, endogenous(d = x z, probit)
```

Random-effects regression of [y_l, y_u] on x with endogenous treatment trtvar modeled by x and z

```
xteintreg y_l y_u x, entreat(trtvar = x z)
```

Same as above, but only the equation for [y_l, y_u] has a random effect

```
xteintreg y_l y_u x, entreat(trtvar = x z, nore)
```

Random-effects regression of [y_l, y_u] on x with endogenous sample-selection indicator selvar modeled by x and z

```
xteintreg y_l y_u x, select(selvar = x z)
```

Same as above, but adding endogenous covariate y2 modeled by x and z2

```
xteintreg y_l y_u x, select(selvar = x z) endogenous(y2 = x z2)
```

Menu

Statistics > Longitudinal/panel data > Endogenous covariates > Models adding selection and treatment > Interval regression (RE)

Syntax

For syntax, methods, and all other information on xteintreg, see [ERM] **eintreg**.

xteoprobit — Extended random-effects ordered probit regression

Description

xteoprobit fits a random-effects ordered probit regression model that accommodates any combination of endogenous covariates, nonrandom treatment assignment, and endogenous sample selection and also accounts for correlation of observations within panels or within groups.

Continuous, binary, and ordinal endogenous covariates are allowed. Treatment assignment may be endogenous or exogenous. A probit or tobit model may be used to account for endogenous sample selection.

xteoprobit fits ordered probit extended regression models for panel data in the same way that eoprobit does for cross-sectional data. See [ERM] **eoprobit** to learn about these models and how to fit them using xteoprobit.

Quick start

Random-effects ordered probit regression of y on x with continuous endogenous covariate y2 modeled by x and z using xtset data

```
xteoprobit y x, endogenous(y2 = x z)
```

Same as above, but with binary endogenous covariate d modeled by x and z

```
xteoprobit y x, endogenous(d = x z, probit)
```

Random-effects ordered probit regression of y on x with endogenous treatment trtvar modeled by x and z

```
xteoprobit y x, entreat(trtvar = x z)
```

Same as above, but only the equation for y has a random effect

```
xteoprobit y x, entreat(trtvar = x z, nore)
```

Random-effects ordered probit regression of y on x with endogenous sample-selection indicator selvar modeled by x and z

```
xteoprobit y x, select(selvar = x z)
```

Same as above, but adding endogenous covariate y2 modeled by x and z2

```
xteoprobit y x, select(selvar = x z) endogenous(y2 = x z2)
```

Menu

Statistics > Longitudinal/panel data > Endogenous covariates > Models adding selection and treatment > Ordered probit regression (RE)

Syntax

For syntax, methods, and all other information on xteoprobit, see [ERM] **eoprobit**.

Description

`xteprobit` fits a random-effects probit model that accommodates any combination of endogenous covariates, nonrandom treatment assignment, and endogenous sample selection and also accounts for correlation of observations within panels or within groups.

Continuous, binary, and ordinal endogenous covariates are allowed. Treatment assignment may be endogenous or exogenous. A probit or tobit model may be used to account for endogenous sample selection.

`xteprobit` fits probit extended regression models for panel data in the same way that `eprobit` does for cross-sectional data. See [ERM] **eprobit** to learn about these models and how to fit them using `xteprobit`.

Quick start

Random-effects probit regression of `y` on `x` with continuous endogenous covariate `y2` modeled by `x` and `z` using `xtset` data

```
xteprobit y x, endogenous(y2 = x z)
```

Same as above, but with binary endogenous covariate `d` modeled by `x` and `z`

```
xteprobit y x, endogenous(d = x z, probit)
```

Random-effects probit regression of `y` on `x` with endogenous treatment `trtvar` modeled by `x` and `z`

```
xteprobit y x, entreat(trtvar = x z)
```

Same as above, but only the equation for `y` has a random effect

```
xteprobit x, entreat(trtvar = x z, nore)
```

Random-effects probit regression of `y` on `x` with endogenous sample-selection indicator `selvar` modeled by `x` and `z`

```
xteprobit y x, select(selvar = x z)
```

Same as above, but adding endogenous covariate `y2` modeled by `x` and `z2`

```
xteprobit y x, select(selvar = x z) endogenous(y2 = x z2)
```

Menu

Statistics > Longitudinal/panel data > Endogenous covariates > Models adding selection and treatment > Probit regression (RE)

Syntax

For syntax, methods, and all other information on `xteprobit`, see [ERM] **eprobit**.

xteregress — Extended random-effects linear regression

Description

xteregress fits a random-effects linear regression model that accommodates any combination of endogenous covariates, nonrandom treatment assignment, and endogenous sample selection and also accounts for correlation of observations within panels or within groups.

Continuous, binary, and ordinal endogenous covariates are allowed. Treatment assignment may be endogenous or exogenous. A probit or tobit model may be used to account for endogenous sample selection.

xteregress fits linear extended regression models for panel data in the same way that eregress does for cross-sectional data. See [ERM] **eregress** to learn about these models and how to fit them using xteregress.

Quick start

Random-effects linear regression of y on x with continuous endogenous covariate y2 modeled by x and z using xtset data

```
xteregress y x, endogenous(y2 = x z)
```

Same as above, but with binary endogenous covariate d modeled by x and z

```
xteregress y x, endogenous(d = x z, probit)
```

Random-effects regression of y on x with endogenous treatment trtvar modeled by x and z

```
xteregress y x, entreat(trtvar = x z)
```

Same as above, but only the equation for y has a random effect

```
xteregress y x, entreat(trtvar = x z, nore)
```

Random-effects regression of y on x with endogenous sample-selection indicator selvar modeled by x and z

```
xteregress y x, select(selvar = x z)
```

Same as above, but adding endogenous covariate y2 modeled by x and z2

```
xteregress y x, select(selvar = x z) endogenous(y2 = x z2)
```

Menu

Statistics > Longitudinal/panel data > Endogenous covariates > Models adding selection and treatment > Linear regression (RE)

Syntax

For syntax, methods, and all other information on xteregress, see [ERM] **eregress**.

xtfrontier — Stochastic frontier models for panel data

Description

`xtfrontier` fits stochastic production or cost frontier models for panel data where the disturbance term is a mixture of an inefficiency term and the idiosyncratic error. `xtfrontier` can fit a time-invariant model, in which the inefficiency term is assumed to have a truncated-normal distribution, or a time-varying decay model, in which the inefficiency term is modeled as a truncated-normal random variable multiplied by a function of time.

`xtfrontier` expects that the dependent variable and independent variables are on the natural logarithm scale; this transformation must be performed before estimation takes place.

Quick start

Stochastic production frontier regression of `lny` on `lnx1` and `lnx2` with time-invariant inefficiency using `xtset` data

```
xtfrontier lny lnx1 lnx2, ti
```

Stochastic cost frontier regression of `lny` on `lnx1` and `lnx2` with time-invariant inefficiency

```
xtfrontier lny lnx1 lnx2, ti cost
```

Time-varying decay model for production

```
xtfrontier lny lnx1 lnx2, tvd
```

Menu

Statistics > Longitudinal/panel data > Frontier models

Syntax

Time-invariant model

`xtfrontier` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] , `ti` [*ti_options*]

Time-varying decay model

`xtfrontier` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] , `tvd` [*tvd_options*]

ti_options	Description
Model	
`noconstant`	suppress constant term
`ti`	use time-invariant model
`cost`	fit cost frontier model
`constraints(`*constraints*`)`	apply specified linear constraints
SE	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`nocnsreport`	do not display constraints
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
maximize_options	control the maximization process; seldom used
`collinear`	keep collinear variables
`coeflegend`	display legend instead of statistics

tvd_options	Description
Model	
noconstant	suppress constant term
tvd	use time-varying decay model
cost	fit cost frontier model
constraints(*constraints*)	apply specified linear constraints
SE	
vce(*vcetype*)	*vcetype* may be oim, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
nocnsreport	do not display constraints
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
maximize_options	control the maximization process; seldom used
collinear	keep collinear variables
coeflegend	display legend instead of statistics

A panel variable must be specified. For xtfrontier, tvd, a time variable must also be specified. Use xtset; see [XT] **xtset**.

indepvars may contain factor variables; see [U] **11.4.3 Factor variables**.

depvar and *indepvars* may contain time-series operators; see [U] **11.4.4 Time-series varlists**.

by, collect, fp, and statsby are allowed; see [U] **11.1.10 Prefix commands**.

fweights and iweights are allowed; see [U] **11.1.6 weight**. Weights must be constant within panel.

collinear and coeflegend do not appear in the dialog box.

See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options for time-invariant model

Model

noconstant; see [R] **Estimation options**.

ti specifies that the parameters of the time-invariant technical inefficiency model be estimated.

cost specifies that the frontier model be fit in terms of a cost function instead of a production function. By default, xtfrontier fits a production frontier model.

constraints(*constraints*); see [R] **Estimation options**.

SE

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (oim) and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] ***vce_options***.

Reporting

level(#); see [R] **Estimation options**.

nocnsreport; see [R] **Estimation options**.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Maximization

maximize_options: difficult, technique(*algorithm_spec*) iterate(#), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), nonrtolerance, and from(*init_specs*); see [R] **Maximize**. These options are seldom used.

The following options are available with xtfrontier but are not shown in the dialog box:

collinear, coeflegend; see [R] **Estimation options**.

Options for time-varying decay model

Model

noconstant; see [R] **Estimation options**.

tvd specifies that the parameters of the time-varying decay model be estimated.

cost specifies that the frontier model be fit in terms of a cost function instead of a production function. By default, xtfrontier fits a production frontier model.

constraints(*constraints*); see [R] **Estimation options**.

SE

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (oim) and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] ***vce_options***.

Reporting

level(#); see [R] **Estimation options**.

nocnsreport; see [R] **Estimation options**.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Maximization

maximize_options: difficult, technique(*algorithm_spec*), iterate(#), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), nonrtolerance, and from(*init_specs*); see [R] **Maximize**. These options are seldom used.

The following options are available with `xtfrontier` but are not shown in the dialog box:

`collinear`, `coeflegend`; see [R] **Estimation options**.

Remarks and examples

Remarks are presented under the following headings:

Introduction

Stochastic production frontier models were introduced by Aigner, Lovell, and Schmidt (1977) and Meeusen and van den Broeck (1977). Since then, stochastic frontier models have become a popular subfield in econometrics; see Kumbhakar and Lovell (2000) for an introduction. `xtfrontier` fits two stochastic frontier models with distinct specifications of the inefficiency term and can fit both production- and cost-frontier models.

Let's review the nature of the stochastic frontier problem. Suppose that a producer has a production function $f(\mathbf{z}_{it}, \boldsymbol{\beta})$. In a world without error or inefficiency, in time t, the ith firm would produce

$$q_{it} = f(\mathbf{z}_{it}, \boldsymbol{\beta})$$

A fundamental element of stochastic frontier analysis is that each firm potentially produces less than it might because of a degree of inefficiency. Specifically,

$$q_{it} = f(\mathbf{z}_{it}, \boldsymbol{\beta})\xi_{it}$$

where ξ_{it} is the level of efficiency for firm i at time t; ξ_i must be in the interval $(0, 1\,]$. If $\xi_{it} = 1$, the firm is achieving the optimal output with the technology embodied in the production function $f(\mathbf{z}_{it}, \boldsymbol{\beta})$. When $\xi_{it} < 1$, the firm is not making the most of the inputs $\mathbf{z}_{it}$ given the technology embodied in the production function $f(\mathbf{z}_{it}, \boldsymbol{\beta})$. Because the output is assumed to be strictly positive (that is, $q_{it} > 0$), the degree of technical efficiency is assumed to be strictly positive (that is, $\xi_{it} > 0$).

Output is also assumed to be subject to random shocks, implying that

$$q_{it} = f(\mathbf{z}_{it}, \boldsymbol{\beta})\xi_{it}\exp(v_{it})$$

Taking the natural log of both sides yields

$$\ln(q_{it}) = \ln\{f(\mathbf{z}_{it}, \boldsymbol{\beta})\} + \ln(\xi_{it}) + v_{it}$$

Assuming that there are k inputs and that the production function is linear in logs, defining $u_{it} = -\ln(\xi_{it})$ yields

$$\ln(q_{it}) = \beta_0 + \sum_{j=1}^{k} \beta_j \ln(z_{jit}) + v_{it} - u_{it} \tag{1}$$

Because u_{it} is subtracted from $\ln(q_{it})$, restricting $u_{it} \geq 0$ implies that $0 < \xi_{it} \leq 1$, as specified above.

Kumbhakar and Lovell (2000) provide a detailed version of this derivation, and they show that performing an analogous derivation in the dual cost function problem allows us to specify the problem as

$$\ln(c_{it}) = \beta_0 + \beta_q \ln(q_{it}) + \sum_{j=1}^{k} \beta_j \ln(p_{jit}) + v_{it} - su_{it} \tag{2}$$

where q_{it} is output, the z_{jit} are input quantities, c_{it} is cost, the p_{jit} are input prices, and

$$s = \begin{cases} 1, & \text{for production functions} \\ -1, & \text{for cost functions} \end{cases}$$

Intuitively, the inefficiency effect is required to lower output or raise expenditure, depending on the specification.

❏ Technical note

The model that `xtfrontier` actually fits has the form

$$y_{it} = \beta_0 + \sum_{j=1}^{k} \beta_j x_{jit} + v_{it} - su_{it}$$

so in the context of the discussion above, $y_{it} = \ln(q_{it})$ and $x_{jit} = \ln(z_{jit})$ for a production function; for a cost function, $y_{it} = \ln(c_{it})$, the x_{jit} are the $\ln(p_{jit})$, and $\ln(q_{it})$. You must perform the natural logarithm transformation of the data before estimation to interpret the estimation results correctly for a stochastic frontier production or cost model. `xtfrontier` does not perform any transformations on the data.

❏

As shown above, the disturbance term in a stochastic frontier model is assumed to have two components. One component is assumed to have a strictly nonnegative distribution, and the other component is assumed to have a symmetric distribution. In the econometrics literature, the nonnegative component is often referred to as the *inefficiency term*, and the component with the symmetric distribution as the *idiosyncratic error*.

Equation (2) is a variant of a panel-data model in which v_{it} is the idiosyncratic error and u_{it} is a time-varying panel-level effect. Much of the literature on this model has focused on deriving estimators for different specifications of the u_{it} term. Kumbhakar and Lovell (2000) provide a survey of this literature.

`xtfrontier` provides estimators for two different specifications of u_{it}. To facilitate the discussion, let $N^+(\mu, \sigma^2)$ denote the truncated-normal distribution, which is truncated at zero with mean μ and variance σ^2, and let $\overset{\text{iid}}{\sim}$ stand for independent and identically distributed.

Consider the simplest specification in which the inefficiency term u_{it} is a time-invariant truncated-normal random variable. In the time-invariant model, $u_{it} = u_i$, $u_i \overset{\text{iid}}{\sim} N^+(\mu, \sigma_u^2)$, $v_{it} \overset{\text{iid}}{\sim} N(0, \sigma_v^2)$, and u_i and v_{it} are distributed independently of each other and the covariates in the model. Specifying the `ti` option causes `xtfrontier` to estimate the parameters of this model.

In the Battese–Coelli (1992) parameterization of time effects, the inefficiency term is modeled as a truncated-normal random variable multiplied by a specific function of time. In the time-varying decay specification,

$$u_{it} = \exp\{-\eta(t - T_i)\}u_i$$

where T_i is the last period in the ith panel, η is the decay parameter, $u_i \stackrel{\text{iid}}{\sim} N^+(\mu, \sigma_u^2)$, $v_{it} \stackrel{\text{iid}}{\sim} N(0, \sigma_v^2)$, and u_i and v_{it} are distributed independently of each other and the covariates in the model. Specifying the `tvd` option causes `xtfrontier` to estimate the parameters of this model.

Time-invariant model

▷ Example 1

`xtfrontier, ti` provides maximum likelihood estimates for the parameters of the time-invariant decay model. In this model, the inefficiency effects are modeled as $u_{it} = u_i$, $u_i \stackrel{\text{iid}}{\sim} N^+(\mu, \sigma_u^2)$, $v_{it} \stackrel{\text{iid}}{\sim} N(0, \sigma_v^2)$, and u_i and v_{it} are distributed independently of each other and the covariates in the model. In this example, firms produce a product called a widget, using a constant-returns-to-scale technology. We have 948 observations—91 firms, with 6–14 observations per firm. Our dataset contains variables representing the quantity of widgets produced, the number of machine hours used in production, the number of labor hours used in production, and three additional variables that are the natural logarithm transformations of the three aforementioned variables.

We fit a time-invariant model using the transformed variables:

```
. use https://www.stata-press.com/data/r19/xtfrontier1
. xtfrontier lnwidgets lnmachines lnworkers, ti
Iteration 0:  Log likelihood = -1473.8703
Iteration 1:  Log likelihood = -1473.0565
Iteration 2:  Log likelihood = -1472.6155
Iteration 3:  Log likelihood =  -1472.607
Iteration 4:  Log likelihood = -1472.6069
Time-invariant inefficiency model                    Number of obs    =    948
Group variable: id                                   Number of groups =     91
                                                     Obs per group:
                                                                  min =      6
                                                                  avg =   10.4
                                                                  max =     14
                                                     Wald chi2(2)     = 661.76
Log likelihood = -1472.6069                          Prob > chi2      = 0.0000
```

lnwidgets	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
lnmachines	.2904551	.0164219	17.69	0.000	.2582688	.3226415
lnworkers	.2943333	.0154352	19.07	0.000	.2640808	.3245858
_cons	3.030983	.1441022	21.03	0.000	2.748548	3.313418
/mu	1.125667	.6479217	1.74	0.082	-.144236	2.39557
/lnsigma2	1.421979	.2672745	5.32	0.000	.898131	1.945828
/lgtgamma	1.138685	.3562642	3.20	0.001	.4404204	1.83695
sigma2	4.145318	1.107938			2.455011	6.999424
gamma	.7574382	.0654548			.6083592	.8625876
sigma_u2	3.139822	1.107235			.9696821	5.309962
sigma_v2	1.005496	.0484143			.9106055	1.100386

In addition to the coefficients, the output reports estimates for the parameters `sigma_v2`, `sigma_u2`, `gamma`, `sigma2`, `lgtgamma`, `lnsigma2`, and `mu`. `sigma_v2` is the estimate of σ_v^2. `sigma_u2` is the estimate of σ_u^2. `gamma` is the estimate of $\gamma = \sigma_u^2/\sigma_S^2$. `sigma2` is the estimate of $\sigma_S^2 = \sigma_v^2 + \sigma_u^2$. Because γ must be between 0 and 1, the optimization is parameterized in terms of the logit of γ, and this estimate is reported as `lgtgamma`. Because σ_S^2 must be positive, the optimization is parameterized in terms of $\ln(\sigma_S^2)$, and this estimate is reported as `lnsigma2`. Finally, `mu` is the estimate of μ.

◁

❑ Technical note

Our simulation results indicate that this estimator requires relatively large samples to achieve any reasonable degree of precision in the estimates of μ and σ_u^2.

❑

Time-varying decay model

`xtfrontier, tvd` provides maximum likelihood estimates for the parameters of the time-varying decay model. In this model, the inefficiency effects are modeled as

$$u_{it} = \exp\{-\eta(t - T_i)\}u_i$$

where $u_i \overset{\text{iid}}{\sim} N^+(\mu, \sigma_u^2)$.

When $\eta > 0$, the degree of inefficiency decreases over time; when $\eta < 0$, the degree of inefficiency increases over time. Because $t = T_i$ in the last period, the last period for firm i contains the base level of inefficiency for that firm. If $\eta > 0$, the level of inefficiency decays toward the base level. If $\eta < 0$, the level of inefficiency increases to the base level.

▷ Example 2

When $\eta = 0$, the time-varying decay model reduces to the time-invariant model. The following example illustrates this property and demonstrates how to specify constraints and starting values in these models.

Let's begin by fitting the time-varying decay model on the same data that were used in the previous example for the time-invariant model.

```
. xtfrontier lnwidgets lnmachines lnworkers, tvd
Iteration 0:  Log likelihood = -1551.3798  (not concave)
Iteration 1:  Log likelihood = -1502.2637
Iteration 2:  Log likelihood = -1476.3093  (not concave)
Iteration 3:  Log likelihood = -1472.9845
Iteration 4:  Log likelihood = -1472.5365
Iteration 5:  Log likelihood =  -1472.529
Iteration 6:  Log likelihood = -1472.5289
Time-varying decay inefficiency model           Number of obs    =     948
Group variable: id                              Number of groups =      91
Time variable: t                                Obs per group:
                                                             min =       6
                                                             avg =    10.4
                                                             max =      14
                                                Wald chi2(2)     =  661.93
Log likelihood = -1472.5289                     Prob > chi2      =  0.0000
```

lnwidgets	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
lnmachines	.2907555	.0164376	17.69	0.000	.2585384	.3229725
lnworkers	.2942412	.0154373	19.06	0.000	.2639846	.3244978
_cons	3.028939	.1436046	21.09	0.000	2.74748	3.310399
/mu	1.110831	.6452809	1.72	0.085	-.1538967	2.375558
/eta	.0016764	.00425	0.39	0.693	-.0066535	.0100064
/lnsigma2	1.410723	.2679485	5.26	0.000	.885554	1.935893
/lgtgamma	1.123982	.3584243	3.14	0.002	.4214828	1.82648
sigma2	4.098919	1.098299			2.424327	6.930228
gamma	.7547265	.0663495			.603838	.8613419
sigma_u2	3.093563	1.097606			.9422943	5.244832
sigma_v2	1.005356	.0484079			.9104785	1.100234

The estimate of η is close to zero, and the other estimates are not too far from those of the time-invariant model.

We can use `constraint` to constrain $\eta = 0$ and obtain the same results produced by the time-invariant model. Although there is only one statistical equation to be estimated in this model, the model fits five of Stata's [R] **ml** equations; see [R] **ml** or Pitblado, Poi, and Gould (2024). The equation names can be seen by listing the matrix of estimated coefficients.

```
. matrix list e(b)
e(b)[1,7]
     lnwidgets:  lnwidgets:  lnwidgets:   lnsigma2:   lgtgamma:          mu:
    lnmachines   lnworkers       _cons       _cons       _cons       _cons
y1   .29075546    .2942412   3.0289395   1.4107233   1.1239816   1.1108307

           eta:
         _cons
y1   .00167642
```

To constrain a parameter to a particular value in any equation, except the first equation, you must specify both the equation name and the parameter name by using the syntax

`constraint` # `[`*eqname*`]_b[`*varname*`]` = *value*   or

`constraint` # `[`*eqname*`]`*coefficient* = *value*

where *eqname* is the equation name, *varname* is the name of the variable in a linear equation, and *coefficient* refers to any parameter that has been estimated. More elaborate specifications with expressions are possible; see the example with constant returns to scale below, and see [R] **constraint** for general reference.

Suppose that we impose the constraint $\eta = 0$; we get the same results as those reported above for the time-invariant model, except for some minute differences attributable to an alternate convergence path in the optimization.

```
. constraint 1 [eta]_cons = 0
. xtfrontier lnwidgets lnmachines lnworkers, tvd constraints(1)
Iteration 0:  Log likelihood = -1540.7124  (not concave)
Iteration 1:  Log likelihood = -1515.7726
Iteration 2:  Log likelihood = -1473.0162
Iteration 3:  Log likelihood = -1472.9223
Iteration 4:  Log likelihood = -1472.6254
Iteration 5:  Log likelihood =  -1472.607
Iteration 6:  Log likelihood = -1472.6069
Time-varying decay inefficiency model            Number of obs     =    948
Group variable: id                               Number of groups =     91
Time variable: t                                 Obs per group:
                                                              min =      6
                                                              avg =   10.4
                                                              max =     14
                                                 Wald chi2(2)      = 661.76
Log likelihood = -1472.6069                      Prob > chi2       = 0.0000
 ( 1)  [eta]_cons = 0
```

lnwidgets	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
lnmachines	.2904551	.0164219	17.69	0.000	.2582688	.3226414
lnworkers	.2943332	.0154352	19.07	0.000	.2640807	.3245857
_cons	3.030963	.1440995	21.03	0.000	2.748534	3.313393
/mu	1.125507	.6480444	1.74	0.082	-.1446369	2.39565
/eta	0	(omitted)				
/lnsigma2	1.422039	.2673128	5.32	0.000	.8981155	1.945962
/lgtgamma	1.138764	.3563076	3.20	0.001	.4404135	1.837114
sigma2	4.145565	1.108162			2.454972	7.000366
gamma	.7574526	.0654602			.6083575	.862607
sigma_u2	3.140068	1.107459			.9694878	5.310649
sigma_v2	1.005496	.0484143			.9106057	1.100386

◁

Stored results

`xtfrontier` stores the following in `e()`:

Scalars	
`e(N)`	number of observations
`e(N_g)`	number of groups
`e(k)`	number of parameters
`e(k_eq)`	number of equations in `e(b)`
`e(k_eq_model)`	number of equations in overall model test
`e(k_dv)`	number of dependent variables
`e(df_m)`	model degrees of freedom
`e(ll)`	log likelihood
`e(g_min)`	minimum number of observations per group
`e(g_avg)`	average number of observations per group
`e(g_max)`	maximum number of observations per group
`e(sigma2)`	sigma2
`e(gamma)`	gamma
`e(Tcon)`	`1` if panels balanced, `0` otherwise
`e(sigma_u)`	standard deviation of technical inefficiency
`e(sigma_v)`	standard deviation of random error
`e(chi2)`	χ^2
`e(p)`	p-value for model test
`e(rank)`	rank of `e(V)`
`e(ic)`	number of iterations
`e(rc)`	return code
`e(converged)`	`1` if converged, `0` otherwise
Macros	
`e(cmd)`	`xtfrontier`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(ivar)`	variable denoting groups
`e(tvar)`	variable denoting time within groups
`e(function)`	`production` or `cost`
`e(model)`	`ti`, after time-invariant model; `tvd`, after time-varying decay model
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(title)`	title in estimation output
`e(chi2type)`	`Wald`; type of model χ^2 test
`e(vce)`	*vcetype* specified in `vce()`
`e(opt)`	type of optimization
`e(which)`	`max` or `min`; whether optimizer is to perform maximization or minimization
`e(ml_method)`	type of `ml` method
`e(user)`	name of likelihood-evaluator program
`e(technique)`	maximization technique
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`
Matrices	
`e(b)`	coefficient vector
`e(Cns)`	constraints matrix
`e(ilog)`	iteration log (up to 20 iterations)
`e(gradient)`	gradient vector
`e(V)`	variance–covariance matrix of the estimators
Functions	
`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices
`r(table)` matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

`xtfrontier` fits stochastic frontier models for panel data that can be expressed as

$$y_{it} = \beta_0 + \sum_{j=1}^{k} \beta_j x_{jit} + v_{it} - s u_{it}$$

where y_{it} is the natural logarithm of output, the x_{jit} are the natural logarithm of the input quantities for the production efficiency problem, y_{it} is the natural logarithm of costs, the x_{it} are the natural logarithm of input prices for the cost efficiency problem, and

$$s = \begin{cases} 1, & \text{for production functions} \\ -1, & \text{for cost functions} \end{cases}$$

For the time-varying decay model, the log-likelihood function is derived as

$$\begin{aligned} \ln L = &-\frac{1}{2}\left(\sum_{i=1}^{N} T_i\right)\{\ln(2\pi) + \ln(\sigma_S^2)\} - \frac{1}{2}\sum_{i=1}^{N}(T_i - 1)\ln(1-\gamma) \\ &- \frac{1}{2}\sum_{i=1}^{N} \ln\left\{1 + \left(\sum_{t=1}^{T_i}\eta_{it}^2 - 1\right)\gamma\right\} - N\ln\{1-\Phi(-\tilde{z})\} - \frac{1}{2}N\tilde{z}^2 \\ &+ \sum_{i=1}^{N}\ln\{1-\Phi(-z_i^*)\} + \frac{1}{2}\sum_{i=1}^{N} z_i^{*2} - \frac{1}{2}\sum_{i=1}^{N}\sum_{t=1}^{T_i}\frac{\epsilon_{it}^2}{(1-\gamma)\,\sigma_S^2} \end{aligned}$$

where $\sigma_S = (\sigma_u^2 + \sigma_v^2)^{1/2}$, $\gamma = \sigma_u^2/\sigma_S^2$, $\epsilon_{it} = y_{it} - \mathbf{x}_{it}\boldsymbol{\beta}$, $\eta_{it} = \exp\{-\eta(t - T_i)\}$, $\tilde{z} = \mu/\left(\gamma\sigma_S^2\right)^{1/2}$, $\Phi()$ is the cumulative distribution function of the standard normal distribution, and

$$z_i^* = \frac{\mu(1-\gamma) - s\gamma\sum_{t=1}^{T_i}\eta_{it}\epsilon_{it}}{\left[\gamma(1-\gamma)\,\sigma_S^2\left\{1 + \left(\sum_{t=1}^{T_i}\eta_{it}^2 - 1\right)\gamma\right\}\right]^{1/2}}$$

Maximizing the above log likelihood estimates the coefficients η, μ, σ_v, and σ_u.

References

Aigner, D. J., C. A. K. Lovell, and P. Schmidt. 1977. Formulation and estimation of stochastic frontier production function models. *Journal of Econometrics* 6: 21–37. https://doi.org/10.1016/0304-4076(77)90052-5.

Battese, G. E., and T. J. Coelli. 1992. Frontier production functions, technical efficiency and panel data: With application to paddy farmers in India. *Journal of Productivity Analysis* 3: 153–169. https://doi.org/10.1007/BF00158774.

———. 1995. A model for technical inefficiency effects in a stochastic frontier production function for panel data. *Empirical Economics* 20: 325–332. https://doi.org/10.1007/BF01205442.

Belotti, F., S. Daidone, G. Ilardi, and V. Atella. 2013. Stochastic frontier analysis using Stata. *Stata Journal* 13: 719–758.

Caudill, S. B., J. M. Ford, and D. M. Gropper. 1995. Frontier estimation and firm-specific inefficiency measures in the presence of heteroscedasticity. *Journal of Business and Economic Statistics* 13: 105–111. https://doi.org/10.2307/1392525.

Cococcioni, M., M. Grazzi, L. Li, and F. Ponchio. 2022. A toolbox for measuring heterogeneity and efficiency using zonotopes. *Stata Journal* 22: 25–59.

Coelli, T. J. 1995. Estimators and hypothesis tests for a stochastic frontier function: A Monte Carlo analysis. *Journal of Productivity Analysis* 6: 247–268. https://doi.org/10.1007/BF01076978.

Coelli, T. J., D. S. P. Rao, C. J. O'Donnell, and G. E. Battese. 2005. *An Introduction to Efficiency and Productivity Analysis*. 2nd ed. New York: Springer. https://doi.org/10.1007/b136381.

Fé, E., and R. Hofler. 2020. sfcount: Command for count-data stochastic frontiers and underreported and overreported counts. *Stata Journal* 20: 532–547.

Karakaplan, M. U. 2017. Fitting endogenous stochastic frontier models in Stata. *Stata Journal* 17: 39–55.

———. 2022. Panel stochastic frontier models with endogeneity. *Stata Journal* 22: 643–663.

Kumbhakar, S. C., and C. A. K. Lovell. 2000. *Stochastic Frontier Analysis*. Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9781139174411.

Kumbhakar, S. C., H.-J. Wang, and A. P. Horncastle. 2015. *A Practitioner's Guide to Stochastic Frontier Analysis Using Stata*. New York: Cambridge University Press.

Meeusen, W., and J. van den Broeck. 1977. Efficiency estimation from Cobb–Douglas production functions with composed error. *International Economic Review* 18: 435–444. https://doi.org/10.2307/2525757.

Pitblado, J. S., B. P. Poi, and W. W. Gould. 2024. *Maximum Likelihood Estimation with Stata*. 5th ed. College Station, TX: Stata Press.

Rovigatti, G., and V. Mollisi. 2018. Theory and practice of total-factor productivity estimation: The control function approach using Stata. *Stata Journal* 18: 618–662.

Wang, D., K. Du, and N. Zhang. 2022. Measuring technical efficiency and total factor productivity change with undesirable outputs in Stata. *Stata Journal* 22: 103–124.

Zellner, A., and N. S. Revankar. 1969. Generalized production functions. *Review of Economic Studies* 36: 241–250. https://doi.org/10.2307/2296840.

Also see

[XT] **xtfrontier postestimation** — Postestimation tools for xtfrontier

[XT] **xtset** — Declare data to be panel data

[R] **frontier** — Stochastic frontier models

[U] 20 Estimation and postestimation commands

Postestimation commands   predict   margins   Remarks and examples
Methods and formulas   Also see

Postestimation commands

The following postestimation commands are available after `xtfrontier`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of parameters
`estat ic`	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC, respectively)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
`forecast`	dynamic forecasts and simulations
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
`lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	linear predictions and their SEs, technical efficiency
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of parameters
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

predict

Description for predict

predict creates a new variable containing predictions such as linear predictions, standard errors, and technical efficiencies.

Menu for predict

Statistics > Postestimation

Syntax for predict

predict [*type*] *newvar* [*if*] [*in*] [, *statistic*]

statistic	Description
Main	
xb	linear prediction; the default
stdp	standard error of the linear prediction
u	minus the natural log of the technical efficiency via $E(u_{it} \mid \epsilon_{it})$
m	minus the natural log of the technical efficiency via $M(u_{it} \mid \epsilon_{it})$
te	the technical efficiency via $E\{\exp(-su_{it}) \mid \epsilon_{it}\}$

where

$$s = \begin{cases} 1, & \text{for production functions} \\ -1, & \text{for cost functions} \end{cases}$$

Options for predict

Main

xb, the default, calculates the linear prediction.

stdp calculates the standard error of the linear prediction.

u produces estimates of minus the natural log of the technical efficiency via $E(u_{it} \mid \epsilon_{it})$.

m produces estimates of minus the natural log of the technical efficiency via the mode, $M(u_{it} \mid \epsilon_{it})$.

te produces estimates of the technical efficiency via $E\{\exp(-su_{it}) \mid \epsilon_{it}\}$.

margins

Description for margins

`margins` estimates margins of response for linear predictions.

Menu for margins

Statistics > Postestimation

Syntax for margins

margins [*marginlist*] [, *options*]

margins [*marginlist*] , predict(*statistic* ...) [*options*]

statistic	Description
xb	linear prediction; the default
stdp	not allowed with margins
u	not allowed with margins
m	not allowed with margins
te	not allowed with margins

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.

For the full syntax, see [R] **margins**.

Remarks and examples

▷ Example 1

A production function exhibits *constant returns to scale* if doubling the amount of each input results in a doubling in the quantity produced. When the production function is linear in logs, constant returns to scale implies that the sum of the coefficients on the inputs is one. In example 2 of [XT] **xtfrontier**, we fit a time-varying decay model. Here we test whether the estimated production function exhibits constant returns:

```
. use https://www.stata-press.com/data/r19/xtfrontier1
. xtfrontier lnwidgets lnmachines lnworkers, tvd
  (output omitted)
. test lnmachines + lnworkers = 1
 ( 1)  [lnwidgets]lnmachines + [lnwidgets]lnworkers = 1
           chi2(  1) =  331.55
         Prob > chi2 =    0.0000
```

The test statistic is highly significant, so we reject the null hypothesis and conclude that this production function does not exhibit constant returns to scale.

The previous Wald χ^2 test indicated that the sum of the coefficients does not equal one. An alternative is to use `lincom` to compute the sum explicitly:

```
. lincom lnmachines + lnworkers
 ( 1)  [lnwidgets]lnmachines + [lnwidgets]lnworkers = 0
```

lnwidgets	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
(1)	.5849967	.0227918	25.67	0.000	.5403256	.6296677

The sum of the coefficients is significantly less than one, so this production function exhibits *decreasing returns to scale*. If we doubled the number of machines and workers, we would obtain less than twice as much output.

◁

Methods and formulas

Continuing from the *Methods and formulas* section of [XT] **xtfrontier**, estimates for u_{it} can be obtained from the mean or the mode of the conditional distribution $f(u|\epsilon)$.

$$E(u_{it} \mid \epsilon_{it}) = \tilde{\mu}_i + \tilde{\sigma}_i \left\{ \frac{\phi(-\tilde{\mu}_i/\tilde{\sigma}_i)}{1 - \Phi(-\tilde{\mu}_i/\tilde{\sigma}_i)} \right\}$$

$$M(u_{it} \mid \epsilon_{it}) = \begin{cases} -\tilde{\mu}_i, & \text{if } \tilde{\mu}_i >= 0 \\ 0, & \text{otherwise} \end{cases}$$

where

$$\tilde{\mu}_i = \frac{\mu\sigma_v^2 - s\sum_{t=1}^{T_i} \eta_{it}\epsilon_{it}\sigma_u^2}{\sigma_v^2 + \sum_{t=1}^{T_i} \eta_{it}^2\sigma_u^2}$$

$$\tilde{\sigma}_i^2 = \frac{\sigma_v^2\sigma_u^2}{\sigma_v^2 + \sum_{t=1}^{T_i} \eta_{it}^2\sigma_u^2}$$

These estimates can be obtained from `predict` *newvar*`, u` and `predict` *newvar*`, m`, respectively, and are calculated by plugging in the estimated parameters.

`predict` *newvar*`, te` produces estimates of the technical-efficiency term. These estimates are obtained from

$$E\{\exp(-su_{it}) \mid \epsilon_{it}\} = \left[\frac{1 - \Phi\{s\eta_{it}\tilde{\sigma}_i - (\tilde{\mu}_i/\,\tilde{\sigma}_i)\}}{1 - \Phi(-\tilde{\mu}_i/\,\tilde{\sigma}_i)} \right] \exp\left(-s\eta_{it}\tilde{\mu}_i + \frac{1}{2}\eta_{it}^2\tilde{\sigma}_i^2 \right)$$

Replacing $\eta_{it} = 1$ and $\eta = 0$ in these formulas produces the formulas for the time-invariant models.

Also see

[XT] **xtfrontier** — Stochastic frontier models for panel data

[U] 20 Estimation and postestimation commands

xtgee — GEE population-averaged panel-data models

Description

`xtgee` fits population-averaged panel-data models. In particular, `xtgee` fits generalized linear models and allows you to specify the within-group correlation structure for the panels.

Quick start

Population-averaged linear regression of `y` on `x1` and `x2`

```
xtgee y x1 x2
```

Same as above, but estimate time-varying intragroup correlations

```
xtgee y x1 x2, corr(unstructured)
```

Same as above, but estimate a common second-order autoregression structure for the within-panel correlation

```
xtgee y x1 x2, corr(ar 2)
```

Population-averaged negative binomial regression of `y2` on `x3` and `x4` equivalent to `xtnbreg, pa`

```
xtgee y2 x3 x4, family(nbinomial 1)
```

Population-averaged logistic regression of `y3` on `x5` and `x6` when `y3` is the number of events out of 10 trials

```
xtgee y3 x5 x6, family(binomial 10)
```

Menu

Statistics > Longitudinal/panel data > Generalized estimating equations (GEE) > Generalized estimating equations (GEE)

Syntax

`xtgee` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, *options*]

options	Description
Model	
`family(`*family*`)`	distribution of *depvar*
`link(`*link*`)`	link function
Model 2	
`exposure(`*varname*`)`	include ln(*varname*) in model with coefficient constrained to 1
`offset(`*varname*`)`	include *varname* in model with coefficient constrained to 1
`noconstant`	suppress constant term
`asis`	retain perfect predictor variables
`force`	estimate even if observations unequally spaced in time
Correlation	
`corr(`*correlation*`)`	within-group correlation structure
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `conventional`, `robust`, `bootstrap`, or `jackknife`
`nmp`	use divisor $N - P$ instead of the default N
`rgf`	multiply the robust variance estimate by $(N-1)/(N-P)$
`scale(`*parm*`)`	overrides the default scale parameter; *parm* may be `x2`, `dev`, `phi`, or #
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`eform`	report exponentiated coefficients
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Optimization	
optimize_options	control the optimization process; seldom used
`nodisplay`	suppress display of header and coefficients
`coeflegend`	display legend instead of statistics

A panel variable must be specified. Correlation structures other than `exchangeable` and `independent` require that a time variable also be specified. Use `xtset`; see [XT] **xtset**.

indepvars may contain factor variables; see [U] **11.4.3 Factor variables**.

depvar and *indepvars* may contain time-series operators; see [U] **11.4 varname and varlists**.

`by`, `collect`, `mfp`, `mi estimate`, and `statsby` are allowed; see [U] **11.1.10 Prefix commands**.

`vce(bootstrap)` and `vce(jackknife)` are not allowed with the `mi estimate` prefix; see [MI] **mi estimate**.

`iweight`s, `fweight`s, and `pweight`s are allowed; see [U] **11.1.6 weight**. Weights must be constant within panel.

`nodisplay` and `coeflegend` do not appear in the dialog box.

See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

family	Description
gaussian	Gaussian (normal); family(normal) is a synonym
igaussian	inverse Gaussian
binomial [#| *varname*]	Bernoulli/binomial
poisson	Poisson
nbinomial [#]	negative binomial
gamma	gamma

link	Link function/definition
identity	identity; $y = y$
log	log; $\ln(y)$
logit	logit; $\ln\{y/(1-y)\}$, natural log of the odds
probit	probit; $\Phi^{-1}(y)$, where $\Phi(\,)$ is the normal cumulative distribution
cloglog	cloglog; $\ln\{-\ln(1-y)\}$
power [#]	power; y^k with $k = \#$; $\# = 1$ if not specified
opower [#]	odds power; $[\{y/(1-y)\}^k - 1]/k$ with $k = \#$; $\# = 1$ if not specified
nbinomial	negative binomial; $\ln\{y/(y+\alpha)\}$
reciprocal	reciprocal; $1/y$

correlation	Description
exchangeable	exchangeable
independent	independent
unstructured	unstructured
fixed *matname*	user-specified
ar #	autoregressive of order #
stationary #	stationary of order #
nonstationary #	nonstationary of order #

Options

Model

family(*family*) specifies the distribution of *depvar*; family(gaussian) is the default.

link(*link*) specifies the link function; the default is the canonical link for the family() specified (except for family(nbinomial)).

Model 2

exposure(*varname*) and offset(*varname*) are different ways of specifying the same thing. exposure() specifies a variable that reflects the amount of exposure over which the *depvar* events were observed for each observation; ln(*varname*) with coefficient constrained to be 1 is entered into the regression equation. offset() specifies a variable that is to be entered directly into the log-link function with its coefficient constrained to be 1; thus, exposure is assumed to be $e^{varname}$. If you were fitting a Poisson regression model, family(poisson) link(log), for instance, you would account for exposure time by specifying offset() containing the log of exposure time.

`noconstant` specifies that the linear predictor has no intercept term, thus forcing it through the origin on the scale defined by the link function.

`asis` forces retention of perfect predictor variables and their associated, perfectly predicted observations and may produce instabilities in maximization; see [R] **probit**. This option is only allowed with option `family(binomial)` with a denominator of 1.

`force` specifies that estimation be forced even though the time variable is not equally spaced. This is relevant only for correlation structures that require knowledge of the time variable. These correlation structures require that observations be equally spaced so that calculations based on lags correspond to a constant time change. If you specify a time variable indicating that observations are not equally spaced, the (time dependent) model will not be fit. If you also specify `force`, the model will be fit, and it will be assumed that the lags based on the data ordered by the time variable are appropriate.

Correlation

`corr(`*correlation*`)` specifies the within-group correlation structure; the default corresponds to the equal-correlation model, `corr(exchangeable)`.

When you specify a correlation structure that requires a lag, you indicate the lag after the structure's name with or without a blank; for example, `corr(ar 1)` or `corr(ar1)`.

If you specify the fixed correlation structure, you specify the name of the matrix containing the assumed correlations following the word `fixed`, for example, `corr(fixed myr)`.

SE/Robust

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`conventional`), that are robust to some kinds of misspecification (`robust`), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [XT] ***vce_options***.

`vce(conventional)`, the default, uses the conventionally derived variance estimator for generalized least-squares regression.

`vce(robust)` specifies that the Huber/White/sandwich estimator of variance is to be used in place of the default conventional variance estimator (see *Methods and formulas* below). Use of this option causes `xtgee` to produce valid standard errors even if the correlations within group are not as hypothesized by the specified correlation structure. Under a noncanonical link, it does, however, require that the model correctly specifies the mean. The resulting standard errors are thus labeled "semirobust" instead of "robust" in this case. Although there is no `vce(cluster` *clustvar*`)` option, results are as if this option were included and you specified clustering on the panel variable.

`nmp`; see [XT] ***vce_options***.

`rgf` specifies that the robust variance estimate is multiplied by $(N-1)/(N-P)$, where N is the total number of observations and P is the number of coefficients estimated. This option can be used only with `family(gaussian)` when `vce(robust)` is either specified or implied by the use of `pweights`. Using this option implies that the robust variance estimate is not invariant to the scale of any weights used.

`scale(x2 | dev | phi | #)`; see [XT] ***vce_options***.

Reporting

`level(#)`; see [R] **Estimation options**.

eform displays the exponentiated coefficients and corresponding standard errors and confidence intervals as described in [R] **Maximize**. For family(binomial) link(logit) (that is, logistic regression), exponentiation results in odds ratios; for family(poisson) link(log) (that is, Poisson regression), exponentiated coefficients are incidence-rate ratios.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Optimization

optimize_options control the iterative optimization process. These options are seldom used.

iterate(#) specifies the maximum number of iterations. When the number of iterations equals #, the optimization stops and presents the current results, even if convergence has not been reached. The default is iterate(100).

tolerance(#) specifies the tolerance for the coefficient vector. When the relative change in the coefficient vector from one iteration to the next is less than or equal to #, the optimization process is stopped. tolerance(1e-6) is the default.

log and nolog specify whether to display the iteration log. The iteration log is displayed by default unless you used set iterlog off to suppress it; see set iterlog in [R] ***set iter***.

trace specifies that the current estimates be printed at each iteration.

The following options are available with xtgee but are not shown in the dialog box:

nodisplay is for programmers. It suppresses display of the header and coefficients.

coeflegend; see [R] **Estimation options**.

Remarks and examples

For a thorough introduction to GEE in the estimation of GLM, see Hardin and Hilbe (2013). More information on linear models is presented in Nelder and Wedderburn (1972). Finally, there have been several illuminating articles on various applications of GEE in Zeger, Liang, and Albert (1988); Zeger and Liang (1986), and Liang (1987). Pendergast et al. (1996) surveys the current methods for analyzing clustered data in regard to binary response data. Our implementation follows that of Liang and Zeger (1986).

xtgee fits generalized linear models of y_{it} with covariates $\mathbf{x}_{it}$

$$g\{E(y_{it})\} = \mathbf{x}_{it}\boldsymbol{\beta}, \qquad y \sim F \text{ with parameters } \theta_{it}$$

for $i = 1, \ldots, m$ and $t = 1, \ldots, n_i$, where there are n_i observations for each group identifier i. $g(\,)$ is called the link function, and F is the distributional family. Substituting various definitions for $g(\,)$ and F results in a wide array of models. For instance, if y_{it} is distributed Gaussian (normal) and $g(\,)$ is the identity function, we have

$$E(y_{it}) = \mathbf{x}_{it}\boldsymbol{\beta}, \qquad y \sim N(\,)$$

yielding linear regression, random-effects regression, or other regression-related models, depending on what we assume for the correlation structure.

If $g(\cdot)$ is the logit function and y_{it} is distributed Bernoulli (binomial), we have

$$\text{logit}\{E(y_{it})\} = \mathbf{x}_{it}\boldsymbol{\beta}, \qquad y \sim \text{Bernoulli}$$

or logistic regression. If $g(\cdot)$ is the natural log function and y_{it} is distributed Poisson, we have

$$\ln\{E(y_{it})\} = \mathbf{x}_{it}\boldsymbol{\beta}, \qquad y \sim \text{Poisson}$$

or Poisson regression, also known as the log-linear model. Other combinations are possible.

You specify the link function with the `link()` option, the distributional family with `family()`, and the assumed within-group correlation structure with `corr()`.

The binomial distribution can be specified as case 1 `family(binomial)`, case 2 `family(binomial #)`, or case 3 `family(binomial` *varname*`)`. In case 2, # is the value of the binomial denominator N, the number of trials. Specifying `family(binomial 1)` is the same as specifying `family(binomial)`; both mean that y has the Bernoulli distribution with values 0 and 1 only. In case 3, *varname* is the variable containing the binomial denominator, thus allowing the number of trials to vary across observations.

The negative binomial distribution must be specified as `family(nbinomial #)`, where # denotes the value of the parameter α in the negative binomial distribution. The results will be conditional on this value.

You do not have to specify both `family()` and `link()`; the default `link()` is the canonical link for the specified `family()` (excluding `family(nbinomial)`):

Family	Default link
`family(binomial)`	`link(logit)`
`family(gamma)`	`link(reciprocal)`
`family(gaussian)`	`link(identity)`
`family(igaussian)`	`link(power -2)`
`family(nbinomial)`	`link(log)`
`family(poisson)`	`link(log)`

The canonical link for the negative binomial family is obtained by specifying `link(nbinomial)`. If you specify both `family()` and `link()`, not all combinations make sense. You may choose among the following combinations:

	Gaussian	Inverse Gaussian	Binomial	Poisson	Negative Binomial	Gamma
Identity	x	x	x	x	x	x
Log	x	x	x	x	x	x
Logit			x			
Probit			x			
C. log–log			x			
Power	x	x	x	x	x	x
Odds Power			x			
Neg. binom.					x	
Reciprocal	x		x	x		x

You specify the assumed within-group correlation structure with the `corr()` option.

For example, call $\mathbf{R}$ the working correlation matrix for modeling the within-group correlation, a square $\max\{n_i\} \times \max\{n_i\}$ matrix. `corr()` specifies the structure of $\mathbf{R}$. Let $\mathbf{R}_{t,s}$ denote the t,s element.

The `independent` structure is defined as

$$\mathbf{R}_{t,s} = \begin{cases} 1 & \text{if } t = s \\ 0 & \text{otherwise} \end{cases}$$

The `corr(exchangeable)` structure (corresponding to equal-correlation models) is defined as

$$\mathbf{R}_{t,s} = \begin{cases} 1 & \text{if } t = s \\ \rho & \text{otherwise} \end{cases}$$

The `corr(ar` g`)` structure is defined as the usual correlation matrix for an AR(g) model. This is sometimes called multiplicative correlation. For example, an AR(1) model is given by

$$\mathbf{R}_{t,s} = \begin{cases} 1 & \text{if } t = s \\ \rho^{|t-s|} & \text{otherwise} \end{cases}$$

The `corr(stationary` g`)` structure is a stationary(g) model. For example, a stationary(1) model is given by

$$\mathbf{R}_{t,s} = \begin{cases} 1 & \text{if } t = s \\ \rho & \text{if } |t-s| = 1 \\ 0 & \text{otherwise} \end{cases}$$

The `corr(nonstationary` g`)` structure is a nonstationary(g) model that imposes only the constraints that the elements of the working correlation matrix along the diagonal be 1 and the elements outside the gth band be zero,

$$\mathbf{R}_{t,s} = \begin{cases} 1 & \text{if } t = s \\ \rho_{ts} & \text{if } 0 < |t-s| \leq g, \, \rho_{ts} = \rho_{st} \\ 0 & \text{otherwise} \end{cases}$$

`corr(unstructured)` imposes only the constraint that the diagonal elements of the working correlation matrix be 1.

$$\mathbf{R}_{t,s} = \begin{cases} 1 & \text{if } t = s \\ \rho_{ts} & \text{otherwise, } \rho_{ts} = \rho_{st} \end{cases}$$

The `corr(fixed` *matname*`)` specification is taken from the user-supplied matrix, such that

$$\mathbf{R} = \mathit{matname}$$

Here the correlations are not estimated from the data. The user-supplied matrix must be a valid correlation matrix with 1s on the diagonal.

Full formulas for all the correlation structures are provided in the *Methods and formulas* below.

❑ Technical note

Some `family()`, `link()`, and `corr()` combinations result in models already fit by Stata:

`family()`	`link()`	`corr()`	Other Stata estimation command
`gaussian`	`identity`	`independent`	`regress`
`gaussian`	`identity`	`exchangeable`	`xtreg, re`
`gaussian`	`identity`	`exchangeable`	`xtreg, pa`
`binomial`	`cloglog`	`independent`	`cloglog` (see note 1)
`binomial`	`cloglog`	`exchangeable`	`xtcloglog, pa`
`binomial`	`logit`	`independent`	`logit` or `logistic`
`binomial`	`logit`	`exchangeable`	`xtlogit, pa`
`binomial`	`probit`	`independent`	`probit` (see note 2)
`binomial`	`probit`	`exchangeable`	`xtprobit, pa`
`nbinomial`	`log`	`independent`	`nbreg` (see note 3)
`poisson`	`log`	`independent`	`poisson`
`poisson`	`log`	`exchangeable`	`xtpoisson, pa`
`gamma`	`log`	`independent`	`streg, dist(exp) nohr` (see note 4)
family	*link*	`independent`	`glm, irls` (see note 5)

Notes:

1. For cloglog estimation, `xtgee` with `corr(independent)` and `cloglog` (see [R] **cloglog**) will produce the same coefficients, but the standard errors will be only asymptotically equivalent because cloglog is not the canonical link for the binomial family.

2. For probit estimation, `xtgee` with `corr(independent)` and `probit` will produce the same coefficients, but the standard errors will be only asymptotically equivalent because probit is not the canonical link for the binomial family. If the binomial denominator is not 1, the equivalent maximum-likelihood command is `glm` with options `family(binomial` #`)` or `family(binomial` *varname*`)` and `link(probit)`; see [R] **probit** and [R] **glm**.

3. Fitting a negative binomial model by using `xtgee` (or using `glm`) will yield results conditional on the specified value of α. The `nbreg` command, however, estimates that parameter and provides unconditional estimates; see [R] **nbreg**.

4. `xtgee` with `corr(independent)` can be used to fit exponential regressions, but this requires specifying `scale(1)`. As with probit, the `xtgee`-reported standard errors will be only asymptotically equivalent to those produced by `streg, dist(exp) nohr` (see [ST] **streg**) because log is not the canonical link for the gamma family. `xtgee` cannot be used to fit exponential regressions on censored data.

   Using the `independent` correlation structure, the `xtgee` command will fit the same model fit with the `glm, irls` command if the family–link combination is the same.

5. If the `xtgee` command is equivalent to another command, using `corr(independent)` and the `vce(robust)` option with `xtgee` corresponds to using the `vce(cluster` *clustvar*`)` option in the equivalent command, where *clustvar* corresponds to the panel variable.

❑

`xtgee` is a generalization of the `glm, irls` command and gives the same output when the same family and link are specified together with an independent correlation structure. What makes `xtgee` useful is

- the number of statistical models that it generalizes for use with panel data, many of which are not otherwise available in Stata;
- the richer correlation structure `xtgee` allows, even when models are available through other xt commands; and

- the availability of robust standard errors (see [U] **20.22 Obtaining robust variance estimates**), even when the model and correlation structure are available through other xt commands.

In the following examples, we illustrate the relationships of xtgee with other Stata estimation commands. Remember that, although xtgee generalizes many other commands, the computational algorithm is different; therefore, the answers you obtain will not be identical. The dataset we are using is a subset of the nlswork data (see [XT] **xt**); we are looking at observations before 1980.

▷ Example 1

We can use xtgee to perform ordinary least squares by regress:

```
. use https://www.stata-press.com/data/r19/nlswork2
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)
. regress ln_w grade age c.age#c.age

      Source |       SS           df       MS      Number of obs   =    16,085
-------------+----------------------------------   F(3, 16081)     =   1413.68
       Model |   597.54468         3  199.18156   Prob > F        =    0.0000
    Residual |  2265.74584    16,081  .14089583   R-squared       =    0.2087
-------------+----------------------------------   Adj R-squared   =    0.2085
       Total |  2863.29052    16,084  .178021047  Root MSE        =    .37536

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
       grade |   .0724483   .0014229    50.91   0.000     .0696592    .0752374
         age |   .1064874   .0083644    12.73   0.000     .0900922    .1228825
             |
 c.age#c.age |  -.0016931   .0001655   -10.23   0.000    -.0020174   -.0013688
             |
       _cons |  -.8681487   .1024896    -8.47   0.000     -1.06904   -.6672577
------------------------------------------------------------------------------

. xtgee ln_w grade age c.age#c.age, corr(indep) nmp
Iteration 1:  Tolerance = 8.684e-13

GEE population-averaged model                   Number of obs     =   16,085
Group variable: idcode                          Number of groups  =    3,913
Family: Gaussian                                Obs per group:
Link:   Identity                                              min =        1
Correlation: independent                                      avg =      4.1
                                                              max =        9
                                                Wald chi2(3)      =  4241.04
Scale parameter = .1408958                      Prob > chi2       =   0.0000

Pearson chi2(16081)   =   2265.75               Deviance          =  2265.75
Dispersion (Pearson) = .1408958                 Dispersion        = .1408958

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
       grade |   .0724483   .0014229    50.91   0.000     .0696594    .0752372
         age |   .1064874   .0083644    12.73   0.000     .0900935    .1228812
             |
 c.age#c.age |  -.0016931   .0001655   -10.23   0.000    -.0020174   -.0013688
             |
       _cons |  -.8681487   .1024896    -8.47   0.000    -1.069025   -.6672728
------------------------------------------------------------------------------
```

When nmp is specified, the coefficients and the standard errors produced by the estimators are the same. Moreover, the scale parameter estimate from the xtgee command equals the MSE calculation from regress; both are estimates of the variance of the residuals.

◁

▷ Example 2

The identity link and Gaussian family produce regression-type models. With the independent correlation structure, we reproduce ordinary least squares. With the exchangeable correlation structure, we produce an equal-correlation linear regression estimator.

xtgee, fam(gauss) link(ident) corr(exch) is asymptotically equivalent to the weighted-GLS estimator provided by xtreg, re and to the full maximum-likelihood estimator provided by xtreg, mle. In balanced data, xtgee, fam(gauss) link(ident) corr(exch) and xtreg, mle produce the same results. With unbalanced data, the results are close but differ because the two estimators handle unbalanced data differently. For both balanced and unbalanced data, the results produced by xtgee, fam(gauss) link(ident) corr(exch) and xtreg, mle differ from those produced by xtreg, re. Below we demonstrate the use of the three estimators with unbalanced data. We begin with xtgee; show the maximum likelihood estimator xtreg, mle; show the GLS estimator xtreg, re; and finally show xtgee with the vce(robust) option.

```
. xtgee ln_w grade age c.age#c.age, nolog

GEE population-averaged model                     Number of obs     =  16,085
Group variable: idcode                            Number of groups  =   3,913
Family: Gaussian                                  Obs per group:
Link:   Identity                                                min =       1
Correlation: exchangeable                                       avg =     4.1
                                                                max =       9
                                                  Wald chi2(3)      = 2918.26
Scale parameter = .1416586                        Prob > chi2       =  0.0000

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
       grade |   .0717731     .00211    34.02   0.000     .0676377    .0759086
         age |   .1077645    .006885    15.65   0.000     .0942701    .1212589
             |
 c.age#c.age |  -.0016381   .0001362   -12.03   0.000     -.001905   -.0013712
             |
       _cons |  -.9480449   .0869277   -10.91   0.000     -1.11842   -.7776698
------------------------------------------------------------------------------
```

```
. xtreg ln_w grade age c.age#c.age, mle
Fitting constant-only model:
Iteration 0:  Log likelihood = -5868.3483
Iteration 1:  Log likelihood = -5858.8833
Iteration 2:  Log likelihood = -5858.8244
Fitting full model:
Iteration 0:  Log likelihood = -4591.9241
Iteration 1:  Log likelihood = -4562.4406
Iteration 2:  Log likelihood = -4562.3526
Iteration 3:  Log likelihood = -4562.3525
Random-effects ML regression                    Number of obs     =  16,085
Group variable: idcode                          Number of groups =   3,913
Random effects u_i ~ Gaussian                   Obs per group:
                                                             min =       1
                                                             avg =     4.1
                                                             max =       9
                                                LR chi2(3)        = 2592.94
Log likelihood = -4562.3525                     Prob > chi2       =  0.0000

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
       grade |   .0717747    .002142    33.51   0.000     .0675765     .075973
         age |   .1077899   .0068266    15.79   0.000     .0944101    .1211697
             |
 c.age#c.age |  -.0016364    .000135   -12.12   0.000    -.0019011   -.0013718
             |
       _cons |  -.9500833    .086384   -11.00   0.000    -1.119393   -.7807737
-------------+----------------------------------------------------------------
    /sigma_u |   .2689639   .0040854                      .2610748    .2770915
    /sigma_e |   .2669944   .0017113                      .2636613    .2703696
         rho |   .5036748   .0086449                      .4867329      .52061
------------------------------------------------------------------------------
LR test of sigma_u=0: chibar2(01) = 4996.22           Prob >= chibar2 = 0.000
```

```
. xtreg ln_w grade age c.age#c.age, re

Random-effects GLS regression                   Number of obs     =     16,085
Group variable: idcode                          Number of groups  =      3,913

R-squared:                                      Obs per group:
     Within  = 0.0983                                         min =          1
     Between = 0.2946                                         avg =        4.1
     Overall = 0.2076                                         max =          9

                                                Wald chi2(3)      =    2875.02
corr(u_i, X) = 0 (assumed)                      Prob > chi2       =     0.0000

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
       grade |   .0717757   .0021666    33.13   0.000     .0675294    .0760221
         age |   .1078042   .0068125    15.82   0.000     .0944519    .1211566
             |
 c.age#c.age |  -.0016355   .0001347   -12.14   0.000    -.0018996   -.0013714
             |
       _cons |  -.9512118   .0863139   -11.02   0.000    -1.120384   -.7820397
-------------+----------------------------------------------------------------
     sigma_u |  .27383747
     sigma_e |  .26624266
         rho |  .51405959   (fraction of variance due to u_i)
------------------------------------------------------------------------------

. xtgee ln_w grade age c.age#c.age, vce(robust) nolog

GEE population-averaged model                       Number of obs    = 16,085
Group variable: idcode                              Number of groups =  3,913
Family: Gaussian                                    Obs per group:
Link:   Identity                                                 min =      1
Correlation: exchangeable                                        avg =    4.1
                                                                 max =      9
                                                    Wald chi2(3)     = 2031.28
Scale parameter = .1416586                          Prob > chi2      =  0.0000

                             (Std. err. adjusted for clustering on idcode)
------------------------------------------------------------------------------
             |               Robust
     ln_wage | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
       grade |   .0717731   .0023341    30.75   0.000     .0671983    .0763479
         age |   .1077645   .0098097    10.99   0.000     .0885379    .1269911
             |
 c.age#c.age |  -.0016381   .0001964    -8.34   0.000     -.002023   -.0012532
             |
       _cons |  -.9480449   .1195009    -7.93   0.000    -1.182262   -.7138274
------------------------------------------------------------------------------
```

In [R] **regress**, `regress, vce(cluster` *clustvar*`)` may produce inefficient coefficient estimates with valid standard errors for random-effects models. These standard errors are robust to model misspecification. The `vce(robust)` option of `xtgee`, on the other hand, requires that the model correctly specify the mean and the link function when the noncanonical link is used.

◁

Stored results

xtgee stores the following in e():

Scalars

e(N)	number of observations
e(N_g)	number of groups
e(df_m)	model degrees of freedom
e(chi2)	χ^2
e(p)	p-value for model test
e(df_pear)	degrees of freedom for Pearson χ^2
e(chi2_dev)	χ^2 test of deviance
e(chi2_dis)	χ^2 test of deviance dispersion
e(deviance)	deviance
e(dispers)	deviance dispersion
e(phi)	scale parameter
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(tol)	target tolerance
e(dif)	achieved tolerance
e(rank)	rank of e(V)
e(rc)	return code

Macros

e(cmd)	xtgee
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(ivar)	variable denoting groups
e(tvar)	variable denoting time within groups
e(model)	pa
e(family)	distribution family
e(link)	link function
e(corr)	correlation structure
e(scale)	x2, dev, phi, or #; scale parameter
e(wtype)	weight type
e(wexp)	weight expression
e(offset)	linear offset variable
e(chi2type)	Wald; type of model χ^2 test
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(nmp)	nmp, if specified
e(properties)	b V
e(estat_cmd)	program used to implement estat
e(predict)	program used to implement predict
e(marginsnotok)	predictions disallowed by margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved

Matrices

e(b)	coefficient vector
e(R)	estimated working correlation matrix
e(V)	variance–covariance matrix of the estimators
e(V_modelbased)	model-based variance

Functions

e(sample)	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices
`r(table)` matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

Methods and formulas are presented under the following headings:

Introduction

`xtgee` fits generalized linear models for panel data with the GEE approach described in Liang and Zeger (1986). A related method, referred to as GEE2, is described in Zhao and Prentice (1990) and Prentice and Zhao (1991). The GEE2 method attempts to gain efficiency in the estimation of $\boldsymbol{\beta}$ by specifying a parametric model for $\boldsymbol{\alpha}$ and then assumes that the models for both the mean and dependency parameters are correct. Thus there is a tradeoff in robustness for efficiency. The preliminary work of Liang, Zeger, and Qaqish (1992), however, indicates that there is little efficiency gained with this alternative approach.

In the GLM approach (see McCullagh and Nelder [1989]), we assume that

$$
\begin{aligned}
h(\boldsymbol{\mu}_{i,j}) &= x_{i,j}^{\mathrm{T}}\boldsymbol{\beta} \\
\mathrm{Var}(y_{i,j}) &= g(\mu_{i,j})\phi \\
\boldsymbol{\mu}_i = E(\mathbf{y}_i) &= \{h^{-1}(x_{i,1}^{\mathrm{T}}\boldsymbol{\beta}), \dots, h^{-1}(x_{i,n_i}^{\mathrm{T}}\boldsymbol{\beta})\}^{\mathrm{T}} \\
\mathbf{A}_i &= \mathrm{diag}\{g(\mu_{i,1}), \dots, g(\mu_{i,n_i})\} \\
\mathrm{Cov}(\mathbf{y}_i) &= \phi\mathbf{A}_i \quad \text{for independent observations.}
\end{aligned}
$$

In the absence of a convenient likelihood function with which to work, we can rely on a multivariate analog of the quasiscore function introduced by Wedderburn (1974):

$$
\mathbf{S}_{\boldsymbol{\beta}}(\boldsymbol{\beta},\boldsymbol{\alpha}) = \sum_{i=1}^{m}\left(\frac{\partial\boldsymbol{\mu}_i}{\partial\boldsymbol{\beta}}\right)^{\mathrm{T}}\mathrm{Var}(\mathbf{y}_i)^{-1}(\mathbf{y}_i-\boldsymbol{\mu}_i) = 0
$$

We can solve for correlation parameters $\boldsymbol{\alpha}$ by simultaneously solving

$$
\mathbf{S}_{\boldsymbol{\alpha}}(\boldsymbol{\beta},\boldsymbol{\alpha}) = \sum_{i=1}^{m}\left(\frac{\partial\boldsymbol{\eta}_i}{\partial\boldsymbol{\alpha}}\right)^{\mathrm{T}}\mathbf{H}_i^{-1}(\mathbf{W}_i-\boldsymbol{\eta}_i) = 0
$$

In the GEE approach to GLM, we let $\mathbf{R}_i(\boldsymbol{\alpha})$ be a "working" correlation matrix depending on the parameters in $\boldsymbol{\alpha}$ (see the *Correlation structures* section for the number of parameters), and we estimate $\boldsymbol{\beta}$ by solving the GEE,

$$\mathbf{U}(\boldsymbol{\beta}) = \sum_{i=1}^{m} \left(\frac{\partial \boldsymbol{\mu}_i}{\partial \beta}\right)^{\mathrm{T}} \mathbf{V}_i^{-1}(\boldsymbol{\alpha})(\mathbf{y}_i - \boldsymbol{\mu}_i) = 0$$
$$\text{where } \mathbf{V}_i(\boldsymbol{\alpha}) = \mathbf{A}_i^{1/2}\mathbf{R}_i(\boldsymbol{\alpha})\mathbf{A}_i^{1/2}$$

To solve this equation, we need only a crude approximation of the variance matrix, which we can obtain from a Taylor series expansion, where

$$\begin{aligned}\mathrm{Cov}(\mathbf{y}_i) &= \mathbf{L}_i\mathbf{Z}_i\mathbf{D}_i\mathbf{Z}_i^{\mathrm{T}}\mathbf{L}_i + \phi\mathbf{A}_i = \widetilde{\mathbf{V}}_i \\ \mathbf{L}_i &= \mathrm{diag}\{\partial h^{-1}(u)/\partial u, u = x_{i,j}^{\mathrm{T}}\boldsymbol{\beta}, j = 1,\dots,n_i\}\end{aligned}$$

which allows that

$$\widehat{\mathbf{D}}_i \approx (\mathbf{Z}_i^{\mathrm{T}}\mathbf{Z}_i)^{-1}\mathbf{Z}_i\widehat{\mathbf{L}}_i^{-1}\left\{(\mathbf{y}_i - \widehat{\boldsymbol{\mu}}_i)(\mathbf{y}_i - \widehat{\boldsymbol{\mu}}_i)^{\mathrm{T}} - \widehat{\phi}\widehat{\mathbf{A}}_i\right\}\widehat{\mathbf{L}}_i^{-1}\mathbf{Z}_i^{\mathrm{T}}(\mathbf{Z}_i'\mathbf{Z}_i)^{-1}$$
$$\hat{\phi} = \sum_{i=1}^{m}\sum_{j=1}^{n_i} \frac{(y_{i,j} - \hat{\mu}_{i,j})^2 - (\widehat{\mathbf{L}}_{i,j})^2\mathbf{Z}_{i,j}^{\mathrm{T}}\widehat{\mathbf{D}}_i\mathbf{Z}_{i,j}}{g(\hat{\mu}_{i,j})}$$

Calculating GEE for GLM

Using the notation from Liang and Zeger (1986), let $\mathbf{y}_i = (y_{i,1},\dots,y_{i,n_i})^{\mathrm{T}}$ be the $n_i \times 1$ vector of outcome values, and let $\mathbf{X}_i = (x_{i,1},\dots,x_{i,n_i})^{\mathrm{T}}$ be the $n_i \times p$ matrix of covariate values for the ith subject $i = 1,\dots,m$. We assume that the marginal density for $y_{i,j}$ may be written in exponential family notation as

$$f(y_{i,j}) = \exp\left[\left\{y_{i,j}\theta_{i,j} - a(\theta_{i,j}) + b(y_{i,j})\right\}\phi\right]$$

where $\theta_{i,j} = h(\eta_{i,j}), \eta_{i,j} = x_{i,j}\boldsymbol{\beta}$. Under this formulation, the first two moments are given by

$$E(y_{i,j}) = a'(\theta_{i,j}), \qquad \mathrm{Var}(y_{i,j}) = a''(\theta_{i,j})/\phi$$

In what follows, we let $n_i = n$ without loss of generality. We define the quantities, assuming that we have an $n \times n$ working correlation matrix $\mathbf{R}(\boldsymbol{\alpha})$,

$$\begin{aligned}
\boldsymbol{\Delta}_i &= \mathrm{diag}(d\theta_{i,j}/d\eta_{i,j}) && n \times n \text{ matrix}\\
\mathbf{A}_i &= \mathrm{diag}\{a''(\theta_{i,j})\} && n \times n \text{ matrix}\\
\mathbf{S}_i &= \mathbf{y}_i - a'(\boldsymbol{\theta}_i) && n \times 1 \text{ matrix}\\
\mathbf{D}_i &= \mathbf{A}_i\boldsymbol{\Delta}_i\mathbf{X}_i && n \times p \text{ matrix}\\
\mathbf{V}_i &= \mathbf{A}_i^{1/2}\mathbf{R}(\boldsymbol{\alpha})\mathbf{A}_i^{1/2} && n \times n \text{ matrix}
\end{aligned}$$

such that the GEE becomes

$$\sum_{i=1}^{m} \mathbf{D}_i^{\mathrm{T}}\mathbf{V}_i^{-1}\mathbf{S}_i = 0$$

We then have that

$$\widehat{\boldsymbol{\beta}}_{j+1} = \widehat{\boldsymbol{\beta}}_j - \left\{\sum_{i=1}^{m} \mathbf{D}_i^{\mathrm{T}}(\widehat{\boldsymbol{\beta}}_j)\widetilde{\mathbf{V}}_i^{-1}(\widehat{\boldsymbol{\beta}}_j)\mathbf{D}_i(\widehat{\boldsymbol{\beta}}_j)\right\}^{-1}\left\{\sum_{i=1}^{m} \mathbf{D}_i^{\mathrm{T}}(\widehat{\boldsymbol{\beta}}_j)\widetilde{\mathbf{V}}_i^{-1}(\widehat{\boldsymbol{\beta}}_j)\mathbf{S}_i(\widehat{\boldsymbol{\beta}}_j)\right\}$$

where the term

$$\left\{\sum_{i=1}^{m} \mathbf{D}_i^{\mathrm{T}}(\widehat{\boldsymbol{\beta}}_j)\widetilde{\mathbf{V}}_i^{-1}(\widehat{\boldsymbol{\beta}}_j)\mathbf{D}_i(\widehat{\boldsymbol{\beta}}_j)\right\}^{-1}$$

is what we call the conventional variance estimate. It is used to calculate the standard errors if the `vce(robust)` option is not specified. This command supports the clustered version of the Huber/White/sandwich estimator of the variance with panels treated as clusters when `vce(robust)` is specified. See [P] **_robust**, particularly *Maximum likelihood estimators* and *Methods and formulas*. Liang and Zeger (1986) also discuss the calculation of the robust variance estimator.

Define the following:

$$\begin{aligned}
\mathbf{D} &= (\mathbf{D}_1^{\mathrm{T}}, \ldots, \mathbf{D}_m^{\mathrm{T}}) \\
\mathbf{S} &= (\mathbf{S}_1^{\mathrm{T}}, \ldots, \mathbf{S}_m^{\mathrm{T}})^{\mathrm{T}} \\
\widetilde{\mathbf{V}} &= nm \times nm \text{ block diagonal matrix with } \widetilde{\mathbf{V}}_i \\
\mathbf{Z} &= \mathbf{D}\boldsymbol{\beta} - \mathbf{S}
\end{aligned}$$

At a given iteration, the correlation parameters $\boldsymbol{\alpha}$ and scale parameter ϕ can be estimated from the current Pearson residuals, defined by

$$\widehat{r}_{i,j} = \{y_{i,j} - a'(\widehat{\theta}_{i,j})\}/\{a''(\widehat{\theta}_{i,j})\}^{1/2}$$

where $\widehat{\theta}_{i,j}$ depends on the current value for $\widehat{\boldsymbol{\beta}}$. We can then estimate ϕ by

$$\widehat{\phi}^{-1} = \sum_{i=1}^{m}\sum_{j=1}^{n_i} \widehat{r}_{i,j}^{\,2}/(N-p)$$

As this general derivation is complicated, let's follow the derivation of the Gaussian family with the identity link (regression) to illustrate the generalization. After making appropriate substitutions, we will see a familiar updating equation. First, we rewrite the updating equation for $\boldsymbol{\beta}$ as

$$\widehat{\boldsymbol{\beta}}_{j+1} = \widehat{\boldsymbol{\beta}}_j - \mathbf{Z}_1^{-1}\mathbf{Z}_2$$

and then derive $\mathbf{Z}_1$ and $\mathbf{Z}_2$.

$$\begin{aligned}
\mathbf{Z}_1 &= \sum_{i=1}^{m} \mathbf{D}_i^{\mathrm{T}}(\widehat{\boldsymbol{\beta}}_j)\widetilde{\mathbf{V}}_i^{-1}(\widehat{\boldsymbol{\beta}}_j)\mathbf{D}_i(\widehat{\boldsymbol{\beta}}_j) = \sum_{i=1}^{m} \mathbf{X}_i^{\mathrm{T}}\boldsymbol{\Delta}_i^{\mathrm{T}}\mathbf{A}_i^{\mathrm{T}}\{\mathbf{A}_i^{1/2}\mathbf{R}(\boldsymbol{\alpha})\mathbf{A}_i^{1/2}\}^{-1}\mathbf{A}_i\boldsymbol{\Delta}_i\mathbf{X}_i \\
&= \sum_{i=1}^{m} \mathbf{X}_i^{\mathrm{T}} \operatorname{diag}\left\{\frac{\partial\theta_{i,j}}{\partial(\mathbf{X}\boldsymbol{\beta})}\right\} \operatorname{diag}\left\{a''(\theta_{i,j})\right\} \left[\operatorname{diag}\left\{a''(\theta_{i,j})\right\}^{1/2} \mathbf{R}(\boldsymbol{\alpha}) \operatorname{diag}\left\{a''(\theta_{i,j})\right\}^{1/2}\right]^{-1} \\
&\qquad \operatorname{diag}\left\{a''(\theta_{i,j})\right\} \operatorname{diag}\left\{\frac{\partial\theta_{i,j}}{\partial(\mathbf{X}\boldsymbol{\beta})}\right\}\mathbf{X}_i \\
&= \sum_{i=1}^{m} \mathbf{X}_i^{\mathrm{T}}\mathbf{II}(\mathbf{III})^{-1}\mathbf{II}\mathbf{X}_i = \sum_{i=1}^{m} \mathbf{X}_i^{\mathrm{T}}\mathbf{X}_i = \mathbf{X}^{\mathrm{T}}\mathbf{X}
\end{aligned}$$

$$
\begin{aligned}
\mathbf{Z}_2 &= \sum_{i=1}^{m} \mathbf{D}_i^{\mathrm{T}}(\widehat{\boldsymbol{\beta}}_j)\widetilde{\mathbf{V}}_i^{-1}(\widehat{\boldsymbol{\beta}}_j)\mathbf{S}_i(\widehat{\boldsymbol{\beta}}_j) = \sum_{i=1}^{m} \mathbf{X}_i^{\mathrm{T}}\boldsymbol{\Delta}_i^{\mathrm{T}}\mathbf{A}_i^{\mathrm{T}}\{\mathbf{A}_i^{1/2}\mathbf{R}(\boldsymbol{\alpha})\mathbf{A}_i^{1/2}\}^{-1}\left(\mathbf{y}_i - \mathbf{X}_i\widehat{\boldsymbol{\beta}}_j\right) \\
&= \sum_{i=1}^{m} \mathbf{X}_i^{\mathrm{T}} \operatorname{diag}\left\{\frac{\partial \theta_{i,j}}{\partial(\mathbf{X}\boldsymbol{\beta})}\right\} \operatorname{diag}\left\{a''(\theta_{i,j})\right\} \left[\operatorname{diag}\left\{a''(\theta_{i,j})\right\}^{1/2} \mathbf{R}(\boldsymbol{\alpha}) \operatorname{diag}\left\{a''(\theta_{i,j})\right\}^{1/2}\right]^{-1} \\
&\qquad \left(\mathbf{y}_i - \mathbf{X}_i\widehat{\boldsymbol{\beta}}_j\right) \\
&= \sum_{i=1}^{m} \mathbf{X}_i^{\mathrm{T}}\mathbf{II}(\mathbf{III})^{-1}(\mathbf{y}_i - \mathbf{X}_i\widehat{\boldsymbol{\beta}}_j) = \sum_{i=1}^{m} \mathbf{X}_i^{\mathrm{T}}(\mathbf{y}_i - \mathbf{X}_i\widehat{\boldsymbol{\beta}}_j) = \mathbf{X}^{\mathrm{T}}\widehat{s}_j
\end{aligned}
$$

So, we may write the update formula as

$$
\widehat{\boldsymbol{\beta}}_{j+1} = \widehat{\boldsymbol{\beta}}_j - (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\widehat{s}_j
$$

which is the same formula for GLS in regression.

Correlation structures

The working correlation matrix $\mathbf{R}$ is a function of $\boldsymbol{\alpha}$ and is more accurately written as $\mathbf{R}(\boldsymbol{\alpha})$. Depending on the assumed correlation structure, $\boldsymbol{\alpha}$ might be

Independent	no parameters to estimate
Exchangeable	$\boldsymbol{\alpha}$ is a scalar
Autoregressive	$\boldsymbol{\alpha}$ is a vector
Stationary	$\boldsymbol{\alpha}$ is a vector
Nonstationary	$\boldsymbol{\alpha}$ is a matrix
Unstructured	$\boldsymbol{\alpha}$ is a matrix

Also, throughout the estimation of a general unbalanced panel, it is more proper to discuss $\mathbf{R}_i$, which is the upper left $n_i \times n_i$ submatrix of the ultimately stored matrix in `e(R)`, $\max\{n_i\} \times \max\{n_i\}$.

The only panels that enter into the estimation for a lag-dependent correlation structure are those with $n_i > g$ (assuming a lag of g). `xtgee` drops panels with too few observations (and mentions when it does so).

Independent

The working correlation matrix $\mathbf{R}$ is an identity matrix.

Exchangeable

$$
\boldsymbol{\alpha} = \frac{\sum_{i=1}^{m}\left(\sum_{j=1}^{n_i}\sum_{k=1}^{n_i}\widehat{r}_{i,j}\widehat{r}_{i,k} - \sum_{j=1}^{n_i}\widehat{r}_{i,j}^{\,2}\right)}{\sum_{i=1}^{m}\{n_i(n_i-1)\}} \Bigg/ \frac{\sum_{i=1}^{m}\left(\sum_{j=1}^{n_i}\widehat{r}_{i,j}^{\,2}\right)}{\sum_{i=1}^{m} n_i}
$$

and the working correlation matrix is given by

$$
\mathbf{R}_{s,t} = \begin{cases} 1 & s = t \\ \alpha & \text{otherwise} \end{cases}
$$

Autoregressive and stationary

These two structures require g parameters to be estimated so that $\boldsymbol{\alpha}$ is a vector of length $g+1$ (the first element of $\boldsymbol{\alpha}$ is 1).

$$\boldsymbol{\alpha} = \sum_{i=1}^{m} \left(\frac{\sum_{j=1}^{n_i} \widehat{r}_{i,j}^{\,2}}{n_i} \;,\; \frac{\sum_{j=1}^{n_i-1} \widehat{r}_{i,j}\widehat{r}_{i,j+1}}{n_i} \;,\; \dots \;,\; \frac{\sum_{j=1}^{n_i-g} \widehat{r}_{i,j}\widehat{r}_{i,j+g}}{n_i} \right) \Bigg/ \left(\sum_{i=1}^{m} \frac{\sum_{j=1}^{n_i} \widehat{r}_{i,j}^{\,2}}{n_i} \right)$$

The working correlation matrix for the AR model is calculated as a function of Toeplitz matrices formed from the $\boldsymbol{\alpha}$ vector; see Newton (1988). The working correlation matrix for the stationary model is given by

$$\mathbf{R}_{s,t} = \begin{cases} \boldsymbol{\alpha}_{1,|s-t|} & \text{if } |s-t| \le g \\ 0 & \text{otherwise} \end{cases}$$

Nonstationary and unstructured

These two correlation structures require a matrix of parameters. $\boldsymbol{\alpha}$ is estimated (where we replace $\widehat{r}_{i,j} = 0$ whenever $i > n_i$ or $j > n_i$) as

$$\boldsymbol{\alpha} = \sum_{i=1}^{m} m \begin{pmatrix} N_{1,1}^{-1}\widehat{r}_{i,1}^{\,2} & N_{1,2}^{-1}\widehat{r}_{i,1}\widehat{r}_{i,2} & \cdots & N_{1,n}^{-1}\widehat{r}_{i,1}\widehat{r}_{i,n} \\ N_{2,1}^{-1}\widehat{r}_{i,2}\widehat{r}_{i,1} & N_{2,2}^{-1}\widehat{r}_{i,2}^{\,2} & \cdots & N_{2,n}^{-1}\widehat{r}_{i,2}\widehat{r}_{i,n} \\ \vdots & \vdots & \ddots & \vdots \\ N_{n,1}^{-1}\widehat{r}_{i,n_i}\widehat{r}_{i,1} & N_{n,2}^{-1}\widehat{r}_{i,n_i}\widehat{r}_{i,2} & \cdots & N_{n,n}^{-1}\widehat{r}_{i,n}^{\,2} \end{pmatrix} \Bigg/ \left(\sum_{i=1}^{m} \frac{\sum_{j=1}^{n_i} \widehat{r}_{i,j}^{\,2}}{n_i} \right)$$

where $N_{p,q} = \sum_{i=1}^{m} I(i,p,q)$ and

$$I(i,p,q) = \begin{cases} 1 & \text{if panel } i \text{ has valid observations at times } p \text{ and } q \\ 0 & \text{otherwise} \end{cases}$$

where $N_{i,j} = \min(N_i, N_j)$, $N_i =$ number of panels observed at time i, and $n = \max(n_1, n_2, \dots, n_m)$.

The working correlation matrix for the nonstationary model is given by

$$\mathbf{R}_{s,t} = \begin{cases} 1 & \text{if } s = t \\ \boldsymbol{\alpha}_{s,t} & \text{if } 0 < |s-t| \le g \\ 0 & \text{otherwise} \end{cases}$$

The working correlation matrix for the unstructured model is given by

$$\mathbf{R}_{s,t} = \begin{cases} 1 & \text{if } s = t \\ \boldsymbol{\alpha}_{s,t} & \text{otherwise} \end{cases}$$

such that the unstructured model is equal to the nonstationary model at lag $g = n - 1$, where the panels are balanced with $n_i = n$ for all i.

References

Aloisio, K. M., N. Micali, S. A. Swanson, A. Field, and N. J. Horton. 2014. Analysis of partially observed clustered data using generalized estimating equations and multiple imputation. *Stata Journal* 14: 863–883.

Cabanillas, O. B., J. D. Michler, A. Michuda, and E. Tjernström. 2018. Fitting and interpreting correlated random-coefficient models using Stata. *Stata Journal* 18: 159–173.

Caria, M. P., M. R. Galanti, R. Bellocco, and N. J. Horton. 2011. The impact of different sources of body mass index assessment on smoking onset: An application of multiple-source information models. *Stata Journal* 11: 386–402.

Cui, J. 2007. QIC program and model selection in GEE analyses. *Stata Journal* 7: 209–220.

Earnest, A. 2017. *Essentials of a Successful Biostatistical Collaboration*. Boca Raton, FL: CRC Press.

Gallis, J. A., F. Li, and E. L. Turner. 2020. xtgeebcv: A command for bias-corrected sandwich variance estimation for GEE analyses of cluster randomized trials. *Stata Journal* 20: 363–381.

Hardin, J. W., and J. M. Hilbe. 2013. *Generalized Estimating Equations*. 2nd ed. Boca Raton, FL: Chapman and Hall/CRC.

Hosmer, D. W., Jr., S. A. Lemeshow, and R. X. Sturdivant. 2013. *Applied Logistic Regression*. 3rd ed. Hoboken, NJ: Wiley.

Lenzi, J., and S. Pildava. 2019. Tips for calculating and displaying risk-standardized hospital outcomes in Stata. *Stata Journal* 19: 477–496.

Liang, K.-Y. 1987. Estimating functions and approximate conditional likelihood. *Biometrika* 4: 695–702. https://doi.org/10.2307/2336463.

Liang, K.-Y., and S. L. Zeger. 1986. Longitudinal data analysis using generalized linear models. *Biometrika* 73: 13–22. https://doi.org/10.1093/biomet/73.1.13.

Liang, K.-Y., S. L. Zeger, and B. Qaqish. 1992. Multivariate regression analyses for categorical data. *Journal of the Royal Statistical Society*, B ser., 54: 3–40. https://doi.org/10.1111/j.2517-6161.1992.tb01862.x.

McCullagh, P., and J. A. Nelder. 1989. *Generalized Linear Models*. 2nd ed. London: Chapman and Hall/CRC.

Nelder, J. A., and R. W. M. Wedderburn. 1972. Generalized linear models. *Journal of the Royal Statistical Society*, A ser., 135: 370–384. https://doi.org/10.2307/2344614.

Newton, H. J. 1988. *TIMESLAB: A Time Series Analysis Laboratory*. Belmont, CA: Wadsworth.

Pendergast, J. F., S. J. Gange, M. A. Newton, M. J. Lindstrom, M. Palta, and M. R. Fisher. 1996. A survey of methods for analyzing clustered binary response data. *International Statistical Review* 64: 89–118. https://doi.org/10.2307/1403425.

Prentice, R. L., and L. P. Zhao. 1991. Estimating equations for parameters in means and covariances of multivariate discrete and continuous responses. *Biometrics* 47: 825–839. https://doi.org/10.2307/2532642.

Rabe-Hesketh, S., A. Skrondal, and A. Pickles. 2002. Reliable estimation of generalized linear mixed models using adaptive quadrature. *Stata Journal* 2: 1–21.

Shults, J., and J. M. Hilbe. 2014. *Quasi-Least Squares Regression*. Boca Raton, FL: Chapman and Hall/CRC.

Shults, J., S. J. Ratcliffe, and M. Leonard. 2007. Improved generalized estimating equation analysis via xtqls for quasi-least squares in Stata. *Stata Journal* 7: 147–166.

Twisk, J. W. R. 2013. *Applied Longitudinal Data Analysis for Epidemiology: A Practical Guide*. 2nd ed. Cambridge: Cambridge University Press.

Wedderburn, R. W. M. 1974. Quasi-likelihood functions, generalized linear models, and the Gauss–Newton method. *Biometrika* 61: 439–447. https://doi.org/10.2307/2334725.

Zeger, S. L., and K.-Y. Liang. 1986. Longitudinal data analysis for discrete and continuous outcomes. *Biometrics* 42: 121–130. https://doi.org/10.2307/2531248.

Zeger, S. L., K.-Y. Liang, and P. S. Albert. 1988. Models for longitudinal data: A generalized estimating equation approach. *Biometrics* 44: 1049–1060. https://doi.org/10.2307/2531734.

Zhao, L. P., and R. L. Prentice. 1990. Correlated binary regression using a quadratic exponential model. *Biometrika* 77: 642–648. https://doi.org/10.2307/2337004.

Also see

[XT] **xtgee postestimation** — Postestimation tools for xtgee

[XT] **xtcloglog** — Random-effects and population-averaged cloglog models

[XT] **xtlogit** — Fixed-effects, random-effects, and population-averaged logit models

[XT] **xtnbreg** — Fixed-effects, random-effects, & population-averaged negative binomial models

[XT] **xtpoisson** — Fixed-effects, random-effects, and population-averaged Poisson models

[XT] **xtprobit** — Random-effects and population-averaged probit models

[XT] **xtreg** — Linear models for panel data

[XT] **xtregar** — Fixed- and random-effects linear models with an AR(1) disturbance

[XT] **xtset** — Declare data to be panel data

[MI] **Estimation** — Estimation commands for use with mi estimate

[R] **glm** — Generalized linear models

[R] **logistic** — Logistic regression, reporting odds ratios

[R] **regress** — Linear regression

[U] 20 Estimation and postestimation commands

xtgee postestimation — Postestimation tools for xtgee

Postestimation commands | predict | margins | estat
Remarks and examples | Also see

Postestimation commands

The following postestimation command is of special interest after `xtgee`:

Command	Description
`estat wcorrelation`	estimated matrix of the within-group correlations

The following standard postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of parameters
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
* `forecast`	dynamic forecasts and simulations
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	means, rates, probabilities, etc.
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of parameters
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

*`forecast` is not appropriate with `mi` estimation results.

predict

Description for predict

`predict` creates a new variable containing predictions such as predicted values, probabilities, linear predictions, standard errors, and the equation-level score.

Menu for predict

Statistics > Postestimation

Syntax for predict

`predict` [*type*] *newvar* [*if*] [*in*] [, *statistic* `nooffset`]

statistic	Description
Main	
`mu`	predicted value of *depvar*; considers the `offset()` or `exposure()`; the default
`rate`	predicted value of *depvar*
`pr(`*n*`)`	probability $\Pr(y_{it} = n)$ for `family(poisson) link(log)`
`pr(`*a*`,`*b*`)`	probability $\Pr(a \le y_{it} \le b)$ for `family(poisson) link(log)`
`xb`	linear prediction
`stdp`	standard error of the linear prediction
`score`	first derivative of the log likelihood with respect to $\mathbf{x}_{it}\boldsymbol{\beta}$

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

Options for predict

Main

`mu`, the default, and `rate` calculate the predicted value of *depvar*. `mu` takes into account the `offset()` or `exposure()` together with the denominator if the family is binomial; `rate` ignores those adjustments. `mu` and `rate` are equivalent if you did not specify `offset()` or `exposure()` when you fit the `xtgee` model and you did not specify `family(binomial #)` or `family(binomial` *varname*`)`, meaning the binomial family and a denominator not equal to one.

Thus `mu` and `rate` are the same for `family(gaussian) link(identity)`.

`mu` and `rate` are not equivalent for `family(binomial pop) link(logit)`. Then `mu` would predict the number of positive outcomes and `rate` would predict the probability of a positive outcome.

`mu` and `rate` are not equivalent for `family(poisson) link(log) exposure(time)`. Then `mu` would predict the number of events given exposure time and `rate` would calculate the incidence rate—the number of events given an exposure time of 1.

`pr(`*n*`)` calculates the probability $\Pr(y_{it} = n)$ for `family(poisson) link(log)`, where *n* is a nonnegative integer that may be specified as a number or a variable.

`pr(`*a*`,`*b*`)` calculates the probability $\Pr(a \leq y_{it} \leq b)$ for `family(poisson) link(log)`, where *a* and *b* are nonnegative integers that may be specified as numbers or variables;

b missing ($b \geq .$) means $+\infty$;
`pr(20,.)` calculates $\Pr(y_{it} \geq 20)$;
`pr(20,`*b*`)` calculates $\Pr(y_{it} \geq 20)$ in observations for which $b \geq .$ and calculates $\Pr(20 \leq y_{it} \leq b)$ elsewhere.

`pr(.,`*b*`)` produces a syntax error. A missing value in an observation of the variable *a* causes a missing value in that observation for `pr(`*a*`,`*b*`)`.

`xb` calculates the linear prediction.

`stdp` calculates the standard error of the linear prediction.

`score` calculates the equation-level score, $u_{it} = \partial \ln L(\mathbf{x}_{it}\boldsymbol{\beta})/\partial(\mathbf{x}_{it}\boldsymbol{\beta})$.

`nooffset` is relevant only if you specified `offset(`*varname*`)`, `exposure(`*varname*`)`, `family(binomial #)`, or `family(binomial` *varname*`)` when you fit the model. It modifies the calculations made by `predict` so that they ignore the offset or exposure variable and the binomial denominator. Thus `predict` ..., `mu nooffset` produces the same results as `predict` ..., `rate`.

margins

Description for margins

`margins` estimates margins of response for predicted values, probabilities, and linear predictions.

Menu for margins

Statistics > Postestimation

Syntax for margins

`margins` [*marginlist*] [, *options*]

`margins` [*marginlist*] , `predict(`*statistic* ...`)` [`predict(`*statistic* ...`)` ...] [*options*]

statistic	Description
`mu`	predicted value of *depvar*; considers the `offset()` or `exposure()`; the default
`rate`	predicted value of *depvar*
`pr(`*n*`)`	probability $\Pr(y_{it} = n)$ for `family(poisson) link(log)`
`pr(`*a*`,`*b*`)`	probability $\Pr(a \leq y_{it} \leq b)$ for `family(poisson) link(log)`
`xb`	linear prediction
`stdp`	not allowed with `margins`
`score`	not allowed with `margins`

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.

For the full syntax, see [R] **margins**.

estat

Description for estat

`estat wcorrelation` displays the estimated matrix of the within-group correlations.

Menu for estat

Statistics > Postestimation

Syntax for estat

`estat wcorrelation` [, `compact` `format(%`*fmt*`)`]

`collect` is allowed with `estat wcorrelation`; see [U] **11.1.10 Prefix commands**.

Options for estat

`compact` specifies that only the parameters (alpha) of the estimated matrix of within-group correlations be displayed rather than the entire matrix.

`format(%`*fmt*`)` overrides the display format; see [D] **format**.

Remarks and examples

▷ Example 1

`xtgee` can estimate rich correlation structures. In example 2 of [XT] **xtgee**, we fit the model

```
. use https://www.stata-press.com/data/r19/nlswork2
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)

. xtgee ln_w grade age c.age#c.age
  (output omitted)
```

After estimation, `estat wcorrelation` reports the working correlation matrix **R**:

```
. estat wcorrelation

Estimated within-idcode correlation matrix R:

        |        c1         c2         c3         c4         c5         c6
   -----+------------------------------------------------------------------
     r1 |         1
     r2 |  .4851356          1
     r3 |  .4851356   .4851356          1
     r4 |  .4851356   .4851356   .4851356          1
     r5 |  .4851356   .4851356   .4851356   .4851356          1
     r6 |  .4851356   .4851356   .4851356   .4851356   .4851356          1
     r7 |  .4851356   .4851356   .4851356   .4851356   .4851356   .4851356
     r8 |  .4851356   .4851356   .4851356   .4851356   .4851356   .4851356
     r9 |  .4851356   .4851356   .4851356   .4851356   .4851356   .4851356

        |        c7         c8         c9
   -----+--------------------------------
     r7 |         1
     r8 |  .4851356          1
     r9 |  .4851356   .4851356          1
```

The equal-correlation model corresponds to an exchangeable correlation structure, meaning that the correlation of observations within person is a constant. The working correlation estimated by `xtgee` is 0.4851. (`xtreg, re`, by comparison, reports 0.5141; see the `xtreg` command in example 2 of [XT] **xtgee**.) We constrained the model to have this simple correlation structure. What if we relaxed the constraint? To go to the other extreme, let's place no constraints on the matrix (other than its being symmetric). We do this by specifying `correlation(unstructured)`, although we can abbreviate the option.

```
. xtgee ln_w grade age c.age#c.age, corr(unstructured) nolog
GEE population-averaged model                   Number of obs     =   16,085
Group and time vars: idcode year                Number of groups =    3,913
Family: Gaussian                                Obs per group:
Link:    Identity                                            min =        1
Correlation: unstructured                                    avg =      4.1
                                                             max =        9
                                                Wald chi2(3)      = 2405.20
Scale parameter = .1418513                      Prob > chi2       =  0.0000

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
       grade |   .0720684    .002151    33.50   0.000     .0678525    .0762843
         age |   .1008095   .0081471    12.37   0.000     .0848416    .1167775
             |
 c.age#c.age |  -.0015104   .0001617    -9.34   0.000    -.0018272   -.0011936
             |
       _cons |  -.8645484   .1009488    -8.56   0.000    -1.062404   -.6666923
------------------------------------------------------------------------------

. estat wcorrelation
Estimated within-idcode correlation matrix R:
           |        c1          c2          c3          c4          c5          c6
-----------+------------------------------------------------------------------------
        r1 |         1
        r2 |  .4354838           1
        r3 |  .4280248    .5597329           1
        r4 |  .3772342    .5012129    .5475113           1
        r5 |  .4031433    .5301403     .502668    .6216227           1
        r6 |  .3663686    .4519138    .4783186    .5685009    .7306005           1
        r7 |  .2819915    .3605743    .3918118    .4012104    .4642561      .50219
        r8 |  .3162028    .3445668    .4285424    .4389241    .4696792    .5222537
        r9 |  .2148737    .3078491    .3337292    .3584013    .4865802    .4613128

           |        c7          c8          c9
-----------+------------------------------------
        r7 |         1
        r8 |  .6475654           1
        r9 |  .5791417    .7386595           1
```

This correlation matrix looks different from the previously constrained one and shows, in particular, that the serial correlation of the residuals diminishes as the lag increases, although residuals separated by small lags are more correlated than, say, AR(1) would imply.

◁

▷ Example 2

In example 1 of [XT] **xtprobit**, we showed a random-effects model of unionization using the `union` data described in [XT] **xt**. We performed the estimation using `xtprobit` but said that we could have used `xtgee` as well. Here we fit a population-averaged (equal correlation) model for comparison:

```
. use https://www.stata-press.com/data/r19/union
(NLS Women 14-24 in 1968)

. xtgee union age grade i.not_smsa south##c.year, family(binomial) link(probit)

Iteration 1:   Tolerance = .12544249
Iteration 2:   Tolerance = .0034686
Iteration 3:   Tolerance = .00017448
Iteration 4:   Tolerance = 8.382e-06
Iteration 5:   Tolerance = 3.997e-07

GEE population-averaged model                     Number of obs     = 26,200
Group variable: idcode                            Number of groups =   4,434
Family: Binomial                                  Obs per group:
Link:    Probit                                                 min =      1
Correlation: exchangeable                                       avg =    5.9
                                                                max =     12
                                                  Wald chi2(6)      = 242.57
Scale parameter = 1                               Prob > chi2       = 0.0000

------------------------------------------------------------------------------
       union | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
         age |   .0089699   .0053208     1.69   0.092    -.0014586    .0193985
       grade |   .0333174   .0062352     5.34   0.000     .0210966    .0455382
  1.not_smsa |  -.0715717    .027543    -2.60   0.009    -.1255551   -.0175884
     1.south |  -1.017368    .207931    -4.89   0.000    -1.424905   -.6098308
        year |  -.0062708   .0055314    -1.13   0.257    -.0171122    .0045706
             |
south#c.year |
          1  |   .0086294     .00258     3.34   0.001     .0035727     .013686
             |
       _cons |  -.8670997    .294771    -2.94   0.003     -1.44484   -.2893592
------------------------------------------------------------------------------
```

Let's look at the correlation structure and then relax it:

```
. estat wcorrelation, format(%8.4f)

Estimated within-idcode correlation matrix R:

             |     c1       c2       c3       c4       c5       c6       c7
-------------+----------------------------------------------------------------
          r1 |  1.0000
          r2 |  0.4615   1.0000
          r3 |  0.4615   0.4615   1.0000
          r4 |  0.4615   0.4615   0.4615   1.0000
          r5 |  0.4615   0.4615   0.4615   0.4615   1.0000
          r6 |  0.4615   0.4615   0.4615   0.4615   0.4615   1.0000
          r7 |  0.4615   0.4615   0.4615   0.4615   0.4615   0.4615   1.0000
          r8 |  0.4615   0.4615   0.4615   0.4615   0.4615   0.4615   0.4615
          r9 |  0.4615   0.4615   0.4615   0.4615   0.4615   0.4615   0.4615
         r10 |  0.4615   0.4615   0.4615   0.4615   0.4615   0.4615   0.4615
         r11 |  0.4615   0.4615   0.4615   0.4615   0.4615   0.4615   0.4615
         r12 |  0.4615   0.4615   0.4615   0.4615   0.4615   0.4615   0.4615

             |     c8       c9      c10      c11      c12
-------------+---------------------------------------------
          r8 |  1.0000
          r9 |  0.4615   1.0000
         r10 |  0.4615   0.4615   1.0000
         r11 |  0.4615   0.4615   0.4615   1.0000
         r12 |  0.4615   0.4615   0.4615   0.4615   1.0000
```

We estimate the fixed correlation between observations within person to be 0.4615. We have many data (an average of 5.9 observations on 4,434 women), so estimating the full correlation matrix is feasible. Let's do that and then examine the results:

```
. xtgee union age grade i.not_smsa south##c.year, family(binomial) link(probit)
> corr(unstructured) nolog
GEE population-averaged model                   Number of obs     = 26,200
Group and time vars: idcode year                Number of groups =  4,434
Family: Binomial                                Obs per group:
Link:    Probit                                               min =      1
Correlation: unstructured                                     avg =    5.9
                                                              max =     12
                                                Wald chi2(6)      = 198.45
Scale parameter = 1                             Prob > chi2       = 0.0000

------------------------------------------------------------------------------
       union | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
         age |   .0096612   .0053366     1.81   0.070    -.0007984    .0201208
       grade |   .0352762   .0065621     5.38   0.000     .0224148    .0481377
  1.not_smsa |   -.093073   .0291971    -3.19   0.001    -.1502983   -.0358478
     1.south |  -1.028526    .278802    -3.69   0.000    -1.574968   -.4820839
        year |  -.0088187    .005719    -1.54   0.123    -.0200278    .0023904
             |
south#c.year |
          1  |   .0089824   .0034865     2.58   0.010      .002149    .0158158
             |
       _cons |  -.7306192    .316757    -2.31   0.021    -1.351451    -.109787
------------------------------------------------------------------------------

. estat wcorrelation, format(%8.4f)
Estimated within-idcode correlation matrix R:
           |       c1        c2        c3        c4        c5        c6        c7
-----------+----------------------------------------------------------------------
        r1 |   1.0000
        r2 |   0.6667    1.0000
        r3 |   0.6151    0.6523    1.0000
        r4 |   0.5268    0.5717    0.6101    1.0000
        r5 |   0.3309    0.3669    0.4005    0.4783    1.0000
        r6 |   0.3000    0.3706    0.4237    0.4562    0.6426    1.0000
        r7 |   0.2995    0.3568    0.3851    0.4279    0.4931    0.6384    1.0000
        r8 |   0.2759    0.3021    0.3225    0.3751    0.4682    0.5597    0.7009
        r9 |   0.2989    0.2981    0.3021    0.3806    0.4605    0.5068    0.6090
       r10 |   0.2285    0.2597    0.2748    0.3637    0.3981    0.4909    0.5889
       r11 |   0.2325    0.2289    0.2696    0.3246    0.3551    0.4426    0.5103
       r12 |   0.2359    0.2351    0.2544    0.3134    0.3474    0.3822    0.4788
           |       c8        c9       c10       c11       c12
-----------+--------------------------------------------------
        r8 |   1.0000
        r9 |   0.6714    1.0000
       r10 |   0.5973    0.6325    1.0000
       r11 |   0.5625    0.5756    0.5738    1.0000
       r12 |   0.4999    0.5412    0.5329    0.6428    1.0000
```

As before, we find that the correlation of residuals decreases as the lag increases, but more slowly than an AR(1) process.

◁

▷ Example 3

In this example, we examine injury incidents among 20 airlines in each of 4 years. The data are fictional, and, as a matter of fact, are really from a random-effects model.

```
. use https://www.stata-press.com/data/r19/airacc
. generate lnpm = ln(pmiles)
. xtgee i_cnt inprog, family(poisson) eform offset(lnpm) nolog
GEE population-averaged model                      Number of obs    =     80
Group variable: airline                            Number of groups =     20
Family: Poisson                                    Obs per group:
Link:   Log                                                   min =      4
Correlation: exchangeable                                     avg =    4.0
                                                              max =      4
                                                   Wald chi2(1)     =   5.27
Scale parameter = 1                                Prob > chi2      = 0.0217

------------------------------------------------------------------------------
       i_cnt |        IRR   Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      inprog |   .9059936   .0389528    -2.30   0.022     .8327758    .9856487
       _cons |   .0080065   .0002912  -132.71   0.000     .0074555    .0085981
        lnpm |          1  (offset)
------------------------------------------------------------------------------
Note: _cons estimates baseline incidence rate (conditional on zero random
      effects).
. estat wcorrelation
Estimated within-airline correlation matrix R:

      |        c1          c2          c3          c4
------+------------------------------------------------
   r1 |         1
   r2 |  .4606406           1
   r3 |  .4606406    .4606406           1
   r4 |  .4606406    .4606406    .4606406           1
```

Now there are not really enough data here to reliably estimate the correlation without any constraints of structure, but here is what happens if we try:

```
. xtgee i_cnt inprog, family(poisson) eform offset(lnpm) corr(unstructured) nolog
GEE population-averaged model                      Number of obs    =     80
Group and time vars: airline time                  Number of groups =     20
Family: Poisson                                    Obs per group:
Link:   Log                                                   min =      4
Correlation: unstructured                                     avg =    4.0
                                                              max =      4
                                                   Wald chi2(1)     =   0.36
Scale parameter = 1                                Prob > chi2      = 0.5496

------------------------------------------------------------------------------
       i_cnt |        IRR   Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      inprog |   .9791082   .0345486    -0.60   0.550     .9136826    1.049219
       _cons |   .0078716   .0002787  -136.82   0.000     .0073439    .0084373
        lnpm |          1  (offset)
------------------------------------------------------------------------------
Note: _cons estimates baseline incidence rate (conditional on zero random
      effects).
```

```
. estat wcorrelation
Estimated within-airline correlation matrix R:

             c1         c2         c3         c4
r1            1
r2     .5700298          1
r3      .716356   .4192126          1
r4     .2383264   .3839863   .3521287          1
```

There is no sensible pattern to the correlations.

We created this dataset from a random-effects Poisson model. We reran our data-creation program and this time had it create 400 airlines rather than 20, still with 4 years of data each. Here are the equal-correlation model and estimated correlation structure:

```
. use https://www.stata-press.com/data/r19/airacc2, clear
. xtgee i_cnt inprog, family(poisson) eform offset(lnpm) nolog
GEE population-averaged model                     Number of obs    =  1,600
Group variable: airline                           Number of groups =    400
Family: Poisson                                   Obs per group:
Link:   Log                                                    min =      4
Correlation: exchangeable                                      avg =    4.0
                                                               max =      4
                                                  Wald chi2(1)     = 111.80
Scale parameter = 1                               Prob > chi2      = 0.0000

       i_cnt         IRR   Std. err.       z    P>|z|     [95% conf. interval]

      inprog    .8915304   .0096807   -10.57   0.000     .8727571    .9107076
       _cons    .0071357   .0000629  -560.57   0.000     .0070134    .0072601
        lnpm           1  (offset)

Note: _cons estimates baseline incidence rate (conditional on zero random
      effects).
. estat wcorrelation
Estimated within-airline correlation matrix R:

             c1         c2         c3         c4
r1            1
r2     .5291707          1
r3     .5291707   .5291707          1
r4     .5291707   .5291707   .5291707          1
```

The following estimation results assume unstructured correlation:

```
. xtgee i_cnt inprog, family(poisson) corr(unstructured) eform offset(lnpm) nolog
GEE population-averaged model                   Number of obs    =  1,600
Group and time vars: airline time               Number of groups =    400
Family: Poisson                                 Obs per group:
Link:   Log                                                  min =      4
Correlation: unstructured                                    avg =    4.0
                                                             max =      4
                                                Wald chi2(1)     = 113.43
Scale parameter = 1                             Prob > chi2      = 0.0000

------------------------------------------------------------------------------
       i_cnt |        IRR   Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      inprog |   .8914155   .0096208   -10.65   0.000     .8727572    .9104728
       _cons |   .0071402   .0000628  -561.50   0.000     .0070181    .0072645
        lnpm |          1  (offset)
------------------------------------------------------------------------------
Note: _cons estimates baseline incidence rate (conditional on zero random
      effects).

. estat wcorrelation
Estimated within-airline correlation matrix R:

      |        c1          c2          c3          c4
------+------------------------------------------------
   r1 |         1
   r2 |  .4733189           1
   r3 |  .5240576    .5748868           1
   r4 |  .5139748    .5048895    .5840707           1
```

The equal-correlation model estimated a fixed correlation of 0.5292, and above we have correlations ranging between 0.4733 and 0.5841 with little pattern in their structure.

◁

Also see

[XT] **xtgee** — GEE population-averaged panel-data models

[U] 20 Estimation and postestimation commands

xtgls — GLS linear model with heteroskedastic and correlated errors

Description

`xtgls` fits panel-data linear models by using feasible generalized least squares. This command allows estimation in the presence of AR(1) autocorrelation within panels and cross-sectional correlation and heteroskedasticity across panels.

Quick start

GLS regression of `y` on `x1`, `x2`, and indicators for levels of categorical variable `a` using `xtset` data

```
xtgls y x1 x2 i.a
```

With heteroskedastic but uncorrelated errors across panels

```
xtgls y x1 x2 i.a, panels(heteroskedastic)
```

With heteroskedastic and correlated errors across panels

```
xtgls y x1 x2 i.a, panels(correlated)
```

Three-stage GLS with a common first-order autocorrelation within panels

```
xtgls y x1 x2 i.a, panels(correlated) corr(ar1)
```

Same as above, but let autocorrelation structure be panel-specific

```
xtgls y x1 x2 i.a, panels(correlated) corr(psar1)
```

Same as above, but estimate by iterated GLS

```
xtgls y x1 x2 i.a, panels(correlated) corr(psar1) igls
```

Menu

Statistics > Longitudinal/panel data > Contemporaneous correlation > GLS regression with correlated errors

Syntax

xtgls *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, *options*]

options	Description
Model	
noconstant	suppress constant term
panels(iid)	use i.i.d. error structure
panels(heteroskedastic)	use heteroskedastic but uncorrelated error structure
panels(correlated)	use heteroskedastic and correlated error structure
corr(independent)	use independent autocorrelation structure
corr(ar1)	use AR1 autocorrelation structure
corr(psar1)	use panel-specific AR1 autocorrelation structure
rhotype(*calc*)	specify method to compute autocorrelation parameter; see *Options* for details; seldom used
igls	use iterated GLS estimator instead of two-step GLS estimator
force	estimate even if observations unequally spaced in time
SE	
nmk	normalize standard error by $N - k$ instead of N
Reporting	
level(#)	set confidence level; default is level(95)
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Optimization	
optimize_options	control the optimization process; seldom used
coeflegend	display legend instead of statistics

A panel variable must be specified. For correlation structures other than independent, a time variable must be specified. A time variable must also be specified if panels(correlated) is specified. Use xtset; see [XT] **xtset**.

indepvars may contain factor variables; see [U] **11.4.3 Factor variables**.

depvar and *indepvars* may contain time-series operators; see [U] **11.4.4 Time-series varlists**.

by, collect, and statsby are allowed; see [U] **11.1.10 Prefix commands**.

aweights are allowed; see [U] **11.1.6 weight**.

coeflegend does not appear in the dialog box.

See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

noconstant; see [R] **Estimation options**.

panels(*pdist*) specifies the error structure across panels.

panels(iid) specifies a homoskedastic error structure with no cross-sectional correlation. This is the default.

panels(heteroskedastic) specifies a heteroskedastic error structure with no cross-sectional correlation.

panels(correlated) specifies a heteroskedastic error structure with cross-sectional correlation. If p(c) is specified, you must also specify a time variable (use xtset). The results will be based on a generalized inverse of a singular matrix unless $T \geq m$ (the number of periods is greater than or equal to the number of panels).

corr(*corr*) specifies the assumed autocorrelation within panels.

corr(independent) specifies that there is no autocorrelation. This is the default.

corr(ar1) specifies that, within panels, there is AR(1) autocorrelation and that the coefficient of the AR(1) process is common to all the panels. If c(ar1) is specified, you must also specify a time variable (use xtset).

corr(psar1) specifies that, within panels, there is AR(1) autocorrelation and that the coefficient of the AR(1) process is specific to each panel. psar1 stands for panel-specific AR(1). If c(psar1) is specified, a time variable must also be specified; use xtset.

rhotype(*calc*) specifies the method to be used to calculate the autocorrelation parameter:

regress	regression using lags; the default
dw	Durbin–Watson calculation
freg	regression using leads
nagar	Nagar calculation
theil	Theil calculation
tscorr	time-series autocorrelation calculation

All the calculations are asymptotically equivalent and consistent; this is a rarely used option.

igls requests an iterated GLS estimator instead of the two-step GLS estimator for a nonautocorrelated model or instead of the three-step GLS estimator for an autocorrelated model. The iterated GLS estimator converges to the MLE for the corr(independent) models but does not for the other corr() models.

force specifies that estimation be forced even though the time variable is not equally spaced. This is relevant only for correlation structures that require knowledge of the time variable. These correlation structures require that observations be equally spaced so that calculations based on lags correspond to a constant time change. If you specify a time variable indicating that observations are not equally spaced, the (time dependent) model will not be fit. If you also specify force, the model will be fit, and it will be assumed that the lags based on the data ordered by the time variable are appropriate.

SE

nmk specifies that standard errors be normalized by $N-k$, where k is the number of parameters estimated, rather than N, the number of observations. Different authors have used one or the other normalization. Greene (2018, 313) remarks that whether a degree-of-freedom correction improves the small-sample properties is an open question.

Reporting

level(#); see [R] **Estimation options**.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Optimization

optimize_options control the iterative optimization process. These options are seldom used.

iterate(#) specifies the maximum number of iterations. When the number of iterations equals #, the optimization stops and presents the current results, even if convergence has not been reached. The default is iterate(100).

tolerance(#) specifies the tolerance for the coefficient vector. When the relative change in the coefficient vector from one iteration to the next is less than or equal to #, the optimization process is stopped. tolerance(1e-7) is the default.

log and nolog specify whether to display the iteration log. The iteration log is displayed by default unless you used set iterlog off to suppress it; see set iterlog in [R] ***set iter***.

The following option is available with xtgls but is not shown in the dialog box:

coeflegend; see [R] **Estimation options**.

Remarks and examples

Remarks are presented under the following headings:

Introduction

Information on GLS can be found in Greene (2018), Maddala and Lahiri (2006), Davidson and MacKinnon (1993), and Judge et al. (1985).

If you have many panels relative to periods, see [XT] **xtreg** and [XT] **xtgee**. xtgee, in particular, provides capabilities similar to those of xtgls but does not allow cross-sectional correlation. On the other hand, xtgee allows a richer description of the correlation within panels as long as the same correlations apply to all panels. xtgls provides two unique features:

1. Cross-sectional correlation may be modeled (panels(correlated)).
2. Within panels, the AR(1) correlation coefficient may be unique (corr(psar1)).

xtgls allows models with heteroskedasticity and no cross-sectional correlation, but, strictly speaking, xtgee does not. xtgee with the vce(robust) option relaxes the assumption of equal variances, at least as far as the standard error calculation is concerned.

Also, xtgls, panels(iid) corr(independent) nmk is equivalent to regress.

The nmk option uses $n - k$ rather than n to normalize the variance calculation.

To fit a model with autocorrelated errors (corr(ar1) or corr(psar1)), the data must be equally spaced in time. To fit a model with cross-sectional correlation (panels(correlated)), panels must have the same number of observations (be balanced).

The equation from which the models are developed is given by

$$y_{it} = \mathbf{x}_{it}\boldsymbol{\beta} + \epsilon_{it}$$

where $i = 1, \ldots, m$ is the number of units (or panels) and $t = 1, \ldots, T_i$ is the number of observations for panel i. This model can equally be written as

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_m \end{bmatrix} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_m \end{bmatrix} \boldsymbol{\beta} + \begin{bmatrix} \boldsymbol{\epsilon}_1 \\ \boldsymbol{\epsilon}_2 \\ \vdots \\ \boldsymbol{\epsilon}_m \end{bmatrix}$$

The variance matrix of the disturbance terms can be written as

$$E[\boldsymbol{\epsilon}\boldsymbol{\epsilon}'] = \boldsymbol{\Omega} = \begin{bmatrix} \sigma_{1,1}\boldsymbol{\Omega}_{1,1} & \sigma_{1,2}\boldsymbol{\Omega}_{1,2} & \cdots & \sigma_{1,m}\boldsymbol{\Omega}_{1,m} \\ \sigma_{2,1}\boldsymbol{\Omega}_{2,1} & \sigma_{2,2}\boldsymbol{\Omega}_{2,2} & \cdots & \sigma_{2,m}\boldsymbol{\Omega}_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{m,1}\boldsymbol{\Omega}_{m,1} & \sigma_{m,2}\boldsymbol{\Omega}_{m,2} & \cdots & \sigma_{m,m}\boldsymbol{\Omega}_{m,m} \end{bmatrix}$$

For the $\boldsymbol{\Omega}_{i,j}$ matrices to be parameterized to model cross-sectional correlation, they must be square (balanced panels).

In these models, we assume that the coefficient vector $\boldsymbol{\beta}$ is the same for all panels and consider a variety of models by changing the assumptions on the structure of $\boldsymbol{\Omega}$.

For the classic OLS regression model, we have

$$\begin{aligned} E[\epsilon_{i,t}] &= 0 \\ \text{Var}[\epsilon_{i,t}] &= \sigma^2 \\ \text{Cov}[\epsilon_{i,t}, \epsilon_{j,s}] &= 0 \qquad \text{if } t \neq s \text{ or } i \neq j \end{aligned}$$

This amounts to assuming that $\boldsymbol{\Omega}$ has the structure given by

$$\boldsymbol{\Omega} = \begin{bmatrix} \sigma^2\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \sigma^2\mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \sigma^2\mathbf{I} \end{bmatrix}$$

whether or not the panels are balanced (the $\mathbf{0}$ matrices may be rectangular). The classic OLS assumptions are the default `panels(iid)` and `corr(independent)` options for this command.

Heteroskedasticity across panels

In many cross-sectional datasets, the variance for each of the panels differs. It is common to have data on countries, states, or other units that have variation of scale. The heteroskedastic model is specified by including the `panels(heteroskedastic)` option, which assumes that

$$\boldsymbol{\Omega} = \begin{bmatrix} \sigma_1^2\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \sigma_2^2\mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \sigma_m^2\mathbf{I} \end{bmatrix}$$

▷ Example 1

Greene (2012, 1112) reprints data in a classic study of investment demand by Grunfeld and Griliches (1960). Below we allow the variances to differ for each of the five companies.

```
. use https://www.stata-press.com/data/r19/invest2
. xtgls invest market stock, panels(hetero)
Cross-sectional time-series FGLS regression

Coefficients:  generalized least squares
Panels:        heteroskedastic
Correlation:   no autocorrelation

Estimated covariances      =         5          Number of obs     =        100
Estimated autocorrelations =         0          Number of groups  =          5
Estimated coefficients     =         3          Time periods      =         20
                                                Wald chi2(2)      =     865.38
                                                Prob > chi2       =     0.0000

------------------------------------------------------------------------------
      invest | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      market |   .0949905    .007409    12.82   0.000     .0804692    .1095118
       stock |   .3378129   .0302254    11.18   0.000     .2785722    .3970535
       _cons |   -36.2537   6.124363    -5.92   0.000    -48.25723   -24.25017
------------------------------------------------------------------------------
```

◁

Correlation across panels (cross-sectional correlation)

We may wish to assume that the error terms of panels are correlated, in addition to having different scale variances. The variance structure is specified by including the `panels(correlated)` option and is given by

$$\boldsymbol{\Omega} = \begin{bmatrix} \sigma_1^2\mathbf{I} & \sigma_{1,2}\mathbf{I} & \cdots & \sigma_{1,m}\mathbf{I} \\ \sigma_{2,1}\mathbf{I} & \sigma_2^2\mathbf{I} & \cdots & \sigma_{2,m}\mathbf{I} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{m,1}\mathbf{I} & \sigma_{m,2}\mathbf{I} & \cdots & \sigma_m^2\mathbf{I} \end{bmatrix}$$

Because we must estimate cross-sectional correlation in this model, the panels must be balanced (and $T \geq m$ for valid results). A time variable must also be specified so that `xtgls` knows how the observations within panels are ordered. `xtset` shows us that this is true.

▷ Example 2

```
. xtset
Panel variable: company (strongly balanced)
 Time variable: time, 1 to 20
         Delta: 1 unit
. xtgls invest market stock, panels(correlated)
Cross-sectional time-series FGLS regression
Coefficients:  generalized least squares
Panels:        heteroskedastic with cross-sectional correlation
Correlation:   no autocorrelation
Estimated covariances      =        15          Number of obs     =        100
Estimated autocorrelations =         0          Number of groups  =          5
Estimated coefficients     =         3          Time periods      =         20
                                                Wald chi2(2)      =    1285.19
                                                Prob > chi2       =     0.0000

------------------------------------------------------------------------------
      invest | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      market |   .0961894   .0054752    17.57   0.000     .0854583    .1069206
       stock |   .3095321   .0179851    17.21   0.000     .2742819    .3447822
       _cons |  -38.36128   5.344871    -7.18   0.000    -48.83703   -27.88552
------------------------------------------------------------------------------
```

The estimated cross-sectional covariances are stored in `e(Sigma)`.

```
. matrix list e(Sigma)
symmetric e(Sigma)[5,5]
             _ee         _ee2         _ee3         _ee4         _ee5
 _ee   9410.9061
_ee2  -168.04631    755.85077
_ee3  -1915.9538   -4163.3434     34288.49
_ee4  -1129.2896   -80.381742    2259.3242    633.42367
_ee5   258.50132     4035.872   -27898.235   -1170.6801    33455.511
```

◁

▷ Example 3

We can obtain the MLE results by specifying the `igls` option, which iterates the GLS estimation technique to convergence:

```
. xtgls invest market stock, panels(correlated) igls
Iteration 1:  Tolerance = .2127384
Iteration 2:  Tolerance = .22817
  (output omitted)
Iteration 1046: Tolerance = 1.000e-07
Cross-sectional time-series FGLS regression
Coefficients:  generalized least squares
Panels:        heteroskedastic with cross-sectional correlation
Correlation:   no autocorrelation
Estimated covariances      =        15          Number of obs     =        100
Estimated autocorrelations =         0          Number of groups  =          5
Estimated coefficients     =         3          Time periods      =         20
                                                Wald chi2(2)      =     558.51
Log likelihood             = -515.4222          Prob > chi2       =     0.0000

------------------------------------------------------------------------------
      invest | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      market |    .023631    .004291     5.51   0.000     .0152207    .0320413
       stock |   .1709472   .0152526    11.21   0.000     .1410526    .2008417
       _cons |  -2.216508   1.958845    -1.13   0.258    -6.055774    1.622759
------------------------------------------------------------------------------
```

Here the log likelihood is reported in the header of the output.

◁

Autocorrelation within panels

The individual identity matrices along the diagonal of $\boldsymbol{\Omega}$ may be replaced with more general structures to allow for serial correlation. `xtgls` allows three options so that you may assume a structure with `corr(independent)` (no autocorrelation); `corr(ar1)` (serial correlation where the correlation parameter is common for all panels); or `corr(psar1)` (serial correlation where the correlation parameter is unique for each panel).

The restriction of a common autocorrelation parameter is reasonable when the individual correlations are nearly equal and the time series are short.

If the restriction of a common autocorrelation parameter is reasonable, this allows us to use more information in estimating the autocorrelation parameter to produce a more reasonable estimate of the regression coefficients.

When you specify `corr(ar1)` or `corr(psar1)`, the iterated GLS estimator does not converge to the MLE.

▷ Example 4

If corr(ar1) is specified, each group is assumed to have errors that follow the same AR(1) process; that is, the autocorrelation parameter is the same for all groups.

```
. xtgls invest market stock, panels(hetero) corr(ar1)
Cross-sectional time-series FGLS regression

Coefficients:  generalized least squares
Panels:        heteroskedastic
Correlation:   common AR(1) coefficient for all panels  (0.8651)

Estimated covariances      =         5          Number of obs     =        100
Estimated autocorrelations =         1          Number of groups  =          5
Estimated coefficients     =         3          Time periods      =         20
                                                Wald chi2(2)      =     119.69
                                                Prob > chi2       =     0.0000

------------------------------------------------------------------------------
      invest | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      market |   .0744315   .0097937     7.60   0.000     .0552362    .0936268
       stock |   .2874294   .0475391     6.05   0.000     .1942545    .3806043
       _cons |  -18.96238   17.64943    -1.07   0.283    -53.55464    15.62987
------------------------------------------------------------------------------
```

◁

▷ Example 5

If corr(psar1) is specified, each group is assumed to have errors that follow a different AR(1) process.

```
. xtgls invest market stock, panels(iid) corr(psar1)
Cross-sectional time-series FGLS regression

Coefficients:  generalized least squares
Panels:        homoskedastic
Correlation:   panel-specific AR(1)

Estimated covariances      =         1          Number of obs     =        100
Estimated autocorrelations =         5          Number of groups  =          5
Estimated coefficients     =         3          Time periods      =         20
                                                Wald chi2(2)      =     252.93
                                                Prob > chi2       =     0.0000

------------------------------------------------------------------------------
      invest | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      market |   .0934343   .0097783     9.56   0.000     .0742693    .1125993
       stock |   .3838814   .0416775     9.21   0.000      .302195    .4655677
       _cons |   -10.1246   34.06675    -0.30   0.766     -76.8942    56.64499
------------------------------------------------------------------------------
```

◁

Stored results

`xtgls` stores the following in `e()`:

Scalars

`e(N)`	number of observations
`e(N_ic)`	number of observations used to compute information criteria
`e(N_g)`	number of groups
`e(N_t)`	number of periods
`e(N_miss)`	number of missing observations
`e(n_cf)`	number of estimated coefficients
`e(n_cv)`	number of estimated covariances
`e(n_cr)`	number of estimated correlations
`e(df)`	degrees of freedom
`e(df_pear)`	degrees of freedom for Pearson χ^2
`e(df_ic)`	degrees of freedom for information criteria
`e(ll)`	log likelihood
`e(chi2)`	χ^2
`e(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(g_max)`	largest group size
`e(rank)`	rank of `e(V)`
`e(rc)`	return code

Macros

`e(cmd)`	`xtgls`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(ivar)`	variable denoting groups
`e(tvar)`	variable denoting time within groups
`e(coefftype)`	estimation scheme
`e(corr)`	correlation structure
`e(vt)`	panel option
`e(rhotype)`	type of estimated correlation
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(title)`	title in estimation output
`e(chi2type)`	`Wald`; type of model χ^2 test
`e(rho)`	ρ
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`

Matrices

`e(b)`	coefficient vector
`e(Sigma)`	$\widehat{\boldsymbol{\Sigma}}$ matrix
`e(V)`	variance–covariance matrix of the estimators

Functions

`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

The GLS results are given by

$$\widehat{\beta}_{\text{GLS}} = (\mathbf{X}'\widehat{\boldsymbol{\Omega}}^{-1}\mathbf{X})^{-1}\mathbf{X}'\widehat{\boldsymbol{\Omega}}^{-1}\mathbf{y}$$
$$\widehat{\text{Var}}(\widehat{\beta}_{\text{GLS}}) = (\mathbf{X}'\widehat{\boldsymbol{\Omega}}^{-1}\mathbf{X})^{-1}$$

For all our models, the $\boldsymbol{\Omega}$ matrix may be written in terms of the Kronecker product:

$$\boldsymbol{\Omega} = \boldsymbol{\Sigma}_{m\times m} \otimes \mathbf{I}_{T_i\times T_i}$$

The estimated variance matrix is obtained by substituting the estimator $\widehat{\boldsymbol{\Sigma}}$ for $\boldsymbol{\Sigma}$, where

$$\widehat{\boldsymbol{\Sigma}}_{i,j} = \frac{\widehat{\boldsymbol{\epsilon}}_i{}'\,\widehat{\boldsymbol{\epsilon}}_j}{T}$$

The residuals used in estimating $\boldsymbol{\Sigma}$ are first obtained from OLS regression. If the estimation is iterated, residuals are obtained from the last fitted model.

Maximum likelihood estimates may be obtained by iterating the FGLS estimates to convergence for models with no autocorrelation, `corr(independent)`.

The GLS estimates and their associated standard errors are calculated using $\widehat{\boldsymbol{\Sigma}}^{-1}$. As Beck and Katz (1995) point out, the $\boldsymbol{\Sigma}$ matrix is of rank at most $\min(T, m)$ when you use the `panels(correlated)` option. For the GLS results to be valid (not based on a generalized inverse), T must be at least as large as m, as you need at least as many period observations as there are panels.

Beck and Katz (1995) suggest using OLS parameter estimates with asymptotic standard errors that are corrected for correlation between the panels. This estimation can be performed with the `xtpcse` command; see [XT] **xtpcse**.

References

Baum, C. F. 2001. Residual diagnostics for cross-section time series regression models. *Stata Journal* 1: 101–104.

Beck, N. L., and J. N. Katz. 1995. What to do (and not to do) with time-series cross-section data. *American Political Science Review* 89: 634–647. https://doi.org/10.2307/2082979.

Blackwell, J. L., III. 2005. Estimation and testing of fixed-effect panel-data systems. *Stata Journal* 5: 202–207.

Davidson, R., and J. G. MacKinnon. 1993. *Estimation and Inference in Econometrics*. New York: Oxford University Press.

Greene, W. H. 2012. *Econometric Analysis*. 7th ed. Upper Saddle River, NJ: Prentice Hall.

———. 2018. *Econometric Analysis*. 8th ed. New York: Pearson.

Grunfeld, Y., and Z. Griliches. 1960. Is aggregation necessarily bad? *Review of Economics and Statistics* 42: 1–13. https://doi.org/10.2307/1926089.

Herwartz, H., S. Maxand, F. H. C. Raters, and Y. M. Walle. 2018. Panel unit-root tests for heteroskedastic panels. *Stata Journal* 18: 184–196.

Hoechle, D. 2007. Robust standard errors for panel regressions with cross-sectional dependence. *Stata Journal* 7: 281–312.

Judge, G. G., W. E. Griffiths, R. C. Hill, H. Lütkepohl, and T.-C. Lee. 1985. *The Theory and Practice of Econometrics*. 2nd ed. New York: Wiley.

Maddala, G. S., and K. Lahiri. 2006. *Introduction to Econometrics*. 4th ed. New York: Wiley.

Also see

[XT] **xtgls postestimation** — Postestimation tools for xtgls

[XT] **xtpcse** — Linear regression with panel-corrected standard errors

[XT] **xtreg** — Linear models for panel data

[XT] **xtregar** — Fixed- and random-effects linear models with an AR(1) disturbance

[XT] **xtset** — Declare data to be panel data

[R] **regress** — Linear regression

[TS] **newey** — Regression with Newey–West standard errors

[TS] **prais** — Prais–Winsten and Cochrane–Orcutt regression

[U] 20 Estimation and postestimation commands

Postestimation commands

The following postestimation commands are available after `xtgls`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of parameters
* `estat ic`	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC, respectively)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
`forecast`	dynamic forecasts and simulations
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
* `lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	predictions and their SEs, etc.
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of parameters
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

* `estat ic` and `lrtest` are available only if `igls` and `corr(independent)` were specified at estimation.

predict

Description for predict

predict creates a new variable containing predictions such as linear predictions and standard errors.

Menu for predict

Statistics > Postestimation

Syntax for predict

predict [*type*] *newvar* [*if*] [*in*] [, xb stdp]

These statistics are available both in and out of sample; type predict ... if e(sample) ... if wanted only for the estimation sample.

Options for predict

Main

xb, the default, calculates the linear prediction.

stdp calculates the standard error of the linear prediction.

margins

Description for margins

margins estimates margins of response for linear predictions.

Menu for margins

Statistics > Postestimation

Syntax for margins

margins [*marginlist*] [, *options*]

margins [*marginlist*] , predict(*statistic* ...) [*options*]

statistic	Description
xb	linear prediction; the default
stdp	not allowed with margins

Statistics not allowed with margins are functions of stochastic quantities other than e(b).

For the full syntax, see [R] **margins**.

Also see

[XT] **xtgls** — GLS linear model with heteroskedastic and correlated errors

[U] **20 Estimation and postestimation commands**

xthdidregress — Heterogeneous difference in differences for panel data

Description

`xthdidregress` estimates average treatment effect on the treated (ATETs) that may vary over time and over treatment cohorts. Treatment cohorts are groups subject to treatment at different points in time. `xthdidregress` provides four estimators: extended two-way fixed effects (TWFE), regression adjustment (RA), inverse-probability weighting (IPW), and augmented inverse-probability weighting (AIPW). See [CAUSAL] **teffects intro** for a discussion of RA, IPW, and AIPW estimators.

Quick start

Estimate ATETs of treatment `treat` on outcome `y` with group `grpvar`; use the RA estimator and model `y` with covariate `x` on `xtset` data

```
xthdidregress ra (y x) (treat), group(grpvar)
```

Same as above, but use the TWFE estimator

```
xthdidregress twfe (y x) (treat), group(grpvar)
```

Use the IPW estimator and model `treat` using covariate `z`

```
xthdidregress ipw (y) (treat z), group(grpvar)
```

Use the AIPW estimator, model `y` using covariate `x`, and model `treat` using covariate `z`

```
xthdidregress aipw (y x) (treat z), group(grpvar)
```

Same as above, but use the not-yet-treated group as the control group

```
xthdidregress aipw (y x) (treat z), group(grpvar) ///
    controlgroup(notyet)
```

Same as above, but cluster at the `county` level

```
xthdidregress aipw (y x) (treat z), group(grpvar) ///
    controlgroup(notyet) vce(cluster county)
```

Menu

Statistics > Longitudinal/panel data > Difference in differences (DID) > Heterogeneous DID (TWFE)

Statistics > Longitudinal/panel data > Difference in differences (DID) > Heterogeneous DID (RA)

Statistics > Longitudinal/panel data > Difference in differences (DID) > Heterogeneous DID (IPW)

Statistics > Longitudinal/panel data > Difference in differences (DID) > Heterogeneous DID (AIPW)

Syntax

For syntax, methods, and all other information on `xthdidregress`, see [CAUSAL] **xthdidregress**.

xtheckman — Random-effects regression with sample selection

Description

`xtheckman` fits a random-effects linear regression model with endogenous sample selection.

Quick start

Random-effects model of `y` on `x1` using `xtset` data, with selection indicated by binary variable `selected` and predicted by `v1` and `x1`

```
xtheckman y x1, select(selected = v1 x1)
```

Same as above, but constraining random effects to be independent

```
xtheckman y x1, select(selected = v1 x1) norecorrelation
```

Same as above, but omit random effects from selection model

```
xtheckman y x1, select(selected = v1 x1, nore)
```

Menu

Statistics > Longitudinal/panel data > Sample-selection models > Linear regression with sample selection (RE)

Syntax

`xtheckman` *depvar* [*indepvars*] [*if*] [*in*] ,
  `select(`*depvar*$_s$ = *varlist*$_s$ [, *sel_options*]`)` [*options*]

options	Description
Model	
* `select()`	specify selection equation: dependent and independent variables; whether to have constant term and offset variable or include random effect
`noconstant`	suppress constant term
`norecorrelation`	constrain the random effects to be independent
`offset(`*varname*$_o$`)`	include *varname*$_o$ in model with coefficient constrained to 1
`constraints(`*numlist*`)`	apply specified linear constraints
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `robust`, `cluster` *clustvar*, `opg`, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`nocnsreport`	do not display constraints
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Integration	
`intmethod(`*intmethod*`)`	integration method for random effects; *intmethod* may be `mvaghermite` (the default) or `ghermite`
`intpoints(#)`	set the number of integration (quadrature) points for random-effects integration; default is `intpoints(7)`
Maximization	
maximize_options	control the maximization process; seldom used
`collinear`	keep collinear variables
`coeflegend`	display legend instead of statistics

sel_options	Description
Model	
`noconstant`	suppress constant term
`nore`	do not include random effects in selection model
`offset(`*varname*$_o$`)`	include *varname*$_o$ in model with coefficient constrained to 1

*`select()` is required.

indepvars and *varlist*$_s$ may contain factor variables; see **[U] 11.4.3 Factor variables**.

depvar, *indepvars*, *depvar*$_s$, and *varlist*$_s$ may contain time-series operators; see **[U] 11.4.4 Time-series varlists**.

`bootstrap`, `by`, `collect`, `jackknife`, and `statsby` are allowed; see **[U] 11.1.10 Prefix commands**.

`collinear` and `coeflegend` do not appear in the dialog box.

See **[U] 20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

`select(`*depvar*$_s$ = *varlist*$_s$ [, *sel_options*]`)` specifies a random-effects probit model for sample selection with *varlist*$_s$ as the covariates for the selection model. When *depvar*$_s$ $= 1$, the model's dependent variable is treated as observed (selected); when *depvar*$_s$ $= 0$, it is treated as unobserved (not selected). `select()` is required.

sel_options are the following:

`noconstant` suppresses the constant term (intercept) in the selection model.

`nore` specifies that a random effect not be included in the selection equation.

`offset(`*varname*$_o$`)` specifies that *varname*$_o$ be included in the selection model with the coefficient constrained to 1.

`noconstant`; see [R] **Estimation options**.

`norecorrelation` constrains the random effects in the outcome and selection equations to be independent.

`offset(`*varname*$_o$`)`, `constraints(`*numlist*`)`; see [R] **Estimation options**.

SE/Robust

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`oim`, `opg`), that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster` *clustvar*), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [XT] ***vce_options***.

Reporting

`level(#)`, `nocnsreport`; see [R] **Estimation options**.

display_options: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(`*style*`)`, `cformat(`*%fmt*`)`, `pformat(`*%fmt*`)`, `sformat(`*%fmt*`)`, and `nolstretch`; see [R] **Estimation options**.

Integration

`intmethod(`*intmethod*`)` and `intpoints(#)` control how the integration of random effects is numerically calculated.

`intmethod()` specifies the integration method. The default method is mean-variance adaptive Gauss–Hermite quadrature, `intmethod(mvaghermite)`. We recommend this method. `intmethod(ghermite)` specifies that nonadaptive Gauss–Hermite quadrature be used. This method is less computationally intensive and less accurate. It is sometimes useful to try `intmethod(ghermite)` to get the model to converge and then perhaps use the results as initial values specified in `from` when fitting the model using the more accurate `intmethod(mvaghermite)`. See *Methods and formulas* for more details.

`intpoints()` sets the number of integration (quadrature) points used for integration of the random effects. The default is `intpoints(7)`. Increasing the number increases accuracy but also increases computational time. Computational time is roughly proportional to the number specified. See *Methods and formulas* for more details.

Maximization

maximize_options: difficult, technique(*algorithm_spec*), iterate(#), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), nonrtolerance, and from(*init_specs*); see [R] **Maximize**. These options are seldom used.

The default technique is technique(bhhh 10 nr 2).

Setting the optimization type to technique(bhhh) resets the default *vcetype* to vce(opg).

The following options are available with xtheckman but are not shown in the dialog box:

collinear, coeflegend; see [R] **Estimation options**.

Remarks and examples

xtheckman fits a panel-data model with endogenous sample selection. Endogenous sample selection is sometimes called nonignorability of selection, missing not at random, or selection bias. Within-panel correlation is accounted for by using panel-level random effects.

The outcome of interest y_{it} is modeled as

$$y_{it} = \mathbf{x}_{it}\boldsymbol{\beta} + \nu_{1i} + \epsilon_{1it}$$

where $\mathbf{x}_{it}$ are the covariates modeling the outcome, ν_{1i} is the panel-level random effect, and ϵ_{1it} is the observation-level error.

We model the selection process for the outcome by

$$s_{it} = 1\left(\mathbf{z}_{it}\boldsymbol{\alpha} + \nu_{2i} + \epsilon_{2it} > 0\right)$$

where $s_{it} = 1$ if we observe y_{it} and 0 otherwise, $\mathbf{z}_{it}$ are the covariates modeling selection, ν_{2i} is the panel-level random effect for selection, and ϵ_{2it} is the observation-level selection error.

The random effects ν_{1i} and ν_{2i} are bivariate normal with mean 0 and variance

$$\begin{bmatrix} \sigma^2_{1\nu} & \rho_\nu\sigma_{1\nu}\sigma_{2\nu} \\ \rho_\nu\sigma_{1\nu}\sigma_{2\nu} & \sigma^2_{2\nu} \end{bmatrix}$$

The observation-level errors ϵ_{1it} and ϵ_{2it} are bivariate normal, with mean 0 and variance

$$\begin{bmatrix} \sigma^2_1 & \rho\sigma_1 \\ \rho\sigma_1 & 1 \end{bmatrix}$$

These observation-level errors are independent of the random effects.

Using the Heckman estimator (heckman) for this model will provide inefficient estimates because it ignores the within-panel correlation. Instead, we use maximum likelihood to model both the selection and outcome equations and account for the panel structure of the data. This random-effects estimator is used by xteregress and was discussed in Rabe-Hesketh, Skrondal, and Pickles (2002). There is no parametric fixed-effects estimator for panel data from an endogenously selected sample. See Honoré, Kyriazidou, and Powell (2000) and Kyriazidou (1997) for semiparametric fixed-effects estimators of panel-data endogenous sample-selection models.

▷ Example 1

Suppose that we wish to study the relationship between wage, job tenure, and age for college-educated adults. We have fictional data on 600 adults observed from 2013 to 2016. We use these data to model hourly wage as a function of age, age squared, and job tenure. However, an individual's wage is observed only if he or she works, and not everyone was employed on the dates the data were collected. We are not interested in modeling only the subpopulation of individuals who were employed at the time. We are also interested in the relationship of job tenure and age with the wage an individual would have received if he or she had been employed.

We suspect that the unobserved factors that affect an individual's wage are related to the unobserved factors that affect employment status. These unobserved factors could include person-level characteristics like ability and time-varying factors like an individual's family situation. We suspect that we have an endogenously selected sample. We have data on the local job market conditions (`market`). This variable is used with age and tenure to model the employment status of an individual.

Before we can fit a random-effects model to our data, we need to specify the panel structure of the data using `xtset`. Our panel variable is `personid`, the identification code for the individual. The time variable is `year`, and it ranges from 2013 to 2016.

```
. use https://www.stata-press.com/data/r19/wagework
(Wages for 20 to 77 year olds, 2013-2016)
. xtset personid year
Panel variable: personid (strongly balanced)
 Time variable: year, 2013 to 2016
         Delta: 1 unit
```

We are now ready to fit our model.

```
. xtheckman wage c.age##c.age tenure, select(working = c.age##c.age market)
(setting technique to bhhh)
Iteration 0:  Log likelihood = -5384.5076
Iteration 1:  Log likelihood = -5377.4625
Iteration 2:  Log likelihood = -5376.4805
Iteration 3:  Log likelihood = -5376.4505
Iteration 4:  Log likelihood = -5376.4464
Iteration 5:  Log likelihood = -5376.4454
Iteration 6:  Log likelihood = -5376.4451
Iteration 7:  Log likelihood = -5376.4451
Iteration 8:  Log likelihood =  -5376.445
Iteration 9:  Log likelihood =  -5376.445
(switching technique to nr)
Iteration 10: Log likelihood =  -5376.445
Random-effects regression with selection        Number of obs     =    2,400
                                                      Selected =    1,928
                                                   Nonselected =      472

Group variable: personid                        Number of groups  =      600

                                                Obs per group:
                                                           min =        4
                                                           avg =      4.0
                                                           max =        4

Integration method: mvaghermite                 Integration pts.  =        7

                                                Wald chi2(3)      =  2827.78
Log likelihood = -5376.445                      Prob > chi2       =   0.0000
```

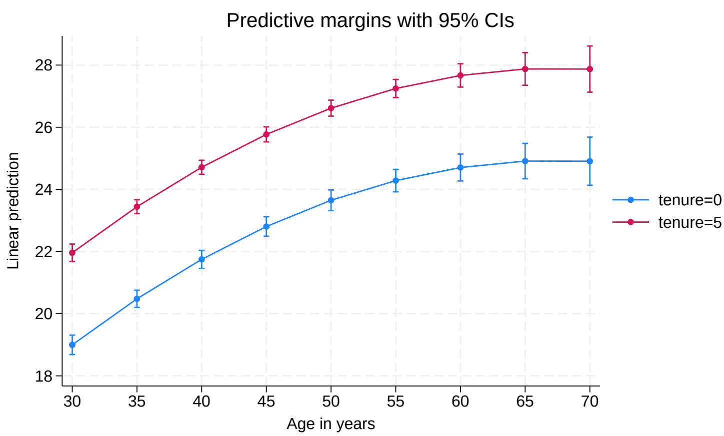
	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
wage						
age	.5722234	.0477613	11.98	0.000	.4786129	.6658339
c.age#c.age	-.0042448	.0005329	-7.97	0.000	-.0052893	-.0032003
tenure	.5927719	.0169866	34.90	0.000	.5594787	.626065
_cons	5.651812	1.038011	5.44	0.000	3.617347	7.686277
working						
age	.2305309	.0207988	11.08	0.000	.1897661	.2712958
c.age#c.age	-.0026832	.0002241	-11.97	0.000	-.0031225	-.0022439
market	.1894934	.019038	9.95	0.000	.1521796	.2268072
_cons	-3.276904	.4352836	-7.53	0.000	-4.130045	-2.423764
var(e.wage)	4.458219	.2235342			4.040939	4.918588
corr(e.wor~g, e.wage)	.4091115	.1391856	2.94	0.003	.1065022	.642359
var(wage[pers~d])	2.493737	.2547628			2.041226	3.046564
var(wor~g[per~d])	.3831411	.0830963			.250466	.5860961
corr(wor~g[per~d], wage[pers~d])	.6021096	.0845675	7.12	0.000	.4106863	.7426953

The first two sections of the output provide the estimated coefficients for the `wage` equation and the selection (`working`) equation. We can interpret the coefficients in the `wage` equation using the standard linear regression interpretation. For example, we expect an increase of $0.59 per hour for an additional year of job tenure.

Next we see `var(e.wage)`, an estimate of the variance of the observation-level error for wage; this is followed by `corr(e.working,e.wage)`, an estimate of its correlation with the observation-level error for the selection model. The next section of the output reports estimates of the variances of the random effects, `var(wage[personid])` and `var(working[personid])`. The last section reports an estimate of the correlation of these random effects. If at least one of the correlations is significantly different from zero, we can conclude that we have endogenous sample selection. In our case, the correlation between the observation-level errors is 0.41, and the correlation between the random effects is 0.60. Because both are positive and significantly different from zero, we conclude that we have endogenous selection and that unobserved individual-level factors that increase the chance of being employed tend to increase wage. Additionally, unobserved observation-level (time-varying) factors that increase the chance of being employed tend to increase wage.

We estimated coefficients for age and age squared. We can use `margins` and `marginsplot` to gain a clearer understanding of the effect of the individuals' age on hourly wage. We use `margins` with `at()` to profile the expected wages for individuals between ages 30 and 70 and with 0 and 5 years of job tenure. Then, we use `marginsplot` to graph the estimates.

```
. margins, at(age=(30(5)70) tenure =(0 5))
. marginsplot
```



Based on this model, and assuming the data are from a random or otherwise representative sample, the plotted points represent the expected wage for individuals with the specified job tenure and age. We see that age has an increasing effect on expected wage until the mid-60s and then attenuates. Having 5 years of job tenure instead of none shifts the curve up by about $3.00 per hour.

◁

Stored results

xtheckman stores the following in e():

Scalars

e(N)	number of observations
e(N_g)	number of groups
e(N_selected)	number of selected observations
e(N_nonselected)	number of nonselected observations
e(k)	number of parameters
e(k_eq)	number of equations in e(b)
e(k_eq_model)	number of equations in overall model test
e(k_aux)	number of auxiliary parameters
e(k_dv)	number of dependent variables
e(df_m)	model degrees of freedom
e(ll)	log likelihood
e(N_clust)	number of clusters
e(chi2)	χ^2
e(p)	p-value for model test
e(n_requad)	number of integration points for random effects
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(rank)	rank of e(V)
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros

e(cmd)	xtheckman
e(cmdline)	command as typed
e(depvar)	names of dependent variables
e(ivar)	variable denoting groups
e(title)	title in estimation output
e(clustvar)	name of cluster variable
e(offset#)	offset for the #th *depvar*, where # is determined by equation order in output
e(chi2type)	Wald; type of model χ^2 test
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(reintmethod)	integration method for random effects
e(opt)	type of optimization
e(which)	max or min; whether optimizer is to perform maximization or minimization
e(ml_method)	type of ml method
e(user)	name of likelihood-evaluator program
e(technique)	maximization technique
e(properties)	b V
e(estat_cmd)	program used to implement estat
e(predict)	program used to implement predict
e(marginsok)	predictions allowed by margins
e(marginsnotok)	predictions disallowed by margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved

Matrices

`e(b)`	coefficient vector
`e(Cns)`	constraints matrix
`e(ilog)`	iteration log (up to 20 iterations)
`e(gradient)`	gradient vector
`e(V)`	variance–covariance matrix of the estimators
`e(V_modelbased)`	model-based variance

Functions

`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

`xtheckman` fits a random-effects linear regression model with endogenous sample selection via maximum likelihood estimation. For details on maximum likelihood estimators, see the results in Wooldridge (2010, chap. 13) and White (1996).

The log-likelihood function maximized by `xtheckman` is implied by the triangular structure of the model. Specifically, the joint distribution of the outcome and selection variables is a product of conditional and marginal distributions because the model is triangular. For a few of the many relevant applications of this result in literature, see Amemiya (1985, chap. 10); Heckman (1976, 1979); Maddala (1983, chap. 5); Maddala and Lee (1976); Wooldridge (2010, sec. 15.7.2, 15.7.3, 16.3.3, 17.5.2, and 19.7.1; 2014). Roodman (2011) and Bartus and Roodman (2014) used this result to derive the formulas discussed below.

We have panels $i = 1, \ldots, N$ and observations $t = 1, \ldots, N_i$. We model y_{it} as

$$y_{it} = \mathbf{x}_{it}\boldsymbol{\beta} + \nu_{1i} + \epsilon_{1it}$$

where $\mathbf{x}_{it}$ are the outcome covariates, ν_{1i} is the panel-level random effect, and ϵ_{1it} is the observation-level error.

The selection process for the outcome is modeled by

$$s_{it} = 1\left(\mathbf{z}_{it}\boldsymbol{\alpha} + \nu_{2i} + \epsilon_{2it} > 0\right)$$

where $s_{it} = 1$ if we observe y_{it} and 0 otherwise, $\mathbf{z}_{it}$ are the selection covariates, ν_{2i} is the panel-level random effect for selection, and ϵ_{2it} is the observation-level selection error.

The random effects ν_{1i} and ν_{2i} are bivariate normal with mean 0 and variance

$$\boldsymbol{\Sigma}_{\nu} = \begin{bmatrix} \sigma^2_{1\nu} & \rho_{\nu}\sigma_{1\nu}\sigma_{2\nu} \\ \rho_{\nu}\sigma_{1\nu}\sigma_{2\nu} & \sigma^2_{2\nu} \end{bmatrix}$$

The observation-level errors ϵ_{1it} and ϵ_{2it} are independent of the random effects and are also bivariate normal, with mean 0 and variance

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma^2_1 & \rho\sigma_1 \\ \rho\sigma_1 & 1 \end{bmatrix}$$

When we condition on the random effects ν_{1i} and ν_{2i}, we can write the joint density of the y_{it} and s_{it} using the conditional density of the selection error ϵ_{2it} on the outcome error ϵ_{1it}.

For the selection indicator s_i, we have lower limit l_{it} and upper limit u_{it},

$$l_{it} = \begin{cases} -\infty & s_{it} = 0 \\ -\mathbf{z}_{it}\boldsymbol{\alpha} - \nu_{2i} - \frac{\rho}{\sigma_1}\left(y_{it} - \mathbf{x}_{it}\boldsymbol{\beta} - \nu_{1i}\right) & s_{it} = 1 \end{cases} \qquad u_{it} = \begin{cases} -\mathbf{z}_{it}\boldsymbol{\alpha} - \nu_{2i} & s_{it} = 0 \\ \infty & s_{it} = 1 \end{cases}$$

Then, the joint density of y_{it} and s_{it} conditional on the random effects is

$$f(y_{it}, s_{it}|\nu_{1i}, \nu_{2i}) = \begin{cases} \Phi\left(\frac{-l_{it}}{\sqrt{1-\rho^2}}\right) + \phi\left(\frac{y_{it}-\mathbf{x}_{it}\boldsymbol{\beta}}{\sigma_1}\right) & s_{it} = 1 \\ \Phi\left(\frac{u_{it}}{\sqrt{1-\rho^2}}\right) & s_{it} = 0 \end{cases}$$

Note that each panel has the same random effects for each observation. So the likelihood for panel i is

$$L_i = \int_{\Re^2} \left[\prod_{t=1}^{N_i} f(y_{it}, s_{it}|\nu_{1i}, \nu_{2i})\phi_2\{(\nu_{1i}, \nu_{2i}), \boldsymbol{\Sigma}_\nu\}\, d\nu_{1i} d\nu_{2i}\right] \tag{1}$$

This multivariate integral is generally not tractable. We can use a change-of-variables technique to transform it into a set of nested univariate integrals. Let $\mathbf{L}$ be the Cholesky decomposition of $\boldsymbol{\Sigma}_\nu$; that is, $\boldsymbol{\Sigma}_\nu = \mathbf{L}\mathbf{L}'$. It follows that $(\nu_{1i}, \nu_{2i})' = \mathbf{L}\boldsymbol{\psi}_i$, where $\boldsymbol{\psi}_i$ is a vector of independent standard normal random variables.

We can rewrite (1) as

$$L_i = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left\{\prod_{t=1}^{N_i} f(y_{it}, s_{it}|(\nu_{1i}, \nu_{2i})' = \mathbf{L}\boldsymbol{\psi}_i)\right\} \phi(\psi_{1i})\phi(\psi_{2i}) d\psi_{1i} d\psi_{2i}$$

Now the univariate integral can be approximated using Gauss–Hermite quadrature (GHQ). For q-point GHQ, let the abscissa and weight pairs be denoted by (a_k^*, w_k^*), where $k = 1, \ldots, q$. The GHQ approximation is then

$$\int_{-\infty}^{\infty} f(x)\exp(-x^2)\, dx \approx \sum_{k=1}^{q} w_k^* f(a_k^*)$$

Consider a 2-dimensional quadrature grid containing q quadrature points in both dimensions. Let the vector of abscissas $\mathbf{a}_k = (a_{k_1}, a_{k_2})'$ be a point in this grid, and let $\mathbf{w}_k = (w_{k_1}, w_{k_2})'$ be the vector of corresponding weights.

The GHQ approximation to the likelihood for a given panel is

$$L_i = \sum_{k_1=1}^{q}\sum_{k_2=1}^{q}\left[\left\{\prod_{t=1}^{N_i} f(y_{it}, s_{it}|(\nu_{1i}, \nu_{2i})' = \mathbf{L}\mathbf{a}_k)\right\}\left\{\prod_{s=1}^{2} w_{k_s}\right\}\right]$$

Rather than using regular GHQ, we can use mean–variance adaptive GHQ. Fixing the observed variables and model parameters in the integrand of (1), we see the posterior density for $\boldsymbol{\psi}_i$ is proportional to

$$\left\{\prod_{t=1}^{N_i} f(y_{it}, s_{it}|(\nu_{1i}, \nu_{2i})' = \mathbf{L}\boldsymbol{\psi}_i)\right\}\phi(\boldsymbol{\psi}_i)$$

It is reasonable to assume that this posterior density can be approximated by a multivariate normal density with mean vector $\boldsymbol{\mu}_{vi}$ and variance matrix $\boldsymbol{\tau}_{vi}$. Instead of using the prior density of $\boldsymbol{\psi}_i$ as the weighting distribution in the integral, we can use our approximation for the posterior density,

$$L_i = \int_{\Re^2} \frac{\left\{\prod_{t=1}^{N_i} f(y_{it}, s_{it} | (\nu_{1i}, \nu_{2i})' = \mathbf{L}\boldsymbol{\psi}_i)\right\} \phi(\boldsymbol{\psi}_i)}{\phi(\boldsymbol{\psi}_i, \boldsymbol{\mu}_{vi}, \boldsymbol{\tau}_{vi})} \phi(\boldsymbol{\psi}_i, \boldsymbol{\mu}_{vi}, \boldsymbol{\tau}_{vi})\, d\boldsymbol{\psi}_i$$

The likelihood is then approximated by

$$L_i = \sum_{k_1=1}^{q} \cdots \sum_{k_2=1}^{q} \left[\left\{\prod_{t=1}^{N_i} f(y_{it}, s_{it} | (\nu_{1i}, \nu_{2i})' = \mathbf{L}\boldsymbol{\alpha}_k)\right\}\left\{\prod_{s=1}^{2} \omega_{k_s}\right\}\right]$$

where $\boldsymbol{\alpha}_k$ and ω_{k_s} are the adaptive versions of the abscissas and weights after an orthogonalizing transformation, which eliminates posterior covariances between elements of $\boldsymbol{\psi}_i$. The posterior means $\boldsymbol{\mu}_{vi}$ and posterior variances $\boldsymbol{\tau}_{vi}$ are computed iteratively by updating the posterior moments by using the mean–variance adaptive GHQ approximation, starting with a 0 mean vector and identity variance matrix.

The log likelihood for all panels is then

$$\ln L = \sum_{i=1}^{N} \left(\ln \sum_{k_1=1}^{q} \cdots \sum_{k_2=1}^{q} \left[\left\{\prod_{t=1}^{N_i} f(y_{it}, s_{it} | (\nu_{1i}, \nu_{2i})' = \mathbf{L}\boldsymbol{\alpha}_k)\right\}\left\{\prod_{s=1}^{2} \omega_{k_s}\right\}\right]\right)$$

The conditional mean of y_{it} is

$$E(y_{it}|\mathbf{x}_{it}) = \mathbf{x}_{it}\boldsymbol{\beta}$$

`xtheckman` results are obtained using `xteregress`; see *Methods and formulas* of [ERM] **eregress**.

References

Amemiya, T. 1985. *Advanced Econometrics*. Cambridge, MA: Harvard University Press.

Bartus, T., and D. Roodman. 2014. Estimation of multiprocess survival models with cmp. *Stata Journal* 14: 756–777.

Heckman, J. J. 1976. "The common structure of statistical models of truncation, sample selection and limited dependent variables and a simple estimator for such models". In *Annals of Economic and Social Measurement*, edited by S. V. Berg, vol. 5: 475–492. Cambridge, MA: National Bureau of Economic Research.

———. 1979. Sample selection bias as a specification error. *Econometrica* 47: 153–161. https://doi.org/10.2307/1912352.

Honoré, B. E., E. Kyriazidou, and J. L. Powell. 2000. Estimation of tobit-type models with individual specific effects. *Econometric Reviews* 19: 341–366. https://doi.org/10.1080/07474930008800476.

Kyriazidou, E. 1997. Estimation of a panel data sample selection model. *Econometrica* 65: 1335–1364. https://doi.org/10.2307/2171739.

Maddala, G. S. 1983. *Limited-Dependent and Qualitative Variables in Econometrics*. Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9780511810176.

Maddala, G. S., and L.-F. Lee. 1976. "Recursive models with qualitative endogenous variables". In *Annals of Economic and Social Measurement*, edited by S. V. Berg, vol. 5: 525–545. Cambridge, MA: National Bureau of Economic Research.

Rabe-Hesketh, S., A. Skrondal, and A. Pickles. 2002. Multilevel selection models using gllamm. Presented at the 2002 Dutch and German Stata Users Group meeting, Maastricht, The Netherlands, May 23. https://www.stata.com/meeting/2dutch/select.pdf.

Roodman, D. 2011. Fitting fully observed recursive mixed-process models with cmp. *Stata Journal* 11: 159–206.

White, H. L., Jr. 1996. *Estimation, Inference and Specification Analysis*. Cambridge: Cambridge University Press. https://doi.org/10.1017/CCOL0521252806.

Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

———. 2014. Quasi-maximum likelihood estimation and testing for nonlinear models with endogenous explanatory variables. *Journal of Econometrics* 182: 226–234. https://doi.org/10.1016/j.jeconom.2014.04.020.

Also see

[XT] **xtheckman postestimation** — Postestimation tools for xtheckman

[XT] **xteregress** — Extended random-effects linear regression

[XT] **xtreg** — Linear models for panel data

[XT] **xtset** — Declare data to be panel data

[ERM] **eregress** — Extended linear regression

[R] **heckman** — Heckman selection model

[R] **regress** — Linear regression

[U] 20 Estimation and postestimation commands

xtheckman postestimation — Postestimation tools for xtheckman

Postestimation commands   predict   margins   Remarks and examples   Also see

Postestimation commands

The following postestimation commands are available after `xtheckman`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of parameters
`estat ic`	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC, respectively)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
`forecast`	dynamic forecasts and simulations
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
`lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	linear predictions, probabilities, etc.
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of parameters
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

predict

Description for predict

`predict` creates a new variable containing predictions such as linear predictions, probabilities, and expected values.

Menu for predict

Statistics > Postestimation

Syntax for predict

predict [*type*] *newvar* [*if*] [*in*] [, *statistic* nooffset]

predict [*type*] *stub** [*if*] [*in*], scores

statistic	Description
Main	
xb	linear prediction; the default
xbsel	linear prediction for selection equation
pr(*a*,*b*)	$\Pr(y_{it} \mid a < y_{it} < b)$
e(*a*,*b*)	$E(y_{it} \mid a < y_{it} < b)$
ystar(*a*,*b*)	$E(y^*_{it})$, $y^*_{it} = \max\{a, \min(y_{it}, b)\}$
ycond	$E(y_{it} \vert y_{it}$ observed)
psel	$\Pr(y_{it}$ observed)

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

where *a* and *b* may be numbers or variables; *a* missing ($a \geq .$) means $-\infty$, and *b* missing ($b \geq .$) means $+\infty$; see **[U] 12.2.1 Missing values**.

Options for predict

Main

`xb`, the default, calculates the linear prediction $\mathbf{x}_{it}\mathbf{b}$.

`xbsel` calculates the linear prediction for the selection equation.

`pr(`*a*,*b*`)` calculates $\Pr(a < \mathbf{x}_{it}\mathbf{b} + \nu_{1i} + \epsilon_{1it} < b)$, the probability that $y_{it}|\mathbf{x}_{it}$ would be observed in the interval (a, b).

a and *b* may be specified as numbers or variable names; *lb* and *ub* are variable names; `pr(20,30)` calculates $\Pr(20 < \mathbf{x}_{it}\mathbf{b} + \nu_{1i} + \epsilon_{1it} < 30)$; `pr(`*lb*,*ub*`)` calculates $\Pr(lb < \mathbf{x}_{it}\mathbf{b} + \nu_{1i} + \epsilon_{1it} < ub)$; and `pr(20,`*ub*`)` calculates $\Pr(20 < \mathbf{x}_{it}\mathbf{b} + \nu_{1i} + \epsilon_{1it} < ub)$.

a missing ($a \geq .$) means $-\infty$; `pr(.,30)` calculates $\Pr(-\infty < \mathbf{x}_{it}\mathbf{b} + \nu_{1i} + \epsilon_{1it} < 30)$; `pr(`*lb*`,30)` calculates $\Pr(-\infty < \mathbf{x}_{it}\mathbf{b} + \nu_{1i} + \epsilon_{1it} < 30)$ in observations for which $lb \geq .$ and calculates $\Pr(lb < \mathbf{x}_{it}\mathbf{b} + \nu_{1i} + \epsilon_{1it} < 30)$ elsewhere.

b missing ($b \geq .$) means $+\infty$; `pr(20,.)` calculates $\Pr(+\infty > \mathbf{x}_{it}\mathbf{b} + \nu_{1i} + \epsilon_{1it} > 20)$; `pr(20,`*ub*`)` calculates $\Pr(+\infty > \mathbf{x}_{it}\mathbf{b} + \nu_{1i} + \epsilon_{1it} > 20)$ in observations for which $ub \geq .$ and calculates $\Pr(20 < \mathbf{x}_{it}\mathbf{b} + \nu_{1i} + \epsilon_{1it} < ub)$ elsewhere.

`e(`*a*`,`*b*`)` calculates $E(\mathbf{x}_{it}\mathbf{b} + \nu_{1i} + \epsilon_{1it} \mid a < \mathbf{x}_{it}\mathbf{b} + \nu_{1i} + \epsilon_{1it} < b)$, the expected value of $y_{it}|\mathbf{x}_{it}$ conditional on $y_{it}|\mathbf{x}_{it}$ being in the interval (a, b), meaning that $y_{it}|\mathbf{x}_{it}$ is truncated. *a* and *b* are specified as they are for `pr()`.

`ystar(`*a*`,`*b*`)` calculates $E(y^*_{it})$, where $y^*_{it} = a$ if $\mathbf{x}_{it}\mathbf{b} + \nu_{1i} + \epsilon_{1it} \leq a$, $y^*_{it} = b$ if $\mathbf{x}_{it}\mathbf{b} + \nu_{1i} + \epsilon_{1it} \geq b$, and $y^*_{it} = \mathbf{x}_{it}\mathbf{b} + \nu_{1i} + \epsilon_{1it}$ otherwise, meaning that y^*_{it} is not selected. *a* and *b* are specified as they are for `pr()`.

`ycond` calculates the expected value of the dependent variable conditional on the dependent variable being observed, that is, selected; $E(y_{it} \mid y_{it} \text{ observed})$.

`psel` calculates the probability of selection (or being observed): $\Pr(y_{it} \text{ observed}) = \Pr(\mathbf{z}_{it}\boldsymbol{\alpha} + \nu_{2i} + \epsilon_{2it} > 0)$.

`nooffset` is relevant when you specify `offset(`*varname*`)` for `xtheckman`. It modifies the calculations made by `predict` so that they ignore the offset variable; the linear prediction is treated as $\mathbf{x}_{it}\mathbf{b}$ rather than as $\mathbf{x}_{it}\mathbf{b} + \text{offset}_{it}$.

`scores` calculates parameter-level score variables.

margins

Description for margins

`margins` estimates margins of response for linear predictions, probabilities, and expected values.

Menu for margins

Statistics > Postestimation

Syntax for margins

`margins` [*marginlist*] [, *options*]

`margins` [*marginlist*] , `predict(`*statistic* ...`)` [`predict(`*statistic* ...`)` ...] [*options*]

statistic	Description
`xb`	linear prediction; the default
`xbsel`	linear prediction for selection equation
`pr(`*a*`,`*b*`)`	$\Pr(y_{it} \mid a < y_{it} < b)$
`e(`*a*`,`*b*`)`	$E(y_{it} \mid a < y_{it} < b)$
`ystar(`*a*`,`*b*`)`	$E(y^*_{it})$, $y^*_{it} = \max\{a, \min(y_{it}, b)\}$
`ycond`	$E(y_{it} \mid y_{it} \text{ observed})$
`psel`	$\Pr(y_{it} \text{ observed})$

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.

For the full syntax, see [R] **margins**.

Remarks and examples

The default statistic produced by `predict` after `xtheckman` is the expected value of the dependent variable from the underlying distribution of the regression model. See example 1 of [XT] **xtheckman** for an example where `margins` is used to predict the conditional mean.

Also see

[XT] **xtheckman** — Random-effects regression with sample selection

[U] 20 Estimation and postestimation commands

xthtaylor — Hausman–Taylor estimator for error-components model

Description

xthtaylor fits a random-effects model for panel data in which some of the covariates are correlated with the unobserved individual-level random effects. The command implements the Hausman–Taylor estimator by default, but the Amemiya–MaCurdy estimator is available for balanced panels.

Quick start

Hausman–Taylor model of y as a function of time-varying exogenous variable x1, time-invariant binary variable a, and time-varying endogenous variable x2 using xtset data

```
xthtaylor y x1 x2 a, endog(x2)
```

Same as above, and verify that a is the only time-invariant variable in the model

```
xthtaylor y x1 x2 a, endog(x2) constant(a)
```

Add time-invariant x3 as an endogenous covariate, but do not verify that a and x3 are the only time-invariant variables

```
xthtaylor y x1 x2 a x3, endog(x2 x3)
```

Same as above, but use Amemiya–MaCurdy estimator for balanced panels

```
xthtaylor y x1 x2 a x3, endog(x2 x3) am
```

Menu

Statistics > Longitudinal/panel data > Endogenous covariates > Hausman–Taylor regression (RE)

Syntax

`xthtaylor` *depvar indepvars* [*if*] [*in*] [*weight*] , `endog(`*varlist*`)` [*options*]

options	Description
Model	
`noconstant`	suppress constant term
* `endog(`*varlist*`)`	explanatory variables in *indepvars* to be treated as endogenous
`constant(`$varlist_{ti}$`)`	independent variables that are constant within panel
`varying(`$varlist_{tv}$`)`	independent variables that are time varying within panel
`amacurdy`	fit model based on Amemiya and MaCurdy estimator
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `conventional`, `robust`, `cluster` *clustvar*, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`small`	report small-sample statistics

* `endog(`*varlist*`)` is required.

A panel variable must be specified. For `xthtaylor, amacurdy`, a time variable must also be specified. Use `xtset`; see [XT] **xtset**.

depvar, *indepvars*, and all *varlist*s may contain time-series operators; see [U] **11.4.4 Time-series varlists**.

`by`, `collect`, `statsby`, and `xi` are allowed; see [U] **11.1.10 Prefix commands**.

`iweight`s and `fweight`s are allowed unless the `amacurdy` option is specified. Weights must be constant within panel; see [U] **11.1.6 weight**.

See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

`noconstant`; see [R] **Estimation options**.

`endog(`*varlist*`)` specifies that a subset of explanatory variables in *indepvars* be treated as endogenous variables, that is, the explanatory variables that are assumed to be correlated with the unobserved random effect. `endog()` is required.

`constant(`$varlist_{ti}$`)` specifies the subset of variables in *indepvars* that are time invariant, that is, constant within panel. By using this option, you assert not only that the variables specified in $varlist_{ti}$ are time invariant but also that all other variables in *indepvars* are time varying. If this assertion is false, `xthtaylor` does not perform the estimation and will issue an error message. `xthtaylor` automatically detects which variables are time invariant and which are not. However, users may want to check their understanding of the data and specify which variables are time invariant and which are not.

`varying(`$varlist_{tv}$`)` specifies the subset of variables in *indepvars* that are time varying. By using this option, you assert not only that the variables specified in $varlist_{tv}$ are time varying but also that all other variables in *indepvars* are time invariant. If this assertion is false, `xthtaylor` does not perform the estimation and will issue an error message. `xthtaylor` automatically detects which variables are time varying and which are not. However, users may want to check their understanding of the data and specify which variables are time varying and which are not.

`amacurdy` specifies that the Amemiya–MaCurdy estimator be used. This estimator uses extra instruments to gain efficiency at the cost of additional assumptions on the data-generating process. This option may be specified only for samples containing balanced panels, and weights may not be specified. The panels must also have a common initial period.

SE/Robust

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`conventional`), that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster` *clustvar*), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [XT] ***vce_options***.

`vce(conventional)`, the default, uses the conventionally derived variance estimator for this Hausman–Taylor model.

Specifying `vce(robust)` is equivalent to specifying `vce(cluster` *panelvar*`)`; see *xtpoisson, re and the robust VCE estimator* in *Methods and formulas* of [XT] **xtpoisson**.

Reporting

`level(#)`; see [R] **Estimation options**.

`small` specifies that the p-values from the Wald tests in the output and all subsequent Wald tests obtained via `test` use t and F distributions instead of the large-sample normal and χ^2 distributions. By default, the p-values are obtained using the normal and χ^2 distributions.

Remarks and examples

If you have not read [XT] **xt**, please do so.

Consider a random-effects model of the form

$$y_{it} = \mathbf{X}_{1it}\boldsymbol{\beta}_1 + \mathbf{X}_{2it}\boldsymbol{\beta}_2 + \mathbf{Z}_{1i}\boldsymbol{\delta}_1 + \mathbf{Z}_{2i}\boldsymbol{\delta}_2 + \mu_i + \epsilon_{it}$$

where

$\mathbf{X}_{1it}$ is a $1 \times k_1$ vector of observations on exogenous, time-varying variables assumed to be uncorrelated with μ_i and ϵ_{it};

$\mathbf{X}_{2it}$ is a $1 \times k_2$ vector of observations on endogenous, time-varying variables assumed to be (possibly) correlated with μ_i but orthogonal to ϵ_{it};

$\mathbf{Z}_{1i}$ is a $1 \times g_1$ vector of observations on exogenous, time-invariant variables assumed to be uncorrelated with μ_i and ϵ_{it};

$\mathbf{Z}_{2i}$ is a $1 \times g_2$ vector of observations on endogenous, time-invariant variables assumed to be (possibly) correlated μ_i but orthogonal to ϵ_{it};

μ_i is the unobserved, panel-level random effect that is assumed to have zero mean and finite variance σ^2_μ and to be independent and identically distributed (i.i.d.) over the panels;

ϵ_{it} is the idiosyncratic error that is assumed to have zero mean and finite variance σ^2_ϵ and to be i.i.d. over all the observations in the data;

$\boldsymbol{\beta}_1$, $\boldsymbol{\beta}_2$, $\boldsymbol{\delta}_1$, and $\boldsymbol{\delta}_2$ are $k_1 \times 1$, $k_2 \times 1$, $g_1 \times 1$, and $g_2 \times 1$ coefficient vectors, respectively; and

$i = 1, \dots, n$, where n is the number of panels in the sample and, for each i, $t = 1, \dots, T_i$.

Because $\mathbf{X}_{2it}$ and $\mathbf{Z}_{2i}$ may be correlated with μ_i, the simple random-effects estimators—`xtreg, re` and `xtreg, mle`—are generally not consistent for the parameters in this model. Because the within estimator, `xtreg, fe`, removes the μ_i by mean-differencing the data before estimating $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$, it is consistent for these parameters. However, in the process of removing the μ_i, the within estimator also eliminates the $\mathbf{Z}_{1i}$ and the $\mathbf{Z}_{2i}$. Thus it cannot estimate $\boldsymbol{\delta}_1$ nor $\boldsymbol{\delta}_2$. The Hausman–Taylor and Amemiya–MaCurdy estimators implemented in `xthtaylor` are designed to resolve this problem.

The within estimator consistently estimates $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$. Using these estimates, we can obtain the within residuals, called $\widehat{d}_i$. Intermediate, albeit consistent, estimates of $\boldsymbol{\delta}_1$ and $\boldsymbol{\delta}_2$—called $\widehat{\boldsymbol{\delta}}_{1\text{IV}}$ and $\widehat{\boldsymbol{\delta}}_{2\text{IV}}$, respectively—are obtained by regressing the within residuals on $\mathbf{Z}_{1i}$ and $\mathbf{Z}_{2i}$, using $\mathbf{X}_{1it}$ and $\mathbf{Z}_{1i}$ as instruments. The order condition for identification requires that the number of variables in $\mathbf{X}_{1it}$, k_1, be at least as large as the number of elements in $\mathbf{Z}_{2i}$, g_2 and that there be sufficient correlation between the instruments and $\mathbf{Z}_{2i}$ to avoid a weak-instrument problem.

The within estimates of $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ and the intermediate estimates $\widehat{\boldsymbol{\delta}}_{1\text{IV}}$ and $\widehat{\boldsymbol{\delta}}_{2\text{IV}}$ can be used to obtain sets of within and overall residuals. These two sets of residuals can be used to estimate the variance components (see *Methods and formulas* for details).

The estimated variance components can then be used to perform a GLS transform on each of the variables. For what follows, define the general notation $\breve{w}_{it}$ to represent the GLS transform of the variable w_{it}, $\overline{w}_i$ to represent the within-panel mean of w_{it}, and $\widetilde{w}_{it}$ to represent the within transform of w_{it}. With this notational convention, the Hausman–Taylor (1981) estimator of the coefficients of interest can be obtained by the instrumental-variables regression

$$\breve{y}_{it} = \breve{\mathbf{X}}_{1it}\boldsymbol{\beta}_1 + \breve{\mathbf{X}}_{2it}\boldsymbol{\beta}_2 + \breve{\mathbf{Z}}_{1i}\boldsymbol{\delta}_1 + \breve{\mathbf{Z}}_{2i}\boldsymbol{\delta}_2 + \breve{\mu}_i + \breve{\epsilon}_{it} \tag{1}$$

using $\widetilde{\mathbf{X}}_{1it}$, $\widetilde{\mathbf{X}}_{2it}$, $\overline{\mathbf{X}}_{1i}$, $\overline{\mathbf{X}}_{2i}$, and $\mathbf{Z}_{1i}$ as instruments.

For the instruments to be valid, this estimator requires that $\overline{\mathbf{X}}_{1i.}$ and $\mathbf{Z}_{1i}$ be uncorrelated with the random-effect μ_i. More precisely, the instruments are valid when

$$\text{plim}_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} \overline{\mathbf{X}}_{1i.}\mu_i = 0$$

and

$$\text{plim}_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} \mathbf{Z}_{1i}\mu_i = 0$$

Amemiya and MaCurdy (1986) place stricter requirements on the instruments that vary within panels to obtain a more efficient estimator. Specifically, Amemiya and MaCurdy (1986) assume that $\mathbf{X}_{1it}$ is orthogonal to μ_i in every period; that is, $\text{plim}_{n\to\infty} 1/n\sum_{i=1}^{n} \mathbf{X}_{1it}\mu_i = 0$ for $t = 1, \dots, T$. With this restriction, they derive the Amemiya–MaCurdy estimator as the instrumental-variables regression of (1) using instruments $\widetilde{\mathbf{X}}_{1it}$, $\widetilde{\mathbf{X}}_{2it}$, $\mathbf{X}^*_{1it}$, and $\mathbf{Z}_{1i}$. The order condition for the Amemiya–MaCurdy estimator is now $Tk_1 > g_2$. `xthtaylor` uses the Amemiya–MaCurdy estimator when the `amacurdy` option is specified.

Although the estimators implemented in `xthtaylor` and `xtivreg` (see [XT] **xtivreg**) use the method of instrumental variables, each command is designed for different problems. The estimators implemented in `xtivreg` assume that a subset of the explanatory variables in the model are correlated with the idiosyncratic error ϵ_{it}. In contrast, the Hausman–Taylor and Amemiya–MaCurdy estimators that are implemented in `xthtaylor` assume that some of the explanatory variables are correlated with the individual-level random effects, u_i, but that none of the explanatory variables are correlated with the idiosyncratic error, ϵ_{it}.

▷ Example 1

This example replicates the results of Baltagi and Khanti-Akom (1990, table II, column HT) using 595 observations on individuals over 1976–1982 that were extracted from the Panel Study of Income Dynamics (PSID). In the model, the log-transformed wage `lwage` is assumed to be a function of how long the person has worked for a firm, `wks`; binary variables indicating whether a person lives in a large metropolitan area or in the south, `smsa` and `south`; marital status is `ms`; years of education, `ed`; a quadratic of work experience, `exp` and `exp2`; occupation, `occ`; a binary variable indicating employment in a manufacture industry, `ind`; a binary variable indicating that wages are set by a union contract, `union`; a binary variable indicating gender, `fem`; and a binary variable indicating whether the individual is African American, `blk`.

We suspect that the time-varying variables `exp`, `exp2`, `wks`, `ms`, and `union` are all correlated with the unobserved individual random effect. We can inspect these variables to see if they exhibit sufficient within-panel variation to serve as their own instruments.

```
. use https://www.stata-press.com/data/r19/psidextract

. xtsum exp exp2 wks ms union

Variable         |      Mean   Std. dev.       Min        Max |    Observations
-----------------+--------------------------------------------+----------------
exp      overall |  19.85378   10.96637          1         51 |     N =    4165
         between |             10.79018          4         48 |     n =     595
         within  |              2.00024   16.85378   22.85378 |     T =       7
                 |                                            |
exp2     overall |   514.405   496.9962          1       2601 |     N =    4165
         between |             489.0495         20       2308 |     n =     595
         within  |             90.44581    231.405    807.405 |     T =       7
                 |                                            |
wks      overall |  46.81152   5.129098          5         52 |     N =    4165
         between |             3.284016   31.57143   51.57143 |     n =     595
         within  |             3.941881    12.2401   63.66867 |     T =       7
                 |                                            |
ms       overall |  .8144058   .3888256          0          1 |     N =    4165
         between |             .3686109          0          1 |     n =     595
         within  |             .1245274  -.0427371   1.671549 |     T =       7
                 |                                            |
union    overall |  .3639856   .4812023          0          1 |     N =    4165
         between |             .4543848          0          1 |     n =     595
         within  |             .1593351  -.4931573   1.221128 |     T =       7
```

We are also going to assume that the exogenous variables `occ`, `south`, `smsa`, `ind`, `fem`, and `blk` are instruments for the endogenous, time-invariant variable `ed`. The output below indicates that although `fem` appears to be a weak instrument, the remaining instruments are probably sufficiently correlated to identify the coefficient on `ed`. (See Baltagi and Khanti-Akom [1990] for more discussion.)

```
. correlate fem blk occ south smsa ind ed
(obs=4,165)

             |      fem      blk      occ    south     smsa      ind       ed
-------------+---------------------------------------------------------------
         fem |   1.0000
         blk |   0.2086   1.0000
         occ |  -0.0847   0.0837   1.0000
       south |   0.0516   0.1218   0.0413   1.0000
        smsa |   0.1044   0.1154  -0.2018  -0.1350   1.0000
         ind |  -0.1778  -0.0475   0.2260  -0.0769  -0.0689   1.0000
          ed |  -0.0012  -0.1196  -0.6194  -0.1216   0.1843  -0.2365   1.0000
```

We will assume that the correlations are strong enough and proceed with the estimation. The output below gives the Hausman–Taylor estimates for this model.

```
. xthtaylor lwage occ south smsa ind exp exp2 wks ms union fem blk ed,
> endog(exp exp2 wks ms union ed)

Hausman-Taylor estimation                       Number of obs     =      4,165
Group variable: id                              Number of groups  =        595

                                                Obs per group:
                                                              min =          7
                                                              avg =          7
                                                              max =          7

Random effects u_i ~ i.i.d.                     Wald chi2(12)     =    6891.87
                                                Prob > chi2       =     0.0000

------------------------------------------------------------------------------
       lwage | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
TVexogenous  |
         occ |  -.0207047   .0137809    -1.50   0.133    -.0477149    .0063055
       south |   .0074398    .031955     0.23   0.816    -.0551908    .0700705
        smsa |  -.0418334   .0189581    -2.21   0.027    -.0789906   -.0046761
         ind |   .0136039   .0152374     0.89   0.372    -.0162608    .0434686
TVendogenous |
         exp |   .1131328    .002471    45.79   0.000     .1082898    .1179758
        exp2 |  -.0004189   .0000546    -7.67   0.000    -.0005259   -.0003119
         wks |   .0008374   .0005997     1.40   0.163    -.0003381    .0020129
          ms |  -.0298508     .01898    -1.57   0.116    -.0670508    .0073493
       union |   .0327714   .0149084     2.20   0.028     .0035514    .0619914
TIexogenous  |
         fem |  -.1309236    .126659    -1.03   0.301    -.3791707    .1173234
         blk |  -.2857479   .1557019    -1.84   0.066    -.5909179    .0194221
TIendogenous |
          ed |    .137944   .0212485     6.49   0.000     .0962977    .1795902
             |
       _cons |   2.912726   .2836522    10.27   0.000     2.356778    3.468674
-------------+----------------------------------------------------------------
     sigma_u |  .94180304
     sigma_e |  .15180273
         rho |  .97467788   (fraction of variance due to u_i)
------------------------------------------------------------------------------
Note: TV refers to time varying; TI refers to time invariant.
```

The estimated σ_μ and σ_ϵ are 0.9418 and 0.1518, respectively, indicating that a large fraction of the total error variance is attributed to μ_i. The z statistics indicate that several the coefficients may not be significantly different from zero. Whereas the coefficients on the time-invariant variables `fem` and `blk` have relatively large standard errors, the standard error for the coefficient on `ed` is relatively small.

Baltagi and Khanti-Akom (1990) also present evidence that the efficiency gains of the Amemiya–MaCurdy estimator over the Hausman–Taylor estimator are small for these data. This point is especially important given the additional restrictions that the estimator places on the data-generating process. The output below replicates the Baltagi and Khanti-Akom (1990) results from column AM of table II.

```
. xthtaylor lwage occ south smsa ind exp exp2 wks ms union fem blk ed,
> endog(exp exp2 wks ms union ed) amacurdy

Amemiya-MaCurdy estimation                      Number of obs     =      4,165
Group variable: id                              Number of groups  =        595

Time variable: t                                Obs per group:
                                                              min =          7
                                                              avg =          7
                                                              max =          7

Random effects u_i ~ i.i.d.                     Wald chi2(12)     =    6879.20
                                                Prob > chi2       =     0.0000

------------------------------------------------------------------------------
       lwage | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
TVexogenous  |
         occ |  -.0208498   .0137653    -1.51   0.130    -.0478292    .0061297
       south |   .0072818   .0319365     0.23   0.820    -.0553126    .0698761
        smsa |  -.0419507   .0189471    -2.21   0.027    -.0790864   -.0048149
         ind |   .0136289    .015229     0.89   0.371    -.0162194    .0434771
TVendogenous |
         exp |   .1129704   .0024688    45.76   0.000     .1081316    .1178093
        exp2 |  -.0004214   .0000546    -7.72   0.000    -.0005283   -.0003145
         wks |   .0008381   .0005995     1.40   0.162    -.0003368     .002013
          ms |  -.0300894   .0189674    -1.59   0.113    -.0672649    .0070861
       union |   .0324752   .0148939     2.18   0.029     .0032837    .0616667
TIexogenous  |
         fem |   -.132008   .1266039    -1.04   0.297     -.380147    .1161311
         blk |  -.2859004   .1554857    -1.84   0.066    -.5906468    .0188459
TIendogenous |
          ed |   .1372049   .0205695     6.67   0.000     .0968894    .1775205
             |
       _cons |   2.927338   .2751274    10.64   0.000     2.388098    3.466578
-------------+----------------------------------------------------------------
     sigma_u |  .94180304
     sigma_e |  .15180273
         rho |  .97467788   (fraction of variance due to u_i)
------------------------------------------------------------------------------
Note: TV refers to time varying; TI refers to time invariant.
```

◁

❑ Technical note

We mentioned earlier that insufficient correlation between an endogenous variable and the instruments can give rise to a weak-instrument problem. Suppose that we simulate data for a model of the form

$$y = 3 + 3x_{1a} + 3x_{1b} + 3x_2 + 3z_1 + 3z_2 + u_i + e_{it}$$

and purposely construct the instruments so that they exhibit little correlation with the endogenous variable z_2.

```
. use https://www.stata-press.com/data/r19/xthtaylor1
. correlate ui z1 z2 x1a x1b x2 eit
(obs=10,000)
```

	ui	z1	z2	x1a	x1b	x2	eit
ui	1.0000						
z1	0.0268	1.0000					
z2	0.8777	0.0286	1.0000				
x1a	-0.0145	0.0065	-0.0034	1.0000			
x1b	0.0026	0.0079	0.0038	-0.0030	1.0000		
x2	0.8765	0.0191	0.7671	-0.0192	0.0037	1.0000	
eit	0.0060	-0.0198	0.0123	-0.0100	-0.0138	0.0092	1.0000

In the output below, weak instruments have serious consequences on the estimates produced by `xthtaylor`. The estimate of the coefficient on `z2` is three times larger than its true value, and its standard error is rather large. Without sufficient correlation between the endogenous variable and its instruments in a given sample, there is insufficient information for identifying the parameter. Also, given the results of Stock, Wright, and Yogo (2002), weak instruments will cause serious size distortions in any tests performed.

```
. xthtaylor yit x1a x1b x2 z1 z2, endog(x2 z2)
Hausman-Taylor estimation                       Number of obs     =     10,000
Group variable: id                              Number of groups  =      1,000
                                                Obs per group:
                                                              min =         10
                                                              avg =         10
                                                              max =         10
Random effects u_i ~ i.i.d.                     Wald chi2(5)      =   24172.91
                                                Prob > chi2       =     0.0000
```

yit	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
TVexogenous						
x1a	2.959736	.0330233	89.63	0.000	2.895011	3.02446
x1b	2.953891	.0333051	88.69	0.000	2.888614	3.019168
TVendogenous						
x2	3.022685	.033085	91.36	0.000	2.957839	3.08753
TIexogenous						
z1	2.709179	.587031	4.62	0.000	1.55862	3.859739
TIendogenous						
z2	9.525973	8.572966	1.11	0.266	-7.276732	26.32868
_cons	2.837072	.4276595	6.63	0.000	1.998875	3.675269
sigma_u	8.729479					
sigma_e	3.1657492					
rho	.88377062	(fraction of variance due to u_i)				

```
Note: TV refers to time varying; TI refers to time invariant.
```

❑

▷ Example 2

Now let's consider why we might want to specify the `constant(`*varlist*$_{ti}$`)` option. For this example, we will use simulated data. In the output below, we fit a model over the full sample. Note the placement in the output of the coefficient on the exogenous variable `x1c`.

```
. use https://www.stata-press.com/data/r19/xthtaylor2
. xthtaylor yit x1a x1b x1c x2 z1 z2, endog(x2 z2)
Hausman-Taylor estimation                        Number of obs     =     10,000
Group variable: id                               Number of groups  =      1,000

                                                 Obs per group:
                                                              min =         10
                                                              avg =         10
                                                              max =         10

Random effects u_i ~ i.i.d.                      Wald chi2(6)      =   10341.63
                                                 Prob > chi2       =     0.0000

------------------------------------------------------------------------------
         yit | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
TVexogenous  |
         x1a |   3.023647   .0570274    53.02   0.000     2.911875    3.135418
         x1b |   2.966666   .0572659    51.81   0.000     2.854427    3.078905
         x1c |   .2355318    .123502     1.91   0.057    -.0065276    .4775912
TVendogenous |
          x2 |   14.17476   3.128385     4.53   0.000     8.043234    20.30628
TIexogenous  |
          z1 |   1.741709   .4280022     4.07   0.000     .9028398    2.580578
TIendogenous |
          z2 |   7.983849   .6970903    11.45   0.000     6.617577    9.350121
             |
       _cons |   2.146038   .3794179     5.66   0.000     1.402393    2.889684
-------------+----------------------------------------------------------------
     sigma_u |  5.6787791
     sigma_e |  3.1806188
         rho |  .76120931   (fraction of variance due to u_i)
------------------------------------------------------------------------------
Note: TV refers to time varying; TI refers to time invariant.
```

Now suppose that we want to fit the model using only the first eight periods. Below, `x1c` now appears under the `TIexogenous` heading rather than the `TVexogenous` heading because `x1c` is time invariant in the subsample defined by `t<9`.

```
. xthtaylor yit x1a x1b x1c x2 z1 z2 if t<9, endog(x2 z2)

Hausman-Taylor estimation                       Number of obs     =      8,000
Group variable: id                              Number of groups  =      1,000

                                                Obs per group:
                                                              min =          8
                                                              avg =          8
                                                              max =          8

Random effects u_i ~ i.i.d.                     Wald chi2(6)      =   15354.87
                                                Prob > chi2       =     0.0000

------------------------------------------------------------------------------
         yit | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
TVexogenous  |
         x1a |   3.051966   .0367026    83.15   0.000      2.98003    3.123901
         x1b |   2.967822   .0368144    80.62   0.000     2.895667    3.039977
TVendogenous |
          x2 |   .7361217   3.199764     0.23   0.818      -5.5353    7.007543
TIexogenous  |
         x1c |   3.215907   .5657191     5.68   0.000     2.107118    4.324696
          z1 |   3.347644   .5819756     5.75   0.000     2.206992    4.488295
TIendogenous |
          z2 |   2.010578   1.143982     1.76   0.079     -.231586    4.252742
             |
       _cons |   3.257004   .5295828     6.15   0.000     2.219041    4.294967
-------------+----------------------------------------------------------------
     sigma_u |  15.445594
     sigma_e |   3.175083
         rho |  .95945606   (fraction of variance due to u_i)
------------------------------------------------------------------------------
Note: TV refers to time varying; TI refers to time invariant.
```

To prevent a variable from becoming time invariant, you can use either `constant(`*varlist*$_{\text{ti}}$`)` or `varying(`*varlist*$_{\text{tv}}$`)`. `constant(`*varlist*$_{\text{ti}}$`)` specifies the subset of variables in *varlist* that are time invariant and requires the remaining variables in *varlist* to be time varying. If you specify `constant(`*varlist*$_{\text{ti}}$`)` and any of the variables contained in *varlist*$_{\text{ti}}$ are time varying, or if any of the variables not contained in *varlist*$_{\text{ti}}$ are time invariant, `xthtaylor` will not perform the estimation and will issue an error message.

```
. xthtaylor yit x1a x1b x1c x2 z1 z2 if t<9, endog(x2 z2) constant(z1 z2)
x1c not included in constant().
r(198);
```

The same thing happens when you use the `varying(`*varlist*$_{\text{tv}}$`)` option.

◁

Stored results

`xthtaylor` stores the following in `e()`:

Scalars	
`e(N)`	number of observations
`e(N_g)`	number of groups
`e(df_m)`	model degrees of freedom
`e(df_r)`	residual degrees of freedom (`small` only)
`e(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(g_max)`	largest group size
`e(Tcon)`	`1` if panels balanced, `0` otherwise
`e(N_clust)`	number of clusters
`e(sigma_u)`	panel-level standard deviation
`e(sigma_e)`	standard deviation of ϵ_{it}
`e(chi2)`	χ^2
`e(rho)`	ρ
`e(F)`	model F (`small` only)
`e(Tbar)`	harmonic mean of group sizes
`e(rank)`	rank of `e(V)`
Macros	
`e(cmd)`	`xthtaylor`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(ivar)`	variable denoting groups
`e(tvar)`	variable denoting time within groups, `amacurdy` only
`e(TVexogenous)`	exogenous time-varying variables
`e(TIexogenous)`	exogenous time-invariant variables
`e(TVendogenous)`	endogenous time-varying variables
`e(TIendogenous)`	endogenous time-invariant variables
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(title)`	`Hausman-Taylor` or `Amemiya-MaCurdy`
`e(clustvar)`	name of cluster variable
`e(chi2type)`	`Wald`; type of model χ^2 test
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. err.
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
Matrices	
`e(b)`	coefficient vector
`e(V)`	variance–covariance matrix of the estimators
`e(V_modelbased)`	model-based variance
Functions	
`e(sample)`	marks estimation sample

Methods and formulas

Consider an error-components model of the form

$$y_{it} = \mathbf{X}_{1it}\boldsymbol{\beta}_1 + \mathbf{X}_{2it}\boldsymbol{\beta}_2 + \mathbf{Z}_{1i}\boldsymbol{\delta}_1 + \mathbf{Z}_{2i}\boldsymbol{\delta}_2 + \mu_i + \epsilon_{it} \tag{2}$$

for $i = 1, \dots, n$ and, for each i, $t = 1, \dots, T_i$, of which T_i periods are observed; n is the number of panels in the sample. The covariates in $\mathbf{X}$ are time varying, and the covariates in $\mathbf{Z}$ are time invariant. Both $\mathbf{X}$ and $\mathbf{Z}$ are decomposed into two parts. The covariates in $\mathbf{X}_1$ and $\mathbf{Z}_1$ are assumed to be uncorrelated with μ_i and e_{it}, whereas the covariates in $\mathbf{X}_2$ and $\mathbf{Z}_2$ are allowed to be correlated with μ_i but not with ϵ_{it}. Hausman and Taylor (1981) suggest an instrumental-variable estimator for this model.

For some variable w, the within transformation of w is defined as

$$\widetilde{w}_{it} = w_{it} - \overline{w}_{i.} \qquad \overline{w}_{i.} = \frac{1}{T_i}\sum_{t=1}^{T_i} w_{it}$$

Because the within estimator removes $\mathbf{Z}$, the within transformation reduces the model to

$$\tilde{y}_{it} = \widetilde{\mathbf{X}}_{1it}\boldsymbol{\beta}_1 + \tilde{\mathbf{X}}_{2it}\boldsymbol{\beta}_2 + \tilde{\epsilon}_{it}$$

The within estimators $\widehat{\beta}_{1w}$ and $\widehat{\beta}_{2w}$ are consistent for $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$, but they may not be efficient. Also, note that the within estimator cannot estimate $\boldsymbol{\delta}_1$ and $\boldsymbol{\delta}_2$.

From the within estimator, we can obtain an estimate of the idiosyncratic error component, σ^2_ϵ, as

$$\widehat{\sigma}^2_\epsilon = \frac{\text{RSS}}{N - n}$$

where RSS is the residual sum of squares from the within regression and N is the total number of observations in the sample.

Using the results of the within estimation, we can define

$$\overline{d}_{it} = \overline{y}_{it} - \overline{X}_{1it}\widehat{\beta}_{1w} - \overline{X}_{2it}\widehat{\beta}_{2w}$$

where $\overline{y}_{it}$, $\overline{X}_{1it}$, and $\overline{X}_{2it}$ contain the panel level means of these variables in all observations.

Regressing $\overline{d}_{it}$ on $\mathbf{Z}_1$ and $\mathbf{Z}_2$, using $\mathbf{X}_1$ and $\mathbf{Z}_1$ as instruments, provides intermediate, consistent estimates of $\boldsymbol{\delta}_1$ and $\boldsymbol{\delta}_2$, which we will call $\widehat{\boldsymbol{\delta}}_{1\text{IV}}$ and $\widehat{\boldsymbol{\delta}}_{2\text{IV}}$.

Using the within estimates, $\widehat{\boldsymbol{\delta}}_{1\text{IV}}$, and $\widehat{\boldsymbol{\delta}}_{2\text{IV}}$, we can obtain an estimate of the variance of the random effect, σ^2_μ. First, let

$$\widehat{e}_{it} = \left(y_{it} - \mathbf{X}_{1it}\widehat{\boldsymbol{\beta}}_{1w} - \mathbf{X}_{2it}\widehat{\boldsymbol{\beta}}_{2w} - \mathbf{Z}_{1it}\widehat{\boldsymbol{\delta}}_{1\text{IV}} - \mathbf{Z}_{2it}\widehat{\boldsymbol{\delta}}_{2\text{IV}}\right)$$

Then define

$$s^2 = \frac{1}{N}\sum_{i=1}^{n}\sum_{t=1}^{T_i}\left(\frac{1}{T_i}\sum_{t=1}^{T_i}\widehat{e}_{it}\right)^2$$

Hausman and Taylor (1981) showed that, for balanced panels,

$$\text{plim}_{n\to\infty} s^2 = T\sigma^2_\mu + \sigma^2_\epsilon$$

For unbalanced panels,

$$\operatorname{plim}_{n\to\infty} s^2 = \overline{T}\sigma_\mu^2 + \sigma_\epsilon^2$$

where

$$\overline{T} = \frac{n}{\sum_{i=1}^{n} \frac{1}{T_i}}$$

After we plug in $\widehat{\sigma}_\epsilon^2$, our consistent estimate for σ_ϵ^2, a little algebra suggests the estimate

$$\widehat{\sigma}_\mu^2 = (s^2 - \widehat{\sigma}_\epsilon^2)(\overline{T})^{-1}$$

Define $\widehat{\theta}_i$ as

$$\widehat{\theta}_i = 1 - \left(\frac{\widehat{\sigma}_\epsilon^2}{\widehat{\sigma}_\epsilon^2 + T_i \widehat{\sigma}_\mu^2} \right)^{\frac{1}{2}}$$

With $\widehat{\theta_i}$ in hand, we can perform the standard random-effects GLS transform on each of the variables. The transform is given by

$$w_{it}^* = w_{it} - \widehat{\theta_i}\overline{w}_{i.}$$

where $\overline{w}_{i.}$ is the within-panel mean.

We can then obtain the Hausman–Taylor estimates of the coefficients in (2) and the conventional VCE by fitting an instrumental-variables regression of the GLS-transformed y_{it}^* on $\mathbf{X}_{it}^*$ and $\mathbf{Z}_{it}^*$, with instruments $\widetilde{\mathbf{X}}_{it}$, $\overline{\mathbf{X}}_{1i.}$, and $\mathbf{Z}_{1i}$.

We can obtain Amemiya–MaCurdy estimates of the coefficients in (2) and the conventional VCE by fitting an instrumental-variables regression of the GLS-transformed y_{it}^* on $\mathbf{X}_{it}^*$ and $\mathbf{Z}_{it}^*$, using $\widetilde{\mathbf{X}}_{it}$, $\breve{\mathbf{X}}_{1it}$, and $\mathbf{Z}_{1i}$ as instruments, where $\breve{\mathbf{X}}_{1it} = \mathbf{X}_{1i1}, \mathbf{X}_{1i2}, \ldots, \mathbf{X}_{1iT_i}$. The order condition for the Amemiya–MaCurdy estimator is $Tk_1 > g_2$, and this estimator is available only for balanced panels.

References

Amemiya, T., and T. E. MaCurdy. 1986. Instrumental-variable estimation of an error-components model. *Econometrica* 54: 869–880. https://doi.org/10.2307/1912840.

Baltagi, B. H. 2009. *A Companion to Econometric Analysis of Panel Data*. Chichester, UK: Wiley.

———. 2013. *Econometric Analysis of Panel Data*. 5th ed. Chichester, UK: Wiley.

Baltagi, B. H., and S. Khanti-Akom. 1990. On efficient estimation with panel data: An empirical comparison of instrumental variables estimators. *Journal of Applied Econometrics* 5: 401–406. https://doi.org/10.1002/jae.3950050408.

Hausman, J. A., and W. E. Taylor. 1981. Panel data and unobservable individual effects. *Econometrica* 49: 1377–1398. https://doi.org/10.2307/1911406.

Stock, J. H., J. H. Wright, and M. Yogo. 2002. A survey of weak instruments and weak identification in generalized method of moments. *Journal of Business and Economic Statistics* 20: 518–529. https://doi.org/10.1198/073500102288618658.

Also see

[XT] **xthtaylor postestimation** — Postestimation tools for xthtaylor

[XT] **xtivreg** — Instrumental variables and two-stage least squares for panel-data models

[XT] **xtreg** — Linear models for panel data

[XT] **xtset** — Declare data to be panel data

[U] 20 Estimation and postestimation commands

xthtaylor postestimation — Postestimation tools for xthtaylor

Postestimation commands predict margins Remarks and examples
References Also see

Postestimation commands

The following postestimation commands are available after `xthtaylor`:

Command	Description
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
`forecast`	dynamic forecasts and simulations
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	predictions and their SEs, residuals, etc.
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

predict

Description for predict

`predict` creates a new variable containing predictions such as fitted values, standard errors, combined residuals, predictions, random-error components, and idiosyncratic error components.

Menu for predict

Statistics > Postestimation

Syntax for predict

`predict` [*type*] *newvar* [*if*] [*in*] [, *statistic*]

statistic	Description
Main	
`xb`	$\mathbf{X}_{it}\widehat{\boldsymbol{\beta}} + \mathbf{Z}_i\widehat{\boldsymbol{\delta}}$, fitted values; the default
`stdp`	standard error of the fitted values
`ue`	$\widehat{\mu}_i + \widehat{\epsilon}_{it}$, the combined residual
* `xbu`	$\mathbf{X}_{it}\widehat{\boldsymbol{\beta}} + \mathbf{Z}_i\widehat{\boldsymbol{\delta}} + \widehat{\mu}_i$, prediction including effect
* `u`	$\widehat{\mu}_i$, the random-error component
* `e`	$\widehat{\epsilon}_{it}$, prediction of the idiosyncratic error component

Unstarred statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample. Starred statistics are calculated only for the estimation sample, even when `if e(sample)` is not specified.

Options for predict

Main

`xb`, the default, calculates the linear prediction, that is, $\mathbf{X}_{it}\widehat{\boldsymbol{\beta}} + \mathbf{Z}_{it}\widehat{\boldsymbol{\delta}}$.

`stdp` calculates the standard error of the linear prediction.

`ue` calculates the prediction of $\widehat{\mu}_i + \widehat{\epsilon}_{it}$.

`xbu` calculates the prediction of $\mathbf{X}_{it}\widehat{\boldsymbol{\beta}} + \mathbf{Z}_{it}\widehat{\boldsymbol{\delta}} + \widehat{\nu}_i$, the prediction including the random effect.

`u` calculates the prediction of $\widehat{\mu}_i$, the estimated random effect.

`e` calculates the prediction of $\widehat{\epsilon}_{it}$.

margins

Description for margins

`margins` estimates margins of response for fitted values.

Menu for margins

Statistics > Postestimation

Syntax for margins

`margins` [*marginlist*] [, *options*]

`margins` [*marginlist*] , `predict(`*statistic* ...`)` [*options*]

statistic	Description
`xb`	$\mathbf{X}_{it}\widehat{\boldsymbol{\beta}} + \mathbf{Z}_i\widehat{\boldsymbol{\delta}}$, fitted values; the default
`stdp`	not allowed with `margins`
`ue`	not allowed with `margins`
`xbu`	not allowed with `margins`
`u`	not allowed with `margins`
`e`	not allowed with `margins`

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.

For the full syntax, see [R] **margins**.

Remarks and examples

▷ Example 1

Continuing with example 1 of [XT] **xthtaylor**, we use `hausman` to test whether we should use the Hausman–Taylor estimator instead of the fixed-effects estimator. We follow the empirical illustration in Baltagi (2013, sec. 7.5), but we fit the model without including the `exp2` and `wks` variables.

We first fit the model with `xthtaylor` and then with `xtreg, fe`:

```
. use https://www.stata-press.com/data/r19/psidextract
. xthtaylor lwage occ south smsa ind exp ms union fem blk ed,
> endog(exp ms union ed)
  (output omitted)
. estimates store eq_ht
. xtreg lwage occ south smsa ind exp ms union fem blk ed, fe
  (output omitted)
. estimates store eq_fe
```

We can now use `hausman` to compare the two estimators, but we need to specify the `df()` to indicate the degrees of freedom for the χ^2 statistic, which would be determined by the overidentifying restrictions in the Hausman–Taylor estimation. In this case, there are three degrees of freedom because there are four time-varying exogenous variables (`occ`, `south`, `smsa`, `ind`) that can be used as instruments for only one time-invariant endogenous variable (`ed`).

```
. hausman eq_fe eq_ht, df(3)

                 ---- Coefficients ----
             |      (b)          (B)            (b-B)     sqrt(diag(V_b-V_B))
             |     eq_fe        eq_ht        Difference       Std. err.
-------------+----------------------------------------------------------------
         occ |   -.0239323    -.0231694      -.0007629        .0002395
       south |   -.0037282     .0062699      -.0099982        .0124188
        smsa |   -.0436251    -.0433518      -.0002733        .0042296
         ind |     .021184     .0156376       .0055465        .0025159
         exp |    .0965738     .0964748       .0000991         .000063
          ms |   -.0299908    -.0300703       .0000795         .000321
       union |    .0349156     .0348494       .0000662        .0006336
------------------------------------------------------------------------------
                         b = Consistent under H0 and Ha; obtained from xtreg.
          B = Inconsistent under Ha, efficient under H0; obtained from xthtaylor.

Test of H0: Difference in coefficients not systematic

    chi2(3) = (b-B)'[(V_b-V_B)^(-1)](b-B)
            =   5.22
Prob > chi2 = 0.1567
(V_b-V_B is not positive definite)
```

The p-value for the test provides evidence favoring the null hypothesis; therefore, in this case, the Hausman–Taylor estimation is adequate.

Notice that the variance–covariance matrix for the difference (b-B) is not positive definite. As Greene (2012, 237) points out, this kind of result is due to finite-sample conditions. He also states that Hausman considers it preferable to take the test statistic as zero and, therefore, not to reject the null hypothesis.

◁

▷ Example 2

We now want to determine whether the Amemiya–MaCurdy estimator produces significant efficiency gains with respect to the Hausman–Taylor estimator. We refit the two models, and we use the Hausman test again:

```
. use https://www.stata-press.com/data/r19/psidextract

. xthtaylor lwage occ south smsa ind exp ms union fem blk ed,
> endog(exp ms union ed)
  (output omitted)

. estimates store eq_ht

. xthtaylor lwage occ south smsa ind exp ms union fem blk ed,
> endog(exp ms union ed) amacurdy
  (output omitted)

. estimates store eq_am
```

```
. hausman eq_ht eq_am

                 ---- Coefficients ----
             |      (b)          (B)            (b-B)     sqrt(diag(V_b-V_B))
             |     eq_ht        eq_am        Difference       Std. err.
-------------+----------------------------------------------------------------
         occ |   -.0231694     -.023354        .0001846        .0006485
       south |    .0062699     .0060857        .0001842        .0010641
        smsa |   -.0433518    -.0434638        .0001121        .0006297
         ind |    .0156376     .0156602       -.0000226         .000492
         exp |    .0964748     .0962147          .00026        .0000694
          ms |   -.0300703    -.0303139        .0002436        .0006735
       union |    .0348494     .0345742        .0002752        .0006471
         fem |   -.1277756    -.1287857        .0010101        .0036717
         blk |   -.2911574     -.291645        .0004876        .0082831
          ed |    .1390257     .1380699        .0009558         .005436
------------------------------------------------------------------------------
                       b = Consistent under H0 and Ha; obtained from xthtaylor.
        B = Inconsistent under Ha, efficient under H0; obtained from xthtaylor.

Test of H0: Difference in coefficients not systematic

   chi2(10) = (b-B)'[(V_b-V_B)^(-1)](b-B)
            =  14.42
Prob > chi2 = 0.1548
```

The result indicates that we should use the more efficient estimation produced by the Amemiya–MaCurdy estimator.

◁

References

Baltagi, B. H. 2013. *Econometric Analysis of Panel Data*. 5th ed. Chichester, UK: Wiley.

Greene, W. H. 2012. *Econometric Analysis*. 7th ed. Upper Saddle River, NJ: Prentice Hall.

Also see

[XT] **xthtaylor** — Hausman–Taylor estimator for error-components model

[U] 20 Estimation and postestimation commands

xtintreg — Random-effects interval-data regression model

Description

xtintreg fits a random-effects regression model in which the dependent variable may be measured as point data, interval data, left-censored data, or right-censored data. The dependent variable must be specified using two *depvar*s that indicate how the dependent variable was measured. The user can request that a likelihood-ratio test comparing the panel interval regression model with the pooled model be conducted at estimation time.

Quick start

Regression on x of an interval-measured dependent variable with lower endpoint y_lower and upper endpoint y_upper using xtset data

    xtintreg y_lower y_upper x

Add indicators for levels of categorical variable a as covariates

    xtintreg y_lower y_upper x i.a

Perform likelihood-ratio test against pooled model

    xtintreg y_lower y_upper x i.a, intreg

Menu

Statistics > Longitudinal/panel data > Censored outcomes > Interval regression (RE)

Syntax

`xtintreg` *depvar*$_{\text{lower}}$ *depvar*$_{\text{upper}}$ [*indepvars*] [*if*] [*in*] [*weight*] [, *options*]

The values in *depvar*$_{\text{lower}}$ and *depvar*$_{\text{upper}}$ should have the following form:

Type of data		*depvar*$_{\text{lower}}$	*depvar*$_{\text{upper}}$
point data	$a = [\,a, a\,]$	a	a
interval data	$[\,a, b\,]$	a	b
left-censored data	$(\,-\infty, b\,]$	.	b
right-censored data	$[\,a, +\infty\,)$	a	.
missing		.	.

options	Description
Model	
`noconstant`	suppress constant term
`offset(`*varname*`)`	include *varname* in model with coefficient constrained to 1
`constraints(`*constraints*`)`	apply specified linear constraints
SE	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`lrmodel`	perform the likelihood-ratio model test instead of the default Wald test
`intreg`	perform likelihood-ratio test against pooled model
`nocnsreport`	do not display constraints
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Integration	
`intmethod(`*intmethod*`)`	integration method; *intmethod* may be `mvaghermite` (the default) or `ghermite`
`intpoints(#)`	use # quadrature points; default is `intpoints(12)`
Maximization	
maximize_options	control the maximization process; see [R] **Maximize**
`collinear`	keep collinear variables
`coeflegend`	display legend instead of statistics

A panel variable must be specified; use `xtset`; see [XT] **xtset**.

indepvars may contain factor variables; see [U] **11.4.3 Factor variables**.

depvar$_{\text{lower}}$, *depvar*$_{\text{upper}}$, and *indepvars* may contain time-series operators; see [U] **11.4.4 Time-series varlists**.

`by`, `collect`, and `statsby` are allowed; see [U] **11.1.10 Prefix commands**.

`iweight`s are allowed; see [U] **11.1.6 weight**. Weights must be constant within panel.

`collinear` and `coeflegend` do not appear in the dialog box.

See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

noconstant, offset(*varname*), constraints(*constraints*); see [R] **Estimation options**.

SE

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (oim) and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] ***vce_options***.

Reporting

level(#), lrmodel; see [R] **Estimation options**.

intreg specifies that a likelihood-ratio test comparing the random-effects model with the pooled (intreg) model be included in the output.

nocnsreport; see [R] **Estimation options**.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Integration

intmethod(*intmethod*), intpoints(#); see [R] **Estimation options**.

Maximization

maximize_options: difficult, technique(*algorithm_spec*), iterate(#), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), nonrtolerance, and from(*init_specs*); see [R] **Maximize**. These options are seldom used.

The following options are available with xtintreg but are not shown in the dialog box:

collinear, coeflegend; see [R] **Estimation options**.

Remarks and examples

xtintreg may be used to fit a random-effects interval regression model. Consider the linear regression model with panel-level random effects

$$y_{it} = \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i + \epsilon_{it}$$

for $i = 1, \dots, n$ panels, where $t = 1, \dots, n_i$. The random effects, ν_i, are i.i.d., $N(0, \sigma_\nu^2)$, and ϵ_{it} are i.i.d., $N(0, \sigma_\epsilon^2)$ independently of ν_i. The observed data consist of the couples, (y_{1it}, y_{2it}), such that all that is known is that $y_{1it} \le y_{it} \le y_{2it}$, where y_{1it} is possibly $-\infty$ and y_{2it} is possibly $+\infty$.

▷ Example 1: Random-effects interval regression

We begin with the nlswork dataset described in [XT] **xt** and create two fictional dependent variables, where the wages are instead reported sometimes as ranges. The wages have been adjusted to 1988 dollars and have further been recoded such that some of the observations are known exactly, some are left-censored, some are right-censored, and some are known only in an interval.

We wish to fit a random-effects interval regression model of adjusted (log) wages. We specify the `intreg` option to test our random-effects model against our pooled estimator.

```
. use https://www.stata-press.com/data/r19/nlswork5
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)
. xtintreg ln_wage1 ln_wage2 i.union age grade not_smsa south##c.year, intreg
```

(output omitted)

```
Random-effects interval regression               Number of obs     =  19,224
                                                       Uncensored =   4,810
                                                    Left-censored =   4,781
                                                   Right-censored =   4,848
                                                   Interval-cens. =   4,785

Group variable: idcode                           Number of groups  =   4,148
Random effects u_i ~ Gaussian                    Obs per group:
                                                              min =       1
                                                              avg =     4.6
                                                              max =      12

Integration method: mvaghermite                  Integration pts.  =      12

                                                 Wald chi2(7)      = 2461.69
Log likelihood = -23260.672                      Prob > chi2       =  0.0000

------------------------------------------------------------------------------
             | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
     1.union |   .1229681   .0092943    13.23   0.000     .1047516    .1411846
         age |   .0096333      .0019     5.07   0.000     .0059094    .0133572
       grade |   .0756045   .0023828    31.73   0.000     .0709343    .0802747
    not_smsa |  -.1481304    .011433   -12.96   0.000    -.1705387   -.1257221
     1.south |  -.3586443   .0977512    -3.67   0.000    -.5502331   -.1670555
        year |   .0029219   .0020353     1.44   0.151    -.0010671    .0069109
             |
south#c.year |
           1 |   .0032699   .0012076     2.71   0.007      .000903    .0056368
             |
       _cons |   .2747391   .1141328     2.41   0.016     .0510429    .4984352
-------------+----------------------------------------------------------------
    /sigma_u |   .3044775   .0052644    57.84   0.000     .2941594    .3147956
    /sigma_e |   .3516248     .00307   114.54   0.000     .3456078    .3576418
-------------+----------------------------------------------------------------
         rho |   .4285095   .0101261                      .4087613    .4484385
------------------------------------------------------------------------------
LR test of sigma_u=0: chibar2(01) = 2683.77           Prob >= chibar2 = 0.000
```

The results from an interval regression can be interpreted as we would those from a linear regression. Because the dependent variable is log transformed, the coefficients can be interpreted in terms of a percentage change. We see, for example, that on average, union members make 12.3% more than nonunion members.

The output also includes the overall and panel-level variance components (labeled `sigma_e` and `sigma_u`, respectively) together with ρ (labeled `rho`),

$$\rho = \frac{\sigma_\nu^2}{\sigma_\epsilon^2 + \sigma_\nu^2}$$

which is the proportion of the total variance contributed by the panel-level variance component.

When rho is zero, the panel-level variance component is unimportant, and the panel estimator is not different from the pooled estimator. A likelihood-ratio test of this is included at the bottom of the output. This test formally compares the pooled estimator (interval regression) with the panel estimator. In this case, we reject the null hypothesis that there are no panel-level effects.

◁

❑ Technical note

The random-effects model is calculated using quadrature, which is an approximation whose accuracy depends partially on the number of integration points used. We can use the quadchk command to see if changing the number of integration points affects the results. If the results change, the quadrature approximation is not accurate given the number of integration points. Try increasing the number of integration points using the intpoints() option and run quadchk again. Do not attempt to interpret the results of estimates when the coefficients reported by quadchk differ substantially. See [XT] **quadchk** for details and [XT] **xtprobit** for an example.

Because the xtintreg likelihood function is calculated by Gauss–Hermite quadrature, on large problems the computations can be slow. Computation time is roughly proportional to the number of points used for the quadrature.

❑

Stored results

xtintreg stores the following in e():

Scalars

e(N)	number of observations
e(N_g)	number of groups
e(N_unc)	number of uncensored observations
e(N_lc)	number of left-censored observations
e(N_rc)	number of right-censored observations
e(N_int)	number of interval observations
e(k)	number of parameters
e(k_aux)	number of auxiliary parameters
e(k_eq)	number of equations in e(b)
e(k_eq_model)	number of equations in overall model test
e(k_dv)	number of dependent variables
e(df_m)	model degrees of freedom
e(ll)	log likelihood
e(ll_0)	log likelihood, constant-only model
e(ll_c)	log likelihood, comparison model
e(chi2)	χ^2
e(chi2_c)	χ^2 for comparison test
e(rho)	ρ
e(sigma_u)	panel-level standard deviation
e(sigma_e)	standard deviation of ϵ_{it}
e(n_quad)	number of quadrature points
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(p)	p-value for model test
e(rank)	rank of e(V)
e(rank0)	rank of e(V) for constant-only model
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros

`e(cmd)`	`xtintreg`
`e(cmdline)`	command as typed
`e(depvar)`	names of dependent variables
`e(ivar)`	variable denoting groups
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(title)`	title in estimation output
`e(offset)`	offset
`e(chi2type)`	`Wald` or `LR`; type of model χ^2 test
`e(chi2_ct)`	`Wald` or `LR`; type of model χ^2 test corresponding to `e(chi2_c)`
`e(vce)`	*vcetype* specified in `vce()`
`e(intmethod)`	integration method
`e(distrib)`	`Gaussian`; the distribution of the random effect
`e(opt)`	type of optimization
`e(which)`	`max` or `min`; whether optimizer is to perform maximization or minimization
`e(ml_method)`	type of `ml` method
`e(user)`	name of likelihood-evaluator program
`e(technique)`	maximization technique
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(marginsok)`	predictions allowed by `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`

Matrices

`e(b)`	coefficient vector
`e(Cns)`	constraints matrix
`e(ilog)`	iteration log
`e(gradient)`	gradient vector
`e(V)`	variance–covariance matrix of the estimators

Functions

`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

Assuming a normal distribution, $N(0, \sigma_\nu^2)$, for the random effects ν_i, we have the joint (unconditional of ν_i) density of the observed data for the ith panel

$$f\left\{(y_{1i1}, y_{2i1}), \dots, (y_{1in_i}, y_{2in_i}) | \mathbf{x}_{1i}, \dots, \mathbf{x}_{in_i}\right\} = \int_{-\infty}^{\infty} \frac{e^{-\nu_i^2/2\sigma_\nu^2}}{\sqrt{2\pi}\sigma_\nu} \left\{\prod_{t=1}^{n_i} F(y_{1it}, y_{2it}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i)\right\} d\nu_i$$

where

$$F(y_{1it}, y_{2it}, \Delta_{it}) = \begin{cases} \left(\sqrt{2\pi}\sigma_\epsilon\right)^{-1} e^{-(y_{1it}-\Delta_{it})^2/(2\sigma_\epsilon^2)} & \text{if } (y_{1it}, y_{2it}) \in C \\ \Phi\left(\frac{y_{2it}-\Delta_{it}}{\sigma_\epsilon}\right) & \text{if } (y_{1it}, y_{2it}) \in L \\ 1-\Phi\left(\frac{y_{1it}-\Delta_{it}}{\sigma_\epsilon}\right) & \text{if } (y_{1it}, y_{2it}) \in R \\ \Phi\left(\frac{y_{2it}-\Delta_{it}}{\sigma_\epsilon}\right) - \Phi\left(\frac{y_{1it}-\Delta_{it}}{\sigma_\epsilon}\right) & \text{if } (y_{1it}, y_{2it}) \in I \end{cases}$$

where C is the set of noncensored observations ($y_{1it} = y_{2it}$ and both nonmissing), L is the set of left-censored observations (y_{1it} missing and y_{2it} nonmissing), R is the set of right-censored observations (y_{1it} nonmissing and y_{2it} missing), I is the set of interval observations ($y_{1it} < y_{2it}$ and both nonmissing), and $\Phi()$ is the cumulative normal distribution.

The panel-level likelihood l_i is given by

$$\begin{aligned} l_i &= \int_{-\infty}^{\infty} \frac{e^{-\nu_i^2/2\sigma_\nu^2}}{\sqrt{2\pi}\sigma_\nu} \left\{ \prod_{t=1}^{n_i} F(y_{1it}, y_{2it}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i) \right\} d\nu_i \\ &\equiv \int_{-\infty}^{\infty} g(y_{1it}, y_{2it}, x_{it}, \nu_i) d\nu_i \end{aligned}$$

This integral can be approximated with M-point Gauss–Hermite quadrature

$$\int_{-\infty}^{\infty} e^{-x^2} h(x) dx \approx \sum_{m=1}^{M} w_m^* h(a_m^*)$$

This is equivalent to

$$\int_{-\infty}^{\infty} f(x) dx \approx \sum_{m=1}^{M} w_m^* \exp\left\{(a_m^*)^2\right\} f(a_m^*)$$

where the w_m^* denote the quadrature weights and the a_m^* denote the quadrature abscissas. The log likelihood, L, is the sum of the logs of the panel-level likelihoods l_i.

The default approximation of the log likelihood is by adaptive Gauss–Hermite quadrature, which approximates the panel-level likelihood with

$$l_i \approx \sqrt{2}\widehat{\sigma}_i \sum_{m=1}^{M} w_m^* \exp\left\{(a_m^*)^2\right\} g(y_{1it}, y_{2it}, x_{it}, \sqrt{2}\widehat{\sigma}_i a_m^* + \widehat{\mu}_i)$$

where $\widehat{\sigma}_i$ and $\widehat{\mu}_i$ are the adaptive parameters for panel i. Therefore, using the definition of $g(y_{1it}, y_{2it}, x_{it}, \nu_i)$, the total log likelihood is approximated by

$$L \approx \sum_{i=1}^{n} w_i \log\left[\sqrt{2}\widehat{\sigma}_i \sum_{m=1}^{M} w_m^* \exp\{(a_m^*)^2\} \frac{\exp\{-(\sqrt{2}\widehat{\sigma}_i a_m^* + \widehat{\mu}_i)^2/2\sigma_\nu^2\}}{\sqrt{2\pi}\sigma_\nu} \right.$$
$$\left. \prod_{t=1}^{n_i} F(y_{1it}, y_{2it}, x_{it}\boldsymbol{\beta} + \sqrt{2}\widehat{\sigma}_i a_m^* + \widehat{\mu}_i)\right] \tag{1}$$

where w_i is the user-specified weight for panel i; if no weights are specified, $w_i = 1$.

The default method of adaptive Gauss–Hermite quadrature is to calculate the posterior mean and variance and use those parameters for $\widehat{\mu}_i$ and $\widehat{\sigma}_i$ by following the method of Naylor and Smith (1982), further discussed in Skrondal and Rabe-Hesketh (2004). We start with $\widehat{\sigma}_{i,0} = 1$ and $\widehat{\mu}_{i,0} = 0$, and the posterior means and variances are updated in the kth iteration. That is, at the kth iteration of the optimization for l_i we use

$$l_{i,k} \approx \sum_{m=1}^{M} \sqrt{2}\widehat{\sigma}_{i,k-1} w_m^* \exp\{a_m^*)^2\} g(y_{1it}, y_{2it}, x_{it}, \sqrt{2}\widehat{\sigma}_{i,k-1} a_m^* + \widehat{\mu}_{i,k-1})$$

Letting

$$\tau_{i,m,k-1} = \sqrt{2}\widehat{\sigma}_{i,k-1} a_m^* + \widehat{\mu}_{i,k-1}$$

$$\widehat{\mu}_{i,k} = \sum_{m=1}^{M} (\tau_{i,m,k-1}) \frac{\sqrt{2}\widehat{\sigma}_{i,k-1} w_m^* \exp\{(a_m^*)^2\} g(y_{1it}, y_{2it}, x_{it}, \tau_{i,m,k-1})}{l_{i,k}}$$

and

$$\widehat{\sigma}_{i,k} = \sum_{m=1}^{M} (\tau_{i,m,k-1})^2 \frac{\sqrt{2}\widehat{\sigma}_{i,k-1} w_m^* \exp\{(a_m^*)^2\} g(y_{1it}, y_{2it}, x_{it}, \tau_{i,m,k-1})}{l_{i,k}} - (\widehat{\mu}_{i,k})^2$$

and this is repeated until $\widehat{\mu}_{i,k}$ and $\widehat{\sigma}_{i,k}$ have converged for this iteration of the maximization algorithm. This adaptation is applied on every iteration until the log-likelihood change from the preceding iteration is less than a relative difference of 1e–6; after this, the quadrature parameters are fixed.

The log likelihood can also be calculated by nonadaptive Gauss–Hermite quadrature if the `intmethod(ghermite)` option is specified. For nonadaptive Gauss–Hermite quadrature, the following formula for the log likelihood is used in place of (1).

$$L = \sum_{i=1}^{n} w_i \log f\left\{(y_{1i1}, y_{2i1}), \ldots, (y_{1in_i}, y_{2in_i}) | \mathbf{x}_{1i}, \ldots, \mathbf{x}_{in_i}\right\}$$
$$\approx \sum_{i=1}^{n} w_i \log\left\{\frac{1}{\sqrt{\pi}} \sum_{m=1}^{M} w_m^* \prod_{t=1}^{n_i} F\left(y_{1it}, y_{2it}, \mathbf{x}_{it}\boldsymbol{\beta} + \sqrt{2}\sigma_\nu a_m^*\right)\right\}$$

Both quadrature formulas require that the integrated function be well approximated by a polynomial of degree equal to the number of quadrature points. Panel size can affect whether

$$\prod_{t=1}^{n_i} F(y_{1it}, y_{2it}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i)$$

is well approximated by a polynomial. As panel size and ρ increase, the quadrature approximation can become less accurate. For large ρ, the random-effects model can also become unidentified. Adaptive quadrature gives better results for correlated data and large panels than nonadaptive quadrature; however, we recommend that you use the `quadchk` command (see [XT] **quadchk**) to verify the quadrature approximation used in this command, whichever approximation you choose.

References

Naylor, J. C., and A. F. M. Smith. 1982. Applications of a method for the efficient computation of posterior distributions. *Journal of the Royal Statistical Society,* C ser., 31: 214–225. https://doi.org/10.2307/2347995.

Neuhaus, J. M. 1992. Statistical methods for longitudinal and clustered designs with binary responses. *Statistical Methods in Medical Research* 1: 249–273. https://doi.org/10.1177/096228029200100303.

Pendergast, J. F., S. J. Gange, M. A. Newton, M. J. Lindstrom, M. Palta, and M. R. Fisher. 1996. A survey of methods for analyzing clustered binary response data. *International Statistical Review* 64: 89–118. https://doi.org/10.2307/1403425.

Skrondal, A., and S. Rabe-Hesketh. 2004. *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models*. Boca Raton, FL: Chapman and Hall/CRC.

Also see

[XT] **xtintreg postestimation** — Postestimation tools for xtintreg

[XT] **quadchk** — Check sensitivity of quadrature approximation

[XT] **xteintreg** — Extended random-effects interval regression

[XT] **xtreg** — Linear models for panel data

[XT] **xtset** — Declare data to be panel data

[XT] **xttobit** — Random-effects tobit model

[ME] **meintreg** — Multilevel mixed-effects interval regression

[R] **intreg** — Interval regression

[R] **tobit** — Tobit regression

[ST] **stintreg** — Parametric models for interval-censored survival-time data

[U] 20 Estimation and postestimation commands

xtintreg postestimation — Postestimation tools for xtintreg

Postestimation commands

The following postestimation commands are available after `xtintreg`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of parameters
`estat ic`	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC, respectively)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
`lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	predictions and their SEs, etc.
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of parameters
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

predict

Description for predict

`predict` creates a new variable containing predictions such as linear predictions, standard errors, probabilities, and expected values.

Menu for predict

Statistics > Postestimation

Syntax for predict

`predict` [*type*] *newvar* [*if*] [*in*] [, *statistic* `nooffset`]

statistic	Description
Main	
`xb`	linear prediction; the default
`stdp`	standard error of the linear prediction
`stdf`	standard error of the linear forecast
`pr(`a`,`b`)`	$\Pr(a < y < b)$, marginal with respect to the random effect
`e(`a`,`b`)`	$E(y \mid a < y < b)$, marginal with respect to the random effect
`ystar(`a`,`b`)`	$E(y^*)$, $y^* = \max\{a, \min(y, b)\}$, marginal with respect to the random effect

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

where a and b may be numbers or variables; a missing ($a \geq .$) means $-\infty$, and b missing ($b \geq .$) means $+\infty$; see **[U] 12.2.1 Missing values**.

Options for predict

Main

`xb`, the default, calculates the linear prediction $\mathbf{x}_{it}\boldsymbol{\beta}$ using the estimated fixed effects (coefficients) in the model. This is equivalent to fixing all random effects in the model to their theoretical (prior) mean value of zero.

`stdp` calculates the standard error of the linear prediction. It can be thought of as the standard error of the predicted expected value or mean for the observation's covariate pattern. The standard error of the prediction is also referred to as the standard error of the fitted value.

`stdf` calculates the standard error of the linear forecast. This is the standard error of the point prediction for 1 observation. It is commonly referred to as the standard error of the future or forecast value. By construction, the standard errors produced by `stdf` are always larger than those produced by `stdp`; see *Methods and formulas* in [R] **regress**.

`pr(`*a*`,`*b*`)` calculates estimates of $\Pr(a < y < b|\mathbf{x} = \mathbf{x}_{it})$, which is the probability that y would be observed in the interval (a, b), given the current values of the predictors, $\mathbf{x}_{it}$. The predictions are calculated marginally with respect to the random effect. That is, the random effect is integrated out of the prediction function. In the discussion that follows, these two conditions are implied.

a and *b* may be specified as numbers or variable names; *lb* and *ub* are variable names;
`pr(20,30)` calculates $\Pr(20 < y < 30)$;
`pr(`*lb*`,`*ub*`)` calculates $\Pr(lb < y < ub)$; and
`pr(20,`*ub*`)` calculates $\Pr(20 < y < ub)$.

a missing ($a \geq .$) means $-\infty$; `pr(.,30)` calculates $\Pr(-\infty < y < 30)$;
`pr(`*lb*`,30)` calculates $\Pr(-\infty < y < 30)$ in observations for which $lb \geq .$
(and calculates $\Pr(lb < y < 30)$ elsewhere).

b missing ($b \geq .$) means $+\infty$; `pr(20,.)` calculates $\Pr(+\infty > y > 20)$;
`pr(20,`*ub*`)` calculates $\Pr(+\infty > y > 20)$ in observations for which $ub \geq .$
(and calculates $\Pr(20 < y < ub)$ elsewhere).

`e(`*a*`,`*b*`)` calculates estimates of $E(y \mid a < y < b, \mathbf{x} = \mathbf{x}_{it})$, which is the expected value of y conditional on y being in the interval (a, b), meaning that y is truncated. *a* and *b* are specified as they are for `pr()`. The predictions are calculated marginally with respect to the random effect. That is, the random effect is integrated out of the prediction function.

`ystar(`*a*`,`*b*`)` calculates estimates of $E(y^*|\mathbf{x} = \mathbf{x}_{it})$, where $y^* = a$ if $y \leq a$, $y^* = b$ if $y \geq b$, and $y^* = y$ otherwise, meaning that y^* is the censored version of y. *a* and *b* are specified as they are for `pr()`. The predictions are calculated marginally with respect to the random effect. That is, the random effect is integrated out of the prediction function.

`nooffset` is relevant only if you specified `offset(`*varname*`)` for `xtintreg`. It modifies the calculations made by `predict` so that they ignore the offset variable; the linear prediction is treated as $\mathbf{x}_{it}\boldsymbol{\beta}$ rather than $\mathbf{x}_{it}\boldsymbol{\beta} + \text{offset}_{it}$.

margins

Description for margins

`margins` estimates margins of response for linear predictions, probabilities, and expected values.

Menu for margins

Statistics > Postestimation

Syntax for margins

`margins` [*marginlist*] [, *options*]

`margins` [*marginlist*] , `predict(`*statistic* ...`)` [`predict(`*statistic* ...`)` ...] [*options*]

statistic	Description
`xb`	linear prediction; the default
`pr(`a`,`b`)`	$\Pr(a < y < b)$, marginal with respect to the random effect
`e(`a`,`b`)`	$E(y \mid a < y < b)$, marginal with respect to the random effect
`ystar(`a`,`b`)`	$E(y^*)$, $y^* = \max\{a, \min(y, b)\}$, marginal with respect to the random effect
`stdp`	not allowed with `margins`
`stdf`	not allowed with `margins`

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.

For the full syntax, see [R] **margins**.

Remarks and examples

▷ Example 1: Average marginal probabilities at specified covariate values

In example 1 of [XT] **xtintreg**, we fit a random-effects model of wages. Say that we want to know how union membership status affects the probability that a worker's wage will be "low", where low means a log wage that is less than the 20th percentile of all observations in our dataset. First, we use `centile` to find the 20th percentile of `ln_wage`:

```
. use https://www.stata-press.com/data/r19/nlswork5
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)
. xtintreg ln_wage1 ln_wage2 i.union age grade not_smsa south##c.year, intreg
  (output omitted)
. centile ln_wage, centile(20)

                                                          Binom. interp.
    Variable |       Obs  Percentile    Centile        [95% conf. interval]
-------------+-------------------------------------------------------------
     ln_wage |    28,534          20    1.301507        1.297063    1.308635
```

Now we use `margins` to obtain the effect of union status on the probability that the log of wages is in the bottom 20% of women. Given the results from `centile` that corresponds to the log of wages being below 1.30. We evaluate the effect for two groups: 1) women age 30 living in the south in 1988 who graduated high school, but had no more schooling, and 2) the same group of women, but who are instead college graduates (`grade=16`).

```
. margins, dydx(union) predict(pr(.,1.30))
> at(age=30 south=1 year=88 grade=12 union=0)
> at(age=30 south=1 year=88 grade=16 union=0)

Average marginal effects                               Number of obs = 19,224
Model VCE: OIM

Expression: Pr(ln_wage1<1.30), predict(pr(.,1.30))
dy/dx wrt:  1.union
1._at: union =  0
       age   = 30
       grade = 12
       south =  1
       year  = 88
2._at: union =  0
       age   = 30
       grade = 16
       south =  1
       year  = 88
```

	dy/dx	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
0.union	(base outcome)					
1.union						
_at						
1	-.0755536	.0058942	-12.82	0.000	-.0871059	-.0640012
2	-.0368238	.0034632	-10.63	0.000	-.0436114	-.0300361

```
Note: dy/dx for factor levels is the discrete change from the base level.
```

For the first group of women, according to our fitted model, being in a union lowers the probability of being classified as a low-wage worker by almost 7.6 percentage points. Being a college graduate attenuates this effect to just under 3.7 percentage points.

◁

Methods and formulas

Methods and formulas for calculating the available predictions are given in *Methods and formulas* of [XT] **xttobit postestimation**.

Also see

[XT] **xtintreg** — Random-effects interval-data regression model

[U] **20 Estimation and postestimation commands**

xtivreg — Instrumental variables and two-stage least squares for panel-data models

Description

`xtivreg` offers five different estimators for fitting panel-data models in which some of the right-hand-side covariates are endogenous. These estimators are two-stage least-squares generalizations of simple panel-data estimators for exogenous variables. `xtivreg` with the `be` option uses the two-stage least-squares between estimator. `xtivreg` with the `fe` option uses the two-stage least-squares within estimator. `xtivreg` with the `re` option uses a two-stage least-squares random-effects estimator. There are two implementations: G2SLS from Balestra and Varadharajan-Krishnakumar (1987) and EC2SLS from Baltagi. The Balestra and Varadharajan-Krishnakumar G2SLS is the default because it is computationally less expensive. Baltagi's EC2SLS can be obtained by specifying the `ec2sls` option. `xtivreg` with the `fd` option requests the two-stage least-squares first-differenced estimator.

See Baltagi (2013) for an introduction to panel-data models with endogenous covariates. For the derivation and application of the first-differenced estimator, see Anderson and Hsiao (1981).

Quick start

Random-effects linear panel-data model with outcome `y`, exogenous `x1`, and `x2` instrumented by `x3` using `xtset` data

```
xtivreg y x1 (x2 = x3)
```

Use fixed-effects estimator and include indicators for each level of categorical variable `a`

```
xtivreg y x1 i.a (x2 = x3), fe
```

Use between-effects estimator and include indicators for levels of `b` as instruments

```
xtivreg y x1 i.a (x2 = x3 i.b), be
```

First-differenced model of `y` as a function of `x1` and `x2` and the lag of `y` instrumented by its third lag

```
xtivreg y x1 x2 (L.y = L3.y), fd
```

Menu

Statistics > Longitudinal/panel data > Endogenous covariates > Instrumental-variables regression (FE, RE, BE, FD)

Syntax

GLS random-effects (RE) model

`xtivreg` *depvar* [*varlist*$_1$] (*varlist*$_2$ = *varlist*$_{iv}$) [*if*] [*in*] [, `re` *RE_options*]

Between-effects (BE) model

`xtivreg` *depvar* [*varlist*$_1$] (*varlist*$_2$ = *varlist*$_{iv}$) [*if*] [*in*] , `be` [*BE_options*]

Fixed-effects (FE) model

`xtivreg` *depvar* [*varlist*$_1$] (*varlist*$_2$ = *varlist*$_{iv}$) [*if*] [*in*] , `fe` [*FE_options*]

First-differenced (FD) estimator

`xtivreg` *depvar* [*varlist*$_1$] (*varlist*$_2$ = *varlist*$_{iv}$) [*if*] [*in*] , `fd` [*FD_options*]

RE_options	Description
Model	
`re`	use random-effects estimator; the default
`ec2sls`	use Baltagi's EC2SLS random-effects estimator
`nosa`	use the Baltagi–Chang estimators of the variance components
`regress`	treat covariates as exogenous and ignore instrumental variables
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `conventional`, `robust`, `cluster` *clustvar*, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`first`	report first-stage estimates
`small`	report t and F statistics instead of Z and χ^2 statistics
`theta`	report θ
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
`coeflegend`	display legend instead of statistics

BE_options	Description
Model	
`be`	use between-effects estimator
`regress`	treat covariates as exogenous and ignore instrumental variables
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `conventional`, `robust`, `cluster` *clustvar*, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`first`	report first-stage estimates
`small`	report t and F statistics instead of Z and χ^2 statistics
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
`coeflegend`	display legend instead of statistics

FE_options	Description
Model	
`fe`	use fixed-effects estimator
`regress`	treat covariates as exogenous and ignore instrumental variables
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `conventional`, `robust`, `cluster` *clustvar*, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`first`	report first-stage estimates
`small`	report t and F statistics instead of Z and χ^2 statistics
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
`coeflegend`	display legend instead of statistics

FD_options	Description
Model	
noconstant	suppress constant term
fd	use first-differenced estimator
regress	treat covariates as exogenous and ignore instrumental variables
SE/Robust	
vce(*vcetype*)	*vcetype* may be conventional, robust, cluster *clustvar*, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
first	report first-stage estimates
small	report t and F statistics instead of Z and χ^2 statistics
display_options	control columns and column formats, row spacing, line width, and display of omitted variables
coeflegend	display legend instead of statistics

A panel variable must be specified. For xtivreg, fd, a time variable must also be specified. Use xtset; see [XT] **xtset**.

varlist$_1$ and *varlist*$_{iv}$ may contain factor variables, except for the fd estimator; see **[U] 11.4.3 Factor variables**.

depvar, *varlist*$_1$, *varlist*$_2$, and *varlist*$_{iv}$ may contain time-series operators; see **[U] 11.4.4 Time-series varlists**.

by, collect, and statsby are allowed; see **[U] 11.1.10 Prefix commands**.

coeflegend does not appear in the dialog box.

See **[U] 20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options for RE model

Model

re requests the G2SLS random-effects estimator. re is the default.

ec2sls requests Baltagi's EC2SLS random-effects estimator instead of the default Balestra and Varadharajan-Krishnakumar estimator.

nosa specifies that the Baltagi–Chang estimators of the variance components be used instead of the default adapted Swamy–Arora estimators.

regress specifies that all the covariates be treated as exogenous and that the instrument list be ignored. Specifying regress causes xtivreg to fit the requested panel-data regression model of *depvar* on *varlist*$_1$ and *varlist*$_2$, ignoring *varlist*$_{iv}$.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (conventional), that are robust to some kinds of misspecification (robust), that allow for intragroup correlation (cluster *clustvar*), and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] ***vce_options***.

vce(conventional), the default, uses the conventionally derived variance estimator for generalized least-squares regression.

Specifying vce(robust) is equivalent to specifying vce(cluster *panelvar*); see xtreg, re in *Methods and formulas* of [XT] **xtreg**.

Reporting

level(#); see [R] **Estimation options**.

first specifies that the first-stage regressions be displayed.

small specifies that t statistics be reported instead of Z statistics and that F statistics be reported instead of χ^2 statistics.

theta specifies that the output include the estimated value of θ used in combining the between and fixed estimators. For balanced data, this is a constant, and for unbalanced data, a summary of the values is presented in the header of the output.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

The following option is available with xtivreg but is not shown in the dialog box:

coeflegend; see [R] **Estimation options**.

Options for BE model

Model

be requests the between regression estimator.

regress specifies that all the covariates be treated as exogenous and that the instrument list be ignored. Specifying regress causes xtivreg to fit the requested panel-data regression model of *depvar* on $varlist_1$ and $varlist_2$, ignoring $varlist_{iv}$.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (conventional), that are robust to some kinds of misspecification (robust), that allow for intragroup correlation (cluster *clustvar*), and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] ***vce_options***.

vce(conventional), the default, uses the conventionally derived variance estimator for generalized least-squares regression.

Specifying vce(robust) is equivalent to specifying vce(cluster *panelvar*); see xtreg, fe in *Methods and formulas* of [XT] **xtreg**.

Reporting

level(#); see [R] **Estimation options**.

first specifies that the first-stage regressions be displayed.

small specifies that t statistics be reported instead of Z statistics and that F statistics be reported instead of χ^2 statistics.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

The following option is available with `xtivreg` but is not shown in the dialog box:

`coeflegend`; see [R] **Estimation options**.

Options for FE model

Model

`fe` requests the fixed-effects (within) regression estimator.

`regress` specifies that all the covariates be treated as exogenous and that the instrument list be ignored. Specifying `regress` causes `xtivreg` to fit the requested panel-data regression model of *depvar* on $varlist_1$ and $varlist_2$, ignoring $varlist_{iv}$.

SE/Robust

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`conventional`), that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster` *clustvar*), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [XT] ***vce_options***.

`vce(conventional)`, the default, uses the conventionally derived variance estimator for generalized least-squares regression.

Specifying `vce(robust)` is equivalent to specifying `vce(cluster` *panelvar*`)`; see *xtreg, fe* in *Methods and formulas* of [XT] **xtreg**.

Reporting

`level(#)`; see [R] **Estimation options**.

`first` specifies that the first-stage regressions be displayed.

`small` specifies that t statistics be reported instead of Z statistics and that F statistics be reported instead of χ^2 statistics.

display_options: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(`*style*`)`, `cformat(`*%fmt*`)`, `pformat(`*%fmt*`)`, `sformat(`*%fmt*`)`, and `nolstretch`; see [R] **Estimation options**.

The following option is available with `xtivreg` but is not shown in the dialog box:

`coeflegend`; see [R] **Estimation options**.

Options for FD model

Model

`noconstant`; see [R] **Estimation options**.

`fd` requests the first-differenced regression estimator.

`regress` specifies that all the covariates be treated as exogenous and that the instrument list be ignored. Specifying `regress` causes `xtivreg` to fit the requested panel-data regression model of *depvar* on $varlist_1$ and $varlist_2$, ignoring $varlist_{iv}$.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (conventional), that are robust to some kinds of misspecification (robust), that allow for intragroup correlation (cluster *clustvar*), and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] ***vce_options***.

vce(conventional), the default, uses the conventionally derived variance estimator for generalized least-squares regression.

Specifying vce(robust) is equivalent to specifying vce(cluster *panelvar*); see *xtreg, fe* in *Methods and formulas* of [XT] **xtreg**.

Reporting

level(#); see [R] **Estimation options**.

first specifies that the first-stage regressions be displayed.

small specifies that t statistics be reported instead of Z statistics and that F statistics be reported instead of χ^2 statistics.

display_options: noci, nopvalues, noomitted, vsquish, cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

The following option is available with xtivreg but is not shown in the dialog box:

coeflegend; see [R] **Estimation options**.

Remarks and examples

If you have not read [XT] **xt**, please do so.

Consider an equation of the form

$$y_{it} = \mathbf{Y}_{it}\gamma + \mathbf{X}_{1it}\boldsymbol{\beta} + \mu_i + \nu_{it} = \mathbf{Z}_{it}\boldsymbol{\delta} + \mu_i + \nu_{it} \tag{1}$$

where

y_{it} is the dependent variable;

$\mathbf{Y}_{it}$ is an $1 \times g_2$ vector of observations on g_2 endogenous variables included as covariates, and these variables are allowed to be correlated with the ν_{it};

$\mathbf{X}_{1it}$ is an $1 \times k_1$ vector of observations on the exogenous variables included as covariates;

$\mathbf{Z}_{it} = [\mathbf{Y}_{it}\ \mathbf{X}_{it}]$;

γ is a $g_2 \times 1$ vector of coefficients;

$\boldsymbol{\beta}$ is a $k_1 \times 1$ vector of coefficients; and

$\boldsymbol{\delta}$ is a $K \times 1$ vector of coefficients, where $K = g_2 + k_1$.

Assume that there is a $1 \times k_2$ vector of observations on the k_2 instruments in $\mathbf{X}_{2it}$. The order condition is satisfied if $k_2 \geq g_2$. Let $\mathbf{X}_{it} = [\mathbf{X}_{1it}\ \mathbf{X}_{2it}]$. xtivreg handles exogenously unbalanced panel data. Thus define T_i to be the number of observations on panel i, n to be the number of panels and N to be the total number of observations; that is, $N = \sum_{i=1}^{n} T_i$.

xtivreg offers five different estimators, which may be applied to models having the form of (1). The first-differenced estimator (FD2SLS) removes the μ_i by fitting the model in first differences. The within estimator (FE2SLS) fits the model after sweeping out the μ_i by removing the panel-level means from each variable. The between estimator (BE2SLS) models the panel averages. The two random-effects estimators, G2SLS and EC2SLS, treat the μ_i as random variables that are independent and identically distributed (i.i.d.) over the panels. Except for (FD2SLS), all of these estimators are generalizations of estimators in xtreg. See [XT] **xtreg** for a discussion of these estimators for exogenous covariates.

Although the estimators allow for different assumptions about the μ_i, all the estimators assume that the idiosyncratic error term ν_{it} has zero mean and is uncorrelated with the variables in $\mathbf{X}_{it}$. Just as when there are no endogenous covariates, as discussed in [XT] **xtreg**, there are various perspectives on what assumptions should be placed on the μ_i. If they are assumed to be fixed, the μ_i may be correlated with the variables in $\mathbf{X}_{it}$, and the within estimator is efficient within a class of limited information estimators. Alternatively, if the μ_i are assumed to be random, they are also assumed to be i.i.d. over the panels. If the μ_i are assumed to be uncorrelated with the variables in $\mathbf{X}_{it}$, the GLS random-effects estimators are more efficient than the within estimator. However, if the μ_i are correlated with the variables in $\mathbf{X}_{it}$, the random-effects estimators are inconsistent but the within estimator is consistent. The price of using the within estimator is that it is not possible to estimate coefficients on time-invariant variables, and all inference is conditional on the μ_i in the sample. See Mundlak (1978) and Hsiao (2014) for discussions of this interpretation of the within estimator.

▷ Example 1: Fixed-effects model

For the within estimator, consider another version of the wage equation discussed in [XT] **xtreg**. The data for this example come from an extract of women from the National Longitudinal Survey of Youth that was described in detail in [XT] **xt**. Restricting ourselves to only time-varying covariates, we might suppose that the log of the real wage was a function of the individual's age, age^2, her tenure in the observed place of employment, whether she belonged to union, whether she lives in metropolitan area, and whether she lives in the south. The variables for these are, respectively, age, c.age#c.age, tenure, union, not_smsa, and south. If we treat all the variables as exogenous, we can use the one-stage within estimator from xtreg, yielding

```
. use https://www.stata-press.com/data/r19/nlswork
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)

. xtreg ln_w age c.age#c.age tenure not_smsa union south, fe

Fixed-effects (within) regression               Number of obs     =     19,007
Group variable: idcode                          Number of groups  =      4,134

R-squared:                                      Obs per group:
     Within  = 0.1333                                         min =          1
     Between = 0.2375                                         avg =        4.6
     Overall = 0.2031                                         max =         12

                                                F(6, 14867)       =     381.19
corr(u_i, Xb) = 0.2074                          Prob > F          =     0.0000

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
         age |   .0311984   .0033902     9.20   0.000     .0245533    .0378436
             |
 c.age#c.age |  -.0003457   .0000543    -6.37   0.000    -.0004522   -.0002393
             |
      tenure |   .0176205   .0008099    21.76   0.000     .0160331    .0192079
    not_smsa |  -.0972535   .0125377    -7.76   0.000    -.1218289    -.072678
       union |   .0975672   .0069844    13.97   0.000     .0838769    .1112576
       south |  -.0620932    .013327    -4.66   0.000    -.0882158   -.0359706
       _cons |   1.091612   .0523126    20.87   0.000     .9890729    1.194151
-------------+----------------------------------------------------------------
     sigma_u |   .3910683
     sigma_e |  .25545969
         rho |  .70091004   (fraction of variance due to u_i)
------------------------------------------------------------------------------
F test that all u_i=0: F(4133, 14867) = 8.31                 Prob > F = 0.0000
```

All the coefficients are statistically significant and have the expected signs.

Now suppose that we wish to model tenure as a function of union and south and that we believe that the errors in the two equations are correlated. Because we are still interested in the within estimates, we now need a two-stage least-squares estimator. The following output shows the command and the results from fitting this model:

```
. xtivreg ln_w age c.age#c.age not_smsa (tenure =  union south), fe

Fixed-effects (within) IV regression            Number of obs     =     19,007
Group variable: idcode                          Number of groups  =      4,134

R-squared:                                      Obs per group:
     Within  =      .                                         min =          1
     Between = 0.1304                                         avg =        4.6
     Overall = 0.0897                                         max =         12

                                                Wald chi2(4)      =  147926.58
corr(u_i, Xb) = -0.6843                         Prob > chi2       =     0.0000

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      tenure |   .2403531   .0373419     6.44   0.000     .1671643    .3135419
         age |   .0118437   .0090032     1.32   0.188    -.0058023    .0294897
             |
 c.age#c.age |  -.0012145   .0001968    -6.17   0.000    -.0016003   -.0008286
             |
    not_smsa |  -.0167178   .0339236    -0.49   0.622    -.0832069    .0497713
       _cons |   1.678287   .1626657    10.32   0.000     1.359468    1.997106
-------------+----------------------------------------------------------------
     sigma_u |  .70661941
     sigma_e |  .63029359
         rho |  .55690561   (fraction of variance due to u_i)
------------------------------------------------------------------------------
 F test that all u_i=0: F(4133,14869) =     1.44          Prob > F    = 0.0000
------------------------------------------------------------------------------
Endogenous: tenure
Exogenous:  age c.age#c.age not_smsa union south
```

Although all the coefficients still have the expected signs, the coefficients on `age` and `not_smsa` are no longer statistically significant. Given that these variables have been found to be important in many other studies, we might want to rethink our specification.

◁

If we are willing to assume that the μ_i are uncorrelated with the other covariates, we can fit a random-effects model. The model is frequently known as the variance-components or error-components model. `xtivreg` has estimators for two-stage least-squares one-way error-components models. In the one-way framework, there are two variance components to estimate, the variance of the μ_i and the variance of the ν_{it}. Because the variance components are unknown, consistent estimates are required to implement feasible GLS. `xtivreg` offers two choices: a Swamy–Arora method and simple consistent estimators from Baltagi and Chang (2000).

Baltagi and Chang (1994) derived the Swamy–Arora estimators of the variance components for unbalanced panels. By default, `xtivreg` uses estimators that extend these unbalanced Swamy–Arora estimators to the case with instrumental variables. The default Swamy–Arora method contains a degree-of-freedom correction to improve its performance in small samples. Baltagi and Chang (2000) use variance-components estimators, which are based on the ideas of Amemiya (1971) and Swamy and Arora (1972), but they do not attempt to make small-sample adjustments. These consistent estimators of the variance components will be used if the `nosa` option is specified.

Using either estimator of the variance components, `xtivreg` offers two GLS estimators of the random-effects model. These two estimators differ only in how they construct the GLS instruments from the exogenous and instrumental variables contained in $\mathbf{X}_{it} = [\mathbf{X}_{1it}\ \mathbf{X}_{2it}]$. The default method, G2SLS, which is from Balestra and Varadharajan-Krishnakumar, uses the exogenous variables after they have been passed through the feasible GLS transform. In math, G2SLS uses $\mathbf{X}^*_{it}$ for the GLS instruments, where $\mathbf{X}^*_{it}$ is constructed by passing each variable in $\mathbf{X}_{it}$ through the GLS transform in (3) given in *Methods and formulas*. If the `ec2sls` option is specified, `xtivreg` performs Baltagi's EC2SLS. In EC2SLS, the instruments are $\widetilde{\mathbf{X}}_{it}$ and $\overline{\mathbf{X}}_{it}$, where $\widetilde{\mathbf{X}}_{it}$ is constructed by passing each of the variables in $\mathbf{X}_{it}$ through the within transform, and $\overline{\mathbf{X}}_{it}$ is constructed by passing each variable through the between transform. The within and between transforms are given in the *Methods and formulas* section. Baltagi and Li (1992) show that, although the G2SLS instruments are a subset of those contained in EC2SLS, the extra instruments in EC2SLS are redundant in the sense of White (2001). Given the extra computational cost, G2SLS is the default.

▷ Example 2: GLS random-effects model

Here is the output from applying the G2SLS estimator to this model:

```
. xtivreg ln_w age c.age#c.age not_smsa 2.race (tenure = union birth south), re
G2SLS random-effects IV regression              Number of obs     =     19,007
Group variable: idcode                          Number of groups  =      4,134

R-squared:                                      Obs per group:
     Within  = 0.0664                                        min =          1
     Between = 0.2098                                        avg =        4.6
     Overall = 0.1463                                        max =         12

                                                Wald chi2(5)      =    1446.37
corr(u_i, X) = 0 (assumed)                      Prob > chi2       =     0.0000

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      tenure |   .1391798   .0078756    17.67   0.000      .123744    .1546157
         age |   .0279649   .0054182     5.16   0.000     .0173454    .0385843
             |
 c.age#c.age |  -.0008357   .0000871    -9.60   0.000    -.0010063    -.000665
             |
    not_smsa |  -.2235103   .0111371   -20.07   0.000    -.2453386   -.2016821
             |
        race |
      Black  |  -.2078613   .0125803   -16.52   0.000    -.2325183   -.1832044
       _cons |   1.337684   .0844988    15.83   0.000     1.172069    1.503299
-------------+----------------------------------------------------------------
     sigma_u |  .36582493
     sigma_e |  .63031479
         rho |  .25197078   (fraction of variance due to u_i)
------------------------------------------------------------------------------
Endogenous: tenure
Exogenous:  age c.age#c.age not_smsa 2.race union birth_yr south
```

We have included two time-invariant covariates, `birth_yr` and `2.race`. All the coefficients are statistically significant and are of the expected sign.

Applying the EC2SLS estimator yields similar results:

```
. xtivreg ln_w age c.age#c.age not_smsa 2.race (tenure = union birth south), re
> ec2sls

EC2SLS random-effects IV regression             Number of obs     =     19,007
Group variable: idcode                          Number of groups  =      4,134

R-squared:                                      Obs per group:
     Within  = 0.0898                                         min =          1
     Between = 0.2608                                         avg =        4.6
     Overall = 0.1926                                         max =         12

                                                Wald chi2(5)      =    2721.92
corr(u_i, X) = 0 (assumed)                      Prob > chi2       =     0.0000

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      tenure |    .064822   .0025647    25.27   0.000     .0597953    .0698486
         age |   .0380048   .0039549     9.61   0.000     .0302534    .0457562
             |
 c.age#c.age |  -.0006676   .0000632   -10.56   0.000    -.0007915   -.0005438
             |
    not_smsa |  -.2298961   .0082993   -27.70   0.000    -.2461625   -.2136297
             |
        race |
      Black  |  -.1823627   .0092005   -19.82   0.000    -.2003954     -.16433
       _cons |   1.110564   .0606538    18.31   0.000     .9916849    1.229443
-------------+----------------------------------------------------------------
     sigma_u |  .36582493
     sigma_e |  .63031479
         rho |  .25197078   (fraction of variance due to u_i)
------------------------------------------------------------------------------
Endogenous: tenure
Exogenous:  age c.age#c.age not_smsa 2.race union birth_yr south
```

Fitting the same model as above with the G2SLS estimator and the consistent variance components estimators yields

```
. xtivreg ln_w age c.age#c.age not_smsa 2.race (tenure = union birth south),
> re nosa

G2SLS random-effects IV regression              Number of obs     =     19,007
Group variable: idcode                          Number of groups  =      4,134

R-squared:                                      Obs per group:
     Within  = 0.0664                                         min =          1
     Between = 0.2098                                         avg =        4.6
     Overall = 0.1463                                         max =         12

                                                Wald chi2(5)      =    1446.93
corr(u_i, X) = 0 (assumed)                      Prob > chi2       =     0.0000

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      tenure |   .1391859    .007873    17.68   0.000     .1237552    .1546166
         age |   .0279697    .005419     5.16   0.000     .0173486    .0385909
             |
 c.age#c.age |  -.0008357   .0000871    -9.60   0.000    -.0010064    -.000665
             |
    not_smsa |  -.2235738   .0111344   -20.08   0.000    -.2453967   -.2017508
             |
        race |
      Black  |  -.2078733   .0125751   -16.53   0.000    -.2325201   -.1832265
       _cons |   1.337522   .0845083    15.83   0.000     1.171889    1.503155
-------------+----------------------------------------------------------------
     sigma_u |  .36535633
     sigma_e |  .63020883
         rho |   .2515512   (fraction of variance due to u_i)
------------------------------------------------------------------------------
Endogenous: tenure
Exogenous:  age c.age#c.age not_smsa 2.race union birth_yr south
```

◁

▷ Example 3: First-differenced estimator

The two-stage least-squares first-differenced estimator (FD2SLS) has been used to fit both fixed-effect and random-effect models. If the μ_i are truly fixed-effects, the FD2SLS estimator is not as efficient as the two-stage least-squares within estimator for finite T_i. Similarly, if none of the endogenous variables are lagged dependent variables, the exogenous variables are all strictly exogenous, and the random effects are i.i.d. and independent of the $\mathbf{X}_{it}$, the two-stage GLS estimators are more efficient than the FD2SLS estimator. However, the FD2SLS estimator has been used to obtain consistent estimates when one of these conditions fails. Anderson and Hsiao (1981) used a version of the FD2SLS estimator to fit a panel-data model with a lagged dependent variable.

Arellano and Bond (1991) develop new one-step and two-step GMM estimators for dynamic panel data. See [XT] **xtabond** for a discussion of these estimators and Stata's implementation of them. In their article, Arellano and Bond (1991) apply their new estimators to a model of dynamic labor demand that had previously been considered by Layard and Nickell (1986). They also compare the results of their estimators with those from the Anderson–Hsiao estimator using data from an unbalanced panel of firms from the United Kingdom. As is conventional, all variables are indexed over the firm i and time t. In this dataset, $\mathtt{n}_{it}$ is the log of employment in firm i inside the United Kingdom at time t, $\mathtt{w}_{it}$ is the natural log of the real product wage, $\mathtt{k}_{it}$ is the natural log of the gross capital stock, and $\mathtt{ys}_{it}$ is the natural log of industry output. The model also includes time dummies `yr1980`, `yr1981`, `yr1982`, `yr1983`, and `yr1984`. In Arellano and Bond (1991, table 5, column e), the authors present the results from applying one version of the Anderson–Hsiao estimator to these data. This example reproduces their results for the coefficients, though standard errors are slightly different because Arellano and Bond are using robust standard errors from GMM while we obtain our robust standard errors from 2SLS.

```
. use https://www.stata-press.com/data/r19/abdata
. xtivreg n l2.n l(0/1).w l(0/2).(k ys) yr1981-yr1984 (l.n = l3.n), fd vce(robust)
First-differenced IV regression
Group variable: id                           Number of obs      =        471
Time variable: year                          Number of groups   =        140
R-squared:                                   Obs per group:
     Within  = 0.0141                                       min =          3
     Between = 0.9165                                       avg =        3.4
     Overall = 0.9892                                       max =          5

                                             Wald chi2(14)      =     259.49
corr(u_i, Xb) = 0.9239                       Prob > chi2        =     0.0000
                                  (Std. err. adjusted for 140 clusters in id)

                            Robust
         D.n   Coefficient  std. err.      z    P>|z|     [95% conf. interval]

           n
         LD.     1.422765   1.019992     1.39   0.163    -.5763824    3.421913
        L2D.    -.1645517   .1300598    -1.27   0.206    -.4194643    .0903609

           w
         D1.    -.7524675   .2341305    -3.21   0.001    -1.211355     -.29358
         LD.     .9627611   .7828358     1.23   0.219    -.5715688    2.497091

           k
         D1.     .3221686   .1066645     3.02   0.003     .1131099    .5312273
         LD.    -.3248778   .3933448    -0.83   0.409    -1.095819    .4460637
        L2D.    -.0953947   .1257672    -0.76   0.448    -.3418938    .1511045

          ys
         D1.     .7660906   .3172664     2.41   0.016       .14426    1.387921
         LD.    -1.361881   .8980497    -1.52   0.129    -3.122026    .3982639
        L2D.     .3212993   .4234835     0.76   0.448     -.508713    1.151312

      yr1981
         D1.    -.0574197   .0323419    -1.78   0.076    -.1208088    .0059693

      yr1982
         D1.    -.0882952   .0580339    -1.52   0.128    -.2020395    .0254491

      yr1983
         D1.    -.1063153   .0934136    -1.14   0.255    -.2894026    .0767719

      yr1984
         D1.    -.1172108   .1150944    -1.02   0.308    -.3427917    .1083701

       _cons     .0161204    .025376     0.64   0.525    -.0336155    .0658564

     sigma_u    .29069213
     sigma_e    .34152632
         rho    .42011045   (fraction of variance due to u_i)

Endogenous: L.n
Exogenous:  L2.n w L.w k L.k L2.k ys L.ys L2.ys yr1981 yr1982 yr1983 yr1984
            L3.n
```

Stored results

xtivreg, re stores the following in e():

Scalars	
e(N)	number of observations
e(N_g)	number of groups
e(df_m)	model degrees of freedom
e(df_rz)	residual degrees of freedom
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(Tcon)	1 if panels balanced, 0 otherwise
e(N_clust)	number of clusters
e(sigma)	ancillary parameter (gamma, lnormal)
e(sigma_u)	panel-level standard deviation
e(sigma_e)	standard deviation of ϵ_{it}
e(r2_w)	R^2 for within model
e(r2_o)	R^2 for overall model
e(r2_b)	R^2 for between model
e(chi2)	χ^2
e(rho)	ρ
e(F)	model F (small only)
e(m_p)	p-value from model test
e(thta_min)	minimum θ
e(thta_5)	θ, 5th percentile
e(thta_50)	θ, 50th percentile
e(thta_95)	θ, 95th percentile
e(thta_max)	maximum θ
e(rank)	rank of e(V)
Macros	
e(cmd)	xtivreg
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(ivar)	variable denoting groups
e(tvar)	variable denoting time within groups
e(endog)	names of endogenous variables
e(exog)	names of exogenous variables
e(model)	g2sls or ec2sls
e(small)	small, if specified
e(clustvar)	name of cluster variable
e(chi2type)	Wald; type of model χ^2 test
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(properties)	b V
e(predict)	program used to implement predict
e(marginsok)	predictions allowed by margins
e(marginsnotok)	predictions disallowed by margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved
Matrices	
e(b)	coefficient vector
e(V)	variance–covariance matrix of the estimators
e(V_modelbased)	model-based variance
Functions	
e(sample)	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

`xtivreg, be` stores the following in `e()`:

Scalars

`e(N)`	number of observations
`e(N_g)`	number of groups
`e(mss)`	model sum of squares
`e(df_m)`	model degrees of freedom
`e(rss)`	residual sum of squares
`e(df_r)`	residual degrees of freedom
`e(df_rz)`	residual degrees of freedom for the between-transformed regression
`e(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(g_max)`	largest group size
`e(rs_a)`	adjusted R^2
`e(r2_w)`	R^2 for within model
`e(r2_o)`	R^2 for overall model
`e(r2_b)`	R^2 for between model
`e(N_clust)`	number of clusters
`e(chi2)`	model Wald
`e(chi2_p)`	p-value for model χ^2 test
`e(F)`	F statistic (`small` only)
`e(rmse)`	root mean squared error
`e(rank)`	rank of `e(V)`

Macros

`e(cmd)`	`xtivreg`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(ivar)`	variable denoting groups
`e(tvar)`	variable denoting time within groups
`e(endog)`	names of endogenous variables
`e(exog)`	names of exogenous variables
`e(model)`	`be`
`e(small)`	`small`, if specified
`e(clustvar)`	name of cluster variable
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. err.
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(marginsok)`	predictions allowed by `margins`
`e(marginsnotok)`	predictions disallowed by `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`

Matrices

`e(b)`	coefficient vector
`e(V)`	variance–covariance matrix of the estimators
`e(V_modelbased)`	model-based variance

Functions

`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

`xtivreg, fe` stores the following in `e()`:

Scalars

`e(N)`	number of observations
`e(N_g)`	number of groups
`e(df_m)`	model degrees of freedom
`e(rss)`	residual sum of squares
`e(df_r)`	residual degrees of freedom (`small` only)
`e(df_rz)`	residual degrees of freedom for the within-transformed regression
`e(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(g_max)`	largest group size
`e(N_clust)`	number of clusters
`e(sigma)`	ancillary parameter (`gamma`, `lnormal`)
`e(corr)`	corr(u_i, `Xb`)
`e(sigma_u)`	panel-level standard deviation
`e(sigma_e)`	standard deviation of ϵ_{it}
`e(r2_w)`	R^2 for within model
`e(r2_o)`	R^2 for overall model
`e(r2_b)`	R^2 for between model
`e(chi2)`	model Wald (not `small`)
`e(chi2_p)`	p-value for model χ^2 test
`e(rho)`	ρ
`e(F)`	F statistic (`small` only)
`e(F_f)`	F for $H_0: u_i = 0$
`e(F_fp)`	p-value for F for $H_0: u_i = 0$
`e(df_a)`	degrees of freedom for absorbed effect
`e(rank)`	rank of `e(V)`

Macros

`e(cmd)`	`xtivreg`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(ivar)`	variable denoting groups
`e(tvar)`	variable denoting time within groups
`e(endog)`	names of endogenous variables
`e(exog)`	names of exogenous variables
`e(model)`	`fe`
`e(small)`	`small`, if specified
`e(clustvar)`	name of cluster variable
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. err.
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(marginsok)`	predictions allowed by `margins`
`e(marginsnotok)`	predictions disallowed by `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`

Matrices	
e(b)	coefficient vector
e(V)	variance–covariance matrix of the estimators
e(V_modelbased)	model-based variance
Functions	
e(sample)	marks estimation sample

In addition to the above, the following is stored in r():

Matrices	
r(table)	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

xtivreg, fd stores the following in e():

Scalars	
e(N)	number of observations
e(N_g)	number of groups
e(df_m)	model degrees of freedom
e(rss)	residual sum of squares
e(df_r)	residual degrees of freedom (small only)
e(df_rz)	residual degrees of freedom for first-differenced regression
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(N_clust)	number of clusters
e(sigma)	ancillary parameter (gamma, lnormal)
e(corr)	corr(u_i, Xb)
e(sigma_u)	panel-level standard deviation
e(sigma_e)	standard deviation of ϵ_{it}
e(r2_w)	R^2 for within model
e(r2_o)	R^2 for overall model
e(r2_b)	R^2 for between model
e(chi2)	model Wald (not small)
e(chi2_p)	p-value for model χ^2 test
e(rho)	ρ
e(F)	F statistic (small only)
e(rank)	rank of e(V)
Macros	
e(cmd)	xtivreg
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(ivar)	variable denoting groups
e(tvar)	variable denoting time within groups
e(endog)	names of endogenous variables
e(exog)	names of exogenous variables
e(model)	fd
e(small)	small, if specified
e(clustvar)	name of cluster variable
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(properties)	b V
e(predict)	program used to implement predict
e(marginsok)	predictions allowed by margins

Matrices

`e(b)`	coefficient vector
`e(V)`	variance–covariance matrix of the estimators
`e(V_modelbased)`	model-based variance

Functions

`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

Consider an equation of the form

$$y_{it} = \mathbf{Y}_{it}\boldsymbol{\gamma} + \mathbf{X}_{1it}\boldsymbol{\beta} + \mu_i + \nu_{it} = \mathbf{Z}_{it}\boldsymbol{\delta} + \mu_i + \nu_{it} \tag{2}$$

where

y_{it} is the dependent variable;
$\mathbf{Y}_{it}$ is an $1 \times g_2$ vector of observations on g_2 endogenous variables included as covariates, and these variables are allowed to be correlated with the ν_{it};
$\mathbf{X}_{1it}$ is an $1 \times k_1$ vector of observations on the exogenous variables included as covariates;
$\mathbf{Z}_{it} = [\mathbf{Y}_{it}\ \mathbf{X}_{it}]$;
$\boldsymbol{\gamma}$ is a $g_2 \times 1$ vector of coefficients;
$\boldsymbol{\beta}$ is a $k_1 \times 1$ vector of coefficients; and
$\boldsymbol{\delta}$ is a $K \times 1$ vector of coefficients, where $K = g_2 + k_1$.

Assume that there is a $1 \times k_2$ vector of observations on the k_2 instruments in $\mathbf{X}_{2it}$. The order condition is satisfied if $k_2 \geq g_2$. Let $\mathbf{X}_{it} = [\mathbf{X}_{1it}\ \mathbf{X}_{2it}]$. `xtivreg` handles exogenously unbalanced panel data. Thus define T_i to be the number of observations on panel i, n to be the number of panels, and N to be the total number of observations; that is, $N = \sum_{i=1}^{n} T_i$.

Methods and formulas are presented under the following headings:

xtivreg, fd
xtivreg, fe
xtivreg, be
xtivreg, re

xtivreg, fd

As the name implies, this estimator obtains its estimates and conventional VCE from an instrumental-variables regression on the first-differenced data. Specifically, first differencing the data yields

$$y_{it} - y_{it-1} = \left(\mathbf{Z}_{it} - \mathbf{Z}_{i,t-1}\right)\delta + \nu_{it} - \nu_{i,t-1}$$

With the μ_i removed by differencing, we can obtain the estimated coefficients and their estimated variance–covariance matrix from a two-stage least-squares regression of Δy_{it} on $\Delta\mathbf{Z}_{it}$ with instruments $\Delta\mathbf{X}_{it}$.

R^2 within is reported as $\left[\text{corr}\{(\mathbf{Z}_{it} - \overline{\mathbf{Z}}_i)\widehat{\boldsymbol{\delta}}, y_{it} - \overline{y}_i\}\right]^2$.

R^2 between is reported as $\left\{\text{corr}(\overline{\mathbf{Z}}_i\widehat{\boldsymbol{\delta}}, \overline{y}_i)\right\}^2$.

R^2 overall is reported as $\left\{\text{corr}(\mathbf{Z}_{it}\widehat{\boldsymbol{\delta}}, y_{it})\right\}^2$.

xtivreg, fe

At the heart of this model is the within transformation. The within transform of a variable w is

$$\widetilde{w}_{it} = w_{it} - \overline{w}_{i.} + \overline{w}$$

where

$$\overline{w}_{i.} = \frac{1}{T_i}\sum_{t=1}^{T_i} w_{it}$$

$$\overline{w} = \frac{1}{N}\sum_{i=1}^{n}\sum_{t=1}^{T_i} w_{it}$$

and n is the number of groups and N is the total number of observations on the variable.

The within transform of (2) is

$$\widetilde{y}_{it} = \widetilde{\mathbf{Z}}_{it} + \widetilde{\nu}_{it}$$

The within transform has removed the μ_i. With the μ_i gone, the within 2SLS estimator can be obtained from a two-stage least-squares regression of $\widetilde{y}_{it}$ on $\widetilde{\mathbf{Z}}_{it}$ with instruments $\widetilde{\mathbf{X}}_{it}$.

Suppose that there are K variables in $\mathbf{Z}_{it}$, including the mandatory constant. There are $K + n - 1$ parameters estimated in the model, and the conventional VCE for the within estimator is

$$\frac{N-K}{N-n-K+1}V_{IV}$$

where V_{IV} is the VCE from the above two-stage least-squares regression. The robust and cluster–robust variance–covariance matrices are the robust and cluster–robust variance–covariance matrices from a two-stage least-squares regression of $\widetilde{y}_{it}$ on $\widetilde{\mathbf{Z}}_{it}$ with instruments $\widetilde{\mathbf{X}}_{it}$.

From the estimate of $\widehat{\boldsymbol{\delta}}$, estimates $\widehat{\mu}_i$ of μ_i are obtained as $\widehat{\mu}_i = \overline{y}_i - \overline{\mathbf{Z}}_i\widehat{\boldsymbol{\delta}}$. Reported from the calculated $\widehat{\mu}_i$ is its standard deviation and its correlation with $\overline{\mathbf{Z}}_i\widehat{\boldsymbol{\delta}}$. Reported as the standard deviation of ν_{it} is the regression's estimated root mean squared error, s^2, which is adjusted (as previously stated) for the $n-1$ estimated means.

R^2 within is reported as the R^2 from the mean-deviated regression.

R^2 between is reported as $\left\{\text{corr}(\overline{\mathbf{Z}}_i\widehat{\boldsymbol{\delta}}, \overline{y}_i)\right\}^2$.

R^2 overall is reported as $\left\{\text{corr}(\mathbf{Z}_{it}\widehat{\boldsymbol{\delta}}, y_{it})\right\}^2$.

At the bottom of the output, an F statistics against the null hypothesis that all the μ_i are zero is reported. This F statistic is an application of the results in Wooldridge (1990).

xtivreg, be

After passing (2) through the between transform, we are left with

$$\overline{y}_i = \alpha + \overline{\mathbf{Z}}_i \boldsymbol{\delta} + \mu_i + \overline{\nu}_i \tag{3}$$

where

$$\overline{w}_i = \frac{1}{T_i} \sum_{t=1}^{T_i} w_{it} \qquad \text{for } w \in \{y, \mathbf{Z}, \nu\}$$

Similarly, define $\overline{\mathbf{X}}_i$ as the matrix of instruments $\mathbf{X}_{it}$ after they have been passed through the between transform.

The BE2SLS estimator of (3) obtains its coefficient estimates and its VCE, a two-stage least-squares regression of $\overline{y}_i$ on $\overline{Z}_i$ with instruments $\overline{\mathbf{X}}_i$ in which each average appears T_i times.

R^2 between is reported as the R^2 from the fitted regression.

R^2 within is reported as $\left[\text{corr}\{(\mathbf{Z}_{it} - \overline{\mathbf{Z}}_i)\widehat{\boldsymbol{\delta}}, y_{it} - \overline{y}_i\}\right]^2$.

R^2 overall is reported as $\left\{\text{corr}(\mathbf{Z}_{it}\widehat{\boldsymbol{\delta}}, y_{it})\right\}^2$.

xtivreg, re

Per Baltagi and Chang (2000), let

$$u = \mu_i + \nu_{it}$$

be the $N \times 1$ vector of combined errors. Then under the assumptions of the random-effects model,

$$E(uu') = \sigma_\nu^2 \text{diag}\left[I_{T_i} - \frac{1}{T_i}\boldsymbol{\iota}_{T_i}\boldsymbol{\iota}_{T_i}'\right] + \text{diag}\left[w_i \frac{1}{T_i}\boldsymbol{\iota}_{T_i}\boldsymbol{\iota}_{T_i}'\right]$$

where

$$\omega_i = T_i \sigma_\mu^2 + \sigma_\nu^2$$

and $\boldsymbol{\iota}_{T_i}$ is a vector of ones of dimension T_i.

Because the variance components are unknown, consistent estimates are required to implement feasible GLS. `xtivreg` offers two choices. The default is a simple extension of the Swamy–Arora method for unbalanced panels.

Let

$$u_{it}^w = \tilde{y}_{it} - \widetilde{\mathbf{Z}}_{it}\widehat{\boldsymbol{\delta}}_w$$

be the combined residuals from the within estimator. Let $\tilde{u}_{it}$ be the within-transformed u_{it}. Then

$$\widehat{\sigma}_\nu = \frac{\sum_{i=1}^n \sum_{t=1}^{T_i} \tilde{u}_{it}^2}{N - n - K + 1}$$

Let

$$u_{it}^b = y_{it} - \mathbf{Z}_{it}\boldsymbol{\delta}_b$$

be the combined residual from the between estimator. Let $\overline{u}_{i.}^b$ be the between residuals after they have been passed through the between transform. Then

$$\widehat{\sigma}_\mu^2 = \frac{\sum_{i=1}^n \sum_{t=1}^{T_i} \overline{u}_{it}^2 - (n - K)\widehat{\sigma}_\nu^2}{N - r}$$

where

$$r = \text{trace}\left\{\left(\overline{\mathbf{Z}}_i'\overline{\mathbf{Z}}_i\right)^{-1}\overline{\mathbf{Z}}_i'\mathbf{Z}_\mu\mathbf{Z}_\mu'\overline{\mathbf{Z}}_i\right\}$$

where

$$\mathbf{Z}_\mu = \text{diag}\left(\boldsymbol{\iota}_{T_i}\boldsymbol{\iota}_{T_i}'\right)$$

If the `nosa` option is specified, the consistent estimators described in Baltagi and Chang (2000) are used. These are given by

$$\widehat{\sigma}_\nu = \frac{\sum_{i=1}^n \sum_{t=1}^{T_i} \widetilde{u}_{it}^2}{N-n}$$

and

$$\widehat{\sigma}_\mu^2 = \frac{\sum_{i=1}^n \sum_{t=1}^{T_i} \overline{u}_{it}^2 - n\widehat{\sigma}_\nu^2}{N}$$

The default Swamy–Arora method contains a degree-of-freedom correction to improve its performance in small samples.

Given estimates of the variance components, $\widehat{\sigma}_\nu^2$ and $\widehat{\sigma}_\mu^2$, the feasible GLS transform of a variable w is

$$w^* = w_{it} - \widehat{\theta}_{it}\overline{w}_{i.} \tag{4}$$

where

$$\overline{w}_{i.} = \frac{1}{T_i}\sum_{t=1}^{T_i} w_{it}$$

$$\widehat{\theta}_{it} = 1 - \left(\frac{\widehat{\sigma}_\nu^2}{\widehat{\omega}_i}\right)^{-\frac{1}{2}}$$

and

$$\widehat{\omega}_i = T_i\widehat{\sigma}_\mu^2 + \widehat{\sigma}_\nu^2$$

Using either estimator of the variance components, `xtivreg` contains two GLS estimators of the random-effects model. These two estimators differ only in how they construct the GLS instruments from the exogenous covariates and instrumental variables contained in $\mathbf{X}_{it} = [\mathbf{X}_{1it}\mathbf{X}_{2it}]$. The default method, G2SLS, which is from Balestra and Varadharajan-Krishnakumar, uses the exogenous variables after they have been passed through the feasible GLS transform. Mathematically, G2SLS uses $\mathbf{X}^*$ for the GLS instruments, where $\mathbf{X}^*$ is constructed by passing each variable in $\mathbf{X}$ though the GLS transform in (4). The G2SLS estimator obtains its coefficient estimates and VCE from an instrumental variable regression of y_{it}^* on $\mathbf{Z}_{it}^*$ with instruments $\mathbf{X}_{it}^*$.

If the `ec2sls` option is specified, `xtivreg` performs Baltagi's EC2SLS. In EC2SLS, the instruments are $\widetilde{\mathbf{X}}_it$ and $\overline{\mathbf{X}}_{it}$, where $\widetilde{X}_{it}$ is constructed by each of the variables in $\mathbf{X}_{it}$ throughout the GLS transform in (4), and $\overline{\mathbf{X}}_{it}$ is made of the group means of each variable in $\mathbf{X}_{it}$. The EC2SLS estimator obtains its coefficient estimates and its VCE from an instrumental variables regression of y_{it}^* on $\mathbf{Z}_{it}^*$ with instruments $\widetilde{\mathbf{X}}_{it}$ and $\overline{\mathbf{X}}_{it}$.

Baltagi and Li (1992) show that although the G2SLS instruments are a subset of those in EC2SLS, the extra instruments in EC2SLS are redundant in the sense of White (2001). Given the extra computational cost, G2SLS is the default.

The standard deviation of $\mu_i + \nu_{it}$ is calculated as $\sqrt{\hat{\sigma}_\mu^2 + \hat{\sigma}_\nu^2}$.

R^2 between is reported as $\left\{\mathrm{corr}(\overline{\mathbf{Z}}_i\hat{\boldsymbol{\delta}}, \overline{y}_i)\right\}^2$.

R^2 within is reported as $\left[\mathrm{corr}\{(\mathbf{Z}_{it} - \overline{\mathbf{Z}}_i)\hat{\boldsymbol{\delta}}, y_{it} - \overline{y}_i\}\right]^2$.

R^2 overall is reported as $\left\{\mathrm{corr}(\mathbf{Z}_{it}\hat{\boldsymbol{\delta}}, y_{it})\right\}^2$.

Acknowledgment

We thank Mead Over of the Center for Global Development, who wrote an early implementation of `xtivreg`.

References

Amemiya, T. 1971. The estimation of the variances in a variance-components model. *International Economic Review* 12: 1–13. https://doi.org/10.2307/2525492.

Anderson, T. W., and C. Hsiao. 1981. Estimation of dynamic models with error components. *Journal of the American Statistical Association* 76: 598–606. https://doi.org/10.2307/2287517.

Arellano, M., and S. Bond. 1991. Some tests of specification for panel data: Monte Carlo evidence and an application to employment equations. *Review of Economic Studies* 58: 277–297. https://doi.org/10.2307/2297968.

Balestra, P., and J. Varadharajan-Krishnakumar. 1987. Full information estimations of a system of simultaneous equations with error component structure. *Econometric Theory* 3: 223–246. https://doi.org/10.1017/S0266466600010318.

Baltagi, B. H. 2009. *A Companion to Econometric Analysis of Panel Data*. Chichester, UK: Wiley.

———. 2013. *Econometric Analysis of Panel Data*. 5th ed. Chichester, UK: Wiley.

Baltagi, B. H., and Y.-J. Chang. 1994. Incomplete panels: A comparative study of alternative estimators for the unbalanced one-way error component regression model. *Journal of Econometrics* 62: 67–89. https://doi.org/10.1016/0304-4076(94)90017-5.

———. 2000. Simultaneous equations with incomplete panels. *Econometric Theory* 16: 269–279. https://doi.org/10.1017/S0266466600162073.

Baltagi, B. H., and Q. Li. 1992. A note on the estimation of simultaneous equations with error components. *Econometric Theory* 8: 113–119. https://doi.org/10.1017/S0266466600010768.

Du, K., Y. Zhang, and Q. Zhou. 2020. Fitting partially linear functional-coefficient panel-data models with Stata. *Stata Journal* 20: 976–998.

Hsiao, C. 2014. *Analysis of Panel Data*. 3rd ed. New York: Cambridge University Press.

Layard, R., and S. J. Nickell. 1986. Unemployment in Britain. *Economica* 53: S121–S169. https://doi.org/10.2307/2554377.

Mundlak, Y. 1978. On the pooling of time series and cross section data. *Econometrica* 46: 69–85. https://doi.org/10.2307/1913646.

Swamy, P. A. V. B., and S. S. Arora. 1972. The exact finite sample properties of the estimators of coefficients in the error components regression models. *Econometrica* 40: 261–275. https://doi.org/10.2307/1909405.

White, H. L., Jr. 2001. *Asymptotic Theory for Econometricians*. Rev. ed. New York: Academic Press. https://doi.org/10.1016/C2009-0-21869-1.

Wooldridge, J. M. 1990. A note on the Lagrange multiplier and F-statistics for two stage least squares regressions. *Economics Letters* 34: 151–155. https://doi.org/10.1016/0165-1765(90)90236-T.

Also see

[XT] **xtivreg postestimation** — Postestimation tools for xtivreg

[XT] **xtabond** — Arellano–Bond linear dynamic panel-data estimation

[XT] **xteregress** — Extended random-effects linear regression

[XT] **xthtaylor** — Hausman–Taylor estimator for error-components model

[XT] **xtreg** — Linear models for panel data

[XT] **xtset** — Declare data to be panel data

[R] **ivregress** — Single-equation instrumental-variables regression

[U] 20 Estimation and postestimation commands

xtivreg postestimation — Postestimation tools for xtivreg

Postestimation commands   predict   margins   Also see

Postestimation commands

The following postestimation commands are available after `xtivreg`:

Command	Description
* `contrast`	contrasts and ANOVA-style joint tests of parameters
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
`forecast`	dynamic forecasts and simulations
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	linear predictions, first-differenced error components
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
* `pwcompare`	pairwise comparisons of parameters
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

*`contrast` and `pwcompare` are not appropriate after `xtivreg, fd`.

predict

Description for predict

`predict` creates a new variable containing predictions such as fitted values and predictions.

Menu for predict

Statistics > Postestimation

Syntax for predict

For all but the first-differenced estimator

`predict` [*type*] *newvar* [*if*] [*in*] [, *statistic*]

First-differenced estimator

`predict` [*type*] *newvar* [*if*] [*in*] [, *FD_statistic*]

statistic	Description
Main	
`xb`	$\mathbf{Z}_{it}\widehat{\boldsymbol{\delta}}$, fitted values; the default
`ue`	$\widehat{\mu}_i + \widehat{\nu}_{it}$, the combined residual
* `xbu`	$\mathbf{Z}_{it}\widehat{\boldsymbol{\delta}} + \widehat{\mu}_i$, prediction including effect
* `u`	$\widehat{\mu}_i$, the fixed- or random-error component
* `e`	$\widehat{\nu}_{it}$, the overall error component

Unstarred statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample. Starred statistics are calculated only for the estimation sample, even when `if e(sample)` is not specified.

FD_statistic	Description
Main	
`xb`	$\mathbf{x}_j\mathbf{b}$, fitted values for the first-differenced model; the default
`e`	$e_{it} - e_{it-1}$, the first-differenced overall error component

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

Options for predict

Main

`xb`, the default, calculates the linear prediction, that is, $\mathbf{Z}_{it}\widehat{\boldsymbol{\delta}}$.

`ue` calculates the prediction of $\widehat{\mu}_i + \widehat{\nu}_{it}$. This is not available after the first-differenced model.

`xbu` calculates the prediction of $\mathbf{Z}_{it}\widehat{\boldsymbol{\delta}} + \widehat{\mu}_i$, the prediction including the fixed or random component. This is not available after the first-differenced model.

`u` calculates the prediction of $\hat{\mu}_i$, the estimated fixed or random effect. This is not available after the first-differenced model.

`e` calculates the prediction of $\hat{\nu}_{it}$.

margins

Description for margins

`margins` estimates margins of response for fitted values.

Menu for margins

Statistics > Postestimation

Syntax for margins

`margins` [*marginlist*] [, *options*]

`margins` [*marginlist*] , `predict(`*statistic* ...`)` [*options*]

For all but the first-differenced estimator

statistic	Description
`xb`	$\mathbf{Z}_{it}\hat{\boldsymbol{\delta}}$, fitted values; the default
`ue`	not allowed with `margins`
`xbu`	not allowed with `margins`
`u`	not allowed with `margins`
`e`	not allowed with `margins`

First-differenced estimator

statistic	Description
`xb`	$\mathbf{x}_j\mathbf{b}$, fitted values for the first-differenced model; the default
`e`	not allowed with `margins`

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.

For the full syntax, see [R] **margins**.

Also see

[XT] **xtivreg** — Instrumental variables and two-stage least squares for panel-data models

[U] 20 Estimation and postestimation commands

xtline — Panel-data line plots

Description

`xtline` draws line plots for panel data.

Quick start

Matrix of line plots of `y` against time variable `tvar` with panel identifier `pvar`

```
xtline y, i(pvar) t(tvar)
```

Same as above, but using `xtset` data

```
xtline y
```

Same as above, but overlay line plots for each panel identifier `pvar`

```
xtline y, overlay
```

Add “My Title” to graph showing a matrix of line plots

```
xtline y, byopts(title(My Title))
```

Add “My Title” to graph of overlaid line plots

```
xtline y, overlay title(My Title)
```

Menu

Statistics > Longitudinal/panel data > Line plots

Syntax

Graph by panel

`xtline` *varlist* [*if*] [*in*] [, *panel_options*]

Overlaid panels

`xtline` *varname* [*if*] [*in*] , `overlay` [*overlaid_options*]

panel_options	Description
Main	
`i(`*varname*$_i$`)`	use *varname*$_i$ as the panel ID variable
`t(`*varname*$_t$`)`	use *varname*$_t$ as the time variable
Plot	
cline_options	affect rendition of the plotted points connected by lines
Add plots	
`addplot(`*plot*`)`	add other plots to the generated graph
Y axis, Time axis, Titles, Legend, Overall	
twoway_options	any options other than `by()` documented in [G-3] ***twoway_options***
`byopts(`*byopts*`)`	affect appearance of the combined graph

overlaid_options	Description
Main	
`overlay`	overlay each panel on the same graph
`i(`*varname*$_i$`)`	use *varname*$_i$ as the panel ID variable
`t(`*varname*$_t$`)`	use *varname*$_t$ as the time variable
Plots	
`plot#opts(`*cline_options*`)`	affect rendition of the # panel line
Add plots	
`addplot(`*plot*`)`	add other plots to the generated graph
Y axis, Time axis, Titles, Legend, Overall	
twoway_options	any options other than `by()` documented in [G-3] ***twoway_options***

A panel variable and a time variable must be specified. Use `xtset` (see [XT] **xtset**) or specify the `i()` and `t()` options. The `t()` option allows noninteger values for the time variable, whereas `xtset` does not.

Options for graph by panel

Main

i($varname_i$) and t($varname_t$) override the panel settings from xtset; see [XT] **xtset**. $varname_i$ is allowed to be a string variable. $varname_t$ can take on noninteger values and have repeated values within panel. That is to say, it can be any numeric variable that you would like to specify for the x-dimension of the graph. It is an error to specify i() without t() and vice versa.

Plot

cline_options affect the rendition of the plotted points connected by lines; see [G-3] ***cline_options***.

Add plots

addplot(*plot*) provides a way to add other plots to the generated graph; see [G-3] ***addplot_option***.

Y axis, Time axis, Titles, Legend, Overall

twoway_options are any of the options documented in [G-3] ***twoway_options***, excluding by(). These include options for titling the graph (see [G-3] ***title_options***) and for saving the graph to disk (see [G-3] ***saving_option***).

byopts(*byopts*) allows all the options documented in [G-3] ***by_option***. These options affect the appearance of the by-graph. byopts() may not be combined with overlay.

Options for overlaid panels

Main

overlay causes the plot from each panel to be overlaid on the same graph. The default is to generate plots by panel. This option may not be combined with byopts() or be specified when there are multiple variables in *varlist*.

i($varname_i$) and t($varname_t$) override the panel settings from xtset; see [XT] **xtset**. $varname_i$ is allowed to be a string variable. $varname_t$ can take on noninteger values and have repeated values within panel. That is to say, it can be any numeric variable that you would like to specify for the x-dimension of the graph. It is an error to specify i() without t() and vice versa.

Plots

plot#opts(*cline_options*) affect the rendition of the #th panel (in sorted order). The *cline_options* can affect whether and how the points are connected; see [G-3] ***cline_options***.

Add plots

addplot(*plot*) provides a way to add other plots to the generated graph; see [G-3] ***addplot_option***.

Y axis, Time axis, Titles, Legend, Overall

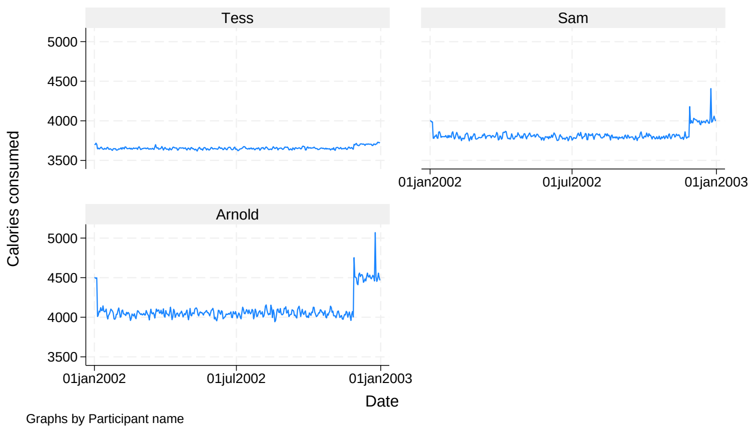
twoway_options are any of the options documented in [G-3] ***twoway_options***, excluding by(). These include options for titling the graph (see [G-3] ***title_options***) and for saving the graph to disk (see [G-3] ***saving_option***).

Remarks and examples

▷ Example 1

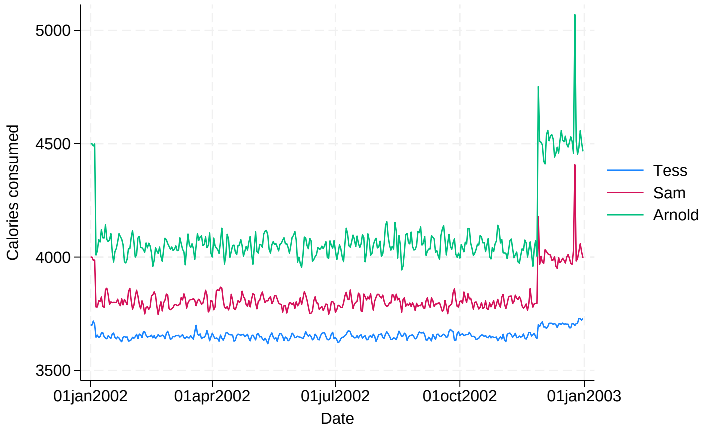
Suppose that Tess, Sam, and Arnold kept a calorie log for an entire calendar year. At the end of the year, if they pooled their data together, they would have a dataset (for example, `xtline1.dta`) that contains the number of calories each of them consumed for 365 days. They could then use `xtset` to identify the date variable and treat each person as a panel and use `xtline` to plot the calories versus time for each person separately.

```
. use https://www.stata-press.com/data/r19/xtline1
(Simulated data of calories consumed for 365 days)
. xtset person day
Panel variable: person (strongly balanced)
 Time variable: day, 01jan2002 to 31dec2002
         Delta: 1 day
. xtline calories, tlabel(#3)
```



Specify the `overlay` option so that the values are plotted on the same graph to provide a better comparison among Tess, Sam, and Arnold.

```
. xtline calories, overlay
```



◁

References

Chatfield, M. D. 2018. Graphing each individual's data over time. *Stata Journal* 18: 503–516.

Gallup, J. L. 2020. Added-variable plots for panel-data estimation. *Stata Journal* 20: 30–50.

Also see

[XT] **xtset** — Declare data to be panel data

[G-2] **graph twoway** — Two-way graphs

[TS] **tsline** — Time-series line plots

Description

`xtlogit` fits random-effects, conditional fixed-effects, and population-averaged logit models for a binary dependent variable. The probability of a positive outcome is assumed to be determined by the logistic cumulative distribution function. Results may be reported as coefficients or odds ratios.

Quick start

Random-effects model of `y` as a function of `x1`, `x2`, and indicators for levels of categorical variable `a` using `xtset` data

```
xtlogit y x1 x2 i.a
```

Same as above, but report odds ratios

```
xtlogit y x1 x2 i.a, or
```

Conditional fixed-effects model

```
xtlogit y x1 x2 i.a, fe
```

Population-averaged model with robust standard errors

```
xtlogit y x1 x2 i.a, pa vce(robust)
```

Random-effects model with cluster–robust standard errors for panels nested within `cvar`

```
xtlogit y x1 x2 i.a, vce(cluster cvar)
```

Menu

Statistics > Longitudinal/panel data > Binary outcomes > Logistic regression (FE, RE, PA)

Syntax

Random-effects (RE) model

`xtlogit` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, `re` *RE_options*]

Conditional fixed-effects (FE) model

`xtlogit` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] , `fe` [*FE_options*]

Population-averaged (PA) model

`xtlogit` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] , `pa` [*PA_options*]

RE_options	Description
Model	
`noconstant`	suppress constant term
`re`	use random-effects estimator; the default
`offset(`*varname*`)`	include *varname* in model with coefficient constrained to 1
`constraints(`*constraints*`)`	apply specified linear constraints
`asis`	retain perfect predictor variables
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `robust`, `cluster` *clustvar*, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`or`	report odds ratios
`lrmodel`	perform the likelihood-ratio model test instead of the default Wald test
`nocnsreport`	do not display constraints
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Integration	
`intmethod(`*intmethod*`)`	integration method; *intmethod* may be `mvaghermite` (the default) or `ghermite`
`intpoints(#)`	use # quadrature points; default is `intpoints(12)`
Maximization	
maximize_options	control the maximization process; seldom used
`nodisplay`	suppress display of header and coefficients
`collinear`	keep collinear variables
`coeflegend`	display legend instead of statistics

FE_options	Description
Model	
`fe`	use fixed-effects estimator
`offset(`*varname*`)`	include *varname* in model with coefficient constrained to 1
`constraints(`*constraints*`)`	apply specified linear constraints
SE	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`or`	report odds ratios
`lrmodel`	perform the likelihood-ratio model test instead of the default Wald test
`nocnsreport`	do not display constraints
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
maximize_options	control the maximization process; seldom used
`nodisplay`	suppress display of header and coefficients
`collinear`	keep collinear variables
`coeflegend`	display legend instead of statistics

PA_options	Description
Model	
noconstant	suppress constant term
pa	use population-averaged estimator
offset(*varname*)	include *varname* in model with coefficient constrained to 1
asis	retain perfect predictor variables
Correlation	
corr(*correlation*)	within-panel correlation structure
force	estimate even if observations unequally spaced in time
SE/Robust	
vce(*vcetype*)	*vcetype* may be conventional, robust, bootstrap, or jackknife
nmp	use divisor $N - P$ instead of the default N
scale(*parm*)	overrides the default scale parameter; *parm* may be x2, dev, phi, or #
Reporting	
level(#)	set confidence level; default is level(95)
or	report odds ratios
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Optimization	
optimize_options	control the optimization process; seldom used
nodisplay	suppress display of header and coefficients
coeflegend	display legend instead of statistics

correlation	Description
`exchangeable`	exchangeable
`independent`	independent
`unstructured`	unstructured
`fixed` *matname*	user-specified
`ar` #	autoregressive of order #
`stationary` #	stationary of order #
`nonstationary` #	nonstationary of order #

A panel variable must be specified. For `xtlogit, pa`, correlation structures other than `exchangeable` and `independent` require that a time variable also be specified. Use `xtset`; see [XT] **xtset**.

indepvars may contain factor variables; see [U] **11.4.3 Factor variables**.

depvar and *indepvars* may contain time-series operators; see [U] **11.4.4 Time-series varlists**.

`by`, `collect`, `mi estimate`, and `statsby` are allowed; see [U] **11.1.10 Prefix commands**. `bayes` is allowed for the random-effects model. For more details, see [BAYES] **bayes: xtlogit**. `fp` is allowed for the random-effects and fixed-effects models.

`vce(bootstrap)` and `vce(jackknife)` are not allowed with the `mi estimate` prefix; see [MI] **mi estimate**.

`iweights`, `fweights`, and `pweights` are allowed for the population-averaged model, and `iweights` are allowed for the fixed-effects and random-effects models; see [U] **11.1.6 weight**. Weights must be constant within panel.

`nodisplay`, `collinear`, and `coeflegend` do not appear in the dialog box.

See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options for RE model

Model

`noconstant`; see [R] **Estimation options**.

`re` requests the random-effects estimator, which is the default.

`offset(`*varname*`) constraints(`*constraints*`)`; see [R] **Estimation options**.

`asis` forces retention of perfect predictor variables and their associated, perfectly predicted observations and may produce instabilities in maximization; see [R] **probit**.

SE/Robust

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`oim`), that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster` *clustvar*), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [XT] ***vce_options***.

Specifying `vce(robust)` is equivalent to specifying `vce(cluster` *panelvar*`)`; see *xtlogit, re and the robust VCE estimator* in *Methods and formulas*.

Reporting

`level(#)`; see [R] **Estimation options**.

`or` reports the estimated coefficients transformed to odds ratios, that is, e^b rather than b. Standard errors and confidence intervals are similarly transformed. This option affects how results are displayed, not how they are estimated. `or` may be specified at estimation or when replaying previously estimated results.

lrmodel, nocnsreport; see [R] **Estimation options**.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Integration

intmethod(*intmethod*), intpoints(#); see [R] **Estimation options**.

Maximization

maximize_options: difficult, technique(*algorithm_spec*), iterate(#), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), nonrtolerance, and from(*init_specs*); see [R] **Maximize**. These options are seldom used.

The following options are available with xtlogit but are not shown in the dialog box:

nodisplay is for programmers. It suppresses the display of the header and the coefficients.

collinear, coeflegend; see [R] **Estimation options**.

Options for FE model

Model

fe requests the fixed-effects estimator.

offset(*varname*), constraints(*constraints*); see [R] **Estimation options**.

SE

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (oim) and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] ***vce_options***.

Reporting

level(#); see [R] **Estimation options**.

or reports the estimated coefficients transformed to odds ratios, that is, e^b rather than b. Standard errors and confidence intervals are similarly transformed. This option affects how results are displayed, not how they are estimated. or may be specified at estimation or when replaying previously estimated results.

lrmodel, nocnsreport; see [R] **Estimation options**.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Maximization

maximize_options: difficult, technique(*algorithm_spec*), iterate(#), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), nonrtolerance, and from(*init_specs*); see [R] **Maximize**. These options are seldom used.

The following options are available with xtlogit but are not shown in the dialog box:

nodisplay is for programmers. It suppresses the display of the header and the coefficients.

collinear, coeflegend; see [R] **Estimation options**.

Options for PA model

Model

noconstant; see [R] **Estimation options**.

pa requests the population-averaged estimator.

offset(*varname*); see [R] **Estimation options**.

asis forces retention of perfect predictor variables and their associated, perfectly predicted observations and may produce instabilities in maximization; see [R] **probit**.

Correlation

corr(*correlation*) specifies the within-panel correlation structure; the default corresponds to the equal-correlation model, corr(exchangeable).

When you specify a correlation structure that requires a lag, you indicate the lag after the structure's name with or without a blank; for example, corr(ar 1) or corr(ar1).

If you specify the fixed correlation structure, you specify the name of the matrix containing the assumed correlations following the word fixed, for example, corr(fixed myr).

force specifies that estimation be forced even though the time variable is not equally spaced. This is relevant only for correlation structures that require knowledge of the time variable. These correlation structures require that observations be equally spaced so that calculations based on lags correspond to a constant time change. If you specify a time variable indicating that observations are not equally spaced, the (time dependent) model will not be fit. If you also specify force, the model will be fit, and it will be assumed that the lags based on the data ordered by the time variable are appropriate.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (conventional), that are robust to some kinds of misspecification (robust), and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] ***vce_options***.

vce(conventional), the default, uses the conventionally derived variance estimator for generalized least-squares regression.

nmp, scale(x2 | dev | phi | #); see [XT] ***vce_options***.

Reporting

level(#); see [R] **Estimation options**.

or reports the estimated coefficients transformed to odds ratios, that is, e^b rather than b. Standard errors and confidence intervals are similarly transformed. This option affects how results are displayed, not how they are estimated. or may be specified at estimation or when replaying previously estimated results.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Optimization

optimize_options control the iterative optimization process. These options are seldom used.

iterate(#) specifies the maximum number of iterations. When the number of iterations equals #, the optimization stops and presents the current results, even if convergence has not been reached. The default is iterate(100).

tolerance(#) specifies the tolerance for the coefficient vector. When the relative change in the coefficient vector from one iteration to the next is less than or equal to #, the optimization process is stopped. tolerance(1e-6) is the default.

log and nolog specify whether to display the iteration log. The iteration log is displayed by default unless you used set iterlog off to suppress it; see set iterlog in [R] ***set iter***.

trace specifies that the current estimates be printed at each iteration.

The following options are available with xtlogit but are not shown in the dialog box:

nodisplay is for programmers. It suppresses the display of the header and the coefficients.

coeflegend; see [R] **Estimation options**.

Remarks and examples

xtlogit fits random-effects, conditional fixed-effects, and population-averaged logit models. Whenever we refer to a fixed-effects model, we mean the conditional fixed-effects model. *depvar* equal to nonzero and nonmissing (typically *depvar* equal to one) indicates a positive outcome, whereas *depvar* equal to zero indicates a negative outcome.

By default, the population-averaged model is an equal-correlation model; that is xtlogit, pa assumes corr(exchangeable). Thus, xtlogit is a convenience command for fitting the population-averaged model using xtgee; see [XT] **xtgee**. Typing

```
. xtlogit ..., pa ...
```

is equivalent to typing

```
. xtgee ..., ... family(binomial) link(logit) corr(exchangeable)
```

It is also a convenience command if you want the fixed-effects model. Typing

```
. xtlogit ..., fe ...
```

is equivalent to typing

```
. clogit ..., group(varname_i) ...
```

See also [XT] **xtgee** and [R] **clogit** for information about xtlogit.

By default or when re is specified, xtlogit fits via maximum likelihood the random-effects model

$$\Pr(y_{it} \neq 0|\mathbf{x}_{it}) = P(\mathbf{x}_{it}\boldsymbol{\beta} + \nu_i)$$

for $i = 1, \ldots, n$ panels, where $t = 1, \ldots, n_i$, ν_i are i.i.d., $N(0, \sigma_\nu^2)$, and $P(z) = \{1 + \exp(-z)\}^{-1}$.

Underlying this model is the variance components model

$$y_{it} \neq 0 \iff \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i + \epsilon_{it} > 0$$

where ϵ_{it} are i.i.d. logistic distributed with mean zero and variance $\sigma_\epsilon^2 = \pi^2/3$, independently of ν_i.

▷ Example 1

We are studying unionization of women in the United States and are using the union dataset; see [XT] **xt**. We wish to fit a random-effects model of union membership:

```
. use https://www.stata-press.com/data/r19/union
(NLS Women 14-24 in 1968)
. xtlogit union age grade not_smsa south##c.year
  (output omitted)
Random-effects logistic regression               Number of obs    = 26,200
Group variable: idcode                           Number of groups =  4,434
Random effects u_i ~ Gaussian                    Obs per group:
                                                              min =      1
                                                              avg =    5.9
                                                              max =     12
Integration method: mvaghermite                  Integration pts. =     12
                                                 Wald chi2(6)     = 227.46
Log likelihood = -10540.274                      Prob > chi2      = 0.0000
```

union	Coefficient	Std. err.	z	P>|z|	[95% conf. interval]	
age	.0156732	.0149895	1.05	0.296	-.0137056	.045052
grade	.0870851	.0176476	4.93	0.000	.0524965	.1216738
not_smsa	-.2511884	.0823508	-3.05	0.002	-.4125929	-.0897839
1.south	-2.839112	.6413116	-4.43	0.000	-4.096059	-1.582164
year	-.0068604	.0156575	-0.44	0.661	-.0375486	.0238277
south#c.year						
1	.0238506	.0079732	2.99	0.003	.0082235	.0394777
_cons	-3.009365	.8414963	-3.58	0.000	-4.658667	-1.360062
/lnsig2u	1.749366	.0470017			1.657245	1.841488
sigma_u	2.398116	.0563577			2.290162	2.511158
rho	.6361098	.0108797			.6145307	.6571548

```
LR test of rho=0: chibar2(01) = 6004.43                 Prob >= chibar2 = 0.000
```

The output includes the additional panel-level variance component. This is parameterized as the log of the variance $\ln(\sigma_\nu^2)$ (labeled `lnsig2u` in the output). The standard deviation σ_ν is also included in the output and labeled `sigma_u` together with ρ (labeled `rho`),

$$\rho = \frac{\sigma_\nu^2}{\sigma_\nu^2 + \sigma_\epsilon^2}$$

which is the proportion of the total variance contributed by the panel-level variance component.

When `rho` is zero, the panel-level variance component is unimportant, and the panel estimator is no different from the pooled estimator. A likelihood-ratio test of this is included at the bottom of the output. This test formally compares the pooled estimator (logit) with the panel estimator.

As an alternative to the random-effects specification, we might want to fit an equal-correlation logit model:

```
. xtlogit union age grade not_smsa south##c.year, pa
Iteration 1:   Tolerance = .14878775
Iteration 2:   Tolerance = .00949339
Iteration 3:   Tolerance = .00040606
Iteration 4:   Tolerance = .00001602
Iteration 5:   Tolerance = 6.628e-07

GEE population-averaged model                       Number of obs     = 26,200
Group variable: idcode                              Number of groups  =  4,434
Family: Binomial                                    Obs per group:
Link:   Logit                                                     min =      1
Correlation: exchangeable                                         avg =    5.9
                                                                  max =     12
                                                    Wald chi2(6)      = 235.08
Scale parameter = 1                                 Prob > chi2       = 0.0000

------------------------------------------------------------------------------
       union | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
         age |   .0165893   .0092229     1.80   0.072    -.0014873    .0346659
       grade |   .0600669   .0108343     5.54   0.000     .0388321    .0813016
    not_smsa |  -.1215445   .0483713    -2.51   0.012    -.2163505   -.0267384
     1.south |  -1.857094    .372967    -4.98   0.000    -2.588096   -1.126092
        year |  -.0121168   .0095707    -1.27   0.205     -.030875    .0066413
             |
south#c.year |
          1  |   .0160193   .0046076     3.48   0.001     .0069886    .0250501
             |
       _cons |   -1.39755   .5089508    -2.75   0.006    -2.395075   -.4000247
------------------------------------------------------------------------------
```

◁

▷ Example 2

`xtlogit` with the pa option allows a `vce(robust)` option, so we can obtain the population-averaged logit estimator with the robust variance calculation by typing

```
. xtlogit union age grade not_smsa south##c.year, pa vce(robust) nolog
GEE population-averaged model                       Number of obs     = 26,200
Group variable: idcode                              Number of groups =  4,434
Family: Binomial                                    Obs per group:
Link:   Logit                                                    min =      1
Correlation: exchangeable                                        avg =    5.9
                                                                 max =     12
                                                    Wald chi2(6)      = 154.88
Scale parameter = 1                                 Prob > chi2       = 0.0000
                                 (Std. err. adjusted for clustering on idcode)
------------------------------------------------------------------------------
             |               Robust
       union | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
         age |   .0165893    .008951     1.85   0.064    -.0009543    .0341329
       grade |   .0600669   .0133193     4.51   0.000     .0339616    .0861722
    not_smsa |  -.1215445   .0613803    -1.98   0.048    -.2418477   -.0012412
     1.south |  -1.857094   .5389238    -3.45   0.001    -2.913366   -.8008231
        year |  -.0121168   .0096998    -1.25   0.212    -.0311282    .0068945
             |
south#c.year |
           1 |   .0160193   .0067217     2.38   0.017      .002845    .0291937
             |
       _cons |   -1.39755   .5603767    -2.49   0.013    -2.495868   -.2992317
------------------------------------------------------------------------------
```

These standard errors are somewhat larger than those obtained without the `vce(robust)` option.

Finally, we can also fit a fixed-effects model to these data (see also [R] **clogit** for details):

```
. xtlogit union age grade not_smsa south##c.year, fe
note: multiple positive outcomes within groups encountered.
note: 2,744 groups (14,165 obs) omitted because of all positive or
      all negative outcomes.
Iteration 0:  Log likelihood = -4516.5881
Iteration 1:  Log likelihood = -4510.8906
Iteration 2:  Log likelihood =  -4510.888
Iteration 3:  Log likelihood =  -4510.888
Conditional fixed-effects logistic regression        Number of obs     = 12,035
Group variable: idcode                               Number of groups =  1,690
                                                     Obs per group:
                                                                  min =      2
                                                                  avg =    7.1
                                                                  max =     12
                                                     LR chi2(6)        =  78.60
Log likelihood = -4510.888                           Prob > chi2       = 0.0000
```

union	Coefficient	Std. err.	z	P>|z|	[95% conf. interval]	
age	.0710973	.0960536	0.74	0.459	-.1171643	.2593589
grade	.0816111	.0419074	1.95	0.051	-.0005259	.163748
not_smsa	.0224809	.1131786	0.20	0.843	-.199345	.2443069
1.south	-2.856488	.6765694	-4.22	0.000	-4.182539	-1.530436
year	-.0636853	.0967747	-0.66	0.510	-.2533602	.1259896
south#c.year						
1	.0264136	.0083216	3.17	0.002	.0101036	.0427235

◁

❑ Technical note

The random-effects model is calculated using quadrature, which is an approximation whose accuracy depends partially on the number of integration points used. We can use the `quadchk` command to see if changing the number of integration points affects the results. If the results change, the quadrature approximation is not accurate given the number of integration points. Try increasing the number of integration points using the `intpoints()` option and run `quadchk` again. Do not attempt to interpret the results of estimates when the coefficients reported by `quadchk` differ substantially. See [XT] **quadchk** for details and [XT] **xtprobit** for an example.

Because the `xtlogit` likelihood function is calculated by Gauss–Hermite quadrature, on large problems the computations can be slow. Computation time is roughly proportional to the number of points used for the quadrature.

❑

Stored results

`xtlogit, re` stores the following in `e()`:

Scalars

`e(N)`	number of observations
`e(N_g)`	number of groups
`e(k)`	number of parameters
`e(k_aux)`	number of auxiliary parameters
`e(k_eq)`	number of equations in `e(b)`
`e(k_eq_model)`	number of equations in overall model test
`e(k_dv)`	number of dependent variables
`e(df_m)`	model degrees of freedom
`e(ll)`	log likelihood
`e(ll_0)`	log likelihood, constant-only model
`e(ll_c)`	log likelihood, comparison model
`e(chi2)`	χ^2
`e(chi2_c)`	χ^2 for comparison test
`e(N_clust)`	number of clusters
`e(rho)`	ρ
`e(sigma_u)`	panel-level standard deviation
`e(n_quad)`	number of quadrature points
`e(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(g_max)`	largest group size
`e(p)`	*p*-value for model test
`e(rank)`	rank of `e(V)`
`e(rank0)`	rank of `e(V)` for constant-only model
`e(ic)`	number of iterations
`e(rc)`	return code
`e(converged)`	`1` if converged, `0` otherwise

Macros

`e(cmd)`	`xtlogit`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(ivar)`	variable denoting groups
`e(model)`	`re`
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(title)`	title in estimation output
`e(clustvar)`	name of cluster variable
`e(offset)`	linear offset variable
`e(chi2type)`	`Wald` or `LR`; type of model χ^2 test
`e(chi2_ct)`	`Wald` or `LR`; type of model χ^2 test corresponding to `e(chi2_c)`
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. err.
`e(intmethod)`	integration method
`e(distrib)`	`Gaussian`; the distribution of the random effect
`e(opt)`	type of optimization
`e(which)`	`max` or `min`; whether optimizer is to perform maximization or minimization
`e(ml_method)`	type of `ml` method
`e(user)`	name of likelihood-evaluator program
`e(technique)`	maximization technique
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(marginsdefault)`	default `predict()` specification for `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`

Matrices	
e(b)	coefficient vector
e(Cns)	constraints matrix
e(ilog)	iteration log
e(gradient)	gradient vector
e(V)	variance–covariance matrix of the estimators
e(V_modelbased)	model-based variance
Functions	
e(sample)	marks estimation sample

In addition to the above, the following is stored in r():

Matrices	
r(table)	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

xtlogit, fe stores the following in e():

Scalars	
e(N)	number of observations
e(N_g)	number of groups
e(N_drop)	number of observations dropped because of all positive or all negative outcomes
e(N_group_drop)	number of groups dropped because of all positive or all negative outcomes
e(k)	number of parameters
e(k_eq)	number of equations in e(b)
e(k_eq_model)	number of equations in overall model test
e(k_dv)	number of dependent variables
e(df_m)	model degrees of freedom
e(r2_p)	pseudo-R^2
e(ll)	log likelihood
e(ll_0)	log likelihood, constant-only model
e(chi2)	χ^2
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(p)	p-value for model test
e(rank)	rank of e(V)
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if converged, 0 otherwise
Macros	
e(cmd)	clogit
e(cmd2)	xtlogit
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(ivar)	variable denoting groups
e(model)	fe
e(wtype)	weight type
e(wexp)	weight expression
e(title)	title in estimation output
e(offset)	linear offset variable
e(chi2type)	LR; type of model χ^2 test
e(vce)	*vcetype* specified in vce()
e(group)	name of group() variable
e(multiple)	multiple if multiple positive outcomes within groups

`e(opt)`	type of optimization
`e(which)`	`max` or `min`; whether optimizer is to perform maximization or minimization
`e(ml_method)`	type of `ml` method
`e(user)`	name of likelihood-evaluator program
`e(technique)`	maximization technique
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(marginsok)`	predictions allowed by `margins`
`e(marginsnotok)`	predictions disallowed by `margins`
`e(marginsdefault)`	default `predict()` specification for `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`

Matrices

`e(b)`	coefficient vector
`e(Cns)`	constraints matrix
`e(ilog)`	iteration log
`e(gradient)`	gradient vector
`e(V)`	variance–covariance matrix of the estimators

Functions

`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

`xtlogit, pa` stores the following in `e()`:

Scalars

`e(N)`	number of observations
`e(N_g)`	number of groups
`e(df_m)`	model degrees of freedom
`e(chi2)`	χ^2
`e(p)`	p-value for model test
`e(df_pear)`	degrees of freedom for Pearson χ^2
`e(chi2_dev)`	χ^2 test of deviance
`e(chi2_dis)`	χ^2 test of deviance dispersion
`e(deviance)`	deviance
`e(dispers)`	deviance dispersion
`e(phi)`	scale parameter
`e(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(g_max)`	largest group size
`e(rank)`	rank of `e(V)`
`e(tol)`	target tolerance
`e(dif)`	achieved tolerance
`e(rc)`	return code

Macros

`e(cmd)`	`xtgee`
`e(cmd2)`	`xtlogit`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(ivar)`	variable denoting groups
`e(tvar)`	variable denoting time within groups

`e(model)`	`pa`
`e(family)`	`binomial`
`e(link)`	`logit`; link function
`e(corr)`	correlation structure
`e(scale)`	`x2`, `dev`, `phi`, or #; scale parameter
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(offset)`	linear offset variable
`e(chi2type)`	`Wald`; type of model χ^2 test
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. err.
`e(nmp)`	`nmp`, if specified
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(marginsnotok)`	predictions disallowed by `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`
Matrices	
`e(b)`	coefficient vector
`e(R)`	estimated working correlation matrix
`e(V)`	variance–covariance matrix of the estimators
`e(V_modelbased)`	model-based variance
Functions	
`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices	
`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

`xtlogit` reports the population-averaged results obtained using `xtgee, family(binomial) link(logit)` to obtain estimates. The fixed-effects results are obtained using `clogit`. See [XT] **xtgee** and [R] **clogit** for details on the methods and formulas.

If we assume a normal distribution, $N(0, \sigma_\nu^2)$, for the random effects ν_i,

$$\Pr(y_{i1}, \ldots, y_{in_i} | \mathbf{x}_{i1}, \ldots, \mathbf{x}_{in_i}) = \int_{-\infty}^{\infty} \frac{e^{-\nu_i^2/2\sigma_\nu^2}}{\sqrt{2\pi}\sigma_\nu} \left\{ \prod_{t=1}^{n_i} F(y_{it}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i) \right\} d\nu_i$$

where

$$F(y, z) = \begin{cases} \dfrac{1}{1 + \exp(-z)} & \text{if } y \neq 0 \\ \dfrac{1}{1 + \exp(z)} & \text{otherwise} \end{cases}$$

The panel-level likelihood l_i is given by

$$l_i = \int_{-\infty}^{\infty} \frac{e^{-\nu_i^2/2\sigma_\nu^2}}{\sqrt{2\pi}\sigma_\nu} \left\{ \prod_{t=1}^{n_i} F(y_{it}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i) \right\} d\nu_i$$
$$\equiv \int_{-\infty}^{\infty} g(y_{it}, x_{it}, \nu_i) d\nu_i$$

This integral can be approximated with M-point Gauss–Hermite quadrature

$$\int_{-\infty}^{\infty} e^{-x^2} h(x) dx \approx \sum_{m=1}^{M} w_m^* h(a_m^*)$$

This is equivalent to

$$\int_{-\infty}^{\infty} f(x) dx \approx \sum_{m=1}^{M} w_m^* \exp\left\{(a_m^*)^2\right\} f(a_m^*)$$

where the w_m^* denote the quadrature weights and the a_m^* denote the quadrature abscissas. The log likelihood, L, is the sum of the logs of the panel-level likelihoods l_i.

The default approximation of the log likelihood is by adaptive Gauss–Hermite quadrature, which approximates the panel-level likelihood with

$$l_i \approx \sqrt{2}\widehat{\sigma}_i \sum_{m=1}^{M} w_m^* \exp\left\{(a_m^*)^2\right\} g(y_{it}, x_{it}, \sqrt{2}\widehat{\sigma}_i a_m^* + \widehat{\mu}_i)$$

where $\widehat{\sigma}_i$ and $\widehat{\mu}_i$ are the adaptive parameters for panel i. Therefore, with the definition of $g(y_{it}, x_{it}, \nu_i)$, the total log likelihood is approximated by

$$L \approx \sum_{i=1}^{n} w_i \log\left[\sqrt{2}\widehat{\sigma}_i \sum_{m=1}^{M} w_m^* \exp\{(a_m^*)^2\} \frac{\exp\{-(\sqrt{2}\widehat{\sigma}_i a_m^* + \widehat{\mu}_i)^2/2\sigma_\nu^2\}}{\sqrt{2\pi}\sigma_\nu} \right.$$
$$\left. \prod_{t=1}^{n_i} F(y_{it}, x_{it}\boldsymbol{\beta} + \sqrt{2}\widehat{\sigma}_i a_m^* + \widehat{\mu}_i) \right]$$

where w_i is the user-specified weight for panel i; if no weights are specified, $w_i = 1$.

The default method of adaptive Gauss–Hermite quadrature is to calculate the posterior mean and variance and use those parameters for $\widehat{\mu}_i$ and $\widehat{\sigma}_i$ by following the method of Naylor and Smith (1982), further discussed in Skrondal and Rabe-Hesketh (2004). We start with $\widehat{\sigma}_{i,0} = 1$ and $\widehat{\mu}_{i,0} = 0$, and the posterior means and variances are updated in the kth iteration. That is, at the kth iteration of the optimization for l_i, we use

$$l_{i,k} \approx \sum_{m=1}^{M} \sqrt{2}\widehat{\sigma}_{i,k-1} w_m^* \exp\{a_m^*)^2\} g(y_{it}, x_{it}, \sqrt{2}\widehat{\sigma}_{i,k-1} a_m^* + \widehat{\mu}_{i,k-1})$$

Letting

$$\tau_{i,m,k-1} = \sqrt{2}\widehat{\sigma}_{i,k-1} a_m^* + \widehat{\mu}_{i,k-1}$$

$$\widehat{\mu}_{i,k} = \sum_{m=1}^{M} (\tau_{i,m,k-1}) \frac{\sqrt{2}\widehat{\sigma}_{i,k-1} w_m^* \exp\{(a_m^*)^2\} g(y_{it}, x_{it}, \tau_{i,m,k-1})}{l_{i,k}}$$

and

$$\widehat{\sigma}_{i,k} = \sum_{m=1}^{M} (\tau_{i,m,k-1})^2 \frac{\sqrt{2}\widehat{\sigma}_{i,k-1} w_m^* \exp\{(a_m^*)^2\} g(y_{it}, x_{it}, \tau_{i,m,k-1})}{l_{i,k}} - (\widehat{\mu}_{i,k})^2$$

and this is repeated until $\widehat{\mu}_{i,k}$ and $\widehat{\sigma}_{i,k}$ have converged for this iteration of the maximization algorithm. This adaptation is applied on every iteration until the log-likelihood change from the preceding iteration is less than a relative difference of 1e–6; after this, the quadrature parameters are fixed.

The log likelihood can also be calculated by nonadaptive Gauss–Hermite quadrature, the `intmethod(ghermite)` option, where $\rho = \sigma_\nu^2/(\sigma_\nu^2 + 1)$:

$$\begin{aligned} L &= \sum_{i=1}^{n} w_i \log\Big\{ \Pr(y_{i1}, \dots, y_{in_i} | \mathbf{x}_{i1}, \dots, \mathbf{x}_{in_i}) \Big\} \\ &\approx \sum_{i=1}^{n} w_i \log \left[\frac{1}{\sqrt{\pi}} \sum_{m=1}^{M} w_m^* \prod_{t=1}^{n_i} F \left\{ y_{it}, \mathbf{x}_{it}\boldsymbol{\beta} + a_m^* \left(\frac{2\rho}{1-\rho} \right)^{1/2} \right\} \right] \end{aligned}$$

Both quadrature formulas require that the integrated function be well approximated by a polynomial of degree equal to the number of quadrature points. The number of periods (panel size) can affect whether

$$\prod_{t=1}^{n_i} F(y_{it}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i)$$

is well approximated by a polynomial. As panel size and ρ increase, the quadrature approximation can become less accurate. For large ρ, the random-effects model can also become unidentified. Adaptive quadrature gives better results for correlated data and large panels than nonadaptive quadrature; however, we recommend that you use the `quadchk` command (see [XT] **quadchk**) to verify the quadrature approximation used in this command, whichever approximation you choose.

xtlogit, re and the robust VCE estimator

Specifying `vce(robust)` or `vce(cluster` *clustvar*`)` causes the Huber/White/sandwich VCE estimator to be calculated for the coefficients estimated in this regression. See [P] **_robust**, particularly *Introduction* and *Methods and formulas*. Wooldridge (2020) and Arellano (2003) discuss this application of the Huber/White/sandwich VCE estimator. As discussed by Wooldridge (2020), Stock and Watson (2008), and Arellano (2003), specifying `vce(robust)` is equivalent to specifying `vce(cluster` *panelvar*`)`, where *panelvar* is the variable that identifies the panels.

Clustering on the panel variable produces a consistent VCE estimator when the disturbances are not identically distributed over the panels or there is serial correlation in ϵ_{it}.

The cluster–robust VCE estimator requires that there are many clusters and the disturbances are uncorrelated across the clusters. The panel variable must be nested within the cluster variable because of the within-panel correlation that is generally induced by the random-effects transform when there is heteroskedasticity or within-panel serial correlation in the idiosyncratic errors.

References

Allison, P. D. 2009. *Fixed Effects Regression Models*. Newbury Park, CA: Sage.

Arellano, M. 2003. *Panel Data Econometrics*. Oxford: Oxford University Press. https://doi.org/10.1093/0199245282.001.0001.

Conway, M. R. 1990. A random effects model for binary data. *Biometrics* 46: 317–328. https://doi.org/10.2307/2531437.

Cruz-Gonzalez, M., I. Fernández-Val, and M. Weidner. 2017. Bias corrections for probit and logit models with two-way fixed effects. *Stata Journal* 17: 517–545.

Liang, K.-Y., and S. L. Zeger. 1986. Longitudinal data analysis using generalized linear models. *Biometrika* 73: 13–22. https://doi.org/10.1093/biomet/73.1.13.

Naylor, J. C., and A. F. M. Smith. 1982. Applications of a method for the efficient computation of posterior distributions. *Journal of the Royal Statistical Society*, C ser., 31: 214–225. https://doi.org/10.2307/2347995.

Neuhaus, J. M. 1992. Statistical methods for longitudinal and clustered designs with binary responses. *Statistical Methods in Medical Research* 1: 249–273. https://doi.org/10.1177/096228029200100303.

Neuhaus, J. M., J. D. Kalbfleisch, and W. W. Hauck. 1991. A comparison of cluster-specific and population-averaged approaches for analyzing correlated binary data. *International Statistical Review* 59: 25–35. https://doi.org/10.2307/1403572.

Pendergast, J. F., S. J. Gange, M. A. Newton, M. J. Lindstrom, M. Palta, and M. R. Fisher. 1996. A survey of methods for analyzing clustered binary response data. *International Statistical Review* 64: 89–118. https://doi.org/10.2307/1403425.

Skrondal, A., and S. Rabe-Hesketh. 2004. *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models*. Boca Raton, FL: Chapman and Hall/CRC.

Stock, J. H., and M. W. Watson. 2008. Heteroskedasticity-robust standard errors for fixed effects panel data regression. *Econometrica* 76: 155–174. https://doi.org/10.1111/j.0012-9682.2008.00821.x.

Twisk, J. W. R. 2013. *Applied Longitudinal Data Analysis for Epidemiology: A Practical Guide*. 2nd ed. Cambridge: Cambridge University Press.

Wooldridge, J. M. 2020. *Introductory Econometrics: A Modern Approach*. 7th ed. Boston: Cengage.

Also see

[XT] **xtlogit postestimation** — Postestimation tools for xtlogit

[XT] **quadchk** — Check sensitivity of quadrature approximation

[XT] **xtcloglog** — Random-effects and population-averaged cloglog models

[XT] **xtgee** — GEE population-averaged panel-data models

[XT] **xtmlogit** — Fixed-effects and random-effects multinomial logit models

[XT] **xtprobit** — Random-effects and population-averaged probit models

[XT] **xtset** — Declare data to be panel data

[BAYES] **bayes: xtlogit** — Bayesian random-effects logit model

[ME] **melogit** — Multilevel mixed-effects logistic regression

[MI] **Estimation** — Estimation commands for use with mi estimate

[R] **clogit** — Conditional (fixed-effects) logistic regression

[R] **logistic** — Logistic regression, reporting odds ratios

[R] **logit** — Logistic regression, reporting coefficients

[U] 20 Estimation and postestimation commands

Postestimation commands

The following postestimation commands are available after `xtlogit`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of parameters
* `estat ic`	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC, respectively)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
† `forecast`	dynamic forecasts and simulations
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
* `lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	linear predictions and their SEs, probabilities
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of parameters
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

*`estat ic` and `lrtest` are not appropriate after `xtlogit, pa`.

†`forecast` is not appropriate with `mi` estimation results or after `xtlogit, fe`.

predict

Description for predict

predict creates a new variable containing predictions such as linear predictions, probabilities, standard errors, and the equation-level score.

Menu for predict

Statistics > Postestimation

Syntax for predict

Random-effects model

predict [*type*] *newvar* [*if*] [*in*] [, *RE_statistic* nooffset]

Fixed-effects model

predict [*type*] *newvar* [*if*] [*in*] [, *FE_statistic* nooffset]

Population-averaged model

predict [*type*] *newvar* [*if*] [*in*] [, *PA_statistic* nooffset]

RE_statistic	Description
Main	
xb	linear prediction; the default
pr	marginal probability of a positive outcome
pu0	probability of a positive outcome assuming that the random effect is zero
stdp	standard error of the linear prediction

FE_statistic	Description
Main	
pc1	predicted probability of a positive outcome conditional on one positive outcome within group; the default
pu0	probability of a positive outcome assuming that the fixed effect is zero
xb	linear prediction
stdp	standard error of the linear prediction
score	first derivative of the log likelihood with respect to $\mathbf{x}_{it}\boldsymbol{\beta}$

The predicted probability for the fixed-effects model is conditional on there being only one outcome per group. See [R] **clogit** for details.

PA_statistic	Description
Main	
mu	predicted probability of *depvar*; considers the offset()
rate	predicted probability of *depvar*
xb	linear prediction
stdp	standard error of the linear prediction
score	first derivative of the log likelihood with respect to $\mathbf{x}_{it}\boldsymbol{\beta}$

These statistics are available both in and out of sample; type predict ... if e(sample) ... if wanted only for the estimation sample.

Options for predict

Main

xb calculates the linear prediction. This is the default for the random-effects model.

pc1 calculates the predicted probability of a positive outcome conditional on one positive outcome within group. This is the default for the fixed-effects model.

mu and rate both calculate the predicted probability of *depvar*. mu takes into account the offset(), and rate ignores those adjustments. mu and rate are equivalent if you did not specify offset(). mu is the default for the population-averaged model.

pr calculates the probability of a positive outcome that is marginal with respect to the random effect, which means that the probability is calculated by integrating the prediction function with respect to the random effect over its entire support.

pu0 calculates the probability of a positive outcome, assuming that the fixed or random effect for that observation's panel is zero ($\nu_i = 0$). This may not be similar to the proportion of observed outcomes in the group.

stdp calculates the standard error of the linear prediction.

nooffset is relevant only if you specified offset(*varname*) for xtlogit. This option modifies the calculations made by predict so that they ignore the offset variable; the linear prediction is treated as $\mathbf{x}_{it}\boldsymbol{\beta}$ rather than $\mathbf{x}_{it}\boldsymbol{\beta} + \text{offset}_{it}$.

score calculates the equation-level score, $u_{it} = \partial \ln L(\mathbf{x}_{it}\boldsymbol{\beta})/\partial(\mathbf{x}_{it}\boldsymbol{\beta})$.

margins

Description for margins

`margins` estimates margins of response for linear predictions and probabilities.

Menu for margins

Statistics > Postestimation

Syntax for margins

`margins` [*marginlist*] [, *options*]

`margins` [*marginlist*] , `predict(`*statistic* ...`)` [`predict(`*statistic* ...`)` ...] [*options*]

Random-effects model

statistic	Description
`pr`	marginal probability of a positive outcome; the default
`pu0`	probability of a positive outcome assuming that the random effect is zero
`xb`	linear prediction
`stdp`	not allowed with `margins`

Fixed-effects model

statistic	Description
`pu0`	probability of a positive outcome assuming that the fixed effect is zero; the default
`xb`	linear prediction
`pc1`	not allowed with `margins`
`stdp`	not allowed with `margins`
`score`	not allowed with `margins`

Population-averaged model

statistic	Description
`mu`	probability of *depvar*; considers the `offset()`
`rate`	probability of *depvar*
`xb`	linear prediction
`stdp`	not allowed with `margins`
`score`	not allowed with `margins`

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.

For the full syntax, see [R] **margins**.

Remarks and examples

▷ Example 1: Conducting hypothesis tests

In example 1 of [XT] **xtlogit**, we fit a random-effects model of union status on the person's age and level of schooling, whether she lived in an urban area, and whether she lived in the south. In fact, we included the full interaction between `south` and `year` to capture both the overall effect of residing in the south and a separate time-trend for southerners. To test whether residing in the south affects union status, we must determine whether `1.south` and `south#c.year` are jointly significant. First, we refit our model, store the estimation results for later use, and use `test` to conduct a Wald test of the joint significance of those two variables' parameters:

```
. use https://www.stata-press.com/data/r19/union
(NLS Women 14-24 in 1968)
. xtlogit union age grade not_smsa south##c.year
  (output omitted)
. estimates store fullmodel
. test 1.south 1.south#c.year
 ( 1)  [union]1.south = 0
 ( 2)  [union]1.south#c.year = 0
           chi2(  2) =  143.93
         Prob > chi2 =    0.0000
```

The test statistic is clearly significant, so we reject the null hypothesis that the coefficients are jointly zero and conclude that living in the south does significantly affect union status.

We can also test our hypothesis with a likelihood-ratio test. Here we fit the model without `south##c.year` and then call `lrtest` to compare this restricted model to the full model:

```
. xtlogit union age grade not_smsa
  (output omitted)
. lrtest fullmodel .
Likelihood-ratio test
Assumption: . nested within fullmodel
 LR chi2(3) = 146.55
Prob > chi2 = 0.0000
```

These results confirm our finding that living in the south affects union status.

◁

Also see

[XT] **xtlogit** — Fixed-effects, random-effects, and population-averaged logit models

[U] 20 Estimation and postestimation commands

xtmlogit — Fixed-effects and random-effects multinomial logit models

Description

`xtmlogit` fits random-effects and conditional fixed-effects multinomial logit models for a categorical dependent variable with unordered outcomes. The actual values taken by the dependent variable are irrelevant.

Quick start

Random-effects model of `y` as a function of `x1`, `x2`, and indicators for levels of categorical variable `a` using `xtset` data

```
xtmlogit y x1 x2 i.a
```

Same as above, but report relative-risk ratios

```
xtmlogit y x1 x2 i.a, rrr
```

Same as above, but with all variances and covariances distinctly estimated

```
xtmlogit y x1 x2 i.a, rrr covariance(unstructured)
```

Conditional fixed-effects model

```
xtmlogit y x1 x2 i.a, fe
```

Random-effects model with cluster–robust standard errors for panels nested within `cvar`

```
xtmlogit y x1 x2 i.a, vce(cluster cvar)
```

Menu

Statistics > Longitudinal/panel data > Categorical outcomes > Multinomial logistic regression (FE, RE)

Syntax

Random-effects model

xtmlogit *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, re *RE_options*]

Conditional fixed-effects model

xtmlogit *depvar* [*indepvars*] [*if*] [*in*] [*weight*] , fe [*FE_options*]

RE_options	Description
Model	
noconstant	suppress constant term
re	use random-effects estimator; the default
baseoutcome(#)	value of *depvar* that will be the base outcome
constraints(*constraints*)	apply specified linear constraints
covariance(*vartype*)	variance–covariance structure of the random effects; default is covariance(independent)
SE/Robust	
vce(*vcetype*)	*vcetype* may be oim, robust, cluster *clustvar*, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
rrr	report relative-risk ratios
lrmodel	perform the likelihood-ratio model test instead of the default Wald test
nocnsreport	do not display constraints
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Integration	
intmethod(*intmethod*)	integration method; *intmethod* may be mvaghermite (the default) or ghermite
intpoints(#)	use # quadrature points; default is intpoints(7)
Maximization	
maximize_options	control the maximization process; seldom used
startgrid(*numlist*)	improve starting values of the random-effects variance parameters by performing a grid search
collinear	keep collinear variables
coeflegend	display legend instead of statistics

vartype	Description
independent	distinct variances for each random effect and all covariances 0; the default
shared	one common random effect
identity	equal variances for random effects and all covariances 0
exchangeable	equal variances for random effects and one common pairwise covariance
unstructured	all variances and covariances to be distinctly estimated

FE_options	Description
Model	
fe	use fixed-effects estimator
baseoutcome(#)	value of *depvar* that will be the base outcome
constraints(*constraints*)	apply specified linear constraints
SE/Robust	
vce(*vcetype*)	*vcetype* may be oim, robust, cluster *clustvar*, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
rrr	report relative-risk ratios
nodots	suppress display of progress bar
nocnsreport	do not display constraints
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Permutations	
rsample(#[, rseed(#$_s$)])	draw sample of permuted outcome sequences at percentage #
favor(speed | space)	favor speed or space when generating permutations of outcome sequences; default is favor(speed)
force	force estimation to proceed even if the number of permutations exceeds 50 million
Maximization	
maximize_options	control the maximization process; seldom used
collinear	keep collinear variables
coeflegend	display legend instead of statistics

A panel variable must be specified; see [XT] **xtset**.

indepvars may contain factor variables and time-series operators; see **[U] 11.4.3 Factor variables** and **[U] 11.4.4 Time-series varlists**.

by, collect, statsby, and svy are allowed; see **[U] 11.1.10 Prefix commands**. bayes is allowed for the random-effects model. For more details, see [BAYES] **bayes: xtmlogit**.

vce() and weights are not allowed with the svy prefix; see [SVY] **svy**.

fweights, iweights, and pweights are allowed; see **[U] 11.1.6 weight**. Weights must be constant within panel.

startgrid(), collinear, and coeflegend do not appear in the dialog box.

See **[U] 20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options for RE model

Model

`noconstant`; see [R] **Estimation options**.

`re` requests the random-effects estimator. This is the default.

`baseoutcome(#)` specifies the value of *depvar* to be treated as the base outcome. The default is to choose the most frequent outcome.

`constraints(`*constraints*`)`; see [R] **Estimation options**.

`covariance(`*vartype*`)` specifies the structure of the covariance matrix for the random effects. A multinomial logit model with J outcomes can have up to $J-1$ random effects. *vartype* determines the structure that is assumed for the random effects and is one of the following: `independent`, `shared`, `identity`, `exchangeable`, or `unstructured`.

`covariance(independent)` estimates distinct variances for each of the $J-1$ random effects and all covariances are 0. This is the default.

`covariance(shared)` has one random effect that is common to all $J-1$ outcome equations. Because there is only one random effect, there is no covariance.

`covariance(identity)` estimates one common variance for all $J-1$ random effects and all covariances are 0.

`covariance(exchangeable)` estimates one common variance for all $J-1$ random effects and one common pairwise covariance.

`covariance(unstructured)` allows for all variances and covariances to be distinct. With $p = J-1$ random-effects terms, the unstructured covariance matrix will have $p(p+1)/2$ distinct parameters.

SE/Robust

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`oim`), that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster` *clustvar*), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [XT] ***vce_options***.

Specifying `vce(robust)` is equivalent to specifying `vce(cluster` *panelvar*`)`.

If `vce(bootstrap)` or `vce(jackknife)` is specified, you must also specify `baseoutcome()`.

Reporting

`level(#)`; see [R] **Estimation options**.

`rrr` reports the estimated coefficients transformed to relative-risk ratios, that is, e^b rather than b. Standard errors and confidence intervals are transformed accordingly. This option affects how results are displayed, not how they are estimated. `rrr` may be specified at estimation or when replaying previously estimated results.

`lrmodel`, `nocnsreport`; see [R] **Estimation options**.

display_options: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(`*style*`)`, `cformat(%`*fmt*`)`, `pformat(%`*fmt*`)`, `sformat(%`*fmt*`)`, and `nolstretch`; see [R] **Estimation options**.

Integration

intmethod(*intmethod*), intpoints(#); see [R] **Estimation options**.

Maximization

maximize_options: difficult, technique(*algorithm_spec*), iterate(#), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), nonrtolerance, and from(*init_specs*); see [R] **Maximize**. These options are seldom used.

The following options are available with xtmlogit but are not shown in the dialog box:

startgrid(*numlist*) performs a grid search to improve starting values of the random-effects parameters. By default, xtmlogit performs a grid search on startgrid(0.2 1).

collinear, coeflegend; see [R] **Estimation options**.

Options for FE model

Model

fe requests the fixed-effects estimator.

baseoutcome(#) specifies the value of *depvar* to be treated as the base outcome. The default is to choose the most frequent outcome.

constraints(*constraints*); see [R] **Estimation options**.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (oim), that are robust to some kinds of misspecification (robust), that allow for intragroup correlation (cluster *clustvar*), and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] ***vce_options***.

Specifying vce(robust) is equivalent to specifying vce(cluster *panelvar*).

If the rsample() option is specified, the default is vce(robust) rather than vce(oim).

If vce(bootstrap) or vce(jackknife) is specified, you must also specify baseoutcome().

Reporting

level(#); see [R] **Estimation options**.

rrr reports the estimated coefficients transformed to relative-risk ratios, that is, e^b rather than b. Standard errors and confidence intervals are transformed accordingly. This option affects how results are displayed, not how they are estimated. rrr may be specified at estimation or when replaying previously estimated results.

nodots suppresses the display of the dots that show the progress of permuting the observed outcomes.

nocnsreport; see [R] **Estimation options**.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(*%fmt*), pformat(*%fmt*), sformat(*%fmt*), and nolstretch; see [R] **Estimation options**.

Permutations

`rsample(#[ , rseed(#`$_s$`) ])` specifies that a random subset be drawn from the set of all permutations of the observed sequence of outcomes for each panel. Optionally, a random-number seed, $\#_s$, can be specified to ensure reproducibility.

The size of the random subset is given as a percentage # of K_i, where K_i is the total number of permutations of the outcome sequence in the ith panel. The resulting subset is of size $L_i = \text{ceil}\{(\#/100)K_i\}$. The observed outcome sequence is also included for a total of $L_i + 1$ sequences. If `rsample()` is not specified, `xtmlogit` uses all K_i permutations in the conditional likelihood calculation.

Specifying `rsample()` requires setting a time variable with `xtset` so that the order of the observed outcome sequence is known.

If `rsample()` is specified, the default standard error type is `vce(robust)` rather than `vce(oim)`.

`favor(speed | space)` instructs `xtmlogit` to favor either speed or space when generating the permutations of the outcome sequences. `favor(speed)` is the default. When favoring speed, the permuted sequences are generated once and stored in memory, thus increasing the speed of evaluating the likelihood. This speed increase can be seen when the number of observations per panel is relatively high. When favoring space, the permutations are generated repeatedly with each likelihood evaluation.

`force` forces estimation to proceed even if the total number of permutations ($\sum_i K_i$) exceeds 50 million. Without specification of `force`, the fixed-effects estimator issues an error message if the number of permutations exceeds 50 million. Estimation with this many permutations requires a considerable amount of memory and is computationally intensive.

Maximization

maximize_options: `difficult`, `technique(`*algorithm_spec*`)`, `iterate(#)`, `[no]log`, `trace`, `gradient`, `showstep`, `hessian`, `showtolerance`, `tolerance(#)`, `ltolerance(#)`, `nrtolerance(#)`, `nonrtolerance`, and `from(`*init_specs*`)`; see [R] **Maximize**. These options are seldom used.

The following options are available with `xtmlogit` but are not shown in the dialog box:

`collinear`, `coeflegend`; see [R] **Estimation options**.

Remarks and examples

Remarks are presented under the following headings:

Introduction

`xtmlogit` fits random-effects and conditional fixed-effects multinomial logit (MNL) models. Whenever we refer to a fixed-effects model, we mean the conditional fixed-effects model.

Both the conditional fixed-effects and the random-effects estimators produce valid estimates in the presence of unobserved heterogeneity at the panel level. The fixed-effects estimator is described in Chamberlain (1980) and Pforr (2014). For a description of the random-effects estimator, see Hartzel, Agresti, and Caffo (2001). For an application of the fixed-effects estimator, see Börsch-Supan (1990); for an application of the random-effects estimator, see Grilli and Rampichini (2007).

The MNL model is a popular method for modeling categorical outcome variables where the categories have no natural ordering. The MNL model is often used in the context of a random utility framework to analyze choices made by individuals. However, the MNL model can also be found used without an underlying utility theory, and the units of analysis do not necessarily have to be individuals or other decision-making entities. In what follows, however, we will refer to individuals for the sake of simplicity, and the set of choices each individual makes as a "panel".

Unlike in cross-sectional applications of the MNL model, in the context of panel and longitudinal data, we observe a sequence of outcomes for each individual in the dataset rather than just a single observation. Each individual sequence can be thought of as a process that depends on individual characteristics.

For example, if we were to analyze restaurant choices, vegetarians would consistently choose restaurants that offer vegetarian dishes, or health-oriented people would consistently avoid fast-food restaurants. In other words, the choices made by individuals are not independent over time because of underlying individual preferences or characteristics, which often remain unobserved in the data. The fixed- and random-effects MNL estimators discussed here offer a way to explicitly account for this unobserved heterogeneity by including an additional error term at the panel level. This panel-level error term is also known as a heterogeneity term and enters the model in addition to the error term that accounts for heterogeneity at the observation (time) level.

The unobserved-heterogeneity model for both the conditional fixed-effects as well as the random-effects estimator can be written in utility-maximization form as

$$U_{ijt} = \mathbf{x}_{it}\boldsymbol{\beta}_j + u_{ij} + \epsilon_{ijt}$$

Assuming we have a panel dataset with repeated observations from individuals, U_{ijt} is the utility of the ith individual toward outcome j at time t, with $i = 1, \dots, N$, $j = 1, \dots, J$, and $t = 1, \dots, T_i$. The observed component of utility is $\mathbf{x}_{it}\boldsymbol{\beta}_j$, where $\mathbf{x}_{it}$ is a row vector of covariates and $\boldsymbol{\beta}_j$ is a column vector of coefficients for outcome j. The unobserved part consists of error components u_{ij} and ϵ_{ijt}, where u_{ij} is the panel-level heterogeneity term and ϵ_{ijt} is an observation-level error term.

Assuming a type-1 extreme value distribution for ϵ_{ijt}, also known as a standard Gumbel distribution, gives rise to the MNL model

$$\Pr(y_{it} = m \,|\, \mathbf{x}_{it}, \boldsymbol{\beta}_j, u_{ij}) = \frac{\exp(\mathbf{x}_{it}\boldsymbol{\beta}_m + u_{im})}{\sum_{j=1}^{J} \exp(\mathbf{x}_{it}\boldsymbol{\beta}_j + u_{ij})}$$

For model identification, the above equation must be normalized with respect to a base category by setting both the elements in $\boldsymbol{\beta}_j$ as well as u_{ij} to zero for one of the categories of the outcome variable. If—without loss of generality—we let the base outcome be outcome 1, the probability that the ith individual chooses outcome m at time t is

$$\Pr(y_{it} = m \,|\, \mathbf{x}_{it}, \boldsymbol{\beta}_j, u_{ij}) = F(y_{it} = m, \mathbf{x}_{it}\boldsymbol{\beta}_j + u_{ij}) = \begin{cases} \dfrac{1}{1 + \sum_{j=2}^{J} \exp(\mathbf{x}_{it}\boldsymbol{\beta}_j + u_{ij})} & \text{if } m = 1 \\ \dfrac{\exp(\mathbf{x}_{it}\boldsymbol{\beta}_m + u_{im})}{1 + \sum_{j=2}^{J} \exp(\mathbf{x}_{it}\boldsymbol{\beta}_j + u_{ij})} & \text{if } m > 1 \end{cases}$$

Here $F(\cdot)$ is defined as the cumulative logistic distribution function.

The fixed-effects and random-effects estimators differ in their assumptions about the unobservables in $\mathbf{u}_i$ and also differ in their methods that the unobservables are accounted for with respect to estimating the coefficients in $\boldsymbol{\beta}_j$.

The random-effects estimator

The random-effects estimator requires an assumption about the distribution of u_{ij}, and the elements in $\mathbf{u}_i$ are assumed to be uncorrelated with the covariates in $\mathbf{x}_{it}$. The covariates $\mathbf{x}_{it}$ may contain constant terms as well as time-invariant predictor variables. Assuming a normal distribution for u_{ij}, the panel-level likelihood is

$$l_i = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \left\{ \prod_{t=1}^{T_i} F(y_{it} = m, \mathbf{x}_{it}\boldsymbol{\beta}_j + u_{ij}) \right\} \phi(\mathbf{u}_i, \boldsymbol{\Sigma}_u)\, d\mathbf{u}_i \tag{1}$$

where $\phi(\mathbf{u}_i, \boldsymbol{\Sigma}_u)$ is the probability density function of the normal distribution $\mathbf{u}_i \sim N(\mathbf{0}, \boldsymbol{\Sigma}_u)$. This integral of dimension $J-1$ has no closed-form solution and must be approximated numerically. By default, `xtmlogit` uses adaptive Gauss–Hermite quadrature to approximate this integral.

`xtmlogit` allows for imposing a variety of structures on $\boldsymbol{\Sigma}_u$. By default, `xtmlogit` estimates separate, independent variance components for each of the $J-1$ outcome equations. The `covariance(shared)` option estimates a single shared variance component for all $J-1$ outcome equations. The most general case is specified by the option `covariance(unstructured)`, which freely estimates all variances and covariances among the random effects instead of treating them as independent. Not imposing any structure on $\boldsymbol{\Sigma}_u$ can potentially yield more accurate results. However, this is also more computationally intensive, resulting in longer computation times.

The conditional fixed-effects estimator

The advantages of the conditional fixed-effects estimator are that elements in $\mathbf{u}_i$ can be correlated with the covariates in $\mathbf{x}_{it}$ and no distributional assumptions need to be imposed on u_{ij}. Unlike in linear fixed-effects models, the heterogeneity term u_{ij} of the logit model cannot be eliminated by taking deviations from the group mean. Moreover, it is also not feasible to account for the heterogeneity in u_{ij} by distinctly estimating an intercept for each panel because this leads to the incidental parameters problem, which renders the estimator of $\boldsymbol{\beta}_j$ inconsistent for a fixed T_i; see Andersen (1970) and Lancaster (2000). Instead, Chamberlain (1980) suggested the use of a sufficient statistic for the unobserved heterogeneity u_{ij}.

Let $\mathbf{Y}_i = (Y_{i1}, \ldots, Y_{iT_i})$ be the sequence of outcomes of the ith panel, and let $\mathbf{Y}_{it} = (Y_{i1t}, \ldots, Y_{iJt})$ be a vector with elements $Y_{ijt} = 1(i \text{ chooses } j \text{at } t)$ that indicate the chosen outcome of the ith panel at time t. The distribution of times that panel i chose each of the J alternatives over time points T_i is then the sufficient statistic $\Theta_i = \sum_{t=1}^{T_i} \mathbf{Y}_{it} = \mathbf{c}_i = (c_{i1}, \ldots, c_{iJ})$. In other words, the elements in $\mathbf{c}_i$ are sums of occurrences of each of the outcomes over time for the ith panel.

Conditioning on the sufficient statistic Θ_i, the probability of panel i having a sequence $\mathbf{Y}_i = \mathbf{s}_i$ that is consistent with $\mathbf{c}_i$ is

$$\begin{aligned}\Pr(\mathbf{Y}_i = \mathbf{s}_i \mid \Theta_i, \mathbf{u}_i, \mathbf{x}_i, \boldsymbol{\beta}) &= \Pr\{Y_{i1}, \ldots, Y_{iT_i} \mid \Psi(\mathbf{c}_i), \mathbf{u}_i, \mathbf{x}_i, \boldsymbol{\beta}\} \\ &= \frac{\exp\left(\sum\limits_{t=1}^{T_i}\sum\limits_{j=2}^{J} Y_{ijt}\mathbf{x}_{it}\boldsymbol{\beta}_j\right)}{\sum\limits_{\widetilde{Y}_{ijt}\in\Psi(\mathbf{c}_i)} \exp\left(\sum\limits_{t=1}^{T_i}\sum\limits_{j=2}^{J} \widetilde{Y}_{ijt}\mathbf{x}_{it}\boldsymbol{\beta}_j\right)}\end{aligned}$$

where $\Psi(\mathbf{c}_i)$ is the set of all permutations of individual i's observed sequence of outcomes that satisfy the condition $\sum_{t=1}^{T_i} \widetilde{\mathbf{Y}}_{it} = \mathbf{c}_i$. That is,

$$\Psi(\mathbf{c}_i) = \left\{ \widetilde{\mathbf{Y}}_i = (\widetilde{Y}_{i1}, \ldots, \widetilde{Y}_{iT_i}) \;\middle|\; \sum_{t=1}^{T_i} \widetilde{\mathbf{Y}}_{it} = \mathbf{c}_i \right\}$$

and $\widetilde{\mathbf{Y}}_{it} = (\widetilde{Y}_{i1t}, \ldots, \widetilde{Y}_{iJt})$ is a vector of indicators with respect to the permutations of the observed outcome sequence $\mathbf{Y}_i$. The log likelihood of panel i is then the natural logarithm of the above probability

$$\log l_i = \sum_{t=1}^{T_i}\sum_{j=2}^{J} Y_{ijt}\mathbf{x}_{it}\boldsymbol{\beta}_j - \log \sum_{\widetilde{Y}_{ijt}\in\Psi(\mathbf{c}_i)} \exp\left(\sum_{t=1}^{T_i}\sum_{j=2}^{J} \widetilde{Y}_{ijt}\mathbf{x}_{it}\boldsymbol{\beta}_j\right)$$

and the overall log likelihood is $\sum\limits_{i=1}^{N} \log l_i$.

To illustrate the concept of permutations in this context, let us suppose we had a panel dataset with three observations per individual and an outcome variable with four categories, $j = 1, 2, 3, 4$. Let us further assume that for some individual in the dataset we observe the sequence $\mathbf{Y}_i = (3, 2, 3)$. This sequence has a total of three permutations, so the set of all permutations (which includes the original sequence) for this individual consists of $(2, 3, 3)$, $(3, 2, 3)$, and $(3, 3, 2)$. Notice that in all three permutations, outcome 3 occurs twice, and outcome 2 occurs once, just as in the original sequence.

Curse of dimensionality

Both the random-effects and fixed-effects estimators suffer from the curse of dimensionality. For the random-effects estimator, the curse is rooted in J, the number of outcomes, because the integral in (1) is a $J-1$ dimensional integral unless one uses a common heterogeneity component for all outcomes. This means that the computation time can be high for more than just three or four outcomes. For example, if we had a dataset with six outcomes, we would have to approximate a five-dimensional integral. If we were to use the default seven quadrature integration points, which are integration points per dimension, we would end up with a total of $7^5 = 16807$ integration points, resulting in substantial computation time. If computation time becomes infeasible, one might consider using a single, shared variance component, if appropriate.

For the fixed-effects estimator, the curse of dimensionality is rooted mainly in T_i, the number of repeated observations and potentially in J. The problem is that the number of permutations in $\Psi(\mathbf{c}_i)$ grows exponentially with T_i and can become infeasibly large. The number of permutations of panel i's observed vector of outcomes is

$$K_i = \frac{T_i!}{c_{i1}! \cdots c_{ij}! \cdots c_{iJ}!}$$

For instance, suppose we observed an individual with 15 repeated observations in a dataset with 6 outcomes, $j = 1, 2, \dots, 6$, with the sequence of outcomes $\mathbf{Y}_i = (3, 3, 3, 2, 4, 1, 1, 5, 4, 6, 6, 1, 1, 2, 4)$. Here $\sum_{t=1}^{T_i} Y_{i1t} = 4$, which is to say that outcome $Y_{it} = 1$ is observed 4 times, $\sum_{t=1}^{T_i} Y_{i2t} = 2$, and so on. The size of the set of permutations of this outcome vector is

$$K_i = \frac{15!}{4!\,2!\,3!\,3!\,1!\,2!} = 378{,}378{,}000$$

Notice that this number in the hundreds of millions is the size of the permutation set of just a single panel in the dataset, and clearly this number can quickly become infeasibly large.

A potential solution that can alleviate this problem to some degree is to use a random subset of permutations (D'Haultfœuille and Iaria 2016). The `rsample()` option can be used to specify the size of the random subset as a percentage of the full set of permutations. Realistically, however, the fixed-effects estimator is really feasible only with shorter panels where the number of repeated observations does not exceed $T_i = 9$ or $T_i = 10$, depending on J, the size of the dataset, and possibly other features of the data.

Examples

▷ Example 1: MNL model with random effects

We have a (fictitious) unbalanced panel dataset of 800 women aged 18 to 40 at the time of the first interview. We wish to estimate the effect of having children under the age of 18 in the household on the women's employment status. Specifically, we wish to find out whether women become more likely not to participate in the labor force in response to having children in the household. And if so, how much more unlikely is it?

The survey was repeated every two years, and the women were asked about their main employment status during the year preceding each of the interviews. The employment status response categories were employed (full time, part time, or self-employed), unemployed (job seeking), and out of the labor force. Here is an excerpt of the dataset, showing the employment history for three individuals:

```
. use https://www.stata-press.com/data/r19/estatus
(Fictional employment status data)
. list id year estatus hhchild age in 22/41, sepby(id) noobs
```

id	year	estatus	hhchild	age
5	2002	Employed	Yes	38
5	2004	Employed	No	40
5	2006	Employed	No	42
5	2008	Employed	No	44
5	2010	Out of labor force	No	46
5	2012	Out of labor force	No	48
5	2014	Unemployed	No	50
6	2002	Unemployed	Yes	31
6	2004	Employed	Yes	33
6	2006	Out of labor force	Yes	35
6	2008	Unemployed	Yes	37
6	2010	Out of labor force	Yes	39
6	2012	Unemployed	No	41
7	2002	Out of labor force	Yes	33
7	2004	Employed	Yes	35
7	2006	Employed	Yes	37
7	2008	Out of labor force	Yes	39
7	2010	Employed	No	41
7	2012	Employed	No	43
7	2014	Employed	No	45

The first person shown in the above excerpt (`id==5`) was observed between years 2002 and 2014. The variable `estatus` records the employment history over these years. In this case, the person has been employed between 2002 and 2008, was out of the labor force between 2010 and 2012, and was unemployed prior to the interview in 2014.

The variable `hhchild` records whether at least one child under the age of 18 was living with the surveyee in the same household at the time of the interview. Looking at the data of the first person in the above excerpt, we see that there was one or more children in the household in 2002, but no children in the household between 2004 and 2014. The variable `age` records the age of the women at each interview. In this case, the woman was observed between 38 and 50 years of age.

To inspect the distribution of employment status over the entire sample, we can use the `tabulate` command:

```
. tabulate estatus
```

Employment status	Freq.	Percent	Cum.
Out of labor force	1,682	35.33	35.33
Unemployed	703	14.77	50.09
Employed	2,376	49.91	100.00
Total	4,761	100.00	

We can see that in 35% of all observations, the interviewed women reported to be out of the labor force, 15% of the time the women were unemployed, and 50% of the time the women were employed.

As with other panel-data estimators, we first need to declare our dataset to be panel data by using the `xtset` command. Here we do not plan to use any lagged covariates, so it is sufficient to `xtset` our dataset with just the person identifier `id` and without a variable for time:

```
. xtset id
Panel variable: id (unbalanced)
```

We can now go ahead and fit our model using `xtmlogit`. We will also include a number of control variables: `age`, a person's annual household income at the time of interview (`hhincome`), whether a significant other was also living in the household at the time of interview (`hhsigno`), and whether the surveyee was the sole or primary breadwinner in her household at the time of interview (`bwinner`).

We use the variable `estatus` as our dependent variable, and `hhchild` is our independent variable of interest. Because `hhchild`, `hhsigno`, and `bwinner` are binary variables, we specify them as factor variables.

```
. xtmlogit estatus i.hhchild age hhincome i.hhsigno i.bwinner
Fitting comparison model ...
Refining starting values:
Grid node 0:   Log likelihood = -4483.1721
Grid node 1:   Log likelihood = -4516.6753
Fitting full model:
Iteration 0:   Log likelihood = -4483.1721
Iteration 1:   Log likelihood = -4474.3849
Iteration 2:   Log likelihood = -4468.9353
Iteration 3:   Log likelihood = -4468.8415
Iteration 4:   Log likelihood = -4468.8413
Random-effects multinomial logistic regression       Number of obs    =  4,761
Group variable: id                                   Number of groups =    800
Random effects u_i ~ Gaussian                        Obs per group:
                                                                  min =      5
                                                                  avg =    6.0
                                                                  max =      7
Integration method: mvaghermite                      Integration pts. =      7
                                                     Wald chi2(10)    = 239.26
Log likelihood = -4468.8413                          Prob > chi2      = 0.0000

------------------------------------------------------------------------------
     estatus | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
Out_of_lab~e |
     hhchild |
         Yes |   .4628125   .0962758     4.81   0.000     .2741154    .6515096
         age |  -.004825    .0066428    -0.73   0.468    -.0178446    .0081946
    hhincome |  -.0046922    .001839    -2.55   0.011    -.0082965   -.0010879
             |
     hhsigno |
         Yes |   .4967056   .0946442     5.25   0.000     .3112063    .6822049
             |
     bwinner |
         Yes |  -.4740919   .0727992    -6.51   0.000    -.6167756   -.3314082
       _cons |  -.4787579   .2845139    -1.68   0.092    -1.036395    .0788792
-------------+----------------------------------------------------------------
Unemployed   |
     hhchild |
         Yes |  -.0401989    .119596    -0.34   0.737    -.2746027    .1942049
         age |   .0042644   .0081818     0.52   0.602    -.0117716    .0203004
    hhincome |  -.0308468   .0026529   -11.63   0.000    -.0360463   -.0256473
             |
     hhsigno |
         Yes |      .0968   .1192659     0.81   0.417    -.1369568    .3305568
             |
     bwinner |
         Yes |  -.2252587   .0951984    -2.37   0.018    -.4118441   -.0386733
       _cons |  -.0953821   .3508736    -0.27   0.786    -.7830817    .5923175
-------------+----------------------------------------------------------------
Employed     |  (base outcome)
-------------+----------------------------------------------------------------
     var(u1)|   .8587807   .1090216                      .6696113    1.101392
     var(u2)|   .7370366   .1388917                      .5094287    1.066338
------------------------------------------------------------------------------
LR test vs. multinomial logit: chi2(2) = 225.31           Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

Looking at the table header, we can find some useful information about the model we just fit. For example, we can see that the estimation sample consists of 4,761 observations from 800 groups (800 individuals in this case), with between 5 and 7 observations per group. The model test right above the table on the right is a joint test of all model coefficients except the constants.

Looking at the output table itself, we see the results for all $J-1$ equations. Because we have three outcome categories, we see the coefficient estimates for two of the outcomes, while employment is our base outcome. Here using employment as the base makes sense given our research question, and we would have chosen this as a base if we had to specify it explicitly. In this case, however, employment was chosen automatically because it is the most frequent category in our dataset, which is what `xtmlogit` defaults to. If we had wanted to specify a different category as the base, we would have used the `baseoutcome()` option.

Below the model coefficient estimates, we find the estimated variances of the random effects. In this case, we have two estimates that correspond to the nonbase equations. By default, `xtmlogit` assumes that the random effects are uncorrelated across the equations. We will see in the next example how to use the `covariance()` option to specify a different covariance structure. Here we can see that there is some considerable variance of the panel-level unobservables. The lower bound of the 95% confidence interval is not close to zero relative to their estimated standard errors. This observation is confirmed by the likelihood-ratio test shown beneath the table, which is a test of our model against the MNL model without random effects.

Let us get back to our initial research question: what is the effect of having children under the age of 18 in the household on employment status? The interpretation of the coefficients is the same as in a conventional cross-sectional MNL model, except that, in the random-effects case, they are to be interpreted as conditional on the random effects, while they naturally have a population-average interpretation in the cross-sectional case. Either way, the coefficients are difficult to interpret. They can be thought of as the natural logarithm of a double ratio: the logarithm of the relative risk, relative to the base category. Realistically, only the sign of these coefficients can be interpreted usefully. Looking at the results, we can see that the coefficient of `hhchild` in the first equation (out of labor force) is around 0.46. Thus, we can say that women with children under 18 in the household are more likely not to participate in the labor force than women with no young children in the household, relative to being employed full time.

A more informative way to interpret the results would be to use relative-risk ratios (RRRs) instead of log relative-risk ratios by exponentiating the coefficients. That is, instead of β_j, we use $\exp(\beta_j)$ to interpret the results. With `xtmlogit`, we can use the `rrr` option for that purpose. This option can be used at the time of estimation or when replaying results. Here we use it as a replay option:

```
. xtmlogit, rrr

Random-effects multinomial logistic regression       Number of obs    =  4,761
Group variable: id                                   Number of groups =    800

Random effects u_i ~ Gaussian                        Obs per group:
                                                                  min =      5
                                                                  avg =    6.0
                                                                  max =      7

Integration method: mvaghermite                      Integration pts. =      7

                                                     Wald chi2(10)    = 239.26
Log likelihood = -4468.8413                          Prob > chi2      = 0.0000

------------------------------------------------------------------------------
     estatus |        RRR   Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
Out_of_lab~e |
     hhchild |
        Yes  |   1.588535   .1529375     4.81   0.000     1.315367    1.918435
         age |   .9951866   .0066108    -0.73   0.468     .9823137    1.008228
    hhincome |   .9953188   .0018303    -2.55   0.011     .9917379    .9989127
             |
     hhsigno |
        Yes  |   1.643299   .1555288     5.25   0.000     1.365071    1.978235
             |
     bwinner |
        Yes  |   .6224501   .0453138    -6.51   0.000     .5396818    .7179121
       _cons |   .6195525   .1762713    -1.68   0.092     .3547312    1.082074
-------------+----------------------------------------------------------------
Unemployed   |
     hhchild |
        Yes  |   .9605983   .1148837    -0.34   0.737     .7598739    1.214345
         age |   1.004274   .0082168     0.52   0.602     .9882974    1.020508
    hhincome |   .9696241   .0025723   -11.63   0.000     .9645956    .9746788
             |
     hhsigno |
        Yes  |    1.10164   .1313881     0.81   0.417     .8720079    1.391743
             |
     bwinner |
        Yes  |   .7983097   .0759978    -2.37   0.018     .6624275    .9620649
       _cons |   .9090255   .3189531    -0.27   0.786     .4569955    1.808174
-------------+----------------------------------------------------------------
Employed     |  (base outcome)
-------------+----------------------------------------------------------------
     var(u1) |   .8587807   .1090216                      .6696113    1.101392
     var(u2) |   .7370366   .1388917                      .5094287    1.066338
------------------------------------------------------------------------------
Note: Estimates are transformed only in the first 3 equations to
      relative-risk ratios.
Note: _cons estimates baseline relative risk (conditional on zero random
      effects).
LR test vs. multinomial logit: chi2(2) = 225.31           Prob > chi2 = 0.0000

Note: LR test is conservative and provided only for reference.
```

Looking at the RRRs of `hhchild` in the out-of-labor-force equation, which is around 1.6, we can say that the relative risk of being out of the labor force for women having at least one child under the age of 18 in the household versus having no children under 18 in the household is 1.6 times as large as the relative risk in the case of employment. While this provides a little bit more information, it still does not provide a very intuitive way to interpret our results. It would be easier if we could just see the actual risks for each of the outcomes with respect to the `hhchild` variable and then also the risk differences rather than risk ratios. To that end, we can use `margins`:

```
. margins hhchild

Predictive margins                                     Number of obs = 4,761
Model VCE: OIM

1._predict: Pr(estatus==Out_of_labor_force), predict(pr outcome(1))
2._predict: Pr(estatus==Unemployed), predict(pr outcome(2))
3._predict: Pr(estatus==Employed), predict(pr outcome(3))

------------------------------------------------------------------------------
             |            Delta-method
             |     Margin   std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
   _predict# |
     hhchild |
       1#No  |   .3021986   .0131047    23.06   0.000     .2765138    .3278834
      1#Yes  |   .3912783   .0119865    32.64   0.000     .3677852    .4147714
       2#No  |   .1630791   .0101239    16.11   0.000     .1432367    .1829216
      2#Yes  |    .139782   .0079417    17.60   0.000     .1242167    .1553474
       3#No  |   .5347223   .0136504    39.17   0.000      .507968    .5614766
      3#Yes  |   .4689397   .0116018    40.42   0.000     .4462006    .4916787
------------------------------------------------------------------------------
```

By default, `margins` uses predicted probabilities that account for the random effects. The probabilities are obtained by integrating out the random effects such that their averages can be used to make population-average inferences. Starting with our third outcome, employment, we can see that the averaged probability of being employed full time is around 0.47 in the presence of children under the age of 18 in the household, whereas this probability is around 0.53 in the absence of young children. Thus, women have a higher chance of being employed full time if they have no young children living with them in the same household.

We can further quantify the difference in chance by calculating the risk difference, which here is around 0.07. Using a percentage scale rather than probability scale, we can say that the chance of being employed is higher by about 7 percentage points if no young child is in the household. Looking at the other outcome of interest, we can see that the chance of being out of the labor force is about 39% in the presence of young children in the household and around 30% otherwise, resulting in a risk difference of around 9 percentage points.

We could also compute these risk differences directly by using the contrast operator `r.`:

```
. margins r.hhchild

Contrasts of predictive margins                          Number of obs = 4,761
Model VCE: OIM

1._predict: Pr(estatus==Out_of_labor_force), predict(pr outcome(1))
2._predict: Pr(estatus==Unemployed), predict(pr outcome(2))
3._predict: Pr(estatus==Employed), predict(pr outcome(3))

──────────────────────────────────────────────────
                 │         df        chi2     P>chi2
─────────────────┼────────────────────────────────
hhchild@_predict │
  (Yes vs No) 1  │          1       26.36     0.0000
  (Yes vs No) 2  │          1        3.28     0.0700
  (Yes vs No) 3  │          1       13.33     0.0003
          Joint  │          2       26.40     0.0000
──────────────────────────────────────────────────

─────────────────────────────────────────────────────────────────
                 │            Delta-method
                 │   Contrast   std. err.     [95% conf. interval]
─────────────────┼───────────────────────────────────────────────
hhchild@_predict │
  (Yes vs No) 1  │   .0890797   .0173496      .0550752    .1230843
  (Yes vs No) 2  │  -.0232971   .0128562     -.0484948    .0019005
  (Yes vs No) 3  │  -.0657826   .0180195     -.1011001   -.0304651
─────────────────────────────────────────────────────────────────
```

We can see that the results match the differences from the previous `margins` call. The predicted probabilities underlying the `margins` analysis are also the default predictions of `predict` after `xtmlogit, re`.

◁

▷ Example 2: Covariance structure of the random effects

As mentioned in the previous example, `xtmlogit` by default uses an independent covariance structure for the random effects, which is to say that the random effects for each of the $J - 1$ equations are assumed to be uncorrelated. A more general case here would be to not impose any structure on the random effects and freely estimate the covariances among the random effects rather than assuming that the covariances are zero. To fit our model with an unstructured covariance matrix, we use the `covariance(unstructured)` option:

```
. xtmlogit estatus i.hhchild age hhincome i.hhsigno i.bwinner,
> covariance(unstructured)
  (output omitted)
Random-effects multinomial logistic regression        Number of obs    =  4,761
Group variable: id                                    Number of groups =    800

Random effects u_i ~ Gaussian                         Obs per group:
                                                                   min =      5
                                                                   avg =    6.0
                                                                   max =      7

Integration method: mvaghermite                       Integration pts. =      7

                                                      Wald chi2(10)    = 242.93
Log likelihood = -4438.2887                           Prob > chi2      = 0.0000

------------------------------------------------------------------------------
     estatus | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
Out_of_lab~e |
     hhchild |
        Yes  |   .4924799   .1002988     4.91   0.000      .295898    .6890619
         age |   -.004219   .0070064    -0.60   0.547    -.0179513    .0095133
    hhincome |   -.006046    .001992    -3.04   0.002    -.0099503   -.0021417
             |
     hhsigno |
        Yes  |   .5036976   .0966982     5.21   0.000     .3141726    .6932225
             |
     bwinner |
        Yes  |   -.489057   .0745454    -6.56   0.000    -.6351632   -.3429507
       _cons |  -.3930378    .298386    -1.32   0.188    -.9778636     .191788
-------------+----------------------------------------------------------------
Unemployed   |
     hhchild |
        Yes  |   .0399687   .1238417     0.32   0.747    -.2027565    .2826939
         age |   .0045538   .0085081     0.54   0.592    -.0121219    .0212294
    hhincome |  -.0315377   .0027426   -11.50   0.000    -.0369131   -.0261624
             |
     hhsigno |
        Yes  |   .1495817   .1214242     1.23   0.218    -.0884053    .3875687
             |
     bwinner |
        Yes  |  -.2552257   .0968165    -2.64   0.008    -.4449826   -.0654689
       _cons |  -.0417024   .3633406    -0.11   0.909    -.7538368     .670432
-------------+----------------------------------------------------------------
Employed     |  (base outcome)
-------------+----------------------------------------------------------------
     var(u1) |   1.132081   .1331468                       .899012    1.425572
     var(u2) |   1.102612   .1698422                      .8152803     1.49121
-------------+----------------------------------------------------------------
   cov(u1,u2)|   .7871916   .1222148     6.44   0.000      .547655    1.026728
------------------------------------------------------------------------------
LR test vs. multinomial logit: chi2(3) = 286.41          Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

At the bottom of the table, we can see the additional estimate for the covariance among the random effects. When we look at the estimate relative to its standard error, or at the corresponding test result, it looks as though the random effects are correlated considerably. To get a better idea of how strongly the random effects are correlated, we might want to look at standard deviations and correlations, rather than variances and covariances. We can do that by using the `estat sd` postestimation command:

```
. estat sd
```

estatus	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
sd(u1)	1.063993	.0625694			.9481624	1.193973
sd(u2)	1.050053	.0808731			.9029287	1.221151
corr(u1,u2)	.7045801	.0632646	11.14	0.000	.5581225	.8084624

The results of `estat sd` show that the correlation between the random effects, `u1` and `u2`, is around 0.7, which appears rather substantial. If we had more than one estimated covariance and wanted to test the inclusion of covariance estimates as a whole, we could perform a joint test on the covariances against zero using the `test` command. Testing covariances against zero is straightforward because they are not bounded, unlike the variances. Because here we have only a single covariance estimate, we can simply take the test result reported by `xtmlogit`. The results show that we can reject the hypothesis of the covariance being zero.

Alternatively, we could perform a likelihood-ratio test here because the model with independent covariance structure is a special case of the model with no structure imposed. We will fit the model with uncorrelated random effects again, store the results, and use the `lrtest` command to perform the likelihood-ratio test:

```
. estimates store unstr
. xtmlogit estatus i.hhchild age hhincome i.hhsigno i.bwinner, baseoutcome(3)
  (output omitted)
. estimates store indep
. lrtest unstr indep
Likelihood-ratio test
Assumption: indep nested within unstr
 LR chi2(1) =  61.11
Prob > chi2 = 0.0000
```

The conclusion here is the same as before: the model with no structure imposed on the random effects covariance matrix appears to be preferable. However, if we compare the results with those from our previous model, we can see that the model with unstructured covariance matrix would not necessarily lead to substantially different conclusions, judging by the differences in relative-risk ratios between the two models. This becomes even more apparent if we were to look at the differences in the averaged marginal probabilities. For example, the difference between having and not having a child in the household with respect to not participating in the labor force was 0.089 on the probability scale in the previous example with independent covariance structure. If we were to compute this risk difference again for the unstructured model, we would find a difference of 0.092 with a similar standard error.

As an aside, notice that when we refit the model with uncorrelated random effects, we specified the `baseoutcome(3)` option. We would not have to do this because we already knew that `xtmlogit` would choose the third outcome as base, but we did so anyway to point out that it is good practice to be explicit about this in this context. It is important that the models that are compared with a likelihood-ratio test use the same base outcome. This is because, unlike in a conventional cross-sectional MNL model, the likelihood solution differs with different base outcomes because the modeling of random effects depends on what category is selected as the reference category.

◁

▷ Example 3: MNL model with conditional fixed effects

We will now use the conditional fixed-effects estimator instead of the random-effects estimator to fit our model. To do so, all we need to do is to specify the `fe` option of `xtmlogit`. However, because we have seen that the results are easier to interpret with relative-risk ratios, we will specify the `rrr` option right away:

```
. xtmlogit estatus i.hhchild age hhincome i.hhsigno i.bwinner, fe rrr
note: 80 groups (451 obs) omitted because of no variation in the outcome
      variable over time.
Computing initial values ...
Setting up 26,168 permutations:
....10%....20%....30%....40%....50%....60%....70%....80%....90%....100%
Fitting full model:
Iteration 0:  Log likelihood = -2136.5919
Iteration 1:  Log likelihood = -2136.2728
Iteration 2:  Log likelihood = -2136.2728
Fixed-effects multinomial logistic regression        Number of obs    =  4,310
Group variable: id                                   Number of groups =    720
                                                     Obs per group:
                                                                  min =      5
                                                                  avg =    6.0
                                                                  max =      7
                                                     LR chi2(10)      = 103.29
Log likelihood = -2136.2728                          Prob > chi2      = 0.0000

------------------------------------------------------------------------------
     estatus |        RRR   Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
Out_of_lab~e |
     hhchild |
        Yes  |   1.800717   .2266555     4.67   0.000     1.407036    2.304549
         age |   .9996159   .0147684    -0.03   0.979     .9710854    1.028985
    hhincome |   .9878698   .0087391    -1.38   0.168     .9708891    1.005148
             |
     hhsigno |
        Yes  |   1.663632    .166548     5.08   0.000     1.367233    2.024287
             |
     bwinner |
        Yes  |   .6277743   .0491447    -5.95   0.000     .5384781    .7318786
-------------+----------------------------------------------------------------
Unemployed   |
     hhchild |
        Yes  |   1.177757   .1930267     1.00   0.318     .8541801    1.623911
         age |   1.006356   .0195273     0.33   0.744     .9688014    1.045366
    hhincome |   .9706959   .0116513    -2.48   0.013     .9481262    .9938029
             |
     hhsigno |
        Yes  |   1.124478   .1463356     0.90   0.367     .8713222    1.451187
             |
     bwinner |
        Yes  |   .7795833   .0802992    -2.42   0.016      .637069    .9539784
-------------+----------------------------------------------------------------
Employed     |  (base outcome)
------------------------------------------------------------------------------
```

Starting with the table header, we can see that our estimation sample consists of 4,310 observations from 720 women. We saw earlier that we had 800 women in our dataset, so why do we now have only 720? The answer to this question is given in the note near the top of the output, which lets us know that 451 observations from 80 groups were dropped for the analysis because in these cases, there is no variation in the outcome variable over time. Technically, these observations could have been kept in the estimation sample, but with no variation in the outcome, these observations would not contribute anything to the likelihood, so they can as well be excluded. Looking at the relative-risk ratios, we see the results are fairly similar to our random-effects estimates. We observe an RRR for our variable of interest `hhchild` of around 1.8 for the out-of-labor-force category. The interpretation of the RRRs here is the same as with the random-effects model from the earlier examples.

An unfortunate side effect of the fixed-effects estimator is that we cannot make predictions that account for the panel-level unobservables. That is because we do not estimate the unobservables explicitly. Therefore, unfortunately, we also cannot perform useful marginal analyses using the `margins` command.

◁

▷ Example 4: Fixed effects estimation with random permutation sampling

As noted earlier, if the number of repeated observations, T_i, becomes larger than, say, $T_i = 10$, the set of permutations can become very large, resulting in computations that may become infeasibly intensive. In that case, a potential solution could be to take a random sample of the set of permutations. This can be done by using the `rsample()` option, which allows one to specify the size of the sample as a percentage of the full set of permutations. Here we will fit the same model as in the previous example, except that we take a 10% random sample of the permutations.

Before we do so, however, note that in this case we have to `xtset` the data with a time variable. The reason for this is that we have to determine the observed sequence of outcomes that has to be included in the set of permutations that we use in the denominator of the formula. This is not necessary without sampling, because the full set of permutations always includes the observed sequence. Without determining the observed sequence, estimation results would randomly depend on the sort order of the data. To specify drawing a 10% random sample and to also set a random-number seed for reproducibility, we just add the `rsample(10, rseed(123))` option.

```
. xtset id year

Panel variable: id (unbalanced)
 Time variable: year, 2002 to 2014, but with gaps
         Delta: 1 unit

. xtmlogit estatus i.hhchild age hhincome i.hhsigno i.bwinner, fe rrr
> rsample(10, rseed(123))
note: option vce() set to vce(robust) because of permutation sampling.
note: 80 groups (451 obs) omitted because of no variation in the outcome
      variable over time.

Computing initial values ...

Setting up 3,495 permutations:
....10%....20%....30%....40%....50%....60%....70%....80%....90%....100%

Fitting full model:

Iteration 0:  Log pseudolikelihood = -908.26163
Iteration 1:  Log pseudolikelihood =  -906.4585
Iteration 2:  Log pseudolikelihood = -906.45801
Iteration 3:  Log pseudolikelihood = -906.45801

Fixed-effects multinomial logistic regression       Number of obs    =  4,310
Group variable: id                                  Number of groups =    720

                                                    Obs per group:
                                                                 min =      5
                                                                 avg =    6.0
                                                                 max =      7

                                                    Wald chi2(10)    =  72.91
Log pseudolikelihood = -906.45801                   Prob > chi2      = 0.0000

                                     (Std. err. adjusted for 720 clusters in id)
--------------------------------------------------------------------------------
               |               Robust
       estatus |        RRR   std. err.      z    P>|z|     [95% conf. interval]
---------------+----------------------------------------------------------------
Out_of_lab~e   |
       hhchild |
          Yes  |   1.790876   .2663706     3.92   0.000     1.338011    2.397017
           age |    .994506   .0167663    -0.33   0.744     .9621816    1.027916
      hhincome |   .9858517   .0099036    -1.42   0.156     .9666309    1.005455
               |
       hhsigno |
          Yes  |   1.559166   .1891864     3.66   0.000     1.229162    1.977769
               |
       bwinner |
          Yes  |   .6304536   .0616622    -4.72   0.000     .5204757    .7636702
---------------+----------------------------------------------------------------
Unemployed     |
       hhchild |
          Yes  |   1.186982   .2173595     0.94   0.349     .8290349    1.699479
           age |   .9953453   .0215995    -0.21   0.830     .9538986    1.038593
      hhincome |   .9661192   .0127244    -2.62   0.009     .9414989    .9913833
               |
       hhsigno |
          Yes  |   .9267669   .1294269    -0.54   0.586     .7048498    1.218553
               |
       bwinner |
          Yes  |   .7490293    .088281    -2.45   0.014     .5945326    .9436738
---------------+----------------------------------------------------------------
Employed       |  (base outcome)
--------------------------------------------------------------------------------
```

Looking at the output, we can see that the results are very close to the results from the previous example, with standard errors being slightly larger, as we would expect. Notice also that, by default, `xtmlogit` computes cluster–robust standard errors in this case because the likelihood function is not the true likelihood because a term that is the sum over all permutations is replaced by a sum over a sample of the permutations.

◁

▷ Example 5: Choosing between fixed- and random-effects models

It can be challenging to decide a priori whether to use the fixed- or the random-effects estimator. If the assumptions of the random-effects estimator hold, then both the random and fixed-effects estimators are consistent. However, in that case, the random-effects estimator is more efficient and thus preferable. On the other hand, if the assumptions of the random-effects estimator do not hold, it becomes inconsistent, and we should use the fixed-effects estimator. In that sense, one could get the idea that we should err on the side of caution and always use the fixed-effects estimator. However, the random-effects estimator has a couple of practical advantages beyond efficiency.

For example, if we were interested in estimating the effect of a variable that is constant over time (for all observations in the dataset), then we could include that variable in the random-effects model, but not in the fixed-effects model. With the fixed-effects model, variables that are constant over time are absorbed into the fixed effects. Also, with the random-effects estimator, we can predict probabilities that account for panel-level unobservables and that lend themselves to a population-average interpretation when we use `margins`—something we cannot do with the fixed-effects estimator because the unobservables are not estimated.

A possible solution to this dilemma is to use a Hausman test. In our context here, our null hypothesis (H_0) is that the panel-level unobservables are uncorrelated with the covariates in the model, while the alternative hypothesis (H_a) is that the unobservables are correlated with the covariates. The fixed-effects estimator is consistent under both H_0 and H_a, while the random-effects estimator is inconsistent under H_a but efficient under H_0. To apply the Hausman test, we first fit both the fixed- and random-effects models, store their results, and then use the `hausman` command:

```
. xtmlogit estatus i.hhchild age hhincome i.hhsigno i.bwinner, fe
  (output omitted)
. estimates store FE
. xtmlogit estatus i.hhchild age hhincome i.hhsigno i.bwinner
  (output omitted)
. estimates store RE
. hausman FE RE, alleqs
```

	—— Coefficients ——			
	(b) FE	(B) RE	(b-B) Difference	sqrt(diag(V_b-V_B)) Std. err.
Out_of_lab~e				
1.hhchild	.5881852	.4628125	.1253727	.0810809
age	-.0003842	-.004825	.0044408	.0131965
hhincome	-.0122043	-.0046922	-.0075122	.0086532
1.hhsigno	.5090034	.4967056	.0122977	.0326296
1.bwinner	-.4655745	-.4740919	.0085173	.0287868
Unemployed				
1.hhchild	.163612	-.0401989	.203811	.1120618
age	.0063355	.0042644	.0020711	.0175947
hhincome	-.029742	-.0308468	.0011048	.0117062
1.hhsigno	.1173192	.0968	.0205192	.0520686
1.bwinner	-.2489958	-.2252587	-.0237371	.0393297

```
                         b = Consistent under H0 and Ha; obtained from xtmlogit.
          B = Inconsistent under Ha, efficient under H0; obtained from xtmlogit.

Test of H0: Difference in coefficients not systematic

    chi2(10) = (b-B)'[(V_b-V_B)^(-1)](b-B)
             =   8.05
Prob > chi2 = 0.6238
```

Notice that we specified hausman such that we gave it the results of the estimator that is consistent under both H_0 and H_a first (the fixed-effects estimator) and the results of the estimator that is efficient under H_0 second (the random-effects estimator). We also specified the alleqs option to apply the test to all equations present in both models. The result of the test, $\chi^2 = 8.05$ with $\text{df} = 10$ yielding $p = 0.62$, suggests that we do not reject H_0. In other words, here we may proceed with the random-effects estimator.

◁

❑ Technical note

The random-effects model is calculated using quadrature, which is an approximation whose accuracy depends partially on the number of integration points used. We can use the quadchk command to see if changing the number of integration points affects the results. If the results change, the quadrature approximation is not accurate given the number of integration points. Try increasing the number of integration points using the intpoints() option, and run quadchk again. Do not attempt to interpret the results of estimates when the coefficients reported by quadchk differ substantially. See [XT] **quadchk** for details and [XT] **xtprobit** for an example.

Because the xtmlogit likelihood function is calculated by Gauss–Hermite quadrature, on large problems computations can be slow. Computation time is roughly proportional to the number of points used for the quadrature.

❑

Stored results

xtmlogit, re stores the following in e():

Scalars

e(N)	number of observations
e(N_g)	number of groups
e(k)	number of parameters
e(k_out)	number of outcomes
e(k_eq)	number of equations in e(b)
e(k_eq_model)	number of equations in overall model test
e(k_eq_base)	equation number of the base outcome
e(baseout)	the value of *depvar* to be treated as the base outcome
e(ibaseout)	index of the base outcome
e(k_dv)	number of dependent variables
e(df_m)	model degrees of freedom
e(ll)	log likelihood
e(ll_0)	log likelihood, constant-only model
e(ll_c)	log likelihood, comparison model
e(chi2)	χ^2
e(chi2_c)	χ^2 for comparison test
e(p)	p-value for model test
e(p_c)	p-value for comparison test
e(df_c)	comparison test degrees of freedom
e(N_clust)	number of clusters
e(n_quad)	number of quadrature points
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(rank)	rank of e(V)
e(rank0)	rank of e(V) for constant-only model
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros

e(cmd)	gsem
e(cmd2)	xtmlogit
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(wtype)	weight type
e(wexp)	weight expression
e(covariance)	random-effects covariance structure
e(ivar)	variable denoting groups
e(model)	re
e(title)	title in estimation output
e(distrib)	Gaussian; the distribution of the random effect
e(clustvar)	name of cluster variable
e(eqnames)	names of equations
e(baselab)	value label corresponding to base outcome
e(chi2type)	Wald or LR; type of model χ^2 test
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(intmethod)	integration method
e(opt)	type of optimization
e(which)	max or min; whether optimizer is to perform maximization or minimization
e(ml_method)	type of ml method
e(user)	name of likelihood-evaluator program
e(technique)	maximization technique
e(properties)	b V

e(predict)	program used to implement predict
e(estat_cmd)	program used to implement estat
e(marginsok)	predictions allowed by margins
e(marginsnotok)	predictions disallowed by margins
e(marginsdefault)	default predict() specification for margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved
Matrices	
e(b)	coefficient vector
e(Cns)	constraints matrix
e(out)	outcome values
e(ilog)	iteration log
e(gradient)	gradient vector
e(V)	variance–covariance matrix of the estimators
e(V_modelbased)	model-based variance
Functions	
e(sample)	marks estimation sample

In addition to the above, the following is stored in r():

Matrices	
r(table)	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

xtmlogit, fe stores the following in e():

Scalars	
e(N)	number of observations
e(N_g)	number of groups
e(N_drop)	number of observations dropped because of no variation in outcome
e(N_group_drop)	number of groups dropped because of no variation in outcome
e(k)	number of parameters
e(k_out)	number of outcomes
e(k_eq)	number of equations in e(b)
e(k_eq_model)	number of equations in overall model test
e(k_eq_base)	equation number of the base outcome
e(baseout)	the value of *depvar* to be treated as the base outcome
e(ibaseout)	index of the base outcome
e(nperm)	number of permutations
e(nperm_sampled)	number of sampled permutations
e(rsample)	1 if permutation sampling, 0 otherwise
e(rssize)	percentage of sampled permutations
e(k_dv)	number of dependent variables
e(df_m)	model degrees of freedom
e(ll)	log likelihood
e(ll_0)	log likelihood, baseline model
e(chi2)	χ^2
e(p)	p-value for model test
e(N_clust)	number of clusters
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(rank)	rank of e(V)
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros

`e(cmd)`	`xtmlogit`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(ivar)`	variable denoting groups
`e(model)`	`fe`
`e(title)`	title in estimation output
`e(clustvar)`	name of cluster variable
`e(eqnames)`	names of equations
`e(baselab)`	value label corresponding to base outcome
`e(chi2type)`	`Wald` or `LR`; type of model χ^2 test
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. err.
`e(rs_rngstate)`	random-number state used
`e(opt)`	type of optimization
`e(which)`	`max` or `min`; whether optimizer is to perform maximization or minimization
`e(ml_method)`	type of `ml` method
`e(user)`	name of likelihood-evaluator program
`e(technique)`	maximization technique
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(marginsok)`	predictions allowed by `margins`
`e(marginsnotok)`	predictions disallowed by `margins`
`e(marginsdefault)`	default `predict()` specification for `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`

Matrices

`e(b)`	coefficient vector
`e(Cns)`	constraints matrix
`e(out)`	outcome values
`e(ilog)`	iteration log
`e(gradient)`	gradient vector
`e(V)`	variance–covariance matrix of the estimators
`e(V_modelbased)`	model-based variance

Functions

`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

For panels $i = 1, \dots, N$ with outcomes $j = 1, \dots, J$ observed at times $t = 1, \dots, T_i$, the model with unobserved heterogeneity at the panel level is

$$U_{ijt} = \mathbf{x}_{it}\boldsymbol{\beta}_j + u_{ij} + \epsilon_{ijt} \tag{2}$$

The variable U_{ijt} measures the utility of the ith panel toward outcome j at time t and is the sum of observed and unobserved components. The observed part of U_{ijt} consists of $\mathbf{x}_{it}$, a row vector of observed covariates of the ith panel at time t, and $\boldsymbol{\beta}_j$, a column vector of coefficients for the jth outcome. The vector of covariates is the same for each outcome, and the covariates do not vary over the outcomes for a given panel at a given time point. The unobserved part of U_{ijt} consists of u_{ij} and ϵ_{ijt}, where u_{ij} is a panel-level unobserved heterogeneity term and ϵ_{ijt} is the observation-level error term. For model identification, (2) must be normalized with respect to a base category.

Assuming a type-1 extreme value distribution for ϵ_{ijt} gives rise to the MNL model

$$\Pr(y_{it} = m \,|\, \mathbf{x}_{it}, \boldsymbol{\beta}, u_{ij}) = F(y_{it} = m, \mathbf{x}_{it}\boldsymbol{\beta}_j + u_{ij}) = \frac{\exp(\mathbf{x}_{it}\boldsymbol{\beta}_m + u_{im})}{\sum_{j=1}^{J} \exp(\mathbf{x}_{it}\boldsymbol{\beta}_j + u_{ij})}$$

For normalization, $\boldsymbol{\beta}_j$ and u_{ij} are set to zero for $j = b$, where b is the base outcome.

The random-effects estimator of `xtmlogit` assumes that $\mathbf{u}_i$ is distributed $\mathbf{u}_i \sim N(\mathbf{0}, \boldsymbol{\Sigma}_u)$. The likelihood for the ith panel is

$$\begin{aligned} l_i &= \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \left\{ \prod_{t=1}^{T_i} F(y_{it} = m, \mathbf{x}_{it}\boldsymbol{\beta}_j + u_{ij}) \right\} \phi(\mathbf{u}_i, \boldsymbol{\Sigma}_u)\, d\mathbf{u}_i \\ &\equiv \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(y_{it} = m, \eta_{ijt})\, d\mathbf{u}_i \end{aligned}$$

where ϕ is the probability density function of the normal distribution and $\eta_{ijt} = \mathbf{x}_{it}\boldsymbol{\beta}_j + u_{ij}$. This integral of dimension $J - 1$ must be approximated numerically because it has no closed-form solution.

In the univariate case, the integral of a function multiplied by the kernel of the standard normal distribution can be approximated using Gauss–Hermite quadrature. For q-point Gauss–Hermite quadrature, let the abscissa and weight pairs be denoted by (a_k^*, w_k^*), $k = 1, \dots, q$. The Gauss–Hermite quadrature approximation is then

$$\int_{-\infty}^{\infty} f(x) \exp(-x^2)\, dx \approx \sum_{k=1}^{q} w_k^* f(a_k^*)$$

Using the standard normal distribution yields the approximation

$$\int_{-\infty}^{\infty} f(x)\phi(x)\, dx \approx \sum_{k=1}^{q} w_k f(a_k)$$

where $a_k = \sqrt{2}a_k^*$ and $w_k = w_k^*/\sqrt{\pi}$.

We can use a change-of-variables technique to transform the multivariate integral into a set of nested univariate integrals. Each univariate integral can then be evaluated using Gauss–Hermite quadrature. Let $\boldsymbol{v}$ be a random vector whose elements are independently standard normal, and let $\mathbf{L}$ be the Cholesky decomposition of $\boldsymbol{\Sigma}_u$; that is, $\boldsymbol{\Sigma}_u = \mathbf{L}\mathbf{L}'$. In the distribution, we have that $\mathbf{u}_i \approx \mathbf{L}\boldsymbol{v}$, and the linear predictions vector as a function of $\boldsymbol{v}$ is

$$\tilde{\eta}_{ijt} = \mathbf{x}_{it}\boldsymbol{\beta}_j + \mathbf{L}\boldsymbol{v}$$

so the likelihood for a given panel is

$$l_i = (2\pi)^{-r/2} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \exp\left\{ \log f(\mathbf{y}_i, \boldsymbol{\eta}_i) - \frac{1}{2}\sum_{k=1}^{r} v_k^2 \right\} dv_1 \dots dv_r$$

Consider an r-dimensional quadrature grid, $r = J - 1$, containing q quadrature points in each dimension. Let the vector of abscissas $\boldsymbol{a}_\mathbf{k} = (a_{k_1}, \dots, a_{k_r})'$ be a point in this grid, and let $\boldsymbol{w}_\mathbf{k} = (w_{k_1}, \dots, w_{k_r})'$ be the vector of corresponding weights. The Gauss–Hermite quadrature approximation to the likelihood for a given panel is

$$\dot{l}_i = \sum_{k_1=1}^{q} \dots \sum_{k_r=1}^{q} \left[\exp\left\{ \sum_{t=1}^{T_i} \log f(y_{it} = m, \tilde{\eta}_{ijt\mathbf{k}}) \right\} \prod_{s=1}^{r} w_{k_s} \right]$$

where

$$\tilde{\eta}_{ijt\mathbf{k}} = \mathbf{x}_{it}\boldsymbol{\beta}_j + \mathbf{L}\boldsymbol{a}_\mathbf{k}$$

In the case of adaptive Gauss–Hermite quadrature, the likelihood is approximated with

$$\ddot{l}_i = \sum_{k_1=1}^{q} \dots \sum_{k_r=1}^{q} \left[\exp\left\{ \sum_{t=1}^{T_i} \log f(y_{it} = m, \check{\eta}_{ijt\mathbf{k}}) \right\} \prod_{s=1}^{r} \omega_{k_s} \right]$$

where

$$\check{\eta}_{ijt\mathbf{k}} = \mathbf{x}_{it}\boldsymbol{\beta}_j + \mathbf{L}\boldsymbol{\alpha}_\mathbf{k}$$

and $\boldsymbol{\alpha}_\mathbf{k}$ and the ω_{k_s} are the adaptive versions of the abscissas and weights after an orthogonalizing transformation, which eliminates posterior covariances between the latent variables. $\boldsymbol{\alpha}_\mathbf{k}$ and the ω_{k_s} are functions of $\boldsymbol{a}_\mathbf{k}$ and $\boldsymbol{w}_\mathbf{k}$ as well as the posterior mean and the posterior variance of $\boldsymbol{v}$.

The fixed-effects estimator follows the derivations in Chamberlain (1980) and Pforr (2014). Let $\mathbf{Y}_i = (Y_{i1}, \dots, Y_{iT_i})$ be the sequence of outcomes of the ith panel, and let $\mathbf{Y}_{it} = (Y_{i1t}, \dots, Y_{iJt})$ be a vector with elements $Y_{ijt} = 1(i$ chooses $j$at $t)$ that indicate the chosen outcome of the ith panel at time t.

The distribution of times that panel i chose each of the J alternatives over time points T_i is the sufficient statistic $\Theta_i = \sum_{t=1}^{T_i} \mathbf{Y}_{it} = \mathbf{c}_i = (c_{i1}, \dots, c_{iJ})$. Conditioning on the sufficient statistic Θ_i, we find the probability of panel i choosing a sequence $\mathbf{Y}_i = \mathbf{s}_i$ that is consistent with $\mathbf{c}_i$ is

$$\begin{aligned}
\Pr(\mathbf{Y}_i = \mathbf{s}_i \mid \Theta_i, \mathbf{u}_i, \mathbf{x}_i, \boldsymbol{\beta}) &= \Pr\{Y_{i1}, \dots, Y_{iT_i} \mid \Psi(\mathbf{c}_i), \mathbf{u}_i, \mathbf{x}_i, \boldsymbol{\beta}\} \\
&= \frac{\exp\left(\sum_{t=1}^{T_i} \sum_{j=1, j\neq b}^{J} Y_{ijt}\mathbf{x}_{it}\boldsymbol{\beta}_j \right)}{\sum_{\widetilde{Y}_{ijt} \in \Psi(\mathbf{c}_i)} \exp\left(\sum_{t=1}^{T_i} \sum_{j=1, j\neq b}^{J} \widetilde{Y}_{ijt}\mathbf{x}_{it}\boldsymbol{\beta}_j \right)}
\end{aligned}$$

where $\Psi(\mathbf{c}_i)$ is the set of all permutations of individual i's observed sequence of outcomes that satisfy the condition $\sum_{t=1}^{T_i} \widetilde{\mathbf{Y}}_{it} = \mathbf{c}_i$. That is,

$$\Psi(\mathbf{c}_i) = \left\{ \widetilde{\mathbf{Y}}_i = (\widetilde{Y}_{i1}, \dots, \widetilde{Y}_{iT_i}) \,\middle|\, \sum_{t=1}^{T_i} \widetilde{\mathbf{Y}}_{it} = \mathbf{c}_i \right\}$$

and $\widetilde{\mathbf{Y}}_{it} = (\widetilde{Y}_{i1t}, \ldots, \widetilde{Y}_{iJt})$ is a vector of indicators with respect to the permutations of the observed outcome sequence $\mathbf{Y}_i$. The log likelihood of panel i is then the natural logarithm of the above probability

$$\log l_i = \sum_{t=1}^{T_i} \sum_{j=1,j\neq b}^{J} Y_{ijt}\mathbf{x}_{it}\boldsymbol{\beta}_j - \log \sum_{\widetilde{Y}_{ijt}\in\Psi(\mathbf{c}_i)} \exp\left(\sum_{t=1}^{T_i} \sum_{j=1,j\neq b}^{J} \widetilde{Y}_{ijt}\mathbf{x}_{it}\boldsymbol{\beta}_j\right)$$

and the overall log likelihood is $\sum_{i=1}^{N} \log l_i$.

Consistent, albeit less efficient, estimates of the parameters in $\boldsymbol{\beta}_j$ can be obtained by taking a random sample of the elements in $\Psi(\mathbf{c}_i)$. The total number of permutations in $\Psi(\mathbf{c}_i)$ is

$$K_i = \frac{T_i!}{c_{i1}!\cdots c_{ij}!\cdots c_{iJ}!}$$

Let $\ddot{\Psi}(\mathbf{c}_i)$ be a random subset of $\Psi(\mathbf{c}_i)$. $\ddot{\Psi}(\mathbf{c}_i)$ consists of $L_i + 1$ elements, where L_i elements are randomly drawn without replacement and equal probability from the set $\Psi(\mathbf{c}_i)$ that has the observed sequence of outcomes removed, and then the observed sequence is added such that $\ddot{\Psi}(\mathbf{c}_i)$ always contains the observed outcome sequence. The log likelihood with sampled permutations is

$$\log l_i = \sum_{t=1}^{T_i} \sum_{j=1,j\neq b}^{J} Y_{ijt}\mathbf{x}_{it}\boldsymbol{\beta}_j - \log \sum_{\widetilde{Y}_{ijt}\in\ddot{\Psi}(\mathbf{c}_i)} \exp\left(\sum_{t=1}^{T_i} \sum_{j=1,j\neq b}^{J} \widetilde{Y}_{ijt}\mathbf{x}_{it}\boldsymbol{\beta}_j\right)$$

The above is a consistent estimator of $\boldsymbol{\beta}_j$ but is less efficient compared with using the full set of permutations $\Psi(\mathbf{c}_i)$ because of the added Monte Carlo variance. The smaller the size of the sample relative to K_i, the number of all permutations, the less efficient it becomes. The permutation sampling implemented in `xtmlogit` follows the approach outlined in D'Haultfœuille and Iaria (2016).

References

Andersen, E. B. 1970. Asymptotic properties of conditional maximum-likelihood estimators. *Journal of the Royal Statistical Society*, B ser., 32: 283–301. https://doi.org/10.1111/j.2517-6161.1970.tb00842.x.

Börsch-Supan, A. 1990. Panel data analysis of housing choices. *Regional Science and Urban Economics* 20: 65–82. https://doi.org/10.1016/0166-0462(90)90025-X.

Chamberlain, G. 1980. Analysis of covariance with qualitative data. *Review of Economic Studies* 47: 225–238. https://doi.org/10.2307/2297110.

D'Haultfœuille, X., and A. Iaria. 2016. A convenient method for the estimation of the multinomial logit model with fixed effects. *Economics Letters* 141: 77–79. https://doi.org/10.1016/j.econlet.2016.02.002.

Grilli, L., and C. Rampichini. 2007. A multilevel multinomial logit model for the analysis of graduates' skills. *Statistical Methods and Applications* 16: 381–393. https://doi.org/10.1007/s10260-006-0039-z.

Hartzel, J., A. Agresti, and B. S. Caffo. 2001. Multinomial logit random effects models. *Statistical Modelling* 1: 81–102. https://doi.org/10.1177/1471082X0100100201.

Lancaster, T. 2000. The incidental parameter problem since 1948. *Journal of Econometrics* 95: 391–413. https://doi.org/10.1016/S0304-4076(99)00044-5.

Pforr, K. 2014. femlogit—Implementation of the multinomial logit model with fixed effects. *Stata Journal* 14: 847–862.

Also see

[XT] **xtmlogit postestimation** — Postestimation tools for xtmlogit

[XT] **quadchk** — Check sensitivity of quadrature approximation

[XT] **xtlogit** — Fixed-effects, random-effects, and population-averaged logit models

[XT] **xtset** — Declare data to be panel data

[BAYES] **bayes: xtmlogit** — Bayesian random-effects multinomial logit model

[R] **clogit** — Conditional (fixed-effects) logistic regression

[R] **mlogit** — Multinomial (polytomous) logistic regression

[R] **mprobit** — Multinomial probit regression

[SVY] **svy estimation** — Estimation commands for survey data

[U] 20 Estimation and postestimation commands

Postestimation commands

The following postestimation command is of special interest after `xtmlogit`:

Command	Description
`estat sd`	display variance components as standard deviations and correlations

The following standard postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of parameters
`estat ic`	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC, respectively)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
* `hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
* `lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	probabilities, etc.
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of parameters
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

*`hausman` and `lrtest` are not appropriate with `svy` estimation results.

predict

Description for predict

predict creates a new variable containing predictions such as probabilities and linear predictions.

Menu for predict

Statistics > Postestimation

Syntax for predict

Random-effects model

predict [*type*] { *stub*∗ | *newvar* | *newvarlist* } [*if*] [*in*] [, *RE_statistic*
  outcome(*outcome*)]

predict [*type*] { *stub*∗ | *newvarlist* } [*if*] [*in*] , scores

Fixed-effects model

predict [*type*] { *stub*∗ | *newvar* | *newvarlist* } [*if*] [*in*] [, *FE_statistic*
  outcome(*outcome*)]

predict [*type*] { *stub*∗ | *newvarlist* } [*if*] [*in*] , scores

RE_statistic	Description
Main	
pr	marginal probability of the specified outcome; the default
pcr	conditional probability of the specified outcome
pu0	probability of the specified outcome, assuming zero random effects
xb	linear prediction of the specified outcome, including random effects
xb0	linear prediction of the specified outcome, assuming zero random effects

FE_statistic	Description
Main	
pu0	probability of the specified outcome, assuming zero fixed effects; the default
xb	linear prediction for the specified outcome, assuming zero fixed effects

You specify one or k new variables, where k is the number of outcomes. If you specify one new variable and you do not specify outcome(), then outcome(#1) is assumed.

These statistics are available both in and out of sample; type predict ... if e(sample) ... if wanted only for the estimation sample.

Options for predict

Main

`pr` (after `xtmlogit, re` only) calculates predicted probabilities that are marginal with respect to the random effects, which means that the probabilities are calculated by integrating the prediction function with respect to the random effect over its entire support. This prediction method is of primary interest if population-averaged probabilities are desired. If `outcome()` is not specified, `pr` defaults to the first outcome. This is the default for the random-effects model.

`pcr` (after `xtmlogit, re` only) calculates predicted probabilities that are conditional on the random effects. In contrast to `predict, pr`, the random effects are not integrated out but are set to their predicted value when predictions are computed from the logistic cumulative distribution function. The random effects are calculated using their posterior means, which are approximated using mean-variance adaptive Gaussian quadrature. This prediction method is useful if observation-level predictions, rather than averaged predicted probabilities, are of interest. If `outcome()` is not specified, `pcr` defaults to the first outcome.

`pu0` calculates predicted probabilities, assuming that the fixed or random effect for that observation's panel is zero ($u_i = 0$). If `outcome()` is not specified, `pu0` defaults to the first outcome. This is the default for the fixed-effects model.

`xb` calculates the linear prediction. This includes the random effect in the case of `xtmlogit, re`. In the case of `xtmlogit, fe`, the fixed effect is assumed to be zero. If `outcome()` is not specified, `xb` defaults to the first outcome.

`xb0` (after `xtmlogit, re` only) calculates the linear prediction, excluding the random effect. If `outcome()` is not specified, `xb0` defaults to the first outcome.

`scores` calculates parameter-level scores, the first derivatives of the log likelihood with respect to β_j.

`outcome(`*outcome*`)` specifies the outcome for which the predicted probabilities or linear predictions are to be calculated. `outcome()` can only be used when one variable is specified. `outcome()` should contain either one value of the dependent variable or one of `#1`, `#2`, ..., with `#1` meaning the first category of the dependent variable, `#2` meaning the second category, etc.

margins

Description for margins

`margins` estimates margins of response for probabilities and linear predictions.

Menu for margins

Statistics > Postestimation

Syntax for margins

`margins` [*marginlist*] [, *options*]

`margins` [*marginlist*] , `predict(`*statistic* ...`)` [`predict(`*statistic* ...`)` ...] [*options*]

Random-effects model

statistic	Description
default	marginal probability for each outcome
`pr`	marginal probability of the specified outcome
`pu0`	probability of the specified outcome, assuming random effect is zero
`xb`	linear prediction of the specified outcome equation, including random effect
`xb0`	linear prediction of the specified outcome equation, assuming random effect is zero
`pcr`	not allowed with `margins`

Fixed-effects model

statistic	Description
default	probability for each outcome, assuming fixed effect is zero
`pu0`	probability of the specified outcome, assuming fixed effect is zero
`xb`	linear prediction of the specified outcome equation, assuming fixed effect is zero

`pr`, `pu0`, `xb`, and `xb0` default to the first outcome when `outcome()` is not specified.

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.

For the full syntax, see [R] **margins**.

Also see

[XT] **xtmlogit** — Fixed-effects and random-effects multinomial logit models

[U] 20 Estimation and postestimation commands

xtnbreg — Fixed-effects, random-effects, & population-averaged negative binomial models

Description

`xtnbreg` fits random-effects and conditional fixed-effects overdispersion models where the random effects or fixed effects apply to the distribution of the dispersion parameter. The dispersion is the same for all observations in the same panel. In the random-effects model, the dispersion varies randomly from group to group, such that the inverse of one plus the dispersion follows a Beta distribution. In the fixed-effects model, the dispersion parameter in a group can take on any value.

`xtnbreg` also fits a population-averaged negative binomial model for a nonnegative count dependent variable with overdispersion.

Quick start

Random-effects negative-binomial regression of `y` on `x` and indicators for levels of categorical variable `a` using `xtset` data

```
xtnbreg y x i.a
```

Same as above, but report incidence-rate ratios

```
xtnbreg y x i.a, irr
```

Conditional fixed-effects model with exposure variable `evar`

```
xtnbreg y x i.a, fe exposure(evar)
```

Population-averaged model with robust standard errors

```
xtnbreg y x i.a, pa vce(robust)
```

Menu

Statistics > Longitudinal/panel data > Count outcomes > Negative binomial regression (FE, RE, PA)

Syntax

Random-effects (RE) and conditional fixed-effects (FE) overdispersion models

`xtnbreg` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, [`re` | `fe`] *RE/FE_options*]

Population-averaged (PA) model

`xtnbreg` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] , `pa` [*PA_options*]

RE/FE_options	Description
Model	
`noconstant`	suppress constant term; not available with `fe`
`re`	use random-effects estimator; the default
`fe`	use fixed-effects estimator
`exposure(`*varname*`)`	include ln(*varname*) in model with coefficient constrained to 1
`offset(`*varname*`)`	include *varname* in model with coefficient constrained to 1
`constraints(`*constraints*`)`	apply specified linear constraints
SE	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`irr`	report incidence-rate ratios
`lrmodel`	perform the likelihood-ratio model test instead of the default Wald test
`nocnsreport`	do not display constraints
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
maximize_options	control the maximization process; seldom used
`collinear`	keep collinear variables
`coeflegend`	display legend instead of statistics

PA_options	Description
Model	
noconstant	suppress constant term
pa	use population-averaged estimator
exposure(*varname*)	include ln(*varname*) in model with coefficient constrained to 1
offset(*varname*)	include *varname* in model with coefficient constrained to 1
Correlation	
corr(*correlation*)	within-panel correlation structure
force	estimate even if observations unequally spaced in time
SE/Robust	
vce(*vcetype*)	*vcetype* may be conventional, robust, bootstrap, or jackknife
nmp	use divisor $N-P$ instead of the default N
scale(*parm*)	overrides the default scale parameter; *parm* may be x2, dev, phi, or #
Reporting	
level(#)	set confidence level; default is level(95)
irr	report incidence-rate ratios
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Optimization	
optimize_options	control the optimization process; seldom used
coeflegend	display legend instead of statistics

correlation	Description
exchangeable	exchangeable
independent	independent
unstructured	unstructured
fixed *matname*	user-specified
ar #	autoregressive of order #
stationary #	stationary of order #
nonstationary #	nonstationary of order #

A panel variable must be specified. For xtnbreg, pa, correlation structures other than exchangeable and independent require that a time variable also be specified. Use xtset; see [XT] **xtset**.

indepvars may contain factor variables; see [U] **11.4.3 Factor variables**.

depvar and *indepvars* may contain time-series operators; see [U] **11.4.4 Time-series varlists**.

by, collect, mi estimate, and statsby are allowed; see [U] **11.1.10 Prefix commands**. bayes is allowed for the random-effects model. For more details, see [BAYES] **bayes: xtnbreg**. fp is allowed for the random-effects and fixed-effects models.

vce(bootstrap) and vce(jackknife) are not allowed with the mi estimate prefix; see [MI] **mi estimate**.

iweights, fweights, and pweights are allowed for the population-averaged model, and iweights are allowed in the random-effects and fixed-effects models; see [U] **11.1.6 weight**. Weights must be constant within panel.

collinear and coeflegend do not appear in the dialog box.

See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options for RE/FE models

Model

noconstant; see [R] **Estimation options**.

re requests the random-effects estimator, which is the default.

fe requests the conditional fixed-effects estimator.

exposure(*varname*), offset(*varname*), constraints(*constraints*); see [R] **Estimation options**.

SE

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (oim) and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] ***vce_options***.

Reporting

level(#); see [R] **Estimation options**.

irr reports exponentiated coefficients e^b rather than coefficients b. For the negative binomial model, exponentiated coefficients have the interpretation of incidence-rate ratios.

lrmodel, nocnsreport; see [R] **Estimation options**.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Maximization

maximize_options: difficult, technique(*algorithm_spec*), iterate(#), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), nonrtolerance, and from(*init_specs*); see [R] **Maximize**. These options are seldom used.

The following options are available with xtnbreg but are not shown in the dialog box:

collinear, coeflegend; see [R] **Estimation options**.

Options for PA model

Model

noconstant; see [R] **Estimation options**.

pa requests the population-averaged estimator.

exposure(*varname*), offset(*varname*); see [R] **Estimation options**.

Correlation

corr(*correlation*) specifies the within-panel correlation structure; the default corresponds to the equal-correlation model, corr(exchangeable).

When you specify a correlation structure that requires a lag, you indicate the lag after the structure's name with or without a blank; for example, corr(ar 1) or corr(ar1).

If you specify the fixed correlation structure, you specify the name of the matrix containing the assumed correlations following the word `fixed`, for example, `corr(fixed myr)`.

`force` specifies that estimation be forced even though the time variable is not equally spaced. This is relevant only for correlation structures that require knowledge of the time variable. These correlation structures require that observations be equally spaced so that calculations based on lags correspond to a constant time change. If you specify a time variable indicating that observations are not equally spaced, the (time dependent) model will not be fit. If you also specify `force`, the model will be fit, and it will be assumed that the lags based on the data ordered by the time variable are appropriate.

SE/Robust

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`conventional`), that are robust to some kinds of misspecification (`robust`), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [XT] ***vce_options***.

`vce(conventional)`, the default, uses the conventionally derived variance estimator for generalized least-squares regression.

`nmp`, `scale(x2` | `dev` | `phi` | #`)`; see [XT] ***vce_options***.

Reporting

`level(`#`)`; see [R] **Estimation options**.

`irr` reports exponentiated coefficients e^b rather than coefficients b. For the negative binomial model, exponentiated coefficients have the interpretation of incidence-rate ratios.

display_options: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(`#`)`, `fvwrapon(`*style*`)`, `cformat(`%*fmt*`)`, `pformat(`%*fmt*`)`, `sformat(`%*fmt*`)`, and `nolstretch`; see [R] **Estimation options**.

Optimization

optimize_options control the iterative optimization process. These options are seldom used.

`iterate(`#`)` specifies the maximum number of iterations. When the number of iterations equals #, the optimization stops and presents the current results, even if convergence has not been reached. The default is `iterate(100)`.

`tolerance(`#`)` specifies the tolerance for the coefficient vector. When the relative change in the coefficient vector from one iteration to the next is less than or equal to #, the optimization process is stopped. `tolerance(1e-6)` is the default.

`log` and `nolog` specify whether to display the iteration log. The iteration log is displayed by default unless you used `set iterlog off` to suppress it; see `set iterlog` in [R] ***set iter***.

`trace` specifies that the current estimates be printed at each iteration.

The following option is available with `xtnbreg` but is not shown in the dialog box:

`coeflegend`; see [R] **Estimation options**.

Remarks and examples

xtnbreg fits random-effects overdispersion models, conditional fixed-effects overdispersion models, and population-averaged negative binomial models. Here “random effects” and “fixed effects” apply to the distribution of the dispersion parameter, not to the $\mathbf{x}\boldsymbol{\beta}$ term in the model. In the random-effects and fixed-effects overdispersion models, the dispersion is the same for all elements in the same group (that is, elements with the same value of the panel variable). In the random-effects model, the dispersion varies randomly from group to group, such that the inverse of one plus the dispersion follows a Beta(r, s) distribution. In the fixed-effects model, the dispersion parameter in a group can take on any value, because a conditional likelihood is used in which the dispersion parameter drops out of the estimation.

By default, the population-averaged model is an equal-correlation model; `xtnbreg, pa` assumes `corr(exchangeable)`. Thus, `xtnbreg` is a convenience command for fitting the population-averaged using `xtgee`; see [XT] **xtgee**. Typing

```
. xtnbreg ..., ... pa exposure(time)
```

is equivalent to typing

```
. xtgee ..., ... family(nbinomial) link(log) corr(exchangeable) exposure(time)
```

See also [XT] **xtgee** for information about `xtnbreg`.

By default, or when `re` is specified, `xtnbreg` fits a maximum-likelihood random-effects overdispersion model.

▷ Example 1

You have (fictional) data on injury “incidents” incurred among 20 airlines in each of 4 years. (Incidents range from major injuries to exceedingly minor ones.) The government agency in charge of regulating airlines has run an experimental safety training program, and, in each of the years, some airlines have participated and some have not. You now wish to analyze whether the “incident” rate is affected by the program. You choose to estimate using random-effects negative binomial regression, as the dispersion might vary across the airlines for unidentified airline-specific reasons. Your measure of exposure is passenger miles for each airline in each year.

```
. use https://www.stata-press.com/data/r19/airacc
. xtnbreg i_cnt inprog, exposure(pmiles) irr
Fitting negative binomial (constant dispersion) model:
Iteration 0:  Log likelihood = -293.57997
Iteration 1:  Log likelihood = -293.57997
  (output omitted)
Fitting full model:
Iteration 0:  Log likelihood = -295.72633
Iteration 1:  Log likelihood = -270.49929  (not concave)
  (output omitted)
Random-effects negative binomial regression         Number of obs    =     80
Group variable: airline                             Number of groups =     20
Random effects u_i ~ Beta                           Obs per group:
                                                                 min =      4
                                                                 avg =    4.0
                                                                 max =      4
                                                    Wald chi2(1)     =   2.04
Log likelihood = -265.38202                         Prob > chi2      = 0.1532

------------------------------------------------------------------------------
       i_cnt |        IRR   Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      inprog |    .911673   .0590277    -1.43   0.153     .8030206    1.035027
       _cons |   .0367524   .0407032    -2.98   0.003     .0041936    .3220983
  ln(pmiles) |          1  (exposure)
-------------+----------------------------------------------------------------
       /ln_r |   4.794991    .951781                      2.929535    6.660448
       /ln_s |   3.268052   .4709033                      2.345098    4.191005
-------------+----------------------------------------------------------------
           r |   120.9033   115.0735                      18.71892    780.9007
           s |   26.26013   12.36598                       10.4343    66.08918
------------------------------------------------------------------------------
Note: Estimates are transformed only in the first equation to incidence-rate
      ratios.
Note: _cons estimates baseline incidence rate (conditional on zero random
      effects).
LR test vs. pooled: chibar2(01) = 19.03               Prob >= chibar2 = 0.000
```

In the output above, the `/ln_r` and `/ln_s` lines refer to $\ln(r)$ and $\ln(s)$, where the inverse of one plus the dispersion is assumed to follow a Beta(r,s) distribution. The output also includes a likelihood-ratio test, which compares the panel estimator with the pooled estimator (that is, a negative binomial estimator with constant dispersion).

You find that the incidence rate for accidents is not significantly different for participation in the program and that the panel estimator is significantly different from the pooled estimator.

We may alternatively fit a fixed-effects overdispersion model:

```
. xtnbreg i_cnt inprog, exposure(pmiles) irr fe nolog

Conditional FE negative binomial regression         Number of obs    =     80
Group variable: airline                             Number of groups =     20

                                                    Obs per group:
                                                                 min =      4
                                                                 avg =    4.0
                                                                 max =      4

                                                    Wald chi2(1)     =   2.11
Log likelihood = -174.25143                         Prob > chi2      = 0.1463

------------------------------------------------------------------------------
       i_cnt |        IRR   Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      inprog |   .9062669   .0613917    -1.45   0.146      .793587    1.034946
       _cons |   .0329025   .0331262    -3.39   0.001     .0045734    .2367111
  ln(pmiles) |          1  (exposure)
------------------------------------------------------------------------------
Note: _cons estimates baseline incidence rate (conditional on zero random
      effects).
```

◁

▷ Example 2

We rerun our previous example, but this time we fit a robust equal-correlation population-averaged model:

```
. xtnbreg i_cnt inprog, exposure(pmiles) irr vce(robust) pa

Iteration 1:  Tolerance = .02499392
Iteration 2:  Tolerance = .0000482
Iteration 3:  Tolerance = 2.929e-07

GEE population-averaged model                       Number of obs    =     80
Group variable: airline                             Number of groups =     20
Family: Negative binomial(k=1)                      Obs per group:
Link:   Log                                                      min =      4
Correlation: exchangeable                                        avg =    4.0
                                                                 max =      4
                                                    Wald chi2(1)     =   1.28
Scale parameter = 1                                 Prob > chi2      = 0.2571

                               (Std. err. adjusted for clustering on airline)
------------------------------------------------------------------------------
             |             Semirobust
       i_cnt |        IRR   std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      inprog |    .927275   .0617857    -1.13   0.257     .8137513    1.056636
       _cons |   .0080211   .0004117   -94.02   0.000     .0072535      .00887
  ln(pmiles) |          1  (exposure)
------------------------------------------------------------------------------
Note: _cons estimates baseline incidence rate (conditional on zero random
      effects).
```

We compare this with a pooled estimator with clustered robust-variance estimates:

```
. nbreg i_cnt inprog, exposure(pmiles) irr vce(cluster airline)
Fitting Poisson model:
Iteration 0:  Log pseudolikelihood = -293.57997
Iteration 1:  Log pseudolikelihood = -293.57997
Fitting constant-only model:
Iteration 0:  Log pseudolikelihood = -335.13615
Iteration 1:  Log pseudolikelihood = -279.43327
Iteration 2:  Log pseudolikelihood = -276.09296
Iteration 3:  Log pseudolikelihood = -274.84036
Iteration 4:  Log pseudolikelihood = -274.81076
Iteration 5:  Log pseudolikelihood = -274.81075
Fitting full model:
Iteration 0:  Log pseudolikelihood = -274.56985
Iteration 1:  Log pseudolikelihood = -274.55077
Iteration 2:  Log pseudolikelihood = -274.55077
Negative binomial regression                            Number of obs =     80
                                                        Wald chi2(1)  =   0.60
Dispersion: mean                                        Prob > chi2   = 0.4369
Log pseudolikelihood = -274.55077                       Pseudo R2     = 0.0009
                               (Std. err. adjusted for 20 clusters in airline)
------------------------------------------------------------------------------
             |               Robust
       i_cnt |        IRR   std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      inprog |   .9429015   .0713091    -0.78   0.437     .8130032    1.093555
       _cons |    .007956   .0004237   -90.77   0.000     .0071674    .0088314
 ln(pmiles) |          1  (exposure)
-------------+----------------------------------------------------------------
    /lnalpha |  -2.835089   .3351784                     -3.492027   -2.178151
-------------+----------------------------------------------------------------
       alpha |   .0587133   .0196794                      .0304391    .1132507
------------------------------------------------------------------------------
Note: Estimates are transformed only in the first equation to incidence-rate
      ratios.
Note: _cons estimates baseline incidence rate.
```

◁

Stored results

xtnbreg, re stores the following in e():

Scalars

e(N)	number of observations
e(N_g)	number of groups
e(k)	number of parameters
e(k_aux)	number of auxiliary parameters
e(k_eq)	number of equations in e(b)
e(k_eq_model)	number of equations in overall model test
e(k_dv)	number of dependent variables
e(df_m)	model degrees of freedom
e(ll)	log likelihood
e(ll_0)	log likelihood, constant-only model
e(ll_c)	log likelihood, comparison model
e(chi2)	χ^2
e(chi2_c)	χ^2 for comparison test
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(r)	value of r in Beta(r, s)
e(s)	value of s in Beta(r, s)
e(p)	p-value for model test
e(rank)	rank of e(V)
e(rank0)	rank of e(V) for constant-only model
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros

e(cmd)	xtnbreg
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(ivar)	variable denoting groups
e(model)	re
e(wtype)	weight type
e(wexp)	weight expression
e(title)	title in estimation output
e(offset)	linear offset variable
e(chi2type)	Wald or LR; type of model χ^2 test
e(chi2_ct)	Wald or LR; type of model χ^2 test corresponding to e(chi2_c)
e(vce)	*vcetype* specified in vce()
e(method)	estimation method
e(distrib)	Beta; the distribution of the random effect
e(opt)	type of optimization
e(which)	max or min; whether optimizer is to perform maximization or minimization
e(ml_method)	type of ml method
e(user)	name of likelihood-evaluator program
e(technique)	maximization technique
e(properties)	b V
e(predict)	program used to implement predict
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved

Matrices

e(b)	coefficient vector
e(Cns)	constraints matrix
e(ilog)	iteration log
e(gradient)	gradient vector
e(V)	variance–covariance matrix of the estimators

Functions

`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

`xtnbreg, fe` stores the following in `e()`:

Scalars

`e(N)`	number of observations
`e(N_g)`	number of groups
`e(k)`	number of parameters
`e(k_eq)`	number of equations in `e(b)`
`e(k_eq_model)`	number of equations in overall model test
`e(k_dv)`	number of dependent variables
`e(df_m)`	model degrees of freedom
`e(ll)`	log likelihood
`e(ll_0)`	log likelihood, constant-only model
`e(chi2)`	χ^2
`e(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(g_max)`	largest group size
`e(p)`	p-value for model test
`e(rank)`	rank of `e(V)`
`e(ic)`	number of iterations
`e(rc)`	return code
`e(converged)`	`1` if converged, `0` otherwise

Macros

`e(cmd)`	`xtnbreg`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(ivar)`	variable denoting groups
`e(model)`	`fe`
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(title)`	title in estimation output
`e(offset)`	linear offset variable
`e(chi2type)`	`LR`; type of model χ^2 test
`e(vce)`	*vcetype* specified in `vce()`
`e(method)`	requested estimation method
`e(opt)`	type of optimization
`e(which)`	`max` or `min`; whether optimizer is to perform maximization or minimization
`e(ml_method)`	type of `ml` method
`e(user)`	name of likelihood-evaluator program
`e(technique)`	maximization technique
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`

Matrices

`e(b)`	coefficient vector
`e(Cns)`	constraints matrix

`e(ilog)`	iteration log
`e(gradient)`	gradient vector
`e(V)`	variance–covariance matrix of the estimators
Functions	
`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices	
`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

`xtnbreg, pa` stores the following in `e()`:

Scalars	
`e(N)`	number of observations
`e(N_g)`	number of groups
`e(df_m)`	model degrees of freedom
`e(chi2)`	χ^2
`e(p)`	p-value for model test
`e(df_pear)`	degrees of freedom for Pearson χ^2
`e(chi2_dev)`	χ^2 test of deviance
`e(chi2_dis)`	χ^2 test of deviance dispersion
`e(deviance)`	deviance
`e(dispers)`	deviance dispersion
`e(phi)`	scale parameter
`e(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(g_max)`	largest group size
`e(rank)`	rank of `e(V)`
`e(tol)`	target tolerance
`e(dif)`	achieved tolerance
`e(rc)`	return code
Macros	
`e(cmd)`	`xtgee`
`e(cmd2)`	`xtnbreg`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(ivar)`	variable denoting groups
`e(tvar)`	variable denoting time within groups
`e(model)`	`pa`
`e(family)`	`negative binomial`($k = 1$)
`e(link)`	`log`; link function
`e(corr)`	correlation structure
`e(scale)`	`x2`, `dev`, `phi`, or #; scale parameter
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(offset)`	linear offset variable
`e(chi2type)`	`Wald`; type of model χ^2 test
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. err.
`e(nmp)`	`nmp`, if specified
`e(nbalpha)`	α
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`

`e(marginsnotok)`	predictions disallowed by `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`

Matrices

`e(b)`	coefficient vector
`e(R)`	estimated working correlation matrix
`e(V)`	variance–covariance matrix of the estimators
`e(V_modelbased)`	model-based variance

Functions

`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

`xtnbreg, pa` reports the population-averaged results obtained by using `xtgee, family(nbinomial) link(log)` to obtain estimates. See [XT] **xtgee** for details on the methods and formulas.

For the random-effects and fixed-effects overdispersion models, let y_{it} be the count for the tth observation in the ith group. We begin with the model $y_{it}\,|\,\gamma_{it} \sim \text{Poisson}(\gamma_{it})$, where $\gamma_{it}\,|\,\delta_i \sim \text{gamma}(\lambda_{it}, \delta_i)$ with $\lambda_{it} = \exp(\mathbf{x}_{it}\boldsymbol{\beta} + \text{offset}_{it})$ and δ_i is the dispersion parameter. This yields the model

$$\Pr(Y_{it} = y_{it} \mid \mathbf{x}_{it}, \delta_i) = \frac{\Gamma(\lambda_{it} + y_{it})}{\Gamma(\lambda_{it})\Gamma(y_{it}+1)} \left(\frac{1}{1+\delta_i}\right)^{\lambda_{it}} \left(\frac{\delta_i}{1+\delta_i}\right)^{y_{it}}$$

(See Hausman, Hall, and Griliches [1984, eq. 3.1, 922]; our δ is the inverse of their δ.) Looking at within-panel effects only, we find that this specification yields a negative binomial model for the ith group with dispersion (variance divided by the mean) equal to $1 + \delta_i$, that is, constant dispersion within group. This parameterization of the negative binomial model differs from the default parameterization of `nbreg`, which has dispersion equal to $1 + \alpha\exp(\mathbf{x}\boldsymbol{\beta} + \text{offset})$; see [R] **nbreg**.

For a random-effects overdispersion model, we allow δ_i to vary randomly across groups; namely, we assume that $1/(1+\delta_i) \sim \text{Beta}(r, s)$. The joint probability of the counts for the ith group is

$$\begin{aligned}\Pr(Y_{i1} = y_{i1}, \ldots, Y_{in_i} = y_{in_i} | \mathbf{X}_i) &= \int_0^\infty \prod_{t=1}^{n_i} \Pr(Y_{it} = y_{it} \mid \mathbf{x}_{it}, \delta_i)\, f(\delta_i)\, d\delta_i \\ &= \frac{\Gamma(r+s)\Gamma(r + \sum_{t=1}^{n_i} \lambda_{it})\Gamma(s + \sum_{t=1}^{n_i} y_{it})}{\Gamma(r)\Gamma(s)\Gamma(r + s + \sum_{t=1}^{n_i} \lambda_{it} + \sum_{t=1}^{n_i} y_{it})} \prod_{t=1}^{n_i} \frac{\Gamma(\lambda_{it} + y_{it})}{\Gamma(\lambda_{it})\Gamma(y_{it}+1)}\end{aligned}$$

for $\mathbf{X}_i = (\mathbf{x}_{i1}, \dots, \mathbf{x}_{in_i})$ and where f is the probability density function for δ_i. The resulting log likelihood is

$$\begin{aligned} \ln L = \sum_{i=1}^{n} w_i \Bigg[& \ln\Gamma(r+s) + \ln\Gamma\left(r + \sum_{k=1}^{n_i} \lambda_{ik}\right) + \ln\Gamma\left(s + \sum_{k=1}^{n_i} y_{ik}\right) - \ln\Gamma(r) - \ln\Gamma(s) \\ & - \ln\Gamma\left(r + s + \sum_{k=1}^{n_i} \lambda_{ik} + \sum_{k=1}^{n_i} y_{ik}\right) + \sum_{t=1}^{n_i} \Big\{ \ln\Gamma(\lambda_{it} + y_{it}) - \ln\Gamma(\lambda_{it}) - \ln\Gamma(y_{it}+1) \Big\} \Bigg] \end{aligned}$$

where $\lambda_{it} = \exp(\mathbf{x}_{it}\boldsymbol{\beta} + \text{offset}_{it})$ and w_i is the weight for the ith group (Hausman, Hall, and Griliches 1984, eq. 3.5, 927).

For the fixed-effects overdispersion model, we condition the joint probability of the counts for each group on the sum of the counts for the group (that is, the observed $\sum_{t=1}^{n_i} y_{it}$). This yields

$$\begin{aligned} \Pr(Y_{i1} = y_{i1}, \dots, Y_{in_i} = y_{in_i} \mid \mathbf{X}_i, \textstyle\sum_{t=1}^{n_i} Y_{it} = \sum_{t=1}^{n_i} y_{it}) \\ = \frac{\Gamma(\sum_{t=1}^{n_i} \lambda_{it})\Gamma(\sum_{t=1}^{n_i} y_{it} + 1)}{\Gamma(\sum_{t=1}^{n_i} \lambda_{it} + \sum_{t=1}^{n_i} y_{it})} \prod_{t=1}^{n_i} \frac{\Gamma(\lambda_{it} + y_{it})}{\Gamma(\lambda_{it})\Gamma(y_{it}+1)} \end{aligned}$$

The conditional log likelihood is

$$\begin{aligned} \ln L = \sum_{i=1}^{n} w_i \Bigg[& \ln\Gamma\left(\sum_{t=1}^{n_i} \lambda_{it}\right) + \ln\Gamma\left(\sum_{t=1}^{n_i} y_{it} + 1\right) - \ln\Gamma\left(\sum_{t=1}^{n_i} \lambda_{it} + \sum_{t=1}^{n_i} y_{it}\right) \\ & + \sum_{t=1}^{n_i} \Big\{ \ln\Gamma(\lambda_{it} + y_{it}) - \ln\Gamma(\lambda_{it}) - \ln\Gamma(y_{it}+1) \Big\} \Bigg] \end{aligned}$$

See Hausman, Hall, and Griliches (1984) for a more thorough development of the random-effects and fixed-effects models. Also see Cameron and Trivedi (2013) for a good textbook treatment of this model.

References

Cameron, A. C., and P. K. Trivedi. 2013. *Regression Analysis of Count Data*. 2nd ed. New York: Cambridge University Press.

Guimarães, P. 2005. A simple approach to fit the beta-binomial model. *Stata Journal* 5: 385–394.

Hausman, J. A., B. H. Hall, and Z. Griliches. 1984. Econometric models for count data with an application to the patents–R & D relationship. *Econometrica* 52: 909–938. https://doi.org/10.2307/1911191.

Liang, K.-Y., and S. L. Zeger. 1986. Longitudinal data analysis using generalized linear models. *Biometrika* 73: 13–22. https://doi.org/10.1093/biomet/73.1.13.

Also see

[XT] **xtnbreg postestimation** — Postestimation tools for xtnbreg

[XT] **xtgee** — GEE population-averaged panel-data models

[XT] **xtpoisson** — Fixed-effects, random-effects, and population-averaged Poisson models

[XT] **xtset** — Declare data to be panel data

[BAYES] **bayes: xtnbreg** — Bayesian random-effects negative binomial model

[ME] **menbreg** — Multilevel mixed-effects negative binomial regression

[MI] **Estimation** — Estimation commands for use with mi estimate

[R] **nbreg** — Negative binomial regression

[U] 20 Estimation and postestimation commands

Postestimation commands

The following postestimation commands are available after `xtnbreg`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of parameters
* `estat ic`	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC, respectively)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
† `forecast`	dynamic forecasts and simulations
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
* `lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	linear predictions and their SEs, number of events, incidence rates, probabilities
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of parameters
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

*`estat ic` and `lrtest` are not appropriate after `xtnbreg, pa`.

†`forecast` is not appropriate with `mi` estimation results.

predict

Description for predict

`predict` creates a new variable containing predictions such as linear predictions, standard errors, numbers of events, incidence rates, probabilities, and the equation-level score.

Menu for predict

Statistics > Postestimation

Syntax for predict

Random-effects (RE) and conditional fixed-effects (FE) overdispersion models

`predict` [*type*] *newvar* [*if*] [*in*] [, *RE/FE_statistic* `nooffset`]

Population-averaged (PA) model

`predict` [*type*] *newvar* [*if*] [*in*] [, *PA_statistic* `nooffset`]

RE/FE_statistic	Description
Main	
`xb`	linear prediction; the default
`stdp`	standard error of the linear prediction
`nu0`	predicted number of events; assumes fixed or random effect is zero
`iru0`	predicted incidence rate; assumes fixed or random effect is zero
`pr0(`*n*`)`	probability $\Pr(y = n)$ assuming the random effect is zero; only allowed after `xtnbreg, re`
`pr0(`*a*`,`*b*`)`	probability $\Pr(a \leq y \leq b)$ assuming the random effect is zero; only allowed after `xtnbreg, re`

PA_statistic	Description
Main	
`mu`	predicted number of events; considers the `offset()`; the default
`rate`	predicted number of events
`xb`	linear prediction
`stdp`	standard error of the linear prediction
`score`	first derivative of the log likelihood with respect to $\mathbf{x}_{it}\boldsymbol{\beta}$

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

Options for predict

Main

`xb` calculates the linear prediction. This is the default for the random-effects and fixed-effects models.

`mu` and `rate` both calculate the predicted number of events. `mu` takes into account the `offset()`, and `rate` ignores those adjustments. `mu` and `rate` are equivalent if you did not specify `offset()`. `mu` is the default for the population-averaged model.

`stdp` calculates the standard error of the linear prediction.

`nu0` calculates the predicted number of events, assuming a zero random or fixed effect.

`iru0` calculates the predicted incidence rate, assuming a zero random or fixed effect.

`pr0(`*n*`)` calculates the probability $\Pr(y = n)$ assuming the random effect is zero, where *n* is a nonnegative integer that may be specified as a number or a variable (only allowed after `xtnbreg, re`).

`pr0(`*a*`,`*b*`)` calculates the probability $\Pr(a \leq y \leq b)$ assuming the random effect is zero, where *a* and *b* are nonnegative integers that may be specified as numbers or variables (only allowed after `xtnbreg, re`);

b missing ($b \geq .$) means $+\infty$;
`pr0(20,.)` calculates $\Pr(y \geq 20)$;
`pr0(20,`*b*`)` calculates $\Pr(y \geq 20)$ in observations for which $b \geq .$ and calculates $\Pr(20 \leq y \leq b)$ elsewhere.

`pr0(.,`*b*`)` produces a syntax error. A missing value in an observation on the variable *a* causes a missing value in that observation for `pr0(`*a*`,`*b*`)`.

`score` calculates the equation-level score, $u_{it} = \partial \ln L(\mathbf{x}_{it}\boldsymbol{\beta})/\partial(\mathbf{x}_{it}\boldsymbol{\beta})$.

`nooffset` is relevant only if you specified `offset(`*varname*`)` for `xtnbreg`. It modifies the calculations made by `predict` so that they ignore the offset variable; the linear prediction is treated as $\mathbf{x}_{it}\boldsymbol{\beta}$ rather than $\mathbf{x}_{it}\boldsymbol{\beta} + \text{offset}_{it}$.

margins

Description for margins

`margins` estimates margins of response for linear predictions, numbers of events, incidence rates, and probabilities.

Menu for margins

Statistics > Postestimation

Syntax for margins

`margins` [*marginlist*] [, *options*]

`margins` [*marginlist*] , `predict(`*statistic* ...`)` [`predict(`*statistic* ...`)` ...] [*options*]

Random-effects (RE) and conditional fixed-effects (FE) overdispersion models

statistic	Description
`xb`	linear prediction; the default
`nu0`	predicted number of events; assumes fixed or random effect is zero
`iru0`	predicted incidence rate; assumes fixed or random effect is zero
`pr0(`n`)`	probability $\Pr(y = n)$ assuming the random effect is zero; only allowed after `xtnbreg, re`
`pr0(`a`,`b`)`	probability $\Pr(a \leq y \leq b)$ assuming the random effect is zero; only allowed after `xtnbreg, re`
`stdp`	not allowed with `margins`

Population-averaged (PA) model

statistic	Description
`mu`	predicted number of events; considers the `offset()`; the default
`rate`	predicted number of events
`xb`	linear prediction
`stdp`	not allowed with `margins`
`score`	not allowed with `margins`

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.

For the full syntax, see [R] **margins**.

Methods and formulas

The probabilities calculated using the `pr0(`n`)` option are the probability $\Pr(y_{it} = n)$ for a RE model assuming the random effect is zero. A negative binomial model is an overdispersed Poisson model, and the nominal overdispersion can be calculated as $\delta = s/(r-1)$, where r and s are as given in the estimation results. Define $\mu_{it} = \exp(\mathbf{x}_{it}\boldsymbol{\beta} + \text{offset}_{it})$. Then the probabilities in `pr0(`n`)` are calculated as the probability that $y_{it} = n$, where y_{it} has a negative binomial distribution with mean $\delta\mu_{it}$ and variance $\delta(1+\delta)\mu_{it}$.

Also see

[XT] **xtnbreg** — Fixed-effects, random-effects, & population-averaged negative binomial models

[U] 20 Estimation and postestimation commands

xtologit — Random-effects ordered logistic model

Description	Quick start	Menu	Syntax
Options	Remarks and examples	Stored results	Methods and formulas
References	Also see		

Description

`xtologit` fits a random-effects ordered logistic model. The actual values taken on by the dependent variable are irrelevant, although larger values are assumed to correspond to "higher" outcomes. The conditional distribution of the dependent variable given the random effects is assumed to be multinomial with success probability determined by the logistic cumulative distribution function.

Quick start

Random-effects ordered logistic model of `y` as a function of `x` using `xtset` data

```
xtologit y x
```

Add indicators for levels of categorical variable `a`

```
xtologit y x i.a
```

Same as above, but report odds ratios

```
xtologit y x i.a, or
```

With cluster–robust standard errors for panels nested within `cvar`

```
xtologit y x i.a, vce(cluster cvar)
```

Menu

Statistics > Longitudinal/panel data > Ordinal outcomes > Logistic regression (RE)

Syntax

`xtologit` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, *options*]

options	Description
Model	
`offset(`*varname*`)`	include *varname* in model with coefficient constrained to 1
`constraints(`*constraints*`)`	apply specified linear constraints
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `robust`, `cluster` *clustvar*, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`or`	report odds ratios
`lrmodel`	perform the likelihood-ratio model test instead of the default Wald test
`nocnsreport`	do not display constraints
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Integration	
`intmethod(`*intmethod*`)`	integration method; *intmethod* may be `mvaghermite` (the default) or `ghermite`
`intpoints(#)`	use # quadrature points; default is `intpoints(12)`
Maximization	
maximize_options	control the maximization process; seldom used
`startgrid(`*numlist*`)`	improve starting value of the random-intercept parameter by performing a grid search
`nodisplay`	suppress display of header and coefficients
`collinear`	keep collinear variables
`coeflegend`	display legend instead of statistics

A panel variable must be specified; see **[XT] xtset**.

indepvars may contain factor variables; see **[U] 11.4.3 Factor variables**.

depvar and *indepvars* may contain time-series operators; see **[U] 11.4.4 Time-series varlists**.

`bayes`, `by`, `collect`, `fp`, and `statsby` are allowed; see **[U] 11.1.10 Prefix commands**. For more details, see **[BAYES] bayes: xtologit**.

`fweight`s, `iweight`s, and `pweight`s are allowed; see **[U] 11.1.6 weight**.

`startgrid()`, `nodisplay`, `collinear`, and `coeflegend` do not appear in the dialog box.

See **[U] 20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

offset(*varname*), constraints(*constraints*); see [R] **Estimation options**.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (oim), that are robust to some kinds of misspecification (robust), that allow for intragroup correlation (cluster *clustvar*), and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] ***vce_options***.

Specifying vce(robust) is equivalent to specifying vce(cluster *panelvar*); see *xtologit and the robust VCE estimator* in *Methods and formulas*.

Reporting

level(#); see [R] **Estimation options**.

or reports the estimated coefficients transformed to odds ratios, that is, e^b rather than b. Standard errors and confidence intervals are similarly transformed. This option affects how results are displayed, not how they are estimated. or may be specified at estimation or when replaying previously estimated results.

lrmodel, nocnsreport; see [R] **Estimation options**.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(*%fmt*), pformat(*%fmt*), sformat(*%fmt*), and nolstretch; see [R] **Estimation options**.

Integration

intmethod(*intmethod*), intpoints(#); see [R] **Estimation options**.

Maximization

maximize_options: difficult, technique(*algorithm_spec*), iterate(#), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), nonrtolerance, and from(*init_specs*); see [R] **Maximize**. These options are seldom used.

The following options are available with xtologit but are not shown in the dialog box:

startgrid(*numlist*) performs a grid search to improve the starting value of the random-intercept parameter. No grid search is performed by default unless the starting value is found to not be feasible; in this case, xtologit runs startgrid(0.1 1 10) and chooses the value that works best. You may already be using a default form of startgrid() without knowing it. If you see xtologit displaying Grid node 1, Grid node 2, ... following Grid node 0 in the iteration log, that is xtologit doing a default search because the original starting value was not feasible.

nodisplay is for programmers. It suppresses the display of the header and the coefficients.

collinear, coeflegend; see [R] **Estimation options**.

Remarks and examples

Remarks are presented under the following headings:

Overview

`xtologit` fits a random-effects ordered logistic model. Ordered logistic models are used to estimate relationships between an ordinal dependent variable and a set of independent variables. An *ordinal* variable is a variable that is categorical and ordered, for instance, "poor", "good", and "excellent", which might indicate a person's current health status or the repair record of a car. If there are only two outcomes, see [XT] **xtlogit**, [XT] **xtprobit**, and [XT] **xtcloglog**. This entry is concerned only with more than two outcomes.

▷ Example 1

We use the data from the "Television, School, and Family Smoking Prevention and Cessation Project" (Flay et al. 1988; Rabe-Hesketh and Skrondal 2022, chap. 11), where schools were randomly assigned into one of four groups defined by two treatment variables. Students within each school are nested in classes, and classes are nested in schools. In this example, we ignore the variability of classes within schools; see example 2 of [ME] **meologit** for a model that incorporates the additional class-level variance component. The dependent variable is the tobacco and health knowledge score (`thk`) collapsed into four ordered categories. We regress the outcome on the treatment variables and their interaction and control for the pretreatment score.

```
. use https://www.stata-press.com/data/r19/tvsfpors
(Television, School, and Family Project)

. xtset school

Panel variable: school (unbalanced)

. xtologit thk prethk cc##tv

Fitting comparison model:

Iteration 0:  Log likelihood =  -2212.775
Iteration 1:  Log likelihood =  -2125.509
Iteration 2:  Log likelihood = -2125.1034
Iteration 3:  Log likelihood = -2125.1032

Refining starting values:

Grid node 0:  Log likelihood = -2136.2426

Fitting full model:

Iteration 0:  Log likelihood = -2136.2426  (not concave)
Iteration 1:  Log likelihood = -2120.2577
Iteration 2:  Log likelihood = -2119.7574
Iteration 3:  Log likelihood = -2119.7428
Iteration 4:  Log likelihood = -2119.7428

Random-effects ordered logistic regression          Number of obs    =  1,600
Group variable: school                              Number of groups =     28

Random effects u_i ~ Gaussian                       Obs per group:
                                                                 min =     18
                                                                 avg =   57.1
                                                                 max =    137

Integration method: mvaghermite                     Integration pts. =     12

                                                    Wald chi2(4)     = 128.06
Log likelihood = -2119.7428                         Prob > chi2      = 0.0000

------------------------------------------------------------------------------
         thk | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      prethk |   .4032892     .03886    10.38   0.000      .327125    .4794534
        1.cc |   .9237904    .204074     4.53   0.000     .5238127    1.323768
        1.tv |   .2749937   .1977424     1.39   0.164    -.1125744    .6625618
             |
       cc#tv |
         1 1 |  -.4659256   .2845963    -1.64   0.102    -1.023724    .0918728
-------------+----------------------------------------------------------------
       /cut1 |  -.0884493   .1641062                     -.4100916     .233193
       /cut2 |   1.153364    .165616                      .8287625    1.477965
       /cut3 |    2.33195   .1734199                      1.992053    2.671846
-------------+----------------------------------------------------------------
    /sigma2_u |   .0735112   .0383106                      .0264695    .2041551
------------------------------------------------------------------------------
LR test vs. ologit model: chibar2(01) = 10.72         Prob >= chibar2 = 0.0005
```

The estimation table reports the parameter estimates, the estimated cutpoints $(\kappa_1, \kappa_2, \kappa_3)$, and the estimated panel-level variance component labeled `sigma2_u`. The parameter estimates can be interpreted just as the output from a standard ordered logistic regression would be interpreted; see [R] **ologit**. For example, we find that students with higher preintervention scores tend to have higher postintervention scores.

Underneath the parameter estimates and the cutpoints, the table shows the estimated variance component. The estimate of σ_u^2 is 0.074 with standard error 0.038. The reported likelihood-ratio test shows that there is enough variability between schools to favor a random-effects ordered logistic regression over a standard ordered logistic regression.

◁

❑ Technical note

The random-effects model is calculated using quadrature, which is an approximation whose accuracy depends partially on the number of integration points used. We can use the `quadchk` command to see if changing the number of integration points affects the results. If the results change, the quadrature approximation is not accurate given the number of integration points. Try increasing the number of integration points using the `intpoints()` option and run `quadchk` again. Do not attempt to interpret the results of estimates when the coefficients reported by `quadchk` differ substantially. See [XT] **quadchk** for details and [XT] **xtprobit** for an example.

Because the `xtologit` likelihood function is calculated by Gauss–Hermite quadrature, on large problems the computations can be slow. Computation time is roughly proportional to the number of points used for the quadrature.

❑

Video example

Ordered logistic and probit for panel data

Stored results

`xtologit` stores the following in `e()`:

Scalars

`e(N)`	number of observations
`e(N_g)`	number of groups
`e(k)`	number of parameters
`e(k_aux)`	number of auxiliary parameters
`e(k_eq)`	number of equations in `e(b)`
`e(k_eq_model)`	number of equations in overall model test
`e(k_dv)`	number of dependent variables
`e(k_cat)`	number of categories
`e(df_m)`	model degrees of freedom
`e(ll)`	log likelihood
`e(ll_0)`	log likelihood, constant-only model
`e(ll_c)`	log likelihood, comparison model
`e(chi2)`	χ^2
`e(chi2_c)`	χ^2 for comparison test
`e(N_clust)`	number of clusters
`e(sigma_u)`	panel-level standard deviation
`e(n_quad)`	number of quadrature points
`e(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(g_max)`	largest group size
`e(p)`	p-value for model test
`e(rank)`	rank of `e(V)`
`e(rank0)`	rank of `e(V)` for constant-only model
`e(ic)`	number of iterations

`e(rc)`	return code
`e(converged)`	`1` if converged, `0` otherwise

Macros

`e(cmd)`	`meglm`
`e(cmd2)`	`xtologit`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(covariates)`	list of covariates
`e(ivar)`	variable denoting groups
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(title)`	title in estimation output
`e(clustvar)`	name of cluster variable
`e(offset)`	linear offset variable
`e(chi2type)`	`Wald` or `LR`; type of model χ^2 test
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. err.
`e(intmethod)`	integration method
`e(distrib)`	`Gaussian`; the distribution of the random effect
`e(opt)`	type of optimization
`e(which)`	`max` or `min`; whether optimizer is to perform maximization or minimization
`e(ml_method)`	type of `ml` method
`e(user)`	name of likelihood-evaluator program
`e(technique)`	maximization technique
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(marginsok)`	predictions allowed by `margins`
`e(marginswtype)`	weight type for `margins`
`e(marginswexp)`	weight expression for `margins`
`e(marginsdefault)`	default `predict()` specification for `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`

Matrices

`e(b)`	coefficient vector
`e(Cns)`	constraints matrix
`e(ilog)`	iteration log
`e(gradient)`	gradient vector
`e(cat)`	category values
`e(V)`	variance–covariance matrix of the estimators
`e(V_modelbased)`	model-based variance

Functions

`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

`xtologit` fits via maximum likelihood the random-effects model

$$\Pr(y_{it} > k|\boldsymbol{\kappa}, \mathbf{x}_{it}, \nu_i) = H(\mathbf{x}_{it}\boldsymbol{\beta} + \nu_i - \kappa_k)$$

for $i = 1, \ldots, n$ panels, where $t = 1, \ldots, n_i$, ν_i are independent and identically distributed $N(0, \sigma_\nu^2)$, and $\boldsymbol{\kappa}$ is a set of cutpoints κ_1, κ_2, ..., κ_{K-1}, where K is the number of possible outcomes; and $H(\cdot)$ is the logistic cumulative distribution function.

From the above, we can derive the probability of observing outcome k for response y_{it} as

$$\begin{aligned} p_{itk} \equiv \Pr(y_{it} = k|\boldsymbol{\kappa}, \mathbf{x}_{it}, \nu_i) &= \Pr(\kappa_{k-1} < \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i + \epsilon_{it} \leq \kappa_k) \\ &= \Pr(\kappa_{k-1} - \mathbf{x}_{it}\boldsymbol{\beta} - \nu_i < \epsilon_{it} \leq \kappa_k - \mathbf{x}_{it}\boldsymbol{\beta} - \nu_i) \\ &= H(\kappa_k - \mathbf{x}_{it}\boldsymbol{\beta} - \nu_i) - H(\kappa_{k-1} - \mathbf{x}_{it}\boldsymbol{\beta} - \nu_i) \\ &= \frac{1}{1 + \exp(-\kappa_k + \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i)} - \frac{1}{1 + \exp(-\kappa_{k-1} + \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i)} \end{aligned}$$

where κ_0 is taken as $-\infty$ and κ_K is taken as $+\infty$. Here $\mathbf{x}_{it}$ does not contain a constant term, because its effect is absorbed into the cutpoints.

We may also express this model in terms of a latent linear response, where observed ordinal responses y_{it} are generated from the latent continuous responses, such that

$$y_{it}^* = \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i + \epsilon_{it}$$

and

$$y_{it} = \begin{cases} 1 & \text{if} \quad y_{it}^* \leq \kappa_1 \\ 2 & \text{if} \quad \kappa_1 < y_{it}^* \leq \kappa_2 \\ \vdots & \\ K & \text{if} \quad \kappa_{K-1} < y_{it}^* \end{cases}$$

The errors ϵ_{it} are distributed as logistic with mean zero and variance $\pi^2/3$ and are independent of ν_i.

Given a set of panel-level random effects ν_i, we can define the conditional distribution for response y_{it} as

$$\begin{aligned} f(y_{it}, \boldsymbol{\kappa}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i) &= \prod_{k=1}^{K} p_{itk}^{I_k(y_{it})} \\ &= \exp \sum_{k=1}^{K} \left\{ I_k(y_{it}) \log(p_{itk}) \right\} \end{aligned}$$

where

$$I_k(y_{it}) = \begin{cases} 1 & \text{if } y_{it} = k \\ 0 & \text{otherwise} \end{cases}$$

For panel i, $i = 1, \ldots, M$, the conditional distribution of $\mathbf{y}_i = (y_{i1}, \ldots, y_{in_i})'$ is

$$\prod_{t=1}^{n_i} f(y_{it}, \boldsymbol{\kappa}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i)$$

and the panel-level likelihood l_i is given by

$$l_i(\boldsymbol{\beta},\boldsymbol{\kappa},\sigma_\nu^2) = \int_{-\infty}^{\infty} \frac{e^{-\nu_i^2/2\sigma_\nu^2}}{\sqrt{2\pi}\sigma_\nu} \left\{ \prod_{t=1}^{n_i} f(y_{it},\boldsymbol{\kappa},\mathbf{x}_{it}\boldsymbol{\beta}+\nu_i) \right\} d\nu_i$$

$$\equiv \int_{-\infty}^{\infty} g(y_{it},\boldsymbol{\kappa},x_{it},\nu_i) d\nu_i$$

This integral can be approximated with M-point Gauss–Hermite quadrature

$$\int_{-\infty}^{\infty} e^{-x^2} h(x) dx \approx \sum_{m=1}^{M} w_m^* h(a_m^*)$$

This is equivalent to

$$\int_{-\infty}^{\infty} f(x) dx \approx \sum_{m=1}^{M} w_m^* \exp\left\{(a_m^*)^2\right\} f(a_m^*)$$

where the w_m^* denote the quadrature weights and the a_m^* denote the quadrature abscissas. The log likelihood, L, is the sum of the logs of the panel-level likelihoods l_i.

The default approximation of the log likelihood is by mean–variance adaptive Gauss–Hermite quadrature, which approximates the panel-level likelihood with

$$l_i \approx \sqrt{2}\widehat{\sigma}_i \sum_{m=1}^{M} w_m^* \exp\left\{(a_m^*)^2\right\} g(y_{it},\boldsymbol{\kappa},x_{it},\sqrt{2}\widehat{\sigma}_i a_m^* + \widehat{\mu}_i)$$

where $\widehat{\sigma}_i$ and $\widehat{\mu}_i$ are the adaptive parameters for panel i. The method of calculating the posterior mean and variance and using those parameters for $\widehat{\mu}_i$ and $\widehat{\sigma}_i$ is described in detail in Naylor and Smith (1982) and Skrondal and Rabe-Hesketh (2004). We start with $\widehat{\sigma}_{i,0} = 1$ and $\widehat{\mu}_{i,0} = 0$, and the posterior means and variances are updated in the jth iteration. That is, at the jth iteration of the optimization for l_i, we use

$$l_{i,j} \approx \sum_{m=1}^{M} \sqrt{2}\widehat{\sigma}_{i,j-1} w_m^* \exp\{(a_m^*)^2\} g(y_{it},\boldsymbol{\kappa},x_{it},\sqrt{2}\widehat{\sigma}_{i,j-1} a_m^* + \widehat{\mu}_{i,j-1})$$

Letting

$$\tau_{i,m,j-1} = \sqrt{2}\widehat{\sigma}_{i,j-1} a_m^* + \widehat{\mu}_{i,j-1}$$

$$\widehat{\mu}_{i,j} = \sum_{m=1}^{M} (\tau_{i,m,j-1}) \frac{\sqrt{2}\widehat{\sigma}_{i,j-1} w_m^* \exp\{(a_m^*)^2\} g(y_{it},\boldsymbol{\kappa},x_{it},\tau_{i,m,j-1})}{l_{i,j}}$$

and

$$\widehat{\sigma}_{i,j} = \sum_{m=1}^{M} (\tau_{i,m,j-1})^2 \frac{\sqrt{2}\widehat{\sigma}_{i,j-1} w_m^* \exp\{(a_m^*)^2\} g(y_{it},\boldsymbol{\kappa},x_{it},\tau_{i,m,j-1})}{l_{i,j}} - \left(\widehat{\mu}_{i,j}\right)^2$$

This is repeated until $\widehat{\mu}_{i,j}$ and $\widehat{\sigma}_{i,j}$ have converged for this iteration of the maximization algorithm. This adaptation is applied on every iteration.

The log likelihood can also be calculated by nonadaptive Gauss–Hermite quadrature with the `intmethod(ghermite)` option, where $\rho = \sigma_\nu^2/(\sigma_\nu^2 + 1)$:

$$L = \sum_{i=1}^{n} w_i \log\Big\{ \Pr(y_{i1}, \ldots, y_{in_i} | \boldsymbol{\kappa}, \mathbf{x}_{i1}, \ldots, \mathbf{x}_{in_i}) \Big\}$$

$$\approx \sum_{i=1}^{n} w_i \log \left[\frac{1}{\sqrt{\pi}} \sum_{m=1}^{M} w_m^* \prod_{t=1}^{n_i} f \left\{ y_{it}, \boldsymbol{\kappa}, \mathbf{x}_{it}\boldsymbol{\beta} + a_m^* \left(\frac{2\rho}{1-\rho} \right)^{1/2} \right\} \right]$$

Both quadrature formulas require that the integrated function be well approximated by a polynomial of degree equal to the number of quadrature points. The number of periods (panel size) can affect whether

$$\prod_{t=1}^{n_i} f(y_{it}, \boldsymbol{\kappa}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i)$$

is well approximated by a polynomial. As panel size and ρ increase, the quadrature approximation can become less accurate. For large ρ, the random-effects model can also become unidentified. Adaptive quadrature gives better results for correlated data and large panels than nonadaptive quadrature; however, we recommend that you use the `quadchk` command (see [XT] **quadchk**) to verify the quadrature approximation used in this command, whichever approximation you choose.

xtologit and the robust VCE estimator

Specifying `vce(robust)` or `vce(cluster` *clustvar*`)` causes the Huber/White/sandwich VCE estimator to be calculated for the coefficients estimated in this regression. See [P] **_robust**, particularly *Introduction* and *Methods and formulas*. Wooldridge (2020) and Arellano (2003) discuss this application of the Huber/White/sandwich VCE estimator. As discussed by Wooldridge (2020), Stock and Watson (2008), and Arellano (2003), specifying `vce(robust)` is equivalent to specifying `vce(cluster` *panelvar*`)`, where *panelvar* is the variable that identifies the panels.

Clustering on the panel variable produces a consistent VCE estimator when the disturbances are not identically distributed over the panels or there is serial correlation in ϵ_{it}.

The cluster–robust VCE estimator requires that there are many clusters and the disturbances are uncorrelated across the clusters. The panel variable must be nested within the cluster variable because of the within-panel correlation that is generally induced by the random-effects transform when there is heteroskedasticity or within-panel serial correlation in the idiosyncratic errors.

References

Allison, P. D. 2009. *Fixed Effects Regression Models*. Newbury Park, CA: Sage.

Arellano, M. 2003. *Panel Data Econometrics*. Oxford: Oxford University Press. https://doi.org/10.1093/0199245282.001.0001.

Baetschmann, G., A. Ballantyne, K. E. Staub, and R. Winkelmann. 2020. feologit: A new command for fitting fixed-effects ordered logit models. *Stata Journal* 20: 253–275.

Conway, M. R. 1990. A random effects model for binary data. *Biometrics* 46: 317–328. https://doi.org/10.2307/2531437.

Flay, B. R., B. R. Brannon, C. A. Johnson, W. B. Hansen, A. L. Ulene, D. A. Whitney-Saltiel, L. R. Gleason, S. Sussman, M. D. Gavin, K. M. Glowacz, D. F. Sobol, and D. C. Spiegel. 1988. The television, school, and family smoking cessation and prevention project: I. Theoretical basis and program development. *Preventive Medicine* 17: 585–607. https://doi.org/10.1016/0091-7435(88)90053-9.

Liang, K.-Y., and S. L. Zeger. 1986. Longitudinal data analysis using generalized linear models. *Biometrika* 73: 13–22. https://doi.org/10.1093/biomet/73.1.13.

Naylor, J. C., and A. F. M. Smith. 1982. Applications of a method for the efficient computation of posterior distributions. *Journal of the Royal Statistical Society,* C ser., 31: 214–225. https://doi.org/10.2307/2347995.

Neuhaus, J. M. 1992. Statistical methods for longitudinal and clustered designs with binary responses. *Statistical Methods in Medical Research* 1: 249–273. https://doi.org/10.1177/096228029200100303.

Neuhaus, J. M., J. D. Kalbfleisch, and W. W. Hauck. 1991. A comparison of cluster-specific and population-averaged approaches for analyzing correlated binary data. *International Statistical Review* 59: 25–35. https://doi.org/10.2307/1403572.

Pendergast, J. F., S. J. Gange, M. A. Newton, M. J. Lindstrom, M. Palta, and M. R. Fisher. 1996. A survey of methods for analyzing clustered binary response data. *International Statistical Review* 64: 89–118. https://doi.org/10.2307/1403425.

Rabe-Hesketh, S., and A. Skrondal. 2022. *Multilevel and Longitudinal Modeling Using Stata*. 4th ed. College Station, TX: Stata Press.

Skrondal, A., and S. Rabe-Hesketh. 2004. *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models*. Boca Raton, FL: Chapman and Hall/CRC.

Stock, J. H., and M. W. Watson. 2008. Heteroskedasticity-robust standard errors for fixed effects panel data regression. *Econometrica* 76: 155–174. https://doi.org/10.1111/j.0012-9682.2008.00821.x.

Twisk, J. W. R. 2013. *Applied Longitudinal Data Analysis for Epidemiology: A Practical Guide*. 2nd ed. Cambridge: Cambridge University Press.

Wooldridge, J. M. 2020. *Introductory Econometrics: A Modern Approach*. 7th ed. Boston: Cengage.

Also see

[XT] **xtologit postestimation** — Postestimation tools for xtologit

[XT] **quadchk** — Check sensitivity of quadrature approximation

[XT] **xtoprobit** — Random-effects ordered probit model

[XT] **xtset** — Declare data to be panel data

[BAYES] **bayes: xtologit** — Bayesian random-effects ordered logistic model

[ME] **meologit** — Multilevel mixed-effects ordered logistic regression

[R] **logistic** — Logistic regression, reporting odds ratios

[R] **logit** — Logistic regression, reporting coefficients

[U] 20 Estimation and postestimation commands

Postestimation commands   predict   margins   Remarks and examples   Also see

Postestimation commands

The following postestimation commands are available after `xtologit`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of parameters
`estat ic`	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC, respectively)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
`lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	linear predictions and their SEs, probabilities
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of parameters
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

predict

Description for predict

`predict` creates a new variable containing predictions such as linear predictions, probabilities, and standard errors.

Menu for predict

Statistics > Postestimation

Syntax for predict

`predict` [*type*] { *stub** | *newvar* | *newvarlist* } [*if*] [*in*] [, *statistic*
    `outcome(`*outcome*`)` `nooffset`]

statistic	Description
Main	
`xb`	linear prediction; the default
`pr`	marginal probability of the specified outcome (`outcome()`)
`pu0`	probability of the specified outcome (`outcome()`) assuming that the random effect is zero
`stdp`	standard error of the linear prediction

If you do not specify `outcome()`, `pr` and `pu0` (with one new variable specified) assume `outcome(#1)`.
You specify one or k new variables with `pr` and `pu0`, where k is the number of outcomes.
You specify one new variable with `xb` and `stdp`.
These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

Options for predict

Main

`xb`, the default, calculates the linear prediction.

`pr` calculates predicted probabilities that are marginal with respect to the random effect, which means that the probabilities are calculated by integrating the prediction function with respect to the random effect over its entire support. Unless otherwise specified, `pr` defaults to the first outcome.

`pu0` calculates predicted probabilities, assuming that the random effect for that observation's panel is zero ($\nu_i = 0$). Unless otherwise specified, `pu0` defaults to the first outcome.

`stdp` calculates the standard error of the linear prediction.

`outcome(`*outcome*`)` specifies the outcome for which the predicted probabilities are to be calculated. `outcome()` should contain either one value of the dependent variable or one of `#1`, `#2`, ..., with `#1` meaning the first category of the dependent variable, `#2` meaning the second category, etc.

`nooffset` is relevant only if you specified `offset(`*varname*`)` for `xtologit`. This option modifies the calculations made by `predict` so that they ignore the offset variable; the linear prediction is treated as $\mathbf{x}_{it}\boldsymbol{\beta}$ rather than $\mathbf{x}_{it}\boldsymbol{\beta} + \text{offset}_{it}$.

margins

Description for margins

`margins` estimates margins of response for linear predictions and probabilities.

Menu for margins

Statistics > Postestimation

Syntax for margins

`margins` [*marginlist*] [, *options*]

`margins` [*marginlist*] , `predict(`*statistic* ...`)` [`predict(`*statistic* ...`)` ...] [*options*]

statistic	Description
default	marginal probability for each outcome
`pr`	marginal probability of the specified outcome (`outcome()`)
`pu0`	probability of the specified outcome (`outcome()`) assuming that the random effect is zero
`xb`	linear prediction
`stdp`	not allowed with `margins`

`pr` and `pu0` default to the first outcome.

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.

For the full syntax, see [R] **margins**.

Remarks and examples

▷ Example 1: Predicted marginal probabilities

In example 1 of [XT] **xtologit**, we modeled the tobacco and health knowledge score (`thk`)—coded 1, 2, 3, 4—among students as a function of two treatments (`cc` and `tv`) using a random-effects ordered logistic model. Here we refit the model, obtain the predicted probabilities for all 4 outcomes, and list the first 10 observations.

```
. use https://www.stata-press.com/data/r19/tvsfpors
(Television, School, and Family Project)

. xtset school

Panel variable: school (unbalanced)

. xtologit thk prethk cc##tv
  (output omitted)

. predict pr*, pr
(using 12 quadrature points)

. list thk pr1-pr4 in 1/10
```

	thk	pr1	pr2	pr3	pr4
1.	3	.1427312	.2191935	.281714	.3563613
2.	4	.0695318	.1343722	.2459221	.5501739
3.	3	.0695318	.1343722	.2459221	.5501739
4.	4	.1003876	.1756773	.2724781	.4514571
5.	4	.1003876	.1756773	.2724781	.4514571
6.	3	.0695318	.1343722	.2459221	.5501739
7.	2	.1427312	.2191935	.281714	.3563613
8.	4	.0695318	.1343722	.2459221	.5501739
9.	4	.0476276	.0990209	.2082012	.6451503
10.	4	.1003876	.1756773	.2724781	.4514571

For each observation, our best guess for the predicted outcome is the one with the highest predicted probability. For example, for the very first observation in the table above, we would choose outcome 4 as the most likely to occur.

◁

Also see

[XT] **xtologit** — Random-effects ordered logistic model

[U] 20 Estimation and postestimation commands

xtoprobit — Random-effects ordered probit model

Description

xtoprobit fits a random-effects ordered probit model. The actual values taken on by the dependent variable are irrelevant, although larger values are assumed to correspond to “higher” outcomes. The conditional distribution of the dependent variable given the random effects is assumed to be multinomial, with success probability determined by the standard normal cumulative distribution function.

Quick start

Random-effects ordered probit model of y as a function of x using xtset data

```
xtoprobit y x
```

Add indicators for levels of categorical variable a

```
xtoprobit y x i.a
```

With cluster–robust standard errors for panels nested within cvar

```
xtoprobit y x i.a, vce(cluster cvar)
```

Menu

Statistics > Longitudinal/panel data > Ordinal outcomes > Probit regression (RE)

Syntax

`xtoprobit` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, *options*]

options	Description
Model	
offset(*varname*)	include *varname* in model with coefficient constrained to 1
constraints(*constraints*)	apply specified linear constraints
SE/Robust	
vce(*vcetype*)	*vcetype* may be oim, robust, cluster *clustvar*, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
lrmodel	perform the likelihood-ratio model test instead of the default Wald test
nocnsreport	do not display constraints
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Integration	
intmethod(*intmethod*)	integration method; *intmethod* may be mvaghermite (the default) or ghermite
intpoints(#)	use # quadrature points; default is intpoints(12)
Maximization	
maximize_options	control the maximization process; seldom used
startgrid(*numlist*)	improve starting value of the random-intercept parameter by performing a grid search
nodisplay	suppress display of header and coefficients
collinear	keep collinear variables
coeflegend	display legend instead of statistics

A panel variable must be specified; see [XT] **xtset**.

indepvars may contain factor variables; see **[U] 11.4.3 Factor variables**.

depvar and *indepvars* may contain time-series operators; see **[U] 11.4.4 Time-series varlists**.

bayes, by, collect, fp, and statsby are allowed; see **[U] 11.1.10 Prefix commands**. For more details, see [BAYES] **bayes: xtoprobit**.

fweights, iweights, and pweights are allowed; see **[U] 11.1.6 weight**.

startgrid(), nodisplay, collinear, and coeflegend do not appear in the dialog box.

See **[U] 20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

offset(*varname*), constraints(*constraints*); see [R] **Estimation options**.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (oim), that are robust to some kinds of misspecification (robust), that allow for intragroup correlation (cluster *clustvar*), and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] ***vce_options***.

Specifying vce(robust) is equivalent to specifying vce(cluster *panelvar*); see *xtoprobit and the robust VCE estimator* in *Methods and formulas*.

Reporting

level(#); see [R] **Estimation options**.

lrmodel, nocnsreport; see [R] **Estimation options**.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Integration

intmethod(*intmethod*), intpoints(#); see [R] **Estimation options**.

Maximization

maximize_options: difficult, technique(*algorithm_spec*), iterate(#), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), nonrtolerance, and from(*init_specs*); see [R] **Maximize**. These options are seldom used.

The following options are available with xtoprobit but are not shown in the dialog box:

startgrid(*numlist*) performs a grid search to improve the starting value of the random-intercept parameter. No grid search is performed by default unless the starting value is found to not be feasible; in this case, xtoprobit runs startgrid(0.1 1 10) and chooses the value that works best. You may already be using a default form of startgrid() without knowing it. If you see xtoprobit displaying Grid node 1, Grid node 2, ... following Grid node 0 in the iteration log, that is xtoprobit doing a default search because the original starting value was not feasible.

nodisplay is for programmers. It suppresses the display of the header and the coefficients.

collinear, coeflegend; see [R] **Estimation options**.

Remarks and examples

Remarks are presented under the following headings:

Overview

`xtoprobit` fits a random-effects ordered probit model. Ordered probit models are used to estimate relationships between an ordinal dependent variable and a set of independent variables. An *ordinal* variable is a variable that is categorical and ordered, for instance, "poor", "good", and "excellent", which might indicate a person's current health status or the repair record of a car. If there are only two outcomes, see [XT] **xtprobit**, [XT] **xtlogit**, and [XT] **xtcloglog**. This entry is concerned only with more than two outcomes.

▷ Example 1

We use the data from the "Television, School, and Family Smoking Prevention and Cessation Project" (Flay et al. 1988; Rabe-Hesketh and Skrondal 2022, chap. 11), where schools were randomly assigned into one of four groups defined by two treatment variables. Students within each school are nested in classes, and classes are nested in schools. In this example, we ignore the variability of classes within schools; see example 2 of [ME] **meoprobit** for a model that incorporates the additional class-level variance component. The dependent variable is the tobacco and health knowledge score (`thk`) collapsed into four ordered categories. We regress the outcome on the treatment variables and their interaction and control for the pretreatment score.

```
. use https://www.stata-press.com/data/r19/tvsfpors
(Television, School, and Family Project)

. xtset school

Panel variable: school (unbalanced)

. xtoprobit thk prethk cc##tv

Fitting comparison model:

Iteration 0:  Log likelihood =  -2212.775
Iteration 1:  Log likelihood = -2127.8111
Iteration 2:  Log likelihood = -2127.7612
Iteration 3:  Log likelihood = -2127.7612

Refining starting values:

Grid node 0:  Log likelihood = -2149.7302

Fitting full model:

Iteration 0:  Log likelihood = -2149.7302  (not concave)
Iteration 1:  Log likelihood = -2129.6838  (not concave)
Iteration 2:  Log likelihood = -2123.5143
Iteration 3:  Log likelihood = -2122.2896
Iteration 4:  Log likelihood = -2121.7949
Iteration 5:  Log likelihood = -2121.7716
Iteration 6:  Log likelihood = -2121.7715

Random-effects ordered probit regression          Number of obs     =  1,600
Group variable: school                            Number of groups  =     28

Random effects u_i ~ Gaussian                     Obs per group:
                                                               min =     18
                                                               avg =   57.1
                                                               max =    137

Integration method: mvaghermite                   Integration pts.  =     12

                                                  Wald chi2(4)      = 128.05
Log likelihood = -2121.7715                       Prob > chi2       = 0.0000

------------------------------------------------------------------------------
         thk | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      prethk |   .2369804   .0227739    10.41   0.000     .1923444    .2816164
        1.cc |   .5490957   .1255108     4.37   0.000      .303099    .7950923
        1.tv |   .1695405   .1215889     1.39   0.163    -.0687693    .4078504
             |
       cc#tv |
        1 1  |  -.2951837   .1751969    -1.68   0.092    -.6385634    .0481959
-------------+----------------------------------------------------------------
       /cut1 |  -.0682011   .1003374                     -.2648587    .1284565
       /cut2 |     .67681   .1008836                      .4790817    .8745382
       /cut3 |   1.390649   .1037494                      1.187304    1.593995
-------------+----------------------------------------------------------------
   /sigma2_u |   .0288527   .0146201                      .0106874    .0778937
------------------------------------------------------------------------------
LR test vs. oprobit model: chibar2(01) = 11.98        Prob >= chibar2 = 0.0003
```

The estimation table reports the parameter estimates, the estimated cutpoints $(\kappa_1, \kappa_2, \kappa_3)$, and the estimated panel-level variance component labeled `sigma2_u`. The parameter estimates can be interpreted just as the output from a standard ordered probit regression would be interpreted; see [R] **oprobit**. For example, we find that students with higher preintervention scores tend to have higher postintervention scores.

Underneath the parameter estimates and the cutpoints, the table shows the estimated variance component. The estimate of σ_u^2 is 0.029 with standard error 0.015. The reported likelihood-ratio test shows that there is enough variability between schools to favor a random-effects ordered probit regression over a standard ordered probit regression.

◁

❑ Technical note

The random-effects model is calculated using quadrature, which is an approximation whose accuracy depends partially on the number of integration points used. We can use the `quadchk` command to see if changing the number of integration points affects the results. If the results change, the quadrature approximation is not accurate given the number of integration points. Try increasing the number of integration points using the `intpoints()` option and run `quadchk` again. Do not attempt to interpret the results of estimates when the coefficients reported by `quadchk` differ substantially. See [XT] **quadchk** for details and [XT] **xtprobit** for an example.

Because the `xtoprobit` likelihood function is calculated by Gauss–Hermite quadrature, on large problems the computations can be slow. Computation time is roughly proportional to the number of points used for the quadrature.

❑

Video example

Ordered logistic and probit for panel data

Stored results

`xtoprobit` stores the following in `e()`:

Scalars

`e(N)`	number of observations
`e(N_g)`	number of groups
`e(k)`	number of parameters
`e(k_aux)`	number of auxiliary parameters
`e(k_eq)`	number of equations in `e(b)`
`e(k_eq_model)`	number of equations in overall model test
`e(k_dv)`	number of dependent variables
`e(k_cat)`	number of categories
`e(df_m)`	model degrees of freedom
`e(ll)`	log likelihood
`e(ll_0)`	log likelihood, constant-only model
`e(ll_c)`	log likelihood, comparison model
`e(chi2)`	χ^2
`e(chi2_c)`	χ^2 for comparison test
`e(N_clust)`	number of clusters
`e(sigma_u)`	panel-level standard deviation
`e(n_quad)`	number of quadrature points
`e(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(g_max)`	largest group size
`e(p)`	p-value for model test
`e(rank)`	rank of `e(V)`
`e(rank0)`	rank of `e(V)` for constant-only model
`e(ic)`	number of iterations

`e(rc)`	return code
`e(converged)`	`1` if converged, `0` otherwise

Macros

`e(cmd)`	`meglm`
`e(cmd2)`	`xtoprobit`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(covariates)`	list of covariates
`e(ivar)`	variable denoting groups
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(title)`	title in estimation output
`e(clustvar)`	name of cluster variable
`e(offset)`	linear offset variable
`e(chi2type)`	`Wald` or `LR`; type of model χ^2 test
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. err.
`e(intmethod)`	integration method
`e(distrib)`	`Gaussian`; the distribution of the random effect
`e(opt)`	type of optimization
`e(which)`	`max` or `min`; whether optimizer is to perform maximization or minimization
`e(ml_method)`	type of `ml` method
`e(user)`	name of likelihood-evaluator program
`e(technique)`	maximization technique
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(marginsok)`	predictions allowed by `margins`
`e(marginswtype)`	weight type for `margins`
`e(marginswexp)`	weight expression for `margins`
`e(marginsdefault)`	default `predict()` specification for `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`

Matrices

`e(b)`	coefficient vector
`e(Cns)`	constraints matrix
`e(ilog)`	iteration log
`e(gradient)`	gradient vector
`e(cat)`	category values
`e(V)`	variance–covariance matrix of the estimators
`e(V_modelbased)`	model-based variance

Functions

`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

`xtoprobit` fits via maximum likelihood the random-effects model

$$\Pr(y_{it} > k|\boldsymbol{\kappa}, \mathbf{x}_{it}, \nu_i) = \Phi(\mathbf{x}_{it}\boldsymbol{\beta} + \nu_i - \kappa_k)$$

for $i = 1, \ldots, n$ panels, where $t = 1, \ldots, n_i$, ν_i are independent and identically distributed $N(0, \sigma^2_\nu)$, and $\boldsymbol{\kappa}$ is a set of cutpoints κ_1, κ_2, …, κ_{K-1}, where K is the number of possible outcomes; and $\Phi(\cdot)$ is the standard normal cumulative distribution function.

From the above, we can derive the probability of observing outcome k for response y_{it} as

$$\begin{aligned} p_{itk} \equiv \Pr(y_{it} = k|\boldsymbol{\kappa}, \mathbf{x}_{it}, \nu_i) &= \Pr(\kappa_{k-1} < \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i + \epsilon_{it} \le \kappa_k) \\ &= \Pr(\kappa_{k-1} - \mathbf{x}_{it}\boldsymbol{\beta} - \nu_i < \epsilon_{it} \le \kappa_k - \mathbf{x}_{it}\boldsymbol{\beta} - \nu_i) \\ &= \Phi(\kappa_k - \mathbf{x}_{it}\boldsymbol{\beta} - \nu_i) - \Phi(\kappa_{k-1} - \mathbf{x}_{it}\boldsymbol{\beta} - \nu_i) \end{aligned}$$

where κ_0 is taken as $-\infty$, and κ_K is taken as $+\infty$. Here $\mathbf{x}_{it}$ does not contain a constant term, because its effect is absorbed into the cutpoints.

We may also express this model in terms of a latent linear response, where observed ordinal responses y_{it} are generated from the latent continuous responses, such that

$$y^*_{it} = \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i + \epsilon_{it}$$

and

$$y_{it} = \begin{cases} 1 & \text{if} \quad y^*_{it} \le \kappa_1 \\ 2 & \text{if} \quad \kappa_1 < y^*_{it} \le \kappa_2 \\ \vdots & \\ K & \text{if} \quad \kappa_{K-1} < y^*_{it} \end{cases}$$

The errors ϵ_{it} are distributed as standard normal with mean zero and variance one and are independent of ν_i.

Given a set of panel-level random effects ν_i, we can define the conditional distribution for response y_{it} as

$$\begin{aligned} f(y_{it}, \boldsymbol{\kappa}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i) &= \prod_{k=1}^{K} p_{itk}^{I_k(y_{it})} \\ &= \exp \sum_{k=1}^{K} \left\{ I_k(y_{it}) \log(p_{itk}) \right\} \end{aligned}$$

where

$$I_k(y_{it}) = \begin{cases} 1 & \text{if } y_{it} = k \\ 0 & \text{otherwise} \end{cases}$$

For panel i, $i = 1, \ldots, M$, the conditional distribution of $\mathbf{y}_i = (y_{i1}, \ldots, y_{in_i})'$ is

$$\prod_{t=1}^{n_i} f(y_{it}, \boldsymbol{\kappa}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i)$$

and the panel-level likelihood l_i is given by

$$l_i(\boldsymbol{\beta}, \boldsymbol{\kappa}, \sigma_\nu^2) = \int_{-\infty}^{\infty} \frac{e^{-\nu_i^2/2\sigma_\nu^2}}{\sqrt{2\pi}\sigma_\nu} \left\{ \prod_{t=1}^{n_i} f(y_{it}, \boldsymbol{\kappa}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i) \right\} d\nu_i$$

$$\equiv \int_{-\infty}^{\infty} g(y_{it}, \boldsymbol{\kappa}, x_{it}, \nu_i) d\nu_i$$

This integral can be approximated with M-point Gauss–Hermite quadrature

$$\int_{-\infty}^{\infty} e^{-x^2} h(x) dx \approx \sum_{m=1}^{M} w_m^* h(a_m^*)$$

This is equivalent to

$$\int_{-\infty}^{\infty} f(x) dx \approx \sum_{m=1}^{M} w_m^* \exp\left\{(a_m^*)^2\right\} f(a_m^*)$$

where the w_m^* denote the quadrature weights and the a_m^* denote the quadrature abscissas. The log likelihood, L, is the sum of the logs of the panel-level likelihoods l_i.

The default approximation of the log likelihood is by mean–variance adaptive Gauss–Hermite quadrature, which approximates the panel-level likelihood with

$$l_i \approx \sqrt{2}\widehat{\sigma}_i \sum_{m=1}^{M} w_m^* \exp\left\{(a_m^*)^2\right\} g(y_{it}, \boldsymbol{\kappa}, x_{it}, \sqrt{2}\widehat{\sigma}_i a_m^* + \widehat{\mu}_i)$$

where $\widehat{\sigma}_i$ and $\widehat{\mu}_i$ are the adaptive parameters for panel i. The method of calculating the posterior mean and variance and using those parameters for $\widehat{\mu}_i$ and $\widehat{\sigma}_i$ is described in detail in Naylor and Smith (1982) and Skrondal and Rabe-Hesketh (2004). We start with $\widehat{\sigma}_{i,0} = 1$ and $\widehat{\mu}_{i,0} = 0$, and the posterior means and variances are updated in the jth iteration. That is, at the jth iteration of the optimization for l_i, we use

$$l_{i,j} \approx \sum_{m=1}^{M} \sqrt{2}\widehat{\sigma}_{i,j-1} w_m^* \exp\{(a_m^*)^2\} g(y_{it}, \boldsymbol{\kappa}, x_{it}, \sqrt{2}\widehat{\sigma}_{i,j-1} a_m^* + \widehat{\mu}_{i,j-1})$$

Letting

$$\tau_{i,m,j-1} = \sqrt{2}\widehat{\sigma}_{i,j-1} a_m^* + \widehat{\mu}_{i,j-1}$$

$$\widehat{\mu}_{i,j} = \sum_{m=1}^{M} (\tau_{i,m,j-1}) \frac{\sqrt{2}\widehat{\sigma}_{i,j-1} w_m^* \exp\{(a_m^*)^2\} g(y_{it}, \boldsymbol{\kappa}, x_{it}, \tau_{i,m,j-1})}{l_{i,j}}$$

and

$$\widehat{\sigma}_{i,j} = \sum_{m=1}^{M} (\tau_{i,m,j-1})^2 \frac{\sqrt{2}\widehat{\sigma}_{i,j-1} w_m^* \exp\{(a_m^*)^2\} g(y_{it}, \boldsymbol{\kappa}, x_{it}, \tau_{i,m,j-1})}{l_{i,j}} - \left(\widehat{\mu}_{i,j}\right)^2$$

This is repeated until $\widehat{\mu}_{i,j}$ and $\widehat{\sigma}_{i,j}$ have converged for this iteration of the maximization algorithm. This adaptation is applied on every iteration.

The log likelihood can also be calculated by nonadaptive Gauss–Hermite quadrature with the `intmethod(ghermite)` option, where $\rho = \sigma_\nu^2/(\sigma_\nu^2+1)$:

$$\begin{aligned} L &= \sum_{i=1}^{n} w_i \log\Big\{ \Pr(y_{i1},\dots,y_{in_i}|\boldsymbol{\kappa},\mathbf{x}_{i1},\dots,\mathbf{x}_{in_i}) \Big\} \\ &\approx \sum_{i=1}^{n} w_i \log\left[\frac{1}{\sqrt{\pi}} \sum_{m=1}^{M} w_m^* \prod_{t=1}^{n_i} f\left\{ y_{it}, \boldsymbol{\kappa}, \mathbf{x}_{it}\boldsymbol{\beta} + a_m^* \left(\frac{2\rho}{1-\rho}\right)^{1/2} \right\}\right] \end{aligned}$$

Both quadrature formulas require that the integrated function be well approximated by a polynomial of degree equal to the number of quadrature points. The number of periods (panel size) can affect whether

$$\prod_{t=1}^{n_i} f(y_{it}, \boldsymbol{\kappa}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i)$$

is well approximated by a polynomial. As panel size and ρ increase, the quadrature approximation can become less accurate. For large ρ, the random-effects model can also become unidentified. Adaptive quadrature gives better results for correlated data and large panels than nonadaptive quadrature; however, we recommend that you use the `quadchk` command (see [XT] **quadchk**) to verify the quadrature approximation used in this command, whichever approximation you choose.

xtoprobit and the robust VCE estimator

Specifying `vce(robust)` or `vce(cluster` *clustvar*`)` causes the Huber/White/sandwich VCE estimator to be calculated for the coefficients estimated in this regression. See [P] **_robust**, particularly *Introduction* and *Methods and formulas*. Wooldridge (2020) and Arellano (2003) discuss this application of the Huber/White/sandwich VCE estimator. As discussed by Wooldridge (2020), Stock and Watson (2008), and Arellano (2003), specifying `vce(robust)` is equivalent to specifying `vce(cluster` *panelvar*`)`, where *panelvar* is the variable that identifies the panels.

Clustering on the panel variable produces a consistent VCE estimator when the disturbances are not identically distributed over the panels or there is serial correlation in ϵ_{it}.

The cluster–robust VCE estimator requires that there are many clusters and the disturbances are uncorrelated across the clusters. The panel variable must be nested within the cluster variable because of the within-panel correlation that is generally induced by the random-effects transform when there is heteroskedasticity or within-panel serial correlation in the idiosyncratic errors.

References

Allison, P. D. 2009. *Fixed Effects Regression Models*. Newbury Park, CA: Sage.

Arellano, M. 2003. *Panel Data Econometrics*. Oxford: Oxford University Press. https://doi.org/10.1093/0199245282.001.0001.

Conway, M. R. 1990. A random effects model for binary data. *Biometrics* 46: 317–328. https://doi.org/10.2307/2531437.

Flay, B. R., B. R. Brannon, C. A. Johnson, W. B. Hansen, A. L. Ulene, D. A. Whitney-Saltiel, L. R. Gleason, S. Sussman, M. D. Gavin, K. M. Glowacz, D. F. Sobol, and D. C. Spiegel. 1988. The television, school, and family smoking cessation and prevention project: I. Theoretical basis and program development. *Preventive Medicine* 17: 585–607. https://doi.org/10.1016/0091-7435(88)90053-9.

Liang, K.-Y., and S. L. Zeger. 1986. Longitudinal data analysis using generalized linear models. *Biometrika* 73: 13–22. https://doi.org/10.1093/biomet/73.1.13.

Naylor, J. C., and A. F. M. Smith. 1982. Applications of a method for the efficient computation of posterior distributions. *Journal of the Royal Statistical Society*, C ser., 31: 214–225. https://doi.org/10.2307/2347995.

Neuhaus, J. M. 1992. Statistical methods for longitudinal and clustered designs with binary responses. *Statistical Methods in Medical Research* 1: 249–273. https://doi.org/10.1177/096228029200100303.

Neuhaus, J. M., J. D. Kalbfleisch, and W. W. Hauck. 1991. A comparison of cluster-specific and population-averaged approaches for analyzing correlated binary data. *International Statistical Review* 59: 25–35. https://doi.org/10.2307/1403572.

Pendergast, J. F., S. J. Gange, M. A. Newton, M. J. Lindstrom, M. Palta, and M. R. Fisher. 1996. A survey of methods for analyzing clustered binary response data. *International Statistical Review* 64: 89–118. https://doi.org/10.2307/1403425.

Rabe-Hesketh, S., and A. Skrondal. 2022. *Multilevel and Longitudinal Modeling Using Stata*. 4th ed. College Station, TX: Stata Press.

Skrondal, A., and S. Rabe-Hesketh. 2004. *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models*. Boca Raton, FL: Chapman and Hall/CRC.

Stock, J. H., and M. W. Watson. 2008. Heteroskedasticity-robust standard errors for fixed effects panel data regression. *Econometrica* 76: 155–174. https://doi.org/10.1111/j.0012-9682.2008.00821.x.

Twisk, J. W. R. 2013. *Applied Longitudinal Data Analysis for Epidemiology: A Practical Guide*. 2nd ed. Cambridge: Cambridge University Press.

Wooldridge, J. M. 2020. *Introductory Econometrics: A Modern Approach*. 7th ed. Boston: Cengage.

Also see

[XT] **xtoprobit postestimation** — Postestimation tools for xtoprobit

[XT] **quadchk** — Check sensitivity of quadrature approximation

[XT] **xteoprobit** — Extended random-effects ordered probit regression

[XT] **xtologit** — Random-effects ordered logistic model

[XT] **xtset** — Declare data to be panel data

[BAYES] **bayes: xtoprobit** — Bayesian random-effects ordered probit model

[ME] **meoprobit** — Multilevel mixed-effects ordered probit regression

[R] **probit** — Probit regression

[U] 20 Estimation and postestimation commands

Postestimation commands

The following postestimation commands are available after `xtoprobit`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of parameters
`estat ic`	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC, respectively)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
`lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	linear predictions and their SEs, probabilities
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of parameters
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

predict

Description for predict

`predict` creates a new variable containing predictions such as linear predictions, probabilities, and standard errors.

Menu for predict

Statistics > Postestimation

Syntax for predict

`predict` [*type*] {*stub**|*newvar*|*newvarlist*} [*if*] [*in*] [, *statistic*
    `outcome(`*outcome*`)` `nooffset`]

statistic	Description
Main	
`xb`	linear prediction; the default
`pr`	marginal probability of the specified outcome (`outcome()`)
`pu0`	probability of the specified outcome (`outcome()`) assuming that the random effect is zero
`stdp`	standard error of the linear prediction

If you do not specify `outcome()`, `pr` and `pu0` (with one new variable specified) assume `outcome(#1)`.
You specify one or k new variables with `pr` and `pu0`, where k is the number of outcomes.
You specify one new variable with `xb` and `stdp`.
These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

Options for predict

Main

`xb`, the default, calculates the linear prediction.

`pr` calculates predicted probabilities that are marginal with respect to the random effect, which means that the probabilities are calculated by integrating the prediction function with respect to the random effect over its entire support. Unless otherwise specified, `pr` defaults to the first outcome.

`pu0` calculates predicted probabilities, assuming that the random effect for that observation's panel is zero ($\nu_i = 0$). Unless otherwise specified, `pu0` defaults to the first outcome.

`stdp` calculates the standard error of the linear prediction.

`outcome(`*outcome*`)` specifies the outcome for which the predicted probabilities are to be calculated. `outcome()` should contain either one value of the dependent variable or one of `#1`, `#2`, ..., with `#1` meaning the first category of the dependent variable, `#2` meaning the second category, etc.

`nooffset` is relevant only if you specified `offset(`*varname*`)` for `xtoprobit`. This option modifies the calculations made by `predict` so that they ignore the offset variable; the linear prediction is treated as $\mathbf{x}_{it}\boldsymbol{\beta}$ rather than $\mathbf{x}_{it}\boldsymbol{\beta} + \text{offset}_{it}$.

margins

Description for margins

margins estimates margins of response for linear predictions and probabilities.

Menu for margins

Statistics > Postestimation

Syntax for margins

margins [*marginlist*] [, *options*]

margins [*marginlist*] , predict(*statistic* ...) [predict(*statistic* ...) ...] [*options*]

statistic	Description
default	marginal probability for each outcome
pr	marginal probability of the specified outcome (outcome())
pu0	probability of the specified outcome (outcome()) assuming that the random effect is zero
xb	linear prediction
stdp	not allowed with margins

pr and pu0 default to the first outcome.

Statistics not allowed with margins are functions of stochastic quantities other than e(b).

For the full syntax, see [R] **margins**.

Remarks and examples

▷ Example 1: Predicted marginal probabilities

In example 1 of [XT] **xtoprobit**, we modeled the tobacco and health knowledge score (thk)—coded 1, 2, 3, 4—among students as a function of two treatments (cc and tv) using a random-effects ordered probit model. Here we refit the model, obtain the predicted probabilities for all 4 outcomes, and list the first 10 observations.

```
. use https://www.stata-press.com/data/r19/tvsfpors
(Television, School, and Family Project)

. xtset school

Panel variable: school (unbalanced)

. xtoprobit thk prethk cc##tv
  (output omitted)

. predict pr*, pr
(using 12 quadrature points)

. list thk pr1-pr4 in 1/10
```

	thk	pr1	pr2	pr3	pr4
1.	3	.1409978	.2254197	.2750568	.3585258
2.	4	.0614014	.1479641	.2488757	.5417588
3.	3	.0614014	.1479641	.2488757	.5417588
4.	4	.0951857	.187463	.2685807	.4487706
5.	4	.0951857	.187463	.2685807	.4487706
6.	3	.0614014	.1479641	.2488757	.5417588
7.	2	.1409978	.2254197	.2750568	.3585258
8.	4	.0614014	.1479641	.2488757	.5417588
9.	4	.0378048	.1108411	.2188475	.6325067
10.	4	.0951857	.187463	.2685807	.4487706

For each observation, our best guess for the predicted outcome is the one with the highest predicted probability. For example, for the very first observation in the table above, we would choose outcome 4 as the most likely to occur.

◁

Also see

[XT] **xtoprobit** — Random-effects ordered probit model

[U] 20 Estimation and postestimation commands

xtpcse — Linear regression with panel-corrected standard errors

Description | Quick start | Menu | Syntax
Options | Remarks and examples | Stored results | Methods and formulas
Acknowledgments | References | Also see

Description

`xtpcse` calculates panel-corrected standard error (PCSE) estimates for linear cross-sectional time-series models where the parameters are estimated by either OLS or Prais–Winsten regression. When computing the standard errors and the variance–covariance estimates, `xtpcse` assumes that the disturbances are, by default, heteroskedastic and contemporaneously correlated across panels.

See [XT] **xtgls** for the generalized least-squares estimator for these models.

Quick start

Linear regression of `y` on `x1` and `x2` with panel-corrected standard errors and assuming no within-panel autocorrelation using `xtset` data

```
xtpcse y x1 x2
```

Same as above, but specify a common first-order autocorrelation within panels

```
xtpcse y x1 x2, correlation(ar1)
```

Within-panel heteroskedastic errors but no contemporaneous correlation between panels

```
xtpcse y x1 x2, hetonly
```

Let autocorrelation structure be panel-specific estimated by time-series methods

```
xtpcse y x1 x2, correlation(psar1) rhotype(tscorr)
```

Menu

Statistics > Longitudinal/panel data > Contemporaneous correlation > Regression with panel-corrected standard errors (PCSE)

Syntax

xtpcse *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, *options*]

options	Description
Model	
noconstant	suppress constant term
correlation(independent)	use independent autocorrelation structure
correlation(ar1)	use AR1 autocorrelation structure
correlation(psar1)	use panel-specific AR1 autocorrelation structure
rhotype(*calc*)	specify method to compute autocorrelation parameter; seldom used
np1	weight panel-specific autocorrelations by panel sizes
hetonly	assume panel-level heteroskedastic errors
independent	assume independent errors across panels
by/if/in	
casewise	include only observations with complete cases
pairwise	include all available observations with nonmissing pairs
SE	
nmk	normalize standard errors by $N - k$ instead of N
Reporting	
level(#)	set confidence level; default is level(95)
detail	report list of gaps in time series
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
coeflegend	display legend instead of statistics

A panel variable and a time variable must be specified; use xtset; see [XT] **xtset**.

indepvars may contain factor variables; see [U] **11.4.3 Factor variables**.

depvar and *indepvars* may contain time-series operators; see [U] **11.4.4 Time-series varlists**.

by, collect, and statsby are allowed; see [U] **11.1.10 Prefix commands**.

iweights and aweights are allowed; see [U] **11.1.6 weight**.

coeflegend does not appear in the dialog box.

See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

noconstant; see [R] **Estimation options**.

correlation(*corr*) specifies the form of assumed autocorrelation within panels.

correlation(independent), the default, specifies that there is no autocorrelation.

correlation(ar1) specifies that, within panels, there is first-order autocorrelation AR(1) and that the coefficient of the AR(1) process is common to all the panels.

correlation(psar1) specifies that, within panels, there is first-order autocorrelation and that the coefficient of the AR(1) process is specific to each panel. psar1 stands for panel-specific AR(1).

rhotype(*calc*) specifies the method to be used to calculate the autocorrelation parameter. Allowed strings for *calc* are

regress	regression using lags; the default
freg	regression using leads
tscorr	time-series autocorrelation calculation
dw	Durbin–Watson calculation

All the above methods are consistent and asymptotically equivalent; this is a rarely used option.

np1 specifies that the panel-specific autocorrelations be weighted by T_i rather than by the default $T_i - 1$ when estimating a common ρ for all panels, where T_i is the number of observations in panel i. This option has an effect only when panels are unbalanced and the correlation(ar1) option is specified.

hetonly and independent specify alternative forms for the assumed covariance of the disturbances across the panels. If neither is specified, the disturbances are assumed to be heteroskedastic (each panel has its own variance) and contemporaneously correlated across the panels (each pair of panels has its own covariance). This is the standard PCSE model.

hetonly specifies that the disturbances are assumed to be panel-level heteroskedastic only with no contemporaneous correlation across panels.

independent specifies that the disturbances are assumed to be independent across panels; that is, there is one disturbance variance common to all observations.

by/if/in

casewise and pairwise specify how missing observations in unbalanced panels are to be treated when estimating the interpanel covariance matrix of the disturbances. The default is casewise selection.

casewise specifies that the entire covariance matrix be computed only on the observations (periods) that are available for all panels. If an observation has missing data, all observations of that period are excluded when estimating the covariance matrix of disturbances. Specifying casewise ensures that the estimated covariance matrix will be of full rank and will be positive definite.

pairwise specifies that, for each element in the covariance matrix, all available observations (periods) that are common to the two panels contributing to the covariance be used to compute the covariance.

The casewise and pairwise options have an effect only when the panels are unbalanced and neither hetonly nor independent is specified.

SE

nmk specifies that standard errors be normalized by $N-k$, where k is the number of parameters estimated, rather than N, the number of observations. Different authors have used one or the other normalization. Greene (2018, 313) remarks that whether a degree-of-freedom correction improves the small-sample properties is an open question.

Reporting

level(#); see [R] **Estimation options**.

detail specifies that a detailed list of any gaps in the series be reported.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

The following option is available with xtpcse but is not shown in the dialog box:

coeflegend; see [R] **Estimation options**.

Remarks and examples

xtpcse is an alternative to feasible generalized least squares (FGLS)—see [XT] **xtgls**—for fitting linear cross-sectional time-series models when the disturbances are not assumed to be independent and identically distributed (i.i.d.). Instead, the disturbances are assumed to be either heteroskedastic across panels or heteroskedastic and contemporaneously correlated across panels. The disturbances may also be assumed to be autocorrelated within panel, and the autocorrelation parameter may be constant across panels or different for each panel.

We can write such models as

$$y_{it} = \mathbf{x}_{it}\boldsymbol{\beta} + \epsilon_{it}$$

where $i = 1, \ldots, m$ is the number of units (or panels); $t = 1, \ldots, T_i$; T_i is the number of periods in panel i; and ϵ_{it} is a disturbance that may be autocorrelated along t or contemporaneously correlated across i.

This model can also be written panel by panel as

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_m \end{bmatrix} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_m \end{bmatrix} \boldsymbol{\beta} + \begin{bmatrix} \boldsymbol{\epsilon}_1 \\ \boldsymbol{\epsilon}_2 \\ \vdots \\ \boldsymbol{\epsilon}_m \end{bmatrix}$$

For a model with heteroskedastic disturbances and contemporaneous correlation but with no autocorrelation, the disturbance covariance matrix is assumed to be

$$E[\boldsymbol{\epsilon}\boldsymbol{\epsilon}'] = \boldsymbol{\Omega} = \begin{bmatrix} \sigma_{11}\mathbf{I}_{11} & \sigma_{12}\mathbf{I}_{12} & \cdots & \sigma_{1m}\mathbf{I}_{1m} \\ \sigma_{21}\mathbf{I}_{21} & \sigma_{22}\mathbf{I}_{22} & \cdots & \sigma_{2m}\mathbf{I}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{m1}\mathbf{I}_{m1} & \sigma_{m2}\mathbf{I}_{m2} & \cdots & \sigma_{mm}\mathbf{I}_{mm} \end{bmatrix}$$

where σ_{ii} is the variance of the disturbances for panel i, σ_{ij} is the covariance of the disturbances between panel i and panel j when the panels' periods are matched, and $\mathbf{I}$ is a T_i by T_i identity matrix with balanced panels. The panels need not be balanced for xtpcse, but the expression for the covariance of the disturbances will be more general if they are unbalanced.

This could also be written as

$$E[\boldsymbol{\epsilon}\boldsymbol{\epsilon}'] = \boldsymbol{\Sigma}_{m\times m} \otimes \mathbf{I}_{T_i \times T_i}$$

where $\boldsymbol{\Sigma}$ is the panel-by-panel covariance matrix and $\mathbf{I}$ is an identity matrix.

See [XT] **xtgls** for a full taxonomy and description of possible disturbance covariance structures.

xtpcse and xtgls follow two different estimation schemes for this family of models. xtpcse produces OLS estimates of the parameters when no autocorrelation is specified, or Prais–Winsten (see [TS] **prais**) estimates when autocorrelation is specified. If autocorrelation is specified, the estimates of the parameters are conditional on the estimates of the autocorrelation parameter(s). The estimate of the variance–covariance matrix of the parameters is asymptotically efficient under the assumed covariance structure of the disturbances and uses the FGLS estimate of the disturbance covariance matrix; see Kmenta (1997, 121).

xtgls produces full FGLS parameter and variance–covariance estimates. These estimates are conditional on the estimates of the disturbance covariance matrix and are conditional on any autocorrelation parameters that are estimated; see Kmenta (1997), Greene (2018), Davidson and MacKinnon (1993), or Judge et al. (1985).

Both estimators are consistent, as long as the conditional mean ($\mathbf{x}_{it}\boldsymbol{\beta}$) is correctly specified. If the assumed covariance structure is correct, FGLS estimates produced by xtgls are more efficient. Beck and Katz (1995) have shown, however, that the full FGLS variance–covariance estimates are typically unacceptably optimistic (anticonservative) when used with the type of data analyzed by most social scientists—10–20 panels with 10–40 periods per panel. They show that the OLS or Prais–Winsten estimates with PCSEs have coverage probabilities that are closer to nominal.

Because the covariance matrix elements, σ_{ij}, are estimated from panels i and j, using those observations that have common time periods, estimators for this model achieve their asymptotic behavior as the T_is approach infinity. In contrast, the random- and fixed-effects estimators assume a different model and are asymptotic in the number of panels m; see [XT] **xtreg** for details of the random- and fixed-effects estimators.

Although xtpcse allows other disturbance covariance structures, the term PCSE, as used in the literature, refers specifically to models that are both heteroskedastic and contemporaneously correlated across panels, with or without autocorrelation.

▷ Example 1: Controlling for heteroskedasticity and cross-panel correlation

Grunfeld and Griliches (1960) analyzed a company's current-year gross investment (invest) as determined by the company's prior year market value (mvalue) and the prior year's value of the company's plant and equipment (kstock). The dataset includes 10 companies over 20 years, from 1935 through 1954, and is a classic dataset for demonstrating cross-sectional time-series analysis. Greene (2012, 1112) reproduces the dataset.

To use xtpcse, the data must be organized in "long form"; that is, each observation must represent a record for a specific company at a specific time; see [D] **reshape**. In the Grunfeld data, company is a categorical variable identifying the company, and year is a variable recording the year. Here are the first few records:

```
. use https://www.stata-press.com/data/r19/grunfeld
. list in 1/5
```

	company	year	invest	mvalue	kstock	time
1.	1	1935	317.6	3078.5	2.8	1
2.	1	1936	391.8	4661.7	52.6	2
3.	1	1937	410.6	5387.1	156.9	3
4.	1	1938	257.7	2792.2	209.2	4
5.	1	1939	330.8	4313.2	203.4	5

To compute PCSEs, Stata must be able to identify the panel to which each observation belongs and be able to match the periods across the panels. We tell Stata how to do this matching by specifying the panel and time variables with `xtset`; see [XT] **xtset**. Because the data are annual, we specify the `yearly` option.

```
. xtset company year, yearly

Panel variable: company (strongly balanced)
 Time variable: year, 1935 to 1954
         Delta: 1 year
```

We can obtain OLS parameter estimates for a linear model of `invest` on `mvalue` and `kstock` while allowing the standard errors (and variance–covariance matrix of the estimates) to be consistent when the disturbances from each observation are not independent. Specifically, we want the standard errors to be robust to each company having a different variance of the disturbances and to each company's observations being correlated with those of the other companies through time.

This model is fit in Stata by typing

```
. xtpcse invest mvalue kstock

Linear regression, correlated panels corrected standard errors (PCSEs)

Group variable:   company                       Number of obs     =        200
Time variable:    year                          Number of groups  =         10
Panels:           correlated (balanced)         Obs per group:
Autocorrelation:  no autocorrelation                          min =         20
                                                              avg =         20
                                                              max =         20
Estimated covariances      =        55          R-squared         =     0.8124
Estimated autocorrelations =         0          Wald chi2(2)      =     637.41
Estimated coefficients     =         3          Prob > chi2       =     0.0000

------------------------------------------------------------------------------
             |           Panel-corrected
      invest | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      mvalue |   .1155622   .0072124    16.02   0.000      .101426    .1296983
      kstock |   .2306785   .0278862     8.27   0.000     .1760225    .2853345
       _cons |  -42.71437   6.780965    -6.30   0.000    -56.00482   -29.42392
------------------------------------------------------------------------------
```

◁

▷ Example 2: Comparing the FGLS and PCSE approaches

`xtgls` will produce more efficient FGLS estimates of the models' parameters, but with the disadvantage that the standard error estimates are conditional on the estimated disturbance covariance. Beck and Katz (1995) argue that the improvement in power using FGLS with such data is small and that the standard error estimates from FGLS are unacceptably optimistic (anticonservative).

The FGLS model is fit by typing

```
. xtgls invest mvalue kstock, panels(correlated)

Cross-sectional time-series FGLS regression

Coefficients:  generalized least squares
Panels:        heteroskedastic with cross-sectional correlation
Correlation:   no autocorrelation

Estimated covariances      =        55          Number of obs     =        200
Estimated autocorrelations =         0          Number of groups  =         10
Estimated coefficients     =         3          Time periods      =         20
                                                Wald chi2(2)      =    3738.07
                                                Prob > chi2       =     0.0000

------------------------------------------------------------------------------
      invest | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      mvalue |   .1127515   .0022364    50.42   0.000     .1083683    .1171347
      kstock |   .2231176   .0057363    38.90   0.000     .2118746    .2343605
       _cons |  -39.84382   1.717563   -23.20   0.000    -43.21018   -36.47746
------------------------------------------------------------------------------
```

The coefficients between the two models are close; the constants differ substantially, but we are generally not interested in the constant. As Beck and Katz observed, the standard errors for the FGLS model are 50%–100% smaller than those for the OLS model with PCSE.

If we were also concerned about autocorrelation of the disturbances, we could obtain a model with a common AR(1) parameter by specifying `correlation(ar1)`.

```
. xtpcse invest mvalue kstock, correlation(ar1)
note: estimates of rho outside [-1,1] bounded to be in the range [-1,1].

Prais-Winsten regression, correlated panels corrected standard errors (PCSEs)

Group variable:   company                       Number of obs     =        200
Time variable:    year                          Number of groups  =         10
Panels:           correlated (balanced)         Obs per group:
Autocorrelation:  common AR(1)                                min =         20
                                                              avg =         20
                                                              max =         20

Estimated covariances      =        55          R-squared         =     0.5468
Estimated autocorrelations =         1          Wald chi2(2)      =      93.71
Estimated coefficients     =         3          Prob > chi2       =     0.0000

------------------------------------------------------------------------------
             |           Panel-corrected
      invest | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      mvalue |   .0950157   .0129934     7.31   0.000     .0695492    .1204822
      kstock |    .306005   .0603718     5.07   0.000     .1876784    .4243317
       _cons |  -39.12569   30.50355    -1.28   0.200    -98.91154    20.66016
-------------+----------------------------------------------------------------
         rho |   .9059774
------------------------------------------------------------------------------
```

The estimate of the autocorrelation parameter is high (0.906), and the standard errors are larger than for the model without autocorrelation, which is to be expected if there is autocorrelation.

◁

▷ Example 3: Controlling for cross-panel correlation and autocorrelation

Let's estimate panel-specific autocorrelation parameters and change the method of estimating the autocorrelation parameter to the one typically used to estimate autocorrelation in time-series analysis.

```
. xtpcse invest mvalue kstock, correlation(psar1) rhotype(tscorr)
Prais-Winsten regression, correlated panels corrected standard errors (PCSEs)

Group variable:   company                      Number of obs     =        200
Time variable:    year                         Number of groups  =         10
Panels:           correlated (balanced)        Obs per group:
Autocorrelation:  panel-specific AR(1)                       min =         20
                                                             avg =         20
                                                             max =         20
Estimated covariances      =         55        R-squared         =     0.8670
Estimated autocorrelations =         10        Wald chi2(2)      =     444.53
Estimated coefficients     =          3        Prob > chi2       =     0.0000

------------------------------------------------------------------------------
             |           Panel-corrected
      invest | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      mvalue |   .1052613   .0086018    12.24   0.000     .0884021    .1221205
      kstock |   .3386743   .0367568     9.21   0.000     .2666322    .4107163
       _cons |  -58.18714   12.63687    -4.60   0.000    -82.95496   -33.41933
-------------+----------------------------------------------------------------
        rhos =   .5135627     .87017   .9023497     .63368  .8571502 ...  .8752707
------------------------------------------------------------------------------
```

Beck and Katz (1995, 121) make a case against estimating panel-specific AR parameters, as opposed to one AR parameter for all panels.

◁

▷ Example 4: Controlling for heteroskedasticity only; not quite PCSEs

We can also diverge from PCSEs to estimate standard errors that are panel corrected, but only for panel-level heteroskedasticity; that is, each company has a different variance of the disturbances. Allowing also for autocorrelation, we would type

```
. xtpcse invest mvalue kstock, correlation(ar1) hetonly
note: estimates of rho outside [-1,1] bounded to be in the range [-1,1].
Prais-Winsten regression, heteroskedastic panels corrected standard errors

Group variable:   company                      Number of obs     =        200
Time variable:    year                         Number of groups  =         10
Panels:           heteroskedastic (balanced)   Obs per group:
Autocorrelation:  common AR(1)                               min =         20
                                                             avg =         20
                                                             max =         20
Estimated covariances      =         10        R-squared         =     0.5468
Estimated autocorrelations =          1        Wald chi2(2)      =      91.72
Estimated coefficients     =          3        Prob > chi2       =     0.0000

------------------------------------------------------------------------------
             |            Het-corrected
      invest | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      mvalue |   .0950157   .0130872     7.26   0.000     .0693653    .1206661
      kstock |    .306005    .061432     4.98   0.000     .1856006    .4264095
       _cons |  -39.12569   26.16935    -1.50   0.135    -90.41666    12.16529
-------------+----------------------------------------------------------------
         rho |   .9059774
------------------------------------------------------------------------------
```

With this specification, we do not obtain what are referred to in the literature as PCSEs. These standard errors are in the same spirit as PCSEs but are from the asymptotic covariance estimates of OLS without allowing for contemporaneous correlation.

◁

Stored results

xtpcse stores the following in e():

Scalars	
e(N)	number of observations
e(N_g)	number of groups
e(N_gaps)	number of gaps
e(n_cf)	number of estimated coefficients
e(n_cv)	number of estimated covariances
e(n_cr)	number of estimated correlations
e(n_sigma)	observations used to estimate elements of Sigma
e(mss)	model sum of squares
e(df)	degrees of freedom
e(df_m)	model degrees of freedom
e(rss)	residual sum of squares
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(r2)	R^2
e(chi2)	χ^2
e(p)	p-value for model test
e(rmse)	root mean squared error
e(rank)	rank of e(V)
e(rc)	return code
Macros	
e(cmd)	xtpcse
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(ivar)	variable denoting groups
e(tvar)	variable denoting time within groups
e(wtype)	weight type
e(wexp)	weight expression
e(title)	title in estimation output
e(panels)	contemporaneous covariance structure
e(corr)	correlation structure
e(rhotype)	type of estimated correlation
e(rho)	ρ
e(cons)	noconstant or ""
e(missmeth)	casewise or pairwise
e(balance)	balanced or unbalanced
e(chi2type)	Wald; type of model χ^2 test
e(vcetype)	title used to label Std. err.
e(properties)	b V
e(predict)	program used to implement predict
e(marginsok)	predictions allowed by margins
e(marginsnotok)	predictions disallowed by margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved
Matrices	
e(b)	coefficient vector
e(Sigma)	$\widehat{\mathbf{\Sigma}}$ matrix

`e(rhomat)`	vector of autocorrelation parameter estimates
`e(V)`	variance–covariance matrix of the estimators
Functions	
`e(sample)`	marks estimation sample

Methods and formulas

If no autocorrelation is specified, the parameters $\boldsymbol{\beta}$ are estimated by OLS; see [R] **regress**. If autocorrelation is specified, the parameters $\boldsymbol{\beta}$ are estimated by Prais–Winsten; see [TS] **prais**.

When autocorrelation with panel-specific coefficients of correlation is specified (by using the `correlation(psar1)` option), each panel-level ρ_i is computed from the residuals of an OLS regression across all panels; see [TS] **prais**. When autocorrelation with a common coefficient of correlation is specified (by using the `correlation(ar1)` option), the common correlation coefficient is computed as

$$\rho = \frac{\rho_1 + \rho_2 + \cdots + \rho_m}{m}$$

where ρ_i is the estimated autocorrelation coefficient for panel i and m is the number of panels.

The covariance of the OLS or Prais–Winsten coefficients is

$$\text{Var}(\boldsymbol{\beta}) = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\Omega}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}$$

where $\boldsymbol{\Omega}$ is the full covariance matrix of the disturbances.

When the panels are balanced, we can write $\boldsymbol{\Omega}$ as

$$\boldsymbol{\Omega} = \boldsymbol{\Sigma}_{m\times m} \otimes \mathbf{I}_{T_i\times T_i}$$

where $\boldsymbol{\Sigma}$ is the m by m panel-by-panel covariance matrix of the disturbances; see *Remarks and examples*.

`xtpcse` estimates the elements of $\boldsymbol{\Sigma}$ as

$$\widehat{\boldsymbol{\Sigma}}_{ij} = \frac{{\epsilon_i}'\epsilon_j}{T_{ij}}$$

where ϵ_i and ϵ_j are the residuals for panels i and j, respectively, that can be matched by period, and where T_{ij} is the number of residuals between the panels i and j that can be matched by time period.

When the panels are balanced (each panel has the same number of observations and all periods are common to all panels), $T_{ij} = T$, where T is the number of observations per panel.

When panels are unbalanced, `xtpcse` by default uses `casewise` selection, in which only those residuals from periods that are common to all panels are used to compute $\widehat{S}_{ij}$. Here $T_{ij} = T^*$, where T^* is the number of periods common to all panels. When `pairwise` is specified, each $\widehat{S}_{ij}$ is computed using all observations that can be matched by period between the panels i and j.

Acknowledgments

We thank the following people for helpful comments: Nathaniel Beck of the Department of Politics at New York University, Jonathan Katz of the Division of the Humanities and Social Science at California Institute of Technology, and Robert John Franzese Jr. of the Center for Political Studies at the Institute for Social Research at the University of Michigan.

References

Beck, N. L., and J. N. Katz. 1995. What to do (and not to do) with time-series cross-section data. *American Political Science Review* 89: 634–647. https://doi.org/10.2307/2082979.

Blackwell, J. L., III. 2005. Estimation and testing of fixed-effect panel-data systems. *Stata Journal* 5: 202–207.

Davidson, R., and J. G. MacKinnon. 1993. *Estimation and Inference in Econometrics*. New York: Oxford University Press.

Greene, W. H. 2012. *Econometric Analysis*. 7th ed. Upper Saddle River, NJ: Prentice Hall.

———. 2018. *Econometric Analysis*. 8th ed. New York: Pearson.

Grunfeld, Y., and Z. Griliches. 1960. Is aggregation necessarily bad? *Review of Economics and Statistics* 42: 1–13. https://doi.org/10.2307/1926089.

Hoechle, D. 2007. Robust standard errors for panel regressions with cross-sectional dependence. *Stata Journal* 7: 281–312.

Judge, G. G., W. E. Griffiths, R. C. Hill, H. Lütkepohl, and T.-C. Lee. 1985. *The Theory and Practice of Econometrics*. 2nd ed. New York: Wiley.

Kmenta, J. 1997. *Elements of Econometrics*. 2nd ed. Ann Arbor: University of Michigan Press. https://doi.org/10.3998/mpub.15701.

Also see

[XT] **xtpcse postestimation** — Postestimation tools for xtpcse

[XT] **xtgls** — GLS linear model with heteroskedastic and correlated errors

[XT] **xtreg** — Linear models for panel data

[XT] **xtregar** — Fixed- and random-effects linear models with an AR(1) disturbance

[XT] **xtset** — Declare data to be panel data

[R] **regress** — Linear regression

[TS] **newey** — Regression with Newey–West standard errors

[TS] **prais** — Prais–Winsten and Cochrane–Orcutt regression

[U] 20 Estimation and postestimation commands

Postestimation commands   predict   margins   Also see

Postestimation commands

The following postestimation commands are available after `xtpcse`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of parameters
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
`forecast`	dynamic forecasts and simulations
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	linear predictions and their SEs
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of parameters
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

predict

Description for predict

predict creates a new variable containing predictions such as linear predictions and standard errors.

Menu for predict

Statistics > Postestimation

Syntax for predict

predict [*type*] *newvar* [*if*] [*in*] [, xb stdp]

These statistics are available both in and out of sample; type predict ... if e(sample) ... if wanted only for the estimation sample.

Options for predict

Main

xb, the default, calculates the linear prediction.

stdp calculates the standard error of the linear prediction.

margins

Description for margins

margins estimates margins of response for linear predictions.

Menu for margins

Statistics > Postestimation

Syntax for margins

margins [*marginlist*] [, *options*]

margins [*marginlist*] , predict(*statistic* ...) [*options*]

statistic	Description
xb	linear prediction, the default
stdp	not allowed with margins

Statistics not allowed with margins are functions of stochastic quantities other than e(b).

For the full syntax, see [R] **margins**.

Also see

[XT] **xtpcse** — Linear regression with panel-corrected standard errors

[U] 20 Estimation and postestimation commands

xtpoisson — Fixed-effects, random-effects, and population-averaged Poisson models

Description

xtpoisson fits random-effects, conditional fixed-effects, and population-averaged Poisson models. These models are typically used for a nonnegative count dependent variable.

Quick start

Random-effects Poisson regression of y on x and indicators for levels of categorical variable a using xtset data

```
xtpoisson y x i.a
```

Conditional fixed-effects model with exposure variable evar

```
xtpoisson y x i.a, fe exposure(evar)
```

Population-averaged model with robust standard errors

```
xtpoisson y x i.a, pa vce(robust)
```

Same as above, but report incidence-rate ratios

```
xtpoisson y x i.a, pa vce(robust) irr
```

Menu

Statistics > Longitudinal/panel data > Count outcomes > Poisson regression (FE, RE, PA)

Syntax

Random-effects (RE) model

xtpoisson *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, re *RE_options*]

Conditional fixed-effects (FE) model

xtpoisson *depvar* [*indepvars*] [*if*] [*in*] [*weight*] , fe [*FE_options*]

Population-averaged (PA) model

xtpoisson *depvar* [*indepvars*] [*if*] [*in*] [*weight*] , pa [*PA_options*]

RE_options	Description
Model	
noconstant	suppress constant term
re	use random-effects estimator; the default
exposure(*varname*)	include ln(*varname*) in model with coefficient constrained to 1
offset(*varname*)	include *varname* in model with coefficient constrained to 1
normal	use a normal distribution for random effects instead of gamma
constraints(*constraints*)	apply specified linear constraints
SE/Robust	
vce(*vcetype*)	*vcetype* may be oim, robust, cluster *clustvar*, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
irr	report incidence-rate ratios
lrmodel	perform the likelihood-ratio model test instead of the default Wald test
nocnsreport	do not display constraints
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Integration	
intmethod(*intmethod*)	integration method; *intmethod* may be mvaghermite (the default) or ghermite
intpoints(#)	use # quadrature points; default is intpoints(12)
Maximization	
maximize_options	control the maximization process; seldom used
collinear	keep collinear variables
coeflegend	display legend instead of statistics

FE_options	Description
Model	
`fe`	use fixed-effects estimator
`exposure(`*varname*`)`	include ln(*varname*) in model with coefficient constrained to 1
`offset(`*varname*`)`	include *varname* in model with coefficient constrained to 1
`constraints(`*constraints*`)`	apply specified linear constraints
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `robust`, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`irr`	report incidence-rate ratios
`nocnsreport`	do not display constraints
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
maximize_options	control the maximization process; seldom used
`collinear`	keep collinear variables
`coeflegend`	display legend instead of statistics

PA_options	Description
Model	
noconstant	suppress constant term
pa	use population-averaged estimator
exposure(*varname*)	include ln(*varname*) in model with coefficient constrained to 1
offset(*varname*)	include *varname* in model with coefficient constrained to 1
Correlation	
corr(*correlation*)	within-panel correlation structure
force	estimate even if observations unequally spaced in time
SE/Robust	
vce(*vcetype*)	*vcetype* may be conventional, robust, bootstrap, or jackknife
nmp	use divisor $N - P$ instead of the default N
scale(*parm*)	overrides the default scale parameter; *parm* may be x2, dev, phi, or #
Reporting	
level(#)	set confidence level; default is level(95)
irr	report incidence-rate ratios
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Optimization	
optimize_options	control the optimization process; seldom used
coeflegend	display legend instead of statistics

correlation	Description
exchangeable	exchangeable
independent	independent
unstructured	unstructured
fixed *matname*	user-specified
ar #	autoregressive of order #
stationary #	stationary of order #
nonstationary #	nonstationary of order #

A panel variable must be specified. For xtpoisson, pa, correlation structures other than exchangeable and independent require that a time variable also be specified. Use xtset; see [XT] **xtset**.

indepvars may contain factor variables; see **[U] 11.4.3 Factor variables**.

depvar and *indepvars* may contain time-series operators; see **[U] 11.4.4 Time-series varlists**.

by, collect, mi estimate, and statsby are allowed; see **[U] 11.1.10 Prefix commands**. bayes is allowed for the random-effects model. For more details, see [BAYES] **bayes: xtpoisson**. fp is allowed for the random-effects and fixed-effects models.

vce(bootstrap) and vce(jackknife) are not allowed with the mi estimate prefix; see [MI] **mi estimate**.

iweights, fweights, and pweights are allowed for the population-averaged model, and iweights are allowed for the random-effects and fixed-effects models; see **[U] 11.1.6 weight**. Weights must be constant within panel.

collinear and coeflegend do not appear in the dialog box.

See **[U] 20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options for RE model

Model

noconstant; see [R] **Estimation options**.

re, the default, requests the random-effects estimator.

exposure(*varname*), offset(*varname*); see [R] **Estimation options**.

normal specifies that the random effects follow a normal distribution instead of a gamma distribution.

constraints(*constraints*); see [R] **Estimation options**.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (oim), that are robust to some kinds of misspecification (robust), that allow for intragroup correlation (cluster *clustvar*), and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] ***vce_options***.

Specifying vce(robust) is equivalent to specifying vce(cluster *panelvar*); see xtpoisson, re and the robust VCE estimator in *Methods and formulas*.

Reporting

level(#); see [R] **Estimation options**.

irr reports exponentiated coefficients e^b rather than coefficients b. For the Poisson model, exponentiated coefficients are interpreted as incidence-rate ratios.

lrmodel, nocnsreport; see [R] **Estimation options**.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Integration

intmethod(*intmethod*), intpoints(#); see [R] **Estimation options**. normal must also be specified.

Maximization

maximize_options: difficult, technique(*algorithm_spec*), iterate(#), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), nonrtolerance, and from(*init_specs*); see [R] **Maximize**. These options are seldom used.

The following options are available with xtpoisson but are not shown in the dialog box:

collinear, coeflegend; see [R] **Estimation options**.

Options for FE model

Model

fe requests the fixed-effects estimator.

exposure(*varname*), offset(*varname*), constraints(*constraints*); see [R] **Estimation options**.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (oim), that are robust to some kinds of misspecification (robust), and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] ***vce_options***.

vce(robust) invokes a cluster–robust estimate of the VCE in which the ID variable specifies the clusters.

Reporting

level(#); see [R] **Estimation options**.

irr reports exponentiated coefficients e^b rather than coefficients b. For the Poisson model, exponentiated coefficients are interpreted as incidence-rate ratios.

nocnsreport; see [R] **Estimation options**.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Maximization

maximize_options: difficult, technique(*algorithm_spec*), iterate(#), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), nonrtolerance, and from(*init_specs*); see [R] **Maximize**. These options are seldom used.

The following options are available with xtpoisson but are not shown in the dialog box:

collinear, coeflegend; see [R] **Estimation options**.

Options for PA model

Model

noconstant; see [R] **Estimation options**.

pa requests the population-averaged estimator.

exposure(*varname*), offset(*varname*); see [R] **Estimation options**.

Correlation

corr(*correlation*) specifies the within-panel correlation structure; the default corresponds to the equal-correlation model, corr(exchangeable).

When you specify a correlation structure that requires a lag, you indicate the lag after the structure's name with or without a blank; for example, corr(ar 1) or corr(ar1).

If you specify the fixed correlation structure, you specify the name of the matrix containing the assumed correlations following the word fixed, for example, corr(fixed myr).

force specifies that estimation be forced even though the time variable is not equally spaced. This is relevant only for correlation structures that require knowledge of the time variable. These correlation structures require that observations be equally spaced so that calculations based on lags correspond to a constant time change. If you specify a time variable indicating that observations are not equally spaced, the (time dependent) model will not be fit. If you also specify `force`, the model will be fit, and it will be assumed that the lags based on the data ordered by the time variable are appropriate.

SE/Robust

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`conventional`), that are robust to some kinds of misspecification (`robust`), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [XT] ***vce_options***.

`vce(conventional)`, the default, uses the conventionally derived variance estimator for generalized least-squares regression.

`nmp`, `scale(x2 | dev | phi | #)`; see [XT] ***vce_options***.

Reporting

`level(#)`; see [R] **Estimation options**.

`irr` reports exponentiated coefficients e^b rather than coefficients b. For the Poisson model, exponentiated coefficients are interpreted as incidence-rate ratios.

display_options: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(`*style*`)`, `cformat(%`*fmt*`)`, `pformat(%`*fmt*`)`, `sformat(%`*fmt*`)`, and `nolstretch`; see [R] **Estimation options**.

Optimization

optimize_options control the iterative optimization process. These options are seldom used.

`iterate(#)` specifies the maximum number of iterations. When the number of iterations equals #, the optimization stops and presents the current results, even if convergence has not been reached. The default is `iterate(100)`.

`tolerance(#)` specifies the tolerance for the coefficient vector. When the relative change in the coefficient vector from one iteration to the next is less than or equal to #, the optimization process is stopped. `tolerance(1e-6)` is the default.

`log` and `nolog` specify whether to display the iteration log. The iteration log is displayed by default unless you used `set iterlog off` to suppress it; see `set iterlog` in [R] ***set iter***.

`trace` specifies that the current estimates be printed at each iteration.

The following option is available with `xtpoisson` but is not shown in the dialog box:

`coeflegend`; see [R] **Estimation options**.

Remarks and examples

`xtpoisson` fits random-effects, conditional fixed-effects, and population-averaged Poisson models. Whenever we refer to a fixed-effects model, we mean the conditional fixed-effects model. These models are typically used for a nonnegative count dependent variable but may be used for any dependent variable in natural logs. For more information about the assumptions of the Poisson model, see [R] **poisson**.

By default, the population-averaged model is an equal-correlation model; `xtpoisson, pa` assumes `corr(exchangeable)`. Thus, `xtpoisson` is a convenience command for fitting the population-averaged model using `xtgee`; see [XT] **xtgee**. Typing

```
. xtpoisson ..., ... pa exposure(time)
```

is equivalent to typing

```
. xtgee ..., ... family(poisson) link(log) corr(exchangeable) exposure(time)
```

Also see [XT] **xtgee** for information about `xtpoisson`.

By default or when `re` is specified, `xtpoisson` fits via maximum likelihood the random-effects model

$$\Pr(Y_{it} = y_{it}|\mathbf{x}_{it}) = F(y_{it}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i)$$

for $i = 1, \dots, n$ panels, where $t = 1, \dots, n_i$, and $F(x, z) = \Pr(X = x)$, where X is Poisson distributed with mean $\exp(z)$. In the standard random-effects model, ν_i is assumed to be i.i.d. such that $\exp(\nu_i)$ is gamma with mean one and variance α, which is estimated from the data. If `normal` is specified, ν_i is assumed to be i.i.d. $N(0, \sigma_\nu^2)$.

▷ Example 1

We have data on the number of ship accidents for five different types of ships (McCullagh and Nelder 1989, 205). We wish to analyze whether the "incident" rate is affected by the period in which the ship was constructed and operated. Our measure of exposure is months of service for the ship, and in this model, we assume that the exponentiated random effects are distributed as gamma with mean one and variance α.

```
. use https://www.stata-press.com/data/r19/ships
. xtpoisson accident op_75_79 co_65_69 co_70_74 co_75_79, exp(service) irr
Fitting Poisson model:
Iteration 0:  Log likelihood = -147.37993
Iteration 1:  Log likelihood = -80.372714
Iteration 2:  Log likelihood = -80.116093
Iteration 3:  Log likelihood = -80.115916
Iteration 4:  Log likelihood = -80.115916
Fitting full model:
Iteration 0:  Log likelihood = -79.653186
Iteration 1:  Log likelihood = -76.990836  (not concave)
Iteration 2:  Log likelihood = -74.824942
Iteration 3:  Log likelihood = -74.811243
Iteration 4:  Log likelihood = -74.811217
Iteration 5:  Log likelihood = -74.811217
Random-effects Poisson regression                 Number of obs    =     34
Group variable: ship                              Number of groups =      5
Random effects u_i ~ Gamma                        Obs per group:
                                                               min =      6
                                                               avg =    6.8
                                                               max =      7
                                                  Wald chi2(4)     =  50.90
Log likelihood = -74.811217                       Prob > chi2      = 0.0000

------------------------------------------------------------------------------
    accident |        IRR   Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
    op_75_79 |   1.466305   .1734005     3.24   0.001     1.162957    1.848777
    co_65_69 |   2.032543    .304083     4.74   0.000     1.515982     2.72512
    co_70_74 |   2.356853   .3999259     5.05   0.000     1.690033    3.286774
    co_75_79 |   1.641913   .3811398     2.14   0.033      1.04174     2.58786
       _cons |   .0013724   .0002992   -30.24   0.000     .0008952     .002104
ln(service) |          1  (exposure)
-------------+----------------------------------------------------------------
    /lnalpha |  -2.368406   .8474597                     -4.029397   -.7074155
-------------+----------------------------------------------------------------
       alpha |   .0936298   .0793475                      .0177851    .4929165
------------------------------------------------------------------------------
Note: Estimates are transformed only in the first equation to incidence-rate
      ratios.
Note: _cons estimates baseline incidence rate (conditional on zero random
      effects).
LR test of alpha=0: chibar2(01) = 10.61                Prob >= chibar2 = 0.001
```

The output also includes a likelihood-ratio test of $\alpha = 0$, which compares the panel estimator with the pooled (Poisson) estimator.

We find that the incidence rate for accidents is significantly different for the periods of construction and operation of the ships and that the random-effects model is significantly different from the pooled model.

We may alternatively fit a fixed-effects specification instead of a random-effects specification:

```
. xtpoisson accident op_75_79 co_65_69 co_70_74 co_75_79, exp(service) irr fe
Iteration 0:  Log likelihood = -80.738973
Iteration 1:  Log likelihood = -54.857546
Iteration 2:  Log likelihood = -54.641897
Iteration 3:  Log likelihood = -54.641859
Iteration 4:  Log likelihood = -54.641859
Conditional fixed-effects Poisson regression         Number of obs    =     34
Group variable: ship                                 Number of groups =      5
                                                     Obs per group:
                                                                  min =      6
                                                                  avg =    6.8
                                                                  max =      7
                                                     Wald chi2(4)     =  48.44
Log likelihood = -54.641859                          Prob > chi2      = 0.0000

    accident |        IRR   Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
    op_75_79 |   1.468831   .1737218     3.25   0.001     1.164926    1.852019
    co_65_69 |   2.008002   .3004803     4.66   0.000     1.497577    2.692398
    co_70_74 |    2.26693    .384865     4.82   0.000     1.625274    3.161912
    co_75_79 |   1.573695   .3669393     1.94   0.052     .9964273    2.485397
ln(service) |          1  (exposure)
```

Both of these models fit the same thing but will differ in efficiency, depending on whether the assumptions of the random-effects model are true.

We could have assumed that the random effects followed a normal distribution, $N(0, \sigma_\nu^2)$, instead of a "log-gamma" distribution, and obtained

```
. xtpoisson accident op_75_79 co_65_69 co_70_74 co_75_79, exp(service) irr
> normal nolog

Random-effects Poisson regression               Number of obs     =      34
Group variable: ship                            Number of groups  =       5

Random effects u_i ~ Gaussian                   Obs per group:
                                                             min =        6
                                                             avg =      6.8
                                                             max =        7

Integration method: mvaghermite                 Integration pts.  =      12

                                                Wald chi2(4)      =   50.95
Log likelihood = -74.780982                     Prob > chi2       =  0.0000

------------------------------------------------------------------------------
    accident |        IRR   Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
    op_75_79 |   1.466677   .1734403     3.24   0.001     1.163259    1.849236
    co_65_69 |   2.032604   .3040933     4.74   0.000     1.516025    2.725205
    co_70_74 |   2.357045   .3998397     5.05   0.000     1.690338    3.286717
    co_75_79 |   1.646935   .3820235     2.15   0.031     1.045278    2.594905
       _cons |   .0013075   .0002775   -31.28   0.000     .0008625     .001982
ln(service) |          1  (exposure)
-------------+----------------------------------------------------------------
    /lnsig2u |  -2.351868   .8586262                     -4.034745   -.6689918
-------------+----------------------------------------------------------------
     sigma_u |   .3085306   .1324562                      .1330045    .7156988
------------------------------------------------------------------------------
Note: Estimates are transformed only in the first equation to incidence-rate
      ratios.
Note: _cons estimates baseline incidence rate (conditional on zero random
      effects).
LR test of sigma_u=0: chibar2(01) = 10.67             Prob >= chibar2 = 0.001
```

The output includes the additional panel-level variance component. This is parameterized as the log of the variance $\ln(\sigma_\nu^2)$ (labeled `lnsig2u` in the output). The standard deviation σ_ν is also included in the output labeled `sigma_u`.

When `sigma_u` is zero, the panel-level variance component is unimportant and the panel estimator is no different from the pooled estimator. A likelihood-ratio test of this is included at the bottom of the output. This test formally compares the pooled estimator (`poisson`) with the panel estimator. Here σ_ν is significantly greater than zero, so a panel estimator is indicated.

◁

▷ Example 2

This time we fit a robust equal-correlation population-averaged model:

```
. xtpoisson accident op_75_79 co_65_69 co_70_74 co_75_79, exp(service) pa
> vce(robust) eform
Iteration 1:  Tolerance = .04083192
Iteration 2:  Tolerance = .00270188
Iteration 3:  Tolerance = .00030663
Iteration 4:  Tolerance = .00003466
Iteration 5:  Tolerance = 3.891e-06
Iteration 6:  Tolerance = 4.359e-07
GEE population-averaged model                      Number of obs     =     34
Group variable: ship                               Number of groups =      5
Family: Poisson                                    Obs per group:
Link:   Log                                                     min =      6
Correlation: exchangeable                                       avg =    6.8
                                                                max =      7
                                                   Wald chi2(4)      = 252.94
Scale parameter = 1                                Prob > chi2       = 0.0000
                                   (Std. err. adjusted for clustering on ship)

                             Robust
    accident        IRR    std. err.      z    P>|z|     [95% conf. interval]

    op_75_79    1.483299   .1197901     4.88   0.000     1.266153    1.737685
    co_65_69    2.038477   .1809524     8.02   0.000     1.712955    2.425859
    co_70_74    2.643467   .4093947     6.28   0.000     1.951407    3.580962
    co_75_79    1.876656     .33075     3.57   0.000     1.328511    2.650966
       _cons    .0010255   .0000721   -97.90   0.000     .0008935     .001177
 ln(service)           1  (exposure)

Note: _cons estimates baseline incidence rate (conditional on zero random
      effects).
```

We may compare this with a pooled estimator with clustered robust-variance estimates:

```
. poisson accident op_75_79 co_65_69 co_70_74 co_75_79, exp(service)
> vce(cluster ship) irr
Iteration 0:  Log pseudolikelihood = -147.37993
Iteration 1:  Log pseudolikelihood = -80.372714
Iteration 2:  Log pseudolikelihood = -80.116093
Iteration 3:  Log pseudolikelihood = -80.115916
Iteration 4:  Log pseudolikelihood = -80.115916
Poisson regression                                      Number of obs =     34
                                                        Wald chi2(3)  =      .
                                                        Prob > chi2   =      .
Log pseudolikelihood = -80.115916                       Pseudo R2     = 0.3438
                                   (Std. err. adjusted for 5 clusters in ship)

                             Robust
    accident         IRR   std. err.      z    P>|z|     [95% conf. interval]

    op_75_79     1.47324   .1287036     4.44   0.000       1.2414    1.748377
    co_65_69    2.125914   .2850531     5.62   0.000     1.634603    2.764897
    co_70_74    2.860138   .6213563     4.84   0.000     1.868384    4.378325
    co_75_79    2.021926   .4265285     3.34   0.001     1.337221    3.057227
       _cons    .0009609   .0000277  -240.66   0.000      .000908    .0010168
ln(service)            1  (exposure)

Note: _cons estimates baseline incidence rate.
```

◁

❑ Technical note

The random-effects model is calculated using quadrature, which is an approximation whose accuracy depends partially on the number of integration points used. We can use the `quadchk` command to see if changing the number of integration points affects the results. If the results change, the quadrature approximation is not accurate given the number of integration points. Try increasing the number of integration points using the `intpoints()` option and run `quadchk` again. Do not attempt to interpret the results of estimates when the coefficients reported by `quadchk` differ substantially. See [XT] **quadchk** for details and [XT] **xtprobit** for an example.

Because the `xtpoisson, re normal` likelihood function is calculated by Gauss–Hermite quadrature, on large problems the computations can be slow. Computation time is roughly proportional to the number of points used for the quadrature.

❑

Stored results

`xtpoisson, re` stores the following in `e()`:

Scalars

`e(N)`	number of observations
`e(N_g)`	number of groups
`e(k)`	number of parameters
`e(k_aux)`	number of auxiliary parameters
`e(k_eq)`	number of equations in `e(b)`
`e(k_eq_model)`	number of equations in overall model test
`e(k_dv)`	number of dependent variables
`e(df_m)`	model degrees of freedom

`e(ll)`	log likelihood
`e(ll_0)`	log likelihood, constant-only model
`e(ll_c)`	log likelihood, comparison model
`e(chi2)`	χ^2
`e(chi2_c)`	χ^2 for comparison test
`e(N_clust)`	number of clusters
`e(alpha)`	value of alpha
`e(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(g_max)`	largest group size
`e(p)`	p-value for model test
`e(rank)`	rank of `e(V)`
`e(rank0)`	rank of `e(V)` for constant-only model
`e(ic)`	number of iterations
`e(rc)`	return code
`e(converged)`	`1` if converged, `0` otherwise

Macros

`e(cmd)`	`xtpoisson`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(ivar)`	variable denoting groups
`e(model)`	`re`
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(title)`	title in estimation output
`e(clustvar)`	name of cluster variable
`e(offset)`	linear offset variable
`e(chi2type)`	`Wald` or `LR`; type of model χ^2 test
`e(chi2_ct)`	`Wald` or `LR`; type of model χ^2 test corresponding to `e(chi2_c)`
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. err.
`e(method)`	requested estimation method
`e(distrib)`	`Gamma`; the distribution of the random effect
`e(opt)`	type of optimization
`e(which)`	`max` or `min`; whether optimizer is to perform maximization or minimization
`e(ml_method)`	type of `ml` method
`e(user)`	name of likelihood-evaluator program
`e(technique)`	maximization technique
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`

Matrices

`e(b)`	coefficient vector
`e(Cns)`	constraints matrix
`e(ilog)`	iteration log
`e(gradient)`	gradient vector
`e(V)`	variance–covariance matrix of the estimators
`e(V_modelbased)`	model-based variance

Functions

`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

xtpoisson, re normal stores the following in e():

Scalars

e(N)	number of observations
e(N_g)	number of groups
e(k)	number of parameters
e(k_aux)	number of auxiliary parameters
e(k_eq)	number of equations in e(b)
e(k_eq_model)	number of equations in overall model test
e(k_dv)	number of dependent variables
e(df_m)	model degrees of freedom
e(ll)	log likelihood
e(ll_0)	log likelihood, constant-only model
e(ll_c)	log likelihood, comparison model
e(chi2)	χ^2
e(chi2_c)	χ^2 for comparison test
e(N_clust)	number of clusters
e(sigma_u)	panel-level standard deviation
e(n_quad)	number of quadrature points
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(p)	*p*-value for model test
e(rank)	rank of e(V)
e(rank0)	rank of e(V) for constant-only model
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros

e(cmd)	xtpoisson
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(ivar)	variable denoting groups
e(model)	re
e(wtype)	weight type
e(wexp)	weight expression
e(title)	title in estimation output
e(clustvar)	name of cluster variable
e(offset)	linear offset variable
e(chi2type)	Wald or LR; type of model χ^2 test
e(chi2_ct)	Wald or LR; type of model χ^2 test corresponding to e(chi2_c)
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(intmethod)	integration method
e(distrib)	Gaussian; the distribution of the random effect
e(opt)	type of optimization
e(which)	max or min; whether optimizer is to perform maximization or minimization
e(ml_method)	type of ml method
e(user)	name of likelihood-evaluator program
e(technique)	maximization technique
e(properties)	b V
e(predict)	program used to implement predict
e(marginsdefault)	default predict() specification for margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved

Matrices	
e(b)	coefficient vector
e(Cns)	constraints matrix
e(ilog)	iteration log
e(gradient)	gradient vector
e(V)	variance–covariance matrix of the estimators
e(V_modelbased)	model-based variance
Functions	
e(sample)	marks estimation sample

In addition to the above, the following is stored in r():

Matrices	
r(table)	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

xtpoisson, fe stores the following in e():

Scalars	
e(N)	number of observations
e(N_g)	number of groups
e(k)	number of parameters
e(k_eq)	number of equations in e(b)
e(k_eq_model)	number of equations in overall model test
e(k_dv)	number of dependent variables
e(df_m)	model degrees of freedom
e(ll)	log likelihood
e(ll_c)	log likelihood, comparison model
e(chi2)	χ^2
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(p)	p-value for model test
e(rank)	rank of e(V)
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if converged, 0 otherwise
Macros	
e(cmd)	xtpoisson
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(ivar)	variable denoting groups
e(model)	fe
e(wtype)	weight type
e(wexp)	weight expression
e(title)	title in estimation output
e(offset)	linear offset variable
e(chi2type)	Wald; type of model χ^2 test
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(method)	requested estimation method
e(opt)	type of optimization
e(which)	max or min; whether optimizer is to perform maximization or minimization
e(ml_method)	type of ml method
e(user)	name of likelihood-evaluator program

e(technique)	maximization technique
e(properties)	b V
e(predict)	program used to implement predict
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved
Matrices	
e(b)	coefficient vector
e(Cns)	constraints matrix
e(ilog)	iteration log
e(gradient)	gradient vector
e(V)	variance–covariance matrix of the estimators
Functions	
e(sample)	marks estimation sample

In addition to the above, the following is stored in r():

Matrices	
r(table)	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

xtpoisson, pa stores the following in e():

Scalars	
e(N)	number of observations
e(N_g)	number of groups
e(df_m)	model degrees of freedom
e(chi2)	χ^2
e(p)	p-value for model test
e(df_pear)	degrees of freedom for Pearson χ^2
e(chi2_dev)	χ^2 test of deviance
e(chi2_dis)	χ^2 test of deviance dispersion
e(deviance)	deviance
e(dispers)	deviance dispersion
e(phi)	scale parameter
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(rank)	rank of e(V)
e(tol)	target tolerance
e(dif)	achieved tolerance
e(rc)	return code
Macros	
e(cmd)	xtgee
e(cmd2)	xtpoisson
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(ivar)	variable denoting groups
e(tvar)	variable denoting time within groups
e(model)	pa
e(family)	Poisson
e(link)	log; link function
e(corr)	correlation structure
e(scale)	x2, dev, phi, or #; scale parameter
e(wtype)	weight type
e(wexp)	weight expression

`e(offset)`	linear offset variable
`e(chi2type)`	`Wald`; type of model χ^2 test
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	covariance estimation method
`e(nmp)`	`nmp`, if specified
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(marginsnotok)`	predictions disallowed by `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`
Matrices	
`e(b)`	coefficient vector
`e(R)`	estimated working correlation matrix
`e(V)`	variance–covariance matrix of the estimators
`e(V_modelbased)`	model-based variance
Functions	
`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices	
`r(table)`	matrix containing the coefficients with their standard errors, test statistics, *p*-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

`xtpoisson, pa` reports the population-averaged results obtained by using `xtgee, family(poisson) link(log)` to obtain estimates. See [XT] **xtgee** for details about the methods and formulas.

`xtpoisson, fe` with robust standard errors implements the formula presented in Wooldridge (1999). The formula is a cluster–robust estimate of the VCE in which the ID variable specifies the clusters.

Although Hausman, Hall, and Griliches (1984) wrote the seminal article on the random-effects and fixed-effects models, Cameron and Trivedi (2013) provide a good textbook treatment. Allison (2009, chap. 4) succinctly discusses these models and illustrates the differences between them using Stata.

For a random-effects specification, we know that

$$\Pr(y_{i1},\dots,y_{in_i}|\alpha_i,\mathbf{x}_{i1},\dots,\mathbf{x}_{in_i}) = \left(\prod_{t=1}^{n_i}\frac{\lambda_{it}^{y_{it}}}{y_{it}!}\right)\exp\left\{-\exp(\alpha_i)\sum_{t=1}^{n_i}\lambda_{it}\right\}\exp\left(\alpha_i\sum_{t=1}^{n_i}y_{it}\right)$$

where $\lambda_{it} = \exp(\mathbf{x}_{it}\boldsymbol{\beta})$. We may rewrite the above as [defining $\epsilon_i = \exp(\alpha_i)$]

$$\begin{aligned}\Pr(y_{i1},\dots,y_{in_i}|\epsilon_i,\mathbf{x}_{i1},\dots,\mathbf{x}_{in_i}) &= \left\{\prod_{t=1}^{n_i}\frac{(\lambda_{it}\epsilon_i)^{y_{it}}}{y_{it}!}\right\}\exp\left(-\sum_{t=1}^{n_i}\lambda_{it}\epsilon_i\right)\\ &= \left(\prod_{t=1}^{n_i}\frac{\lambda_{it}^{y_{it}}}{y_{it}!}\right)\exp\left(-\epsilon_i\sum_{t=1}^{n_i}\lambda_{it}\right)\epsilon_i^{\sum_{t=1}^{n_i}y_{it}}\end{aligned}$$

We now assume that ϵ_i follows a gamma distribution with mean one and variance $1/\theta$ so that unconditional on ϵ_i

$$\begin{aligned}
\Pr(y_{i1},\dots,y_{in_i}|\mathbf{X}_i) &= \frac{\theta^\theta}{\Gamma(\theta)}\left(\prod_{t=1}^{n_i}\frac{\lambda_{it}^{y_{it}}}{y_{it}!}\right)\int_0^\infty \exp\left(-\epsilon_i\sum_{t=1}^{n_i}\lambda_{it}\right)\epsilon_i^{\sum_{t=1}^{n_i}y_{it}}\epsilon_i^{\theta-1}\exp(-\theta\epsilon_i)d\epsilon_i \\
&= \frac{\theta^\theta}{\Gamma(\theta)}\left(\prod_{t=1}^{n_i}\frac{\lambda_{it}^{y_{it}}}{y_{it}!}\right)\int_0^\infty \exp\left\{-\epsilon_i\left(\theta+\sum_{t=1}^{n_i}\lambda_{it}\right)\right\}\epsilon_i^{\theta+\sum_{t=1}^{n_i}y_{it}-1}d\epsilon_i \\
&= \left(\prod_{t=1}^{n_i}\frac{\lambda_{it}^{y_{it}}}{y_{it}!}\right)\frac{\Gamma\left(\theta+\sum_{t=1}^{n_i}y_{it}\right)}{\Gamma(\theta)}\left(\frac{\theta}{\theta+\sum_{t=1}^{n_i}\lambda_{it}}\right)^\theta\left(\frac{1}{\theta+\sum_{t=1}^{n_i}\lambda_{it}}\right)^{\sum_{t=1}^{n_i}y_{it}}
\end{aligned}$$

for $\mathbf{X}_i=(\mathbf{x}_{i1},\dots,\mathbf{x}_{in_i})$.

The log likelihood (assuming gamma heterogeneity) is then derived using

$$u_i=\frac{\theta}{\theta+\sum_{t=1}^{n_i}\lambda_{it}} \qquad \lambda_{it}=\exp(\mathbf{x}_{it}\boldsymbol{\beta})$$

$$\Pr(Y_{i1}=y_{i1},\dots,Y_{in_i}=y_{in_i}|\mathbf{X}_i)=\frac{\prod_{t=1}^{n_i}\lambda_{it}^{y_{it}}\Gamma\left(\theta+\sum_{t=1}^{n_i}y_{it}\right)}{\prod_{t=1}^{n_i}y_{it}!\Gamma(\theta)\left(\sum_{t=1}^{n_i}\lambda_{it}\right)^{\sum_{t=1}^{n_i}y_{it}}}u_i^\theta(1-u_i)^{\sum_{t=1}^{n_i}y_{it}}$$

such that the log likelihood may be written as

$$\begin{aligned}
L=\sum_{i=1}^n w_i\Bigg\{&\log\Gamma\left(\theta+\sum_{t=1}^{n_i}y_{it}\right)-\sum_{t=1}^{n_i}\log\Gamma(1+y_{it})-\log\Gamma(\theta)+\theta\log u_i \\
&+\log(1-u_i)\sum_{t=1}^{n_i}y_{it}+\sum_{t=1}^{n_i}y_{it}(\mathbf{x}_{it}\boldsymbol{\beta})-\left(\sum_{t=1}^{n_i}y_{it}\right)\log\left(\sum_{t=1}^{n_i}\lambda_{it}\right)\Bigg\}
\end{aligned}$$

where w_i is the user-specified weight for panel i; if no weights are specified, $w_i=1$.

Alternatively, if we assume a normal distribution, $N(0,\sigma_\nu^2)$, for the random effects ν_i

$$\Pr(y_{i1},\dots,y_{in_i}|\mathbf{X}_i)=\int_{-\infty}^\infty\frac{e^{-\nu_i^2/2\sigma_\nu^2}}{\sqrt{2\pi}\sigma_\nu}\left\{\prod_{t=1}^{n_i}F(y_{it},\mathbf{x}_{it}\boldsymbol{\beta}+\nu_i)\right\}d\nu_i$$

where

$$F(y,z)=\exp\Big\{-\exp(z)+yz-\log(y!)\Big\}.$$

The panel-level likelihood l_i is given by

$$\begin{aligned}
l_i&=\int_{-\infty}^\infty\frac{e^{-\nu_i^2/2\sigma_\nu^2}}{\sqrt{2\pi}\sigma_\nu}\left\{\prod_{t=1}^{n_i}F(y_{it},\mathbf{x}_{it}\boldsymbol{\beta}+\nu_i)\right\}d\nu_i \\
&\equiv\int_{-\infty}^\infty g(y_{it},x_{it},\nu_i)d\nu_i
\end{aligned}$$

This integral can be approximated with M-point Gauss–Hermite quadrature

$$\int_{-\infty}^{\infty} e^{-x^2} h(x)dx \approx \sum_{m=1}^{M} w_m^* h(a_m^*)$$

This is equivalent to

$$\int_{-\infty}^{\infty} f(x)dx \approx \sum_{m=1}^{M} w_m^* \exp\left\{(a_m^*)^2\right\} f(a_m^*)$$

where the w_m^* denote the quadrature weights and the a_m^* denote the quadrature abscissas. The log likelihood, L, is the sum of the logs of the panel-level likelihoods l_i.

The default approximation of the log likelihood is by adaptive Gauss–Hermite quadrature, which approximates the panel-level likelihood with

$$l_i \approx \sqrt{2}\widehat{\sigma}_i \sum_{m=1}^{M} w_m^* \exp\left\{(a_m^*)^2\right\} g(y_{it}, x_{it}, \sqrt{2}\widehat{\sigma}_i a_m^* + \widehat{\mu}_i)$$

where $\widehat{\sigma}_i$ and $\widehat{\mu}_i$ are the adaptive parameters for panel i. Therefore, with the definition of $g(y_{it}, x_{it}, \nu_i)$, the total log likelihood is approximated by

$$L \approx \sum_{i=1}^{n} w_i \log\Bigg[\sqrt{2}\widehat{\sigma}_i \sum_{m=1}^{M} w_m^* \exp\{(a_m^*)^2\} \frac{\exp\{-(\sqrt{2}\widehat{\sigma}_i a_m^* + \widehat{\mu}_i)^2/2\sigma_\nu^2\}}{\sqrt{2\pi}\sigma_\nu}$$
$$\prod_{t=1}^{n_i} F(y_{it}, x_{it}\boldsymbol{\beta} + \sqrt{2}\widehat{\sigma}_i a_m^* + \widehat{\mu}_i)\Bigg]$$

where w_i is the user-specified weight for panel i; if no weights are specified, $w_i = 1$.

The default method of adaptive Gauss–Hermite quadrature is to calculate the posterior mean and variance and use those parameters for $\widehat{\mu}_i$ and $\widehat{\sigma}_i$ by following the method of Naylor and Smith (1982), further discussed in Skrondal and Rabe-Hesketh (2004). We start with $\widehat{\sigma}_{i,0} = 1$ and $\widehat{\mu}_{i,0} = 0$, and the posterior means and variances are updated in the kth iteration. That is, at the kth iteration of the optimization for l_i, we use

$$l_{i,k} \approx \sum_{m=1}^{M} \sqrt{2}\widehat{\sigma}_{i,k-1} w_m^* \exp\{a_m^*)^2\} g(y_{it}, x_{it}, \sqrt{2}\widehat{\sigma}_{i,k-1} a_m^* + \widehat{\mu}_{i,k-1})$$

Letting

$$\tau_{i,m,k-1} = \sqrt{2}\widehat{\sigma}_{i,k-1} a_m^* + \widehat{\mu}_{i,k-1}$$

$$\widehat{\mu}_{i,k} = \sum_{m=1}^{M} (\tau_{i,m,k-1}) \frac{\sqrt{2}\widehat{\sigma}_{i,k-1} w_m^* \exp\{(a_m^*)^2\} g(y_{it}, x_{it}, \tau_{i,m,k-1})}{l_{i,k}}$$

and

$$\widehat{\sigma}_{i,k} = \sum_{m=1}^{M} (\tau_{i,m,k-1})^2 \frac{\sqrt{2}\widehat{\sigma}_{i,k-1} w_m^* \exp\{(a_m^*)^2\} g(y_{it}, x_{it}, \tau_{i,m,k-1})}{l_{i,k}} - (\widehat{\mu}_{i,k})^2$$

and this is repeated until $\widehat{\mu}_{i,k}$ and $\widehat{\sigma}_{i,k}$ have converged for this iteration of the maximization algorithm. This adaptation is applied on every iteration until the log-likelihood change from the preceding iteration is less than a relative difference of 1e–6; after this, the quadrature parameters are fixed.

The log likelihood can also be calculated by nonadaptive Gauss–Hermite quadrature, the `intmethod(ghermite)` option, where $\rho = \sigma_\nu^2/(\sigma_\nu^2+1)$:

$$\begin{aligned}L &= \sum_{i=1}^{n} w_i \log\Big\{ \Pr(y_{i1},\dots,y_{in_i}|\mathbf{x}_{i1},\dots,\mathbf{x}_{in_i})\Big\} \\ &\approx \sum_{i=1}^{n} w_i \log\left[\frac{1}{\sqrt{\pi}}\sum_{m=1}^{M} w_m^* \prod_{t=1}^{n_i} F\left\{y_{it}, \mathbf{x}_{it}\boldsymbol{\beta} + a_m^*\left(\frac{2\rho}{1-\rho}\right)^{1/2}\right\}\right]\end{aligned}$$

Both quadrature formulas require that the integrated function be well approximated by a polynomial of degree equal to the number of quadrature points. The number of periods (panel size) can affect whether

$$\prod_{t=1}^{n_i} F(y_{it}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i)$$

is well approximated by a polynomial. As panel size and ρ increase, the quadrature approximation can become less accurate. For large ρ, the random-effects model can also become unidentified. Adaptive quadrature gives better results for correlated data and large panels than nonadaptive quadrature; however, we recommend that you use the `quadchk` command (see [XT] **quadchk**) to verify the quadrature approximation used in this command, whichever approximation you choose.

For a fixed-effects specification, we know that

$$\begin{aligned}\Pr(Y_{it} = y_{it}|\mathbf{x}_{it}) &= \exp\{-\exp(\alpha_i + \mathbf{x}_{it}\boldsymbol{\beta})\}\exp(\alpha_i + \mathbf{x}_{it}\boldsymbol{\beta})^{y_{it}}/y_{it}! \\ &= \frac{1}{y_{it}!}\exp\{-\exp(\alpha_i)\exp(\mathbf{x}_{it}\boldsymbol{\beta}) + \alpha_i y_{it}\}\exp(\mathbf{x}_{it}\boldsymbol{\beta})^{y_{it}} \\ &\equiv F_{it}\end{aligned}$$

Because we know that the observations are independent, we may write the joint probability for the observations within a panel as

$$\begin{aligned}&\Pr\Big(Y_{i1} = y_{i1},\dots,Y_{in_i} = y_{in_i}|\mathbf{X}_i\Big) \\ &\qquad= \prod_{t=1}^{n_i}\frac{1}{y_{it}!}\exp\{-\exp(\alpha_i)\exp(\mathbf{x}_{it}\boldsymbol{\beta}) + \alpha_i y_{it}\}\exp(\mathbf{x}_{it}\boldsymbol{\beta})^{y_{it}} \\ &\qquad= \left(\prod_{t=1}^{n_i}\frac{\exp(\mathbf{x}_{it}\boldsymbol{\beta})^{y_{it}}}{y_{it}!}\right)\exp\left\{-\exp(\alpha_i)\sum_t \exp(\mathbf{x}_{it}\boldsymbol{\beta}) + \alpha_i\sum_t y_{it}\right\}\end{aligned}$$

and we also know that the sum of n_i Poisson independent random variables, each with parameter λ_{it} for $t = 1,\dots,n_i$, is distributed as Poisson with parameter $\sum_t \lambda_{it}$. Thus

$$\begin{aligned}&\Pr\left(\sum_t Y_{it} = \sum_t y_{it}\middle|\mathbf{X}_i\right) = \\ &\qquad\frac{1}{\left(\sum_t y_{it}\right)!}\exp\left\{-\exp(\alpha_i)\sum_t\exp(\mathbf{x}_{it}\boldsymbol{\beta}) + \alpha_i\sum_t y_{it}\right\}\left\{\sum_t\exp(\mathbf{x}_{it}\boldsymbol{\beta})\right\}^{\sum_t y_{it}}\end{aligned}$$

So the conditional likelihood is conditioned on the sum of the outcomes in the set (panel). The appropriate function is given by

$$\Pr\Big(Y_{i1}=y_{i1},\ldots,Y_{in_i}=y_{in_i}\Big|\mathbf{X}_i,\sum_t Y_{it}=\sum_t y_{it}\Big)=$$

$$\left[\left(\prod_{t=1}^{n_i}\frac{\exp(\mathbf{x}_{it}\boldsymbol\beta)^{y_{it}}}{y_{it}!}\right)\exp\left\{-\exp(\alpha_i)\sum_t\exp(\mathbf{x}_{it}\boldsymbol\beta)+\alpha_i\sum_t y_{it}\right\}\right]\Bigg/$$

$$\left[\frac{1}{\left(\sum_t y_{it}\right)!}\exp\left\{-\exp(\alpha_i)\sum_t\exp(\mathbf{x}_{it}\boldsymbol\beta)+\alpha_i\sum_t y_{it}\right\}\left\{\sum_t\exp(\mathbf{x}_{it}\boldsymbol\beta)\right\}^{\sum_t y_{it}}\right]$$

$$=\left(\sum_t y_{it}\right)!\prod_{t=1}^{n_i}\frac{\exp(\mathbf{x}_{it}\boldsymbol\beta)^{y_{it}}}{y_{it}!\left\{\sum_k\exp(\mathbf{x}_{ik}\boldsymbol\beta)\right\}^{y_{it}}}$$

which is free of α_i.

The conditional log likelihood is given by

$$L=\log\prod_{i=1}^{n}\left[\left(\sum_{t=1}^{n_i}y_{it}\right)!\prod_{t=1}^{n_i}\frac{\exp(\mathbf{x}_{it}\boldsymbol\beta)^{y_{it}}}{y_{it}!\left\{\sum_{\ell=1}^{n_\ell}\exp(\mathbf{x}_{i\ell}\boldsymbol\beta)\right\}^{y_{it}}}\right]^{w_i}$$

$$=\log\prod_{i=1}^{n}\left\{\frac{\left(\sum_t y_{it}\right)!}{\prod_{t=1}^{n_i}y_{it}!}\prod_{t=1}^{n_i}p_{it}^{y_{it}}\right\}^{w_i}$$

$$=\sum_{i=1}^{n}w_i\left\{\log\Gamma\left(\sum_{t=1}^{n_i}y_{it}+1\right)-\sum_{t=1}^{n_i}\log\Gamma(y_{it}+1)+\sum_{t=1}^{n_i}y_{it}\log p_{it}\right\}$$

where

$$p_{it}=e^{\mathbf{x}_{it}\boldsymbol\beta}\Big/\sum_\ell e^{\mathbf{x}_{i\ell}\boldsymbol\beta}$$

xtpoisson, re and the robust VCE estimator

Specifying `vce(robust)` or `vce(cluster` *clustvar*`)` causes the Huber/White/sandwich VCE estimator to be calculated for the coefficients estimated in this regression. See [P] **_robust**, particularly *Introduction* and *Methods and formulas*. Wooldridge (2020) and Arellano (2003) discuss this application of the Huber/White/sandwich VCE estimator. As discussed by Wooldridge (2020), Stock and Watson (2008), and Arellano (2003), specifying `vce(robust)` is equivalent to specifying `vce(cluster` *panelvar*`)`, where *panelvar* is the variable that identifies the panels.

Clustering on the panel variable produces a consistent VCE estimator when the disturbances are not identically distributed over the panels or there is serial correlation in ϵ_{it}.

The cluster–robust VCE estimator requires that there are many clusters and the disturbances are uncorrelated across the clusters. The panel variable must be nested within the cluster variable because of the within-panel correlation that is generally induced by the random-effects transform when there is heteroskedasticity or within-panel serial correlation in the idiosyncratic errors.

References

Allison, P. D. 2009. *Fixed Effects Regression Models*. Newbury Park, CA: Sage.

Arellano, M. 2003. *Panel Data Econometrics*. Oxford: Oxford University Press. https://doi.org/10.1093/0199245282.001.0001.

Baltagi, B. H. 2009. *A Companion to Econometric Analysis of Panel Data*. Chichester, UK: Wiley.

———. 2013. *Econometric Analysis of Panel Data*. 5th ed. Chichester, UK: Wiley.

Cameron, A. C., and P. K. Trivedi. 2013. *Regression Analysis of Count Data*. 2nd ed. New York: Cambridge University Press.

Correia, S., P. Guimarães, and T. Zylkin. 2020. Fast Poisson estimation with high-dimensional fixed effects. *Stata Journal* 20: 95–115.

Cummings, P. 2011. Estimating adjusted risk ratios for matched and unmatched data: An update. *Stata Journal* 11: 290–298.

Hardin, J. W., and J. M. Hilbe. 2018. *Generalized Linear Models and Extensions*. 4th ed. College Station, TX: Stata Press.

Hausman, J. A., B. H. Hall, and Z. Griliches. 1984. Econometric models for count data with an application to the patents–R & D relationship. *Econometrica* 52: 909–938. https://doi.org/10.2307/1911191.

Liang, K.-Y., and S. L. Zeger. 1986. Longitudinal data analysis using generalized linear models. *Biometrika* 73: 13–22. https://doi.org/10.1093/biomet/73.1.13.

McCullagh, P., and J. A. Nelder. 1989. *Generalized Linear Models*. 2nd ed. London: Chapman and Hall/CRC.

Naylor, J. C., and A. F. M. Smith. 1982. Applications of a method for the efficient computation of posterior distributions. *Journal of the Royal Statistical Society,* C ser., 31: 214–225. https://doi.org/10.2307/2347995.

Skrondal, A., and S. Rabe-Hesketh. 2004. *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models*. Boca Raton, FL: Chapman and Hall/CRC.

Stock, J. H., and M. W. Watson. 2008. Heteroskedasticity-robust standard errors for fixed effects panel data regression. *Econometrica* 76: 155–174. https://doi.org/10.1111/j.0012-9682.2008.00821.x.

Wooldridge, J. M. 1999. Distribution-free estimation of some nonlinear panel data models. *Journal of Econometrics* 90: 77–97. https://doi.org/10.1016/S0304-4076(98)00033-5.

———. 2020. *Introductory Econometrics: A Modern Approach*. 7th ed. Boston: Cengage.

Also see

[XT] **xtpoisson postestimation** — Postestimation tools for xtpoisson

[XT] **quadchk** — Check sensitivity of quadrature approximation

[XT] **xtgee** — GEE population-averaged panel-data models

[XT] **xtnbreg** — Fixed-effects, random-effects, & population-averaged negative binomial models

[XT] **xtset** — Declare data to be panel data

[BAYES] **bayes: xtpoisson** — Bayesian random-effects Poisson model

[ME] **mepoisson** — Multilevel mixed-effects Poisson regression

[MI] **Estimation** — Estimation commands for use with mi estimate

[R] **poisson** — Poisson regression

[U] 20 Estimation and postestimation commands

Postestimation commands

The following postestimation commands are available after `xtpoisson`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of parameters
* `estat ic`	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC, respectively)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
† `forecast`	dynamic forecasts and simulations
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
* `lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	linear predictions and their SEs, number of events, incidence rates, probabilities
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of parameters
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

*`estat ic` and `lrtest` are not appropriate after `xtpoisson, pa`.

†`forecast` is not appropriate with `mi` estimation results.

predict

Description for predict

`predict` creates a new variable containing predictions such as linear predictions, standard errors, numbers of events, incidence rates, probabilities, and the equation-level score.

Menu for predict

Statistics > Postestimation

Syntax for predict

Random-effects (RE) model

`predict` [*type*] *newvar* [*if*] [*in*] [, *RE_statistic* `nooffset`]

Fixed-effects (FE) model

`predict` [*type*] *newvar* [*if*] [*in*] [, *FE_statistic* `nooffset`]

Population-averaged (PA) model

`predict` [*type*] *newvar* [*if*] [*in*] [, *PA_statistic* `nooffset`]

RE_statistic	Description
Main	
`xb`	linear prediction; the default
`stdp`	standard error of the linear prediction
`n`	predicted number of events marginal with respect to the random effect; only allowed after `xtpoisson, re normal`
`nu0`	predicted number of events assuming the random effect is zero
`iru0`	predicted incidence rate assuming the random effect is zero
`pr0(`*n*`)`	probability $\Pr(y = n)$ assuming the random effect is zero
`pr0(`*a*`,`*b*`)`	probability $\Pr(a \leq y \leq b)$ assuming the random effect is zero

FE_statistic	Description
Main	
`xb`	linear prediction; the default
`stdp`	standard error of the linear prediction
`nu0`	predicted number of events assuming the fixed effect is zero
`iru0`	predicted incidence rate assuming the fixed effect is zero

PA_statistic	Description
Main	
`mu`	predicted number of events; considers the `offset()`; the default
`rate`	predicted number of events
`xb`	linear prediction
`pr(`*n*`)`	probability $\Pr(y = n)$
`pr(`*a*`,`*b*`)`	probability $\Pr(a \le y \le b)$
`stdp`	standard error of the linear prediction
`score`	first derivative of the log likelihood with respect to $\mathbf{x}_{it}\boldsymbol{\beta}$

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

Options for predict

Main

`xb` calculates the linear prediction. This is the default for the random-effects and fixed-effects models.

`mu` and `rate` both calculate the predicted number of events. `mu` takes into account the `offset()`, and `rate` ignores those adjustments. `mu` and `rate` are equivalent if you did not specify `offset()`. `mu` is the default for the population-averaged model.

`stdp` calculates the standard error of the linear prediction.

`n` calculates the predicted number of events marginally with respect to the random effect, which means that the statistic is calculated by integrating the prediction function with respect to the random effect over its entire support. This option is only allowed after `xtpoisson, re normal`.

`nu0` calculates the predicted number of events, assuming a zero random or fixed effect.

`iru0` calculates the predicted incidence rate, assuming a zero random or fixed effect.

`pr0(`*n*`)` calculates the probability $\Pr(y = n)$ assuming the random effect is zero, where *n* is a nonnegative integer that may be specified as a number or a variable (only allowed after `xtpoisson, re`).

`pr0(`*a*`,`*b*`)` calculates the probability $\Pr(a \le y \le b)$ assuming the random effect is zero, where *a* and *b* are nonnegative integers that may be specified as numbers or variables (only allowed after `xtpoisson, re`);

b missing ($b \ge .$) means $+\infty$;
`pr0(20,.)` calculates $\Pr(y \ge 20)$;
`pr0(20,`*b*`)` calculates $\Pr(y \ge 20)$ in observations for which $b \ge .$ and calculates $\Pr(20 \le y \le b)$ elsewhere.

`pr0(.,`*b*`)` produces a syntax error. A missing value in an observation of the variable *a* causes a missing value in that observation for `pr0(`*a*`,`*b*`)`.

`pr(`*n*`)` calculates the probability $\Pr(y = n)$, where *n* is a nonnegative integer that may be specified as a number or a variable (only allowed after `xtpoisson, pa`).

`pr(`*a*`,`*b*`)` calculates the probability $\Pr(a \le y \le b)$ (only allowed after `xtpoisson, pa`). The syntax for this option is analogous to that used with `pr0(`*a*`,`*b*`)`.

`score` calculates the equation-level score, $u_{it} = \partial \ln L(\mathbf{x}_{it}\boldsymbol{\beta})/\partial(\mathbf{x}_{it}\boldsymbol{\beta})$.

`nooffset` is relevant only if you specified `offset(`*varname*`)` for `xtpoisson`. It modifies the calculations made by `predict` so that they ignore the offset variable; the linear prediction is treated as $\mathbf{x}_{it}\boldsymbol{\beta}$ rather than $\mathbf{x}_{it}\boldsymbol{\beta} + \text{offset}_{it}$.

margins

Description for margins

`margins` estimates margins of response for linear predictions, numbers of events, incidence rates, and probabilities.

Menu for margins

Statistics > Postestimation

Syntax for margins

`margins` [*marginlist*] [, *options*]

`margins` [*marginlist*] , `predict(`*statistic* ...`)` [`predict(`*statistic* ...`)` ...] [*options*]

Random-effects (RE) model

statistic	Description
`xb`	linear prediction; the default after `xtpoisson, re`
`n`	predicted number of events marginal with respect to the random effect; the default after a
`nu0`	predicted number of events assuming the random effect is zero
`iru0`	predicted incidence rate assuming the random effect is zero
`pr0(`n`)`	probability $\Pr(y = n)$ assuming the random effect is zero
`pr0(`a`,`b`)`	probability $\Pr(a \leq y \leq b)$ assuming the random effect is zero
`stdp`	not allowed with `margins`

Fixed-effects (FE) model

statistic	Description
`xb`	linear prediction; the default
`nu0`	predicted number of events assuming the fixed effect is zero
`iru0`	predicted incidence rate assuming the fixed effect is zero
`stdp`	not allowed with `margins`

Population-averaged (PA) model

statistic	Description
mu	predicted number of events; considers the offset(); the default
rate	predicted number of events
xb	linear prediction
pr(*n*)	probability $\Pr(y = n)$
pr(*a*,*b*)	probability $\Pr(a \leq y \leq b)$
stdp	not allowed with margins
score	not allowed with margins

Statistics not allowed with margins are functions of stochastic quantities other than e(b).

For the full syntax, see [R] **margins**.

Remarks and examples

Example 1: Predicted number of events and incidence rate with no random effect

In example 1 of [XT] **xtpoisson**, we fit a random-effects model of the number of accidents experienced by five different types of ships on the basis of when the ships were constructed and operated. Here we obtain the predicted number of accidents for each observation, assuming that the random effect for each panel is zero:

```
. use https://www.stata-press.com/data/r19/ships
. xtpoisson accident op_75_79 co_65_69 co_70_74 co_75_79, exposure(service) irr
  (output omitted)
. predict n_acc, nu0
(6 missing values generated)
. summarize n_acc

    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
       n_acc |         34    13.52307    23.15885   .0617592   83.31905
```

From these results, you may be tempted to conclude that some types of ships are safe, with a predicted number of accidents close to zero, whereas others are dangerous, because 1 observation is predicted to have more than 83 accidents.

However, when we fit the model, we specified the exposure(service) option. The variable service records the total number of months of operation for each type of ship constructed in and operated during particular years. Because ships experienced different utilization rates and thus were exposed to different levels of accident risk, we included service as our exposure variable. When comparing different types of ships, we must therefore predict the number of accidents, assuming that all ships faced the same exposure to risk. To do that, we use the iru0 option with predict:

```
. predict acc_rate, iru0
. summarize acc_rate

    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
    acc_rate |         40     .002975    .0010497   .0013724   .0047429
```

These results show that if each ship were used for 1 month, the expected number of accidents is 0.002975. Depending on the type of ship and years of construction and operation, the *incidence rate* of accidents ranges from 0.00137 to 0.00474.

◁

Methods and formulas

The probabilities calculated using the `pr0(`*n*`)` option are the probability $\Pr(y_{it} = n)$ for a RE model assuming the random effect is zero. Define $\mu_{it} = \exp(\mathbf{x}_{it}\boldsymbol{\beta} + \mathrm{offset}_{it})$. The probabilities in `pr0(`*n*`)` are calculated as the probability that $y_{it} = n$, where y_{it} has a Poisson distribution with mean μ_{it}. Specifically,

$$\Pr(y_{it} = n) = (n!)^{-1} \exp(-\mu_{it})(\mu_{it})^n$$

Probabilities calculated using the `pr(`*n*`)` option after fitting a PA model are also calculated as described above.

Also see

[XT] **xtpoisson** — Fixed-effects, random-effects, and population-averaged Poisson models

[U] 20 Estimation and postestimation commands

xtprobit — Random-effects and population-averaged probit models

Description	Quick start	Menu	Syntax
Options for RE model	Options for PA model	Remarks and examples	Stored results
Methods and formulas	References	Also see	

Description

`xtprobit` fits random-effects and population-averaged probit models for a binary dependent variable. The probability of a positive outcome is assumed to be determined by the standard normal cumulative distribution function.

Quick start

Random-effects probit model of `y` as a function of `x1`, `x2`, and indicators for levels of categorical variable `a` using `xtset` data

```
xtprobit y x1 x2 i.a
```

Population-averaged model with robust standard errors

```
xtprobit y x1 x2 i.a, pa vce(robust)
```

Same as above, but specify an autoregressive correlation structure of order 1

```
xtprobit y x1 x2 i.a, pa vce(robust) corr(ar 1)
```

Random-effects model with cluster–robust standard errors for panels nested within `cvar`

```
xtprobit y x1 x2 i.a, vce(cluster cvar)
```

Menu

Statistics > Longitudinal/panel data > Binary outcomes > Probit regression (RE, PA)

Syntax

Random-effects (RE) model

`xtprobit` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, `re` *RE_options*]

Population-averaged (PA) model

`xtprobit` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] , `pa` [*PA_options*]

RE_options	Description
Model	
`noconstant`	suppress constant term
`re`	use random-effects estimator; the default
`offset(`*varname*`)`	include *varname* in model with coefficient constrained to 1
`constraints(`*constraints*`)`	apply specified linear constraints
`asis`	retain perfect predictor variables
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `robust`, `cluster` *clustvar*, `bootstrap`, or `jackknife`
Reporting	
`level(`#`)`	set confidence level; default is `level(95)`
`lrmodel`	perform the likelihood-ratio model test instead of the default Wald test
`nocnsreport`	do not display constraints
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Integration	
`intmethod(`*intmethod*`)`	integration method; *intmethod* may be `mvaghermite` (the default) or `ghermite`
`intpoints(`#`)`	use # quadrature points; default is `intpoints(12)`
Maximization	
maximize_options	control the maximization process; seldom used
`collinear`	keep collinear variables
`coeflegend`	display legend instead of statistics

PA_options	Description
Model	
noconstant	suppress constant term
pa	use population-averaged estimator
offset(*varname*)	include *varname* in model with coefficient constrained to 1
asis	retain perfect predictor variables
Correlation	
corr(*correlation*)	within-panel correlation structure
force	estimate even if observations unequally spaced in time
SE/Robust	
vce(*vcetype*)	*vcetype* may be conventional, robust, bootstrap, or jackknife
nmp	use divisor $N - P$ instead of the default N
scale(*parm*)	overrides the default scale parameter; *parm* may be x2, dev, phi, or #
Reporting	
level(#)	set confidence level; default is level(95)
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Optimization	
optimize_options	control the optimization process; seldom used
coeflegend	display legend instead of statistics

correlation	Description
exchangeable	exchangeable
independent	independent
unstructured	unstructured
fixed *matname*	user-specified
ar #	autoregressive of order #
stationary #	stationary of order #
nonstationary #	nonstationary of order #

A panel variable must be specified. For xtprobit, pa, correlation structures other than exchangeable and independent require that a time variable also be specified. Use xtset; see [XT] **xtset**.

indepvars may contain factor variables; see **[U] 11.4.3 Factor variables**.

depvar and *indepvars* may contain time-series operators; see **[U] 11.4.4 Time-series varlists**.

by, collect, mi estimate, and statsby are allowed; see **[U] 11.1.10 Prefix commands**. bayes is allowed for the random-effects model. For more details, see [BAYES] **bayes: xtprobit**. fp is allowed for the random-effects model.

vce(bootstrap) and vce(jackknife) are not allowed with the mi estimate prefix; see [MI] **mi estimate**.

iweights, fweights, and pweights are allowed for the population-averaged model, and iweights are allowed for the random-effects model; see **[U] 11.1.6 weight**. Weights must be constant within panel.

collinear and coeflegend do not appear in the dialog box.

See **[U] 20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options for RE model

Model

noconstant; see [R] **Estimation options**.

re requests the random-effects estimator. re is the default if neither re nor pa is specified.

offset(*varname*), constraints(*constraints*); see [R] **Estimation options**.

asis forces retention of perfect predictor variables and their associated, perfectly predicted observations and may produce instabilities in maximization; see [R] **probit**.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (oim), that are robust to some kinds of misspecification (robust), that allow for intragroup correlation (cluster *clustvar*), and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] ***vce_options***.

Specifying vce(robust) is equivalent to specifying vce(cluster *panelvar*); see *xtprobit, re and the robust VCE estimator* in *Methods and formulas*.

Reporting

level(#), lrmodel, nocnsreport; see [R] **Estimation options**.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Integration

intmethod(*intmethod*), intpoints(#); see [R] **Estimation options**.

Maximization

maximize_options: difficult, technique(*algorithm_spec*), iterate(#), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), nonrtolerance, and from(*init_specs*); see [R] **Maximize**. These options are seldom used.

The following options are available with xtprobit but are not shown in the dialog box:

collinear, coeflegend; see [R] **Estimation options**.

Options for PA model

Model

noconstant; see [R] **Estimation options**.

pa requests the population-averaged estimator.

offset(*varname*); see [R] **Estimation options**.

asis forces retention of perfect predictor variables and their associated, perfectly predicted observations and may produce instabilities in maximization; see [R] **probit**.

Correlation

corr(*correlation*) specifies the within-panel correlation structure; the default corresponds to the equal-correlation model, corr(exchangeable).

When you specify a correlation structure that requires a lag, you indicate the lag after the structure's name with or without a blank; for example, corr(ar 1) or corr(ar1).

If you specify the fixed correlation structure, you specify the name of the matrix containing the assumed correlations following the word fixed, for example, corr(fixed myr).

force specifies that estimation be forced even though the time variable is not equally spaced. This is relevant only for correlation structures that require knowledge of the time variable. These correlation structures require that observations be equally spaced so that calculations based on lags correspond to a constant time change. If you specify a time variable indicating that observations are not equally spaced, the (time dependent) model will not be fit. If you also specify force, the model will be fit, and it will be assumed that the lags based on the data ordered by the time variable are appropriate.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (conventional), that are robust to some kinds of misspecification (robust), and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] ***vce_options***.

vce(conventional), the default, uses the conventionally derived variance estimator for generalized least-squares regression.

nmp, scale(x2 | dev | phi | #); see [XT] ***vce_options***.

Reporting

level(#); see [R] **Estimation options**.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Optimization

optimize_options control the iterative optimization process. These options are seldom used.

iterate(#) specifies the maximum number of iterations. When the number of iterations equals #, the optimization stops and presents the current results, even if convergence has not been reached. The default is iterate(100).

tolerance(#) specifies the tolerance for the coefficient vector. When the relative change in the coefficient vector from one iteration to the next is less than or equal to #, the optimization process is stopped. tolerance(1e-6) is the default.

log and nolog specify whether to display the iteration log. The iteration log is displayed by default unless you used set iterlog off to suppress it; see set iterlog in [R] ***set iter***.

trace specifies that the current estimates be printed at each iteration.

The following option is available with xtprobit but is not shown in the dialog box:

coeflegend; see [R] **Estimation options**.

Remarks and examples

xtprobit may be used to fit a population-averaged model or a random-effects probit model. There is no command for a conditional fixed-effects model, as there does not exist a sufficient statistic allowing the fixed effects to be conditioned out of the likelihood. Unconditional fixed-effects probit models may be fit with the probit command with indicator variables for the panels. However, unconditional fixed-effects estimates are biased. We do not discuss fixed-effects further in this entry.

By default, the population-averaged model is an equal-correlation model; that is, xtprobit, pa assumes corr(exchangeable). Thus, xtprobit is a convenience command for obtaining the population-averaged model using xtgee; see [XT] **xtgee**. Typing

```
. xtprobit ..., pa ...
```

is equivalent to typing

```
. xtgee ..., ... family(binomial) link(probit) corr(exchangeable)
```

See also [XT] **xtgee** for information about xtprobit.

By default or when re is specified, xtprobit fits via maximum likelihood the random-effects model

$$\Pr(y_{it} \neq 0|\mathbf{x}_{it}) = \Phi(\mathbf{x}_{it}\boldsymbol{\beta} + \nu_i)$$

for $i = 1, \ldots, n$ panels, where $t = 1, \ldots, n_i$, ν_i are i.i.d., $N(0, \sigma^2_\nu)$, and Φ is the standard normal cumulative distribution function.

Underlying this model is the variance components model

$$y_{it} \neq 0 \iff \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i + \epsilon_{it} > 0$$

where ϵ_{it} are i.i.d. Gaussian distributed with mean zero and variance $\sigma^2_\epsilon = 1$, independently of ν_i.

▷ Example 1: Random-effects model

We are studying unionization of women in the United States and are using the union dataset; see [XT] **xt**. We wish to fit a random-effects model of union membership:

```
. use https://www.stata-press.com/data/r19/union
(NLS Women 14-24 in 1968)

. xtprobit union age grade i.not_smsa south##c.year

Fitting comparison model:

Iteration 0:  Log likelihood =  -13864.23
Iteration 1:  Log likelihood = -13545.541
Iteration 2:  Log likelihood = -13544.385
Iteration 3:  Log likelihood = -13544.385

Fitting full model:

rho =  0.0    Log likelihood = -13544.385
rho =  0.1    Log likelihood = -12237.655
rho =  0.2    Log likelihood = -11590.282
rho =  0.3    Log likelihood = -11211.185
rho =  0.4    Log likelihood = -10981.319
rho =  0.5    Log likelihood = -10852.793
rho =  0.6    Log likelihood = -10808.759
rho =  0.7    Log likelihood =  -10865.57

Iteration 0:  Log likelihood = -10807.712
Iteration 1:  Log likelihood = -10599.332
Iteration 2:  Log likelihood = -10552.287
Iteration 3:  Log likelihood = -10552.225
Iteration 4:  Log likelihood = -10552.225

Random-effects probit regression                  Number of obs     = 26,200
Group variable: idcode                            Number of groups  =  4,434

Random effects u_i ~ Gaussian                     Obs per group:
                                                               min =      1
                                                               avg =    5.9
                                                               max =     12

Integration method: mvaghermite                   Integration pts.  =     12

                                                  Wald chi2(6)      = 220.91
Log likelihood = -10552.225                       Prob > chi2       = 0.0000

------------------------------------------------------------------------------
       union | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
         age |   .0082967   .0084599     0.98   0.327    -.0082843    .0248778
       grade |   .0482731   .0099469     4.85   0.000     .0287776    .0677686
  1.not_smsa |   -.139657   .0460548    -3.03   0.002    -.2299227   -.0493913
     1.south |  -1.584394    .358473    -4.42   0.000    -2.286989   -.8818002
        year |  -.0039854   .0088399    -0.45   0.652    -.0213113    .0133406
             |
south#c.year |
           1 |   .0134017   .0044622     3.00   0.003     .0046559    .0221475
             |
       _cons |  -1.668202   .4751819    -3.51   0.000    -2.599542   -.7368628
-------------+----------------------------------------------------------------
    /lnsig2u |   .6103616   .0458783                      .5204418    .7002814
-------------+----------------------------------------------------------------
     sigma_u |    1.35687   .0311255                      1.297217    1.419267
         rho |   .6480233   .0104643                      .6272511    .6682502
------------------------------------------------------------------------------
LR test of rho=0: chibar2(01) = 5984.32                Prob >= chibar2 = 0.000
```

The output includes the additional panel-level variance component, which is parameterized as the log of the variance $\ln(\sigma_\nu^2)$ (labeled `lnsig2u` in the output). The standard deviation σ_ν is also included in the output (labeled `sigma_u`) together with ρ (labeled `rho`), where

$$\rho = \frac{\sigma_\nu^2}{\sigma_\nu^2 + 1}$$

which is the proportion of the total variance contributed by the panel-level variance component.

When `rho` is zero, the panel-level variance component is unimportant, and the panel estimator is not different from the pooled estimator. A likelihood-ratio test of this is included at the bottom of the output. This test formally compares the pooled estimator (probit) with the panel estimator.

◁

❑ Technical note

The random-effects model is calculated using quadrature, which is an approximation whose accuracy depends partially on the number of integration points used. We can use the quadchk command to see if changing the number of integration points affects the results. If the results change, the quadrature approximation is not accurate given the number of integration points. Try increasing the number of integration points using the intpoints() option and run quadchk again. Do not attempt to interpret the results of estimates when the coefficients reported by quadchk differ substantially.

```
. quadchk, nooutput
Refitting model intpoints() =  8
Refitting model intpoints() = 16
                         Quadrature check

                 Fitted       Comparison     Comparison
               quadrature     quadrature     quadrature
               12 points      8 points       16 points
─────────────────────────────────────────────────────────
Log           -10552.225     -10554.496     -10552.399
likelihood                   -2.2712569     -.17396615    Difference
                              .00021524      .00001649    Relative difference
─────────────────────────────────────────────────────────
union:          .00829671      .00828745      .00831488
     age                     -9.265e-06      .00001817    Difference
                              -.0011167      .00218987    Relative difference
─────────────────────────────────────────────────────────
union:           .0482731      .04860277      .04826287
   grade                      .00032967     -.00001023    Difference
                              .00682917     -.00021188    Relative difference
─────────────────────────────────────────────────────────
union:         -.13965702     -.14057441     -.13953521
1.not_smsa                   -.00091739      .00012181    Difference
                              .00656891     -.00087218    Relative difference
─────────────────────────────────────────────────────────
union:         -1.5843944     -1.5909857     -1.5843375
 1.south                     -.00659135      .00005689    Difference
                              .00416017     -.00003591    Relative difference
─────────────────────────────────────────────────────────
union:         -.00398535     -.00397811     -.00400181
    year                      7.237e-06     -.00001646    Difference
                             -.00181578      .00412982    Relative difference
─────────────────────────────────────────────────────────
union:          .01340169      .01344457      .01340388
1.south#c.~r                  .00004288      2.193e-06    Difference
                              .00319946       .0001636    Relative difference
─────────────────────────────────────────────────────────
union:         -1.6682022     -1.6757524     -1.6665327
   _cons                     -.00755024      .00166948    Difference
                              .00452597     -.00100077    Relative difference
─────────────────────────────────────────────────────────
/:              .61036163      .61780789      .60974814
 lnsig2u                      .00744626     -.00061349    Difference
                              .01219976     -.00100513    Relative difference
─────────────────────────────────────────────────────────
```

The results obtained for 12 quadrature points were closer to the results for 16 points than to the results for eight points. Although the relative and absolute differences are a bit larger than we would like, they are not large. We can increase the number of quadrature points with the intpoints() option; if we choose intpoints(20) and do another quadchk we will get acceptable results, with relative differences around 0.01%.

This is not the case if we use nonadaptive quadrature. Then the results we obtain are

```
. xtprobit union age grade i.not_smsa south##c.year, intmethod(ghermite)
Fitting comparison model:
Iteration 0:  Log likelihood =  -13864.23
Iteration 1:  Log likelihood = -13545.541
Iteration 2:  Log likelihood = -13544.385
Iteration 3:  Log likelihood = -13544.385
Fitting full model:
rho =  0.0    Log likelihood = -13544.385
rho =  0.1    Log likelihood = -12237.655
rho =  0.2    Log likelihood = -11590.282
rho =  0.3    Log likelihood = -11211.185
rho =  0.4    Log likelihood = -10981.319
rho =  0.5    Log likelihood = -10852.793
rho =  0.6    Log likelihood = -10808.759
rho =  0.7    Log likelihood =  -10865.57
Iteration 0:  Log likelihood = -10808.759
Iteration 1:  Log likelihood = -10594.349
Iteration 2:  Log likelihood = -10560.913
Iteration 3:  Log likelihood = -10560.876
Iteration 4:  Log likelihood = -10560.876
Random-effects probit regression                 Number of obs    = 26,200
Group variable: idcode                           Number of groups =  4,434
Random effects u_i ~ Gaussian                    Obs per group:
                                                              min =      1
                                                              avg =    5.9
                                                              max =     12
Integration method: ghermite                     Integration pts. =     12
                                                 Wald chi2(6)     = 218.99
Log likelihood = -10560.876                      Prob > chi2      = 0.0000

------------------------------------------------------------------------------
       union | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
         age |   .0093488   .0083385     1.12   0.262    -.0069945     .025692
       grade |   .0488014   .0101168     4.82   0.000     .0289728      .06863
  1.not_smsa |  -.1364862   .0462831    -2.95   0.003    -.2271995    -.045773
     1.south |  -1.592711   .3576715    -4.45   0.000    -2.293734   -.8916877
        year |  -.0053723   .0087219    -0.62   0.538    -.0224668    .0117223
             |
south#c.year |
           1 |   .0136764   .0044532     3.07   0.002     .0049482    .0224046
             |
       _cons |  -1.575539   .4639881    -3.40   0.001    -2.484939   -.6661388
-------------+----------------------------------------------------------------
    /lnsig2u |   .5615976   .0432021                       .476923    .6462722
-------------+----------------------------------------------------------------
     sigma_u |   1.324187   .0286038                      1.269295    1.381453
         rho |   .6368221   .0099918                       .617021    .6561699
------------------------------------------------------------------------------
LR test of rho=0: chibar2(01) = 5967.02                Prob >= chibar2 = 0.000
```

We now check the stability of the quadrature technique for this nonadaptive quadrature model. We expect it to be less stable.

```
. quadchk, nooutput
Refitting model intpoints() =  8
Refitting model intpoints() = 16
                         Quadrature check

                   Fitted      Comparison     Comparison
               quadrature      quadrature     quadrature
                12 points        8 points      16 points
─────────────────────────────────────────────────────────
Log            -10560.876      -10574.239     -10555.792
likelihood                     -13.362535      5.0839579   Difference
                                .00126529     -.0004814    Relative difference
─────────────────────────────────────────────────────────
union:          .00934876       .01264615      .00731888
     age                        .0032974      -.00202987   Difference
                                .35270966     -.21712744   Relative difference
─────────────────────────────────────────────────────────
union:          .04880139       .05710089      .04432417
   grade                        .00829951     -.00447722   Difference
                                .17006703     -.09174372   Relative difference
─────────────────────────────────────────────────────────
union:         -.13648624      -.13327724     -.14094541
1.not_smsa                       .003209      -.00445917   Difference
                               -.0235115       .03267123   Relative difference
─────────────────────────────────────────────────────────
union:          -1.592711      -1.5275627     -1.6059143
 1.south                        .06514823     -.01320331   Difference
                               -.04090399      .00828983   Relative difference
─────────────────────────────────────────────────────────
union:         -.00537226      -.00867673     -.00307042
    year                       -.00330447      .00230184   Difference
                                .61509968     -.4284678    Relative difference
─────────────────────────────────────────────────────────
union:          .01367641       .01278071      .01369009
1.south#c.~r                   -.0008957       .00001368   Difference
                               -.06549266      .00100054   Relative difference
─────────────────────────────────────────────────────────
union:         -1.5755388      -1.4888646     -1.6505526
   _cons                        .08667418     -.0750138    Difference
                               -.0550124       .04761152   Relative difference
─────────────────────────────────────────────────────────
/:              .56159763       .49290978      .58068904
 lnsig2u                       -.06868786      .0190914    Difference
                               -.12230795      .03399481   Relative difference
─────────────────────────────────────────────────────────
```

Once again, the results obtained for 12 quadrature points were closer to the results for 16 points than to the results for eight points. However, here the convergence point seems to be sensitive to the number of quadrature points, so we should not trust these results. We should increase the number of quadrature points with the `intpoints()` option and then use `quadchk` again. We should not use the results of a random-effects specification when there is evidence that the numeric technique for calculating the model is not stable (as shown by `quadchk`).

Generally, the relative differences in the coefficients should not change by more than 1% if the quadrature technique is stable. See [XT] **quadchk** for details. Increasing the number of quadrature points can often improve the stability, and for models with high `rho` we may need many. We can also switch between adaptive and nonadaptive quadrature. As a rule, adaptive quadrature, which is the default integration method, is much more flexible and robust.

Because the `xtprobit, re` likelihood function is calculated by Gauss–Hermite quadrature, on large problems the computations can be slow. Computation time is roughly proportional to the number of points used for the quadrature.

❑

▷ Example 2: Equal-correlation model

As an alternative to the random-effects specification, we can fit an equal-correlation probit model:

```
. xtprobit union age grade i.not_smsa south##c.year, pa
Iteration 1:  Tolerance = .12544249
Iteration 2:  Tolerance = .0034686
Iteration 3:  Tolerance = .00017448
Iteration 4:  Tolerance = 8.382e-06
Iteration 5:  Tolerance = 3.997e-07

GEE population-averaged model                      Number of obs    = 26,200
Group variable: idcode                             Number of groups =  4,434
Family: Binomial                                   Obs per group:
Link:   Probit                                                  min =      1
Correlation: exchangeable                                       avg =    5.9
                                                                max =     12
                                                   Wald chi2(6)     = 242.57
Scale parameter = 1                                Prob > chi2      = 0.0000

------------------------------------------------------------------------------
       union | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
         age |   .0089699   .0053208     1.69   0.092    -.0014586    .0193985
       grade |   .0333174   .0062352     5.34   0.000     .0210966    .0455382
  1.not_smsa |  -.0715717    .027543    -2.60   0.009    -.1255551   -.0175884
     1.south |  -1.017368    .207931    -4.89   0.000    -1.424905   -.6098308
        year |  -.0062708   .0055314    -1.13   0.257    -.0171122    .0045706
             |
south#c.year |
           1 |   .0086294     .00258     3.34   0.001     .0035727     .013686
             |
       _cons |  -.8670997    .294771    -2.94   0.003     -1.44484   -.2893592
------------------------------------------------------------------------------
```

◁

▷ Example 3: Population-averaged model

In example 3 of [R] **probit**, we showed the above results and compared them with `probit, vce(cluster id)`. `xtprobit` with the `pa` option allows a `vce(robust)` option, so we can obtain the population-averaged probit estimator with the robust variance calculation by typing

```
. xtprobit union age grade i.not_smsa south##c.year, pa vce(robust) nolog
GEE population-averaged model                    Number of obs     = 26,200
Group variable: idcode                           Number of groups =  4,434
Family: Binomial                                 Obs per group:
Link:   Probit                                                min =      1
Correlation: exchangeable                                     avg =    5.9
                                                              max =     12
                                                 Wald chi2(6)      = 156.33
Scale parameter = 1                              Prob > chi2       = 0.0000
                                    (Std. err. adjusted for clustering on idcode)

------------------------------------------------------------------------------
             |              Semirobust
       union | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
         age |   .0089699   .0051169     1.75   0.080     -.001059    .0189988
       grade |   .0333174   .0076425     4.36   0.000     .0183383    .0482965
  1.not_smsa |  -.0715717   .0348659    -2.05   0.040    -.1399076   -.0032359
     1.south |  -1.017368   .3026981    -3.36   0.001    -1.610645   -.4240906
        year |  -.0062708   .0055745    -1.12   0.261    -.0171965    .0046549
             |
south#c.year |
           1 |   .0086294   .0037866     2.28   0.023     .0012078    .0160509
             |
       _cons |  -.8670997   .3243959    -2.67   0.008    -1.502904   -.2312955
------------------------------------------------------------------------------
```

These standard errors are similar to those shown for `probit, vce(cluster id)` in [R] **probit**.

◁

▷ Example 4: Random-effects model with stable quadrature

In a previous example, we showed how quadchk indicated that the quadrature technique was numerically unstable. Here we present an example in which the quadrature is stable.

In this example, we have (synthetic) data on whether workers complain to managers at fast-food restaurants. The covariates are age (in years of the worker), grade (years of schooling completed by the worker), south (equal to 1 if the restaurant is located in the South), tenure (the number of years spent on the job by the worker), gender (of the worker), race (of the worker), income (in thousands of dollars by the restaurant), genderm (gender of the manager), burger (equal to 1 if the restaurant specializes in hamburgers), and chicken (equal to 1 if the restaurant specializes in chicken). The model is given by

```
. use https://www.stata-press.com/data/r19/chicken

. xtprobit complain age grade south tenure gender race income genderm burger
> chicken, nolog

Random-effects probit regression                    Number of obs    =  2,763
Group variable: restaurant                          Number of groups =    500

Random effects u_i ~ Gaussian                       Obs per group:
                                                                 min =      3
                                                                 avg =    5.5
                                                                 max =      8

Integration method: mvaghermite                     Integration pts. =     12

                                                    Wald chi2(10)    = 126.59
Log likelihood = -1318.2088                         Prob > chi2      = 0.0000

------------------------------------------------------------------------------
    complain | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
         age |  -.0430409   .0130211    -3.31   0.001    -.0685617    -.01752
       grade |   .0330934   .0264572     1.25   0.211    -.0187618   .0849486
       south |      .1012   .0707196     1.43   0.152     -.037408   .2398079
      tenure |  -.0440079   .0987099    -0.45   0.656    -.2374758     .14946
      gender |   .3318499   .0601382     5.52   0.000     .2139812   .4497185
        race |   .3417901   .0382251     8.94   0.000     .2668703   .4167098
      income |  -.0022702   .0008885    -2.56   0.011    -.0040117  -.0005288
     genderm |   .0524577   .0706585     0.74   0.458    -.0860305   .1909459
      burger |   .0448931   .0956151     0.47   0.639    -.1425091   .2322953
     chicken |   .1904714   .0953067     2.00   0.046     .0036737   .3772691
       _cons |  -.2145311   .6240549    -0.34   0.731    -1.437656   1.008594
-------------+----------------------------------------------------------------
    /lnsig2u |  -1.704494   .2502057                     -2.194888  -1.214099
-------------+----------------------------------------------------------------
     sigma_u |   .4264557   .0533508                       .333723   .5449563
         rho |   .1538793   .0325769                      .1002105   .2289765
------------------------------------------------------------------------------
LR test of rho=0: chibar2(01) = 29.91                  Prob >= chibar2 = 0.000
```

Again we would like to check the stability of the quadrature technique of the model before interpreting the results. Given the estimate of ρ and the small size of the panels (between 3 and 8), we should find that the quadrature technique is numerically stable.

```
. quadchk, nooutput
Refitting model intpoints() =  8
Refitting model intpoints() = 16
                           Quadrature check

                  Fitted      Comparison     Comparison
              quadrature      quadrature     quadrature
               12 points        8 points      16 points
─────────────────────────────────────────────────────────
Log           -1318.2088      -1318.2088     -1318.2088
likelihood                    -2.002e-06     -1.194e-09   Difference
                               1.519e-09      9.061e-13   Relative difference
─────────────────────────────────────────────────────────
complain:     -.04304086      -.04304086     -.04304086
     age                      -3.896e-10     -2.625e-12   Difference
                               9.051e-09      6.100e-11   Relative difference
─────────────────────────────────────────────────────────
complain:       .0330934        .0330934       .0330934
   grade                       2.208e-11      1.867e-12   Difference
                               6.673e-10      5.642e-11   Relative difference
─────────────────────────────────────────────────────────
complain:      .10119998       .10119999      .10119998
   south                       2.369e-09      3.957e-11   Difference
                               2.341e-08      3.910e-10   Relative difference
─────────────────────────────────────────────────────────
complain:     -.04400789       -.0440079     -.04400789
  tenure                      -3.362e-09     -2.250e-11   Difference
                               7.640e-08      5.114e-10   Relative difference
─────────────────────────────────────────────────────────
complain:      .33184986       .33184986      .33184986
  gender                       3.190e-09      2.546e-11   Difference
                               9.612e-09      7.673e-11   Relative difference
─────────────────────────────────────────────────────────
complain:      .34179006       .34179007      .34179006
    race                       3.801e-09      2.990e-11   Difference
                               1.112e-08      8.749e-11   Relative difference
─────────────────────────────────────────────────────────
complain:     -.00227021      -.00227021     -.00227021
  income                      -4.468e-11     -9.252e-13   Difference
                               1.968e-08      4.075e-10   Relative difference
─────────────────────────────────────────────────────────
complain:      .05245769       .05245769      .05245769
 genderm                       1.963e-09      4.481e-11   Difference
                               3.742e-08      8.542e-10   Relative difference
─────────────────────────────────────────────────────────
complain:      .04489311       .04489311      .04489311
  burger                       4.173e-10      6.628e-12   Difference
                               9.296e-09      1.476e-10   Relative difference
─────────────────────────────────────────────────────────
complain:      .19047138       .19047139      .19047138
 chicken                       3.096e-09      4.916e-11   Difference
                               1.625e-08      2.581e-10   Relative difference
─────────────────────────────────────────────────────────
complain:     -.21453112      -.21453111     -.21453112
   _cons                       1.281e-08      2.682e-10   Difference
                              -5.972e-08     -1.250e-09   Relative difference
─────────────────────────────────────────────────────────
/:            -1.7044935      -1.7044934     -1.7044935
 lnsig2u                       1.255e-07     -4.135e-10   Difference
                              -7.365e-08      2.426e-10   Relative difference
─────────────────────────────────────────────────────────
```

The relative and absolute differences are all small between the default 12 quadrature points and the result with 16 points. We do not have any coefficients that have a large difference between the default 12 quadrature points and eight quadrature points.

We conclude that the quadrature technique is stable. Because the differences here are so small, we would plan on using and interpreting these results rather than trying to rerun with more quadrature points.

◁

Stored results

xtprobit, re stores the following in e():

Scalars

e(N)	number of observations
e(N_g)	number of groups
e(k)	number of parameters
e(k_aux)	number of auxiliary parameters
e(k_eq)	number of equations in e(b)
e(k_eq_model)	number of equations in overall model test
e(k_dv)	number of dependent variables
e(df_m)	model degrees of freedom
e(ll)	log likelihood
e(ll_0)	log likelihood, constant-only model
e(ll_c)	log likelihood, comparison model
e(chi2)	χ^2
e(chi2_c)	χ^2 for comparison test
e(N_clust)	number of clusters
e(rho)	ρ
e(sigma_u)	panel-level standard deviation
e(n_quad)	number of quadrature points
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(p)	p-value for model test
e(rank)	rank of e(V)
e(rank0)	rank of e(V) for constant-only model
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros

e(cmd)	xtprobit
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(ivar)	variable denoting groups
e(model)	re
e(wtype)	weight type
e(wexp)	weight expression
e(title)	title in estimation output
e(clustvar)	name of cluster variable
e(offset)	linear offset variable
e(chi2type)	Wald or LR; type of model χ^2 test
e(chi2_ct)	Wald or LR; type of model χ^2 test corresponding to e(chi2_c)
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(intmethod)	integration method
e(distrib)	Gaussian; the distribution of the random effect

e(opt)	type of optimization
e(which)	max or min; whether optimizer is to perform maximization or minimization
e(ml_method)	type of ml method
e(user)	name of likelihood-evaluator program
e(technique)	maximization technique
e(properties)	b V
e(predict)	program used to implement predict
e(marginsdefault)	default predict() specification for margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved
Matrices	
e(b)	coefficient vector
e(Cns)	constraints matrix
e(ilog)	iteration log
e(gradient)	gradient vector
e(V)	variance–covariance matrix of the estimators
e(V_modelbased)	model-based variance
Functions	
e(sample)	marks estimation sample

In addition to the above, the following is stored in r():

Matrices	
r(table)	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

xtprobit, pa stores the following in e():

Scalars	
e(N)	number of observations
e(N_g)	number of groups
e(df_m)	model degrees of freedom
e(chi2)	χ^2
e(p)	p-value for model test
e(df_pear)	degrees of freedom for Pearson χ^2
e(chi2_dev)	χ^2 test of deviance
e(chi2_dis)	χ^2 test of deviance dispersion
e(deviance)	deviance
e(dispers)	deviance dispersion
e(phi)	scale parameter
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(rank)	rank of e(V)
e(tol)	target tolerance
e(dif)	achieved tolerance
e(rc)	return code
Macros	
e(cmd)	xtgee
e(cmd2)	xtprobit
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(ivar)	variable denoting groups
e(tvar)	variable denoting time within groups
e(model)	pa

`e(family)`	`binomial`
`e(link)`	`probit`; link function
`e(corr)`	correlation structure
`e(scale)`	`x2`, `dev`, `phi`, or #; scale parameter
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(offset)`	linear offset variable
`e(chi2type)`	`Wald`; type of model χ^2 test
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. err.
`e(nmp)`	`nmp`, if specified
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(marginsnotok)`	predictions disallowed by `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`
Matrices	
`e(b)`	coefficient vector
`e(R)`	estimated working correlation matrix
`e(V)`	variance–covariance matrix of the estimators
`e(V_modelbased)`	model-based variance
Functions	
`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices	
`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

`xtprobit` reports the population-averaged results obtained by using `xtgee, family(binomial) link(probit)` to obtain estimates.

Assuming a normal distribution, $N(0, \sigma_\nu^2)$, for the random effects ν_i

$$\Pr(y_{i1}, \dots, y_{in_i} | \mathbf{x}_{i1}, \dots, \mathbf{x}_{in_i}) = \int_{-\infty}^{\infty} \frac{e^{-\nu_i^2/2\sigma_\nu^2}}{\sqrt{2\pi}\sigma_\nu} \left\{ \prod_{t=1}^{n_i} F(y_{it}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i) \right\} d\nu_i$$

where

$$F(y, z) = \begin{cases} \Phi(z) & \text{if } y \neq 0 \\ 1 - \Phi(z) & \text{otherwise} \end{cases}$$

where Φ is the cumulative normal distribution.

The panel-level likelihood l_i is given by

$$l_i = \int_{-\infty}^{\infty} \frac{e^{-\nu_i^2/2\sigma_\nu^2}}{\sqrt{2\pi}\sigma_\nu} \left\{ \prod_{t=1}^{n_i} F(y_{it}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i) \right\} d\nu_i$$

$$\equiv \int_{-\infty}^{\infty} g(y_{it}, x_{it}, \nu_i) d\nu_i$$

This integral can be approximated with M-point Gauss–Hermite quadrature

$$\int_{-\infty}^{\infty} e^{-x^2} h(x) dx \approx \sum_{m=1}^{M} w_m^* h(a_m^*)$$

This is equivalent to

$$\int_{-\infty}^{\infty} f(x) dx \approx \sum_{m=1}^{M} w_m^* \exp\left\{(a_m^*)^2\right\} f(a_m^*)$$

where the w_m^* denote the quadrature weights and the a_m^* denote the quadrature abscissas. The log likelihood, L, is the sum of the logs of the panel-level likelihoods l_i.

The default approximation of the log likelihood is by adaptive Gauss–Hermite quadrature, which approximates the panel-level likelihood with

$$l_i \approx \sqrt{2}\hat{\sigma}_i \sum_{m=1}^{M} w_m^* \exp\left\{(a_m^*)^2\right\} g(y_{it}, x_{it}, \sqrt{2}\hat{\sigma}_i a_m^* + \hat{\mu}_i)$$

where $\hat{\sigma}_i$ and $\hat{\mu}_i$ are the adaptive parameters for panel i. Therefore, with the definition of $g(y_{it}, x_{it}, \nu_i)$, the total log likelihood is approximated by

$$L \approx \sum_{i=1}^{n} w_i \log\Bigg[\sqrt{2}\hat{\sigma}_i \sum_{m=1}^{M} w_m^* \exp\{(a_m^*)^2\} \frac{\exp\{-(\sqrt{2}\hat{\sigma}_i a_m^* + \hat{\mu}_i)^2/2\sigma_\nu^2\}}{\sqrt{2\pi}\sigma_\nu}$$

$$\prod_{t=1}^{n_i} F(y_{it}, x_{it}\boldsymbol{\beta} + \sqrt{2}\hat{\sigma}_i a_m^* + \hat{\mu}_i)\Bigg]$$

where w_i is the user-specified weight for panel i; if no weights are specified, $w_i = 1$.

The default method of adaptive Gauss–Hermite quadrature is to calculate the posterior mean and variance and use those parameters for $\widehat{\mu}_i$ and $\widehat{\sigma}_i$ by following the method of Naylor and Smith (1982), further discussed in Skrondal and Rabe-Hesketh (2004). We start with $\widehat{\sigma}_{i,0} = 1$ and $\widehat{\mu}_{i,0} = 0$, and the posterior means and variances are updated in the kth iteration. That is, at the kth iteration of the optimization for l_i, we use

$$l_{i,k} \approx \sum_{m=1}^{M} \sqrt{2}\widehat{\sigma}_{i,k-1} w_m^* \exp\{a_m^*)^2\} g(y_{it}, x_{it}, \sqrt{2}\widehat{\sigma}_{i,k-1} a_m^* + \widehat{\mu}_{i,k-1})$$

Letting

$$\tau_{i,m,k-1} = \sqrt{2}\widehat{\sigma}_{i,k-1} a_m^* + \widehat{\mu}_{i,k-1}$$

$$\widehat{\mu}_{i,k} = \sum_{m=1}^{M} (\tau_{i,m,k-1}) \frac{\sqrt{2}\widehat{\sigma}_{i,k-1} w_m^* \exp\{(a_m^*)^2\} g(y_{it}, x_{it}, \tau_{i,m,k-1})}{l_{i,k}}$$

and

$$\widehat{\sigma}_{i,k} = \sum_{m=1}^{M} (\tau_{i,m,k-1})^2 \frac{\sqrt{2}\widehat{\sigma}_{i,k-1} w_m^* \exp\{(a_m^*)^2\} g(y_{it}, x_{it}, \tau_{i,m,k-1})}{l_{i,k}} - (\widehat{\mu}_{i,k})^2$$

and this is repeated until $\widehat{\mu}_{i,k}$ and $\widehat{\sigma}_{i,k}$ have converged for this iteration of the maximization algorithm. This adaptation is applied on every iteration until the log-likelihood change from the preceding iteration is less than a relative difference of 1e–6; after this, the quadrature parameters are fixed.

The log likelihood can also be calculated by nonadaptive Gauss–Hermite quadrature, the `intmethod(ghermite)` option, where $\rho = \sigma_\nu^2/(\sigma_\nu^2 + 1)$:

$$\begin{aligned} L &= \sum_{i=1}^{n} w_i \log\Big\{ \Pr(y_{i1}, \dots, y_{in_i} | \mathbf{x}_{i1}, \dots, \mathbf{x}_{in_i}) \Big\} \\ &\approx \sum_{i=1}^{n} w_i \log \left[\frac{1}{\sqrt{\pi}} \sum_{m=1}^{M} w_m^* \prod_{t=1}^{n_i} F\left\{ y_{it}, \mathbf{x}_{it}\boldsymbol{\beta} + a_m^* \left(\frac{2\rho}{1-\rho} \right)^{1/2} \right\} \right] \end{aligned}$$

Both quadrature formulas require that the integrated function be well approximated by a polynomial of degree equal to the number of quadrature points. The number of periods (panel size) can affect whether

$$\prod_{t=1}^{n_i} F(y_{it}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i)$$

is well approximated by a polynomial. As panel size and ρ increase, the quadrature approximation can become less accurate. For large ρ, the random-effects model can also become unidentified. Adaptive quadrature gives better results for correlated data and large panels than nonadaptive quadrature; however, we recommend that you use the `quadchk` command (see [XT] **quadchk**) to verify the quadrature approximation used in this command, whichever approximation you choose.

xtprobit, re and the robust VCE estimator

Specifying `vce(robust)` or `vce(cluster` *clustvar*`)` causes the Huber/White/sandwich VCE estimator to be calculated for the coefficients estimated in this regression. See [P] **_robust**, particularly *Introduction* and *Methods and formulas*. Wooldridge (2020) and Arellano (2003) discuss this application of the Huber/White/sandwich VCE estimator. As discussed by Wooldridge (2020), Stock and Watson (2008), and Arellano (2003), specifying `vce(robust)` is equivalent to specifying `vce(cluster` *panelvar*`)`, where *panelvar* is the variable that identifies the panels.

Clustering on the panel variable produces a consistent VCE estimator when the disturbances are not identically distributed over the panels or there is serial correlation in ϵ_{it}.

The cluster–robust VCE estimator requires that there are many clusters and the disturbances are uncorrelated across the clusters. The panel variable must be nested within the cluster variable because of the within-panel correlation that is generally induced by the random-effects transform when there is heteroskedasticity or within-panel serial correlation in the idiosyncratic errors.

References

Arellano, M. 2003. *Panel Data Econometrics*. Oxford: Oxford University Press. https://doi.org/10.1093/0199245282.001.0001.

Baltagi, B. H. 2009. *A Companion to Econometric Analysis of Panel Data*. Chichester, UK: Wiley.

———. 2013. *Econometric Analysis of Panel Data*. 5th ed. Chichester, UK: Wiley.

Conway, M. R. 1990. A random effects model for binary data. *Biometrics* 46: 317–328. https://doi.org/10.2307/2531437.

Cruz-Gonzalez, M., I. Fernández-Val, and M. Weidner. 2017. Bias corrections for probit and logit models with two-way fixed effects. *Stata Journal* 17: 517–545.

Grotti, R., and G. Cutuli. 2018. xtpdyn: A community-contributed command for fitting dynamic random-effects probit models with unobserved heterogeneity. *Stata Journal* 18: 844–862.

Guilkey, D. K., and J. L. Murphy. 1993. Estimation and testing in the random effects probit model. *Journal of Econometrics* 59: 301–317. https://doi.org/10.1016/0304-4076(93)90028-4.

Liang, K.-Y., and S. L. Zeger. 1986. Longitudinal data analysis using generalized linear models. *Biometrika* 73: 13–22. https://doi.org/10.1093/biomet/73.1.13.

Naylor, J. C., and A. F. M. Smith. 1982. Applications of a method for the efficient computation of posterior distributions. *Journal of the Royal Statistical Society, C ser.*, 31: 214–225. https://doi.org/10.2307/2347995.

Neuhaus, J. M. 1992. Statistical methods for longitudinal and clustered designs with binary responses. *Statistical Methods in Medical Research* 1: 249–273. https://doi.org/10.1177/096228029200100303.

Neuhaus, J. M., J. D. Kalbfleisch, and W. W. Hauck. 1991. A comparison of cluster-specific and population-averaged approaches for analyzing correlated binary data. *International Statistical Review* 59: 25–35. https://doi.org/10.2307/1403572.

Pendergast, J. F., S. J. Gange, M. A. Newton, M. J. Lindstrom, M. Palta, and M. R. Fisher. 1996. A survey of methods for analyzing clustered binary response data. *International Statistical Review* 64: 89–118. https://doi.org/10.2307/1403425.

Plum, A. 2016. bireprob: An estimator for bivariate random-effects probit models. *Stata Journal* 16: 96–111.

Skrondal, A., and S. Rabe-Hesketh. 2004. *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models*. Boca Raton, FL: Chapman and Hall/CRC.

Stewart, M. B. 2006. Maximum simulated likelihood estimation of random-effects dynamic probit models with autocorrelated errors. *Stata Journal* 6: 256–272.

Stock, J. H., and M. W. Watson. 2008. Heteroskedasticity-robust standard errors for fixed effects panel data regression. *Econometrica* 76: 155–174. https://doi.org/10.1111/j.0012-9682.2008.00821.x.

Wooldridge, J. M. 2020. *Introductory Econometrics: A Modern Approach*. 7th ed. Boston: Cengage.

Also see

[XT] **xtprobit postestimation** — Postestimation tools for xtprobit

[XT] **quadchk** — Check sensitivity of quadrature approximation

[XT] **xtcloglog** — Random-effects and population-averaged cloglog models

[XT] **xteprobit** — Extended random-effects probit regression

[XT] **xtgee** — GEE population-averaged panel-data models

[XT] **xtlogit** — Fixed-effects, random-effects, and population-averaged logit models

[XT] **xtset** — Declare data to be panel data

[BAYES] **bayes: xtprobit** — Bayesian random-effects probit model

[ME] **meprobit** — Multilevel mixed-effects probit regression

[MI] **Estimation** — Estimation commands for use with mi estimate

[R] **probit** — Probit regression

[U] 20 Estimation and postestimation commands

xtprobit postestimation — Postestimation tools for xtprobit

Postestimation commands   predict   margins   Remarks and examples   Also see

Postestimation commands

The following postestimation commands are available after `xtprobit`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of parameters
* `estat ic`	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC, respectively)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
† `forecast`	dynamic forecasts and simulations
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
* `lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	linear predictions and their SEs, probabilities
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of parameters
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

*`estat ic` and `lrtest` are not appropriate after `xtprobit, pa`.

†`forecast` is not appropriate with `mi` estimation results.

predict

Description for predict

predict creates a new variable containing predictions such as linear predictions, probabilities, standard errors, and the equation-level score.

Menu for predict

Statistics > Postestimation

Syntax for predict

Random-effects model

predict [*type*] *newvar* [*if*] [*in*] [, *RE_statistic* nooffset]

Population-averaged model

predict [*type*] *newvar* [*if*] [*in*] [, *PA_statistic* nooffset]

RE_statistic	Description
Main	
xb	linear prediction; the default
pr	marginal probability of a positive outcome
pu0	probability of a positive outcome
stdp	standard error of the linear prediction

PA_statistic	Description
Main	
mu	probability of *depvar*; considers the offset(); the default
rate	probability of *depvar*
xb	linear prediction
stdp	standard error of the linear prediction
score	first derivative of the log likelihood with respect to $\mathbf{x}_{it}\boldsymbol{\beta}$

These statistics are available both in and out of sample; type predict ... if e(sample) ... if wanted only for the estimation sample.

Options for predict

Main

`xb` calculates the linear prediction. This is the default for the random-effects model.

`pr` calculates the probability of a positive outcome that is marginal with respect to the random effect, which means that the probability is calculated by integrating the prediction function with respect to the random effect over its entire support.

`pu0` calculates the probability of a positive outcome, assuming that the random effect for that observation's panel is zero ($\nu_i = 0$). This probability may not be similar to the proportion of observed outcomes in the group.

`mu` and `rate` both calculate the predicted probability of *depvar*. `mu` takes into account the `offset()`, and `rate` ignores those adjustments. `mu` and `rate` are equivalent if you did not specify `offset()`. `mu` is the default for the population-averaged model.

`stdp` calculates the standard error of the linear prediction.

`score` calculates the equation-level score, $u_{it} = \partial \ln L(\mathbf{x}_{it}\boldsymbol{\beta})/\partial(\mathbf{x}_{it}\boldsymbol{\beta})$.

`nooffset` is relevant only if you specified `offset(`*varname*`)` for `xtprobit`. It modifies the calculations made by `predict` so that they ignore the offset variable; the linear prediction is treated as $\mathbf{x}_{it}\boldsymbol{\beta}$ rather than $\mathbf{x}_{it}\boldsymbol{\beta} + \text{offset}_{it}$.

margins

Description for margins

`margins` estimates margins of response for linear predictions and probabilities.

Menu for margins

Statistics > Postestimation

Syntax for margins

`margins` [*marginlist*] [, *options*]

`margins` [*marginlist*] , `predict(`*statistic* ...`)` [`predict(`*statistic* ...`)` ...] [*options*]

Random-effects model

statistic	Description
`pr`	marginal probability of a positive outcome; the default
`pu0`	probability of a positive outcome
`xb`	linear prediction
`stdp`	not allowed with `margins`

Population-averaged model

statistic	Description
`mu`	probability of *depvar*; considers the `offset()`; the default
`rate`	probability of *depvar*
`xb`	linear prediction
`stdp`	not allowed with `margins`
`score`	not allowed with `margins`

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.

For the full syntax, see [R] **margins**.

Remarks and examples

▷ Example 1: Calculating average marginal effects

In example 2 of [XT] **xtprobit**, we fit a population-averaged model of union status on the woman's age and level of schooling, whether she lived in an urban area, whether she lived in the south, and the year observed. Here we compute the average marginal effects from that fitted model on the probability of being in a union.

```
. use https://www.stata-press.com/data/r19/union
(NLS Women 14-24 in 1968)
. xtprobit union age grade i.not_smsa south##c.year, pa
  (output omitted)
. margins, dydx(*)
Average marginal effects                              Number of obs = 26,200
Model VCE: Conventional

Expression: Pr(union != 0), predict()
dy/dx wrt:  age grade 1.not_smsa 1.south year
```

	dy/dx	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
age	.0025337	.0015035	1.69	0.092	-.0004132	.0054805
grade	.0094109	.0017566	5.36	0.000	.005968	.0128537
1.not_smsa	-.0199744	.0075879	-2.63	0.008	-.0348464	-.0051023
1.south	-.0910805	.0073315	-12.42	0.000	-.10545	-.076711
year	-.000938	.0015413	-0.61	0.543	-.0039589	.0020828

```
Note: dy/dx for factor levels is the discrete change from the base level.
```

On average, not living in a metropolitan area (not_smsa $= 1$) lowers the probability of being in a union by about two percentage points.

◁

Also see

[XT] **xtprobit** — Random-effects and population-averaged probit models

[U] 20 Estimation and postestimation commands

xtrc — Random-coefficients model

Description Quick start Menu Syntax Options
Remarks and examples Stored results Methods and formulas References Also see

Description

xtrc fits the Swamy (1970) random-coefficients linear regression model, which does not impose the assumption of constant parameters across panels. Average coefficient estimates are reported by default, but panel-specific coefficients may be requested.

Quick start

Random-coefficients regression of y on x1 and x2 using xtset data

```
xtrc y x1 x2
```

Same as above, but report panel-specific best linear predictors

```
xtrc y x1 x2, betas
```

Multiple-imputation estimates of random-coefficients regression using mi xtset data

```
mi estimate: xtrc y x
```

Menu

Statistics > Longitudinal/panel data > Random-coefficients regression by GLS

Syntax

`xtrc` *depvar indepvars* [*if*] [*in*] [, *options*]

options	Description
Main	
`noconstant`	suppress constant term
`offset(`*varname*`)`	include *varname* in model with coefficient constrained to 1
SE	
`vce(`*vcetype*`)`	*vcetype* may be `conventional`, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`betas`	display group-specific best linear predictors
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
`coeflegend`	display legend instead of statistics

A panel variable must be specified; use `xtset`; see [XT] **xtset**.

indepvars may contain factor variables; see [U] **11.4.3 Factor variables**.

`by`, `collect`, `mi estimate`, and `statsby` are allowed; see [U] **11.1.10 Prefix commands**.

`vce(bootstrap)` and `vce(jackknife)` are not allowed with the `mi estimate` prefix; see [MI] **mi estimate**.

`coeflegend` does not appear in the dialog box.

See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Main

`noconstant`, `offset(`*varname*`)`; see [R] **Estimation options**

SE

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`conventional`) and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [XT] ***vce_options***.

`vce(conventional)`, the default, uses the conventionally derived variance estimator for generalized least-squares regression.

Reporting

`level(#)`; see [R] **Estimation options**.

`betas` requests that the group-specific best linear predictors also be displayed.

display_options: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(`*style*`)`, `cformat(%`*fmt*`)`, `pformat(%`*fmt*`)`, `sformat(%`*fmt*`)`, and `nolstretch`; see [R] **Estimation options**.

The following option is available with xtrc but is not shown in the dialog box:

coeflegend; see [R] **Estimation options**.

Remarks and examples

In random-coefficients models, we wish to treat the parameter vector as a realization (in each panel) of a stochastic process. xtrc fits the Swamy (1970) random-coefficients model, which is suitable for linear regression of panel data. See Greene (2012, chap. 11) and Poi (2003) for more information about this and other panel-data models.

▷ Example 1

Greene (2012, 1112) reprints data from a classic study of investment demand by Grunfeld and Griliches (1960). In [XT] **xtgls**, we use this dataset to illustrate many of the possible models that may be fit with the xtgls command. Although the models included in the xtgls command offer considerable flexibility, they all assume that there is no parameter variation across firms (the cross-sectional units).

To take a first look at the assumption of parameter constancy, we should reshape our data so that we may fit a simultaneous-equation model with sureg; see [R] **sureg**. Because there are only five panels here, this is not too difficult.

```
. use https://www.stata-press.com/data/r19/invest2

. reshape wide invest market stock, i(time) j(company)
(j = 1 2 3 4 5)

Data                               Long   ->   Wide
-----------------------------------------------------------------------------

Number of observations              100   ->   20
Number of variables                   5   ->   16
j variable (5 values)           company   ->   (dropped)
xij variables:
                                 invest   ->   invest1 invest2 ... invest5
                                 market   ->   market1 market2 ... market5
                                  stock   ->   stock1 stock2 ... stock5
-----------------------------------------------------------------------------
```

```
. sureg (invest1 market1 stock1) (invest2 market2 stock2)
> (invest3 market3 stock3) (invest4 market4 stock4) (invest5 market5 stock5)

Seemingly unrelated regression

Equation             Obs   Params         RMSE  "R-squared"      chi2   P>chi2

invest1               20        2     84.94729      0.9207     261.32   0.0000
invest2               20        2     12.36322      0.9119     207.21   0.0000
invest3               20        2     26.46612      0.6876      46.88   0.0000
invest4               20        2     9.742303      0.7264      59.15   0.0000
invest5               20        2     95.85484      0.4220      14.97   0.0006


             Coefficient  Std. err.      z    P>|z|     [95% conf. interval]

invest1
     market1    .120493   .0216291     5.57   0.000     .0781007    .1628853
      stock1   .3827462    .032768    11.68   0.000      .318522    .4469703
       _cons  -162.3641   89.45922    -1.81   0.070    -337.7009    12.97279

invest2
     market2   .0695456   .0168975     4.12   0.000     .0364271    .1026641
      stock2   .3085445   .0258635    11.93   0.000     .2578529    .3592362
       _cons   .5043112   11.51283     0.04   0.965    -22.06042    23.06904

invest3
     market3   .0372914   .0122631     3.04   0.002     .0132561    .0613268
      stock3    .130783   .0220497     5.93   0.000     .0875663    .1739997
       _cons  -22.43892   25.51859    -0.88   0.379    -72.45443    27.57659

invest4
     market4   .0570091   .0113623     5.02   0.000     .0347395    .0792788
      stock4   .0415065   .0412016     1.01   0.314    -.0392472    .1222602
       _cons   1.088878   6.258805     0.17   0.862    -11.17815    13.35591

invest5
     market5   .1014782   .0547837     1.85   0.064    -.0058958    .2088523
      stock5   .3999914   .1277946     3.13   0.002     .1495186    .6504642
       _cons   85.42324   111.8774     0.76   0.445    -133.8525    304.6989
```

Here we instead fit a random-coefficients model:

```
. use https://www.stata-press.com/data/r19/invest2, clear
. xtrc invest market stock
Random-coefficients regression                  Number of obs     =    100
Group variable: company                         Number of groups  =      5
Time variable: time                             Obs per group:
                                                              min =     20
                                                              avg =   20.0
                                                              max =     20
                                                Wald chi2(2)      =  17.55
                                                Prob > chi2       = 0.0002

------------------------------------------------------------------------------
      invest | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      market |   .0807646   .0250829     3.22   0.001     .0316031    .1299261
       stock |   .2839885   .0677899     4.19   0.000     .1511229    .4168542
       _cons |  -23.58361   34.55547    -0.68   0.495    -91.31108    44.14386
------------------------------------------------------------------------------
Test of parameter constancy: chi2(12) = 603.99           Prob > chi2 = 0.0000
```

Just as the results of our simultaneous-equation model do not support the assumption of parameter constancy, the test included with the random-coefficients model also indicates that the assumption is not valid for these data. With large panel datasets, we would not want to take the time to look at a simultaneous-equations model (aside from the fact that our doing so was subjective).

◁

Stored results

xtrc stores the following in e():

Scalars

e(N)	number of observations
e(N_g)	number of groups
e(df_m)	model degrees of freedom
e(chi2)	χ^2
e(chi2_c)	χ^2 for comparison test
e(df_chi2c)	degrees of freedom for comparison χ^2 test
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(rank)	rank of e(V)

Macros

e(cmd)	xtrc
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(ivar)	variable denoting groups
e(tvar)	variable denoting time within groups
e(title)	title in estimation output
e(offset)	linear offset variable
e(chi2type)	Wald; type of model χ^2 test
e(vce)	*vcetype* specified in vce()
e(properties)	b V
e(predict)	program used to implement predict
e(marginsnotok)	predictions disallowed by margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved

Matrices

`e(b)`	coefficient vector
`e(Sigma)`	$\widehat{\mathbf{\Sigma}}$ matrix
`e(beta_ps)`	matrix of best linear predictors
`e(V)`	variance–covariance matrix of the estimators
`e(V_ps)`	matrix of variances for the best linear predictors; row i contains vec of variance matrix for group i predictor

Functions

`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

In a random-coefficients model, the parameter heterogeneity is treated as stochastic variation. Assume that we write

$$\mathbf{y}_i = \mathbf{X}_i \boldsymbol{\beta}_i + \boldsymbol{\epsilon}_i$$

where $i = 1, \ldots, m$, and $\boldsymbol{\beta}_i$ is the coefficient vector ($k \times 1$) for the ith cross-sectional unit, such that

$$\boldsymbol{\beta}_i = \boldsymbol{\beta} + \boldsymbol{\nu}_i \qquad E(\boldsymbol{\nu}_i) = \mathbf{0} \qquad E(\boldsymbol{\nu}_i \boldsymbol{\nu}_i') = \mathbf{\Sigma}$$

Our goal is to find $\widehat{\boldsymbol{\beta}}$ and $\widehat{\mathbf{\Sigma}}$.

The derivation of the estimator assumes that the cross-sectional specific coefficient vector $\boldsymbol{\beta}_i$ is the outcome of a random process with mean vector $\boldsymbol{\beta}$ and covariance matrix $\mathbf{\Sigma}$,

$$\mathbf{y}_i = \mathbf{X}_i \boldsymbol{\beta}_i + \boldsymbol{\epsilon}_i = \mathbf{X}_i (\boldsymbol{\beta} + \boldsymbol{\nu}_i) + \boldsymbol{\epsilon}_i = \mathbf{X}_i \boldsymbol{\beta} + (\mathbf{X}_i \boldsymbol{\nu}_i + \boldsymbol{\epsilon}_i) = \mathbf{X}_i \boldsymbol{\beta} + \boldsymbol{\omega}_i$$

where $E(\boldsymbol{\omega}_i) = \mathbf{0}$ and

$$E(\boldsymbol{\omega}_i \boldsymbol{\omega}_i') = E\Big\{(\mathbf{X}_i \boldsymbol{\nu}_i + \boldsymbol{\epsilon}_i)(\mathbf{X}_i \boldsymbol{\nu}_i + \boldsymbol{\epsilon}_i)'\Big\} = E(\boldsymbol{\epsilon}_i \boldsymbol{\epsilon}_i') + \mathbf{X}_i E(\boldsymbol{\nu}_i \boldsymbol{\nu}_i') \mathbf{X}_i' = \sigma_i^2 \mathbf{I} + \mathbf{X}_i \mathbf{\Sigma} \mathbf{X}_i' = \mathbf{\Pi}_i$$

Stacking the m equations, we have

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\omega}$$

where $\mathbf{\Pi} \equiv E(\boldsymbol{\omega}\boldsymbol{\omega}')$ is a block diagonal matrix with $\mathbf{\Pi}_i$, $i = 1 \ldots m$, along the main diagonal and zeros elsewhere. The GLS estimator of $\widehat{\boldsymbol{\beta}}$ is then

$$\widehat{\boldsymbol{\beta}} = \left(\sum_i \mathbf{X}_i' \mathbf{\Pi}_i^{-1} \mathbf{X}_i \right)^{-1} \sum_i \mathbf{X}_i' \mathbf{\Pi}_i^{-1} \mathbf{y}_i = \sum_{i=1}^m \mathbf{W}_i \mathbf{b}_i$$

where

$$\mathbf{W}_i = \left\{ \sum_{i=1}^m (\mathbf{\Sigma} + \mathbf{V}_i)^{-1} \right\}^{-1} (\mathbf{\Sigma} + \mathbf{V}_i)^{-1}$$

$\mathbf{b}_i = (\mathbf{X}_i'\mathbf{X}_i)^{-1}\mathbf{X}_i'\mathbf{y}_i$ and $\mathbf{V}_i = \sigma_i^2(\mathbf{X}_i'\mathbf{X}_i)^{-1}$, showing that the resulting GLS estimator is a matrix-weighted average of the panel-specific OLS estimators. The variance of $\widehat{\boldsymbol\beta}$ is

$$\text{Var}(\widehat{\boldsymbol\beta}) = \sum_{i=1}^{m}(\boldsymbol\Sigma + \mathbf{V}_i)^{-1}$$

To calculate the above estimator $\widehat{\boldsymbol\beta}$ for the unknown $\boldsymbol\Sigma$ and $\mathbf{V}_i$ parameters, we use the two-step approach suggested by Swamy (1970):

$$\begin{aligned}
\mathbf{b}_i &= \text{OLS panel-specific estimator}\\
\widehat{\sigma}_i^2 &= \frac{\widehat{\boldsymbol\epsilon}_i'\widehat{\boldsymbol\epsilon}_i}{n_i - k}\\
\widehat{\mathbf{V}}_i &= \widehat{\sigma}_i^2(\mathbf{X}_i'\mathbf{X}_i)^{-1}\\
\overline{\mathbf{b}} &= \frac{1}{m}\sum_{i=1}^{m}\mathbf{b}_i\\
\widehat{\boldsymbol\Sigma} &= \frac{1}{m-1}\left(\sum_{i=1}^{m}\mathbf{b}_i\mathbf{b}_i' - m\overline{\mathbf{b}}\,\overline{\mathbf{b}}'\right) - \frac{1}{m}\sum_{i=1}^{m}\widehat{\mathbf{V}}_i
\end{aligned}$$

The two-step procedure begins with the usual OLS estimates of $\boldsymbol\beta_i$. With those estimates, we may proceed by obtaining estimates of $\widehat{\mathbf{V}}_i$ and $\widehat{\boldsymbol\Sigma}$ (and thus $\widehat{\mathbf{W}}_i$) and then obtain an estimate of $\boldsymbol\beta$.

Swamy (1970) further points out that the matrix $\widehat{\boldsymbol\Sigma}$ may not be positive definite and that because the second term is of order $1/(mT)$, it is negligible in large samples. A simple and asymptotically expedient solution is simply to drop this second term and instead use

$$\widehat{\boldsymbol\Sigma} = \frac{1}{m-1}\left(\sum_{i=1}^{m}\mathbf{b}_i\mathbf{b}_i' - m\overline{\mathbf{b}}\,\overline{\mathbf{b}}'\right)$$

As discussed by Judge et al. (1985, 541), the feasible best linear predictor of $\boldsymbol\beta_i$ is given by

$$\begin{aligned}
\widehat{\boldsymbol\beta}_i &= \widehat{\boldsymbol\beta} + \widehat{\boldsymbol\Sigma}\mathbf{X}_i'\left(\mathbf{X}_i\widehat{\boldsymbol\Sigma}\mathbf{X}_i' + \widehat{\sigma}_i^2\mathbf{I}\right)^{-1}\left(\mathbf{y}_i - \mathbf{X}_i\widehat{\boldsymbol\beta}\right)\\
&= \left(\widehat{\boldsymbol\Sigma}^{-1} + \widehat{\mathbf{V}}_i^{-1}\right)^{-1}\left(\widehat{\boldsymbol\Sigma}^{-1}\widehat{\boldsymbol\beta} + \widehat{\mathbf{V}}_i^{-1}\mathbf{b}_i\right)
\end{aligned}$$

The conventional variance of $\widehat{\boldsymbol\beta}_i$ is given by

$$\text{Var}(\widehat{\boldsymbol\beta}_i) = \text{Var}(\widehat{\boldsymbol\beta}) + (\mathbf{I} - \mathbf{A}_i)\left\{\widehat{\mathbf{V}}_i - \text{Var}(\widehat{\boldsymbol\beta})\right\}(\mathbf{I} - \mathbf{A}_i)'$$

where

$$\mathbf{A}_i = \left(\widehat{\boldsymbol\Sigma}^{-1} + \widehat{\mathbf{V}}_i^{-1}\right)^{-1}\widehat{\boldsymbol\Sigma}^{-1}$$

To test the model, we may look at the difference between the OLS estimate of $\boldsymbol\beta$, ignoring the panel structure of the data and the matrix-weighted average of the panel-specific OLS estimators. The test statistic suggested by Swamy (1970) is given by

$$\chi^2_{k(m-1)} = \sum_{i=1}^{m}(\mathbf{b}_i - \overline{\boldsymbol\beta}^*)'\widehat{\mathbf{V}}_i^{-1}(\mathbf{b}_i - \overline{\boldsymbol\beta}^*) \quad \text{where} \quad \overline{\boldsymbol\beta}^* = \left(\sum_{i=1}^{m}\widehat{\mathbf{V}}_i^{-1}\right)^{-1}\sum_{i=1}^{m}\widehat{\mathbf{V}}_i^{-1}\mathbf{b}_i$$

Johnston and DiNardo (1997) have shown that the test is algebraically equivalent to testing

$$H_0 : \boldsymbol{\beta}_1 = \boldsymbol{\beta}_2 = \cdots = \boldsymbol{\beta}_m$$

in the generalized (groupwise heteroskedastic) `xtgls` model, where $\mathbf{V}$ is block diagonal with ith diagonal element $\mathbf{\Pi}_i$.

References

Eberhardt, M. 2012. Estimating panel time-series models with heterogeneous slopes. *Stata Journal* 12: 61–71.

Greene, W. H. 2012. *Econometric Analysis*. 7th ed. Upper Saddle River, NJ: Prentice Hall.

Grunfeld, Y., and Z. Griliches. 1960. Is aggregation necessarily bad? *Review of Economics and Statistics* 42: 1–13. https://doi.org/10.2307/1926089.

Johnston, J., and J. DiNardo. 1997. *Econometric Methods*. 4th ed. New York: McGraw–Hill.

Judge, G. G., W. E. Griffiths, R. C. Hill, H. Lütkepohl, and T.-C. Lee. 1985. *The Theory and Practice of Econometrics*. 2nd ed. New York: Wiley.

Nichols, A. 2007. Causal inference with observational data. *Stata Journal* 7: 507–541.

Poi, B. P. 2003. From the help desk: Swamy's random-coefficients model. *Stata Journal* 3: 302–308.

Swamy, P. A. V. B. 1970. Efficient inference in a random coefficient regression model. *Econometrica* 38: 311–323. https://doi.org/10.2307/1913012.

———. 1971. *Statistical Inference in Random Coefficient Regression Models*. New York: Springer. https://doi.org/10.1007/978-3-642-80653-7.

Also see

[XT] **xtrc postestimation** — Postestimation tools for xtrc

[XT] **xtreg** — Linear models for panel data

[XT] **xtset** — Declare data to be panel data

[ME] **mixed** — Multilevel mixed-effects linear regression

[MI] **Estimation** — Estimation commands for use with mi estimate

[U] 20 Estimation and postestimation commands

xtrc postestimation — Postestimation tools for xtrc

Postestimation commands   predict   margins   Also see

Postestimation commands

The following postestimation commands are available after `xtrc`:

Command	Description
contrast	contrasts and ANOVA-style joint tests of parameters
estat summarize	summary statistics for the estimation sample
estat vce	variance–covariance matrix of the estimators (VCE)
estimates	cataloging estimation results
etable	table of estimation results
* forecast	dynamic forecasts and simulations
hausman	Hausman's specification test
lincom	point estimates, standard errors, testing, and inference for linear combinations of parameters
margins	marginal means, predictive margins, marginal effects, and average marginal effects
marginsplot	graph the results from margins (profile plots, interaction plots, etc.)
nlcom	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
predict	linear predictions and their SEs
predictnl	point estimates, standard errors, testing, and inference for generalized predictions
pwcompare	pairwise comparisons of parameters
test	Wald tests of simple and composite linear hypotheses
testnl	Wald tests of nonlinear hypotheses

*`forecast` is not appropriate with `mi` estimation results.

predict

Description for predict

predict creates a new variable containing predictions such as linear predictions and standard errors.

Menu for predict

Statistics > Postestimation

Syntax for predict

predict [*type*] *newvar* [*if*] [*in*] [, *statistic* nooffset]

statistic	Description
Main	
xb	linear prediction; the default
stdp	standard error of the linear prediction
group(*group*)	linear prediction based on group *group*

These statistics are available both in and out of sample; type predict ... if e(sample) ... if wanted only for the estimation sample.

Options for predict

Main

xb, the default, calculates the linear prediction using the mean parameter vector.

stdp calculates the standard error of the linear prediction.

group(*group*) calculates the linear prediction using the best linear predictors for group *group*.

nooffset is relevant only if you specified offset(*varname*) for xtrc. It modifies the calculations made by predict so that they ignore the offset variable; the linear prediction is treated as $\mathbf{x}_{it}\mathbf{b}$ rather than $\mathbf{x}_{it}\mathbf{b} + \text{offset}_{it}$.

margins

Description for margins

margins estimates margins of response for linear predictions.

Menu for margins

Statistics > Postestimation

Syntax for margins

margins [*marginlist*] [, *options*]

margins [*marginlist*] , predict(*statistic* ...) [*options*]

statistic	Description
xb	linear prediction; the default
stdp	not allowed with margins
group(*group*)	not allowed with margins

Statistics not allowed with margins are functions of stochastic quantities other than e(b).

For the full syntax, see [R] **margins**.

Also see

[XT] **xtrc** — Random-coefficients model

[U] **20 Estimation and postestimation commands**

xtreg — Linear models for panel data

Description

`xtreg` fits linear regression models for panel data. `xtreg` with the `re` option fits random-effects models using generalized least squares (GLS); `xtreg` with the `fe` option fits fixed-effects models using the within regression estimator; and `xtreg` with the `pa` option fits population-averaged models. Alternatively, random-effects models can be fit by using maximum likelihood (`mle` option) or the between-effects estimator (`be` option). `xtreg` with the `cre` option fits correlated random-effects models using a Mundlak regression. See [XT] **xtdata** for a faster way to fit fixed- and random-effects models.

Quick start

Random-effects linear regression by GLS of `y` on `x1` and `x2` using `xtset` data

```
xtreg y x1 x2
```

Same as above, but estimate by maximum likelihood

```
xtreg y x1 x2, mle
```

Fixed-effects model with cluster–robust standard errors for panels nested within `cvar`

```
xtreg y x1 x2, fe vce(cluster cvar)
```

Fixed-effects model absorbing indicator variables for the levels of `cvar`

```
xtreg y x1 x2, fe absorb(cvar)
```

Correlated random-effects model by Mundlak regression

```
xtreg y x1 x2, cre
```

Population-averaged model with an exchangeable within-panel correlation structure

```
xtreg y x1 x2, pa
```

Same as above, but specify an autoregressive correlation structure of order 1

```
xtreg y x1 x2, pa corr(ar 1)
```

Between-effects model

```
xtreg y x1 x2, be
```

Menu

Statistics > Longitudinal/panel data > Linear models > Linear regression (FE, RE, PA, BE, CRE)

Syntax

GLS random-effects (RE) model

xtreg *depvar* [*indepvars*] [*if*] [*in*] [, re *RE_options*]

Correlated random-effects (CRE) model

xtreg *depvar* [*indepvars*] [*if*] [*in*] , cre [*CRE_options*]

Between-effects (BE) model

xtreg *depvar* [*indepvars*] [*if*] [*in*] , be [*BE_options*]

Fixed-effects (FE) model

xtreg *depvar* [*indepvars*] [*if*] [*in*] [*weight*] , fe [*FE_options*]

Random-effects model using maximum likelihood estimation (MLE)

xtreg *depvar* [*indepvars*] [*if*] [*in*] [*weight*] , mle [*MLE_options*]

Population-averaged (PA) model

xtreg *depvar* [*indepvars*] [*if*] [*in*] [*weight*] , pa [*PA_options*]

RE_options	Description
Model	
re	use random-effects estimator; the default
sa	use Swamy–Arora estimator of the variance components
SE/Robust	
vce(*vcetype*)	*vcetype* may be conventional, robust, cluster *clustvar*, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
theta	report θ
eform[(*string*)]	report exponentiated coefficients and, optionally, label as *string*
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
coeflegend	display legend instead of statistics

CRE_options	Description
Model	
`cre`	use correlated random-effects estimator
`sa`	use Swamy–Arora estimator of variance components in Mundlak regression
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `conventional`, `robust`, `cluster` *clustvar*, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`nomeans`	do not display coefficients for panel means
`eform`[`(`*string*`)`]	report exponentiated coefficients and, optionally, label as *string*
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
`coeflegend`	display legend instead of statistics

BE_options	Description
Model	
`be`	use between-effects estimator
`wls`	use weighted least squares
SE	
`vce(`*vcetype*`)`	*vcetype* may be `conventional`, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`eform`[`(`*string*`)`]	report exponentiated coefficients and, optionally, label as *string*
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
`coeflegend`	display legend instead of statistics

FE_options	Description
Model	
`fe`	use fixed-effects estimator
`absorb(`*varlist*[, *method*]`)`	specify categorical variables to be absorbed
† `dfabsorb`	adjust degrees of freedom for collinearity among absorbed variables
`nosigmau`	do not estimate σ_u
`nouitest`	suppress the F test for u_i and do not estimate σ_u
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `conventional`, `robust`, `cluster` *clustvarlist*, `bootstrap`, `jackknife`, or `hc2` [*clustvar*]
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`clustertable`	display table of multiway cluster combinations
`eform`[`(`*string*`)`]	report exponentiated coefficients and, optionally, label as *string*
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Optimization	
`iterate(#)`	maximum number of iterations for alternating projection method (APM); default is `iterate(50)`
`nolog`	suppress the APM iteration log
`tolerance(#)`	convergence tolerance for maximum absolute difference; default is `tolerance(1e-8)`
`coeflegend`	display legend instead of statistics

†Ignored if `absorb()` is not specified.

MLE_options	Description
Model	
`noconstant`	suppress constant term
`mle`	use maximum-likelihood random-effects estimator
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `robust`, `cluster` *clustvar*, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`eform`[`(`*string*`)`]	report exponentiated coefficients and, optionally, label as *string*
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
maximize_options	control the maximization process; seldom used
`coeflegend`	display legend instead of statistics

PA_options	Description
Model	
noconstant	suppress constant term
pa	use population-averaged estimator
offset(*varname*)	include *varname* in model with coefficient constrained to 1
Correlation	
corr(*correlation*)	within-panel correlation structure
force	estimate even if observations unequally spaced in time
SE/Robust	
vce(*vcetype*)	*vcetype* may be conventional, robust, bootstrap, or jackknife
nmp	use divisor $N - P$ instead of the default N
rgf	multiply the robust variance estimate by $(N - 1)/(N - P)$
scale(*parm*)	overrides the default scale parameter; *parm* may be x2, dev, phi, or #
Reporting	
level(#)	set confidence level; default is level(95)
eform[(*string*)]	report exponentiated coefficients and, optionally, label as *string*
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Optimization	
optimize_options	control the optimization process; seldom used
coeflegend	display legend instead of statistics

correlation	Description
exchangeable	exchangeable
independent	independent
unstructured	unstructured
fixed *matname*	user-specified
ar #	autoregressive of order #
stationary #	stationary of order #
nonstationary #	nonstationary of order #

A panel variable must be specified. For `xtreg, pa`, correlation structures other than `exchangeable` and `independent` require that a time variable also be specified. Use `xtset`; see [XT] **xtset**.

indepvars may contain factor variables; see **[U] 11.4.3 Factor variables**.

depvar and *indepvars* may contain time-series operators; see **[U] 11.4.4 Time-series varlists**.

`by`, `collect`, `mi estimate`, and `statsby` are allowed; see **[U] 11.1.10 Prefix commands**. `bayes` is allowed for the GLS random-effects model. For more details, see [BAYES] **bayes: xtreg**. `fp` is allowed for the between-effects, fixed-effects, and maximum-likelihood random-effects models.

`vce(bootstrap)` and `vce(jackknife)` are not allowed with the `mi estimate` prefix; see [MI] **mi estimate**.

`vce()` is not allowed with the `by` prefix.

`aweight`s, `fweight`s, and `pweight`s are allowed for the fixed-effects model. `iweight`s, `fweight`s, and `pweight`s are allowed for the population-averaged model. `iweight`s are allowed for the maximum-likelihood random-effects model. See **[U] 11.1.6 weight**. Weights must be constant within panel.

`coeflegend` does not appear in the dialog box.

See **[U] 20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Options are presented under the following headings:

Options for RE model

Model

`re`, the default, requests the GLS random-effects estimator.

`sa` specifies that the small-sample Swamy–Arora estimator for the individual-level variance component be used instead of the default consistent estimator. See *xtreg, re* in *Methods and formulas* for details.

SE/Robust

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`conventional`), that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster` *clustvar*), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [XT] ***vce_options***.

`vce(conventional)`, the default, uses the conventionally derived variance estimator for generalized least-squares regression.

Specifying `vce(robust)` is equivalent to specifying `vce(cluster` *panelvar*`)`; see *xtreg, re* in *Methods and formulas*.

Reporting

`level(`#`)`; see [R] **Estimation options**.

`theta` specifies that the output include the estimated value of θ used in combining the between and fixed estimators. For balanced data, this is a constant, and for unbalanced data, a summary of the values is presented in the header of the output.

eform and eform(*string*) specify that the coefficient table be displayed in exponentiated form and that exp(b) and *string*, respectively, be used to label the exponentiated coefficients in the table. Standard errors and confidence intervals are also transformed.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

The following option is available with xtreg but is not shown in the dialog box:

coeflegend; see [R] **Estimation options**.

Options for CRE model

Model

cre requests the correlated random-effects estimator.

sa specifies that the small-sample Swamy–Arora estimator for the individual-level variance component be used when estimating the Mundlak regression by random effects. See *xtreg, re* in *Methods and formulas* for details.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (conventional), that are robust to some kinds of misspecification (robust), that allow for intragroup correlation (cluster *clustvar*), and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] ***vce_options***.

vce(conventional), the default, uses the conventionally derived variance estimator for generalized least-squares regression.

Specifying vce(robust) is equivalent to specifying vce(cluster *panelvar*).

Reporting

level(#); see [R] **Estimation options**.

nomeans specifies that the coefficients of the panel means not be displayed in the output table.

eform and eform(*string*) specify that the coefficient table be displayed in exponentiated form and that exp(b) and *string*, respectively, be used to label the exponentiated coefficients in the table. Standard errors and confidence intervals are also transformed.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

The following option is available with xtreg but is not shown in the dialog box:

coeflegend; see [R] **Estimation options**.

Options for BE model

Model

`be` requests the between regression estimator.

`wls` specifies that, for unbalanced data, weighted least squares be used rather than the default OLS. Both methods produce consistent estimates. The true variance of the between-effects residual is $\sigma_\nu^2 + T_i\sigma_\epsilon^2$ (see xtreg, be in *Methods and formulas* below). WLS produces a "stabilized" variance of $\sigma_\nu^2/T_i + \sigma_\epsilon^2$, which is also not constant. Thus the choice between OLS and WLS amounts to which is more stable.

Comment: `xtreg, be` is rarely used anyway, but between estimates are an ingredient in the random-effects estimate. Our implementation of `xtreg, re` uses the OLS estimates for this ingredient, based on our judgment that σ_ν^2 is large relative to σ_ϵ^2 in most models. Formally, only a consistent estimate of the between estimates is required.

SE

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`conventional`) and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [XT] ***vce_options***.

`vce(conventional)`, the default, uses the conventionally derived variance estimator for generalized least-squares regression.

Reporting

`level(#)`; see [R] **Estimation options**.

`eform` and `eform(`*string*`)` specify that the coefficient table be displayed in exponentiated form and that `exp(b)` and *string*, respectively, be used to label the exponentiated coefficients in the table. Standard errors and confidence intervals are also transformed.

display_options: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(`*style*`)`, `cformat(%`*fmt*`)`, `pformat(%`*fmt*`)`, `sformat(%`*fmt*`)`, and `nolstretch`; see [R] **Estimation options**.

The following option is available with `xtreg` but is not shown in the dialog box:

`coeflegend`; see [R] **Estimation options**.

Options for FE model

Model

`fe` requests the fixed-effects (within) regression estimator.

`absorb(`*varlist*[, *method*]`)` specifies the categorical variables to be absorbed. The results are adjusted as if indicator variables for each level of each variable in *varlist* were included in the regression. The panel variable is included as the first absorbed variable.

An APM iterative algorithm is used to project the *depvar* and *indepvars* to absorb these variables and the panel variable. *method* specifies the APM and is one of `halperin` or `cimmino`.

`halperin`, the default, uses the product of the projection matrices.

`cimmino` uses the mean of the projection matrices.

The two methods typically perform similarly. See Stammann (2018) for details.

absorb() may not be specified with the vce(hc2) option.

dfabsorb adjusts the degrees of freedom to account for collinearity among the absorbed variables. The default degrees of freedom assumes all absorbed variables are independent. If absorb() is not also specified, only the panel variable is absorbed, and the dfabsorb option is ignored.

nosigmau prevents the computation of σ_u. When absorb(*varlist*) is specified, the estimation of σ_u requires an iterative search. If σ_u is not of interest, you may specify nosigmau to save computational time.

nouitest prevents computation of σ_u and the F test that all u_i are jointly zero. If σ_u and the F test are of no interest, you may specify nouitest to save computational time.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (conventional), that are robust to some kinds of misspecification (robust), that allow for intragroup correlations (cluster *clustvarlist*), and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] ***vce_options***.

vce(conventional), the default, uses the conventionally derived variance estimator for ordinary least-squares regression.

vce(robust) is equivalent to specifying vce(cluster *panelvar*).

vce(cluster *clustvarlist)* specifies that standard errors allow for intragroup correlation within groups defined by one or more variables in *clustvarlist*, relaxing the usual requirement that the observations be independent. For example, vce(cluster clustvar1) produces cluster–robust standard errors that allow for observations that are independent across groups defined by clustvar1 but not necessarily independent within groups. You could also type vce(cluster clustvar1 clustvar2 ... clusterp) to account for correlation within groups formed by p variables (multiway clustering).

xtreg also allows the following:

vce(hc2 [*clustvar*][, dfadjust]) specifies a cluster–robust variance calculation that uses a bias correction that is an alternative to the one provided by vce(robust) and vce(cluster *clustvarlist)*. Specifying vce(hc2[, dfadjust]) is equivalent to specifying vce(hc2 *panelvar*[, dfadjust]). dfadjust computes the Bell and McCaffrey (2002) adjusted degrees of freedom based on the clusters of *clustvar*; see xtreg, fe in *Methods and formulas*. Note that dfadjust does not affect multiple-imputation results when the command is used with mi estimate.

vce(hc2) may not be specified with the absorb() option.

Reporting

level(#); see [R] **Estimation options**.

clustertable displays a table reporting cluster combinations and the number of clusters per combination. This option is available only when vce(cluster *clustvarlist)* is specified with more than one variable in *clustvarlist* to compute multiway cluster–robust standard errors.

eform and eform(*string*) specify that the coefficient table be displayed in exponentiated form and that exp(b) and *string*, respectively, be used to label the exponentiated coefficients in the table. Standard errors and confidence intervals are also transformed.

display_options: noci, nopvalues, dfci, dfpvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

dfci specifies that parameter degrees of freedom and confidence intervals be reported in the coefficient table.

dfpvalues specifies that parameter degrees of freedom and p-values be reported in the coefficient table.

Optimization

iterate(#) specifies the maximum number of iterations for the APM. The default is iterate(50).

nolog specifies that no APM iterative log be displayed.

tolerance(#) specifies the limit for the maximum absolute difference between iterations for the projected *depvar* and *indepvars*. The default is tolerance(1e-8).

The following option is available with xtreg but is not shown in the dialog box:

coeflegend; see [R] **Estimation options**.

Options for MLE model

Model

noconstant; see [R] **Estimation options**.

mle requests the maximum-likelihood random-effects estimator.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (oim), that are robust to some kinds of misspecification (robust), that allow for intragroup correlation (cluster *clustvar*), and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] ***vce_options***.

Reporting

level(#); see [R] **Estimation options**.

eform and eform(*string*) specify that the coefficient table be displayed in exponentiated form and that exp(b) and *string*, respectively, be used to label the exponentiated coefficients in the table. Standard errors and confidence intervals are also transformed.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Maximization

maximize_options: iterate(#), [no]log, trace, tolerance(#), ltolerance(#), and from(*init_specs*); see [R] **Maximize**. These options are seldom used.

The following option is available with xtreg but is not shown in the dialog box:

coeflegend; see [R] **Estimation options**.

Options for PA model

Model

noconstant; see [R] **Estimation options**.

pa requests the population-averaged estimator. For linear regression, this is the same as a random-effects estimator (both interpretations hold).

xtreg, pa is equivalent to xtgee, family(gaussian) link(id) corr(exchangeable), which are the defaults for the xtgee command. xtreg, pa allows all the relevant xtgee options such as vce(robust). Whether you use xtreg, pa or xtgee makes no difference. See [XT] **xtgee**.

offset(*varname*); see [R] **Estimation options**.

Correlation

corr(*correlation*) specifies the within-panel correlation structure; the default corresponds to the equal-correlation model, corr(exchangeable).

When you specify a correlation structure that requires a lag, you indicate the lag after the structure's name with or without a blank; for example, corr(ar 1) or corr(ar1).

If you specify the fixed correlation structure, you specify the name of the matrix containing the assumed correlations following the word fixed, for example, corr(fixed myr).

force specifies that estimation be forced even though the time variable is not equally spaced. This is relevant only for correlation structures that require knowledge of the time variable. These correlation structures require that observations be equally spaced so that calculations based on lags correspond to a constant time change. If you specify a time variable indicating that observations are not equally spaced, the (time dependent) model will not be fit. If you also specify force, the model will be fit, and it will be assumed that the lags based on the data ordered by the time variable are appropriate.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (conventional), that are robust to some kinds of misspecification (robust), and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] ***vce_options***.

vce(conventional), the default, uses the conventionally derived variance estimator for generalized least-squares regression.

nmp; see [XT] ***vce_options***.

rgf specifies that the robust variance estimate is multiplied by $(N-1)/(N-P)$, where N is the total number of observations and P is the number of coefficients estimated. This option can be used with family(gaussian) only when vce(robust) is either specified or implied by the use of pweights. Using this option implies that the robust variance estimate is not invariant to the scale of any weights used.

scale(x2 | dev | phi | #); see [XT] ***vce_options***.

Reporting

level(#); see [R] **Estimation options**.

eform and eform(*string*) specify that the coefficient table be displayed in exponentiated form and that exp(b) and *string*, respectively, be used to label the exponentiated coefficients in the table. Standard errors and confidence intervals are also transformed.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Optimization

optimize_options control the iterative optimization process. These options are seldom used.

iterate(#) specifies the maximum number of iterations. When the number of iterations equals #, the optimization stops and presents the current results, even if convergence has not been reached. The default is iterate(100).

tolerance(#) specifies the tolerance for the coefficient vector. When the relative change in the coefficient vector from one iteration to the next is less than or equal to #, the optimization process is stopped. tolerance(1e-6) is the default.

log and nolog specify whether to display the iteration log. The iteration log is displayed by default unless you used set iterlog off to suppress it; see set iterlog in [R] ***set iter***.

trace specifies that the current estimates be printed at each iteration.

The following option is available with xtreg but is not shown in the dialog box:

coeflegend; see [R] **Estimation options**.

Remarks and examples

If you have not read [XT] **xt**, please do so.

See Baltagi (2013, chap. 2) and Wooldridge (2020, chap. 14) for good overviews of fixed-effects and random-effects models. Allison (2009) provides perspective on the use of fixed- versus random-effects estimators and provides many examples using Stata.

Consider fitting models of the form

$$y_{it} = \alpha + \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i + \epsilon_{it} \tag{1}$$

In this model, $\nu_i + \epsilon_{it}$ is the error term that we have little interest in; we want estimates of $\boldsymbol{\beta}$. ν_i is the unit-specific error term; it differs between units, but for any particular unit, its value is constant. In the pulmonary data of [XT] **xt**, a person who exercises less would presumably have a lower forced expiratory volume year after year and so would have a negative ν_i.

ϵ_{it} is the "usual" error term with the usual properties (mean 0, uncorrelated with itself, uncorrelated with $\mathbf{x}$, uncorrelated with ν, and homoskedastic), although in a more thorough development, we could decompose $\epsilon_{it} = \upsilon_t + \omega_{it}$, assume that ω_{it} is a conventional error term, and better describe υ_t.

Before making the assumptions necessary for estimation, let's perform some useful algebra on (1). Whatever the properties of ν_i and ϵ_{it}, if (1) is true, it must also be true that

$$\overline{y}_i = \alpha + \overline{\mathbf{x}}_i\boldsymbol{\beta} + \nu_i + \overline{\epsilon}_i \tag{2}$$

where $\overline{y}_i = \sum_t y_{it}/T_i$, $\overline{\mathbf{x}}_i = \sum_t \mathbf{x}_{it}/T_i$, and $\overline{\epsilon}_i = \sum_t \epsilon_{it}/T_i$. Subtracting (2) from (1), it must be equally true that

$$(y_{it} - \overline{y}_i) = (\mathbf{x}_{it} - \overline{\mathbf{x}}_i)\boldsymbol{\beta} + (\epsilon_{it} - \overline{\epsilon}_i) \tag{3}$$

These three equations provide the basis for estimating $\boldsymbol{\beta}$. In particular, `xtreg, fe` provides what is known as the fixed-effects estimator—also known as the within estimator—and amounts to using OLS to perform the estimation of (3). `xtreg, be` provides what is known as the between estimator and amounts to using OLS to perform the estimation of (2). `xtreg, re` provides the random-effects estimator and is a (matrix) weighted average of the estimates produced by the between and within estimators. In particular, the random-effects estimator turns out to be equivalent to estimation of

$$(y_{it} - \theta_i \overline{y}_i) = (1-\theta_i)\alpha + (\mathbf{x}_{it} - \theta_i \overline{\mathbf{x}}_i)\boldsymbol{\beta} + \{(1-\theta_i)\nu_i + (\epsilon_{it} - \theta_i \overline{\epsilon}_i)\} \tag{4}$$

where θ_i is a function of σ_ν^2, σ_ϵ^2 and T_i. If $\sigma_\nu^2 = 0$, meaning that ν_i is always 0, $\theta_i = 0$ for all i and (1) can be estimated by OLS directly. Alternatively, if $\sigma_\epsilon^2 = 0$, meaning that ϵ_{it} is 0, $\theta_i = 1$ for all i and the within estimator returns all the information available (which will, in fact, be a regression with an R^2 of 1).

Consider an extended version of (4), where we add the panel averages of the covariates as regressors, so that

$$(y_{it} - \theta_i \overline{y}_i) = (1-\theta_i)\alpha + (\mathbf{x}_{it} - \theta_i \overline{\mathbf{x}}_i)\boldsymbol{\beta} + (1-\theta_i)\overline{\mathbf{x}}_i\gamma + \{(1-\theta_i)\nu_i + (\epsilon_{it} - \theta_i \overline{\epsilon}_i)\} \tag{5}$$

This extended model is known as the Mundlak regression or, more recently, as the correlated random-effects model. Mundlak (1978) showed that, for balanced panels, the estimation of (5) by OLS produces the same coefficients for time-variant regressors as `xtreg, fe`. This equivalence result was extended to unbalanced panels in Wooldridge (2019). With unbalanced panels, for the equivalence to hold, the panel averages in (5) should be calculated using only observations in the estimation sample. `xtreg, cre` fits the correlated random-effects model.

Few assumptions are required to justify the fixed-effects estimator of (3). The estimates are, however, conditional on the sample in that the ν_i are not assumed to have a distribution but are instead treated as fixed and estimable. This statistical fine point can lead to difficulty when making out-of-sample predictions, but that aside, the fixed-effects estimator has much to recommend it.

More is required to justify the between estimator of (2), but the conditioning on the sample is not assumed because $\nu_i + \overline{\epsilon}_i$ is treated as an error term. Newly required is that we assume that ν_i and $\overline{\mathbf{x}}_i$ are uncorrelated. This follows from the assumptions of the OLS estimator but is also transparent: were ν_i and $\overline{\mathbf{x}}_i$ correlated, the estimator could not determine how much of the change in $\overline{y}_i$, associated with an increase in $\overline{\mathbf{x}}_i$, to assign to $\boldsymbol{\beta}$ versus how much to attribute to the unknown correlation. (This, of course, suggests the use of an instrumental-variable estimator, $\overline{\mathbf{z}}_i$, which is correlated with $\overline{\mathbf{x}}_i$ but uncorrelated with ν_i, though that approach is not implemented here.)

The random-effects estimator of (4) requires the same no-correlation assumption. In comparison with the between estimator, the random-effects estimator produces more efficient results, albeit ones with unknown small-sample properties. The between estimator is less efficient because it discards the over-time information in the data in favor of simple means; the random-effects estimator uses both the within and the between information.

The correlated random-effects estimator of (5) does not require the no-correlation assumption that is required by the random-effects estimator.

All of this would seem to leave the between estimator of (2) with no role (except for a minor, technical part it plays in helping to estimate σ_ν^2 and σ_ϵ^2, which are used in the calculation of θ, on which the random-effects estimates depend). Let's, however, consider a variation on (1):

$$y_{it} = \alpha + \overline{\mathbf{x}}_i\boldsymbol{\beta}_1 + (\mathbf{x}_{it} - \overline{\mathbf{x}}_i)\boldsymbol{\beta}_2 + \nu_i + \epsilon_{it} \tag{1$'$}$$

In this model, we postulate that changes in the average value of $\mathbf{x}$ for an individual have a different effect from temporary departures from the average. In an economic situation, y might be purchases of some item and $\mathbf{x}$ income; a change in average income should have more effect than a transitory change. In a clinical situation, y might be a physical response and $\mathbf{x}$ the level of a chemical in the brain; the model allows a different response to permanent rather than transitory changes.

The variations of (2) and (3) corresponding to ($1'$) are

$$\overline{y}_i = \alpha + \overline{\mathbf{x}}_i\boldsymbol{\beta}_1 + \nu_i + \overline{\epsilon}_i \tag{$2'$}$$

$$(y_{it} - \overline{y}_i) = (\mathbf{x}_{it} - \overline{\mathbf{x}}_i)\boldsymbol{\beta}_2 + (\epsilon_{it} - \overline{\epsilon}_i) \tag{$3'$}$$

That is, the between estimator estimates $\boldsymbol{\beta}_1$ and the within $\boldsymbol{\beta}_2$, and neither estimates the other. Thus even when estimating equations like (1), it is worth comparing the within and between estimators. Differences in results can suggest models like ($1'$), or at the least some other specification error.

Finally, it is worth understanding the role of the between and within estimators with regressors that are constant over time or constant over units. Consider the model

$$y_{it} = \alpha + \mathbf{x}_{it}\boldsymbol{\beta}_1 + \mathbf{s}_i\boldsymbol{\beta}_2 + \mathbf{z}_t\boldsymbol{\beta}_3 + \nu_i + \epsilon_{it} \tag{$1''$}$$

This model is the same as (1), except that we explicitly identify the variables that vary over both time and i ($\mathbf{x}_{it}$, such as output or forced expiratory volume; variables that are constant over time ($\mathbf{s}_i$, such as race or sex); and variables that vary solely over time ($\mathbf{z}_t$, such as the consumer price index or age in a cohort study). The corresponding between and within equations are

$$\overline{y}_i = \alpha + \overline{\mathbf{x}}_i\boldsymbol{\beta}_1 + \mathbf{s}_i\boldsymbol{\beta}_2 + \overline{\mathbf{z}}\boldsymbol{\beta}_3 + \nu_i + \overline{\epsilon}_i \tag{$2''$}$$

$$(y_{it} - \overline{y}_i) = (\mathbf{x}_{it} - \overline{\mathbf{x}}_i)\boldsymbol{\beta}_1 + (\mathbf{z}_t - \overline{\mathbf{z}})\boldsymbol{\beta}_3 + (\epsilon_{it} - \overline{\epsilon}_i) \tag{$3''$}$$

In the between estimator of ($2''$), no estimate of $\boldsymbol{\beta}_3$ is possible because $\overline{\mathbf{z}}$ is a constant across the i observations; the regression-estimated intercept will be an estimate of $\alpha + \overline{\mathbf{z}}\boldsymbol{\beta}_3$. On the other hand, it can provide estimates of $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$. It can estimate effects of factors that are constant over time, such as race and sex, but to do so it must assume that ν_i is uncorrelated with those factors.

The within estimator of ($3''$), like the between estimator, provides an estimate of $\boldsymbol{\beta}_1$ but provides no estimate of $\boldsymbol{\beta}_2$ for time-invariant factors. Instead, it provides an estimate of $\boldsymbol{\beta}_3$, the effects of the time-varying factors. The within estimator can also provide estimates u_i for ν_i. More correctly, the estimator u_i is an estimator of $\nu_i + \mathbf{s}_i\boldsymbol{\beta}_2$. Thus u_i is an estimator of ν_i only if there are no time-invariant variables in the model. If there are time-invariant variables, u_i is an estimate of ν_i plus the effects of the time-invariant variables.

The correlated random-effects estimator of (5) can provide estimates of coefficients on all three types of variables—those that vary across time and panels, those that vary only across time, and those that vary only across panels.

Remarks are presented under the following headings:

Assessing goodness of fit
xtreg and associated commands

Assessing goodness of fit

R^2 is a popular measure of goodness of fit in ordinary regression. In our case, given $\widehat{\alpha}$ and $\widehat{\boldsymbol{\beta}}$ estimates of α and $\boldsymbol{\beta}$, we can assess the goodness of fit with respect to (1), (2), or (3). The prediction equations are, respectively,

$$\widehat{y}_{it} = \widehat{\alpha} + \mathbf{x}_{it}\widehat{\boldsymbol{\beta}} \tag{1'''}$$

$$\widehat{\overline{y}}_i = \widehat{\alpha} + \overline{\mathbf{x}}_i\widehat{\boldsymbol{\beta}} \tag{2'''}$$

$$\widehat{\widetilde{y}}_{it} = (\widehat{y}_{it} - \widehat{\overline{y}}_i) = (\mathbf{x}_{it} - \overline{\mathbf{x}}_i)\widehat{\boldsymbol{\beta}} \tag{3'''}$$

`xtreg` reports "R-squares" corresponding to these three equations. R-squares is in quotes because the R-squares reported do not have all the properties of the OLS R^2.

The ordinary properties of R^2 include being equal to the squared correlation between $\widehat{y}$ and y and being equal to the fraction of the variation in y explained by $\widehat{y}$—formally defined as $\text{Var}(\widehat{y})/\text{Var}(y)$. The identity of the definitions is from a special property of the OLS estimates; in general, given a prediction $\widehat{y}$ for y, the squared correlation is not equal to the ratio of the variances, and the ratio of the variances is not required to be less than 1.

`xtreg` reports R^2 values calculated as correlations squared, calling them R^2 overall, corresponding to (1‴); R^2 between, corresponding to (2‴); and R^2 within, corresponding to (3‴). In fact, you can think of each of these three numbers as having all the properties of ordinary R^2's, if you bear in mind that the prediction being judged is not $\widehat{y}_{it}$, $\widehat{\overline{y}}_i$, and $\widehat{\widetilde{y}}_{it}$, but $\gamma_1\widehat{y}_{it}$ from the regression $y_{it} = \gamma_1\widehat{y}_{it}$; $\gamma_2\widehat{\overline{y}}_i$ from the regression $\overline{y}_i = \gamma_2\widehat{\overline{y}}_i$; and $\gamma_3\widehat{\widetilde{y}}_{it}$ from $\widetilde{y}_{it} = \gamma_3\widehat{\widetilde{y}}_{it}$.

In particular, `xtreg, be` obtains its estimates by performing OLS on (2), and therefore its reported R^2 between is an ordinary R^2. The other two reported R^2's are merely correlations squared, or, if you prefer, R^2's from the second-round regressions $y_{it} = \gamma_{11}\widehat{y}_{it}$ and $\widetilde{y}_{it} = \gamma_{13}\widehat{\widetilde{y}}_{it}$.

`xtreg, fe` obtains its estimates by performing OLS on (3), so its reported R^2 within is an ordinary R^2. As with `be`, the other R^2's are correlations squared, or, if you prefer, R^2's from the second-round regressions $\overline{y_i} = \gamma_{22}\widehat{\overline{y}}_i$ and, as with `be`, $\widetilde{y}_{it} = \gamma_{23}\widehat{\widetilde{y}}_{it}$.

`xtreg, re` obtains its estimates by performing OLS on (4); none of the R^2's corresponding to (1‴), (2‴), or (3‴) correspond directly to this estimator [the "relevant" R^2 is the one corresponding to (4)]. All three reported R^2's are correlations squared, or, if you prefer, from second-round regressions.

`xtreg, cre` obtains its estimates by performing OLS on the extended model in (5). Because this is equivalent to performing a random-effects regression with the panel means included as additional controls, the three R^2's reported by `xtreg, cre` correspond to the R^2's that `xtreg, re` reports for the extended model.

xtreg and associated commands

▷ Example 1: Between-effects model

Using nlswork.dta described in [XT] **xt**, we will model ln_wage in terms of completed years of schooling (grade), current age and age squared, current years worked (experience) and experience squared, current years of tenure on the current job and tenure squared, whether black (race = 2), whether residing in an area not designated a standard metropolitan statistical area (SMSA), and whether residing in the South.

```
. use https://www.stata-press.com/data/r19/nlswork
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)
```

To obtain the between-effects estimates, we use xtreg, be. nlswork.dta has previously been xtset idcode year because that is what is true of the data, but for running xtreg, it would have been sufficient to have xtset idcode by itself.

```
. xtreg ln_w grade age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp tenure
> c.tenure#c.tenure 2.race not_smsa south, be

Between regression (regression on group means)  Number of obs     =     28,091
Group variable: idcode                          Number of groups  =      4,697

R-squared:                                      Obs per group:
     Within  = 0.1591                                         min =          1
     Between = 0.4900                                         avg =        6.0
     Overall = 0.3695                                         max =         15

                                                F(10,4686)        =     450.23
sd(u_i + avg(e_i.)) = .3036114                  Prob > F          =     0.0000

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
       grade |   .0607602   .0020006    30.37   0.000     .0568382    .0646822
         age |   .0323158   .0087251     3.70   0.000     .0152105    .0494211
             |
 c.age#c.age |  -.0005997   .0001429    -4.20   0.000    -.0008799   -.0003194
             |
     ttl_exp |   .0138853   .0056749     2.45   0.014     .0027598    .0250108
             |
   c.ttl_exp#|
   c.ttl_exp |   .0007342   .0003267     2.25   0.025     .0000936    .0013747
             |
      tenure |   .0698419   .0060729    11.50   0.000     .0579361    .0817476
             |
    c.tenure#|
    c.tenure |  -.0028756   .0004098    -7.02   0.000    -.0036789   -.0020722
             |
        race |
       Black |  -.0564167   .0105131    -5.37   0.000    -.0770272   -.0358061
    not_smsa |  -.1860406   .0112495   -16.54   0.000    -.2080949   -.1639862
       south |  -.0993378    .010136    -9.80   0.000    -.1192091   -.0794665
       _cons |   .3339113   .1210434     2.76   0.006     .0966093    .5712133
------------------------------------------------------------------------------
```

The between-effects regression is estimated on person-averages, so the "n = 4697" result is relevant. xtreg, be reports the "number of observations" and group-size information: describe in [XT] **xt** showed that we have 28,534 "observations"—person-years, really—of data. If we take the subsample that has no missing values in ln_wage, grade, …, south leaves us with 28,091 observations on person-years, reflecting 4,697 persons, each observed for an average of 6.0 years.

For goodness of fit, the R^2 between is directly relevant; our R^2 is 0.4900. If, however, we use these estimates to predict the within model, we have an R^2 of 0.1591. If we use these estimates to fit the overall data, our R^2 is 0.3695.

The F statistic tests that the coefficients on the regressors `grade`, `age`, ..., `south` are all jointly zero. Our model is significant.

The root mean squared error of the fitted regression, which is an estimate of the standard deviation of $\nu_i + \overline{\epsilon}_i$, is 0.3036.

For our coefficients, each year of schooling increases hourly wages by 6.1%; age increases wages up to age 26.9 and thereafter decreases them (because the quadratic $ax^2 + bx + c$ turns over at $x = -b/2a$, which for our `age` and `c.age#c.age` coefficients is $0.0323158/(2 \times 0.0005997) \approx 26.9$); total experience increases wages at an increasing rate (which is surprising and bothersome); tenure on the current job increases wages up to a tenure of 12.1 years and thereafter decreases them; wages of blacks are, these things held constant, (approximately) 5.6% below that of nonblacks (approximately because `2.race` is an indicator variable); residing in a non-SMSA (rural area) reduces wages by 18.6%; and residing in the South reduces wages by 9.9%.

◁

▷ Example 2: Fixed-effects model

To fit the same model with the fixed-effects estimator, we specify the `fe` option.

```
. xtreg ln_w grade age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp tenure
> c.tenure#c.tenure 2.race not_smsa south, fe
note: grade omitted because of collinearity.
note: 2.race omitted because of collinearity.

Fixed-effects (within) regression               Number of obs     =     28,091
Group variable: idcode                          Number of groups  =      4,697

R-squared:                                      Obs per group:
     Within  = 0.1727                                         min =          1
     Between = 0.3505                                         avg =        6.0
     Overall = 0.2625                                         max =         15

                                                F(8, 23386)       =     610.12
corr(u_i, Xb) = 0.1936                          Prob > F          =     0.0000

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
       grade |          0  (omitted)
         age |   .0359987   .0033864    10.63   0.000     .0293611    .0426362
             |
 c.age#c.age |   -.000723   .0000533   -13.58   0.000    -.0008274   -.0006186
             |
     ttl_exp |   .0334668   .0029653    11.29   0.000     .0276545     .039279
             |
   c.ttl_exp#|
   c.ttl_exp |   .0002163   .0001277     1.69   0.090    -.0000341    .0004666
             |
      tenure |   .0357539   .0018487    19.34   0.000     .0321303    .0393775
             |
    c.tenure#|
    c.tenure |  -.0019701    .000125   -15.76   0.000    -.0022151   -.0017251
             |
        race |
      Black  |          0  (omitted)
    not_smsa |  -.0890108   .0095316    -9.34   0.000    -.1076933   -.0703282
       south |  -.0606309   .0109319    -5.55   0.000    -.0820582   -.0392036
       _cons |    1.03732   .0485546    21.36   0.000     .9421496     1.13249
-------------+----------------------------------------------------------------
     sigma_u |  .35562203
     sigma_e |  .29068923
         rho |  .59946283   (fraction of variance due to u_i)
------------------------------------------------------------------------------
F test that all u_i=0: F(4696, 23386) = 6.65                 Prob > F = 0.0000
```

The observation summary at the top is the same as for the between-effects model, although this time it is the "`Number of obs`" that is relevant.

Our three R^2's are not too different from those reported previously; the R^2 within is slightly higher (0.1727 versus 0.1591), and the R^2 between is a little lower (0.3505 versus 0.4900), as expected, because the between estimator maximizes R^2 between and the within estimator R^2 within. In terms of overall fit, these estimates are somewhat worse (0.2625 versus 0.3695).

If the unobserved time-invariant component ν is not correlated with the regressors, estimates from the fixed-effects model are consistent but inefficient relative to estimates from the random-effects model. In this case, the interpretation of `sigma_u` in the coefficient table is the same for the fixed-effects and random-effects models. However, `sigma_u` is a nuisance parameter when ν is correlated with the covariates.

The postestimation command estat mundlak performs a Mundlak specification test to help decide between random-effects and fixed-effects estimation. A rejection of the null hypothesis in this test indicates that ν is probably correlated with the regressors and that the random-effects estimator is inconsistent.

```
. estat mundlak
Mundlak specification test
H0: Covariates are uncorrelated with unobserved panel-level effects
    chi2(8) = 151.62
Prob > chi2 = 0.0000
Notes: Fixed effects and correlated random effects are
       consistent under H0 and Ha.
       Random effects are efficient under H0.
```

In this case, the test favors the implemented fixed-effects model.

In the xtreg output, we see that both grade and 2.race were omitted from the model because they do not vary over time. Our estimate u_i is an estimate of ν_i plus the effects of any time-invariant variables, including grade and 2.race, so our estimate of the standard deviation is based on the variation in ν_i, grade, 2.race, and any other time-invariant variables. On the other hand, had 2.race and grade been omitted merely because they were collinear with the other regressors in our model, u_i would be an estimate of ν_i, and 0.355622 would be an estimate of σ_ν. (xtsum and xttab allow you to determine whether a variable is time invariant; see [XT] **xtsum** and [XT] **xttab**.)

Regardless of the status of u_i, our estimate of the standard deviation of ϵ_{it} is valid (and, in fact, is the estimate that would be used by the random-effects estimator to produce its results).

Our estimate of the correlation of u_i with $\mathbf{x}_{it}$ suffers from the problem of what u_i measures. We find correlation but cannot say whether this is correlation of ν_i with $\mathbf{x}_{it}$ or merely correlation of grade and 2.race with $\mathbf{x}_{it}$. In any case, the fixed-effects estimator is robust to such a correlation, and the other estimates it produces are unbiased.

So, although this estimator produces no estimates of the effects of grade and 2.race, it does predict that age has a positive effect on wages up to age 24.9 years (compared with 26.9 years estimated by the between estimator); that total experience still increases wages at an increasing rate (which is still bothersome); that tenure increases wages up to 9.1 years (compared with 12.1); that living in a non-SMSA reduces wages by 8.9% (compared with a more drastic 18.6%); and that living in the South reduces wages by 6.1% (as compared with 9.9%).

◁

▷ Example 3: Fixed-effects models with robust standard errors

If we suspect that there is heteroskedasticity or within-panel serial correlation in the idiosyncratic error term ϵ_{it}, we could specify the `vce(robust)` or the `vce(hc2)` option:

```
. xtreg ln_w grade age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp tenure
> c.tenure#c.tenure 2.race not_smsa south, fe vce(robust)
note: grade omitted because of collinearity.
note: 2.race omitted because of collinearity.

Fixed-effects (within) regression               Number of obs     =     28,091
Group variable: idcode                          Number of groups  =      4,697

R-squared:                                      Obs per group:
     Within  = 0.1727                                         min =          1
     Between = 0.3505                                         avg =        6.0
     Overall = 0.2625                                         max =         15

                                                F(8, 4696)        =     273.86
corr(u_i, Xb) = 0.1936                          Prob > F          =     0.0000

                             (Std. err. adjusted for 4,697 clusters in idcode)
```

ln_wage	Coefficient	Robust std. err.	t	P>|t|	[95% conf.	interval]
grade	0	(omitted)				
age	.0359987	.0052407	6.87	0.000	.0257243	.046273
c.age#c.age	-.000723	.0000845	-8.56	0.000	-.0008887	-.0005573
ttl_exp	.0334668	.004069	8.22	0.000	.0254896	.0414439
c.ttl_exp#c.ttl_exp	.0002163	.0001763	1.23	0.220	-.0001294	.0005619
tenure	.0357539	.0024683	14.49	0.000	.0309148	.040593
c.tenure#c.tenure	-.0019701	.0001696	-11.62	0.000	-.0023026	-.0016376
race						
Black	0	(omitted)				
not_smsa	-.0890108	.0137629	-6.47	0.000	-.1159926	-.062029
south	-.0606309	.0163366	-3.71	0.000	-.0926583	-.0286035
_cons	1.03732	.0739644	14.02	0.000	.8923149	1.182325
sigma_u	.35562203					
sigma_e	.29068923					
rho	.59946283	(fraction of variance due to u_i)				

Although the estimated coefficients are the same with and without the `vce(robust)` option, the robust estimator produced larger standard errors and a p-value for `c.ttl_exp#c.ttl_exp` above the conventional 10%. The F test of $\nu_i = 0$ is suppressed because it is too difficult to compute the robust form of the statistic when there are more than a few panels.

◁

❑ Technical note

The robust standard errors reported above are identical to those obtained by clustering on the panel variable idcode. Clustering on the panel variable produces an estimator of the VCE that is robust to cross-sectional heteroskedasticity and within-panel (serial) correlation that is asymptotically equivalent to that proposed by Arellano (1987). Although the example above applies the fixed-effects estimator, the robust and cluster–robust VCE estimators are also available for the random-effects estimator. Wooldridge (2020) and Arellano (2003) discuss these robust and cluster–robust VCE estimators for the fixed-effects and random-effects estimators. More details are available in *Methods and formulas*.

❑

▷ Example 4: Random-effects model

Refitting our log-wage model with the random-effects estimator, we obtain

```
. xtreg ln_w grade age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp tenure
> c.tenure#c.tenure 2.race not_smsa south, re theta

Random-effects GLS regression                   Number of obs     =     28,091
Group variable: idcode                          Number of groups  =      4,697

R-squared:                                      Obs per group:
     Within  = 0.1715                                         min =          1
     Between = 0.4784                                         avg =        6.0
     Overall = 0.3708                                         max =         15

                                                Wald chi2(10)     =    9244.74
corr(u_i, X) = 0 (assumed)                      Prob > chi2       =     0.0000

------------------- theta -------------------
  min      5%       median        95%      max
0.2520   0.2520     0.5499     0.7016   0.7206

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
       grade |   .0646499   .0017812    36.30   0.000     .0611589    .0681409
         age |   .0368059   .0031195    11.80   0.000     .0306918    .0429201
             |
 c.age#c.age |  -.0007133     .00005   -14.27   0.000    -.0008113   -.0006153
             |
     ttl_exp |   .0290208    .002422    11.98   0.000     .0242739    .0337678
             |
   c.ttl_exp#|
   c.ttl_exp |   .0003049   .0001162     2.62   0.009      .000077    .0005327
             |
      tenure |   .0392519   .0017554    22.36   0.000     .0358113    .0426925
             |
    c.tenure#|
    c.tenure |  -.0020035   .0001193   -16.80   0.000    -.0022373   -.0017697
             |
        race |
       Black |   -.053053   .0099926    -5.31   0.000    -.0726381   -.0334679
    not_smsa |  -.1308252   .0071751   -18.23   0.000    -.1448881   -.1167622
       south |  -.0868922   .0073032   -11.90   0.000    -.1012062   -.0725781
       _cons |   .2387207    .049469     4.83   0.000     .1417633    .3356781
-------------+----------------------------------------------------------------
     sigma_u |  .25790526
     sigma_e |  .29068923
         rho |  .44045273   (fraction of variance due to u_i)
------------------------------------------------------------------------------
```

According to the R^2's, this estimator performs worse within than the within fixed-effects estimator and worse between than the between estimator, as it must, and slightly better overall.

We estimate that σ_ν is 0.2579 and σ_ϵ is 0.2907 and, by assertion, assume that the correlation of ν and **x** is zero.

All that is known about the random-effects estimator is its asymptotic properties, so rather than reporting an F statistic for overall significance, `xtreg, re` reports a χ^2. Taken jointly, our coefficients are significant.

`xtreg, re` also reports a summary of the distribution of θ_i, an ingredient in the estimation of (4). θ is not a constant here because we observe women for unequal periods.

We estimate that schooling has a rate of return of 6.5% (compared with 6.1% between and no estimate within); that the increase of wages with age turns around at 25.8 years (compared with 26.9 between and 24.9 within); that total experience yet again increases wages increasingly; that the effect of job tenure turns around at 9.8 years (compared with 12.1 between and 9.1 within); that being black reduces wages by 5.3% (compared with 5.6% between and no estimate within); that living in a non-SMSA reduces wages 13.1% (compared with 18.6% between and 8.9% within); and that living in the South reduces wages 8.7% (compared with 9.9% between and 6.1% within).

◁

▷ Example 5: Random-effects model fit using MLE

We could also have fit this random-effects model with MLE:

```
. xtreg ln_w grade age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp tenure
> c.tenure#c.tenure 2.race not_smsa south, mle
Fitting constant-only model:
Iteration 0:  Log likelihood = -12663.954
Iteration 1:  Log likelihood = -12649.756
Iteration 2:  Log likelihood = -12649.614
Iteration 3:  Log likelihood = -12649.614
Fitting full model:
Iteration 0:  Log likelihood =  -8922.145
Iteration 1:  Log likelihood = -8853.6409
Iteration 2:  Log likelihood = -8853.4255
Iteration 3:  Log likelihood = -8853.4254
Random-effects ML regression                    Number of obs     =   28,091
Group variable: idcode                          Number of groups  =    4,697
Random effects u_i ~ Gaussian                   Obs per group:
                                                             min =        1
                                                             avg =      6.0
                                                             max =       15
                                                LR chi2(10)       =  7592.38
Log likelihood = -8853.4254                     Prob > chi2       =   0.0000

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
       grade |   .0646093   .0017372    37.19   0.000     .0612044    .0680142
         age |   .0368531   .0031226    11.80   0.000      .030733    .0429732
             |
 c.age#c.age |  -.0007132   .0000501   -14.24   0.000    -.0008113    -.000615
             |
     ttl_exp |   .0288196   .0024143    11.94   0.000     .0240877    .0335515
             |
   c.ttl_exp#|
   c.ttl_exp |    .000309   .0001163     2.66   0.008     .0000811    .0005369
             |
      tenure |   .0394371   .0017604    22.40   0.000     .0359868    .0428875
             |
    c.tenure#|
    c.tenure |  -.0020052   .0001195   -16.77   0.000    -.0022395   -.0017709
             |
        race |
       Black |  -.0533394   .0097338    -5.48   0.000    -.0724172   -.0342615
    not_smsa |  -.1323433   .0071322   -18.56   0.000    -.1463221   -.1183644
       south |  -.0875599   .0072143   -12.14   0.000    -.1016998   -.0734201
       _cons |   .2390837   .0491902     4.86   0.000     .1426727    .3354947
-------------+----------------------------------------------------------------
    /sigma_u |   .2485556   .0035017                      .2417863    .2555144
    /sigma_e |   .2918458    .001352                       .289208    .2945076
         rho |   .4204033   .0074828                      .4057959    .4351212
------------------------------------------------------------------------------
LR test of sigma_u=0: chibar2(01) = 7339.84            Prob >= chibar2 = 0.000
```

The estimates are nearly the same as those produced by `xtreg, re`—the GLS estimator. For instance, `xtreg, re` estimated the coefficient on `grade` to be 0.0646499, `xtreg, mle` estimated 0.0646093, and the ratio is $0.0646499/0.0646093 = 1.001$ to three decimal places. Similarly, the standard errors are nearly equal: $0.0017811/0.0017372 = 1.025$. Below we compare all 11 coefficients:

	Coefficient ratio			SE ratio		
Estimator	mean	min.	max.	mean	min.	max.
`xtreg, mle` (MLE)	1.	1.	1.	1.	1.	1.
`xtreg, re` (GLS)	.997	.987	1.007	1.006	.997	1.027

Example 6: Population-averaged model

We could also have fit this model with the population-averaged estimator:

```
. xtreg ln_w grade age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp tenure
> c.tenure#c.tenure 2.race not_smsa south, pa
Iteration 1:  Tolerance = .0310561
Iteration 2:  Tolerance = .00074898
Iteration 3:  Tolerance = .0000147
Iteration 4:  Tolerance = 2.880e-07
GEE population-averaged model                   Number of obs     =  28,091
Group variable: idcode                          Number of groups  =   4,697
Family: Gaussian                                Obs per group:
Link:    Identity                                             min =       1
Correlation: exchangeable                                     avg =     6.0
                                                              max =      15
                                                Wald chi2(10)     = 9598.89
Scale parameter = .1436709                      Prob > chi2       =  0.0000

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
       grade |   .0645427   .0016829    38.35   0.000     .0612442    .0678412
         age |    .036932   .0031509    11.72   0.000     .0307564    .0431076
             |
 c.age#c.age |  -.0007129   .0000506   -14.10   0.000    -.0008121   -.0006138
             |
     ttl_exp |   .0284878   .0024169    11.79   0.000     .0237508    .0332248
             |
   c.ttl_exp#|
   c.ttl_exp |   .0003158   .0001172     2.69   0.007      .000086    .0005456
             |
      tenure |   .0397468   .0017779    22.36   0.000     .0362621    .0432315
             |
    c.tenure#|
    c.tenure |   -.002008   .0001209   -16.61   0.000    -.0022449   -.0017711
             |
        race |
       Black |  -.0538314   .0094086    -5.72   0.000     -.072272   -.0353909
    not_smsa |  -.1347788   .0070543   -19.11   0.000    -.1486049   -.1209526
       south |  -.0885969   .0071132   -12.46   0.000    -.1025386   -.0746552
       _cons |   .2396286   .0491465     4.88   0.000     .1433034    .3359539
------------------------------------------------------------------------------
```

These results differ from those produced by xtreg, re and xtreg, mle. Coefficients are larger and standard errors smaller. xtreg, pa is simply another way to run the xtgee command. That is, we would have obtained the same output had we typed

```
. xtgee ln_w grade age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp
> tenure c.tenure#c.tenure 2.race not_smsa south
```
(*output omitted because it is the same as above*)

See [XT] **xtgee**. In the language of xtgee, the random-effects model corresponds to an exchangeable correlation structure and identity link, and xtgee also allows other correlation structures. Let's stay with the random-effects model, however. xtgee will also produce robust estimates of variance, and we refit this model that way by typing

```
. xtgee ln_w grade age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp
> tenure c.tenure#c.tenure 2.race not_smsa south, vce(robust)
```
(*output omitted, coefficients the same, standard errors different*)

In the previous example, we presented a table comparing xtreg, re with xtreg, mle. Below we add the results from the estimates shown and the ones we did with xtgee, vce(robust):

		Coefficient ratio			SE ratio		
Estimator		mean	min.	max.	mean	min.	max.
xtreg, mle	(MLE)	1.	1.	1.	1.	1.	1.
xtreg, re	(GLS)	.997	.987	1.007	1.006	.997	1.027
xtreg, pa	(PA)	1.060	.847	1.317	.853	.626	.986
xtgee, vce(robust)	(PA)	1.060	.847	1.317	1.306	.957	1.545

So, which are right? This is a real dataset, and we do not know. However, in example 2 in [XT] **xtreg postestimation**, we will present evidence that the assumptions underlying the xtreg, re and xtreg, mle results are not met.

◁

▷ Example 7: Correlated random-effects models

Below we use the same specification as in the fixed-effects estimation in example 3. Fixed-effects models cannot estimate coefficients for time-invariant regressors such as grade and race. If the effects of these variables are of interest, we can fit a correlated random-effects model by specifying the cre option.

```
. xtreg ln_w grade age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp tenure
> c.tenure#c.tenure 2.race not_smsa south, cre
note: grade omitted from xt_means because of collinearity.
note: 2.race omitted from xt_means because of collinearity.
```

```
Correlated random-effects regression            Number of obs     =     28,091
Group variable: idcode                          Number of groups  =      4,697

R-squared:                                      Obs per group:
     Within  = 0.1727                                         min =          1
     Between = 0.4899                                         avg =        6.0
     Overall = 0.3800                                         max =         15

                                                Wald chi2(10)     =    5984.86
corr(xit_vars*b, xt_means*γ) = 0.3941           Prob > chi2       =     0.0000

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
xit_vars     |
       grade |   .0610924   .0019795    30.86   0.000     .0572127    .0649722
         age |   .0359987   .0033749    10.67   0.000      .029384    .0426134
             |
 c.age#c.age |   -.000723   .0000531   -13.62   0.000     -.000827    -.000619
             |
     ttl_exp |   .0334668   .0029553    11.32   0.000     .0276745     .039259
             |
   c.ttl_exp#|
   c.ttl_exp |   .0002163   .0001273     1.70   0.089    -.0000332    .0004658
             |
      tenure |   .0357539   .0018424    19.41   0.000     .0321428     .039365
             |
    c.tenure#|
    c.tenure |  -.0019701   .0001246   -15.82   0.000    -.0022142    -.001726
             |
        race |
       Black |  -.0603623   .0102465    -5.89   0.000    -.0804451   -.0402795
    not_smsa |  -.0890108   .0094992    -9.37   0.000    -.1076289   -.0703926
       south |  -.0606309   .0108948    -5.57   0.000    -.0819844   -.0392774
       _cons |    .311389   .1282604     2.43   0.015     .0600033    .5627747
-------------+----------------------------------------------------------------
xt_means     |
       grade |          0  (omitted)
         age |  -.0021612    .009836    -0.22   0.826    -.0214394     .017117
             |
 c.age#c.age |   .0001041   .0001603     0.65   0.516    -.0002102    .0004184
             |
     ttl_exp |  -.0194687   .0065442    -2.97   0.003     -.032295   -.0066424
             |
   c.ttl_exp#|
   c.ttl_exp |   .0004933   .0003588     1.38   0.169    -.0002098    .0011965
             |
      tenure |   .0317887   .0064167     4.95   0.000     .0192122    .0443651
             |
    c.tenure#|
    c.tenure |  -.0007977   .0004349    -1.83   0.067      -.00165    .0000547
             |
        race |
       Black |          0  (omitted)
    not_smsa |  -.0982075   .0144899    -6.78   0.000    -.1266072   -.0698077
       south |  -.0405623   .0147223    -2.76   0.006    -.0694174   -.0117072
-------------+----------------------------------------------------------------
     sigma_u |  .25790526
     sigma_e |  .29068923
         rho |  .44045273   (fraction of variance due to u_i)
------------------------------------------------------------------------------
Mundlak test (xt_means = 0): chi2(8) = 151.6234           Prob > chi2 = 0.0000
```

The first section of the coefficient table, labeled `xit_vars`, reports coefficients and related statistics for both the time-variant and time-invariant variables. All the coefficients for time-variant regressors are numerically equivalent to those from `xtreg, fe`. This result is known as the Mundlak equivalence. As for the time-invariant regressors, we estimate that schooling has a rate of return of 6.1% (compared with 6.5% in the random-effects model) and being black reduces wages by 6.0% (compared with 5.3% in the random-effects model). As a result of the Mundlak equivalence, the correlated random-effects model has the same R^2 within, 0.1727, as the fixed-effects model. It also has a higher R^2 overall, 0.3800, because it includes as additional regressors the time-invariant independent variables and the panel means.

The second section of the coefficient table, labeled `xt_means`, reports coefficients and related statistics for the panel means of the time-varying variables. The panel means for time-invariant variables such as `grade` are omitted because the means are collinear with the original variables.

`xtreg, cre` also performs a Mundlak specification test to help decide between estimation using the random-effects model and estimation using the correlated random-effects model or fixed-effects model. This test consists of testing whether all the coefficients of the panel means are equal to zero. The results of the test can be found at the bottom of the output table. In this example, with a Wald test statistic of 151.6234 and a p-value of 0.0000, we reject the null hypothesis and favor this correlated random-effects model over the random-effects model.

◁

▷ Example 8: Controlling for high-dimensional categorical variables

Let's assume that in the labor market model we have been fitting, we would like to account for county effects. The United States, as of 2024, had approximately 3,200 counties. To fit a fixed-effects model, we could type

```
. xtreg ... i.county, fe
```

However, this would add the 3,200 county indicators as regressors in our model. If we are not interested in the coefficients on the county indicators, this creates an unnecessary computational burden. We can fit the model in a more efficient way by absorbing the county effects. To do this, we type

```
. xtreg ..., fe absorb(county)
```

Now, instead of estimating 3,200 coefficients, we absorb the county effects as well as the panel effects.

For illustrative purposes, we generate a county variable that mimics the description above.

```
. set seed 123
. generate county = runiformint(1,3200)
```

Then we fit the model.

```
. set seed 123
. generate county = runiformint(1,3200)
```

Below the command, we see iteration logs showing the maximum absolute difference (MAD) for the APM, which is the algorithm used to absorb the effects of panel and county variables. The smaller the MAD, the closer the APM is to the projection of the model fit by including indicator variables for all levels of `county`. Thus, the MAD should be as close to 0 as possible to absorb county and panel effects and obtain regression coefficients for the other predictors.

Following the iteration logs, we obtain output similar to the output from `xtreg, fe` when the `absorb()` option was not specified. However, we see an additional table that reports which variables were absorbed and the number of categories in each of the variables. No coefficients are reported for the counties. The county effects are absorbed.

Note that this output includes an estimate of σ_u and an F test that all u_is are 0; each of these requires additional computation time. If you are not interested in the estimate of σ_u, you can specify the `nosigmau` option. If you are not interested in σ_u or the F test that the u_is are 0, you can specify the `nouitest` option.

```
. xtreg ln_w age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp tenure
> c.tenure#c.tenure not_smsa south, fe
> absorb(county) nouitest
```

(output omitted)

```
Fixed-effects (within) regression               Number of obs     =     28,093
Group variable: idcode                          Number of groups  =      4,699

R-squared:                                      Obs per group:
     Within  = 0.2881                                         min =          1
     Between = 0.3479                                         avg =        6.0
     Overall = 0.2604                                         max =         15

                                                F(8, 20187)       =     533.12
```

Absorbed variable	Levels
county	3,200

ln_wage	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
age	.0359171	.0036425	9.86	0.000	.0287774	.0430567
c.age#c.age	-.0007245	.0000573	-12.65	0.000	-.0008368	-.0006122
ttl_exp	.0343766	.0031888	10.78	0.000	.0281262	.040627
c.ttl_exp#c.ttl_exp	.0001783	.0001374	1.30	0.194	-.0000911	.0004477
tenure	.0356387	.0019824	17.98	0.000	.0317531	.0395244
c.tenure#c.tenure	-.0019394	.0001339	-14.48	0.000	-.0022019	-.0016769
not_smsa	-.0909885	.0102319	-8.89	0.000	-.1110438	-.0709331
south	-.0514181	.0117182	-4.39	0.000	-.0743867	-.0284494
_cons	1.034046	.0522015	19.81	0.000	.9317273	1.136366
sigma_e	.29022084					

Specifying `nouitest` can be particularly helpful in improving speed when absorbing categorical variables with many levels.

◁

Stored results

`xtreg, re` stores the following in `e()`:

Scalars	
`e(N)`	number of observations
`e(N_g)`	number of groups
`e(df_m)`	model degrees of freedom
`e(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(g_max)`	largest group size
`e(Tcon)`	1 if T is constant
`e(sigma)`	square root of the sum of `e(sigma_e)`^{2} and `e(sigma_u)`^{2}
`e(sigma_u)`	panel-level standard deviation
`e(sigma_e)`	standard deviation of ϵ_{it}
`e(r2_w)`	R^2 for within model
`e(r2_o)`	R^2 for overall model
`e(r2_b)`	R^2 for between model
`e(N_clust)`	number of clusters
`e(chi2)`	χ^2
`e(p)`	p-value for model test
`e(rho)`	ρ
`e(thta_min)`	minimum θ
`e(thta_5)`	θ, 5th percentile
`e(thta_50)`	θ, 50th percentile
`e(thta_95)`	θ, 95th percentile
`e(thta_max)`	maximum θ
`e(rmse)`	root mean squared error of GLS regression
`e(Tbar)`	harmonic mean of group sizes
`e(rank)`	rank of `e(V)`
Macros	
`e(cmd)`	`xtreg`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(ivar)`	variable denoting groups
`e(model)`	`re`
`e(clustvar)`	name of cluster variable
`e(chi2type)`	`Wald`; type of model χ^2 test
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. err.
`e(sa)`	`sa`, if specified
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(marginsnotok)`	predictions disallowed by `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`
Matrices	
`e(b)`	coefficient vector
`e(bf)`	coefficient vector for fixed-effects model
`e(theta)`	θ
`e(V)`	variance–covariance matrix of the estimators
`e(VCEf)`	VCE for fixed-effects model
Functions	
`e(sample)`	marks estimation sample

In addition to the above, the following is stored in r():

Matrices

r(table)	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

xtreg, cre stores the following in e():

Scalars

e(N)	number of observations
e(N_g)	number of groups
e(df_m)	model degrees of freedom
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(Tcon)	1 if T_i is constant
e(sigma)	square root of the sum of e(sigma_e)^2 and e(sigma_u)^2
e(sigma_u)	panel-level standard deviation
e(sigma_e)	standard deviation of ϵ_{it}
e(r2_w)	R^2 for within model
e(r2_o)	R^2 for overall model
e(r2_b)	R^2 for between model
e(N_clust)	number of clusters
e(chi2)	χ^2
e(p)	p-value for model test
e(rho)	ρ
e(rmse)	root mean squared error of Mundlak GLS regression
e(Tbar)	harmonic mean of group sizes
e(rank)	rank of e(V)
e(corr)	corr($\mathbf{x}_{it}\boldsymbol{\beta}$, $\overline{\mathbf{x}}_i\boldsymbol{\gamma}$)
e(chi2_mundlak)	Mundlak test χ^2 statistic
e(p_mundlak)	Mundlak test p-value
e(df_mundlak)	Mundlak test degrees of freedom

Macros

e(cmd)	xtreg
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(covariates)	list of covariates
e(ivar)	variable denoting groups
e(model)	cre
e(clustvar)	name of cluster variable
e(chi2type)	Wald; type of model χ^2 test
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(sa)	sa, if specified
e(properties)	b V
e(predict)	program used to implement predict
e(estat_cmd)	program used to implement estat
e(marginsok)	predictions allowed by margins
e(marginsnotok)	predictions disallowed by margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved

Matrices

e(b)	coefficient vector
e(bf)	coefficient vector for fixed-effects model

`e(V)`	variance–covariance matrix of the estimators
`e(VCEf)`	VCE for fixed-effects model

Functions

`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

`xtreg, be` stores the following in `e()`:

Scalars

`e(N)`	number of observations
`e(N_g)`	number of groups
`e(mss)`	model sum of squares
`e(df_m)`	model degrees of freedom
`e(rss)`	residual sum of squares
`e(df_r)`	residual degrees of freedom
`e(ll)`	log likelihood
`e(ll_0)`	log likelihood, constant-only model
`e(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(g_max)`	largest group size
`e(Tcon)`	`1` if T is constant
`e(r2)`	R^2
`e(r2_a)`	adjusted R^2
`e(r2_w)`	R^2 for within model
`e(r2_o)`	R^2 for overall model
`e(r2_b)`	R^2 for between model
`e(F)`	F statistic
`e(p)`	p-value for model test
`e(rmse)`	root mean squared error
`e(Tbar)`	harmonic mean of group sizes
`e(rank)`	rank of `e(V)`

Macros

`e(cmd)`	`xtreg`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(ivar)`	variable denoting groups
`e(model)`	`be`
`e(typ)`	`WLS`, if `wls` specified
`e(title)`	title in estimation output
`e(vce)`	*vcetype* specified in `vce()`
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(marginsok)`	predictions allowed by `margins`
`e(marginsnotok)`	predictions disallowed by `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`

Matrices

`e(b)`	coefficient vector
`e(V)`	variance–covariance matrix of the estimators

Functions	
e(sample)	marks estimation sample

In addition to the above, the following is stored in r():

Matrices	
r(table)	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

xtreg, fe stores the following in e():

Scalars	
e(N)	number of observations
e(N_g)	number of groups
e(mss)	model sum of squares
e(df_m)	model degrees of freedom
e(rss)	residual sum of squares
e(df_r)	residual degrees of freedom
e(tss)	total sum of squares
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(Tcon)	1 if T is constant
e(corr)	corr(u_i, $\mathbf{x}_{it}\boldsymbol{\beta}$)
e(sigma)	square root of the sum of e(sigma_e)^2 and e(sigma_u)^2
e(sigma_u)	panel-level standard deviation
e(sigma_e)	standard deviation of ϵ_{it}
e(r2)	R^2
e(r2_a)	adjusted R^2
e(r2_w)	R^2 for within model
e(r2_o)	R^2 for overall model
e(r2_b)	R^2 for between model
e(ll)	log likelihood
e(ll_0)	log likelihood, constant-only model
e(N_clust)	number of clusters
e(k_absorb)	total number of absorbed categories, including those of the panel variable
e(rho)	ρ
e(F)	F statistic
e(F_f)	F statistic for test of $u_i = 0$
e(p)	p-value for model test
e(p_f)	p-value for test of $u_i = 0$
e(df_a)	degrees of freedom for absorbed panel effect
e(df_b)	numerator degrees of freedom for F statistic
e(rmse)	root mean squared error
e(Tbar)	mean of group sizes
e(rank)	rank of e(V)
e(converged)	1 if APM converged, 0 otherwise
Macros	
e(cmd)	xtreg
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(ivar)	variable denoting groups
e(absvar)	names of absorbed variables, including the panel variable
e(apm)	alternating projection method

`e(model)`	`fe`
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(clustvar)`	names of cluster variables
`e(cluster#)`	cluster combination #
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. err.
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(marginsnotok)`	predictions disallowed by `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`
Matrices	
`e(b)`	coefficient vector
`e(V)`	variance–covariance matrix of the estimators
`e(V_modelbased)`	model-based variance
`e(adj_df)`	adjusted degrees of freedom when `vce(hc2, dfadjust)` is specified
`e(kcluster)`	cluster sizes, multiway clustering
`e(kabsorb)`	number of levels for each absorbed variable, including the panel variable
`e(dfabsorb)`	adjusted degrees of freedom for each absorbed variable
`e(ksingle)`	number of singletons for each absorbed variable
Functions	
`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices	
`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

`xtreg, mle` stores the following in `e()`:

Scalars	
`e(N)`	number of observations
`e(N_g)`	number of groups
`e(df_m)`	model degrees of freedom
`e(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(g_max)`	largest group size
`e(sigma_u)`	panel-level standard deviation
`e(sigma_e)`	standard deviation of ϵ_{it}
`e(ll)`	log likelihood
`e(ll_0)`	log likelihood, constant-only model
`e(ll_c)`	log likelihood, comparison model
`e(N_clust)`	number of clusters
`e(chi2)`	χ^2
`e(chi2_c)`	χ^2 for comparison test
`e(p)`	p-value for model test
`e(rho)`	ρ
`e(rank)`	rank of `e(V)`
Macros	
`e(cmd)`	`xtreg`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(ivar)`	variable denoting groups

e(model)	ml
e(wtype)	weight type
e(wexp)	weight expression
e(title)	title in estimation output
e(clustvar)	name of cluster variable
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(chi2type)	Wald or LR; type of model χ^2 test
e(chi2_ct)	Wald or LR; type of model χ^2 test corresponding to e(chi2_c)
e(distrib)	Gaussian; the distribution of the random effects
e(properties)	b V
e(predict)	program used to implement predict
e(marginsnotok)	predictions disallowed by margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved

Matrices

e(b)	coefficient vector
e(V)	variance–covariance matrix of the estimators

Functions

e(sample)	marks estimation sample

In addition to the above, the following is stored in r():

Matrices

r(table)	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

xtreg, pa stores the following in e():

Scalars

e(N)	number of observations
e(N_g)	number of groups
e(df_m)	model degrees of freedom
e(chi2)	χ^2
e(p)	p-value for model test
e(df_pear)	degrees of freedom for Pearson χ^2
e(chi2_dev)	χ^2 test of deviance
e(chi2_dis)	χ^2 test of deviance dispersion
e(deviance)	deviance
e(dispers)	deviance dispersion
e(phi)	scale parameter
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(rank)	rank of e(V)
e(tol)	target tolerance
e(dif)	achieved tolerance
e(rc)	return code

Macros

e(cmd)	xtgee
e(cmd2)	xtreg
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(ivar)	variable denoting groups
e(tvar)	variable denoting time within groups

e(model)	pa
e(family)	Gaussian
e(link)	identity; link function
e(corr)	correlation structure
e(scale)	x2, dev, phi, or #; scale parameter
e(wtype)	weight type
e(wexp)	weight expression
e(offset)	linear offset variable
e(chi2type)	Wald; type of model χ^2 test
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(rgf)	rgf, if rgf specified
e(nmp)	nmp, if specified
e(properties)	b V
e(predict)	program used to implement predict
e(marginsnotok)	predictions disallowed by margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved
Matrices	
e(b)	coefficient vector
e(R)	estimated working correlation matrix
e(V)	variance–covariance matrix of the estimators
e(V_modelbased)	model-based variance
Functions	
e(sample)	marks estimation sample

In addition to the above, the following is stored in r():

Matrices	
r(table)	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

The model to be fit is

$$y_{it} = \alpha + \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i + \epsilon_{it}$$

for $i = 1, \dots, n$ and, for each i, $t = 1, \dots, T$, of which T_i periods are actually observed.

Methods and formulas are presented under the following headings:

xtreg, fe

`xtreg, fe` produces estimates by running OLS on

$$(y_{it} - \overline{y}_i + \overline{\overline{y}}) = \alpha + (\mathbf{x}_{it} - \overline{\mathbf{x}}_i + \overline{\overline{\mathbf{x}}})\boldsymbol{\beta} + (\epsilon_{it} - \overline{\epsilon}_i + \overline{\nu}) + \overline{\overline{\epsilon}}$$

where $\overline{y}_i = \sum_{t=1}^{T_i} y_{it}/T_i$, and similarly, $\overline{\overline{y}} = \sum_i \sum_t y_{it}/(nT_i)$. The conventional covariance matrix of the estimators is adjusted for the extra $n-1$ estimated means, so results are the same as using OLS on (1) to estimate ν_i directly. Specifying `vce(robust)` or `vce(cluster` *clustvar*`)` causes the Huber/White/sandwich VCE estimator to be calculated for the coefficients estimated in this regression. See [P] **_robust**, particularly *Introduction* and *Methods and formulas*. Wooldridge (2020) and Arellano (2003) discuss this application of the Huber/White/sandwich VCE estimator. As discussed by Wooldridge (2020), Stock and Watson (2008), and Arellano (2003), specifying `vce(robust)` is equivalent to specifying `vce(cluster` *panelvar*`)`, where *panelvar* is the variable that identifies the panels.

Clustering on the panel variable produces a consistent VCE estimator when the disturbances are not identically distributed over the panels or there is serial correlation in ϵ_{it}.

The cluster–robust VCE estimator requires that there are many clusters and the disturbances are uncorrelated across the clusters. The panel variable must be nested within the cluster variable because of the within-panel correlation induced by the within transform. The panel-nesting restriction is also enforced for multiway clustering, `vce(cluster` *clustvarlist*`)`, which includes all combinations of the cluster variables. See *Methods and formulas* in [R] **regress** for computational details of multiway clustering.

`xtreg, fe` with `vce(hc2` [*clustvar*] [`, dfadjust`]`)` specifies an alternative bias correction for the cluster–robust variance calculation. If *clustvar* is not specified, *panelvar* is used in the VCE computation. See *Robust calculation for regress* in *Methods and formulas* in [R] **regress** for a description of the VCE produced by `vce(hc2)` and the adjusted degrees-of-freedom computations.

From the estimates $\widehat{\alpha}$ and $\widehat{\boldsymbol{\beta}}$, estimates u_i of ν_i are obtained as $u_i = \overline{y}_i - \widehat{\alpha} - \overline{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}$. Reported from the calculated u_i are its standard deviation and its correlation with $\overline{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}$. Reported as the standard deviation of e_{it} is the regression's estimated root mean squared error, s, which is adjusted (as previously stated) for the $n-1$ estimated means.

Reported as R^2 within is the R^2 from the mean-deviated regression.

Reported as R^2 between is corr$(\overline{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}, \overline{y}_i)^2$.

Reported as R^2 overall is corr$(\mathbf{x}_{it}\widehat{\boldsymbol{\beta}}, y_{it})^2$.

Absorbed variables with xtreg, fe

Suppose we have h categorical variables, $C_1, \dots, C_h$, that we would like to include as controls in our fixed-effects regression. Let m_k be the number of categories for variable C_k, and let $\mathbf{d}_{k(it)}$ be the $1 \times m_k$ indicator vector for panel i at time t. Thus, vector $\mathbf{d}_{k(it)}$ is a unit vector with a 1 in the category of variable C_k in panel i at time t, and 0 everywhere else. Let $\mathbf{D}_{k(i)}$ denote the $T_i \times m_k$ matrix of indicators for variable C_k and panel i:

$$\mathbf{D}_{k(i)} = \begin{pmatrix} \mathbf{d}_{k(i1)} \\ \mathbf{d}_{k(i2)} \\ \vdots \\ \mathbf{d}_{k(iT_i)} \end{pmatrix}$$

We would like to fit the model

$$y_{it} = \alpha + \mathbf{x}_{it}\boldsymbol{\beta} + \mathbf{d}_{1(it)}\gamma_1 + \cdots + \mathbf{d}_{h(it)}\gamma_h + \nu_i + \epsilon_{it}$$

for $t = 1, \dots, T_i$, and $i = 1, \dots, n$.

Estimating the $\boldsymbol{\gamma}$ coefficients in the same manner as the $\boldsymbol{\beta}$ coefficients can be computationally expensive if the categorical variables have many levels. Instead, we absorb the effects of these categorical variables along with the panel effects.

When absorbing categorical variables, regression coefficients are computed by first running an APM and then applying Frisch–Waugh–Lovell's theorem (see Hansen [2022, 82]). The APM is used to obtain the residuals of y_{it} and $\mathbf{x}_{it}$ after regressing them (separately) on all the categorical variables. The speed of the APM is further improved by using a conjugate gradient acceleration technique (Hernández-Ramos, Escalante, and Raydan 2011).

Two projection methods are available: Halperin (the default) and Cimmino. You can specify which to use with the `halperin` or `cimmino` suboption within the `absorb()` option. We describe both methods below.

Let $\mathbf{D}_k$ be the $N \times m_k$ matrix of indicators for the kth absorbed variable (where $N = \sum_i^n T_i$):

$$\mathbf{D}_k = \begin{pmatrix} \mathbf{D}_{k(1)} \\ \mathbf{D}_{k(2)} \\ \vdots \\ \mathbf{D}_{k(n)} \end{pmatrix}$$

The orthonormal projection matrix for C_k is therefore given by $\mathbf{P}_k = \mathbf{D}_k(\mathbf{D}_k'\mathbf{D}_k)^{-1}\mathbf{D}_k'$. By convention, we define the first absorbed variable ($k = 1$) as the panel variable. Let $\mathbf{y}$ be the $N \times 1$ vector with the values of the dependent variable in the sample. Similarly, let $\mathbf{X}$ be the matrix with the values of the covariates. Thus, the product $\overline{\mathbf{y}}_k = \mathbf{P}_k\mathbf{y}$ is the projection of the dependent variable onto the column space of $\mathbf{D}_k$. That is, $\overline{\mathbf{y}}_k$ is the $N \times 1$ vector containing the (repeated) means of y_{it} for each level of C_k, in the order that these levels appear in the sample. The same projection can be applied to the columns of covariate matrix $\mathbf{X}$, that is, $\overline{\mathbf{X}}_k = \mathbf{P}_k\mathbf{X}$.

The Halperin algorithm first sets $\tilde{\mathbf{y}}_0^{(1)} = \mathbf{y}$ and loops over the h absorbed variables computing projection residuals $\tilde{\mathbf{y}}_k^{(1)} = (\mathbf{I} - \mathbf{P_k})\tilde{\mathbf{y}}_{k-1}^{(1)}$, for $k = 1, \dots, h$. Then, it repeats the loop with $\tilde{\mathbf{y}}_0^{(j+1)} = \tilde{\mathbf{y}}_h^{(j)}$ until convergence at $j = \jmath$. Convergence is declared when $|\tilde{\mathbf{y}}_h^{(j)} - \tilde{\mathbf{y}}_0^{(j)}|$ is less than the specified tolerance, where the matrix norm $|\cdot|$ is defined as the largest entry in absolute value. The same computations are applied to the columns of covariate matrix $\mathbf{X}$. On convergence, the overall mean $\overline{\overline{y}}$ is added to $\tilde{\mathbf{y}}_h^{(\jmath)}$, and the vector of means $\overline{\overline{\mathbf{x}}}$ is added to the columns of $\widetilde{\mathbf{X}}_h^{(\jmath)}$.

The Cimmino algorithm first sets $\tilde{\mathbf{y}}^{(0)} = \mathbf{y}$ and then iteratively computes

$$\tilde{\mathbf{y}}^{(j+1)} = \left(\mathbf{I} - \frac{1}{h}\sum_{k=1}^{h}\mathbf{P}_k\right)\tilde{\mathbf{y}}^{(j)}$$

until convergence at $j + 1 = \jmath$. It then repeats the process for covariance matrix $\mathbf{X}$ and adds the overall means, $\overline{\overline{y}}$ and $\overline{\overline{\mathbf{x}}}$, to $\tilde{\mathbf{y}}^{(\jmath)}$ and to the columns of $\widetilde{\mathbf{X}}^{(\jmath)}$, respectively.

Efficient computation of the projection $\mathbf{Py}$ can be done in Mata without generating the $N \times N$ matrix $\mathbf{P}$ directly (see [M-5] **panelsum()**, for instance).

An estimate of σ_u, the standard deviation of the panel effects, is obtained by applying the APM algorithm to $\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}$. A discussion of estimating the panel effects can be found in *Remarks and examples* of [XT] **xtreg postestimation**.

See *Methods and formulas* in [R] **areg** for a description of the method used to adjust the degrees of freedom to account for collinearity among absorbed variables when the `dfabsorb` option is specified.

For a discussion of controlling for high-dimensional categorical variables in linear and nonlinear models, see Correia (2016) and Correia, Guimarães, and Zylkin (2020).

xtreg, be

`xtreg, be` fits the following model:

$$\overline{y}_i = \alpha + \overline{\mathbf{x}}_i \boldsymbol{\beta} + \nu_i + \overline{\epsilon}_i$$

Estimation is via OLS unless T_i is not constant and the `wls` option is specified. Otherwise, the estimation is performed via WLS. The estimates and conventional VCE are obtained from `regress` for both cases, but for WLS, `[aweight=`T_i`]` is specified.

Reported as R^2 between is the R^2 from the fitted regression.

Reported as R^2 within is $\text{corr}\{(\mathbf{x}_{it} - \overline{\mathbf{x}}_i)\widehat{\boldsymbol{\beta}}, y_{it} - \overline{y}_i\}^2$.

Reported as R^2 overall is $\text{corr}(\mathbf{x}_{it}\widehat{\boldsymbol{\beta}}, y_{it})^2$.

xtreg, re

The key to the random-effects estimator is the GLS transform. Given estimates of the idiosyncratic component, $\widehat{\sigma}_e^2$, and the individual component, $\widehat{\sigma}_u^2$, the GLS transform of a variable z for the random-effects model is

$$z_{it}^* = z_{it} - \widehat{\theta}_i \overline{z}_i$$

where $\overline{z}_i = 1/T_i \sum_{t=1}^{T_i} z_{it}$ and

$$\hat{\theta}_i = 1 - \sqrt{\frac{\widehat{\sigma}_e^2}{T_i \widehat{\sigma}_u^2 + \widehat{\sigma}_e^2}}$$

Given an estimate of $\widehat{\theta}_i$, one transforms the dependent and independent variables, and then the coefficient estimates and the conventional variance–covariance matrix come from an OLS regression of y_{it}^* on $\mathbf{x}_{it}^*$ and the transformed constant $1 - \widehat{\theta}_i$. Specifying `vce(robust)` or `vce(cluster` *clustvar*`)` causes the Huber/White/sandwich VCE estimator to be calculated for the coefficients estimated in this regression. See [P] **_robust**; in particular, see *Introduction* and *Methods and formulas*. Wooldridge (2020) and Arellano (2003) discuss this application of the Huber/White/sandwich VCE estimator. As discussed by Wooldridge (2020), Stock and Watson (2008), and Arellano (2003), specifying `vce(robust)` is equivalent to specifying `vce(cluster` *panelvar*`)`, where *panelvar* is the variable that identifies the panels.

Clustering on the panel variable produces a consistent VCE estimator when the disturbances are not identically distributed over the panels or there is serial correlation in ϵ_{it}.

The cluster–robust VCE estimator requires that there are many clusters and the disturbances are uncorrelated across the clusters. The panel variable must be nested within the cluster variable because of the within-panel correlation that is generally induced by the random-effects transform when there is heteroskedasticity or within-panel serial correlation in the idiosyncratic errors.

Stata has two implementations of the Swamy–Arora method for estimating the variance components. They produce the same results in balanced panels and share the same estimator of σ_e^2. However, the two methods differ in their estimator of σ_u^2 in unbalanced panels. We call the first $\widehat{\sigma}_{u\overline{T}}^2$ and the second $\widehat{\sigma}_{\text{uSA}}^2$. Both estimators are consistent; however, $\widehat{\sigma}_{\text{uSA}}^2$ has a more elaborate adjustment for small samples than $\widehat{\sigma}_{u\overline{T}}^2$. (See Baltagi [2013], Baltagi and Chang [1994], and Swamy and Arora [1972] for derivations of these methods.)

Both methods use the same function of within residuals to estimate the idiosyncratic error component σ_e. Specifically,

$$\widehat{\sigma}_e^2 = \frac{\sum_{i=1}^n \sum_{t=1}^{T_i} e_{it}^2}{N - n - K + 1}$$

where

$$e_{it} = (y_{it} - \overline{y}_i + \overline{\overline{y}}) - \widehat{\alpha}_w - (\mathbf{x}_{it} - \overline{\mathbf{x}}_i + \overline{\overline{\mathbf{x}}})\widehat{\boldsymbol{\beta}}_w$$

and $\widehat{\alpha}_w$ and $\widehat{\boldsymbol{\beta}}_w$ are the within estimates of the coefficients and $N = \sum_{i=1}^n T_i$. After passing the within residuals through the within transform, only the idiosyncratic errors are left.

The default method for estimating σ_u^2 is

$$\widehat{\sigma}_{u\overline{T}}^2 = \max\left\{0, \frac{\text{SSR}_b}{n-K} - \frac{\widehat{\sigma}_e^2}{\overline{T}}\right\}$$

where

$$\text{SSR}_b = \sum_{i=1}^n \left(\overline{y}_i - \widehat{\alpha}_b - \overline{\mathbf{x}}_i \widehat{\boldsymbol{\beta}}_b\right)^2$$

$\widehat{\alpha}_b$ and $\widehat{\boldsymbol{\beta}}_b$ are coefficient estimates from the between regression and $\overline{T}$ is the harmonic mean of T_i:

$$\overline{T} = \frac{n}{\sum_{i=1}^n \frac{1}{T_i}}$$

This estimator is consistent for σ_u^2 and is computationally less expensive than the second method. The sum of squared residuals from the between model estimate a function of both the idiosyncratic component and the individual component. Using our estimator of σ_e^2, we can remove the idiosyncratic component, leaving only the desired individual component.

The second method is the Swamy–Arora method for unbalanced panels derived by Baltagi and Chang (1994), which has a more precise small-sample adjustment. Using this method,

$$\widehat{\sigma}_{\text{uSA}}^2 = \max\left\{0, \frac{\text{SSR}_b^* - (n-K)\widehat{\sigma}_e^2}{N - c_{\text{tr}}}\right\}$$

where

$$\text{SSR}_b^* = \sum_{i=1}^n T_i \left(\overline{y}_i - \widehat{\alpha}_b - \overline{\mathbf{x}}_i \widehat{\boldsymbol{\beta}}_b\right)^2$$

$$c_{\text{tr}} = \text{trace}\left\{(\mathbf{X}'\mathbf{P}\mathbf{X})^{-1}\mathbf{X}'\mathbf{Z}\mathbf{Z}'\mathbf{X}\right\}$$

$$\mathbf{P} = \text{diag}\left\{\left(\frac{1}{T_i}\right)\boldsymbol{\iota}_{T_i}\boldsymbol{\iota}_{T_i}'\right\}$$

$$\mathbf{Z} = \text{diag}\left[\boldsymbol{\iota}_{T_i}\right]$$

$\mathbf{X}$ is the $N \times K$ matrix of covariates, including the constant, and $\boldsymbol{\iota}_{T_i}$ is a $T_i \times 1$ vector of ones.

The estimated coefficients $(\widehat{\alpha}, \widehat{\boldsymbol{\beta}})$ and their estimated covariance matrix $\widehat{\mathbf{V}}$ are reported together with the previously calculated quantities $\widehat{\sigma}_e$ and $\widehat{\sigma}_u$. The standard deviation of $\nu_i + e_{it}$ is calculated as $\sqrt{\widehat{\sigma}_e^2 + \widehat{\sigma}_u^2}$.

Reported as R^2 between is $\text{corr}(\overline{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}, \overline{y}_i)^2$.

Reported as R^2 within is $\text{corr}\{(\mathbf{x}_{it} - \overline{\mathbf{x}}_i)\widehat{\boldsymbol{\beta}}, y_{it} - \overline{y}_i\}^2$.

Reported as R^2 overall is $\text{corr}(\mathbf{x}_{it}\widehat{\boldsymbol{\beta}}, y_{it})^2$.

xtreg, cre

Let's modify notation slightly to better explain estimation using correlated random effects. Denote by $\mathbf{z}_i$ the time-invariant regressors in the model, by $\mathbf{x}_{it}$ the regressors that vary across time, and by $\overline{\mathbf{x}}_i$ the average of $\mathbf{x}_{it}$ in panel i (using only observations in the estimation sample).

`xtreg, cre` fits a correlated random-effects model by fitting a random-effects regression of y_{it} on $\mathbf{x}_{it}$, $\overline{\mathbf{x}}_i$, $\mathbf{z}_i$, and a constant term. This is equivalent to fitting an OLS regression of $(y_{it} - \widehat{\theta}_i\overline{y}_i)$ on $(\mathbf{x}_{it} - \widehat{\theta}_i\overline{\mathbf{x}}_i)$, $(1 - \widehat{\theta}_i)\overline{\mathbf{x}}_i$, $(1 - \widehat{\theta}_i)\mathbf{z}_i$, and the transformed constant term $(1 - \widehat{\theta}_i)$. The GLS transformation term, $\widehat{\theta}_i$, is a function of T_i and the variance estimators $\widehat{\sigma}_u^2$ and $\widehat{\sigma}_e^2$. Option `sa` requests that the small-sample Swamy–Arora estimator be used for σ_u^2 instead of the default. The formulas for $\widehat{\theta}_i$, $\widehat{\sigma}_e^2$, and the available estimators for σ_u^2 can be found in *xtreg, re*.

Specifying `vce(cluster` *clustvar*`)` requests that the cluster–robust VCE estimator be used in the underlying OLS regression with the transformed variables. The cluster–robust VCE estimator requires that i) the panel variable be nested in the clusters defined by *clustvar*, ii) there be many clusters, and iii) the idiosyncratic errors, ϵ_{it}, be uncorrelated between clusters. Specifying `vce(robust)` is equivalent to specifying `vce(cluster` *panelvar*`)`. See [P] **_robust** for more on the computation of standard errors in Stata and Wooldridge (2020) and Arellano (2003) for a discussion on the application of the Huber/White VCE estimator for panel data.

For balanced panels, Mundlak (1978) showed that the coefficients for $\mathbf{x}_{it}$ obtained from `xtreg, cre` are numerically equivalent to those obtained from a fixed-effects regression of y_{it} on $\mathbf{x}_{it}$. Wooldridge (2019) extended this equivalence result to unbalanced panels. Note that, unlike fixed-effects estimation, `xtreg, cre` returns coefficients for time-invariant regressors.

`xtreg, cre` also performs a Mundlak specification test to choose between random-effects estimation and correlated random-effects or fixed-effects estimation. This is an alternative to the Hausman test for comparing fixed-effects and random-effects models, but unlike the Hausman test, the Mundlak test can be used even with a cluster–robust VCE. The Mundlak test is obtained as a test of $H_0 : \gamma = 0$ in the correlated random-effects model,

$$y_{it} = \alpha + \mathbf{x}_{it}\boldsymbol{\beta} + \overline{\mathbf{x}}_i\gamma + \mathbf{z}_i\boldsymbol{\xi} + \nu_i + \epsilon_{it}$$

The overall model Wald test reported in the header of the `xtreg, cre` output is a test of $H_0 : \beta = \xi = 0$.

Reported as R^2 between is $\text{corr}(\overline{\mathbf{x}}_i(\widehat{\boldsymbol{\beta}} + \widehat{\gamma}) + z_i\widehat{\boldsymbol{\xi}}, \overline{y}_i)^2$.

Reported as R^2 within is $\text{corr}\{(\mathbf{x}_{it} - \overline{\mathbf{x}}_i)\widehat{\boldsymbol{\beta}}, y_{it} - \overline{y}_i\}^2$.

Reported as R^2 overall is $\text{corr}(\mathbf{x}_{it}\widehat{\boldsymbol{\beta}} + \overline{\mathbf{x}}_i\widehat{\gamma} + z_i\widehat{\boldsymbol{\xi}}, y_{it})^2$.

margins, when used after xtreg, cre, recalculates the panel means if working outside the original estimation sample. Once these panel means have been recalculated, however, margins does not recompute the panel means when it adjusts covariate values in the computations of adjusted means, adjusted predictions, and average marginal or partial effects. See Abrevaya and Hsu (2021) for more on estimating partial effects in models with correlated random-effects.

xtreg, mle

The log likelihood for the ith unit is

$$l_i = -\frac{1}{2}\left(\frac{1}{\sigma_e^2}\left[\sum_{t=1}^{T_i}(y_{it}-\mathbf{x}_{it}\boldsymbol{\beta})^2 - \frac{{\sigma_u}^2}{T_i\sigma_u^2+\sigma_e^2}\left\{\sum_{t=1}^{T_i}(y_{it}-\mathbf{x}_{it}\boldsymbol{\beta})\right\}^2\right] + \ln\left(T_i\frac{\sigma_u^2}{\sigma_e^2}+1\right) + T_i\ln(2\pi\sigma_e^2)\right)$$

The mle and re options yield essentially the same results, except when total $N=\sum_i T_i$ is small (200 or less) and the data are unbalanced.

Similarly to xtreg, fe and xtreg, re, specifying vce(robust) or vce(cluster *clustvar*) causes the Huber/White/sandwich VCE estimator to be calculated for the estimated parameters in this regression.

Specifying vce(robust) is equivalent to specifying vce(cluster *panelvar*), where *panelvar* is the variable that identifies the panels.

Clustering on the panel variable produces a consistent VCE estimator when the disturbances are not identically distributed over the panels or there is serial correlation in it.

The cluster–robust VCE estimator requires that there are many clusters and the disturbances are uncorrelated across the clusters. The panel variable must be nested within the cluster variable because of the within-panel correlation that is generally induced by the random-effects transform when there is heteroskedasticity or within-panel serial correlation in the idiosyncratic errors.

xtreg, pa

See [XT] **xtgee** for details on the methods and formulas used to calculate the population-averaged model using a generalized estimating equations approach.

Acknowledgments

We thank Richard Goldstein, who wrote the first draft of the routine that fits random-effects regressions, Badi Baltagi of the Department of Economics at Syracuse University, and Manuelita Ureta of the Department of Economics at Texas A&M University, who assisted us in working our way through the literature. We also thank Sergio Correia at the Federal Reserve Board Paulo Guimarães at the Bank of Portugal, and Fernando Rios-Avila at the Levy Economics Institute at Bard College for their work on controlling for high-dimensional categorical variables.

References

Abrevaya, J., and Y.-C. Hsu. 2021. Partial effects in non-linear panel data models with correlated random effects. *Econometrics Journal* 24: 519–535. https://doi.org/10.1093/ectj/utab004.

Alejo, J., A. F. Galvao, G. Montes-Rojas, and W. Sosa-Escudero. 2015. Tests for normality in linear panel-data models. *Stata Journal* 15: 822–832.

Allison, P. D. 2009. *Fixed Effects Regression Models*. Newbury Park, CA: Sage.

Andrews, M. J., T. Schank, and R. Upward. 2006. Practical fixed-effects estimation methods for the three-way error-components model. *Stata Journal* 6: 461–481.

Arellano, M. 1987. Computing robust standard errors for within-groups estimators. *Oxford Bulletin of Economics and Statistics* 49: 431–434. https://doi.org/10.1111/j.1468-0084.1987.mp49004006.x.

———. 2003. *Panel Data Econometrics*. Oxford: Oxford University Press. https://doi.org/10.1093/0199245282.001.0001.

Baltagi, B. H. 1985. Pooling cross-sections with unequal time-series lengths. *Economics Letters* 18: 133–136. https://doi.org/10.1016/0165-1765(85)90167-3.

———. 2009. *A Companion to Econometric Analysis of Panel Data*. Chichester, UK: Wiley.

———. 2013. *Econometric Analysis of Panel Data*. 5th ed. Chichester, UK: Wiley.

Baltagi, B. H., and Y.-J. Chang. 1994. Incomplete panels: A comparative study of alternative estimators for the unbalanced one-way error component regression model. *Journal of Econometrics* 62: 67–89. https://doi.org/10.1016/0304-4076(94)90017-5.

Baum, C. F. 2001. Residual diagnostics for cross-section time series regression models. *Stata Journal* 1: 101–104.

Bell, R. M., and D. F. McCaffrey. 2002. Bias reduction in standard errors for linear regression with multi-stage samples. *Survey Methodology* 28: 169–181.

Blackwell, J. L., III. 2005. Estimation and testing of fixed-effect panel-data systems. *Stata Journal* 5: 202–207.

Bottai, M., and N. Orsini. 2004. Confidence intervals for the variance component of random-effects linear models. *Stata Journal* 4: 429–435.

Bruno, G. S. F. 2005. Estimation and inference in dynamic unbalanced panel-data models with a small number of individuals. *Stata Journal* 5: 473–500.

Cabanillas, O. B., J. D. Michler, A. Michuda, and E. Tjernström. 2018. Fitting and interpreting correlated random-coefficient models using Stata. *Stata Journal* 18: 159–173.

Christodoulou, D., and V. Sarafidis. 2017. Regression clustering for panel-data models with fixed effects. *Stata Journal* 17: 314–329.

Correia, S. 2016. A feasible estimator for linear models with multi-way fixed effects. Unpublished manuscript, Duke University, https://scorreia.com/research/hdfe.pdf.

Correia, S., P. Guimarães, and T. Zylkin. 2020. Fast Poisson estimation with high-dimensional fixed effects. *Stata Journal* 20: 95–115.

De Hoyos, R. E., and V. Sarafidis. 2006. Testing for cross-sectional dependence in panel-data models. *Stata Journal* 6: 482–496.

Dwyer, J. H., and M. Feinleib. 1992. "Introduction to statistical models for longitudinal observation". In *Statistical Models for Longitudinal Studies of Health*, edited by J. H. Dwyer, M. Feinleib, P. Lippert, and H. Hoffmeister, 3–48. New York: Oxford University Press.

Gardner, J. 2023. Stata tip 149: Weighted estimation of fixed-effects and first-differences models. *Stata Journal* 23: 276–280.

Guimarães, P., and P. Portugal. 2010. A simple feasible procedure to fit models with high-dimensional fixed effects. *Stata Journal* 10: 628–649.

Hansen, B. E. 2022. *Econometrics*. Princeton, NJ: Princeton University Press.

Hernández-Ramos, L. M., R. Escalante, and M. Raydan. 2011. Unconstrained optimization techniques for the acceleration of alternating projection methods. *Numerical Functional Analysis and Optimization* 32: 1041–1066. https://doi.org/10.1080/01630563.2011.591954.

Hoechle, D. 2007. Robust standard errors for panel regressions with cross-sectional dependence. *Stata Journal* 7: 281–312.

Hughes, R. A., M. G. Kenward, J. A. C. Sterne, and K. Tilling. 2017. Analyzing repeated measurements while accounting for derivative tracking, varying within-subject variance, and autocorrelation: The xtmixediou command. *Stata Journal* 17: 573–599.

Judge, G. G., W. E. Griffiths, R. C. Hill, H. Lütkepohl, and T.-C. Lee. 1985. *The Theory and Practice of Econometrics*. 2nd ed. New York: Wiley.

Lee, L.-F., and W. E. Griffiths. 1979. The prior likelihood and best linear unbiased prediction in stochastic coefficient linear models. Working Paper 1, Department of Econometrics, University of New England, Armidale, Australia.

Libois, F., and V. Verardi. 2013. Semiparametric fixed-effects estimator. *Stata Journal* 13: 329–336.

Magazzini, L., R. L. Bruno, and M. Stampini. 2020. Using information from singletons in fixed-effects estimation: xtfesing. *Stata Journal* 20: 965–975.

McCaffrey, D. F., K. Mihaly, J. R. Lockwood, and T. R. Sass. 2012. A review of Stata commands for fixed-effects estimation in normal linear models. *Stata Journal* 12: 406–432.

Mundlak, Y. 1978. On the pooling of time series and cross section data. *Econometrica* 46: 69–85. https://doi.org/10.2307/1913646.

Nichols, A. 2007. Causal inference with observational data. *Stata Journal* 7: 507–541.

Pinzon, E. 2015. Fixed effects or random effects: The Mundlak approach. *The Stata Blog: Not Elsewhere Classified*. https://blog.stata.com/2015/10/29/fixed-effects-or-random-effects-the-mundlak-approach/.

Rios-Avila, F. 2015. Feasible fitting of linear models with N fixed effects. *Stata Journal* 15: 881–898.

Schunck, R. 2013. Within and between estimates in random-effects models: Advantages and drawbacks of correlated random effects and hybrid models. *Stata Journal* 13: 65–76.

Stammann, A. 2018. Fast and feasible estimation of generalized linear models with high-dimensional k-way fixed effects. arXiv:1707.01815 [stat.AP], https://doi.org/10.48550/arXiv.1707.01815.

Stock, J. H., and M. W. Watson. 2008. Heteroskedasticity-robust standard errors for fixed effects panel data regression. *Econometrica* 76: 155–174. https://doi.org/10.1111/j.0012-9682.2008.00821.x.

Swamy, P. A. V. B., and S. S. Arora. 1972. The exact finite sample properties of the estimators of coefficients in the error components regression models. *Econometrica* 40: 261–275. https://doi.org/10.2307/1909405.

Taub, A. J. 1979. Prediction in the context of the variance-components model. *Journal of Econometrics* 10: 103–107. https://doi.org/10.1016/0304-4076(79)90068-X.

Twisk, J. W. R. 2013. *Applied Longitudinal Data Analysis for Epidemiology: A Practical Guide*. 2nd ed. Cambridge: Cambridge University Press.

Wooldridge, J. M. 2019. Correlated random effects models with unbalanced panels. *Journal of Econometrics* 211: 137–150. https://doi.org/10.1016/j.jeconom.2018.12.010.

———. 2020. *Introductory Econometrics: A Modern Approach*. 7th ed. Boston: Cengage.

Wursten, J. 2018. Testing for serial correlation in fixed-effects panel models. *Stata Journal* 18: 76–100.

Also see

[XT] **xtreg postestimation** — Postestimation tools for xtreg

[XT] **xteregress** — Extended random-effects linear regression

[XT] **xtgee** — GEE population-averaged panel-data models

[XT] **xtgls** — GLS linear model with heteroskedastic and correlated errors

[XT] **xtheckman** — Random-effects regression with sample selection

[XT] **xtivreg** — Instrumental variables and two-stage least squares for panel-data models

[XT] **xtregar** — Fixed- and random-effects linear models with an AR(1) disturbance

[XT] **xtset** — Declare data to be panel data

[BAYES] **bayes: xtreg** — Bayesian random-effects linear model

[ME] **mixed** — Multilevel mixed-effects linear regression

[MI] **Estimation** — Estimation commands for use with mi estimate

[R] **areg** — Linear regression with many indicator variables

[R] **regress** — Linear regression

[R] **wildbootstrap** — Wild cluster bootstrap inference

[SP] **spxtregress** — Spatial autoregressive models for panel data

[TS] **forecast** — Econometric model forecasting

[TS] **prais** — Prais–Winsten and Cochrane–Orcutt regression

[U] 20 Estimation and postestimation commands

Postestimation commands

The following postestimation commands are of special interest after `xtreg`:

Command	Description
`xttest0`	Breusch and Pagan LM test for random effects
`estat mundlak`	Mundlak specification test

The following standard postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of parameters
* `estat ic`	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC, respectively)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
† `forecast`	dynamic forecasts and simulations
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
* `lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	linear predictions, residuals, error components
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of parameters
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

* `estat ic` and `lrtest` are not appropriate after `xtreg` with the `pa`, `re`, or `cre` option.

† `forecast` is not appropriate with `mi` estimation results.

predict

Description for predict

predict creates a new variable containing predictions such as fitted values, standard errors, predicted values, linear predictions, and the equation-level score.

Menu for predict

Statistics > Postestimation

Syntax for predict

Fixed-effects (FE) model

predict [*type*] *newvar* [*if*] [*in*] [, *FE_statistic*]

Between-effects and GLS and ML random-effects (RE) model

predict [*type*] *newvar* [*if*] [*in*] [, *RE/CRE_statistic*]

Correlated random-effects (CRE) model

predict [*type*] *newvar* [*if*] [*in*] [, *RE/CRE_statistic*]

Population-averaged (PA) model

predict [*type*] *newvar* [*if*] [*in*] [, *PA_statistic* nooffset]

FE_statistic	Description
Main	
`xb`	$\alpha + \mathbf{x}_{it}\boldsymbol{\beta}$, fitted values; the default
`stdp`	standard error of the fitted values
`ue`	$u_i + e_{it}$, the combined residual
* `xbu`	$\alpha + \mathbf{x}_{it}\boldsymbol{\beta} + u_i$, prediction including panel effect
* `u`	u_i, the fixed error component
* `xbd`	$\alpha + \mathbf{x}_{it}\boldsymbol{\beta} + d_{\text{absorbvars}}$, prediction including effects of absorbed variables
* `d`	$d_{\text{absorbvars}}$, effects of absorbed variables
* `e`	e_{it}, the overall error component

Unstarred statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample. Starred statistics are calculated only for the estimation sample, even when `if e(sample)` is not specified.

RE/CRE_statistic	Description
Main	
`xb`	$\alpha + \mathbf{x}_{it}\boldsymbol{\beta}$ or $\alpha + \mathbf{x}_{it}\boldsymbol{\beta} + \overline{\mathbf{x}}_i\gamma$ for CRE, fitted values; the default
`stdp`	standard error of the fitted values
`ue`	$u_i + e_{it}$, the combined residual
* `xbu`	$\alpha + \mathbf{x}_{it}\boldsymbol{\beta} + u_i$ or $\alpha + \mathbf{x}_{it}\boldsymbol{\beta} + \overline{\mathbf{x}}_i\gamma + u_i$ for CRE, prediction including panel effect
* `u`	u_i, the random-error component
* `e`	e_{it}, the overall error component

Unstarred statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample. Starred statistics are calculated only for the estimation sample, even when `if e(sample)` is not specified.

PA_statistic	Description
Main	
`mu`	predicted value of *depvar*; considers the `offset()`
`rate`	predicted value of *depvar*
`xb`	linear prediction
`stdp`	standard error of the linear prediction
`score`	first derivative of the log likelihood with respect to $\mathbf{x}_{it}\boldsymbol{\beta}$

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

Options for predict

Main

`xb` calculates the linear prediction. This is the default for all except the population-averaged model. The linear prediction equals $\alpha + \mathbf{x}_{it}\boldsymbol{\beta}$ for the fixed-, between-, and random-effects models; and equals $\alpha+\mathbf{x}_{it}\boldsymbol{\beta}+\overline{\mathbf{x}}_i\gamma$ for the correlated random-effects model. Panel means are recalculated when predicting out of sample in the correlated random-effects model.

`stdp` calculates the standard error of the linear prediction. For the fixed-effects model, this excludes the variance due to uncertainty about the estimate of u_i.

mu and rate both calculate the predicted value of *depvar*. mu takes into account the offset(), and rate ignores those adjustments. mu and rate are equivalent if you did not specify offset(). mu is the default for the population-averaged model.

ue calculates the prediction of $u_i + e_{it}$.

xbu calculates the linear prediction including the fixed or random component. This prediction equals $\alpha + \mathbf{x}_{it}\boldsymbol{\beta} + u_i$ for the fixed-, between-, and random-effects models; and equals $\alpha + \mathbf{x}_{it}\boldsymbol{\beta} + \overline{\mathbf{x}}_i\boldsymbol{\gamma} + u_i$ for the correlated random-effects model.

u calculates the prediction of u_i, the estimated fixed or random effect.

xbd calculates the prediction of $\alpha + \mathbf{x}_{it}\boldsymbol{\beta} + d_{\text{absorbvars}}$, the prediction including the absorbed variables' effects.

d calculates the prediction of $d_{\text{absorbvars}}$, the absorbed variables' effects.

e calculates the prediction of e_{it}.

score calculates the equation-level score, $u_{it} = \partial \ln L(\mathbf{x}_{it}\boldsymbol{\beta})/\partial(\mathbf{x}_{it}\boldsymbol{\beta})$.

nooffset is relevant only if you specified offset(*varname*) for xtreg, pa. It modifies the calculations made by predict so that they ignore the offset variable; the linear prediction is treated as $\mathbf{x}_{it}\boldsymbol{\beta}$ rather than $\mathbf{x}_{it}\boldsymbol{\beta} + \text{offset}_{it}$.

margins

Description for margins

margins estimates margins of response for fitted values, probabilities, and linear predictions.

Menu for margins

Statistics > Postestimation

Syntax for margins

margins [*marginlist*] [, *options*]

margins [*marginlist*] , predict(*statistic* ...) [predict(*statistic* ...) ...] [*options*]

Fixed-effects (FE) model

FE_statistic	Description
xb	$\alpha + \mathbf{x}_{it}\boldsymbol{\beta}$, fitted values; the default
stdp	not allowed with margins
ue	not allowed with margins
xbu	not allowed with margins
u	not allowed with margins
xbd	not allowed with margins
d	not allowed with margins
e	not allowed with margins

Between-effects, GLS and ML random-effects (RE), and correlated random-effects (CRE) model

RE_statistic	Description
xb	$\alpha + \mathbf{x}_{it}\boldsymbol{\beta}$ or $\alpha + \mathbf{x}_{it}\boldsymbol{\beta} + \overline{\mathbf{x}}_i\gamma$ for CRE, fitted values; the default
stdp	not allowed with margins
ue	not allowed with margins
xbu	not allowed with margins
u	not allowed with margins
e	not allowed with margins

Population-averaged (PA) model

PA_statistic	Description
mu	probability of *depvar*; considers the offset()
rate	probability of *depvar*
xb	linear prediction
stdp	not allowed with margins
score	not allowed with margins

Statistics not allowed with margins are functions of stochastic quantities other than e(b).

For the full syntax, see [R] **margins**.

xttest0

Description for xttest0

xttest0, for use after xtreg, re, presents the Breusch and Pagan (1980) Lagrange multiplier test for random effects, a test that $\mathrm{Var}(\nu_i) = 0$.

Menu for xttest0

Statistics > Longitudinal/panel data > Linear models > Lagrange multiplier test for random effects

Syntax for xttest0

```
xttest0
```

collect is allowed; see **[U] 11.1.10 Prefix commands**.

estat mundlak

Description for estat mundlak

estat mundlak performs a Mundlak specification test, a test that ν_i is uncorrelated with the regressors in $\mathbf{x}_{it}$. estat mundlak is for use after xtreg with the re, cre, and fe options but not with the absorb() option or the vce(hc2) option.

Menu for estat

Statistics > Postestimation

Syntax for estat mundlak

estat mundlak [, *options*]

options	Description
* reps(#)	perform # bootstrap replications; default is reps(50)
* rseed(#)	set random-number seed to #

*reps() and rseed() allowed only after xtreg, vce(bootstrap).

collect is allowed; see **[U] 11.1.10 Prefix commands**.

Options for estat mundlak

reps(#) specifies the number of bootstrap replications to perform when computing the p-value for the Mundlak test. This option is allowed only after xtreg, vce(bootstrap).

rseed(#) sets the random-number seed. This option is allowed only after xtreg, vce(bootstrap).

Remarks and examples

▷ Example 1: Lagrange multiplier test for random effects

Continuing with our random-effects model from example 4 in xtreg, we can see that xttest0 will report a test of $\nu_i = 0$. In case we have any doubts, we could type

```
. use https://www.stata-press.com/data/r19/nlswork
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)
. xtreg ln_w grade age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp
> tenure c.tenure#c.tenure 2.race not_smsa south, re theta
  (output omitted)
. xttest0
Breusch and Pagan Lagrangian multiplier test for random effects
        ln_wage[idcode,t] = Xb + u[idcode] + e[idcode,t]
        Estimated results:
                         |       Var     SD = sqrt(Var)
                ---------+-----------------------------
                 ln_wage |   .2283326       .4778416
                       e |   .0845002       .2906892
                       u |   .0665151       .2579053
        Test: Var(u) = 0
                             chibar2(01) = 14779.98
                          Prob > chibar2 =   0.0000
```

◁

▷ Example 2: Hausman specification test

More importantly, after xtreg, re estimation, hausman will perform the Hausman specification test. If our model is correctly specified, and if ν_i is uncorrelated with $\mathbf{x}_{it}$, the (subset of) coefficients that are estimated by the fixed-effects estimator and the same coefficients that are estimated here should not statistically differ:

```
. xtreg ln_w grade age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp
> tenure c.tenure#c.tenure 2.race not_smsa south, re
  (output omitted)
. estimates store random_effects
. xtreg ln_w grade age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp
> tenure c.tenure#c.tenure 2.race not_smsa south, fe
  (output omitted)
```

```
. hausman . random_effects

                 ---- Coefficients ----
             |      (b)          (B)            (b-B)     sqrt(diag(V_b-V_B))
             |       .       random_eff~s    Difference       Std. err.
-------------+----------------------------------------------------------------
         age |    .0359987     .0368059       -.0008073        .0013177
 c.age#c.age |    -.000723    -.0007133       -9.68e-06        .0000184
     ttl_exp |    .0334668     .0290208        .0044459         .001711
  c.ttl_exp# |
   c.ttl_exp |    .0002163     .0003049       -.0000886         .000053
      tenure |    .0357539     .0392519        -.003498        .0005797
   c.tenure# |
    c.tenure |   -.0019701    -.0020035        .0000334        .0000373
    not_smsa |   -.0890108    -.1308252        .0418144        .0062745
       south |   -.0606309    -.0868922        .0262613        .0081345
------------------------------------------------------------------------------
                          b = Consistent under H0 and Ha; obtained from xtreg.
           B = Inconsistent under Ha, efficient under H0; obtained from xtreg.

Test of H0: Difference in coefficients not systematic

    chi2(8) = (b-B)'[(V_b-V_B)^(-1)](b-B)
            = 149.43
Prob > chi2 = 0.0000
```

We can reject the hypothesis that the coefficients are the same. Before turning to what this means, note that hausman listed the coefficients estimated by the two models. It did not, however, list grade and 2.race. hausman did not make a mistake; in the Hausman test, we compare only the coefficients estimated by both techniques.

This means that we have an unpleasant choice: we can admit that our model is misspecified—that we have not parameterized it correctly—or we can hold that our specification is correct, in which case the observed differences must be due to the zero correlation of ν_i and the $\mathbf{x}_{it}$ assumption.

◁

❑ Technical note

We can also mechanically explore the underpinnings of the test's dissatisfaction. In the comparison table from hausman, it is the coefficients on not_smsa and south that exhibit the largest differences. In equation (1′) of [XT] **xtreg**, we showed how to decompose a model into within and between effects. Let's do that with these two variables, assuming that changes in the average have one effect, whereas transitional changes have another:

```
. egen avgnsmsa = mean(not_smsa), by(id)

. generate devnsma = not_smsa -avgnsmsa
(8 missing values generated)

. egen avgsouth = mean(south), by(id)

. generate devsouth = south - avgsouth
(8 missing values generated)

. xtreg ln_w grade age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp tenure c.tenure#
> c.tenure 2.race avgnsm devnsm avgsou devsou

Random-effects GLS regression                   Number of obs     =     28,091
Group variable: idcode                          Number of groups  =      4,697

R-squared:                                      Obs per group:
     Within  = 0.1723                                         min =          1
     Between = 0.4809                                         avg =        6.0
     Overall = 0.3737                                         max =         15

                                                Wald chi2(12)     =    9319.56
corr(u_i, X) = 0 (assumed)                      Prob > chi2       =     0.0000

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
       grade |   .0631716   .0017903    35.29   0.000     .0596627    .0666805
         age |   .0375196   .0031186    12.03   0.000     .0314072     .043632
             |
 c.age#c.age |  -.0007248     .00005   -14.50   0.000    -.0008228   -.0006269
             |
     ttl_exp |   .0286543   .0024207    11.84   0.000     .0239098    .0333989
             |
   c.ttl_exp#|
   c.ttl_exp |   .0003222   .0001162     2.77   0.006     .0000945    .0005499
             |
      tenure |   .0394423    .001754    22.49   0.000     .0360044    .0428801
             |
    c.tenure#|
    c.tenure |  -.0020081   .0001192   -16.85   0.000    -.0022417   -.0017746
             |
        race |
       Black |  -.0545936   .0102101    -5.35   0.000     -.074605   -.0345821
    avgnsmsa |  -.1833237   .0109339   -16.77   0.000    -.2047537   -.1618937
     devnsma |  -.0887596   .0095071    -9.34   0.000    -.1073931    -.070126
    avgsouth |  -.1011235   .0098789   -10.24   0.000    -.1204858   -.0817611
    devsouth |  -.0598538   .0109054    -5.49   0.000     -.081228   -.0384797
       _cons |   .2682987   .0495778     5.41   0.000      .171128    .3654694
-------------+----------------------------------------------------------------
     sigma_u |   .2579182
     sigma_e |  .29068923
         rho |  .44047745   (fraction of variance due to u_i)
------------------------------------------------------------------------------
```

We will leave the reinterpretation of this model to you, except that if we were really going to sell this model, we would have to explain why the between and within effects are different. Focusing on residence in a non-SMSA, we might tell a story about rural people being paid less and continuing to get paid less when they move to the SMSA. Given our panel data, we could create variables to measure this (an indicator for moved from non-SMSA to SMSA) and to measure the effects. In our assessment of this model, we should think about women in the cities moving to the country and their relative productivity in a bucolic setting.

In any case, the Hausman test now is

```
. estimates store new_random_effects

. xtreg ln_w grade age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp
> tenure c.tenure#c.tenure 2.race avgnsm devnsm avgsou devsou, fe
  (output omitted)

. hausman . new_random_effects

                 ---- Coefficients ----
             |      (b)          (B)            (b-B)     sqrt(diag(V_b-V_B))
             |       .       new_random~s    Difference       Std. err.
-------------+----------------------------------------------------------------
         age |    .0359987     .0375196       -.0015209        .0013198
 c.age#c.age |    -.000723    -.0007248        1.84e-06        .0000184
     ttl_exp |    .0334668     .0286543        .0048124        .0017127
   c.ttl_exp#|
   c.ttl_exp |    .0002163     .0003222       -.0001059        .0000531
      tenure |    .0357539     .0394423       -.0036884        .0005839
    c.tenure#|
    c.tenure |   -.0019701    -.0020081         .000038        .0000377
     devnsma |   -.0890108    -.0887596       -.0002512         .000683
    devsouth |   -.0606309    -.0598538       -.0007771        .0007618
------------------------------------------------------------------------------
                          b = Consistent under H0 and Ha; obtained from xtreg.
           B = Inconsistent under Ha, efficient under H0; obtained from xtreg.

Test of H0: Difference in coefficients not systematic

    chi2(8) = (b-B)'[(V_b-V_B)^(-1)](b-B)
            =  92.52
Prob > chi2 = 0.0000
```

We have mechanically succeeded in greatly reducing the χ^2, but not by enough. The major differences now are in the age, experience, and tenure effects. We already knew this problem existed because of the ever-increasing effect of experience. More careful parameterization work rather than simply including squares needs to be done.

❑

▷ Example 3: Mundlak specification test

Suppose now that the errors in our random-effects model are serially correlated within `idcode`. For most panel-data structures, this is probably a more realistic assumption than assuming i.i.d. errors. We can no longer rely on the Hausman specification test when the errors are correlated. However, we can still conduct a cluster–robust specification test by using `estat mundlak`.

We first refit the random-effects model with cluster–robust standard errors and then use `estat mundlak` to perform the specification test.

```
. xtreg ln_w grade age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp tenure
> c.tenure#c.tenure 2.race not_smsa south, re vce(cluster idcode)
  (output omitted)
. estat mundlak
Mundlak specification test
H0: Covariates are uncorrelated with unobserved panel-level effects
    chi2(8) = 123.10
Prob > chi2 = 0.0000
Notes: Fixed effects and correlated random effects are
       consistent under H0 and Ha.
       Random effects are efficient under H0.
```

With a Wald test statistic of 123.10 and a p-value of 0.0000, we reject the null hypothesis that ν_i is uncorrelated with the regressors in $\mathbf{x}_{it}$. This result implies that the random-effects model as stated is misspecified and that either the model needs to be reparameterized or estimation by correlated random effects or fixed effects should be considered.

◁

Stored results

`xttest0` stores the following in `r()`:

Scalars

`r(lm)`	Lagrange multiplier statistic
`r(df)`	degrees of freedom
`r(p)`	p-value

`estat mundlak` stores the following in `r()`:

Scalars

`r(chi2_mundlak)`	Mundlak test χ^2 statistic
`r(df_mundlak)`	Mundlak test degrees of freedom
`r(p_mundlak)`	Mundlak test p-value
`r(N_reps)`	number of complete replications
`r(N_clust)`	number of clusters

Macros

`r(rngstate)`	random-number state
`r(clustvar)`	name of cluster variable

Methods and formulas

Methods and formulas are presented under the following headings:

Predictions for fixed-effects model with absorbed variables
xttest0
estat mundlak

Predictions for fixed-effects model with absorbed variables

The following uses the notation introduced in *Methods and formulas* in [XT] **xtreg**.

Suppose we fit the model

$$y_{it} = \alpha + \mathbf{x}_{it}\boldsymbol{\beta} + \eta_{it} + \epsilon_{it}$$

where $\eta_{it} = \mathbf{d}_{1(it)}\gamma_1 + \mathbf{d}_{2(it)}\gamma_2 + \cdots + \mathbf{d}_{h(it)}\gamma_h$ and $\mathbf{d}_{k(it)}$ is an indicator vector for absorbed variable k in panel i at time t. By convention, we define the first absorb variable ($k = 1$) as the panel-ID variable and write $\eta_{it} = \nu_i + \sum_{k=2}^{h} \mathbf{d}_{k(it)}\gamma_k = \nu_i + \Delta_{it}$. Let $\boldsymbol{\eta}$ be the $1 \times N$ vector with the values of η_{it} in the sample. Define vectors Δ and $\boldsymbol{\nu}$ similarly.

We are interested in making predictions for the panel effects $\boldsymbol{\nu}$ (the u option), the absorbed variables' effects $\boldsymbol{\Delta}$ (the d option), and the residuals $\boldsymbol{\epsilon}$ (the e option). These postestimation predictions use alternating projection methods to avoid the computational burden of estimating the $\boldsymbol{\gamma}$'s. Two alternating projection method algorithms are available: Halperin and Cimmino. For a description of these two algorithms, see *Methods and formulas* in [XT] **xtreg**.

Estimates for $\boldsymbol{\nu}$, $\boldsymbol{\Delta}$, and the residuals can be extracted from $\hat{\boldsymbol{v}} = \mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}$, where $\widehat{\boldsymbol{\beta}}$ is the estimated regression coefficient vector. Here is the procedure. First, separate the absorbed variables into two sets: $\iota_1 = \{1\}$, the panel effect, and $\iota_2 = \{2, \ldots, h\}$, the remaining absorbed variables. Let project$(\hat{\boldsymbol{v}},\ \iota_i)$ denote the projection of vector $\hat{\boldsymbol{v}}$ over the variables in set ι_i, using either the Halperin or Cimmino algorithm, and let p_error$(\hat{\boldsymbol{v}},\ \iota_i)$ denote the corresponding projection error. Second, initialize the values of the algorithm to $\tilde{\boldsymbol{\epsilon}}^{(1)} = \hat{\boldsymbol{v}}$, $\tilde{\boldsymbol{\nu}}^{(1)} = \mathbf{0}$, and $\widetilde{\boldsymbol{\Delta}}^{(1)} = \mathbf{0}$. Third and finally, iterate until convergence using the following formulas:

$$\tilde{\boldsymbol{\nu}}^{(j+1)} = \tilde{\boldsymbol{\nu}}^{(j)} + \text{project}(\tilde{\boldsymbol{\epsilon}}^{(j)},\ \iota_1)$$

$$\tilde{\boldsymbol{\epsilon}}^{(j+1)} = \text{p\_error}\{\text{p\_error}(\tilde{\boldsymbol{\epsilon}}^{(j)},\ \iota_1),\ \iota_2\}$$

$$\widetilde{\boldsymbol{\Delta}}^{(j+1)} = \widetilde{\boldsymbol{\Delta}}^{(j)} + \text{project}\{\text{p\_error}(\tilde{\boldsymbol{\epsilon}}^{(j)},\ \iota_1),\ \iota_2\}$$

Convergence is declared at $j = \jmath$ when the elements of project$(\tilde{\boldsymbol{\epsilon}}^{(\jmath)},\ \iota_1)$ and project$(\tilde{\boldsymbol{\epsilon}}^{(\jmath)},\ \iota_2)$ are negligible. At convergence, $\tilde{\boldsymbol{\epsilon}}^{(\jmath)}$ contains the estimated residuals, $\tilde{\boldsymbol{\nu}}^{(\jmath)}$ the estimated panel effects, and $\widetilde{\boldsymbol{\Delta}}^{(\jmath)}$ the estimated absorbed variables' effects.

xttest0

`xttest0` reports the Lagrange multiplier test for random effects developed by Breusch and Pagan (1980) and as modified by Baltagi and Li (1990). The model

$$y_{it} = \alpha + \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i$$

is fit via OLS, and then the quantity

$$\lambda_{\text{LM}} = \frac{(n\overline{T})^2}{2}\left\{\frac{A_1^2}{(\sum_i T_i^2) - n\overline{T}}\right\}$$

is calculated, where

$$A_1 = 1 - \frac{\sum_{i=1}^{n}(\sum_{t=1}^{T_i} v_{it})^2}{\sum_i \sum_t v_{it}^2}$$

The Baltagi and Li modification allows for unbalanced data and reduces to the standard formula

$$\lambda_{\mathrm{LM}} = \begin{cases} \frac{nT}{2(T-1)}\left\{\frac{\sum_i(\sum_t v_{it})^2}{\sum_i\sum_t v_{it}^2} - 1\right\}^2, & \hat{\sigma}_u^2 \geq 0 \\ 0 & , \quad \hat{\sigma}_u^2 < 0 \end{cases}$$

when $T_i = T$ (balanced data). Under the null hypothesis, λ_{LM} is distributed as a 50:50 mixture of a point mass at zero and $\chi^2(1)$.

estat mundlak

`estat mundlak` performs a Mundlak specification test to help decide between estimation by random effects or estimation by correlated random effects or fixed effects. The test consists of first fitting a correlated random-effects model, using the same covariates and the same sample as the previously fitted `xtreg` model, and then testing whether the coefficients of the panel means are all jointly equal to 0. See Mundlak (1978), Wooldridge (2019), and *xtreg, cre* in [XT] **xtreg** for more details on correlated random-effects models and the Mundlak specification test. `estat mundlak` is available after `xtreg, re`; `xtreg, fe`; and `xtreg, cre`.

`estat mundlak` uses the same VCE specified in the estimation step to fit the underlying correlated random-effects model. So, for example, if the `vce(cluster` *clustvar*`)` option was specified in the estimation step, `estat mundlak` will fit the correlated random-effects model using the same level of clustering for the standard errors. If the `vce(bootstrap)` option was specified in the estimation step, `estat mundlak` computes a bootstrapped p-value for the Mundlak test following the procedure for Wald-type bootstrap tests in Hansen (2022, 294). You can set the number of bootstrap replications used in this computation with the `reps()` option.

References

Alejo, J., A. F. Galvao, G. Montes-Rojas, and W. Sosa-Escudero. 2015. Tests for normality in linear panel-data models. *Stata Journal* 15: 822–832.

Baltagi, B. H., and Q. Li. 1990. A Lagrange multiplier test for the error components model with incomplete panels. *Econometric Reviews* 9: 103–107. https://doi.org/10.1080/07474939008800180.

Breusch, T. S., and A. R. Pagan. 1980. The Lagrange multiplier test and its applications to model specification in econometrics. *Review of Economic Studies* 47: 239–253. https://doi.org/10.2307/2297111.

Hansen, B. E. 2022. *Econometrics*. Princeton, NJ: Princeton University Press.

Hausman, J. A. 1978. Specification tests in econometrics. *Econometrica* 46: 1251–1271. https://doi.org/10.2307/1913827.

Mundlak, Y. 1978. On the pooling of time series and cross section data. *Econometrica* 46: 69–85. https://doi.org/10.2307/1913646.

Sosa-Escudero, W., and A. K. Bera. 2008. Tests for unbalanced error-components models under local misspecification. *Stata Journal* 8: 68–78.

Verbeke, G., and G. Molenberghs. 2003. The use of score tests for inference on variance components. *Biometrics* 59: 254–262. https://doi.org/10.1111/1541-0420.00032.

Wooldridge, J. M. 2019. Correlated random effects models with unbalanced panels. *Journal of Econometrics* 211: 137–150. https://doi.org/10.1016/j.jeconom.2018.12.010.

Also see

[XT] **xtreg** — Linear models for panel data

[U] 20 Estimation and postestimation commands

xtregar — Fixed- and random-effects linear models with an AR(1) disturbance

Description

`xtregar` fits cross-sectional time-series regression models when the disturbance term is first-order autoregressive. `xtregar` offers a within estimator for fixed-effects models and a GLS estimator for random-effects models. `xtregar` can accommodate unbalanced panels whose observations are unequally spaced over time.

Quick start

Random-effects model of `y` on `x1` with an AR(1) disturbance using `xtset` data

```
xtregar y x1
```

Add `x2` and `x3` as covariates and perform Baltagi–Wu LBI test

```
xtregar y x1 x2 x3, lbi
```

Fixed-effects model using the within estimator and observations where `tvar` is greater than 2,000

```
xtregar y x1 x2 x3 if tvar > 2000, fe
```

Menu

Statistics > Longitudinal/panel data > Linear models > Linear regression with AR(1) disturbance (FE, RE)

Syntax

GLS random-effects (RE) model

`xtregar` *depvar* [*indepvars*] [*if*] [*in*] [, `re` *options*]

Fixed-effects (FE) model

`xtregar` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] , `fe` [*options*]

options	Description
Model	
`re`	use random-effects estimator; the default
`fe`	use fixed-effects estimator
`rhotype(`*rhomethod*`)`	specify method to compute autocorrelation; seldom used
`rhof(#)`	use # for ρ and do not estimate ρ
`twostep`	perform two-step estimate of correlation
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`lbi`	perform Baltagi–Wu LBI test
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
`coeflegend`	display legend instead of statistics

A panel variable and a time variable must be specified; use `xtset`; see [XT] **xtset**.

indepvars may contain factor variables; see **[U] 11.4.3 Factor variables**.

depvar and *indepvars* may contain time-series operators; see **[U] 11.4.4 Time-series varlists**.

`by`, `collect`, and `statsby` are allowed; see **[U] 11.1.10 Prefix commands**.

`fweight`s and `aweight`s are allowed for the fixed-effects model with `rhotype(regress)` or `rhotype(freg)`, or with a fixed rho; see **[U] 11.1.6 weight**. Weights must be constant within panel.

`coeflegend` does not appear in the dialog box.

See **[U] 20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

re requests the GLS estimator of the random-effects model, which is the default.

fe requests the within estimator of the fixed-effects model.

rhotype(*rhomethod*) allows the user to specify any of the following estimators of ρ:

dw	$\rho_{\text{dw}} = 1 - d/2$, where d is the Durbin–Watson d statistic
regress	$\rho_{\text{reg}} = \beta$ from the residual regression $\epsilon_t = \beta\epsilon_{t-1}$
freg	$\rho_{\text{freg}} = \beta$ from the residual regression $\epsilon_t = \beta\epsilon_{t+1}$
tscorr	$\rho_{\text{tscorr}} = \boldsymbol{\epsilon}'\boldsymbol{\epsilon}_{t-1}/\boldsymbol{\epsilon}'\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is the vector of residuals and $\boldsymbol{\epsilon}_{t-1}$ is the vector of lagged residuals
theil	$\rho_{\text{theil}} = \rho_{\text{tscorr}}(N-k)/N$
nagar	$\rho_{\text{nagar}} = (\rho_{\text{dw}}N^2 + k^2)/(N^2 - k^2)$
onestep	$\rho_{\text{onestep}} = (n/m_c)(\boldsymbol{\epsilon}'\boldsymbol{\epsilon}_{t-1}/\boldsymbol{\epsilon}'\boldsymbol{\epsilon})$, where $\boldsymbol{\epsilon}$ is the vector of residuals, n is the number of observations, and m_c is the number of consecutive pairs of residuals

dw is the default method. Except for onestep, the details of these methods are given in [TS] **prais**. prais handles unequally spaced data. onestep is the one-step method proposed by Baltagi and Wu (1999). More details on this method are available below in *Methods and formulas*.

rhof(#) specifies that the given number be used for ρ and that ρ not be estimated.

twostep requests that a two-step implementation of the *rhomethod* estimator of ρ be used. Unless a fixed value of ρ is specified (with the rhof() option), ρ is estimated by running prais on the de-meaned data. When twostep is specified, prais will stop on the first iteration after the equation is transformed by ρ—the two-step efficient estimator. Although it is customary to iterate these estimators to convergence, they are efficient at each step. When twostep is not specified, the FGLS process iterates to convergence as described in the *Methods and formulas* of [TS] **prais**.

Reporting

level(#); see [R] **Estimation options**.

lbi requests that the Baltagi–Wu (1999) locally best invariant (LBI) test statistic that $\rho = 0$ and a modified version of the Bhargava, Franzini, and Narendranathan (1982) Durbin–Watson statistic be calculated and reported. The default is not to report them. p-values are not reported for either statistic. Although Bhargava, Franzini, and Narendranathan (1982) published critical values for their statistic, no tables are currently available for the Baltagi–Wu LBI. Baltagi and Wu (1999) derive a normalized version of their statistic, but this statistic cannot be computed for datasets of moderate size. You can also specify these options upon replay.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

The following option is available with xtregar but is not shown in the dialog box:

coeflegend; see [R] **Estimation options**.

Remarks and examples

Remarks are presented under the following headings:

Introduction

If you have not read [XT] **xt**, please do so.

`xtregar` fits cross-sectional time-series regression models when the disturbance term is first-order autoregressive. The models of interest are described by

$$y_{it} = \alpha + \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i + \epsilon_{it} \qquad i = 1, \dots, N; \quad t = 1, \dots, T_i \tag{1}$$

where

$$\epsilon_{it} = \rho\epsilon_{i,t-1} + \eta_{it} \tag{2}$$

and where $|\rho| < 1$ and η_{it} is independent and identically distributed (i.i.d.) with mean 0 and variance σ^2_η.

In the fixed-effects model, the ν_i are assumed to be correlated with the covariates $\mathbf{x}_{it}$, whereas in the random-effects model they are assumed to follow an i.i.d. process with mean 0 and variance σ^2_η and to be uncorrelated with the $\mathbf{x}_{it}$.

Similar to other linear panel-data models, any $\mathbf{x}_{it}$ that do not vary over t are collinear with the ν_i and will be omitted from the fixed-effects model. In contrast, the random-effects model can accommodate covariates that are constant over time.

`xtregar` offers a within estimator for the fixed-effect model and the Baltagi–Wu (1999) GLS estimator of the random-effects model. Both of these estimators offer several estimators of ρ.

The Baltagi–Wu (1999) GLS estimator extends the balanced panel estimator in Baltagi and Li (1991) to a case of exogenously unbalanced panels with unequally spaced observations. Specifically, the dataset contains observations on individual i at times t_{ij} for $j = 1, \dots, n_i$. The difference $t_{ij} - t_{i,j-1}$ plays an integral role in the estimation techniques used by `xtregar`.

For this reason, you must specify the `delta()` option when you `xtset` *panelvar timevar* if, for example, you have quarterly data with a monthly *timevar* recorded every three months instead of a quarterly *timevar*; see [XT] **xtset**.

The fixed-effects model

Let's examine the fixed-effect model first. The basic approach is common to all fixed-effects models. The ν_i are treated as nuisance parameters. We use a transformation of the model that removes the nuisance parameters and leaves behind the parameters of interest in an estimable form. Subtracting the group means from (1) removes the ν_i from the model

$$y_{it_{ij}} - \overline{y}_i = \left(\overline{\mathbf{x}}_{it_{ij}} - \overline{\mathbf{x}}_i\right)\boldsymbol{\beta} + \epsilon_{it_{ij}} - \overline{\epsilon}_i \tag{3}$$

where

$$\overline{y}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} y_{it_{ij}} \qquad \overline{\mathbf{x}}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} \mathbf{x}_{it_{ij}} \qquad \overline{\epsilon}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} \epsilon_{it_{ij}}$$

After the transformation, (3) is a linear AR(1) model, potentially with unequally spaced observations. (3) can be used to estimate ρ. Given an estimate of ρ, we must do a Cochrane–Orcutt transformation on each panel and then remove the within-panel means and add back the overall mean for each variable. OLS on the transformed data will produce the within estimates of α and $\boldsymbol{\beta}$.

▷ Example 1: Fixed-effects model

Let's use the Grunfeld investment dataset to illustrate how `xtregar` can be used to fit the fixed-effects model. This dataset contains information on 10 firms' investment, market value, and the value of their capital stocks. The data were collected annually between 1935 and 1954. The following output shows that we have `xtset` our data and gives the results of running a fixed-effects model with investment as a function of market value and the capital stock.

```
. use https://www.stata-press.com/data/r19/grunfeld

. xtset

Panel variable: company (strongly balanced)
 Time variable: year, 1935 to 1954
         Delta: 1 year

. xtregar invest mvalue kstock, fe

FE (within) regression with AR(1) disturbances  Number of obs     =        190
Group variable: company                         Number of groups  =         10

R-squared:                                      Obs per group:
     Within  = 0.5927                                         min =         19
     Between = 0.7989                                         avg =       19.0
     Overall = 0.7904                                         max =         19

                                                F(2,178)          =     129.49
corr(u_i, Xb) = -0.0454                         Prob > F          =     0.0000

------------------------------------------------------------------------------
      invest | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
      mvalue |   .0949999   .0091377    10.40   0.000     .0769677     .113032
      kstock |    .350161   .0293747    11.92   0.000     .2921935    .4081286
       _cons |  -63.22022   5.648271   -11.19   0.000    -74.36641   -52.07402
-------------+----------------------------------------------------------------
      rho_ar |  .67210608
     sigma_u |  91.507609
     sigma_e |  40.992469
     rho_fov |   .8328647   (fraction of variance because of u_i)
------------------------------------------------------------------------------
F test that all u_i=0: F(9,178) = 11.53                      Prob > F = 0.0000
```

Because there are 10 groups, the panel-by-panel Cochrane–Orcutt method decreases the number of available observations from 200 to 190. The above example used the default `dw` estimator of ρ. Using the `tscorr` estimator of ρ yields

```
. xtregar invest mvalue kstock, fe rhotype(tscorr)

FE (within) regression with AR(1) disturbances  Number of obs     =        190
Group variable: company                          Number of groups  =         10

R-squared:                                       Obs per group:
     Within  = 0.6583                                         min =         19
     Between = 0.8024                                         avg =       19.0
     Overall = 0.7933                                         max =         19

                                                 F(2,178)          =     171.47
corr(u_i, Xb) = -0.0709                          Prob > F          =     0.0000

------------------------------------------------------------------------------
      invest | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
      mvalue |   .0978364   .0096786    10.11   0.000     .0787369    .1169359
      kstock |    .346097   .0242248    14.29   0.000     .2982922    .3939018
       _cons |  -61.84403   6.621354    -9.34   0.000    -74.91049   -48.77758
-------------+----------------------------------------------------------------
      rho_ar |  .54131231
     sigma_u |  90.893572
     sigma_e |  41.592151
     rho_fov |  .82686297   (fraction of variance because of u_i)
------------------------------------------------------------------------------
F test that all u_i=0: F(9,178) = 19.73                      Prob > F = 0.0000
```

◁

❑ Technical note

The tscorr estimator of ρ is bounded in $[-1,1]$. The other estimators of ρ are not. In samples with short panels, the estimates of ρ produced by the other estimators of ρ may be outside $[-1,1]$. If this happens, use the tscorr estimator. However, simulations have shown that the tscorr estimator is biased toward zero. dw is the default because it performs well in Monte Carlo simulations. In the example above, the estimate of ρ produced by tscorr is much smaller than the one produced by dw.

❑

▷ Example 2: Using xtset

xtregar will complain if you try to run xtregar on a dataset that has not been xtset:

```
. xtset, clear

. xtregar invest mvalue kstock, fe
must specify panelvar and timevar; use xtset
r(459);
```

You must xtset your data to ensure that xtregar understands the nature of your time variable. Suppose that our observations were taken quarterly instead of annually. We will get the same results with the quarterly variable t2 that we did with the annual variable year.

```
. generate t = year - 1934
. generate t2 = tq(1934q4) + t
. format t2 %tq
. list year t2 in 1/5

     +---------------+
     | year       t2 |
     |---------------|
  1. | 1935   1935q1 |
  2. | 1936   1935q2 |
  3. | 1937   1935q3 |
  4. | 1938   1935q4 |
  5. | 1939   1936q1 |
     +---------------+

. xtset company t2
Panel variable: company (strongly balanced)
 Time variable: t2, 1935q1 to 1939q4
         Delta: 1 quarter

. xtregar invest mvalue kstock, fe
FE (within) regression with AR(1) disturbances  Number of obs     =        190
Group variable: company                         Number of groups  =         10

R-squared:                                      Obs per group:
     Within  = 0.5927                                         min =         19
     Between = 0.7989                                         avg =       19.0
     Overall = 0.7904                                         max =         19

                                                F(2,178)          =     129.49
corr(u_i, Xb) = -0.0454                         Prob > F          =     0.0000

------------------------------------------------------------------------------
      invest | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
      mvalue |   .0949999   .0091377    10.40   0.000     .0769677     .113032
      kstock |    .350161   .0293747    11.92   0.000     .2921935    .4081286
       _cons |  -63.22022   5.648271   -11.19   0.000    -74.36641   -52.07402
-------------+----------------------------------------------------------------
      rho_ar |  .67210608
     sigma_u |  91.507609
     sigma_e |  40.992469
     rho_fov |   .8328647   (fraction of variance because of u_i)
------------------------------------------------------------------------------
F test that all u_i=0: F(9,178) = 11.53                      Prob > F = 0.0000
```

◁

In all the examples thus far, we have assumed that ϵ_{it} is first-order autoregressive. Testing the hypothesis of $\rho = 0$ in a first-order autoregressive process produces test statistics with extremely complicated distributions. Bhargava, Franzini, and Narendranathan (1982) extended the Durbin–Watson statistic to the case of balanced, equally spaced panel datasets. Baltagi and Wu (1999) modify their statistic to account for unbalanced panels with unequally spaced data. In the same article, Baltagi and Wu (1999) derive the locally best invariant test statistic of $\rho = 0$. Both these test statistics have extremely complicated distributions, although Bhargava, Franzini, and Narendranathan (1982) did publish some critical values in their article. Specifying the `lbi` option to `xtregar` causes Stata to calculate and report the modified Bhargava et al. Durbin–Watson and the Baltagi–Wu LBI.

▷ Example 3: Testing for autocorrelation

In this example, we calculate the modified Bhargava et al. Durbin–Watson statistic and the Baltagi–Wu LBI. We exclude periods 9 and 10 from the sample, thereby reproducing the results of Baltagi and Wu (1999, 822). p-values are not reported for either statistic. Although Bhargava, Franzini, and Narendranathan (1982) published critical values for their statistic, no tables are currently available for the Baltagi–Wu (LBI). Baltagi and Wu (1999) did derive a normalized version of their statistic, but this statistic cannot be computed for datasets of moderate size.

```
. xtregar invest mvalue kstock if year !=1943 & year !=1944, fe lbi

FE (within) regression with AR(1) disturbances  Number of obs     =        170
Group variable: company                         Number of groups  =         10

R-squared:                                      Obs per group:
     Within  = 0.5907                                         min =         17
     Between = 0.7938                                         avg =       17.0
     Overall = 0.7879                                         max =         17

                                                F(2,158)          =     114.00
corr(u_i, Xb) = -0.0339                         Prob > F          =     0.0000

------------------------------------------------------------------------------
      invest | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
      mvalue |   .0922066   .0090362    10.20   0.000     .0743593    .1100539
      kstock |   .3509339   .0320278    10.96   0.000      .287676    .4141919
       _cons |  -61.69045   6.192364    -9.96   0.000    -73.92094   -49.45996
-------------+----------------------------------------------------------------
      rho_ar |  .67483913
     sigma_u |  94.568243
     sigma_e |  42.600124
     rho_fov |  .83130847   (fraction of variance because of u_i)
------------------------------------------------------------------------------
F test that all u_i=0: F(9,158) = 10.66                      Prob > F = 0.0000
Modified Bhargava et al. Durbin-Watson = .70578896
Baltagi-Wu LBI = 1.0218978
```

◁

The random-effects model

In the random-effects model, the ν_i are assumed to be realizations of an i.i.d. process with mean 0 and variance σ^2_ν. Furthermore, the ν_i are assumed to be independent of both the ϵ_{it} and the covariates $\mathbf{x}_{it}$. The latter of these assumptions can be strong, but inference is not conditional on the particular realizations of the ν_i in the sample. See Mundlak (1978) for a discussion of this point.

▷ Example 4: Random-effects model

By specifying the `re` option, we obtain the Baltagi–Wu GLS estimator of the random-effects model. This estimator can accommodate unbalanced panels and unequally spaced data. We run this model on the Grunfeld dataset:

```
. xtregar invest mvalue kstock if year !=1943 & year !=1944, re lbi

RE GLS regression with AR(1) disturbances        Number of obs     =        180
Group variable: company                          Number of groups  =         10

R-squared:                                       Obs per group:
     Within  = 0.7718                                          min =         18
     Between = 0.8036                                          avg =       18.0
     Overall = 0.7956                                          max =         18

                                                 Wald chi2(3)      =     335.41
corr(u_i, Xb) = 0 (assumed)                      Prob > chi2       =     0.0000

------------------------------------------------------------------------------
      invest | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      mvalue |   .0948541   .0085443    11.10   0.000     .0781075    .1116007
      kstock |    .322599   .0271626    11.88   0.000     .2693613    .3758368
       _cons |  -44.82233   27.24889    -1.64   0.100    -98.22918    8.584515
-------------+----------------------------------------------------------------
      rho_ar |  .67483913   (estimated autocorrelation coefficient)
     sigma_u |  74.332091
     sigma_e |  43.199999
     rho_fov |  .74751539   (fraction of variance due to u_i)
       theta |  .65649837
------------------------------------------------------------------------------
Modified Bhargava et al. Durbin-Watson = .70578896
Baltagi-Wu LBI = 1.0218978
```

The modified Bhargava et al. Durbin–Watson and the Baltagi–Wu LBI are the same as those reported for the fixed-effects model because the formulas for these statistics do not depend on fitting the fixed-effects model or the random-effects model.

◁

Stored results

xtregar, re stores the following in e():

Scalars	
e(N)	number of observations
e(N_g)	number of groups
e(df_m)	model degrees of freedom
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(d1)	Bhargava et al. Durbin–Watson
e(LBI)	Baltagi–Wu LBI statistic
e(N_LBI)	number of obs used in e(LBI)
e(Tcon)	1 if T is constant
e(sigma_u)	panel-level standard deviation
e(sigma_e)	standard deviation of η_{it}
e(r2_w)	R^2 for within model
e(r2_o)	R^2 for overall model
e(r2_b)	R^2 for between model
e(chi2)	χ^2
e(rho_ar)	autocorrelation coefficient
e(rho_fov)	u_i fraction of variance
e(thta_min)	minimum θ
e(thta_5)	θ, 5th percentile
e(thta_50)	θ, 50th percentile
e(thta_95)	θ, 95th percentile
e(thta_max)	maximum θ
e(Tbar)	harmonic mean of group sizes
e(rank)	rank of e(V)
Macros	
e(cmd)	xtregar
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(ivar)	variable denoting groups
e(tvar)	variable denoting time within groups
e(model)	re
e(rhotype)	method of estimating ρ_{ar}
e(dw)	lbi, if lbi specified
e(chi2type)	Wald; type of model χ^2 test
e(properties)	b V
e(predict)	program used to implement predict
e(marginsok)	predictions allowed by margins
e(marginsnotok)	predictions disallowed by margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved
Matrices	
e(b)	coefficient vector
e(V)	VCE for random-effects model
Functions	
e(sample)	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

`xtregar, fe` stores the following in `e()`:

Scalars

`e(N)`	number of observations
`e(N_g)`	number of groups
`e(df_m)`	model degrees of freedom
`e(mss)`	model sum of squares
`e(rss)`	residual sum of squares
`e(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(g_max)`	largest group size
`e(d1)`	Bhargava et al. Durbin–Watson
`e(LBI)`	Baltagi–Wu LBI statistic
`e(N_LBI)`	number of obs used in `e(LBI)`
`e(Tcon)`	1 if T is constant
`e(corr)`	corr(u_i, `Xb`)
`e(sigma_u)`	panel-level standard deviation
`e(sigma_e)`	standard deviation of ϵ_{it}
`e(r2_a)`	adjusted R^2
`e(r2_w)`	R^2 for within model
`e(r2_o)`	R^2 for overall model
`e(r2_b)`	R^2 for between model
`e(ll)`	log likelihood
`e(ll_0)`	log likelihood, constant-only model
`e(rho_ar)`	autocorrelation coefficient
`e(rho_fov)`	u_i fraction of variance
`e(F)`	F statistic
`e(F_f)`	F for $u_i = 0$
`e(df_r)`	residual degrees of freedom
`e(df_a)`	degrees of freedom for absorbed effect
`e(df_b)`	numerator degrees of freedom for F statistic
`e(rmse)`	root mean squared error
`e(Tbar)`	harmonic mean of group sizes
`e(rank)`	rank of `e(V)`

Macros

`e(cmd)`	`xtregar`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(ivar)`	variable denoting groups
`e(tvar)`	variable denoting time within groups
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(model)`	`fe`
`e(rhotype)`	method of estimating ρ_{ar}
`e(dw)`	`lbi`, if `lbi` specified
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(marginsok)`	predictions allowed by `margins`
`e(marginsnotok)`	predictions disallowed by `margins`

`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`
Matrices	
`e(b)`	coefficient vector
`e(V)`	variance–covariance matrix of the estimators
Functions	
`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices	
`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

Consider a linear panel-data model described by (1) and (2). The data can be unbalanced and unequally spaced. Specifically, the dataset contains observations on individual i at times t_{ij} for $j = 1, \dots, n_i$.

Methods and formulas are presented under the following headings:

Estimating ρ

The estimate of ρ is always obtained after removing the group means. Let $\tilde{y}_{it} = y_{it} - \overline{y}_i$, let $\tilde{\mathbf{x}}_{it} = \mathbf{x}_{it} - \overline{\mathbf{x}}_i$, and let $\tilde{\epsilon}_{it} = \epsilon_{it} - \overline{\epsilon}_i$.

Then, except for the `onestep` method, all the estimates of ρ are obtained by running Stata's `prais` on

$$\tilde{y}_{it} = \tilde{x}_{it}\boldsymbol{\beta} + \tilde{\epsilon}_{it}$$

See [TS] **prais** for the formulas for each of the methods.

When `onestep` is specified, a regression is run on the above equation, and the residuals are obtained. Let $e_{it_{ij}}$ be the residual used to estimate the error $\tilde{\epsilon}_{it_{ij}}$. If $t_{ij} - t_{i,j-1} > 1$, $e_{it_{ij}}$ is set to zero. Given this series of residuals

$$\widehat{\rho}_{\text{onestep}} = \frac{n}{m_c} \frac{\sum_{i=1}^{N} \sum_{t=2}^{T} e_{it} e_{i,t-1}}{\sum_{i=1}^{N} \sum_{t=1}^{T} e_{it}^2}$$

where n is the number of nonzero elements in e and m_c is the number of consecutive pairs of nonzero e_{it}s.

Transforming the data to remove the AR(1) component

After estimating ρ, Baltagi and Wu (1999) derive a transformation of the data that removes the AR(1) component. Their $C_i(\rho)$ can be written as

$$y^*_{it_{ij}} = \begin{cases} (1-\rho^2)^{1/2} y_{it_{ij}} & \text{if } t_{ij}=1 \\ (1-\rho^2)^{1/2}\left\{ y_{i,t_{ij}} \frac{1}{\left(1-\rho^{2(t_{ij}-t_{i,j-1})}\right)^{1/2}} - y_{i,t_{i,j-1}} \frac{\rho^{(t_{ij}-t_{i,j-1})}}{\left(1-\rho^{2(t_{i,j}-t_{i,j-1})}\right)^{1/2}} \right\} & \text{if } t_{ij}>1 \end{cases}$$

Using the analogous transform on the independent variables generates transformed data without the AR(1) component. Performing simple OLS on the transformed data leaves behind the residuals μ^*.

The within estimator of the fixed-effects model

To obtain the within estimator, we must transform the data that come from the AR(1) transform. For the within transform to remove the fixed effects, the first observation of each panel must be omitted. Specifically, let

$$\begin{aligned} \breve{y}_{it_{ij}} &= y^*_{it_{ij}} - \overline{y}^*_i + \overline{\overline{y}}^* \qquad \forall j>1 \\ \breve{\mathbf{x}}_{it_{ij}} &= \mathbf{x}^*_{it_{ij}} - \overline{\mathbf{x}}^*_i + \overline{\overline{\mathbf{x}}}^* \qquad \forall j>1 \\ \breve{\epsilon}_{it_{ij}} &= \epsilon^*_{it_{ij}} - \overline{\epsilon}^*_i + \overline{\overline{\epsilon}}^* \qquad \forall j>1 \end{aligned}$$

where

$$\begin{aligned} \overline{y}^*_i &= \frac{\sum_{j=2}^{n_i-1} y^*_{it_{ij}}}{n_i-1} \\ \overline{\overline{y}}^* &= \frac{\sum_{i=1}^{N}\sum_{j=2}^{n_i-1} y^*_{it_{ij}}}{\sum_{i=1}^{N} n_i-1} \\ \overline{\mathbf{x}}^*_i &= \frac{\sum_{j=2}^{n_i-1} \mathbf{x}^*_{it_{ij}}}{n_i-1} \\ \overline{\overline{\mathbf{x}}}^* &= \frac{\sum_{i=1}^{N}\sum_{j=2}^{n_i-1} \mathbf{x}^*_{it_{ij}}}{\sum_{i=1}^{N} n_i-1} \\ \overline{\epsilon}^*_i &= \frac{\sum_{j=2}^{n_i-1} \epsilon^*_{it_{ij}}}{n_i-1} \\ \overline{\overline{\epsilon}}^* &= \frac{\sum_{i=1}^{N}\sum_{j=2}^{n_i-1} \epsilon^*_{it_{ij}}}{\sum_{i=1}^{N} n_i-1} \end{aligned}$$

The within estimator of the fixed-effects model is then obtained by running OLS on

$$\breve{y}_{it_{ij}} = \alpha + \breve{\mathbf{x}}_{it_{ij}}\boldsymbol{\beta} + \breve{\epsilon}_{it_{ij}}$$

Reported as R^2 within is the R^2 from the above regression.

Reported as R^2 between is $\left\{\text{corr}(\overline{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}, \overline{y}_i)\right\}^2$.

Reported as R^2 overall is $\left\{\text{corr}(\mathbf{x}_{it}\widehat{\boldsymbol{\beta}}, y_{it})\right\}^2$.

The Baltagi–Wu GLS estimator

The residuals μ^* can be used to estimate the variance components. Translating the matrix formulas given in Baltagi and Wu (1999) into summations yields the following variance-components estimators:

$$\widehat{\sigma}^2_\omega = \sum_{i=1}^N \frac{(\mu_i^{*\prime} g_i)^2}{(g_i' g_i)}$$

$$\widehat{\sigma}^2_\epsilon = \frac{\left[\sum_{i=1}^N (\mu_i^{*\prime}\mu_i^*) - \sum_{i=1}^N \left\{\frac{(\mu_i^{*\prime} g_i)^2}{(g_i' g_i)}\right\}\right]}{\sum_{i=1}^N (n_i - 1)}$$

$$\widehat{\sigma}^2_\mu = \frac{\left[\sum_{i=1}^N \left\{\frac{(\mu_i^{*\prime} g_i)^2}{(g_i' g_i)}\right\} - N\widehat{\sigma}^2_\epsilon\right]}{\sum_{i=1}^N (g_i' g_i)}$$

where

$$g_i = \left[1, \frac{\left\{1-\rho^{(t_{i,2}-t_{i,1})}\right\}}{\left\{1-\rho^{2(t_{i,2}-t_{i,1})}\right\}^{\frac{1}{2}}}, \ldots, \frac{\left\{1-\rho^{(t_{i,n_i}-t_{i,n_i-1})}\right\}}{\left\{1-\rho^{2(t_{i,n_i}-t_{i,n_i-1})}\right\}^{\frac{1}{2}}}\right]'$$

and μ_i^* is the $n_i \times 1$ vector of residuals from μ^* that correspond to person i.

Then

$$\widehat{\theta}_i = 1 - \left(\frac{\widehat{\sigma}_\mu}{\widehat{\omega}_i}\right)$$

where

$$\widehat{\omega}_i^2 = g_i' g_i \widehat{\sigma}^2_\mu + \widehat{\sigma}^2_\epsilon$$

With these estimates in hand, we can transform the data via

$$z^{**}_{it_{ij}} = z^*_{it_{ij}} - \widehat{\theta}_i g_{ij} \frac{\sum_{s=1}^{n_i} g_{is} z^*_{it_{is}}}{\sum_{s=1}^{n_i} g_{is}^2}$$

for $z \in \{y, \mathbf{x}\}$.

Running OLS on the transformed data $y^{**}, \mathbf{x}^{**}$ yields the feasible GLS estimator of α and $\boldsymbol{\beta}$.

Reported as R^2 between is $\left\{\text{corr}(\overline{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}, \overline{y}_i)\right\}^2$.

Reported as R^2 within is $\left\{\text{corr}\{(\mathbf{x}_{it} - \overline{\mathbf{x}}_i)\widehat{\boldsymbol{\beta}}, y_{it} - \overline{y}_i\}\right\}^2$.

Reported as R^2 overall is $\left\{\text{corr}(\mathbf{x}_{it}\widehat{\boldsymbol{\beta}}, y_{it})\right\}^2$.

The test statistics

The Baltagi–Wu LBI is the sum of terms

$$d_* = d_1 + d_2 + d_3 + d_4$$

where

$$d_1 = \frac{\sum_{i=1}^N \sum_{j=1}^{n_i} \{\tilde{z}_{it_{ij}} - \tilde{z}_{it_{i,j-1}} I(t_{ij} - t_{i,j-1} = 1)\}^2}{\sum_{i=1}^N \sum_{j=1}^{n_i} \tilde{z}^2_{it_{ij}}}$$

$$d_2 = \frac{\sum_{i=1}^N \sum_{j=1}^{n_i-1} \tilde{z}^2_{it_{ij}} \{1 - I(t_{i,j+1} - t_{ij} = 1)\}}{\sum_{i=1}^N \sum_{j=1}^{n_i} \tilde{z}^2_{it_{ij}}}$$

$$d_3 = \frac{\sum_{i=1}^N \tilde{z}^2_{it_{i1}}}{\sum_{i=1}^N \sum_{j=1}^{n_i} \tilde{z}^2_{it_{ij}}}$$

$$d_4 = \frac{\sum_{i=1}^N \tilde{z}^2_{it_{in_i}}}{\sum_{i=1}^N \sum_{j=1}^{n_i} \tilde{z}^2_{it_{ij}}}$$

$I()$ is the indicator function that takes the value of 1 if the condition is true and 0 otherwise. The $\tilde{z}_{it_{i,j-1}}$ are residuals from the within estimator.

Baltagi and Wu (1999) also show that d_1 is the Bhargava et al. Durbin–Watson statistic modified to handle cases of unbalanced panels and unequally spaced data.

Acknowledgment

We thank Badi Baltagi of the Department of Economics at Syracuse University for his helpful comments.

References

Baltagi, B. H. 2009. *A Companion to Econometric Analysis of Panel Data*. Chichester, UK: Wiley.

———. 2013. *Econometric Analysis of Panel Data*. 5th ed. Chichester, UK: Wiley.

Baltagi, B. H., and Q. Li. 1991. A transformation that will circumvent the problem of autocorrelation in an error-component model. *Journal of Econometrics* 48: 385–393. https://doi.org/10.1016/0304-4076(91)90070-T.

Baltagi, B. H., and P. X. Wu. 1999. Unequally spaced panel data regressions with AR(1) disturbances. *Econometric Theory* 15: 814–823. https://doi.org/10.1017/S0266466699156020.

Bhargava, A., L. Franzini, and W. Narendranathan. 1982. Serial correlation and the fixed effects model. *Review of Economic Studies* 49: 533–549. https://doi.org/10.2307/2297285.

Drukker, D. M. 2003. Testing for serial correlation in linear panel-data models. *Stata Journal* 3: 168–178.

Hoechle, D. 2007. Robust standard errors for panel regressions with cross-sectional dependence. *Stata Journal* 7: 281–312.

Mundlak, Y. 1978. On the pooling of time series and cross section data. *Econometrica* 46: 69–85. https://doi.org/10.2307/1913646.

Sosa-Escudero, W., and A. K. Bera. 2008. Tests for unbalanced error-components models under local misspecification. *Stata Journal* 8: 68–78.

Wursten, J. 2018. Testing for serial correlation in fixed-effects panel models. *Stata Journal* 18: 76–100.

Also see

[XT] **xtregar postestimation** — Postestimation tools for xtregar

[XT] **xtgee** — GEE population-averaged panel-data models

[XT] **xtgls** — GLS linear model with heteroskedastic and correlated errors

[XT] **xtreg** — Linear models for panel data

[XT] **xtset** — Declare data to be panel data

[TS] **newey** — Regression with Newey–West standard errors

[TS] **prais** — Prais–Winsten and Cochrane–Orcutt regression

[U] 20 Estimation and postestimation commands

Postestimation commands   predict   margins   Also see

Postestimation commands

The following postestimation commands are available after `xtregar`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of parameters
* `estat ic`	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC, respectively)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
`forecast`	dynamic forecasts and simulations
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	linear predictions, residuals, error components
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of parameters
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

*`estat ic` is not appropriate after `xtregar, re`.

predict

Description for predict

`predict` creates a new variable containing predictions such as linear predictions and predictions.

Menu for predict

Statistics > Postestimation

Syntax for predict

`predict` [*type*] *newvar* [*if*] [*in*] [, *statistic*]

statistic	Description
Main	
`xb`	$\mathbf{x}_{it}\mathbf{b}$, linear prediction; the default
`ue`	$u_i + e_{it}$, the combined residual
* `u`	u_i, the fixed- or random-error component
* `e`	e_{it}, the overall error component

Unstarred statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample. Starred statistics are calculated only for the estimation sample, even when `if e(sample)` is not specified.

Options for predict

Main

`xb`, the default, calculates the linear prediction, $\mathbf{x}_{it}\boldsymbol{\beta}$.

`ue` calculates the prediction of $u_i + e_{it}$.

`u` calculates the prediction of u_i, the estimated fixed or random effect.

`e` calculates the prediction of e_{it}.

margins

Description for margins

`margins` estimates margins of response for linear predictions.

Menu for margins

Statistics > Postestimation

Syntax for margins

`margins` [*marginlist*] [, *options*]

`margins` [*marginlist*] , `predict(`*statistic* ...`)` [*options*]

statistic	Description
`xb`	$\mathbf{x}_{it}\mathbf{b}$, linear prediction; the default
`ue`	not allowed with `margins`
`u`	not allowed with `margins`
`e`	not allowed with `margins`

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.

For the full syntax, see [R] **margins**.

Also see

[XT] **xtregar** — Fixed- and random-effects linear models with an AR(1) disturbance

[U] 20 Estimation and postestimation commands

Description

`xtset` manages the panel settings of a dataset. You must `xtset` your data before you can use the other `xt` commands. `xtset` *panelvar* declares the data in memory to be a panel in which the order of observations is irrelevant. `xtset` *panelvar timevar* declares the data to be a panel in which the order of observations is relevant. When you specify *timevar*, you can then use Stata's time-series operators and analyze your data with the `ts` commands without having to `tsset` your data.

`xtset` without arguments displays how the data are currently `xtset`. If the data are set with a *panelvar* and a *timevar*, `xtset` also sorts the data by *panelvar timevar* if a *timevar* was specified. If the data are set with a *panelvar* only, the sort order is not changed.

`xtset, clear` is a rarely used programmer's command to declare that the data are no longer to be considered a panel.

Quick start

Declare dataset to be panel data with panel identifier `pvar`

```
xtset pvar
```

Indicate that observations are ordered by year, stored in `tvar1`

```
xtset pvar tvar1
```

Same as above, but indicate that observations are instead made every 2 years

```
xtset pvar tvar1, delta(2)
```

Indicate that observations are made monthly; `tvar2` is not formatted

```
xtset pvar tvar2, monthly
```

Same as above, and apply `%tm` format to `tvar2`

```
xtset pvar tvar2, format(%tm)
```

Menu

Statistics > Longitudinal/panel data > Setup and utilities > Declare dataset to be panel data

Syntax

Declare data to be panel

`xtset` *panelvar*

`xtset` *panelvar timevar* [, *tsoptions*]

Display how data are currently xtset

`xtset`

Clear xt settings

`xtset, clear`

In the declare syntax, *panelvar* identifies the panels and the optional *timevar* identifies the times within panels. *tsoptions* concern *timevar*.

tsoptions	Description
unitoptions	specify units of *timevar*
deltaoption	specify length of period of *timevar*
`noquery`	suppress summary calculations and output

`collect` is allowed; see **[U] 11.1.10 Prefix commands**.
`noquery` is not shown in the dialog box.

unitoptions	Description
(default)	*timevar*'s units from *timevar*'s display format
`clocktime`	*timevar* is `%tc`: 0 = 1jan1960 00:00:00.000, 1 = 1jan1960 00:00:00.001, ...
`daily`	*timevar* is `%td`: 0 = 1jan1960, 1 = 2jan1960, ...
`weekly`	*timevar* is `%tw`: 0 = 1960w1, 1 = 1960w2, ...
`monthly`	*timevar* is `%tm`: 0 = 1960m1, 1 = 1960m2, ...
`quarterly`	*timevar* is `%tq`: 0 = 1960q1, 1 = 1960q2,...
`halfyearly`	*timevar* is `%th`: 0 = 1960h1, 1 = 1960h2,...
`yearly`	*timevar* is `%ty`: 1960 = 1960, 1961 = 1961, ...
`generic`	*timevar* is `%tg`: 0 = ?, 1 = ?, ...
`format(`*%fmt*`)`	specify *timevar*'s format and then apply default rule

In all cases, negative *timevar* values are allowed.

deltaoption specifies the period between observations in *timevar* units and may be specified as

deltaoption	Example
`delta(#)`	`delta(1)` or `delta(2)`
`delta((`*exp*`))`	`delta((7*24))`
`delta(#` *units*`)`	`delta(7 days)` or `delta(15 minutes)` or `delta(7 days 15 minutes)`
`delta((`*exp*`)` *units*`)`	`delta((2+3) weeks)`

Allowed units for `%tc` and `%tC` *timevars* are

`seconds`	`second`	`secs`	`sec`
`minutes`	`minute`	`mins`	`min`
`hours`	`hour`		
`days`	`day`		
`weeks`	`week`		

and for all other `%t` *timevars* are

`days`	`day`
`weeks`	`week`

Options

unitoptions `clocktime`, `daily`, `weekly`, `monthly`, `quarterly`, `halfyearly`, `yearly`, `generic`, and `format(`*%fmt*`)` specify the units in which *timevar* is recorded.

timevar will usually be a variable that counts 1, 2, ..., and is to be interpreted as first year of survey, second year, ..., or first month of treatment, second month, ... In these cases, you do not need to specify a *unitoption*.

In other cases, *timevar* will be a year variable or the like such as 2001, 2002, ..., and is to be interpreted as year of survey or the like. In those cases, you do not need to specify a *unitoption*.

In other, more complicated cases, *timevar* will be a full-blown `%t` variable; see [D] **Datetime**. If *timevar* already has a `%t` display format assigned to it, you do not need to specify a *unitoption*; `xtset` will obtain the units from the format. If you have not yet bothered to assign the appropriate `%t` format to the `%t` variable, however, you can use the *unitoptions* to tell `xtset` the units. Then `xtset` will set *timevar*'s display format for you. Thus, the *unitoptions* are convenience options; they allow you to skip formatting the time variable. The following all have the same net result:

Alternative 1	Alternative 2	Alternative 3
`format t %td`	*(t not formatted)*	*(t not formatted)*
`xtset pid t`	`xtset pid t, daily`	`xtset pid t, format(%td)`

timevar is not required to be a `%t` variable; it can be any variable of your own concocting so long as it takes on only integer values. When you `xtset` a time variable that is not `%t`, the display format does not change unless you specify the *unitoption* `generic` or use the `format()` option.

`delta()` specifies the period between observations in *timevar* and is commonly used when *timevar* is `%tc`. `delta()` is only sometimes used with the other `%t` formats or with generic time variables.

If `delta()` is not specified, `delta(1)` is assumed. This means that at *timevar* $= 5$, the previous time is *timevar* $= 5 - 1 = 4$ and the next time would be *timevar* $= 5 + 1 = 6$. Lag and lead operators, for instance, would work this way. This would be assumed regardless of the units of *timevar*.

If you specified `delta(2)`, then at *timevar* $= 5$, the previous time would be *timevar* $= 5 - 2 = 3$ and the next time would be *timevar* $= 5 + 2 = 7$. Lag and lead operators would work this way. In the observation with *timevar* $= 5$, `L.income` would be the value of `income` in the observation for which *timevar* $= 3$ and `F.income` would be the value of `income` in the observation for which *timevar* $= 7$. If you then add an observation with *timevar* $= 4$, the operators will still work appropriately; that is, at *timevar* $= 5$, `L.income` will still have the value of `income` at *timevar* $= 3$.

There are two aspects of *timevar*: its units and its length of period. The *unitoptions* set the units. `delta()` sets the length of period. You are not required to specify one to specify the other. You might have a generic *timevar* but it counts in 12: 0, 12, 24, ... You would skip specifying *unitoptions* but would specify `delta(12)`.

We mentioned that `delta()` is commonly used with `%tc` *timevars* because Stata's `%tc` variables have units of milliseconds. If `delta()` is not specified and in some model you refer to `L.bp`, you will be referring to the value of `bp` 1 ms ago. Few people have data with periodicity of a millisecond. Perhaps your data are hourly. You could specify `delta(3600000)`. Or you could specify `delta((60*60*1000))`, because `delta()` will allow expressions if you include an extra pair of parentheses. Or you could specify `delta(1 hour)`. They all mean the same thing: *timevar* has periodicity of 3,600,000 ms. In an observation for which *timevar* $=$ 1,489,572,000,000 (corresponding to 15mar2007 10:00:00), `L.bp` would be the observation for which *timevar* $=$ 1,489,572,000,000 $-$ 3,600,000 $=$ 1,489,568,400,000 (corresponding to 15mar2007 9:00:00).

When you `xtset` the data and specify `delta()`, `xtset` verifies that all the observations follow the specified periodicity. For instance, if you specified `delta(2)`, then *timevar* could contain any subset of $\{\ldots, -4, -2, 0, 2, 4, \ldots\}$ or it could contain any subset of $\{\ldots, -3, -1, 1, 3, \ldots\}$. If *timevar* contained a mix of values, `xtset` would issue an error message. The check is made on each panel independently, so one panel might contain *timevar* values from one set and the next, another, and that would be fine.

`clear`—used in `xtset, clear`—makes Stata forget that the data ever were `xtset`. This is a rarely used programmer's option.

The following option is available with `xtset` but is not shown in the dialog box:

`noquery` prevents `xtset` from performing most of its summary calculations and suppresses output. With this option, only the following results are posted:

`r(tdelta)`	`r(tsfmt)`
`r(panelvar)`	`r(unit)`
`r(timevar)`	`r(unit1)`

Remarks and examples

`xtset` declares the dataset in memory to be panel data. You need to do this before you can use the other `xt` commands. The storage types of both *panelvar* and *timevar* must be numeric, and both variables must contain integers only.

There are two syntaxes for setting the data:

`xtset` *panelvar*
`xtset` *panelvar timevar*

In the first syntax—`xtset` *panelvar*—the data are set to be a panel and the order of the observations within panel is considered to be irrelevant. For instance, *panelvar* might be country and the observations within might be city.

In the second syntax—`xtset` *panelvar timevar*—the data are to be a panel and the order of observations within panel are considered ordered by *timevar*. For instance, in data collected from repeated surveying of the same people over various years, *panelvar* might be person and *timevar*, year. When you specify *timevar*, you may then use Stata's time-series operators such as `L.` and `F.` (lag and lead) in other commands. The operators will be interpreted as lagged and lead values within panel.

The storage types of both *panelvar* and *timevar* must be numeric, and both variables must contain integers only.

❑ Technical note

In previous versions of Stata there was no `xtset` command. The other xt commands instead had the `i(`*panelvar*`)` and `t(`*timevar*`)` options. Older commands still have those options, but they are no longer documented and, if you specify them, they just perform the `xtset` for you. Thus, do-files that you previously wrote will continue to work. Modern usage, however, is to `xtset` the data first.

❑

❑ Technical note

`xtset` is related to the `tsset` command, which declares data to be time series. One of the syntaxes of `tsset` is `tsset` *panelvar timevar*, which is identical to one of `xtset`'s syntaxes, namely, `xtset` *panelvar timevar*. Here they are in fact the same command, meaning that `xtset`ting your data is sufficient to allow you to use the `ts` commands and `tsset`ting your data is sufficient to allow you to use the xt commands. You do not need to set both, but it will not matter if you do.

`xtset` and `tsset` are different, however, when you set just a *panelvar*—you type `xtset` *panelvar*—or when you set just a *timevar*—you type `tsset` *timevar*.

❑

If you `save` your data after `xtset`, the data will be remembered to be a panel and you will not have to `xtset` again.

▷ Example 1: Panel data without a time variable

Many panel datasets contain a variable identifying panels but do not contain a time variable. For example, you may have a dataset where each panel is a family, and the observations within panel are family members, or you may have a dataset in which each person made a decision multiple times but the ordering of those decisions is unimportant and perhaps unknown. In this latter case, if the time of the decision were known, we would advise you to `xtset` it. The other xt statistical commands do not do something different because *timevar* has been set—they will ignore *timevar* if *timevar* is irrelevant to the statistical method that you are using. You should always set everything that is true about the data.

In any case, let's consider the case where there is no *timevar*. We have data on US states and cities within states:

```
. list state city in 1/10, sepby(state)
```

	state	city
1.	Alabama	Birmingham
2.	Alabama	Mobile
3.	Alabama	Montgomery
4.	Alabama	Huntsville
5.	Alaska	Anchorage
6.	Alaska	Fairbanks
7.	Arizona	Phoenix
8.	Arizona	Tucson
9.	Arkansas	Fayetteville
10.	Arkansas	Fort Smith

Here we do not type `xtset state city` because `city` is not a time variable. Instead, we type `xtset state`:

```
. xtset state
string variables not allowed in varlist;
state is a string variable
r(109);
```

You cannot `xtset` a string variable. We must make a numeric variable from our string variable and `xtset` that. One alternative is

```
. egen statenum = group(state)
. list state statenum in 1/10, sepby(state)
```

	state	statenum
1.	Alabama	1
2.	Alabama	1
3.	Alabama	1
4.	Alabama	1
5.	Alaska	2
6.	Alaska	2
7.	Arizona	3
8.	Arizona	3
9.	Arkansas	4
10.	Arkansas	4

```
. xtset statenum
Panel variable: statenum (unbalanced)
```

Perhaps a better alternative is

```
. encode state, gen(st)
. list state st in 1/10, sepby(state)

     +---------------------+
     |    state         st |
     |---------------------|
  1. |  Alabama    Alabama |
  2. |  Alabama    Alabama |
  3. |  Alabama    Alabama |
  4. |  Alabama    Alabama |
     |---------------------|
  5. |   Alaska     Alaska |
  6. |   Alaska     Alaska |
     |---------------------|
  7. |  Arizona    Arizona |
  8. |  Arizona    Arizona |
     |---------------------|
  9. | Arkansas   Arkansas |
 10. | Arkansas   Arkansas |
     +---------------------+
```

encode (see [D] **encode**) produces a numeric variable with a value label, so when we list the result, new variable st looks just like our original. It is, however, numeric:

```
. list state st in 1/10, nolabel sepby(state)

     +---------------+
     |    state   st |
     |---------------|
  1. |  Alabama    1 |
  2. |  Alabama    1 |
  3. |  Alabama    1 |
  4. |  Alabama    1 |
     |---------------|
  5. |   Alaska    2 |
  6. |   Alaska    2 |
     |---------------|
  7. |  Arizona    3 |
  8. |  Arizona    3 |
     |---------------|
  9. | Arkansas    4 |
 10. | Arkansas    4 |
     +---------------+
```

We can xtset new variable st:

```
. xtset st
Panel variable: st (unbalanced)
```

◁

▷ Example 2: Panel data with a time variable

Some panel datasets do contain a time variable. Dataset `abdata.dta` contains labor demand data from a panel of firms in the United Kingdom. Here are wage data for the first two firms in the dataset:

```
. use https://www.stata-press.com/data/r19/abdata, clear

. list id year wage if id==1 | id==2, sepby(id)

     +----------------------+
     | id   year      wage  |
     |----------------------|
  1. |  1   1977   13.1516  |
  2. |  1   1978   12.3018  |
  3. |  1   1979   12.8395  |
  4. |  1   1980   13.8039  |
  5. |  1   1981   14.2897  |
  6. |  1   1982   14.8681  |
  7. |  1   1983   13.7784  |
     |----------------------|
  8. |  2   1977   14.7909  |
  9. |  2   1978   14.1036  |
 10. |  2   1979   14.9534  |
 11. |  2   1980    15.491  |
 12. |  2   1981   16.1969  |
 13. |  2   1982   16.1314  |
 14. |  2   1983   16.3051  |
     +----------------------+
```

To declare this dataset as a panel dataset, you type

```
. xtset id year, yearly

Panel variable: id (unbalanced)
 Time variable: year, 1976 to 1984
         Delta: 1 year
```

The output from `list` shows that the last observations for these two firms are for 1983, but `xtset` shows that for some firms data are available for 1984 as well. If one or more panels contain data for nonconsecutive periods, `xtset` will report that gaps exist in the time variable. For example, if we did not have data for firm 1 for 1980 but did have data for 1979 and 1981, `xtset` would indicate that our data have a gap.

For yearly data, we could omit the `yearly` option and just type `xtset id year` because years are stored and listed just like regular integers.

Having declared our data to be a panel dataset, we can use time-series operators to obtain lags:

```
. list id year wage L.wage if id==1 | id==2, sepby(id)

     +-----------------------------------+
     |                                L. |
     | id   year      wage         wage  |
     |-----------------------------------|
  1. |  1   1977   13.1516            . |
  2. |  1   1978   12.3018      13.1516 |
  3. |  1   1979   12.8395      12.3018 |
  4. |  1   1980   13.8039      12.8395 |
  5. |  1   1981   14.2897      13.8039 |
  6. |  1   1982   14.8681      14.2897 |
  7. |  1   1983   13.7784      14.8681 |
     |-----------------------------------|
  8. |  2   1977   14.7909            . |
  9. |  2   1978   14.1036      14.7909 |
 10. |  2   1979   14.9534      14.1036 |
 11. |  2   1980    15.491      14.9534 |
 12. |  2   1981   16.1969       15.491 |
 13. |  2   1982   16.1314      16.1969 |
 14. |  2   1983   16.3051      16.1314 |
     +-----------------------------------+
```

`L.wage` is missing for 1977 in both panels because we have no wage data for 1976. In observation 8, the lag operator did not incorrectly reach back into the previous panel.

◁

❑ Technical note

The terms *balanced* and *unbalanced* are often used to describe whether a panel dataset is missing some observations. If a dataset does not contain a time variable, then panels are considered *balanced* if each panel contains the same number of observations; otherwise, the panels are *unbalanced*.

When the dataset contains a time variable, panels are said to be *strongly balanced* if each panel contains the same time points, *weakly balanced* if each panel contains the same number of observations but not the same time points, and *unbalanced* otherwise.

❑

▷ Example 3: Applying time-series formats to the time variable

If our data are observed more than once per year, applying time-series formats to the time variable can improve readability.

We have a dataset consisting of individuals who joined a gym's weight-loss program that began in January 2005 and ended in December 2005. Each participant's weight was recorded once per month. Some participants did not show up for all the monthly weigh-ins, so we do not have all 12 months' records for each person. The first two people's data are

```
. use https://www.stata-press.com/data/r19/gymdata
. list id month wt if id==1 | id==2, sepby(id)
```

	id	month	wt
1.	1	1	145
2.	1	2	144
	(output omitted)		
11.	1	11	124
12.	1	12	120
13.	2	1	144
14.	2	2	143
	(output omitted)		
23.	2	11	122
24.	2	12	118

To set these data, we can type

```
. xtset id month
Panel variable: id (unbalanced)
 Time variable: month, 1 to 12, but with gaps
         Delta: 1 unit
```

The note "but with gaps" above is no cause for concern. It merely warns us that, within some panels, some time values are missing. We already knew that about our data—some participants did not show up for the monthly weigh-ins.

The rest of this example concerns making output more readable. Month numbers such as 1, 2, ..., 12 are perfectly readable here. In another dataset, where month numbers went to, say 127, they would not be so readable. In such cases, we can make a more readable date—2005m1, 2005m2, ...—by using Stata's `%t` variables. For a discussion, see [D] **Datetime**. We will go quickly here. One of the `%t` formats is `%tm`—monthly—and it says that 1 means 1960m1. Thus, we need to recode our `month` variable so that, rather than taking on values from 1 to 12, it takes on values from 540 to 551. Then we can put a `%tm` format on that variable. Working out 540–551 is subject to mistakes. Stata function `tm(2005m1)` tells us the `%tm` month corresponding to January of 2005, so we can type

```
. generate month2 = month + m(2005m1) - 1
. format month2 %tm
```

New variable `month2` will work just as well as the original `month` in an `xtset`, and even a little better, because output will be a little more readable:

```
. xtset id month2
Panel variable: id (unbalanced)
 Time variable: month2, 2005m1 to 2005m12, but with gaps
         Delta: 1 month
```

By the way, we could have omitted typing `format month2 %tm` and then, rather than typing `xtset id month2`, we would have typed `xtset id month2, monthly`. The `monthly` option specifies that the time variable is `%tm`. When we did not specify the option, `xtset` determined that it was monthly from the display format we had set.

◁

▷ Example 4: Clock times

We have data from a large hotel in Las Vegas that changes the reservation prices for its rooms hourly. A piece of the data looks like

```
. list in 1/5

     ┌──────────────────────────────────────┐
     │ roomtype               time    price │
     ├──────────────────────────────────────┤
  1. │        1   02.13.2007 08:00      140 │
  2. │        1   02.13.2007 09:00      155 │
  3. │        1   02.13.2007 10:00      160 │
  4. │        1   02.13.2007 11:00      155 │
  5. │        1   02.13.2007 12:00      160 │
     └──────────────────────────────────────┘
```

The panel variable is `roomtype` and, although you cannot see it from the output above, it takes on 1, 2, . . . , 20. Variable `time` is a string variable. The first step in making this dataset xt is to translate the string to a numeric variable:

```
. generate double t = clock(time, "MDY hm")

. list in 1/5

     ┌─────────────────────────────────────────────────┐
     │ roomtype               time    price          t │
     ├─────────────────────────────────────────────────┤
  1. │        1   02.13.2007 08:00      140   1.487e+12 │
  2. │        1   02.13.2007 09:00      155   1.487e+12 │
  3. │        1   02.13.2007 10:00      160   1.487e+12 │
  4. │        1   02.13.2007 11:00      155   1.487e+12 │
  5. │        1   02.13.2007 12:00      160   1.487e+12 │
     └─────────────────────────────────────────────────┘
```

See [D] **Datetime conversion** for an explanation of what is going on here. `clock()` is the function that converts strings to datetime (`%tc`) values. We typed `clock(time, "MDY hm")` to convert string variable `time`, and we told `clock()` that the values in `time` were in the order month, day, year, hour, and minute. We stored new variable `t` as a `double` because time values are large and that is required to prevent rounding. Even so, the resulting values 1.487e+12 look rounded, but that is only because of the default display format for new variables. We can see the values better if we change the format:

```
. format t %20.0gc

. list in 1/5

     ┌─────────────────────────────────────────────────────────┐
     │ roomtype               time    price                  t │
     ├─────────────────────────────────────────────────────────┤
  1. │        1   02.13.2007 08:00      140  1,486,972,800,000 │
  2. │        1   02.13.2007 09:00      155  1,486,976,400,000 │
  3. │        1   02.13.2007 10:00      160  1,486,980,000,000 │
  4. │        1   02.13.2007 11:00      155  1,486,983,600,000 │
  5. │        1   02.13.2007 12:00      160  1,486,987,200,000 │
     └─────────────────────────────────────────────────────────┘
```

Even better would be to change the format to %tc—Stata's clock-time format:

```
. format t %tc
. list in 1/5
```

	roomtype	time	price	t
1.	1	02.13.2007 08:00	140	13feb2007 08:00:00
2.	1	02.13.2007 09:00	155	13feb2007 09:00:00
3.	1	02.13.2007 10:00	160	13feb2007 10:00:00
4.	1	02.13.2007 11:00	155	13feb2007 11:00:00
5.	1	02.13.2007 12:00	160	13feb2007 12:00:00

We could drop variable `time`. New variable `t` contains the same information as `time` and `t` is better because it is a Stata time variable, the most important property of which being that it is numeric rather than string. We can `xtset` it. Here, however, we also need to specify the length of the periods with `xtset`'s `delta()` option. Stata's time variables are numeric, but they record milliseconds since 01jan1960 00:00:00. By default, `xtset` uses `delta(1)`, and that means the time-series operators would not work as we want them to work. For instance, `L.price` would look back only 1 ms (and find nothing). We want `L.price` to look back 1 hour (3,600,000 ms):

```
. xtset roomtype t, delta(1 hour)
Panel variable: roomtype (strongly balanced)
 Time variable: t, 13feb2007 08:00:00 to 31mar2007 18:00:00, but with gaps
         Delta: 1 hour
. list t price l.price in 1/5
```

	t	price	L. price
1.	13feb2007 08:00:00	140	.
2.	13feb2007 09:00:00	155	140
3.	13feb2007 10:00:00	160	155
4.	13feb2007 11:00:00	155	160
5.	13feb2007 12:00:00	160	155

◁

▷ Example 5: Clock times must be double

In the previous example, it was of vital importance that when we generated the %tc variable `t`,

```
. generate double t = clock(time, "MDY hm")
```

we generated it as a `double`. Let's see what would have happened had we forgotten and just typed `generate t = clock(time, "MDY hm")`. Let's go back and start with the same original data:

```
. list in 1/5
```

	roomtype	time	price
1.	1	02.13.2007 08:00	140
2.	1	02.13.2007 09:00	155
3.	1	02.13.2007 10:00	160
4.	1	02.13.2007 11:00	155
5.	1	02.13.2007 12:00	160

Remember, variable `time` is a string variable, and we need to translate it to numeric. So we translate, but this time we forget to make the new variable a `double`:

```
. generate t = clock(time, "MDY hm")
. list in 1/5
```

	roomtype	time	price	t
1.	1	02.13.2007 08:00	140	1.49e+12
2.	1	02.13.2007 09:00	155	1.49e+12
3.	1	02.13.2007 10:00	160	1.49e+12
4.	1	02.13.2007 11:00	155	1.49e+12
5.	1	02.13.2007 12:00	160	1.49e+12

We see the first difference—`t` now lists as 1.49e+12 rather than 1.487e+12 as it did previously—but this is nothing that would catch our attention. We would not even know that the value is different. Let's continue.

We next put a `%20.0gc` format on `t` to better see the numerical values. In fact, that is not something we would usually do in an analysis. We did that in the example to emphasize to you that the `t` values were really big numbers. We will repeat the exercise just to be complete, but in real analysis, we would not bother.

```
. format t %20.0gc
. list in 1/5
```

	roomtype	time	price	t
1.	1	02.13.2007 08:00	140	1,486,972,780,544
2.	1	02.13.2007 09:00	155	1,486,976,450,560
3.	1	02.13.2007 10:00	160	1,486,979,989,504
4.	1	02.13.2007 11:00	155	1,486,983,659,520
5.	1	02.13.2007 12:00	160	1,486,987,198,464

Okay, we see big numbers in `t`. Let's continue.

Next we put a `%tc` format on `t`, and that is something we would usually do, and you should always do. You should also list a bit of the data, as we did:

```
. format t %tc
. list in 1/5
```

	roomtype	time	price	t
1.	1	02.13.2007 08:00	140	13feb2007 07:59:40
2.	1	02.13.2007 09:00	155	13feb2007 09:00:50
3.	1	02.13.2007 10:00	160	13feb2007 09:59:49
4.	1	02.13.2007 11:00	155	13feb2007 11:00:59
5.	1	02.13.2007 12:00	160	13feb2007 11:59:58

By now, you should see a problem: the translated datetime values are off by a second or two. That was caused by rounding. Dates and times should be the same, not approximately the same, and when you see a difference like this, you should say to yourself, "The translation is off a little. Why is that?" and then you should think, "Of course, rounding. I bet that I did not create `t` as a `double`."

Let's assume, however, that you do not do this. You instead plow ahead:

```
. xtset roomtype t, delta(1 hour)
time values with period less than delta() found
r(451);
```

And that is what will happen when you forget to create `t` as a `double`. The rounding will cause uneven period, and `xtset` will complain.

By the way, it is important only that clock times (`%tc` and `%tC` variables) be stored as `doubles`. The other date values `%td`, `%tw`, `%tm`, `%tq`, `%th`, and `%ty` are small enough that they can safely be stored as `floats`, although forgetting and storing them as `doubles` does no harm.

◁

❑ Technical note

Stata provides two clock-time formats, `%tc` and `%tC`. `%tC` provides a clock with leap seconds. Leap seconds are occasionally inserted to account for randomness of the earth's rotation, which gradually slows. Unlike the extra day inserted in leap years, the timing of when leap seconds will be inserted cannot be foretold. The authorities in charge of such matters announce a leap second approximately 6 months before insertion. Leap seconds are inserted at the end of the day, and the leap second is called 23:59:60 (that is, 11:59:60 p.m.), which is then followed by the usual 00:00:00 (12:00:00 a.m.). Most nonastronomers find these leap seconds vexing. The added seconds cause problems because of their lack of predictability—knowing how many seconds there will be between 01jan2012 and 01jan2013 is not possible—and because there are not necessarily 24 hours in a day. If you use a leap second–adjusted clock, most days have 24 hours, but a few have 24 hours and 1 second. You must look at a table to find out.

From a time-series analysis point of view, the nonconstant day causes the most problems. Let's say that you have data on blood pressure for a set of patients, taken hourly at 1:00, 2:00, ..., and that you have `xtset` your data with `delta(1 hour)`. On most days, `L24.bp` would be blood pressure at the same time yesterday. If the previous day had a leap second, however, and your data were recorded using a leap second–adjusted clock, there would be no observation `L24.bp` because 86,400 seconds before the current reading does not correspond to an on-the-hour time; 86,401 seconds before the current reading corresponds to yesterday's time. Thus, whenever possible, using Stata's `%tc` encoding rather than `%tC` is better.

When times are recorded by computers using leap second–adjusted clocks, however, avoiding `%tC` is not possible. For performing most time-series analysis, the recommended procedure is to map the `%tC` values to `%tc` and then `xtset` those. You must ask yourself whether the process you are studying is based on the clock—the nurse does something at 2 o'clock every day—or the true passage of time—the emitter spits out an electron every 86,400,000 ms.

When dealing with computer-recorded times, first find out whether the computer (and its time-recording software) use a leap second–adjusted clock. If it does, translate that to a `%tC` value. Then use function `cofC()` to convert to a `%tc` value and `xtset` that. If variable `T` contains the `%tC` value,

```
. generate double t = cofC(T)
. format t %tc
. xtset panelvar t, delta(...)
```

Function `cofC()` moves leap seconds forward: 23:59:60 becomes 00:00:00 of the next day.

❑

Stored results

`xtset` stores the following in `r()`:

Scalars

`r(imin)`	minimum panel ID
`r(imax)`	maximum panel ID
`r(tmin)`	minimum time
`r(tmax)`	maximum time
`r(tdelta)`	delta
`r(gaps)`	`1` if there are gaps, `0` otherwise

Macros

`r(panelvar)`	name of panel variable
`r(timevar)`	name of time variable
`r(tdeltas)`	formatted delta
`r(tmins)`	formatted minimum time
`r(tmaxs)`	formatted maximum time
`r(tsfmt)`	`%`*fmt* of time variable
`r(unit)`	units of time variable: `Clock`, `clock`, `daily`, `weekly`, `monthly`, `quarterly`, `halfyearly`, `yearly`, or `generic`
`r(unit1)`	units of time variable: `C`, `c`, `d`, `w`, `m`, `q`, `h`, `y`, or `""`
`r(balanced)`	`unbalanced`, `weakly balanced`, or `strongly balanced`; panels are strongly balanced if they all have the same time values, weakly balanced if same number of observations but different time values, otherwise unbalanced

References

Cox, N. J. 2024. Stata tip 158: The devil is in the delta. *Stata Journal* 24: 777–783.

Lazzaro, C. 2023. Stata tip 150: When is it appropriate to xtset a panel dataset with panelvar only? *Stata Journal* 23: 281–292.

Also see

[XT] **xtdescribe** — Describe pattern of xt data

[XT] **xtsum** — Summarize xt data

[TS] **tsset** — Declare data to be time-series data

[TS] **tsfill** — Fill in gaps in time variable

xtstreg — Random-effects parametric survival models

Description	Quick start	Menu	Syntax
Options	Remarks and examples	Stored results	Methods and formulas
References	Also see		

Description

`xtstreg` fits random-effects parametric survival-time models. The conditional distribution of the response given the random effects is assumed to be an exponential, a Weibull, a lognormal, a loglogistic, or a gamma distribution. `xtstreg` can be used with single- or multiple-record st data.

Quick start

Declare `pvar` as the panel variable using `xtset`, and declare data to be survival-time data using `stset`

```
xtset pvar
stset tvar, failure(fail)
```

Random-effects Weibull survival model with covariates `x1` and `x2`

```
xtstreg x1 x2, distribution(weibull)
```

Use accelerated failure-time metric instead of proportional-hazards metric

```
xtstreg x1 x2, distribution(weibull) time
```

Same as above, but with panel-level probability weights `wvar`

```
xtstreg x1 x2 [pw=wvar], distribution(weibull) time
```

Random-effects model with cluster–robust standard errors for panels nested within `cvar`

```
xtstreg x1 x2, distribution(weibull) vce(cluster cvar)
```

Menu

Statistics > Longitudinal/panel data > Survival models > Parametric survival models (RE)

Syntax

`xtstreg` [*indepvars*] [*if*] [*in*] [*weight*] , `distribution(`*distname*`)` [*options*]

options	Description
Model	
`noconstant`	suppress constant term
* `distribution(`*distname*`)`	specify survival distribution
`time`	use accelerated failure-time metric
`offset(`*varname*`)`	include *varname* in model with coefficient constrained to 1
`constraints(`*constraints*`)`	apply specified linear constraints
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `robust`, `cluster` *clustvar*, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`nohr`	do not report hazard ratios
`noshow`	do not show st setting information
`lrmodel`	perform the likelihood-ratio model test instead of the default Wald test
`nocnsreport`	do not display constraints
`tratio`	report time ratios
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Integration	
`intmethod(`*intmethod*`)`	integration method; *intmethod* may be `mvaghermite` (the default) or `ghermite`
`intpoints(#)`	use # quadrature points; default is `intpoints(12)`
Maximization	
maximize_options	control the maximization process; seldom used
`startgrid(`*numlist*`)`	improve starting value of the random-intercept parameter by performing a grid search
`nodisplay`	suppress display of header and coefficients
`collinear`	keep collinear variables
`coeflegend`	display legend instead of statistics

*`distribution(`*distname*`)` is required.

distname	Description
exponential	exponential survival distribution
loglogistic	loglogistic survival distribution
llogistic	synonym for loglogistic
weibull	Weibull survival distribution
lognormal	lognormal survival distribution
lnormal	synonym for lognormal
gamma	gamma survival distribution

You must stset your data before using xtstreg; see [ST] **stset**.

A panel variable must be specified. Use xtset; see [XT] **xtset**.

indepvars may contain factor variables; see **[U] 11.4.3 Factor variables**.

varlist may contain time-series operators; see **[U] 11.4.4 Time-series varlists**.

by, collect, fp, and statsby are allowed; see **[U] 11.1.10 Prefix commands**.

fweights, iweights, and pweights are allowed; see **[U] 11.1.6 weight**. Weights must be constant within panel.

startgrid(), nodisplay, collinear, and coeflegend do not appear in the dialog box.

See **[U] 20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

noconstant; see [R] **Estimation options**.

distribution(*distname*) specifies the survival model to be fit. *distname* is one of the following: exponential, loglogistic, llogistic, weibull, lognormal, lnormal, or gamma. This option is required.

time specifies that the model be fit in the accelerated failure-time metric rather than in the log relative-hazard metric. This option is valid only for the exponential and Weibull models because these are the only models that have both a proportional-hazards and an accelerated failure-time parameterization. Regardless of metric, the likelihood function is the same, and models are equally appropriate in either metric; it is just a matter of changing interpretation.

time must be specified at estimation.

offset(*varname*) specifies that *varname* be included in the fixed-effects portion of the model with the coefficient constrained to be 1.

constraints(*constraints*); see [R] **Estimation options**.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (oim), that are robust to some kinds of misspecification (robust), that allow for intragroup correlation (cluster *clustvar*), and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] ***vce_options***.

Specifying vce(robust) is equivalent to specifying vce(cluster *panelvar*); see *xtstreg and the robust VCE estimator* in *Methods and formulas*.

Reporting

level(#); see [R] **Estimation options**.

nohr, which may be specified at estimation or upon redisplaying results, specifies that coefficients rather than exponentiated coefficients be displayed, that is, that coefficients rather than hazard ratios be displayed. This option affects only how coefficients are displayed, not how they are estimated.

This option is valid only for the exponential and Weibull models because they have a natural proportional-hazards parameterization. These two models, by default, report hazards ratios (exponentiated coefficients).

noshow prevents xtstreg from showing the key st variables. This option is rarely used because most users type stset, show or stset, noshow to set once and for all whether they want to see these variables mentioned at the top of the output of every st command; see [ST] **stset**.

lrmodel, nocnsreport; see [R] **Estimation options**.

tratio specifies that exponentiated coefficients, which are interpreted as time ratios, be displayed. tratio is appropriate only for the loglogistic, lognormal, and gamma models or for the exponential and Weibull models when fit in the accelerated failure-time metric.

tratio may be specified at estimation or upon replay.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Integration

intmethod(*intmethod*), intpoints(#); see [R] **Estimation options**.

Maximization

maximize_options: difficult, technique(*algorithm_spec*), iterate(#), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), nonrtolerance, and from(*init_specs*); see [R] **Maximize**. These options are seldom used.

The following options are available with xtstreg but are not shown in the dialog box:

startgrid(*numlist*) performs a grid search to improve the starting value of the random-intercept parameter. No grid search is performed by default unless the starting value is found to be not feasible, in which case xtstreg runs startgrid(0.1 1 10) and chooses the value that works best. You may already be using a default form of startgrid() without knowing it. If you see xtstreg displaying Grid node 1, Grid node 2, ... following Grid node 0 in the iteration log, that is xtstreg doing a default search because the original starting value was not feasible.

nodisplay is for programmers. It suppresses the display of the header and the coefficients.

collinear, coeflegend; see [R] **Estimation options**.

Remarks and examples

xtstreg fits random-effects parametric survival-time models. The following discussion relies on the jargon of survival analysis; if you are not familiar with survival-time data or time-to-event data, see *Introduction* in [ST] **Survival analysis** for an overview of survival analysis in Stata and [ST] **Glossary** for a glossary of technical terms.

xtstreg is suitable only for data that have been stset as well as xtset. stset weights are not used; instead, weights must be specified at estimation. xtstreg requires "subjects", as defined by the id() option in stset, to be nested within panels identified by the *panelvar* specified in xtset. xtstreg can be used with multiple-record data, but it does not allow delayed entry or gaps.

For a Stata-specific introduction to survival analysis, see Cleves, Gould, and Marchenko (2016). Rabe-Hesketh and Skrondal (2022, chap. 14–15) is a good introductory read on applied longitudinal modeling of survival data.

▷ Example 1

In example 11 of [ST] **streg**, we fit a Weibull model with an inverse-Gaussian shared frailty to the recurrence times for catheter-insertion point infection for 38 kidney dialysis patients. In this example, the subjects are the catheter insertions, not the patients themselves. This is a function of how the data were recorded—the onset of risk occurs at the time the catheter is inserted and not, say, at the time of admission of the patient into the study. Thus we have two subjects (insertions) within each panel (patient). Each catheter insertion results in either infection (infect==1) or right-censoring (infect==0). Here we stset the data:

```
. use https://www.stata-press.com/data/r19/catheter
(Kidney data, McGilchrist and Aisbett, Biometrics, 1991)
. stset
-> stset time, failure(infect)
Survival-time data settings
         Failure event: infect!=0 & infect<.
Observed time interval: (0, time]
     Exit on or before: failure
--------------------------------------------------------------------------
         76  total observations
          0  exclusions
--------------------------------------------------------------------------
         76  observations remaining, representing
         58  failures in single-record/single-failure data
      7,424  total analysis time at risk and under observation
                                                At risk from t =         0
                                     Earliest observed entry t =         0
                                          Last observed exit t =       562
```

While it is reasonable to assume independence of patients, we would not want to assume that recurrence times within each patient are independent. The model used in [ST] **streg** allowed us to model the correlation by assuming that it was the result of a latent patient-level effect, or frailty.

Here we use xtstreg to fit a random-effects Weibull model with normally distributed random effects at the patient level. This is done by defining our panels as patients in the dataset.

```
. xtset patient
Panel variable: patient (balanced)
. xtstreg age female, distribution(weibull)
        Failure _d: infect
  Analysis time _t: time
Fitting comparison model:
Iteration 0:  Log likelihood = -1700989.9
Iteration 1:  Log likelihood =  -440.1998
Iteration 2:  Log likelihood = -336.62162
Iteration 3:  Log likelihood = -334.64937
Iteration 4:  Log likelihood = -334.57959
Iteration 5:  Log likelihood = -334.57944
Iteration 6:  Log likelihood = -334.57944
Refining starting values:
Grid node 0:  Log likelihood = -336.03903
Fitting full model:
Iteration 0:  Log likelihood = -336.03903  (not concave)
Iteration 1:  Log likelihood = -333.12188
Iteration 2:  Log likelihood = -330.38629
Iteration 3:  Log likelihood = -329.89525
Iteration 4:  Log likelihood = -329.87946
Iteration 5:  Log likelihood = -329.87938
Iteration 6:  Log likelihood = -329.87938
Random-effects Weibull PH regression            Number of obs     =         76
Group variable: patient                         Number of groups  =         38
                                                Obs per group:
                                                              min =          2
                                                              avg =        2.0
                                                              max =          2
Integration method: mvaghermite                 Integration pts.  =         12
                                                Wald chi2(2)      =      10.17
Log likelihood = -329.87938                     Prob > chi2       =     0.0062

------------------------------------------------------------------------------
          _t | Haz. ratio   Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
         age |   1.007329   .0137828     0.53   0.594     .9806742    1.034708
      female |   .1910581   .0999004    -3.17   0.002     .0685629    .5324042
       _cons |   .0073346   .0072307    -4.99   0.000     .0010623    .0506427
-------------+----------------------------------------------------------------
       /ln_p |    .222825   .1386296                     -.0488841     .494534
-------------+----------------------------------------------------------------
   /sigma2_u |   .8234583   .4812598                      .2619194    2.588902
------------------------------------------------------------------------------
Note: Estimates are transformed only in the first equation to hazard ratios.
Note: _cons estimates baseline hazard (conditional on zero random effects).
LR test vs. Weibull model: chibar2(01) = 9.40         Prob >= chibar2 = 0.0011
```

The results are similar to those in [ST] **streg**. The likelihood-ratio test compares the random-effects model with a survival model with fixed effects only. The results support the random-effects model.

For this model, xtstreg displays exponentiated coefficients, labeled as hazard ratios. These hazard ratios should be interpreted as “conditional hazard ratios”, that is, conditional on the random effects.

For example, the hazard ratio for `age` is 1.01. This means that according to the model, for a given patient, the hazard would increase 1% with each year of age. However, at the population level, marginal hazards corresponding to different levels of the covariates are not necessarily proportional; see example 5 in [ME] **mestreg postestimation** for further discussion.

The exponentiated coefficients of covariates that usually remain constant within a panel do not have a natural interpretation as conditional hazard ratios. However, the magnitude of the exponentiated coefficients always gives an idea of the effect of the covariates. In this example, `female` is constant within the panel. The estimated hazard ratio for `female` is 0.19, which indicates that hazard functions for females tend to be smaller than hazard functions for males.

◁

❑ Technical note

The random-effects model is calculated using quadrature, which is an approximation whose accuracy depends partially on the number of integration points used. We can use the `quadchk` command to see whether changing the number of integration points affects the results. If the results change, the quadrature approximation is not accurate given the number of integration points. Try increasing the number of integration points using the `intpoints()` option, and run `quadchk` again. Do not attempt to interpret the results of estimates when the coefficients reported by `quadchk` differ substantially. See [XT] **quadchk** for details and [XT] **xtprobit** for an example.

Because the `xtstreg` likelihood function is calculated by Gauss–Hermite quadrature, the computations on large problems can be slow. Computation time is roughly proportional to the number of points used for the quadrature.

❑

Stored results

`xtstreg` stores the following in `e()`:

Scalars

`e(N)`	number of observations
`e(N_g)`	number of groups
`e(k)`	number of parameters
`e(k_eq)`	number of equations in `e(b)`
`e(k_eq_model)`	number of equations in overall model test
`e(k_dv)`	number of dependent variables
`e(df_m)`	model degrees of freedom
`e(ll)`	log likelihood
`e(N_clust)`	number of clusters
`e(chi2)`	χ^2
`e(p)`	p-value for model test
`e(ll_c)`	log likelihood, comparison model
`e(chi2_c)`	χ^2, comparison model
`e(sigma_u)`	panel-level standard deviation
`e(n_quad)`	number of quadrature points
`e(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(g_max)`	largest group size
`e(rank)`	rank of `e(V)`
`e(ic)`	number of iterations
`e(rc)`	return code
`e(converged)`	`1` if converged, `0` otherwise

Macros

e(cmd)	gsem
e(cmd2)	xtstreg
e(cmdline)	command as typed
e(depvar)	_t
e(wtype)	weight type
e(wexp)	weight expression (first-level weights)
e(covariates)	list of covariates
e(ivar)	variable denoting groups
e(model)	model name
e(title)	title in estimation output
e(distribution)	distribution
e(clustvar)	name of cluster variable
e(offset)	offset
e(intmethod)	integration method
e(chi2type)	Wald; type of model χ^2
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(frm2)	hazard or time
e(opt)	type of optimization
e(which)	max or min; whether optimizer is to perform maximization or minimization
e(ml_method)	type of ml method
e(user)	name of likelihood-evaluator program
e(technique)	maximization technique
e(properties)	b V
e(predict)	program used to implement predict
e(marginsok)	predictions allowed by margins
e(marginsnotok)	predictions disallowed by margins
e(marginswtype)	weight type for margins
e(marginswexp)	weight expression for margins
e(marginsdefault)	default predict() specification for margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved

Matrices

e(b)	coefficient vector
e(Cns)	constraints matrix
e(ilog)	iteration log (up to 20 iterations)
e(gradient)	gradient vector
e(V)	variance–covariance matrix of the estimator
e(V_modelbased)	model-based variance

Functions

e(sample)	marks estimation sample

In addition to the above, the following is stored in r():

Matrices

r(table)	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

Methods and formulas are presented under the following headings:

Survival models

Survival models have a trivariate response (t_0, t, d):

t_0 is the starting time under observation $t_0 \geq 0$;

t is the ending time under observation $t \geq t_0$; and

d is an indicator for failure $d \in \{0, 1\}$.

You must `stset` your data before using `xtstreg`; see [ST] **stset**. By `stset`ting your data, you define the variables `_t0`, `_t`, and `_d`, which serve as the trivariate response variable.

Let $i = 1, \ldots, n$ panels, $j = 1, \ldots, n_i$, and ν_is be unobservable panel-level random effects that are independent and identically distributed $N(0, \sigma^2_\nu)$.

Two often-used models for adjusting survivor functions for the effects of covariates are the accelerated failure-time (AFT) model and the multiplicative or proportional hazards (PH) model.

In the AFT model, the natural logarithm of the survival time, $\log t$, is expressed as a linear function of the covariates; when we incorporate the random effects, this yields the model

$$\text{log}t_{ij} = \mathbf{x}_{ij}\boldsymbol{\beta} + \nu_i + \epsilon_{ij}$$

The $1 \times p$ row vector $\mathbf{x}_{ij}$ contains the covariates for the fixed effects, with regression coefficients (fixed effects) $\boldsymbol{\beta}$.

The ϵ_{ij} are the observation-level errors with density $\varphi(\cdot)$. The distributional form of the error term determines the regression model. Five regression models are implemented in `xtstreg` using the AFT parameterization: exponential, gamma, loglogistic, lognormal, and Weibull. The lognormal regression model is obtained by letting $\varphi(\cdot)$ be the normal density. Similarly, by letting $\varphi(\cdot)$ be the logistic density, one obtains the loglogistic regression. Setting $\varphi(\cdot)$ equal to the extreme-value density yields the exponential and the Weibull regression models.

In the PH models fit by `xtstreg`, the covariates have a multiplicative effect on the hazard function

$$h(t_{ij}) = h_0(t_{ij}) \exp(\mathbf{x}_{ij}\boldsymbol{\beta} + \nu_i)$$

for some baseline hazard function $h_0(t)$. For the `xtstreg` command, $h_0(t)$ is assumed to be parametric. The exponential and Weibull models are implemented in `xtstreg` for the PH parameterization. These two models are implemented using both the AFT and PH parameterizations.

The survivor function for a given family is the complement of the cumulative distribution function, $S(t) = 1 - F(t)$. The density for a failure at time t is given by

$$g(t) = \frac{\partial F(t)}{\partial t} = -\frac{\partial S(t)}{\partial t}$$

Some distributions contain ancillary parameters that are not denoted here.

The conditional density for a failure at time t is

$$g(t|t \geq t_0, d = 1) = g(t)/S(t_0)$$

and the conditional probability of survival without failure up to time t is

$$P(T \geq t|t \geq t_0, d = 0) = S(t)/S(t_0)$$

The conditional likelihood is given by

$$L(t_0, t, d) = \left\{ \frac{g(t)}{S(t_0)} \right\}^d \left\{ \frac{S(t)}{S(t_0)} \right\}^{1-d}$$

See *Survival distributions* in [SEM] **Methods and formulas for gsem** for the specific density function corresponding to each distribution.

Given the panel-level random effect $\boldsymbol{\nu}_i$, the conditional distribution of $\mathbf{t}_i = (t_{i1}, \ldots, t_{in_i})'$ given $\boldsymbol{\eta}_i = (\mathbf{x}_{i1}\boldsymbol{\beta} + \nu_i, \ldots, \mathbf{x}_{in_i}\boldsymbol{\beta} + \nu_i)$ for panel i is

$$f(\mathbf{t}_i|\boldsymbol{\eta}_i) = \prod_{j=1}^{n_i} f(t_{ij}|\eta_{ij})$$

where $f(t_{ij}|\eta_{ij})$ is the contribution to the likelihood from observation ij; that is,

$$f(t_{ij}|\eta_{ij}) = \left\{ \frac{g(t_{ij}|\mathbf{x}_{ij}\boldsymbol{\beta} + \nu_i)}{S(t_{0ij}|\mathbf{x}_{ij}\boldsymbol{\beta} + \nu_i)} \right\}^{d_{ij}} \left\{ \frac{S(t_{ij}|\mathbf{x}_{ij}\boldsymbol{\beta} + \nu_i)}{S(t_{0ij}|\mathbf{x}_{ij}\boldsymbol{\beta} + \nu_i)} \right\}^{1-d_{ij}}$$

where $g(t|\eta)$ and $S(t|\eta)$ are, respectively, the density and the survivor function conditional on the linear prediction η.

As mentioned above, `xtstreg` does not allow delayed entry or gaps. Therefore, the first observation for a given subject will have a value of $t_0 = 0$, and subsequent spells for the subject must start at the end of the previous spell. That is, if observations ij and $i, j+1$ belong to the same subject, then $t_{0i,j+1} = t_{ij}$.

Given a normal distribution $N(0, \sigma_\nu^2)$ for the random effect ν_i, the panel-level likelihood $\mathcal{L}_i$ is given by

$$\mathcal{L}_i(\boldsymbol{\beta}, \sigma_\nu) = \int_{-\infty}^{\infty} \frac{e^{-\nu_i^2/2\sigma_\nu^2}}{\sqrt{2\pi}\sigma_\nu} \left\{ \prod_{j=1}^{n_i} f(t_{ij}|\mathbf{x}_{ij}\boldsymbol{\beta} + \nu_i) \right\} d\nu_i \tag{1}$$

The integration in (1) has no closed form and thus must be approximated. `xtstreg` offers two approximation methods: mean–variance adaptive Gauss–Hermite quadrature (default) and nonadaptive Gauss–Hermite quadrature. For details on these integration methods, see [SEM] **Methods and formulas for gsem**. To see how panel-level weights are incorporated into the likelihood function, see *Survey data* in [SEM] **Methods and formulas for gsem**.

The log likelihood for the entire dataset is simply the sum of the contributions of the n individual panels; namely, $\mathcal{L}(\boldsymbol{\beta}, \sigma_\nu) = \sum_{i=1}^{n} \mathcal{L}_i(\boldsymbol{\beta}, \sigma_\nu)$.

Maximization of $\mathcal{L}(\boldsymbol{\beta}, \sigma_\nu)$ is performed with respect to $(\boldsymbol{\beta}, {\sigma_\nu}^2)$. Parameter estimates are stored in `e(b)` as $(\widehat{\boldsymbol{\beta}}, \widehat{\sigma}_\nu^2)$, with the corresponding variance–covariance matrix stored in `e(V)`.

xtstreg and the robust VCE estimator

Specifying `vce(robust)` or `vce(cluster` *clustvar*`)` causes the Huber/White/sandwich VCE estimator to be calculated for the coefficients estimated in this regression. See [P] **_robust**, particularly *Introduction* and *Methods and formulas*. Wooldridge (2020) and Arellano (2003) discuss this application of the Huber/White/sandwich VCE estimator. As discussed by Wooldridge (2020), Stock and Watson (2008), and Arellano (2003), specifying `vce(robust)` is equivalent to specifying `vce(cluster` *panelvar*`)`, where *panelvar* is the variable that identifies the panels.

Clustering on the panel variable produces a consistent VCE estimator when the disturbances are not identically distributed over the panels or there is serial correlation in ϵ_{it}.

The cluster–robust VCE estimator requires that there be many clusters and that the disturbances be uncorrelated across the clusters. The panel variable must be nested within the cluster variable because of the within-panel correlation that is generally induced by the random-effects transform when there is heteroskedasticity or within-panel serial correlation in the idiosyncratic errors.

References

Arellano, M. 2003. *Panel Data Econometrics*. Oxford: Oxford University Press. https://doi.org/10.1093/0199245282.001.0001.

Cleves, M. A., W. W. Gould, and Y. V. Marchenko. 2016. *An Introduction to Survival Analysis Using Stata*. Rev. 3rd ed. College Station, TX: Stata Press.

Philips, A. Q. 2020. An easy way to create duration variables in binary cross-sectional time-series data. *Stata Journal* 20: 916–930.

Rabe-Hesketh, S., and A. Skrondal. 2022. *Multilevel and Longitudinal Modeling Using Stata*. 4th ed. College Station, TX: Stata Press.

Stock, J. H., and M. W. Watson. 2008. Heteroskedasticity-robust standard errors for fixed effects panel data regression. *Econometrica* 76: 155–174. https://doi.org/10.1111/j.0012-9682.2008.00821.x.

Wooldridge, J. M. 2020. *Introductory Econometrics: A Modern Approach*. 7th ed. Boston: Cengage.

Also see

[XT] **xtstreg postestimation** — Postestimation tools for xtstreg

[XT] **quadchk** — Check sensitivity of quadrature approximation

[XT] **xtset** — Declare data to be panel data

[ME] **mestreg** — Multilevel mixed-effects parametric survival models

[ST] **streg** — Parametric survival models

[ST] **stset** — Declare data to be survival-time data

[U] 20 Estimation and postestimation commands

Postestimation commands

The following postestimation command is of special interest after `xtstreg`:

Command	Description
`stcurve`	plot the survivor, hazard, and cumulative hazard functions

The following standard postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of parameters
`estat ic`	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC, respectively)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
`lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	linear predictions and their SEs, means, medians
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of parameters
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

predict

Description for predict

`predict` creates a new variable containing predictions such as linear predictions, mean and median survival times, hazard functions, and standard errors.

Menu for predict

Statistics > Postestimation

Syntax for predict

`predict` [*type*] *newvar* [*if*] [*in*] [, *statistic* `nooffset`]

statistic	Description
Main	
`xb`	linear prediction; the default
`mean`	marginal mean survival time
`mean0`	mean survival time assuming that the random effects are zero
`median0`	median survival time assuming that the random effects are zero
`hazard`	marginal hazard
`hazard0`	hazard assuming that the random effects are zero
`surv`	marginal predicted survivor function
`surv0`	predicted survivor function assuming that the random effects are zero
`stdp`	standard error of the linear prediction

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

Options for predict

Main

`xb`, the default, calculates the linear prediction.

`mean` calculates the mean survival time that is marginal with respect to the random effect, which means that the statistic is calculated by integrating the prediction function with respect to the random effect over its entire support.

`mean0` calculates the mean survival time assuming that all random effects are zero.

`median0` calculates the median survival time assuming that all random effects are zero.

`hazard` calculates the hazard function at `_t0` that is marginal with respect to the random effect, which means that the statistic is calculated by integrating the prediction function with respect to the random effect over its entire support.

`hazard0` calculates the hazard function at `_t0`, assuming that all random effects are zero.

`surv` calculates the predicted survivor function at `_t0` that is marginal with respect to the random effect, which means that the statistic is calculated by integrating the prediction function with respect to the random effect over its entire support.

`surv0` calculates the predicted survivor function at `_t0`, assuming that all random effects are zero.

`stdp` calculates the standard error of the linear prediction.

`nooffset` is relevant only if you specified `offset(`*varname*`)` with `xtstreg`. This option modifies the calculations made by `predict` so that they ignore the offset variable; the linear prediction is treated as $\mathbf{x}_{ij}\boldsymbol{\beta}$ rather than as $\mathbf{x}_{ij}\boldsymbol{\beta} + \text{offset}_{ij}$.

margins

Description for margins

`margins` estimates margins of response for linear predictions and mean and median survival times.

Menu for margins

Statistics > Postestimation

Syntax for margins

`margins` [*marginlist*] [, *options*]

`margins` [*marginlist*] , `predict(`*statistic* ...`)` [`predict(`*statistic* ...`)` ...] [*options*]

statistic	Description
`mean`	marginal mean survival time; the default
`mean0`	mean survival time conditional on zero random effects
`median0`	median survival time conditional on zero random effects
`hazard`	marginal hazard
`surv`	marginal predicted survivor function
`xb`	linear predictor for the fixed portion of the model only
`hazard0`	not allowed with `margins`
`surv0`	not allowed with `margins`
`stdp`	not allowed with `margins`

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.

For the full syntax, see [R] **margins**.

Remarks and examples

▷ Example 1

In example 1 of [XT] **xtstreg**, we analyzed the time to infection of the catheter-insertion point for 38 kidney dialysis patients. We fit the following model:

```
. use https://www.stata-press.com/data/r19/catheter
(Kidney data, McGilchrist and Aisbett, Biometrics, 1991)
. xtset patient
  (output omitted)
. xtstreg age female, distribution(weibull)
  (output omitted)
```

The `predict` command allows us to compute the marginal mean and the mean and median survival time assuming that all random effects are zero:

```
. predict mean, mean
. predict mean0, mean0
. predict median0, median0
```

Here we list the predicted mean and median survival times for the first five patients:

```
. list patient mean mean0 median0 in 1/10, sepby(patient)
```

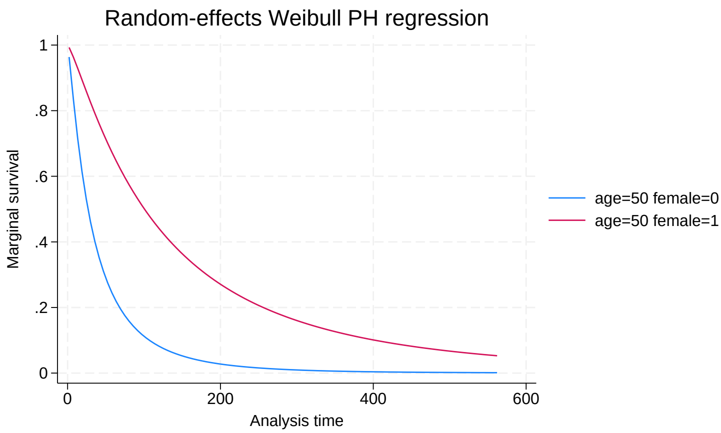
	patient	mean	mean0	median0
1.	1	60.97527	40.39634	32.34459
2.	1	60.97527	40.39634	32.34459
3.	2	204.0082	135.1562	108.217
4.	2	204.0082	135.1562	108.217
5.	3	59.56653	39.46305	31.59731
6.	3	59.56653	39.46305	31.59731
7.	4	224.6581	148.8368	119.1708
8.	4	224.6581	148.8368	119.1708
9.	5	67.7384	44.87694	35.93212
10.	5	67.7384	44.87694	35.93212

This example illustrates that for nonlinear models, the mean computed with the random effects equal to zero is usually not representative of the marginal mean.

`predict` can also compute the predicted survivor function and the predicted hazard function. All of these predictions can be marginal or conditional on the random effects being zero.

Predicted survivor, failure, hazard, or cumulative hazard functions can be visualized with `stcurve`. For example, below we compute marginal predictions for the survivor function for men and women at age 50.

```
. stcurve, survival at1(age=50 female=0) at2(age=50 female=1)
(option marginal assumed)
note: function evaluated at specified values of selected covariates and
      overall means of other covariates (if any).
```



The graph above shows that women who are 50 years old have larger survival probabilities than men of the same age.

◁

Methods and formulas

`predict` *newvar* computes the following predictions:

`mean0`:

$$newvar_{ij} = \int_0^\infty \hat{S}(t|\mathbf{x}_{ij}, u_{ij})dt$$

`median0`:

$$newvar_{ij} = \{t : \hat{S}(t|\mathbf{x}_{ij}, u_{ij}) = 1/2\}$$

`surv0`:

$$newvar_{ij} = \hat{S}(t_{ij}|\mathbf{x}_{ij}, u_{ij})$$

`hazard0`:

$$newvar_{ij} = \hat{g}(t_{ij}|\mathbf{x}_{ij}, u_{ij})/\hat{S}(t_{ij}|\mathbf{x}_{ij}, u_{ij})$$

Here $\hat{S}(t|\mathbf{x}_{ij}, u_{ij})$ is the survivor function $S(t|\mathbf{x}_{ij}\boldsymbol{\beta} + u_{ij})$, and $\hat{g}(t|\mathbf{x}_{ij}, u_{ij})$ is the density $g(t|\mathbf{x}_{ij}\boldsymbol{\beta} + u_{ij})$ with the parameter estimates substituted in for $\boldsymbol{\beta}$ and zero substituted for u_{ij}.

Also see

[XT] **xtstreg** — Random-effects parametric survival models

[ST] **stcurve** — Plot the survivor or related function after streg, stcox, and more

[U] 20 Estimation and postestimation commands

xtsum — Summarize xt data

Description | Quick start | Menu | Syntax
Remarks and examples | Stored results | Also see

Description

`xtsum`, a generalization of `summarize` (see [R] **summarize**), reports means and standard deviations for panel data; it differs from `summarize` in that it decomposes the standard deviation into between and within components.

Quick start

Report means and overall, between, and within standard deviations for all numeric variables in `xtset` data

```
xtsum
```

Same as above, but restrict to `v1`, `v2`, and `v3`

```
xtsum v1 v2 v3
```

Same as above, but calculate statistics separately for each level of `catvar`

```
bysort catvar: xtsum v1 v2 v3
```

Menu

Statistics > Longitudinal/panel data > Setup and utilities > Summarize xt data

Syntax

xtsum [*varlist*] [*if*]

A panel variable must be specified; use xtset; see [XT] **xtset**.

varlist may contain time-series operators; see [U] **11.4.4 Time-series varlists**.

by and collect are allowed; see [U] **11.1.10 Prefix commands**.

Remarks and examples

If you have not read [XT] **xt**, please do so.

xtsum provides an alternative to summarize. For instance, in the nlswork dataset described in [XT] **xt**, hours contains the usual hours worked:

```
. use https://www.stata-press.com/data/r19/nlswork
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)
. summarize hours
    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
       hours |     28,467    36.55956    9.869623          1        168
. xtsum hours
Variable         |      Mean   Std. dev.       Min        Max |    Observations
-----------------+--------------------------------------------+----------------
hours    overall |  36.55956   9.869623          1        168 |     N =   28467
         between |             7.846585          1       83.5 |     n =    4710
         within  |             7.520712  -2.154726   130.0596 | T-bar = 6.04395
```

xtsum provides the same information as summarize and more. It decomposes the variable x_{it} into a between ($\overline{x}_i$) and within ($x_{it} - \overline{x}_i + \overline{\overline{x}}$, the global mean $\overline{\overline{x}}$ being added back in make results comparable). The overall and within are calculated over 28,467 person-years of data. The between is calculated over 4,710 persons, and the average number of years a person was observed in the hours data is 6.

xtsum also reports minimums and maximums. Hours worked last week varied between 1 and (unbelievably) 168. Average hours worked last week for each woman varied between 1 and 83.5. "Hours worked within" varied between −2.15 and 130.1, which is not to say that any woman actually worked negative hours. The within number refers to the deviation from each individual's average, and naturally, some of those deviations must be negative. Then the negative value is not disturbing but the positive value is. Did some woman really deviate from her average by +130.1 hours? No. In our definition of within, we add back in the global average of 36.6 hours. Some woman did deviate from her average by $130.1 - 36.6 = 93.5$ hours, which is still large.

The reported standard deviations tell us something that may surprise you. They say that the variation in hours worked last week across women is nearly equal to that observed within a woman over time. That is, if you were to draw two women randomly from our data, the difference in hours worked is expected to be nearly equal to the difference for the same woman in two randomly selected years.

If a variable does not vary over time, its within standard deviation will be zero:

```
. xtsum birth_yr
Variable         |      Mean   Std. dev.       Min        Max |    Observations
-----------------+--------------------------------------------+----------------
birth_yr overall |  48.08509   3.012837         41         54 |     N =   28534
         between |             3.051795         41         54 |     n =    4711
         within  |                    0   48.08509   48.08509 | T-bar = 6.05689
```

Stored results

xtsum stores the following in r():

Scalars

r(N)	number of observations
r(n)	number of panels
r(Tbar)	average number of years under observation
r(mean)	mean
r(sd)	overall standard deviation
r(min)	overall minimum
r(max)	overall maximum
r(sd_b)	between standard deviation
r(min_b)	between minimum
r(max_b)	between maximum
r(sd_w)	within standard deviation
r(min_w)	within minimum
r(max_w)	within maximum

Also see

[XT] **xtdescribe** — Describe pattern of xt data

[XT] **xttab** — Tabulate xt data

xttab — Tabulate xt data

Description | Quick start | Menu | Syntax
Option | Remarks and examples | Stored results | Also see

Description

`xttab`, a generalization of `tabulate` (see [R] **tabulate oneway**), performs one-way tabulations and decomposes counts into between and within components in panel data.

`xttrans`, another generalization of `tabulate` (see [R] **tabulate oneway**), reports transition probabilities (the change in one categorical variable over time).

Quick start

Overall, between, and within one-way tabulation of `v1` using `xtset` data

```
xttab v1
```

Report transition probabilities for `v2`

```
xttrans v2
```

Add frequency of transitions

```
xttrans v2, freq
```

Same as above, but for each level of `catvar`

```
bysort catvar: xttrans v2, freq
```

Menu

xttab

Statistics > Longitudinal/panel data > Setup and utilities > Tabulate xt data

xttrans

Statistics > Longitudinal/panel data > Setup and utilities > Report transition probabilities

Syntax

xttab *varname* [*if*]

xttrans *varname* [*if*] [, freq]

A panel variable must be specified; use xtset; see [XT] **xtset**.

by and collect are allowed with xttab and xttrans; see [U] **11.1.10 Prefix commands**.

Option

Main

freq, allowed with xttrans only, specifies that frequencies as well as transition probabilities be displayed.

Remarks and examples

If you have not read [XT] **xt**, please do so.

▷ Example 1: xttab

Using the nlswork dataset described in [XT] **xt**, variable msp is 1 if a woman is married and her spouse resides with her, and 0 otherwise:

```
. use https://www.stata-press.com/data/r19/nlswork
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)

. xttab msp

                  Overall              Between            Within
      msp    Freq.  Percent        Freq.  Percent        Percent
-----------------------------------------------------------------
        0    11324    39.71         3113    66.08          62.69
        1    17194    60.29         3643    77.33          75.75
-----------------------------------------------------------------
    Total    28518   100.00         6756   143.41          69.73
                                (n = 4711)
```

The overall part of the table summarizes results in terms of person-years. We have 11,324 person-years of data in which msp is 0 and 17,194 in which it is 1—in 60.3% of our data, the woman is married with her spouse present. Between repeats the breakdown, but this time in terms of women rather than person-years; 3,113 of our women ever had msp 0 and 3,643 ever had msp 1, for a grand total of 6,756 ever having either. We have in our data, however, only 4,711 women. This means that there are women who sometimes have msp 0 and at other times have msp 1.

The within percent tells us the fraction of the time a woman has the specified value of msp. If we take the first line, conditional on a woman ever having msp 0, 62.7% of her observations have msp 0. Similarly, conditional on a woman ever having msp 1, 75.8% of her observations have msp 1. These two numbers are a measure of the stability of the msp values, and, in fact, msp 1 is more stable among these younger women than msp 0, meaning that they tend to marry more than they divorce. The total within of 69.73% is the normalized between weighted average of the within percents, that is, $(3113 \times 62.69 + 3643 \times 75.75)/6756$. It is a measure of the overall stability of the msp variable.

A time-invariant variable will have a tabulation with within percents of 100:

```
. xttab race
```

race	Overall Freq.	Overall Percent	Between Freq.	Between Percent	Within Percent
White	20180	70.72	3329	70.66	100.00
Black	8051	28.22	1325	28.13	100.00
Other	303	1.06	57	1.21	100.00
Total	28534	100.00	4711	100.00	100.00
			(n = 4711)		

◁

▷ Example 2: xttrans

xttrans shows the transition probabilities. In cross-sectional time-series data, we can estimate the probability that $x_{i,t+1} = v_2$ given that $x_{it} = v_1$ by counting transitions. For instance

```
. xttrans msp
```

1 if married, spouse present	1 if married, spouse present 0	1	Total
0	80.49	19.51	100.00
1	7.96	92.04	100.00
Total	37.11	62.89	100.00

The rows reflect the initial values, and the columns reflect the final values. Each year, some 80% of the msp 0 persons in the data remained msp 0 in the next year; the remaining 20% became msp 1. Although msp 0 had a 20% chance of becoming msp 1 in each year, the msp 1 had only an 8% chance of becoming (or returning to) msp 0. The freq option displays the frequencies that go into the calculation:

```
. xttrans msp, freq
```

1 if married, spouse present	1 if married, spouse present 0	1	Total
0	7,697 80.49	1,866 19.51	9,563 100.00
1	1,133 7.96	13,100 92.04	14,233 100.00
Total	8,830 37.11	14,966 62.89	23,796 100.00

◁

❑ Technical note

The transition probabilities reported by `xttrans` are not necessarily the transition probabilities in a Markov sense. `xttrans` counts transitions from each observation to the next once the observations have been put in t order within i. It does not normalize for missing periods. `xttrans` does pay attention to missing values of the variable being tabulated, however, and does not count transitions from nonmissing to missing or from missing to nonmissing. Thus if the data are fully rectangularized, `xttrans` produces (inefficient) estimates of the Markov transition matrix. `fillin` will rectangularize datasets; see [D] **fillin**. Thus the Markov transition matrix could be estimated by typing

```
. fillin idcode year
. xttrans msp
```
(*output omitted*)

❑

Stored results

`xttab` stores the following in `r()`:

Scalars

`r(n_panels)`	number of panels

Matrices

`r(results)`	results matrix

`xttrans` stores the following in `r()`:

Scalars

`r(n_trans)`	number of transitions
`r(n_rows)`	number of initial values (rows in output)
`r(n_cols)`	number of final values (columns in output)

Also see

[XT] **xtdescribe** — Describe pattern of xt data

[XT] **xtsum** — Summarize xt data

Description

`xttobit` fits a random-effects tobit model for panel data where the outcome variable is censored. Censoring limits may be fixed for all observations or vary across observations. The user can request that a likelihood-ratio test comparing the panel tobit model with the pooled tobit model be conducted at estimation time.

Quick start

Tobit model of `y` on `x` where `y` is censored at a lower limit of 5 using `xtset` data

```
xttobit y x, ll(5)
```

Add indicators for levels of categorical variable `a`

```
xttobit y x i.a, ll(5)
```

Same as above, but specify that censoring occurs at 5 and 25

```
xttobit y x i.a, ll(5) ul(25)
```

Same as above, but where `lower` and `upper` are variables containing the censoring limits

```
xttobit y x i.a, ll(lower) ul(upper)
```

Add likelihood-ratio test comparing the random-effects model with the pooled model

```
xttobit y x i.a, ll(lower) ul(upper) tobit
```

Menu

Statistics > Longitudinal/panel data > Censored outcomes > Tobit regression (RE)

Syntax

`xttobit` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, *options*]

options	Description
Model	
`noconstant`	suppress constant term
`ll`[(*varname* | #)]	left-censoring variable or limit
`ul`[(*varname* | #)]	right-censoring variable or limit
`offset(`*varname*`)`	include *varname* in model with coefficient constrained to 1
`constraints(`*constraints*`)`	apply specified linear constraints
SE	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`tobit`	perform likelihood-ratio test comparing against pooled tobit model
`lrmodel`	perform the likelihood-ratio model test instead of the default Wald test
`nocnsreport`	do not display constraints
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Integration	
`intmethod(`*intmethod*`)`	integration method; *intmethod* may be `mvaghermite` (the default) or `ghermite`
`intpoints(#)`	use # quadrature points; default is `intpoints(12)`
Maximization	
maximize_options	control the maximization process; seldom used
`collinear`	keep collinear variables
`coeflegend`	display legend instead of statistics

A panel variable must be specified; use `xtset`; see [XT] **xtset**.

indepvars may contain factor variables; see **[U] 11.4.3 Factor variables**.

depvar and *indepvars* may contain time-series operators; see **[U] 11.4.4 Time-series varlists**.

`by`, `collect`, `fp`, and `statsby` are allowed; see **[U] 11.1.10 Prefix commands**.

`iweight`s are allowed; see **[U] 11.1.6 weight**. Weights must be constant within panel.

`collinear` and `coeflegend` do not appear in the dialog box.

See **[U] 20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

noconstant; see [R] **Estimation options**.

ll[(*varname* | #)] and ul[(*varname* | #)] indicate the lower and upper limits for censoring, respectively. Observations with *depvar* ≤ ll() are left-censored; observations with *depvar* ≥ ul() are right-censored; and remaining observations are not censored. You do not have to specify the censoring values. If you specify ll, the lower limit is the minimum of *depvar*. If you specify ul, the upper limit is the maximum of *depvar*.

offset(*varname*), constraints(*constraints*); see [R] **Estimation options**.

SE

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (oim) and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] ***vce_options***.

Reporting

level(#); see [R] **Estimation options**.

tobit specifies that a likelihood-ratio test comparing the random-effects model with the pooled (tobit) model be included in the output.

lrmodel, nocnsreport; see [R] **Estimation options**.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Integration

intmethod(*intmethod*), intpoints(#); see [R] **Estimation options**.

Maximization

maximize_options: difficult, technique(*algorithm_spec*), iterate(#), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), nonrtolerance, and from(*init_specs*); see [R] **Maximize**. These options are seldom used.

The following options are available with xttobit but are not shown in the dialog box:

collinear, coeflegend; see [R] **Estimation options**.

Remarks and examples

xttobit fits a random-effects tobit model. There is no command for a fixed-effects model, because there does not exist a sufficient statistic allowing the fixed effects to be conditioned out of the likelihood.

Consider the linear regression model with panel-level random effects

$$y_{it} = \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i + \epsilon_{it}$$

for $i = 1, \dots, n$ panels, where $t = 1, \dots, n_i$. The random effects, ν_i, are i.i.d., $N(0, \sigma_\nu^2)$, and ϵ_{it} are i.i.d. $N(0, \sigma_\epsilon^2)$ independently of ν_i.

The observed data, y_{it}^o, represent possibly censored versions of y_{it}. If they are left-censored, all that is known is that $y_{it} \le y_{it}^o$. If they are right-censored, all that is known is that $y_{it} \ge y_{it}^o$. If they are uncensored, $y_{it} = y_{it}^o$. If they are left-censored, y_{it}^o is determined by `ll()`. If they are right-censored, y_{it}^o is determined by `ul()`. If they are uncensored, y_{it}^o is determined by *depvar*.

▷ Example 1: Random-effects tobit regression

Using the `nlswork` data described in [XT] **xt**, we fit a random-effects tobit model of adjusted (log) wages. We use the `ul()` option to impose an upper limit on the recorded log of wages.

```
. use https://www.stata-press.com/data/r19/nlswork3
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)
. xttobit ln_wage i.union age grade not_smsa south##c.year, ul(1.9) tobit
  (output omitted)
Random-effects tobit regression                     Number of obs     =   19,224
                                                        Uncensored =   12,334
Limits: Lower = -inf                                 Left-censored =        0
        Upper = 1.90                                Right-censored =    6,890

Group variable: idcode                              Number of groups  =    4,148
Random effects u_i ~ Gaussian                       Obs per group:
                                                                  min =        1
                                                                  avg =      4.6
                                                                  max =       12

Integration method: mvaghermite                     Integration pts.  =       12

                                                    Wald chi2(7)      = 2925.68
Log likelihood = -6814.4606                         Prob > chi2       =  0.0000

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
     1.union |   .1430527   .0069718    20.52   0.000     .1293883    .1567171
         age |   .0099132   .0017516     5.66   0.000     .0064801    .0133464
       grade |   .0784855   .0022764    34.48   0.000     .0740239    .0829472
    not_smsa |  -.1339978    .009206   -14.56   0.000    -.1520413   -.1159544
     1.south |  -.3507188   .0695554    -5.04   0.000    -.4870449   -.2143928
        year |  -.0008285   .0018371    -0.45   0.652    -.0044292    .0027721
             |
south#c.year |
           1 |   .0031938   .0008606     3.71   0.000     .0015071    .0048805
             |
       _cons |   .5101956   .1006646     5.07   0.000     .3128966    .7074946
-------------+----------------------------------------------------------------
    /sigma_u |   .3045992   .0048344    63.01   0.000     .2951239    .3140745
    /sigma_e |   .2488678   .0018254   136.34   0.000       .24529    .2524455
-------------+----------------------------------------------------------------
         rho |   .5996844   .0084095                       .583118    .6160734
------------------------------------------------------------------------------
LR test of sigma_u=0: chibar2(01) = 6650.63             Prob >= chibar2 = 0.000
```

The results from a tobit regression can be interpreted as we would those from a linear regression. Because the dependent variable is log transformed, the coefficients can be interpreted in terms of a percentage change. We see, for example, that on average, union members make 14.3% more than nonunion members.

The output also includes the overall and panel-level variance components (labeled `sigma_e` and `sigma_u`, respectively) together with ρ (labeled `rho`)

$$\rho = \frac{\sigma_\nu^2}{\sigma_\epsilon^2 + \sigma_\nu^2}$$

which is the percent contribution to the total variance of the panel-level variance component.

When `rho` is zero, the panel-level variance component is unimportant, and the panel estimator is not different from the pooled estimator. A likelihood-ratio test of this is included at the bottom of the output. This test formally compares the pooled estimator (tobit) with the panel estimator. In this case, we reject the null hypothesis that there are no panel-level effects.

◁

❏ Technical note

The random-effects model is calculated using quadrature, which is an approximation whose accuracy depends partially on the number of integration points used. We can use the `quadchk` command to see if changing the number of integration points affects the results. If the results change, the quadrature approximation is not accurate given the number of integration points. Try increasing the number of integration points using the `intpoints()` option and run `quadchk` again. Do not attempt to interpret the results of estimates when the coefficients reported by `quadchk` differ substantially. See [XT] **quadchk** for details and [XT] **xtprobit** for an example.

Because the `xttobit` likelihood function is calculated by Gauss–Hermite quadrature, on large problems the computations can be slow. Computation time is roughly proportional to the number of points used for the quadrature.

❏

Stored results

`xttobit` stores the following in `e()`:

Scalars

`e(N)`	number of observations
`e(N_g)`	number of groups
`e(N_unc)`	number of uncensored observations
`e(N_lc)`	number of left-censored observations
`e(N_rc)`	number of right-censored observations
`e(N_cd)`	number of completely determined observations
`e(k)`	number of parameters
`e(k_eq)`	number of equations in `e(b)`
`e(k_eq_model)`	number of equations in overall model test
`e(k_dv)`	number of dependent variables
`e(df_m)`	model degrees of freedom
`e(ll)`	log likelihood
`e(ll_0)`	log likelihood, constant-only model
`e(ll_c)`	log likelihood, comparison model
`e(chi2)`	χ^2
`e(chi2_c)`	χ^2 for comparison test

`e(rho)`	ρ
`e(sigma_u)`	panel-level standard deviation
`e(sigma_e)`	standard deviation of ϵ_{it}
`e(n_quad)`	number of quadrature points
`e(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(g_max)`	largest group size
`e(p)`	p-value for model test
`e(rank)`	rank of `e(V)`
`e(rank0)`	rank of `e(V)` for constant-only model
`e(ic)`	number of iterations
`e(rc)`	return code
`e(converged)`	`1` if converged, `0` otherwise

Macros

`e(cmd)`	`xttobit`
`e(cmdline)`	command as typed
`e(depvar)`	names of dependent variables
`e(ivar)`	variable denoting groups
`e(llopt)`	contents of `ll()`
`e(ulopt)`	contents of `ul()`
`e(k_aux)`	number of auxiliary parameters
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(title)`	title in estimation output
`e(offset1)`	offset
`e(chi2type)`	`Wald` or `LR`; type of model χ^2 test
`e(chi2_ct)`	`Wald` or `LR`; type of model χ^2 test corresponding to `e(chi2_c)`
`e(vce)`	*vcetype* specified in `vce()`
`e(intmethod)`	integration method
`e(distrib)`	`Gaussian`; the distribution of the random effect
`e(opt)`	type of optimization
`e(which)`	`max` or `min`; whether optimizer is to perform maximization or minimization
`e(ml_method)`	type of `ml` method
`e(user)`	name of likelihood-evaluator program
`e(technique)`	maximization technique
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(marginsok)`	predictions allowed by `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`

Matrices

`e(b)`	coefficient vector
`e(Cns)`	constraints matrix
`e(ilog)`	iteration log
`e(gradient)`	gradient vector
`e(V)`	variance–covariance matrix of the estimators

Functions

`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

Assuming a normal distribution, $N(0, \sigma_\nu^2)$, for the random effects ν_i, we have the joint (unconditional of ν_i) density of the observed data from the ith panel

$$f(y_{i1}^o, \ldots, y_{in_i}^o | \mathbf{x}_{i1}, \ldots, \mathbf{x}_{in_i}) = \int_{-\infty}^{\infty} \frac{e^{-\nu_i^2/2\sigma_\nu^2}}{\sqrt{2\pi}\sigma_\nu} \left\{ \prod_{t=1}^{n_i} F(y_{it}^o, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i) \right\} d\nu_i$$

where

$$F(y_{it}^o, \Delta_{it}) = \begin{cases} \left(\sqrt{2\pi}\sigma_\epsilon\right)^{-1} e^{-(y_{it}^o - \Delta_{it})^2/(2\sigma_\epsilon^2)} & \text{if } y_{it}^o \in C \\ \Phi\left(\frac{y_{it}^o - \Delta_{it}}{\sigma_\epsilon}\right) & \text{if } y_{it}^o \in L \\ 1 - \Phi\left(\frac{y_{it}^o - \Delta_{it}}{\sigma_\epsilon}\right) & \text{if } y_{it}^o \in R \end{cases}$$

where C is the set of noncensored observations, L is the set of left-censored observations, R is the set of right-censored observations, and $\Phi()$ is the cumulative normal distribution.

The panel level likelihood l_i is given by

$$\begin{aligned} l_i &= \int_{-\infty}^{\infty} \frac{e^{-\nu_i^2/2\sigma_\nu^2}}{\sqrt{2\pi}\sigma_\nu} \left\{ \prod_{t=1}^{n_i} F(y_{it}^o, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i) \right\} d\nu_i \\ &\equiv \int_{-\infty}^{\infty} g(y_{it}^o, x_{it}, \nu_i) d\nu_i \end{aligned}$$

This integral can be approximated with M-point Gauss–Hermite quadrature

$$\int_{-\infty}^{\infty} e^{-x^2} h(x) dx \approx \sum_{m=1}^{M} w_m^* h(a_m^*)$$

This is equivalent to

$$\int_{-\infty}^{\infty} f(x) dx \approx \sum_{m=1}^{M} w_m^* \exp\left\{(a_m^*)^2\right\} f(a_m^*)$$

where the w_m^* denote the quadrature weights and the a_m^* denote the quadrature abscissas. The log likelihood, L, is the sum of the logs of the panel level likelihoods l_i.

The default approximation of the log likelihood is by adaptive Gauss–Hermite quadrature, which approximates the panel level likelihood with

$$l_i \approx \sqrt{2}\widehat{\sigma}_i \sum_{m=1}^{M} w_m^* \exp\left\{(a_m^*)^2\right\} g(y_{it}^o, x_{it}, \sqrt{2}\widehat{\sigma}_i a_m^* + \widehat{\mu}_i)$$

where $\widehat{\sigma}_i$ and $\widehat{\mu}_i$ are the adaptive parameters for panel i. Therefore, with the definition of $g(y_{it}^o, x_{it}, \nu_i)$, the total log likelihood is approximated by

$$\begin{aligned} L \approx \sum_{i=1}^{n} w_i \log\Bigg[\sqrt{2}\widehat{\sigma}_i \sum_{m=1}^{M} w_m^* \exp\{(a_m^*)^2\} & \frac{\exp\{-(\sqrt{2}\widehat{\sigma}_i a_m^* + \widehat{\mu}_i)^2/2\sigma_\nu^2\}}{\sqrt{2\pi}\sigma_\nu} \\ & \prod_{t=1}^{n_i} F(y_{it}^o, x_{it}\boldsymbol{\beta} + \sqrt{2}\widehat{\sigma}_i a_m^* + \widehat{\mu}_i) \Bigg] \end{aligned} \tag{1}$$

where w_i is the user-specified weight for panel i; if no weights are specified, $w_i = 1$.

The default method of adaptive Gauss–Hermite quadrature is to calculate the posterior mean and variance and use those parameters for $\widehat{\mu}_i$ and $\widehat{\sigma}_i$ by following the method of Naylor and Smith (1982), further discussed in Skrondal and Rabe-Hesketh (2004). We start with $\widehat{\sigma}_{i,0} = 1$ and $\widehat{\mu}_{i,0} = 0$, and the posterior means and variances are updated in the kth iteration. That is, at the kth iteration of the optimization for l_i we use

$$l_{i,k} \approx \sum_{m=1}^{M} \sqrt{2}\widehat{\sigma}_{i,k-1} w_m^* \exp\{a_m^*)^2\} g(y_{it}^o, x_{it}, \sqrt{2}\widehat{\sigma}_{i,k-1} a_m^* + \widehat{\mu}_{i,k-1})$$

Letting

$$\tau_{i,m,k-1} = \sqrt{2}\widehat{\sigma}_{i,k-1} a_m^* + \widehat{\mu}_{i,k-1}$$

$$\widehat{\mu}_{i,k} = \sum_{m=1}^{M} (\tau_{i,m,k-1}) \frac{\sqrt{2}\widehat{\sigma}_{i,k-1} w_m^* \exp\{(a_m^*)^2\} g(y_{it}^o, x_{it}, \tau_{i,m,k-1})}{l_{i,k}}$$

and

$$\widehat{\sigma}_{i,k} = \sum_{m=1}^{M} (\tau_{i,m,k-1})^2 \frac{\sqrt{2}\widehat{\sigma}_{i,k-1} w_m^* \exp\{(a_m^*)^2\} g(y_{it}^o, x_{it}, \tau_{i,m,k-1})}{l_{i,k}} - (\widehat{\mu}_{i,k})^2$$

and this is repeated until $\widehat{\mu}_{i,k}$ and $\widehat{\sigma}_{i,k}$ have converged for this iteration of the maximization algorithm. This adaptation is applied on every iteration until the log-likelihood change from the preceding iteration is less than a relative difference of 1e–6; after this, the quadrature parameters are fixed.

The log likelihood can also be calculated by nonadaptive Gauss–Hermite quadrature if the `intmethod(ghermite)` option is specified. For nonadaptive Gauss–Hermite quadrature, the following formula for the log likelihood is used in place of (1).

$$\begin{aligned} L &= \sum_{i=1}^{n} w_i \log\Big\{ \Pr(y_{i1}, \ldots, y_{in_i} | \mathbf{x}_{i1}, \ldots, \mathbf{x}_{in_i}) \Big\} \\ &\approx \sum_{i=1}^{n} w_i \log \left[\frac{1}{\sqrt{\pi}} \sum_{m=1}^{M} w_m^* \prod_{t=1}^{n_i} F\left\{ y_{it}^o, \mathbf{x}_{it}\boldsymbol{\beta} + \sqrt{2}\sigma_\nu a_m^* \right\} \right] \end{aligned}$$

Both quadrature formulas require that the integrated function be well approximated by a polynomial of degree equal to the number of quadrature points. Panel size can affect whether

$$\prod_{t=1}^{n_i} F(y_{it}^o, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i)$$

is well approximated by a polynomial. As panel size and ρ increase, the quadrature approximation can become less accurate. For large ρ, the random-effects model can also become unidentified. Adaptive quadrature gives better results for correlated data and large panels than nonadaptive quadrature; however, we recommend that you use the `quadchk` command (see [XT] **quadchk**) to verify the quadrature approximation used in this command, whichever approximation you choose.

References

Naylor, J. C., and A. F. M. Smith. 1982. Applications of a method for the efficient computation of posterior distributions. *Journal of the Royal Statistical Society*, C ser., 31: 214–225. https://doi.org/10.2307/2347995.

Pendergast, J. F., S. J. Gange, M. A. Newton, M. J. Lindstrom, M. Palta, and M. R. Fisher. 1996. A survey of methods for analyzing clustered binary response data. *International Statistical Review* 64: 89–118. https://doi.org/10.2307/1403425.

Skrondal, A., and S. Rabe-Hesketh. 2004. *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models*. Boca Raton, FL: Chapman and Hall/CRC.

Also see

[XT] **xttobit postestimation** — Postestimation tools for xttobit

[XT] **quadchk** — Check sensitivity of quadrature approximation

[XT] **xteintreg** — Extended random-effects interval regression

[XT] **xtintreg** — Random-effects interval-data regression model

[XT] **xtreg** — Linear models for panel data

[XT] **xtset** — Declare data to be panel data

[ME] **metobit** — Multilevel mixed-effects tobit regression

[R] **tobit** — Tobit regression

[U] 20 Estimation and postestimation commands

Postestimation commands   predict   margins   Remarks and examples
Methods and formulas   Also see

Postestimation commands

The following postestimation commands are available after `xttobit`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of parameters
`estat ic`	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC, respectively)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
`forecast`	dynamic forecasts and simulations
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
`lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	predictions and their SEs, etc.
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of parameters
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

predict

Description for predict

`predict` creates a new variable containing predictions such as linear predictions, standard errors, probabilities, and expected values.

Menu for predict

Statistics > Postestimation

Syntax for predict

`predict` [*type*] *newvar* [*if*] [*in*] [, *statistic* `nooffset`]

statistic	Description
Main	
`xb`	linear prediction; the default
`stdp`	standard error of the linear prediction
`stdf`	standard error of the linear forecast
`pr(`*a*`,`*b*`)`	$\Pr(a < y < b)$, marginal with respect to the random effect
`e(`*a*`,`*b*`)`	$E(y \mid a < y < b)$, marginal with respect to the random effect
`ystar(`*a*`,`*b*`)`	$E(y^*)$, $y^* = \max\{a, \min(y, b)\}$, marginal with respect to the random effect

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

where *a* and *b* may be numbers or variables; *a* missing ($a \geq .$) means $-\infty$, and *b* missing ($b \geq .$) means $+\infty$; see [U] **12.2.1 Missing values**.

Options for predict

Main

`xb`, the default, calculates the linear prediction $\mathbf{x}_{it}\boldsymbol{\beta}$ using the estimated fixed effects (coefficients) in the model. This is equivalent to fixing all random effects in the model to their theoretical (prior) mean value of zero.

`stdp` calculates the standard error of the linear prediction. It can be thought of as the standard error of the predicted expected value or mean for the observation's covariate pattern. The standard error of the prediction is also referred to as the standard error of the fitted value.

`stdf` calculates the standard error of the linear forecast. This is the standard error of the point prediction for 1 observation. It is commonly referred to as the standard error of the future or forecast value. By construction, the standard errors produced by `stdf` are always larger than those produced by `stdp`; see *Methods and formulas* in [R] **regress**.

`pr(`a`,`b`)` calculates estimates of $\Pr(a < y < b \mid \mathbf{x} = \mathbf{x}_{it})$, which is the probability that y would be observed in the interval (a, b), given the current values of the predictors, $\mathbf{x}_{it}$. The predictions are calculated marginally with respect to the random effect. That is, the random effect is integrated out of the prediction function. In the discussion that follows, these two conditions are implied.

a and *b* may be specified as numbers or variable names; *lb* and *ub* are variable names;
`pr(20,30)` calculates $\Pr(20 < y < 30)$;
`pr(`*lb*`,`*ub*`)` calculates $\Pr(lb < y < ub)$; and
`pr(20,`*ub*`)` calculates $\Pr(20 < y < ub)$.

a missing ($a \geq .$) means $-\infty$; `pr(.,30)` calculates $\Pr(-\infty < y < 30)$;
`pr(`*lb*`,30)` calculates $\Pr(-\infty < y < 30)$ in observations for which $lb \geq .$
(and calculates $\Pr(lb < y < 30)$ elsewhere).

b missing ($b \geq .$) means $+\infty$; `pr(20,.)` calculates $\Pr(+\infty > y > 20)$;
`pr(20,`*ub*`)` calculates $\Pr(+\infty > y > 20)$ in observations for which $ub \geq .$
(and calculates $\Pr(20 < y < ub)$ elsewhere).

`e(`a`,`b`)` calculates estimates of $E(y \mid a < y < b, \mathbf{x} = \mathbf{x}_{it})$, which is the expected value of y conditional on y being in the interval (a, b), meaning that y is truncated. *a* and *b* are specified as they are for `pr()`. The predictions are calculated marginally with respect to the random effect. That is, the random effect is integrated out of the prediction function.

`ystar(`a`,`b`)` calculates estimates of $E(y^* \mid \mathbf{x} = \mathbf{x}_{it})$, where $y^* = a$ if $y \leq a$, $y^* = b$ if $y \geq b$, and $y^* = y$ otherwise, meaning that y^* is the censored version of y. *a* and *b* are specified as they are for `pr()`. The predictions are calculated marginally with respect to the random effect. That is, the random effect is integrated out of the prediction function.

`nooffset` is relevant only if you specify `offset(`*varname*`)` for `xttobit`. It modifies the calculations made by `predict` so that they ignore the offset variable; the linear prediction is treated as $\mathbf{x}_{it}\boldsymbol{\beta}$ rather than $\mathbf{x}_{it}\boldsymbol{\beta} + \text{offset}_{it}$.

margins

Description for margins

`margins` estimates margins of response for linear predictions, probabilities, and expected values.

Menu for margins

Statistics > Postestimation

Syntax for margins

`margins` [*marginlist*] [, *options*]

`margins` [*marginlist*] , `predict(`*statistic* ...`)` [`predict(`*statistic* ...`)` ...] [*options*]

statistic	Description
`xb`	linear prediction, the default
`pr(`*a*,*b*`)`	$\Pr(a < y < b)$, marginal with respect to the random effect
`e(`*a*,*b*`)`	$E(y \mid a < y < b)$, marginal with respect to the random effect
`ystar(`*a*,*b*`)`	$E(y^*)$, $y^* = \max\{a, \min(y, b)\}$, marginal with respect to the random effect
`stdp`	not allowed with `margins`
`stdf`	not allowed with `margins`

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.

For the full syntax, see [R] **margins**.

Remarks and examples

▷ Example 1: Average marginal probabilities at specified covariate values

In example 1 of [XT] **xttobit**, we fit a random-effects model of wages. Say that we want to know how union membership status affects the probability that a worker's wage will be "low", where low means a log wage that is less than the 20th percentile of all observations in our dataset. First, we use `centile` to find the 20th percentile of `ln_wage`:

```
. use https://www.stata-press.com/data/r19/nlswork3
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)
. xttobit ln_wage i.union age grade not_smsa south##c.year, ul(1.9)
  (output omitted)
. centile ln_wage, centile(20)

                                                      Binom. interp.
    Variable |     Obs  Percentile    Centile      [95% conf. interval]
-------------+----------------------------------------------------------
     ln_wage |  28,534          20   1.301507       1.297063    1.308635
```

Now, we use `margins` to obtain the effect of union status on the probability that the log of wages is in the bottom 20% of women. Given the results from `centile`, that corresponds to the log of wages being below 1.30. We evaluate the effect for two groups: 1) women age 30 living in the south in 1988 who graduated from high school but had no more schooling and 2) the same group of women who instead graduated from college (`grade=16`).

```
. margins, dydx(union) predict(pr(.,1.30))
> at(age=30 south=1 year=88 grade=12 union=0)
> at(age=30 south=1 year=88 grade=16 union=0)

Average marginal effects                               Number of obs = 19,224
Model VCE: OIM

Expression: Pr(ln_wage<1.30), predict(pr(.,1.30))
dy/dx wrt:  1.union
1._at: union =  0
       age   = 30
       grade = 12
       south =  1
       year  = 88
2._at: union =  0
       age   = 30
       grade = 16
       south =  1
       year  = 88

─────────────┬────────────────────────────────────────────────────────────────
             │            Delta-method
             │      dy/dx   std. err.      z    P>|z|     [95% conf. interval]
─────────────┼────────────────────────────────────────────────────────────────
0.union      │  (base outcome)
─────────────┼────────────────────────────────────────────────────────────────
1.union      │
         _at │
          1  │  -.0992088   .0057424   -17.28   0.000    -.1104637   -.0879539
          2  │  -.0374347   .0033407   -11.21   0.000    -.0439823   -.0308871
─────────────┴────────────────────────────────────────────────────────────────
Note: dy/dx for factor levels is the discrete change from the base level.
```

For the first group of women, according to our fitted model, being in a union lowers the probability of being classified as a low-wage worker by almost 10 percentage points. Being a college graduate attenuates this effect to just above 3.7 percentage points.

◁

Methods and formulas

The following uses the notation introduced in *Remarks and examples* of [XT] **xttobit**.

The marginal probability that y_{it} is observed in the interval $(\ell\ell_{it}, u\ell_{it})$, obtained by specifying the `pr(`a`,`b`)` option, is calculated as

$$\begin{aligned} \mathrm{pr}(\ell\ell_{it}, u\ell_{it}) &= \mathrm{Pr}(\ell\ell_{it} < \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i + \epsilon_{it} < u\ell_{it}) \\ &= \Phi\left(\frac{u\ell_{it} - \mathbf{x}_{it}\widehat{\boldsymbol{\beta}}}{\widehat{\sigma}}\right) - \Phi\left(\frac{\ell\ell_{it} - \mathbf{x}_{it}\widehat{\boldsymbol{\beta}}}{\widehat{\sigma}}\right) \end{aligned} \tag{1}$$

where $\widehat{\sigma}$ is the square root of the estimated marginal variance of the linear predictor, $\sqrt{\widehat{\sigma}_\epsilon^2 + \widehat{\sigma}_\nu^2}$.

The `e(`a`,`b`)` option computes the expected value of y_{it} conditional on y_{it} being in the interval $(\ell\ell_{it}, u\ell_{it})$, that is, when y_{it} is truncated. The expected value is calculated as

$$\begin{aligned} e(\ell\ell_{it}, u\ell_{it}) &= E(\mathbf{x}_{it}\boldsymbol{\beta} + \nu_i + \epsilon_{it} \mid \ell\ell_{it} < \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i + \epsilon_{it} < u\ell_{it}) \\ &= \mathbf{x}_{it}\widehat{\boldsymbol{\beta}} - \widehat{\sigma}\frac{\phi\left(\frac{u\ell_{it}-\mathbf{x}_{it}\widehat{\boldsymbol{\beta}}}{\widehat{\sigma}}\right) - \phi\left(\frac{\ell\ell_{it}-\mathbf{x}_{it}\widehat{\boldsymbol{\beta}}}{\widehat{\sigma}}\right)}{\Phi\left(\frac{u\ell_{it}-\mathbf{x}_{it}\widehat{\boldsymbol{\beta}}}{\widehat{\sigma}}\right) - \Phi\left(\frac{\ell\ell_{it}-\mathbf{x}_{it}\widehat{\boldsymbol{\beta}}}{\widehat{\sigma}}\right)} \end{aligned} \tag{2}$$

where ϕ is the normal density and Φ is the cumulative normal distribution.

You can also compute the expected value of y_{it}, where y_{it} is assumed censored at $\ell\ell_{it}$ and $u\ell_{it}$ by specifying the option `ystar(`a, b`)`. This expected value is

$$y^*_{it} = \begin{cases} \ell\ell_{it} & \text{if } y_{it} \le \ell\ell_{it} \\ \mathbf{x}_{it}\boldsymbol{\beta} + \epsilon_{it} & \text{if } \ell\ell_{it} < y_{it} < u\ell_{it} \\ u\ell_{it} & \text{if } y_{it} \ge u\ell_{it} \end{cases}$$

This computation can be expressed in several ways, but the most intuitive formulation involves a combination of (1) and (2):

$$E(y^*_{it}) = \text{pr}(-\infty, \ell\ell_{it})\ell\ell_{it} + \text{pr}(\ell\ell_{it}, u\ell_{it})e(\ell\ell_{it}, u\ell_{it}) + \text{pr}(u\ell_{it}, +\infty)u\ell_{it}$$

Also see

[XT] **xttobit** — Random-effects tobit model

[U] 20 Estimation and postestimation commands

xtunitroot — Panel-data unit-root tests

Description | Quick start | Menu | Syntax
Options | Remarks and examples | Stored results | Methods and formulas
Acknowledgments | References | Also see

Description

xtunitroot performs a variety of tests for unit roots (or stationarity) in panel datasets. The Levin–Lin–Chu (2002), Harris–Tzavalis (1999), Breitung (2000; Breitung and Das 2005), Im–Pesaran–Shin (2003), and Fisher-type (Choi 2001) tests have as the null hypothesis that all the panels contain a unit root. The Hadri (2000) Lagrange multiplier (LM) test has as the null hypothesis that all the panels are (trend) stationary. The top of the output for each test makes explicit the null and alternative hypotheses. Options allow you to include panel-specific means (fixed effects) and time trends in the model of the data-generating process.

Quick start

Levin–Lin–Chu test that each series y within panels contains a unit root using xtset data

```
xtunitroot llc y
```

Same as above, but specify 4 lags for the augmented Dickey–Fuller regressions

```
xtunitroot llc y, lags(4)
```

Harris–Tzavalis unit-root test including a time trend

```
xtunitroot ht y, trend
```

Breitung unit-root test with 4 lags to prewhiten the series

```
xtunitroot breitung y, lags(4)
```

Im–Pesaran–Shin unit-root test for the demeaned series y

```
xtunitroot ips y, demean
```

Philips–Perron unit-root test of y with 1 lag for prewhitening

```
xtunitroot fisher y, pperron lags(1)
```

Hadri Lagrange multiplier stationarity test using Bartlett's kernel with 1 lag to estimate long-run variance

```
xtunitroot hadri y, kernel(bartlett)
```

Menu

Statistics > Longitudinal/panel data > Unit-root tests

Syntax

Levin–Lin–Chu test

`xtunitroot llc` *varname* [*if*] [*in*] [, *LLC_options*]

Harris–Tzavalis test

`xtunitroot ht` *varname* [*if*] [*in*] [, *HT_options*]

Breitung test

`xtunitroot breitung` *varname* [*if*] [*in*] [, *Breitung_options*]

Im–Pesaran–Shin test

`xtunitroot ips` *varname* [*if*] [*in*] [, *IPS_options*]

Fisher-type tests (combining p-values)

`xtunitroot fisher` *varname* [*if*] [*in*] , `{dfuller | pperron} lags(#)`
[*Fisher_options*]

Hadri Lagrange multiplier stationarity test

`xtunitroot hadri` *varname* [*if*] [*in*] [, *Hadri_options*]

LLC_options	Description
`trend`	include a time trend
`noconstant`	suppress panel-specific means
`demean`	subtract cross-sectional means
`lags(`*lag_spec*`)`	specify lag structure for augmented Dickey–Fuller (ADF) regressions
`kernel(`*kernel_spec*`)`	specify method to estimate long-run variance

lag_spec is either a nonnegative integer or one of `aic`, `bic`, or `hqic` followed by a positive integer.
kernel_spec takes the form *kernel maxlags*, where *kernel* is one of `bartlett`, `parzen`, or `quadraticspectral` and *maxlags* is either a positive number or one of `nwest` or `llc`.

HT_options	Description
`trend`	include a time trend
`noconstant`	suppress panel-specific means
`demean`	subtract cross-sectional means
`altt`	make small-sample adjustment to T

Breitung_options	Description
`trend`	include a time trend
`noconstant`	suppress panel-specific means
`demean`	subtract cross-sectional means
`robust`	allow for cross-sectional dependence
`lags(#)`	specify lag structure for prewhitening

IPS_options	Description
trend	include a time trend
demean	subtract cross-sectional means
lags(*lag_spec*)	specify lag structure for ADF regressions

lag_spec is either a nonnegative integer or one of aic, bic, or hqic followed by a positive integer.

Fisher_options	Description
* dfuller	use ADF unit-root tests
* pperron	use Phillips–Perron unit-root tests
* lags(#)	specify lag structure for prewhitening
demean	subtract cross-sectional means
dfuller_opts	any options allowed by the dfuller command
pperron_opts	any options allowed by the pperron command

*Either dfuller or pperron is required.
*lags(#) is required.

Hadri_options	Description
trend	include a time trend
demean	subtract cross-sectional means
robust	allow for cross-sectional dependence
kernel(*kernel_spec*)	specify method to estimate long-run variance

kernel_spec takes the form *kernel* [#], where *kernel* is one of bartlett, parzen, or quadraticspectral and # is a positive number.

varname may contain time-series operators; see **[U] 11.4.4 Time-series varlists**.
collect is allowed with all xtunitroot tests; see **[U] 11.1.10 Prefix commands**.

Options

LLC_options

trend includes a linear time trend in the model that describes the process by which the series is generated.

noconstant suppresses the panel-specific mean term in the model that describes the process by which the series is generated. Specifying noconstant imposes the assumption that the series has a mean of zero for all panels.

demean requests that xtunitroot first subtract the cross-sectional averages from the series. When specified, for each time period xtunitroot computes the mean of the series across panels and subtracts this mean from the series. Levin, Lin, and Chu suggest this procedure to mitigate the impact of cross-sectional dependence.

lags(*lag_spec*) specifies the lag structure to use for the ADF regressions performed in computing the test statistic.

Specifying lags(#) requests that # lags of the series be used in the ADF regressions. The default is lags(1).

Specifying `lags(aic #)` requests that the number of lags of the series be chosen such that the Akaike information criterion (AIC) for the regression is minimized. `xtunitroot llc` will fit ADF regressions with 1 to # lags and choose the regression for which the AIC is minimized. This process is done for each panel so that different panels may use ADF regressions with different numbers of lags.

Specifying `lags(bic #)` is just like specifying `lags(aic #)`, except that the Bayesian information criterion (BIC) is used instead of the AIC.

Specifying `lags(hqic #)` is just like specifying `lags(aic #)`, except that the Hannan–Quinn information criterion is used instead of the AIC.

`kernel(`*kernel_spec*`)` specifies the method used to estimate the long-run variance of each panel's series. *kernel_spec* takes the form *kernel maxlags*. *kernel* is one of `bartlett`, `parzen`, or `quadraticspectral`. *maxlags* is a number, `nwest` to request the Newey and West (1994) bandwidth selection algorithm, or `llc` to request the lag truncation algorithm in Levin, Lin, and Chu (2002).

Specifying, for example, `kernel(bartlett 3)` requests the Bartlett kernel with 3 lags.

Specifying `kernel(bartlett nwest)` requests the Bartlett kernel with the maximum number of lags determined by the Newey and West bandwidth selection algorithm.

Specifying `kernel(bartlett llc)` requests the Bartlett kernel with a maximum lag determined by the method proposed in Levin, Lin, and Chu's (2002) article:

$$\mathit{maxlags} = \mathrm{int}\left(3.21T^{1/3}\right)$$

where T is the number of observations per panel. This is the default.

HT_options

`trend` includes a linear time trend in the model that describes the process by which the series is generated.

`noconstant` suppresses the panel-specific mean term in the model that describes the process by which the series is generated. Specifying `noconstant` imposes the assumption that the series has a mean of zero for all panels.

`demean` requests that `xtunitroot` first subtract the cross-sectional averages from the series. When specified, for each time period `xtunitroot` computes the mean of the series across panels and subtracts this mean from the series. Levin, Lin, and Chu suggest this procedure to mitigate the impact of cross-sectional dependence.

`altt` requests that `xtunitroot` use $T-1$ instead of T in the formulas for the mean and variance of the test statistic under the null hypothesis. When the number of time periods, T, is small (less than 10 or 15), the test suffers from severe size distortions when fixed effects or time trends are included; in these cases, using `altt` results in much improved size properties at the expense of significantly less power.

Breitung_options

`trend` includes a linear time trend in the model that describes the process by which the series is generated.

`noconstant` suppresses the panel-specific mean term in the model that describes the process by which the series is generated. Specifying `noconstant` imposes the assumption that the series has a mean of zero for all panels.

demean requests that xtunitroot first subtract the cross-sectional averages from the series. When specified, for each time period xtunitroot computes the mean of the series across panels and subtracts this mean from the series. Levin, Lin, and Chu suggest this procedure to mitigate the impact of cross-sectional dependence.

robust requests a variant of the test that is robust to cross-sectional dependence.

lags(#) specifies the number of lags used to remove higher-order autoregressive components of the series. The Breitung test assumes the data are generated by an AR(1) process; for higher-order processes, the first-differenced and lagged-level data are replaced by the residuals from regressions of those two series on the first # lags of the first-differenced data. The default is to not perform this prewhitening step.

IPS_options

trend includes a linear time trend in the model that describes the process by which the series is generated.

demean requests that xtunitroot first subtract the cross-sectional averages from the series. When specified, for each time period xtunitroot computes the mean of the series across panels and subtracts this mean from the series. Levin, Lin, and Chu suggest this procedure to mitigate the impact of cross-sectional dependence.

lags(*lag_spec*) specifies the lag structure to use for the ADF regressions performed in computing the test statistic. With this option, xtunitroot reports Im, Pesaran, and Shin's (2003) $W_{t\text{-}bar}$ statistic that is predicated on T going to infinity first, followed by N going to infinity. By default, no lags are included, and xtunitroot instead reports Im, Pesaran, and Shin's $\tilde{t}\text{-}bar$ and $Z_{\tilde{t}\text{-}bar}$ statistics that assume T is fixed while N goes to infinity, as well as the $t\text{-}bar$ statistic and exact critical values that assume both N and T are fixed.

Specifying lags(#) requests that # lags of the series be used in the ADF regressions. By default, no lags are included.

Specifying lags(aic #) requests that the number of lags of the series be chosen such that the AIC for the regression is minimized. xtunitroot llc will fit ADF regressions with 1 to # lags and choose the regression for which the AIC is minimized. This process is done for each panel so that different panels may use ADF regressions with different numbers of lags.

Specifying lags(bic #) is just like specifying lags(aic #), except that the BIC is used instead of the AIC.

Specifying lags(hqic #) is just like specifying lags(aic #), except that the Hannan–Quinn information criterion is used instead of the AIC.

If you specify lags(0), then xtunitroot reports the $W_{t\text{-}bar}$ statistic instead of the $Z_{t\text{-}bar}$, $Z_{\tilde{t}\text{-}bar}$, and $t\text{-}bar$ statistics.

Fisher_options

dfuller requests that xtunitroot conduct ADF unit-root tests on each panel by using the dfuller command. You must specify either the dfuller or the pperron option.

pperron requests that xtunitroot conduct Phillips–Perron unit-root tests on each panel by using the pperron command. You must specify either the pperron or the dfuller option.

`lags(#)` specifies the number of lags used to remove higher-order autoregressive components of the series. The Fisher test assumes the data are generated by an AR(1) process; for higher-order processes, the first-differenced and lagged-level data are replaced by the residuals from regressions of those two series on the first # lags of the first-differenced data. `lags(#)` is required.

`demean` requests that `xtunitroot` first subtract the cross-sectional averages from the series. When specified, for each time period `xtunitroot` computes the mean of the series across panels and subtracts this mean from the series. Levin, Lin, and Chu suggest this procedure to mitigate the impact of cross-sectional dependence.

dfuller_opts are any options accepted by the `dfuller` command, including `noconstant`, `trend`, `drift`, and `lags()`. Because `xtunitroot` calls `dfuller quietly`, the `dfuller` option `regress` has no effect. See [TS] **dfuller**.

pperron_opts are any options accepted by the `pperron` command, including `noconstant`, `trend`, and `lags()`. Because `xtunitroot` calls `pperron quietly`, the `pperron` option `regress` has no effect. See [TS] **pperron**.

Hadri_options

`trend` includes a linear time trend in the model that describes the process by which the series is generated.

`demean` requests that `xtunitroot` first subtract the cross-sectional averages from the series. When specified, for each time period `xtunitroot` computes the mean of the series across panels and subtracts this mean from the series. Levin, Lin, and Chu suggest this procedure to mitigate the impact of cross-sectional dependence.

`robust` requests a variant of the test statistic that is robust to heteroskedasticity across panels.

`kernel(`*kernel_spec*`)` requests a variant of the test statistic that is robust to serially correlated errors. *kernel_spec* specifies the method used to estimate the long-run variance of each panel's series. *kernel_spec* takes the form *kernel* [#]. Three kernels are supported: `bartlett`, `parzen`, and `quadraticspectral`.

Specifying, for example, `kernel(bartlett 3)` requests the Bartlett kernel with 3 lags.

If # is not specified, then 1 lag is used.

Remarks and examples

Remarks are presented under the following headings:

Overview

We consider a simple panel-data model with a first-order autoregressive component:

$$y_{it} = \rho_i y_{i,t-1} + \mathbf{z}'_{it}\boldsymbol{\gamma}_i + \epsilon_{it} \tag{1}$$

where $i = 1, \ldots, N$ indexes panels; $t = 1, \ldots, T_i$ indexes time; y_{it} is the variable being tested; and ϵ_{it} is a stationary error term. The $\mathbf{z}_{it}$ term can represent panel-specific means, panel-specific means and a time trend, or nothing, depending on the options specified to `xtunitroot`. By default, $\mathbf{z}_{it} = 1$, so that the term $\mathbf{z}'_{it}\boldsymbol{\gamma}_i$ represents panel-specific means (fixed effects). If `trend` is specified, $\mathbf{z}'_{it} = (1, t)$ so that $\mathbf{z}'_{it}\boldsymbol{\gamma}_i$ represents panel-specific means and linear time trends. For tests that allow it, specifying `noconstant` omits the $\mathbf{z}'_{it}\boldsymbol{\gamma}_i$ term. The Im–Pesaran–Shin (`xtunitroot ips`) and Fisher-type (`xtunitroot fisher`) tests allow unbalanced panels, while the remaining tests require balanced panels so that $T_i = T$ for all i.

Panel unit-root tests are used to test the null hypothesis $H_0 : \rho_i = 1$ for all i versus the alternative $H_a : \rho_i < 1$. Depending on the test, H_a may hold, for one i, a fraction of all i or all i; the output of the respective test precisely states the alternative hypothesis. Equation (1) is often written as

$$\Delta y_{it} = \phi_i y_{i,t-1} + \mathbf{z}'_{it}\boldsymbol{\gamma}_i + \epsilon_{it} \tag{1'}$$

so that the null hypothesis is then $H_0 : \phi_i = 0$ for all i versus the alternative $H_a : \phi_i < 0$.

The Hadri LM test for panel stationarity instead assumes the null hypothesis that all panels are stationary versus the alternative that at least some of the panels contain unit roots. We discuss the Hadri LM test in detail later, though for now our remarks focus on tests whose null hypothesis is that the panels contain unit roots.

The various panel unit-root tests implemented by `xtunitroot` differ in several key aspects. First, the Levin–Lin–Chu (`xtunitroot llc`), Harris–Tsavalis (`xtunitroot ht`), and Breitung (`xtunitroot breitung`) tests make the simplifying assumption that all panels share the same autoregressive parameter so that $\rho_i = \rho$ for all i. The other tests implemented by `xtunitroot`, however, allow the autoregressive parameter to be panel specific. Maddala and Wu (1999) provide an example of testing whether countries' economic growth rates converge to a long-run value. Imposing the restriction that $\rho_i = \rho$ for all i implies that the rate of convergence would be the same for all countries, an implication that is too restrictive in practice.

Second, the various tests make differing assumptions about the rates at which the number of panels, N, and the number of time periods, T, tend to infinity or whether N or T is fixed. For microeconomic panels of firms, for example, increasing the sample size would involve gathering data on more firms while holding the number of time periods fixed; here N tends to infinity whereas T is fixed. In a macroeconomic analysis of OECD countries, one would typically assume that N is fixed whereas T tends to infinity.

Related to the previous point, the size of one's sample will in large part determine which test is most appropriate in a given situation. If a dataset has a small number of panels and a large number of time periods, then a panel unit-root test that assumes that N is fixed or that N tends to infinity at a slower rate than T will likely perform better than one that is designed for cases where N is large.

Hlouskova and Wagner (2006) provide a good overview of the types of panel unit-root tests available with `xtunitroot`, and they present exhaustive Monte Carlo simulations examining the tests' performance. Baltagi (2013, chap. 12) also concisely discusses the tests implemented by `xtunitroot`.

The following table summarizes some of the key differences among the various tests:

Test	Options	Asymptotics	ρ under H_a	Panels
LLC	`noconstant`	$\sqrt{N}/T \to 0$	common	balanced
LLC		$N/T \to 0$	common	balanced
LLC	`trend`	$N/T \to 0$	common	balanced
HT	`noconstant`	$N \to \infty$, T fixed	common	balanced
HT		$N \to \infty$, T fixed	common	balanced
HT	`trend`	$N \to \infty$, T fixed	common	balanced
Breitung	`noconstant`	$(T, N) \to_{\text{seq}} \infty$	common	balanced
Breitung		$(T, N) \to_{\text{seq}} \infty$	common	balanced
Breitung	`trend`	$(T, N) \to_{\text{seq}} \infty$	common	balanced
IPS		$N \to \infty$, T fixed or N and T fixed	panel-specific	unbalanced
IPS	`trend`	$N \to \infty$, T fixed or N and T fixed	panel-specific	unbalanced
IPS	`lags()`	$(T, N) \to_{\text{seq}} \infty$	panel-specific	unbalanced
IPS	`trend lags()`	$(T, N) \to_{\text{seq}} \infty$	panel-specific	unbalanced
Fisher-type		$T \to \infty$, N finite or infinite	panel-specific	unbalanced
Hadri LM		$(T, N) \to_{\text{seq}} \infty$	(not applicable)	balanced
Hadri LM	`trend`	$(T, N) \to_{\text{seq}} \infty$	(not applicable)	balanced

The first column identifies the test procedure, where we use LLC to denote the Levin–Lin–Chu test, HT to denote the Harris–Tsavalis test, and IPS to denote the Im–Pesaran–Shin test. The second column indicates the deterministic components included in (1) or (1′). The column labeled "Asymptotics" indicates the behavior of the number of panels, N, and time periods, T, required for the test statistic to have a well-defined asymptotic distribution. For example, the LLC test without the `noconstant` option requires that T grow at a faster rate than N so that N/T approaches zero; with the `noconstant` option, we need only for T to grow faster than the square root of N (so T could grow more slowly than N).

The HT tests and the IPS tests without accommodations for serial correlation assume that the number of time periods, T, is fixed, whereas N tends to infinity; `xtunitroot` also reports critical values for the IPS tests that are valid in finite samples (where N and T are fixed).

Many of the tests are justified using sequential limit theory, which we denote as $(T, N) \to_{\text{seq}} \infty$. First, the time dimension goes to infinity, and then the number of panels goes to infinity. As a practical matter, these tests work best with "large" T and at least "moderate" N. See Phillips and Moon (2000) for an introduction to asymptotics that depend on both N and T and their relation to nonstationary panels. Phillips and Moon (1999) contains a more technical discussion of "multi-indexed" asymptotics.

The fourth column refers to the parameter ρ_i in (1) and ϕ_i in (1′). As we mentioned previously, some tests assume that all panels have the same autoregressive parameter under the alternative hypothesis of stationarity (denoted "common" in the table), while others allow for panel-specific autoregressive parameters (denoted "panel-specific" in the table). The Hadri LM tests are not framed in terms of an equation like (1) or (1′), so the distinction based on ρ is not applicable.

The final column indicates whether the panel dataset must be strongly balanced, meaning each panel has the same number of observations covering the same time span. Except for the Fisher tests, all the tests require that there be no gaps in any panel's series.

We now discuss each test in turn.

Levin–Lin–Chu test

The starting point for the Levin–Lin–Chu (LLC) test is (1′) with the restriction that all panels share a common autoregressive parameter. In a regression model like (1), ϵ_{it} is likely to be plagued by serial correlation, so to mitigate this problem, LLC augment the model with additional lags of the dependent variable:

$$\Delta y_{it} = \phi y_{i,t-1} + \mathbf{z}'_{it}\boldsymbol{\gamma}_i + \sum_{j=1}^{p} \theta_{ij}\Delta y_{i,t-j} + u_{it} \tag{2}$$

The number of lags, p, can be specified using the `lags()` option, or you can have `xtunitroot llc` select the number of lags that minimizes one of several information criteria. The LLC test assumes that ϵ_{it} is independently distributed across panels and follows a stationary invertible autoregressive moving-average process for each panel. By including sufficient lags of $\Delta y_{i,t}$ in (2), u_{it} will be white noise; the test does not require u_{it} to have the same variance across panels.

Under the null hypothesis of a unit root, y_{it} is nonstationary, so a standard OLS regression t statistic for ϕ will have a nonstandard distribution that depends in part on the specification of the $\mathbf{z}_{it}$ term. Moreover, the inclusion of a fixed-effect term in a dynamic model like (2) causes the OLS estimate of ϕ to be biased toward zero; see Nickell (1981). The LLC method produces a bias-adjusted t statistic, which the authors denote as t^*_δ, that has an asymptotically normal distribution.

The LLC test without panel-specific intercepts or time trends, requested by specifying the `noconstant` option with `xtunitroot llc`, is justified asymptotically if $\sqrt{N}/T \rightarrow 0$, allowing the time dimension T to grow more slowly than the cross-sectional dimension N; LLC (2002) mention that this assumption is particularly relevant for panel datasets typically encountered in microeconomic applications.

If model (2) includes panel-specific means (the default for `xtunitroot llc`) or time trends (requested with the `trend` option), then you must assume that $N/T \rightarrow 0$ for the t^*_δ statistic to have an asymptotically standard normal distribution. This implies that the time dimension, T, must grow faster than the cross-sectional dimension, N, a situation more plausible with macroeconomic datasets.

LLC (2002) recommend using their test with panels of "moderate" size, which they describe as having between 10 and 250 panels and 25 to 250 observations per panel. Baltagi (2013, sec. 12.2.3) mentions that the requirement $N/T \rightarrow 0$ implies that N should be small relative to T.

❏ Technical note

Panel unit-root tests have frequently been used to test the purchasing power parity (PPP) hypothesis. We use a PPP dataset to illustrate the `xtunitroot` command, but understanding PPP is not required to understand how these tests are applied. Here we outline PPP and explain how to test it using panel unit-root tests; uninterested readers can skip the remainder of this technical note. Our discussion and examples are motivated by those in Oh (1996) and Patterson (2000, chap. 13). Also see Rogoff (1996) for a broader introduction to PPP.

The PPP hypothesis is based on the Law of One Price, which stipulates that the price of a tradable good will be the same everywhere. Absolute PPP stipulates that the nominal exchange rate, E, is

$$E = \frac{P}{P^*}$$

where P is the price of a basket of goods in the home country and P^* is the price of the same basket in the foreign country. The exchange rate, E, indicates the price of a foreign currency in terms of our "home" currency or, equivalently, how many units of the home currency are needed to buy one unit of the foreign currency.

Now consider the real exchange rate, λ, which tells us the prices of goods and services—things we actually consume—in a foreign country relative to their prices at home. We have

$$\lambda = \frac{EP^*}{P} \tag{3}$$

λ in general does not equal unity for many reasons, including the fact that not all goods are tradable across countries (haircuts being the textbook example), trade barriers such as tariffs and quotas, differences among countries in how price indices are constructed, and the Harrod–Balassa–Samuelson effect, which links productivity and price levels; see Obstfeld and Rogoff (1996, 210–216).

Taking logs of both sides of (3), we have

$$y \equiv \ln\lambda = \ln E + \ln P^* - \ln P$$

PPP holds only if the real exchange rate reverts to its equilibrium value over time. Thus, to test for PPP, we test whether y contains a unit root. If y does contain a unit root, we reject PPP.

The dataset `pennxrate.dta` contains real exchange-rate data based on the Penn World Table version 6.2 (Heston, Summers, and Aten 2006). The data are a balanced panel consisting of 151 countries observed over 34 years, from 1970 through 2003. The United States was treated as the domestic country and is therefore not included. The variable `lnrxrate` contains the log of the real exchange rate and is the variable on which we conduct panel unit-root tests in the examples.

Two indicator variables are included in the dataset as well. The variable `oecd` flags 27 countries aside from the United States that are members of the Organisation for Economic Co-operation and Development (OECD). (The Czech Republic and the Slovak Republic are excluded because they did not become independent countries until 1993.) The variable `g7` flags the six countries aside from the United States that are members of the Group of Seven (G7) nations.

❑

▷ Example 1

The dataset `pennxrate.dta` contains real exchange-rate data for a panel of countries observed over 34 years. Here we use the LLC test to determine whether the series `lnrxrate`, the log of real exchange rates, contains a unit root for six nations that are currently in the G7 group of advanced economies. We do not have any reason to believe `lnrxrate` should exhibit a global trend, so we do not include the `trend` option.

Looking at (2), we have no a priori knowledge of the number of lags, p, needed to ensure that u_{it} is white noise, so we let `xtunitroot` choose the number of lags for each panel by minimizing the AIC, subject to a maximum of 10 lags.

We type

```
. use https://www.stata-press.com/data/r19/pennxrate
. xtunitroot llc lnrxrate if g7, lags(aic 10)
Levin-Lin-Chu unit-root test for lnrxrate
─────────────────────────────────────────

H0: Panels contain unit roots               Number of panels  =      6
Ha: Panels are stationary                   Number of periods =     34

AR parameter: Common                        Asymptotics: N/T -> 0
Panel means:  Included
Time trend:   Not included

ADF regressions: 1.00 lags average (chosen by AIC)
LR variance:     Bartlett kernel, 10.00 lags average (chosen by LLC)
──────────────────────────────────────────────────────────────────────────────
                    Statistic      p-value
──────────────────────────────────────────────────────────────────────────────
 Unadjusted t        -6.7538
 Adjusted t*         -4.0277        0.0000
──────────────────────────────────────────────────────────────────────────────
```

The header of the output summarizes the exact specification of the test and dataset. Because we did not specify the `noconstant` option, the test allowed for panel-specific means. On average, $p = 1$ lag of the dependent variable of (2) were included as regressors in the ADF regressions. By default, `xtunitroot` estimated the long-run variance of Δ`lnrxrate`$_{it}$ by using a Bartlett kernel with an average of 10 lags.

The LLC bias-adjusted test statistic $t^*_\delta = -4.0277$ is significantly less than zero ($p < 0.00005$), so we reject the null hypothesis of a unit-root [that is, that $\phi = 0$ in (2)] in favor of the alternative that `lnrxrate` is stationary (that is, that $\phi < 0$). This conclusion supports the PPP hypothesis.

Labeled "Unadjusted t" in the output is a conventional t statistic for testing $H_0: \phi = 0$. When the model does not include panel-specific means or trends, this test statistic has a standard normal limiting distribution and its p-value is shown in the output; the unadjusted statistic, t_δ, diverges to negative infinity if trends or panel-specific constants are included, so a p-value is not displayed in those cases.

Because the G7 economies have many similarities, our results could be affected by cross-sectional correlation in real exchange rates; O'Connell's (1998) results showed that the LLC test exhibits severe size distortions in the presence of cross-sectional correlation. LLC (2002) suggested removing cross-sectional averages from the data to help control for this correlation. We can do this by specifying the `demean` option to `xtunitroot`:

```
. xtunitroot llc lnrxrate if g7, lags(aic 10) demean
Levin-Lin-Chu unit-root test for lnrxrate
-----------------------------------------
H0: Panels contain unit roots                  Number of panels  =      6
Ha: Panels are stationary                      Number of periods =     34

AR parameter: Common                           Asymptotics: N/T -> 0
Panel means:  Included
Time trend:   Not included                     Cross-sectional means removed

ADF regressions: 1.50 lags average (chosen by AIC)
LR variance:     Bartlett kernel, 10.00 lags average (chosen by LLC)
------------------------------------------------------------------------------
                    Statistic      p-value
------------------------------------------------------------------------------
 Unadjusted t        -5.5473
 Adjusted t*         -2.0813        0.0187
------------------------------------------------------------------------------
```

Once we control for cross-sectional correlation by removing cross-sectional means, we can no longer reject the null hypothesis of a unit root at the 1% significance level, though we can reject at the 5% level. ◁

Here we chose the number of lags based on the AIC criterion in an admission that we do not know the true number of lags to include in (2). However, the test statistics are derived under the assumption that the lag order, p, is known. If we happen to choose the wrong number of lags, then the distribution of the test statistic will depart from its expected distribution that assumes p is known.

Harris–Tsavalis test

In many datasets, particularly in microeconomics, the time dimension, T, is small, so tests whose asymptotic properties are established by assuming that T tends to infinity can lead to incorrect inference. HT (1999) derived a unit-root test that assumes that the time dimension, T, is fixed. Their simulation results suggest that the test has favorable size and power properties for N greater than 25, and they report (p. 213) that power improves faster as T increases for a given N than when N increases for a given T.

The HT test statistic is based on the OLS estimator, ρ, in the regression model

$$y_{it} = \rho y_{i,t-1} + \mathbf{z}'_{it}\boldsymbol{\gamma}_i + \epsilon_{it} \tag{4}$$

where the term $\mathbf{z}'_{it}\boldsymbol{\gamma}_i$ allows for panel-specific means and trends and was discussed in *Overview*. Harris and Tsavalis assume that ϵ_{it} is independent and identically distributed (i.i.d.) normal with constant variance across panels. Because of the bias induced by the inclusion of the panel means and time trends in this model, the expected value of the OLS estimator is not equal to unity under the null hypothesis. Harris and Tsavalis derived the mean and standard error of $\widehat{\rho}$ for (4) under the null hypothesis $H_0: \rho = 1$ when neither panel-specific means nor time trends are included (requested with the `noconstant` option), when only panel-specific means are included (the default), and when both panel-specific means and time trends are included (requested with the `trend` option). The asymptotic distribution of the test statistic is justified as $N \to \infty$, so you should have a relatively large number of panels when using this test. Notice that, like the LLC test, the HT test assumes that all panels share the same autoregressive parameter.

▷ Example 2

Because the HT test is designed for cases where N is relatively large, here we test whether the series `lnrxrate` contains a unit root using all 151 countries in our dataset. We will again remove cross-sectional means to help control for contemporaneous correlation. We type

```
. xtunitroot ht lnrxrate, demean
Harris-Tzavalis unit-root test for lnrxrate
-------------------------------------------
H0: Panels contain unit roots                Number of panels  =    151
Ha: Panels are stationary                    Number of periods =     34

AR parameter: Common                         Asymptotics: N -> Infinity
Panel means:  Included                                    T Fixed
Time trend:   Not included                   Cross-sectional means removed
------------------------------------------------------------------------------
                    Statistic           z         p-value
------------------------------------------------------------------------------
 rho                  0.8184      -13.1239         0.0000
------------------------------------------------------------------------------
```

Here we strongly reject the null hypothesis of a unit root, again finding support for PPP. The point estimate of ρ in (4) is 0.8184, and the z statistic is -13.12.

◁

Can we directly compare the results from the LLC and HT tests? We used a subset of the data for the LLC test but used all the data for the HT test. That leads to the obvious answer that no, our results are not entirely comparable. However, a more subtle issue regarding the asymptotic properties of the tests also warrants caution when comparing results.

The LLC test assumes that $N/T \rightarrow 0$, so N should be small relative to T. Moreover, with our exchange-rate dataset, we are much more likely to be able to add more years of data rather than add more countries, because the number of countries in the world is for the most part fixed. Hence, assuming T grows faster than N is certainly plausible.

On the other hand, the HT test assumes that T is fixed whereas N goes to infinity. Is that assumption plausible for our dataset? As we just mentioned, T likely grows faster than N here, so using a test that assumes T is fixed whereas N grows is hard to justify with our dataset.

In short, when selecting a panel unit-root test, you must consider the relative sizes of N and T and the relative speeds at which they tend to infinity or whether either N or T is fixed.

Breitung test

Both the LLC and HT tests take the approach of first fitting a regression model and subsequently adjusting the autoregressive parameter or its t statistic to compensate for the bias induced by having a dynamic regressor and fixed effects in the model. The Breitung (2000; Breitung and Das 2005) test takes a different tact, adjusting the data before fitting a regression model so that bias adjustments are not needed.

In the LLC test, additional lags of the dependent variable could be included in (2) to control for serial correlation. The Breitung procedure instead allows for a prewhitening of the series before computing the test. If the `trend` option is not specified, we regress Δy_{it} and $y_{i,t-1}$ on $\Delta y_{i,t-1}, \ldots, \Delta y_{i,t-p}$ and use the residuals from those regressions in place of $\Delta y_{i,t}$ and $y_{i,t-1}$ in computing the test. You specify the number of lags, p, to use by specifying `lags(#)`. If the `trend` option is specified, then the Breitung method uses a different prewhitening procedure that involves fitting only one (instead of two) preliminary regressions; see *Methods and formulas* for details.

Monte Carlo simulations by Breitung (2000) show that bias-corrected statistics such as LLC's t^*_δ suffer from low power, particularly against alternative hypotheses with autoregressive parameters near one and when panel-specific effects are included. In contrast, the Breitung (2000) test statistic exhibits much higher power in these cases. Moreover, the Breitung test has good power even with small datasets ($N = 25$, $T = 25$), though the power of the test appears to deteriorate when T is fixed and N is increased.

The Breitung test assumes that the error term ϵ_{it} is uncorrelated across both i and t. `xtunitroot breitung` optionally also reports a version of the statistic based on Breitung and Das (2005) that is robust to cross-sectional correlation.

▷ Example 3

Here we test whether `lnrxrate` contains a unit root for the subset of 27 OECD countries in our dataset. We will use the `robust` option to obtain a test statistic that is robust to cross-sectional correlation, so we will not subtract the cross-sectional means via the `demean` option. We type

```
. xtunitroot breitung lnrxrate if oecd, robust
Breitung unit-root test for lnrxrate

H0: Panels contain unit roots                  Number of panels  =     27
Ha: Panels are stationary                      Number of periods =     34

AR parameter: Common                           Asymptotics:   T,N -> Infinity
Panel means:  Included                                            sequentially
Time trend:   Not included                     Prewhitening: Not performed

                    Statistic      p-value

 lambda*             -1.6794        0.0465

 * Lambda robust to cross-sectional correlation
```

We can reject the null of a unit root at the 5% level but not at the 1% level.

◁

Im–Pesaran–Shin test

All the tests we have discussed thus far assume that all panels share a common autoregressive parameter, ρ. Cultural, institutional, and other factors make such an assumption tenuous for both macro- and microeconometric panel datasets. IPS (2003) developed a set of tests that relax the assumption of a common autoregressive parameter. Moreover, the IPS test does not require balanced datasets, though there cannot be gaps within a panel. The starting point for the IPS test is a set of Dickey–Fuller regressions of the form

$$\Delta y_{it} = \phi_i y_{i,t-1} + \mathbf{z}'_{it}\boldsymbol{\gamma}_i + \epsilon_{it} \tag{5}$$

Notice that here ϕ is panel-specific, indexed by i, whereas in (2), ϕ is constant. Im, Pesaran, and Shin assume that ϵ_{it} is independently distributed normal for all i and t, and they allow ϵ_{it} to have heterogeneous variances σ_i^2 across panels.

As described by Maddala and Wu (1999), one way to view the key difference between the IPS and LLC tests is that here we fit (5) to each panel separately and average the resulting t statistics, whereas in the LLC test we pool the data before fitting an equation such as (2) (thus we impose a common autoregressive parameter) and compute a test statistic based on the pooled regression results.

Under the null hypothesis that all panels contain a unit root, we have $\phi_i = 0$ for all i. The alternative is that the fraction of panels that follow stationary processes is nonzero; that is, as N tends to infinity, the fraction N_1/N converges to a nonzero value, where N_1 is the number of panels that are stationary.

Whether you allow for serially correlated errors determines the test statistics produced, and because there are substantive differences in the output, we consider the serially uncorrelated and serially correlated cases separately. First, we consider the serially uncorrelated case, which `xtunitroot` assumes when you do not specify the `lags()` option.

The IPS test allowing for heterogeneous panels with serially uncorrelated errors assumes that the number of time periods, T, is fixed; `xtunitroot ips` produces statistics both for the case where N is fixed and for the case where $N \to \infty$. Under the null hypothesis of a unit root, the usual t statistic, t_i, for testing $H_0 : \phi_i = 0$ in (5) does not have a mean of zero. For the case where N is fixed, IPS used simulation to tabulate "exact" critical values for the average of the t_i statistics when the dataset is balanced; these critical values are not available with unbalanced datasets. The critical values are "exact" only when the error term is normally distributed and when T corresponds to one of the sample sizes used in their simulation studies. For other values of T, `xtunitroot ips` linearly interpolates the values in IPS (2003, table 2).

For the case where $N \to \infty$, they used simulation to tabulate the mean and variance of t_i for various values of T under the null hypothesis and showed that a bias-adjusted average of the t_i's has a standard normal limiting distribution. We illustrate the test with an example.

▷ Example 4

Here we test whether `lnrxrate` contains a unit root for the subset of OECD countries. We type

```
. xtunitroot ips lnrxrate if oecd, demean
Im-Pesaran-Shin unit-root test for lnrxrate
-------------------------------------------
H0: All panels contain unit roots           Number of panels  =     27
Ha: Some panels are stationary              Number of periods =     34

AR parameter: Panel-specific                Asymptotics: T,N -> Infinity
Panel means:  Included                                    sequentially
Time trend:   Not included                  Cross-sectional means removed

ADF regressions: No lags included
------------------------------------------------------------------------------
                                              Fixed-N exact critical values
                  Statistic      p-value          1%        5%       10%
------------------------------------------------------------------------------
 t-bar             -3.1327                     -1.810    -1.730    -1.680
 t-tilde-bar       -2.5771
 Z-t-tilde-bar     -7.3911       0.0000
------------------------------------------------------------------------------
```

As with the other unit-root tests available with `xtunitroot`, the header of the output contains a summary of the dataset's dimensions and the null and alternative hypotheses. First, consider the statistic labeled `t-bar`, which IPS denote as $t\text{-}bar_{NT}$. This statistic is appropriate when you assume that both N and T fixed; exact critical values reported in IPS (2003) are reported immediately to its right. Here, because $t\text{-}bar_{NT}$ is less than even its 1% critical value, we strongly reject the null hypothesis that all series contain a unit root in favor of the alternative that a nonzero fraction of the panels represent stationary processes.

The statistic labeled `t-tilde-bar` is IPS's $\tilde{t}\text{-}bar_{NT}$ statistic and is similar to the $t\text{-}bar_{NT}$ statistic, except that a different estimator of the Dickey–Fuller regression error variance is used. A standardized version of this statistic, $Z_{\tilde{t}\text{-}bar}$, is labeled `Z-t-tilde-bar` in the output and has an asymptotic standard normal distribution. Here the p-value corresponding to `Z-t-tilde-bar` is essentially zero, so we strongly reject the null that all series contain a unit root.

◁

❏ Technical note

Just as the $Z_{\tilde{t}\text{-}bar}$ statistic corresponds to $\tilde{t}\text{-}bar_{NT}$, IPS present a $Z_{t\text{-}bar}$ statistic corresponding to $t\text{-}bar_{NT}$. However, the $Z_{t\text{-}bar}$ statistic does not have an asymptotic normal distribution, and so it is not presented in the output. $Z_{t\text{-}bar}$ is available in the stored results as `r(zt)`.

❏

When serial correlation is present, we augment the Dickey–Fuller regression with further lags of the dependent variable:

$$\Delta y_{it} = \phi_i y_{i,t-1} + \mathbf{z}_{it}'\boldsymbol{\gamma}_i + \sum_{j=1}^{p} \Delta y_{i,t-j} + \epsilon_{it} \tag{6}$$

where the number of lags, p, is specified using the `lags()` option, and if the `trend` option is specified, we also include a time trend with panel-specific slope. You can either specify a number or have `xtunitroot` choose the number of lags for each panel by minimizing an information criterion. Here `xtunitroot` produces the IPS $W_{t\text{-}bar}$ statistic, which has an asymptotically standard normal distribution as $T \rightarrow \infty$ followed by $N \rightarrow \infty$. As a practical matter, this means you should have a reasonably large number of both time periods and panels to use this test.

Part of the computation of the $W_{t\text{-}bar}$ statistic involves retrieving expected values and variances of the t statistic for β_i in (6) in table 3 of IPS (2003). Because expected values have not been computed beyond $p = 8$ lags in (6), you cannot request more than 8 lags in the `lags()` option.

▷ Example 5

We again test whether `lnrxrate` contains a unit root for the subset of OECD countries, except we allow for serially correlated errors. We will choose the number of lags for the ADF regressions by minimizing the AIC criterion, subject to a maximum of 8 lags. We type

```
. xtunitroot ips lnrxrate if oecd, lags(aic 8) demean
Im-Pesaran-Shin unit-root test for lnrxrate
───────────────────────────────────────────
H0: All panels contain unit roots            Number of panels  =     27
Ha: Some panels are stationary               Number of periods =     34
AR parameter: Panel-specific                 Asymptotics: T,N -> Infinity
Panel means:  Included                                    sequentially
Time trend:   Not included                   Cross-sectional means removed
ADF regressions: 1.48 lags average (chosen by AIC)
──────────────────────────────────────────────────────────────────────────────
                     Statistic      p-value
──────────────────────────────────────────────────────────────────────────────
 W-t-bar              -7.3075       0.0000
──────────────────────────────────────────────────────────────────────────────
```

◁

Fisher-type tests

In our discussion of the IPS test, we intimated that the test statistics could be viewed as averages of bias-adjusted t statistics for each panel. As Maddala and Wu (1999, 635) describe the IPS test, "… the IPS test is a way of combining the evidence on the unit-root hypothesis from the N unit-root tests performed on the N cross-section units." Fisher-type panel unit-root tests make this approach explicit.

Meta-analysis, frequently used in biostatistics and medical sciences, is the combination of results from multiple studies designed to test a similar hypothesis to yield a more decisive conclusion. One type of meta-analysis, first proposed by R. A. Fisher, combines the p-values from independent tests to obtain an overall test statistic and is frequently called a Fisher-type test. See Whitehead (2002, sec. 9.8) for an introduction. In the context of panel data unit-root tests, we perform a unit-root test on each panel's series separately, then combine the p-values to obtain an overall test of whether the panel series contains a unit root.

`xtunitroot fisher` performs either ADF or Phillips–Perron unit-root tests on each panel depending on whether you specify the `dfuller` or `pperron` option. The actual tests are conducted by the `dfuller` and `pperron` commands, and you can specify to `xtunitroot fisher` any options those commands take; see [TS] **dfuller** and [TS] **pperron**.

`xtunitroot fisher` combines the p-values from the panel-specific unit-root tests using the four methods proposed by Choi (2001). Three of the methods differ in whether they use the inverse χ^2, inverse-normal, or inverse-logit transformation of p-values, and the fourth is a modification of the inverse χ^2 transformation that is suitable for when N tends to infinity. The inverse-normal and inverse-logit transformations can be used whether N is finite or infinite.

The null hypothesis being tested by `xtunitroot fisher` is that all panels contain a unit root. For a finite number of panels, the alternative is that at least one panel is stationary. As N tends to infinity, the number of panels that do not have a unit root should grow at the same rate as N under the alternative hypothesis.

▷ Example 6

Here we test for a unit root in `lnrxrate` using all 151 countries in our sample. We will use the ADF test. As before, we do not include a trend in real exchange rates and will therefore not specify the `trend` option. However, because the mean real exchange rate for any country is nonzero, we will specify the `drift` option. We will use two lags in the ADF regressions, and we will remove cross-sectional means by using `demean`. We type

```
. xtunitroot fisher lnrxrate, dfuller drift lags(2) demean
Fisher-type unit-root test for lnrxrate
Based on augmented Dickey-Fuller tests
---------------------------------------
H0: All panels contain unit roots           Number of panels  =    151
Ha: At least one panel is stationary        Number of periods =     34

AR parameter: Panel-specific                Asymptotics: T -> Infinity
Panel means:  Included
Time trend:   Not included                  Cross-sectional means removed
Drift term:   Included                      ADF regressions: 2 lags
------------------------------------------------------------------------------
                                  Statistic      p-value
------------------------------------------------------------------------------
 Inverse chi-squared(302)  P       975.9130       0.0000
 Inverse normal            Z       -19.6183       0.0000
 Inverse logit t(759)      L*      -20.9768       0.0000
 Modified inv. chi-squared Pm       27.4211       0.0000
------------------------------------------------------------------------------
 P statistic requires number of panels to be finite.
 Other statistics are suitable for finite or infinite number of panels.
------------------------------------------------------------------------------
```

All four of the tests strongly reject the null hypothesis that all the panels contain unit roots. Choi's (2001) simulation results suggest that the inverse normal Z statistic offers the best tradeoff between size and power, and he recommends using it in applications. We have observed that the inverse logit L^* test typically agrees with the Z test. Under the null hypothesis, Z has a standard normal distribution and L^* has a t distribution with $5N + 4$ degrees of freedom. Low values of Z and L^* cast doubt on the null hypothesis.

When the number of panels is finite, the inverse χ^2 P test is applicable; this statistic has a χ^2 distribution with $2N$ degrees of freedom, and large values are cause to reject the null hypothesis. Under the null hypothesis, as $T \rightarrow \infty$ followed by $N \rightarrow \infty$, P tends to infinity so that P has a degenerate limiting distribution. For large panels, Choi (2001) therefore proposes the modified inverse χ^2 P_m test which converges to a standard normal distribution; a large value of P_m casts doubt on the null hypothesis. Choi's simulation results do not reveal a specific value of N over which P_m should be preferred to P, though he mentions that $N = 100$ is still too small for P_m to have an approximately normal distribution.
◁

Hadri LM test

All the tests we have discussed so far take as the null hypothesis that the series contains a unit root. Classical statistical methods are designed to reject the null hypothesis only when the evidence against the null is sufficiently overwhelming. However, because unit-root tests typically are not very powerful against alternative hypotheses of somewhat persistent but stationary processes, reversing roles and testing the null hypothesis of stationarity against the alternative of a unit root is appealing. For pure time series, the KPSS test of Kwiatkowski et al. (1992) is one such test.

The Hadri (2000) LM test uses panel data to test the null hypothesis that the data are stationary versus the alternative that at least one panel contains a unit root. The test is designed for cases with large T and moderate N. The motivation for the test is straightforward. Suppose we include a panel-specific time trend (using the `trend` option with `xtunitroot hadri`) and write our series, y_{it}, as

$$y_{it} = r_{it} + \beta_i t + \epsilon_{it}$$

where r_{it} is a random walk,

$$r_{it} = r_{i,t-1} + u_{it}$$

and ϵ_{it} and u_{it} are zero-mean i.i.d. normal errors. If the variance of u_{it} were zero, then r_{it} would collapse to a constant; y_{it} would therefore be trend stationary. Using this logic, the Hadri LM test tests the hypothesis

$$H_0: \lambda = \frac{\sigma_u^2}{\sigma_\epsilon^2} = 0 \qquad \text{versus} \qquad H_a: \lambda > 0$$

Two options to `xtunitroot hadri` allow you to relax the assumption that ϵ_{it} is i.i.d., though normality is still required. You can specify the `robust` option to obtain a variant of the test that is robust to heteroskedasticity across panels, or you can specify `kernel()` to obtain a variant that is robust to serial correlation and heteroskedasticity. Asymptotically, the Hadri LM test is justified as $T \rightarrow \infty$ followed by $N \rightarrow \infty$. As a practical matter, Hadri (2000) recommends this test for "large" T and "moderate" N.

▷ Example 7

We now test the null hypothesis that `lnrxrate` is stationary for the subset of OECD countries. To control for serial correlation, we will use a Bartlett kernel with 5 lags. We type

```
. xtunitroot hadri lnrxrate if oecd, kernel(bartlett 5) demean
Hadri LM test for lnrxrate
-------------------------------------
H0: All panels are stationary                Number of panels  =     27
Ha: Some panels contain unit roots           Number of periods =     34

Time trend:          Not included            Asymptotics: T, N -> Infinity
Heteroskedasticity: Robust                                     sequentially
LR variance:         Bartlett kernel, 5 lags Cross-sectional means removed
------------------------------------------------------------------------------
                     Statistic      p-value
------------------------------------------------------------------------------
 z                     9.6473        0.0000
------------------------------------------------------------------------------
```

We strongly reject the null hypothesis that all panels' series are stationary in favor of the alternative that at least one of them contains a unit root. In contrast, the previous examples generally rejected the null hypothesis that all series contain unit roots in favor of the alternative that at least some are stationary. For cautionary remarks on the use of panel unit-root tests in the examination of PPP, see, for example, Banerjee, Marcellino, and Osbat (2005). In short, our results are qualitatively quite similar to those reported in the literature, though Banerjee, Marcellino, and Osbat argue that because of cross-unit cointegration and long-run relationships among countries, panel unit-root tests quite often reject the null hypothesis even when true.

◁

Stored results

`xtunitroot llc` stores the following in `r()`:

Scalars

`r(N)`	number of observations
`r(N_g)`	number of groups
`r(N_t)`	number of time periods
`r(sig_adj)`	standard deviation adjustment
`r(mu_adj)`	mean adjustment
`r(delta)`	pooled estimate of δ
`r(se_delta)`	pooled standard error of $\widehat{\delta}$
`r(Var_ep)`	variance of whitened differenced series

`r(sbar)`	mean of ratio of long-run to innovation standard deviations
`r(ttilde)`	observations per panel after lagging and differencing
`r(td)`	unadjusted t_δ statistic
`r(p_td)`	p-value for t_δ
`r(tds)`	adjusted t^*_δ statistic
`r(p_tds)`	p-value for t^*_δ
`r(hac_lags)`	lags used in HAC variance estimator
`r(hac_lagm)`	average lags used in HAC variance estimator
`r(adf_lags)`	lags used in ADF regressions
`r(adf_lagm)`	average lags used in ADF regressions
Macros	
`r(test)`	`llc`
`r(hac_kernel)`	kernel used in HAC variance estimator
`r(hac_method)`	HAC lag-selection algorithm
`r(adf_method)`	ADF regression lag-selection criterion
`r(demean)`	`demean`, if the data were demeaned
`r(deterministics)`	`noconstant`, `constant`, or `trend`

`xtunitroot ht` stores the following in `r()`:

Scalars	
`r(N)`	number of observations
`r(N_g)`	number of groups
`r(N_t)`	number of time periods
`r(rho)`	estimated ρ
`r(Var_rho)`	variance of ρ under H_0
`r(mean_rho)`	mean of ρ under H_0
`r(z)`	z statistic
`r(p)`	p-value
Macros	
`r(test)`	`ht`
`r(demean)`	`demean`, if the data were demeaned
`r(deterministics)`	`noconstant`, `constant`, or `trend`
`r(altt)`	`altt`, if `altt` was specified

`xtunitroot breitung` stores the following in `r()`:

Scalars	
`r(N)`	number of observations
`r(N_g)`	number of groups
`r(N_t)`	number of time periods
`r(lambda)`	test statistic λ
`r(lrobust)`	robust test statistic λ_R
`r(p)`	p-value for λ
`r(p_lrobust)`	p-value for λ_R
`r(lags)`	lags used for prewhitening
Macros	
`r(test)`	`breitung`
`r(demean)`	`demean`, if the data were demeaned
`r(robust)`	`robust`, if specified
`r(deterministics)`	`noconstant`, `constant`, or `trend`

xtunitroot ips stores the following in r():

Scalars	
r(N)	number of observations
r(N_g)	number of groups
r(N_t)	number of time periods
r(tbar)	test statistic $t\text{-}bar_{NT}$
r(cv_10)	exact 10% critical value for $t\text{-}bar_{NT}$
r(cv_5)	exact 5% critical value for $t\text{-}bar_{NT}$
r(cv_1)	exact 1% critical value for $t\text{-}bar_{NT}$
r(zt)	test statistic $Z_{t\text{-}bar}$
r(ttildebar)	test statistic $\tilde{t}\text{-}bar_{NT}$
r(zttildebar)	test statistic $Z_{\tilde{t}\text{-}bar}$
r(p_zttildebar)	p-value for $Z_{\tilde{t}\text{-}bar}$
r(wtbar)	test statistic $W_{t\text{-}bar}$
r(p_wtbar)	p-value for $W_{t\text{-}bar}$
r(lags)	lags used in ADF regressions
r(lagm)	average lags used in ADF regressions
Macros	
r(test)	ips
r(demean)	demean, if the data were demeaned
r(adf_method)	ADF regression lag-selection criterion
r(deterministics)	constant or trend

xtunitroot fisher stores the following in r():

Scalars	
r(N)	number of observations
r(N_g)	number of groups
r(N_t)	number of time periods
r(P)	inverse χ^2 P statistic
r(df_P)	P statistic degrees of freedom
r(p_P)	p-value for P statistic
r(L)	inverse logit L statistic
r(df_L)	L statistic degrees of freedom
r(p_L)	p-value for L statistic
r(Z)	inverse normal Z statistic
r(p_Z)	p-value for Z statistic
r(Pm)	modified inverse χ^2 P_m statistic
r(p_Pm)	p-value for P_m statistic
Macros	
r(test)	fisher
r(urtest)	dfuller or pperron
r(options)	options passed to dfuller or pperron
r(demean)	demean, if the data were demeaned

xtunitroot hadri stores the following in r():

Scalars	
r(N)	number of observations
r(N_g)	number of groups
r(N_t)	number of time periods
r(var)	variance of z under H_0
r(mu)	mean of z under H_0
r(z)	test statistic z
r(p)	p-value for z
r(lags)	lags used for HAC variance

Macros

`r(test)`	`hadri`
`r(demean)`	`demean`, if the data were demeaned
`r(robust)`	`robust`, if specified
`r(kernel)`	kernel used for HAC variance
`r(deterministics)`	`constant` or `trend`

Methods and formulas

Methods and formulas are presented under the following headings:

We consider a simple panel-data model with a first-order autoregressive component:

$$y_{it} = \rho_i y_{i,t-1} + \mathbf{z}'_{it}\boldsymbol{\gamma}_i + \epsilon_{it}$$

where $i = 1, \dots, N$ indexes panels and $t = 1, \dots, T$ indexes time. For the IPS, Fisher-type, and Hadri LM tests, we instead have $t = 1, \dots, T_i$, because they do not require balanced panels. ϵ_{it} is a zero-mean error term; we discuss the assumptions about ϵ_{it} for each test below. Here we use N to denote the number of panels, not the total number of observations. By default, $\mathbf{z}_{it} = 1$, so that the term $\mathbf{z}'_{it}\boldsymbol{\gamma}_i$ represents panel-specific means (fixed effects). If `noconstant` is specified, $\mathbf{z}'_{it}\boldsymbol{\gamma}_i$ vanishes. If `trend` is specified, $\mathbf{z}'_{it} = (1, t)$ so that $\mathbf{z}'_{it}\boldsymbol{\gamma}_i$ represents panel-specific means and linear time trends.

Levin–Lin–Chu test

The starting point for the LLC test is the regression model

$$\Delta y_{it} = \phi y_{i,t-1} + \mathbf{z}'_{it}\boldsymbol{\gamma}_i + \sum_{j=1}^{p_i} \theta_{ij}\Delta y_{i,t-j} + u_{it} \tag{7}$$

In (1′), LLC assume ϵ_{it} is independently distributed across panels and follows a stationary invertible process so that with sufficient lags of Δy_{it} included in (7), u_{it} will be white noise with potentially heterogeneous variance across panels. If `lags(#)` is specified with `xtunitroot llc`, then we set $p_i = \#$ for all panels $i = 1, \dots, N$. Otherwise, we fit (7) for each panel individually for lags $1 \dots p^{\max}$ and choose the lag length, p_i, that minimizes the information criterion requested by the user. During this step, we restrict estimation to the subset of observations that are valid when $p^{\max}$ lags are included. Information criteria are defined as follows:

$$\begin{aligned} \text{AIC} &= (-2\ln L + 2k)/M \\ \text{BIC} &= (-2\ln L + k\ln M)/M \\ \text{HQIC} &= (-2\ln L + 2k\ln\ln M)/M \end{aligned}$$

where $\ln L$ is the log likelihood assuming Gaussian errors, $M = T - p^{\max} - 2$, and k is the number of parameters in (7).

With the lag orders, p_i, in hand, the test proceeds in three main steps, the first of which is to use panel-by-panel OLS regressions to obtain the orthogonalized residuals

$$\hat{e}_{it} = \Delta y_{it} - \sum_{j=1}^{p_i} \hat{\theta}_{ij} \Delta y_{ij} - \mathbf{z}'_{it} \widehat{\gamma}_i \tag{8}$$

and

$$\hat{v}_{i,t-1} = y_{i,t-1} - \sum_{j=1}^{p_i} \tilde{\theta}_{ij} \Delta y_{ij} - \mathbf{z}_{it} \tilde{\gamma}_i \tag{9}$$

To control for panel-level heterogeneity, compute

$$\tilde{e}_{it} = \hat{e}_{it} / \hat{\sigma}_{\epsilon i} \qquad \text{and} \qquad \tilde{v}_{i,t-1} = \hat{v}_{i,t-1} / \hat{\sigma}_{\epsilon i}$$

where

$$\hat{\sigma}^2_{\epsilon i} = \frac{1}{T - p_i - 1} \sum_{t=p_i}^{T} \left(\hat{e}_{it} - \hat{\delta}_i \hat{v}_{i,t-1} \right)^2$$

and $\hat{\delta}_i$ is the OLS coefficient from a regression of $\hat{e}_{it}$ on $\hat{v}_{i,t-1}$. If time trends are included (by specifying the `trend` option), then a linear time trend is included in regressions (7), (8), and (9).

In the second step, we estimate the ratio of long-run to short-run variances. Under the null hypothesis of a unit root, the long-run variance of the model without panel-specific intercepts or time trends ($z_{it} = \{\emptyset\}$) can be estimated as

$$\hat{\sigma}^2_{yi} = \frac{1}{T-1} \sum_{t=2}^{T} \Delta y^2_{it} + \frac{2}{T-1} \sum_{j=1}^{m} K(j,m) \left(\sum_{t=j+2}^{T} \Delta y_{it} \Delta y_{i,t-j} \right)$$

where m is the maximum number of lags and $K(j,m)$ is the kernel weight function. Define $z = j/(m+1)$. If *kernel* is `bartlett`, then

$$K(j,m) = \begin{cases} 1 - z & 0 \le z \le 1 \\ 0 & \text{otherwise} \end{cases}$$

If *kernel* is `parzen`, then

$$K(j,m) = \begin{cases} 1 - 6z^2 + 6z^3 & 0 \le z \le 0.5 \\ 2(1-z)^3 & 0.5 < z \le 1 \\ 0 & \text{otherwise} \end{cases}$$

If *kernel* is `quadraticspectral`, then

$$K(j,m) = \begin{cases} 1 & z = 0 \\ 3\{\sin(\theta)/\theta - \cos(\theta)\}/\theta^2 & \text{otherwise} \end{cases}$$

where $\theta = 6\pi z/5$. If the user requests automatic bandwidth (lag) selection using the Newey–West algorithm, then we use the method documented in *Methods and formulas* of [R] **ivregress** with $\mathbf{z}_i = \mathbf{h} = 1$. If automatic lag selection with the LLC algorithm is chosen, then $m = \text{int}(3.21T^{1/3})$.

If panel-specific intercepts are included (by not specifying `noconstant`), then in the formula for $\hat{\sigma}^2_{yi}$ we replace Δy_{it} with $\Delta y_{it} - \overline{\Delta y_{it}}$, where $\overline{\Delta y_{it}}$ is the panel-level mean of Δy_{it} for panel i. Let $\hat{s}_i = \hat{\sigma}_{yi} / \hat{\sigma}_{\epsilon i}$, and denote $\widehat{S}_N = N^{-1} \sum_i \hat{s}_i$.

In the third step, we run the OLS regression

$$\tilde{e}_{it} = \delta \tilde{v}_{i,t-1} + \tilde{\epsilon}_{it}$$

Called the "Basic test statistic" in the output of `xtunitroot llc` is the standard t statistic for δ computed as

$$t_\delta = \hat{\delta}/\text{se}_{(\hat{\delta})}$$

where

$$\text{se}_{(\hat{\delta})} = \hat{\sigma}_{\tilde{\epsilon}} \left(\sum_{i=1}^{N} \sum_{t=p_i+2}^{T} \tilde{v}^2_{i,t-1} \right)^{-1/2}$$

$$\hat{\sigma}^2_{\tilde{\epsilon}} = \frac{1}{N\tilde{T}} \sum_{i=1}^{N} \sum_{t=p_i+2} \left(\tilde{e}_{it} - \delta \tilde{v}_{i,t-1} \right)^2$$

and $\tilde{T} = T - \overline{p} - 1$ with $\overline{p}$ the average of $p_1, \ldots, p_N$.

The adjusted test statistic is then computed as

$$t^*_\delta = \frac{t_\delta - N\tilde{T}\hat{S}_N \text{se}_{(\hat{\delta})} \mu^*_{\tilde{T}}}{\sigma^*_{\tilde{T}}}$$

where $\mu^*_{\tilde{T}}$ and $\sigma^*_{\tilde{T}}$ are obtained by linearly interpolating the values in LLC (2002, table 2). t^*_δ is asymptotically $N(0,1)$, with very negative values casting doubt on H_0. If `noconstant` is specified, then the asymptotic properties hold as $\sqrt{N}/T \to \infty$. Otherwise, T must grow at a faster rate so that $N/T \to \infty$.

Harris–Tsavalis test

The starting point for the HT test is (4), where ϵ_{it} is assumed to be i.i.d. normal with constant variance across panels. Denote by $\hat{\rho}$ the least-squares estimate of ρ.

HT show that $\sqrt{N}(\hat{\rho} - \mu) \xrightarrow{D} N(0, \sigma^2)$ as $N \to \infty$ with T fixed, where μ and σ^2 depend on the specification of the deterministic component:

Option	μ	σ^2
`noconstant`	1	$\frac{2}{T(T-1)}$
none	$1 - \frac{3}{T+1}$	$\frac{3(17T^2-20T+17)}{5(T-1)(T+1)^3}$
`trend`	$1 - \frac{15}{2(T+2)}$	$\frac{15(193T^2-728T+1147)}{112(T+2)^3(T-2)}$

Breitung test

Suppose the data are generated by an AR(1) process so that we can express y_{it} as

$$y_{it} = \mathbf{z}'_{it}\boldsymbol{\gamma}_i + x_{it}$$

where

$$x_{it} = \alpha_1 x_{i,t-1} + \alpha_2 x_{i,t-2} + \epsilon_{it}$$

where ϵ_{it} is an error term. A prewhitening step is available to correct for serial correlation. The nonrobust version assumes that ϵ_{it} is uncorrelated across panels, whereas the robust version allows for the panels to be contemporaneously correlated with covariance matrix $\mathbf{\Omega}$.

Under the null hypothesis that y_{it} contains a unit root, that is, that y_{it} is difference stationary, $\alpha_1 + \alpha_2 = 1$. Under the alternative that y_{it} is stationary, $\alpha_1 + \alpha_2 < 1$. Some of the time indices and summation limits of the formulas below appear more complex than those in Breitung (2000) and Breitung and Das (2005) because our formulas make explicit the loss of observations because of the prewhitening step.

Breitung test without trend

Let $y^{\ell}_{i,t} = y_{i,t-1} - y_{i,p+1}$ unless `noconstant` is specified, in which case let $y^{\ell}_{i,t} = y_{i,t-1}$. If the `lags()` option is specified with `xtunitroot breitung`, then we replace Δy_{it} and $y^{\ell}_{i,t}$ in the following description with the residuals from running regressions of Δy_{it} and $y^{\ell}_{i,t}$ on $\Delta y_{i,t-1}, \ldots, \Delta y_{i,t-p}$, where p is the lag order specified in `lags()`.

Define

$$\sigma_i^2 = \frac{1}{T-p-2} \sum_{t=p+2}^{T} (\Delta y_{it})^2$$

Then

$$\lambda = \frac{\sum_{i=1}^{N} \sum_{t=p+2}^{T} y^{\ell}_{it} \cdot \Delta y_{it}/\sigma_i^2}{\sqrt{\sum_{i=1}^{N} \sum_{t=p+2}^{T} (y^{\ell}_{it})^2/\sigma_i^2}}$$

λ is asymptotically distributed $N(0,1)$ as $T \to \infty$ followed by $N \to \infty$; small values of λ cast doubt on H_0.

For the robust version of the test statistic, let

$$\phi = \frac{\sum_{i=1}^{N} \sum_{t=p+2}^{T} y^{\ell}_{it} \cdot \Delta y_{it}/\sigma_i^2}{\sum_{i=1}^{N} \sum_{t=p+2}^{T} (y^{\ell}_{it})^2/\sigma_i^2}$$

and define $u_{it} = \Delta y_{it} - \phi y^{\ell}_{it}$. Let $\mathbf{u}_i = (u_{i,p+2}, \ldots, u_{iT})'$ and let the $N \times N$ matrix $\mathbf{\Omega}$ have typical element $\mathbf{u}_i'\mathbf{u}_j/(T-p-2)$. Let $\Delta \mathbf{y}_t = (\Delta y_{1t}, \ldots, \Delta y_{Nt})'$ and $\mathbf{y}^{\ell}_t = (y_{1,t-1}, \ldots, y_{N,t-1})'$. Then

$$\lambda_{\text{robust}} = \frac{\sum_{t=p+2}^{T} (\Delta \mathbf{y}_t)' \mathbf{y}^{\ell}_t}{\sum_{t=p+2}^{T} (\mathbf{y}^{\ell}_t)' \mathbf{\Omega} \mathbf{y}^{\ell}_t}$$

For $\mathbf{\Omega}$ to be positive definite, we must have $T-p-1 \geq N$. As a practical matter, for $\mathbf{\Omega}$ to have good finite-sample properties, we need $T \gg N$. λ_{robust} is asymptotically distributed $N(0,1)$ as $T \to \infty$ followed by $N \to \infty$; very negative values of λ_{robust} cast doubt on H_0.

Breitung test with trend

Let p denote the number of lags requested in the `lags()` option. We fit the regression

$$\Delta y_{it} = \alpha_{i0} + \sum_{j=1}^{p} \alpha_{ij} \Delta y_{i,t-j} + \nu_{it}$$

and compute the $1 \times (T-p-1)$ vectors $\Delta \mathbf{u}_i$ and $\mathbf{u}_i^\ell$ with typical elements

$$\Delta u_{is} = \Delta y_{is} - \sum_{j=1}^{p} \widehat{\alpha}_{ij} \Delta y_{i,s-j}$$

and

$$u_{is}^\ell = y_{i,s-1} - \sum_{j=1}^{p} \widehat{\alpha}_{ij} y_{i,s-j-1}$$

for $s = 1, \dots, T-p-1$. Let

$$\sigma_i^2 = \frac{1}{T-p-2} \sum_{s=1}^{T-p-1} \left(\Delta u_{is} - \overline{\Delta u}_i \right) \Delta u_{is}$$

where $\overline{\Delta u}_i$ is the mean of Δu_{is} over s. Let $\Delta \mathbf{v}_i$ and $\mathbf{v}_i^\ell$ denote $1 \times (T-p-1)$ vectors with typical elements

$$\Delta v_{is} = \sqrt{\frac{T-p-s-1}{T-p-s}} \left(\Delta u_{is} - \frac{1}{T-p-s-1} \sum_{j=s+1}^{T-p-1} \Delta u_{ij} \right)$$

and

$$v_{is}^\ell = u_{is}^\ell - u_{i1}^\ell - (T-p-1)\overline{\Delta u}_i$$

Now

$$\lambda = \frac{\sum_{i=1}^{N} \sum_{s=1}^{T-p-1} v_{is}^\ell \Delta v_{is} / \sigma_i^2}{\sqrt{\sum_{i=1}^{N} \sum_{s=1}^{T-p-1} (v_{is}^\ell)^2 / \sigma_i^2}}$$

λ is asymptotically distributed $N(0,1)$ as $T \to \infty$ followed by $N \to \infty$; very negative values of λ cast doubt on H_0. The computation of the robust form of the statistic proceeds in a fashion entirely analogous to the case without trend.

Im–Pesaran–Shin test

Write the model as

$$\Delta y_{it} = \phi_i y_{i,t-1} + \mathbf{z}_{it}' \boldsymbol{\gamma}_i + \epsilon_{it}$$

where ϵ_{it} is independently distributed normal for all i and t with panel-specific variance σ_i^2. Denote $\Delta \mathbf{y}_i = (\Delta y_{i2}, \dots, \Delta y_{iT})'$ and $\mathbf{y}_{i,-1} = (y_{i1}, \dots, y_{i,T-1})'$. Note that to be consistent with the notation used in the rest of this documentation, we start the time index at $t = 1$ instead of $t = 0$ as in IPS (2003). Also let τ_T be a conformable vector of ones, $\mathbf{M}_\tau = \mathbf{I} - \tau_T(\tau_T'\tau_T)^{-1}\tau_T'$, $\mathbf{X}_i = (\tau_T, \mathbf{y}_{i,-1})$, and $\mathbf{M}_{X_i} = \mathbf{I} - \mathbf{X}_i(\mathbf{X}_i'\mathbf{X}_i)^{-1}\mathbf{X}_i'$.

First, we consider the case of no serial correlation, where the user does not specify the `lags()` option. Then

$$\tilde{t}\text{-}bar_{NT} = \frac{1}{N} \sum_{i=1}^{N} \tilde{t}_{iT}$$

where

$$\tilde{t}_{iT} = \frac{\mathbf{\Delta y}_i'\mathbf{M}_\tau \mathbf{y}_{i,-1}}{\tilde{\sigma}_{iT}\left(\mathbf{y}_{i,-1}'\mathbf{M}_\tau\mathbf{y}_{i,-1}\right)^{1/2}}$$

and

$$\tilde{\sigma}_{iT}^2 = \frac{\mathbf{\Delta y}_i'\mathbf{M}_\tau\mathbf{\Delta y}_i}{T-1}$$

Also

$$t\text{-}bar_{NT} = \frac{1}{N}\sum_{i=1}^{N} t_{iT}$$

where

$$t_{iT} = \frac{\mathbf{\Delta y}_i'\mathbf{M}_\tau \mathbf{y}_{i,-1}}{\hat{\sigma}_{iT}\left(\mathbf{y}_{i,-1}'\mathbf{M}_\tau\mathbf{y}_{i,-1}\right)^{1/2}}$$

and

$$\tilde{\sigma}_{iT}^2 = \frac{\mathbf{\Delta y}_i'\mathbf{M}_{X_i}\mathbf{\Delta y}_i}{T-1}$$

Now

$$Z_{\tilde{t}\text{-}bar} = \frac{\sqrt{N}\left\{\tilde{t}\text{-}bar_{NT} - N^{-1}\sum_{i=1}^{N} E(\tilde{t}_{T_i})\right\}}{\sqrt{N^{-1}\sum_i \text{Var}(\tilde{t}_{T_i})}}$$

where $E(\tilde{t}_{T_i})$ and $\text{Var}(\tilde{t}_{T_i})$ are obtained by linearly interpolating the values shown in IPS (2003, table 1). $Z_{\tilde{t}\text{-}bar}$ has a standard normal limiting distribution for fixed T and $N \to \infty$; very negative values cast doubt on H_0. Similarly,

$$Z_{t\text{-}bar} = \frac{\sqrt{N}\left\{t\text{-}bar_{NT} - N^{-1}\sum_i E(t_{T_i})\right\}}{\sqrt{N^{-1}\sum_i \text{Var}(t_{T_i})}}$$

If the `lags()` option is specified, then we fit the ADF regressions

$$\Delta y_{it} = \phi_i y_{i,t-1} + \mathbf{z}_{it}'\boldsymbol{\gamma}_i + \sum_{j=1}^{p_i}\rho_{ij}\Delta y_{i,t-j} + \epsilon_i t$$

In matrix form, we can write this more compactly as

$$\mathbf{\Delta y}_i = \phi_i\mathbf{y}_{i,-1} + \mathbf{Q}_i\boldsymbol{\theta}_i + \epsilon_i$$

where $\mathbf{Q}_i = (\tau_t, \Delta\mathbf{y}_{i,-1}, \ldots, \Delta\mathbf{y}_{i,-p_i})$ and $\boldsymbol{\theta}_i = (\alpha_i, \rho_{i1}, \ldots, \rho_{ip_i})'$. Then

$$t\text{-}bar_{NT} = \frac{1}{N}\sum_{i=1}^{N} t_{iT}(p_i)$$

where

$$t_{iT}(p_i) = \frac{\sqrt{T-p_i-2}(\mathbf{y}_{i,-1}'\mathbf{M}_{Q_i}\mathbf{\Delta y}_i)}{(\mathbf{y}_{i,-1}'\mathbf{M}_{Q_i}\mathbf{y}_{i,-1})^{1/2}(\mathbf{\Delta y}_{i,-1}'\mathbf{M}_{Q_i}\mathbf{\Delta y}_{i,-1})^{1/2}}$$

where $\mathbf{M}_{Q_i} = \mathbf{I} - \mathbf{Q}_i(\mathbf{Q}_i'\mathbf{Q}_i)^{-1}\mathbf{Q}_i'$, $\mathbf{M}_{X_i} = \mathbf{I} - \mathbf{X}_i(\mathbf{X}_i'\mathbf{X}_i)^{-1}\mathbf{X}_i'$, and $\mathbf{X}_i = (\mathbf{y}_{i,-1}, \mathbf{Q}_i)$. Finally,

$$\mathbf{W}_{t\text{-}bar}(p) = \frac{\sqrt{N}\left[t\text{-}bar_{NT} - N^{-1}\sum_{i=1}^{N} E\{t_{iT}(p_i)\}\right]}{\sqrt{N^{-1}\sum_{i=1}^{N}\text{Var}\{t_{iT}(p_i)\}}}$$

where $E\{t_{iT}(p_i)\}$ and $\text{Var}\{t_{iT}(p_i)\}$ are obtained by linearly interpolating the values shown in IPS (2003, table 3). $\mathbf{W}_{t\text{-}bar}(p)$ has a standard normal limiting distribution as $T \to \infty$ followed by $N \to \infty$; very negative values cast doubt on H_0.

Fisher-type tests

We use `dfuller` or `pperron` to perform unit-root tests on each panel; denote the p-value for the respective test on the ith panel as p_i. All of these tests are predicated on $T \to \infty$ so that the unit-root test for each panel is consistent. The P test is for finite N; the other tests are valid whether N is finite or infinite. Then

$$P = -2\sum_{i=1}^{N} \ln(p_i)$$

$P \sim \chi^2(2N)$ and large values cast doubt on H_0.

$$Z = \frac{1}{\sqrt{N}}\sum_{i=1}^{N} \Phi^{-1}(p_i)$$

where $\Phi^{-1}(\,)$ is the inverse of the standard normal cumulative distribution function. $Z \sim N(0,1)$; very negative values of Z cast doubt on H_0.

$$L = \sum_{i=1}^{N} \ln\left(\frac{p_i}{1-p_i}\right)$$

$L^* = \sqrt{k}L \sim t(5N+4)$ where

$$k = \frac{3(5N+4)}{\pi^2 N(5N+2)}$$

Very negative values of L^* cast doubt on H_0. Finally,

$$P_m = -\frac{1}{\sqrt{N}}\sum_{i=1}^{N} \{\ln(p_i)+1\}$$

$P_m \sim N(0,1)$; very positive values of P_m cast doubt on H_0.

Hadri LM test

As discussed in the main text, the Hadri LM test can be viewed as a test of $H_0 : \sigma_u^2/\sigma_\epsilon^2 = 0$, where both u_{it} and ϵ_{it} are normally distributed random errors.

Let $\widehat{\epsilon}_{it}$ denote the residuals from a regression of y_{it} on a panel-specific intercept or a panel-specific intercept and time trend if `trend` is specified. Then

$$\widehat{\text{LM}} = \frac{\frac{1}{N}\sum_i \frac{1}{T^2}\sum_t S_{it}^2}{\widehat{\sigma}_\epsilon^2} \tag{10}$$

where

$$S_{it} = \sum_{j=1}^{t} \widehat{\epsilon}_{ij}$$

and

$$\widehat{\sigma}_{\epsilon}^{2}=\frac{1}{NT'}\sum_{i=1}^{N}\sum_{t=1}^{T}\widehat{\epsilon}_{it}^{2}$$

where $T' = T - 2$ if trend is specified and $T' = T - 1$ otherwise. Then

$$Z=\frac{\sqrt{N}\,(\widehat{\mathrm{LM}}-\mu)}{\sigma}$$

where $\mu = 1/15$ and $\sigma^2 = 11/6300$ if trend is specified and $\mu = 1/6$ and $\sigma^2 = 1/45$ otherwise. $Z \sim N(0,1)$ asymptotically as $T \to \infty$ followed by $N \to \infty$. Very positive values of Z cast doubt on H_0. If robust is specified, then we instead use

$$\widehat{\mathrm{LM}}=\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\sum_{t=1}^{T}S_{it}^{2}}{T^{2}\widehat{\sigma}_{\epsilon,i}^{2}}\right)$$

where we calculate $\widehat{\sigma}_{\epsilon,i}^{2}$ individually for each panel:

$$\widehat{\sigma}_{\epsilon,i}^{2}=\frac{1}{T'}\sum_{t=1}^{T}\widehat{\epsilon}_{it}^{2}$$

If kernel() is specified, then we use (10) with

$$\widehat{\sigma}_{\epsilon}^{2}=\frac{1}{N}\sum_{i=1}^{N}\left\{\frac{1}{T}\sum_{t=p+1}^{T}\widehat{\epsilon}_{it}^{2}+\frac{2}{T}\sum_{j=1}^{m}K(j,m)\sum_{t=j+1}^{T}\widehat{\epsilon}_{it}\widehat{\epsilon}_{i,t-j}\right\}$$

where m is the maximum number of lags and $K(.,.)$ is the kernel function defined previously.

Acknowledgments

We gratefully acknowledge users Fabian Bornhorst of the International Monetary Fund and Christopher F. Baum of the Department of Economics at Boston College and author of the Stata Press books *An Introduction to Modern Econometrics Using Stata* and *An Introduction to Stata Programming* and coauthor of the Stata Press book *Environmental Econometrics Using Stata* for the ipshin and levinlin commands; Christopher Baum for the hadrilm command; and Scott Merryman of the Risk Management Agency for the xtfisher command. We also gratefully acknowledge Jörg Breitung of the Faculty of Management, Economics, and Social Sciences at the University of Cologne for useful discussions when implementing his tests. These contributions were beneficial during the development of xtunitroot.

References

Baltagi, B. H. 2013. *Econometric Analysis of Panel Data*. 5th ed. Chichester, UK: Wiley.

Banerjee, A., M. Marcellino, and C. Osbat. 2005. Testing for PPP: Should we use panel methods? *Empirical Economics* 30: 77–91. https://doi.org/10.1007/s00181-004-0222-8.

Baum, C. F., and J. Otero. 2021. Unit-root tests for explosive behavior. *Stata Journal* 21: 999–1020.

Bersvendsen, T., and J. Ditzen. 2021. Testing for slope heterogeneity in Stata. *Stata Journal* 21: 51–80.

Breitung, J. 2000. “The local power of some unit root tests for panel data”. In *Nonstationary Panels, Panel Cointegration, and Dynamic Panels*. Advances in Econometrics, edited by B. H. Baltagi, vol. 15: 161–178. Amsterdam: JAI Press.

Breitung, J., and S. Das. 2005. Panel unit root tests under cross-sectional dependence. *Statistica Neerlandica* 59: 414–433. https://doi.org/10.1111/j.1467-9574.2005.00299.x.

Chen, P., Y. Karavias, and E. Tzavalis. 2022. Panel unit-root tests with structural breaks. *Stata Journal* 22: 664–678.

Choi, I. 2001. Unit root tests for panel data. *Journal of International Money and Finance* 20: 249–272. https://doi.org/10.1016/S0261-5606(00)00048-6.

Ditzen, J. 2018. Estimating dynamic common-correlated effects in Stata. *Stata Journal* 18: 585–617.

———. 2021. Estimating long-run effects and the exponent of cross-sectional dependence: An update to xtdcce2. *Stata Journal* 21: 687–707.

Hadri, K. 2000. Testing for stationarity in heterogeneous panel data. *Econometrics Journal* 3: 148–161. https://doi.org/10.1111/1368-423X.00043.

Harris, R. D. F., and E. Tzavalis. 1999. Inference for unit roots in dynamic panels where the time dimension is fixed. *Journal of Econometrics* 91: 201–226. https://doi.org/10.1016/S0304-4076(98)00076-1.

Heston, A., R. Summers, and B. Aten. 2006. Penn World Table Version 6.2. Center for International Comparisons of Production, Income, and Prices, University of Pennsylvania. https://www.rug.nl/ggdc/productivity/pwt/pwt-releases/pwt-6.2.

Hlouskova, J., and M. Wagner. 2006. The performance of panel unit root and stationarity tests: Results from a large scale simulation study. *Econometric Reviews* 25: 85–116. https://doi.org/10.1080/07474930500545504.

Im, K. S., M. H. Pesaran, and Y. Shin. 2003. Testing for unit roots in heterogeneous panels. *Journal of Econometrics* 115: 53–74. https://doi.org/10.1016/S0304-4076(03)00092-7.

Kwiatkowski, D., P. C. B. Phillips, P. Schmidt, and Y. Shin. 1992. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of Econometrics* 54: 159–178. https://doi.org/10.1016/0304-4076(92)90104-Y.

Levin, A., C.-F. Lin, and C.-S. J. Chu. 2002. Unit root tests in panel data: Asymptotic and finite-sample properties. *Journal of Econometrics* 108: 1–24. https://doi.org/10.1016/S0304-4076(01)00098-7.

Maddala, G. S., and S. Wu. 1999. A comparative study of unit root tests with panel data and a new simple test. *Oxford Bulletin of Economics and Statistics* 61: 631–652. https://doi.org/10.1111/1468-0084.0610s1631.

Neal, T. 2014. Panel cointegration analysis with xtpedroni. *Stata Journal* 14: 684–692.

Newey, W. K., and K. D. West. 1994. Automatic lag selection in covariance matrix estimation. *Review of Economic Studies* 61: 631–653. https://doi.org/10.2307/2297912.

Nickell, S. J. 1981. Biases in dynamic models with fixed effects. *Econometrica* 49: 1417–1426. https://doi.org/10.2307/1911408.

Obstfeld, M., and K. Rogoff. 1996. *Foundations of International Macroeconomics*. Cambridge, MA: MIT Press.

O'Connell, P. G. J. 1998. The overvaluation of purchasing power parity. *Journal of International Economics* 44: 1–19. https://doi.org/10.1016/S0022-1996(97)00017-2.

Oh, K.-Y. 1996. Purchasing power parity and unit root tests using panel data. *Journal of International Money and Finance* 15: 405–418. https://doi.org/10.1016/0261-5606(96)00012-5.

Patterson, K. 2000. *An Introduction to Applied Econometrics: A Time Series Approach*. Basingstoke, UK: Palgrave.

Phillips, P. C. B., and H. R. Moon. 1999. Linear regression limit theory for nonstationary panel data. *Econometrica* 67: 1057–1111. https://doi.org/10.1111/1468-0262.00070.

———. 2000. Nonstationary panel data analysis: An overview of some recent developments. *Econometric Reviews* 19: 263–286. https://doi.org/10.1080/07474930008800473.

Rogoff, K. 1996. The purchasing power parity puzzle. *Journal of Economic Literature* 34: 647–668.

Whitehead, A. 2002. *Meta-Analysis of Controlled Clinical Trials*. Chichester, UK: Wiley.

Also see

[XT] **xtcointtest** — Panel-data cointegration tests

[TS] **dfgls** — DF-GLS unit-root test

[TS] **dfuller** — Augmented Dickey–Fuller unit-root test

[TS] **pperron** — Phillips–Perron unit-root test

xtvar — Panel-data vector autoregressive models

Description

`xtvar` fits vector autoregressive (VAR) models to panel data. Similar to VAR models for time-series data, `xtvar` models each dependent variable as a function of its own lags, the lags of all other dependent variables, and a panel-level fixed effect. Other explanatory variables can be added to the model as well; these variables can be predetermined, fully exogenous, or endogenous.

Quick start

Fit a panel-data VAR model with dependent variables `y1`, `y2`, and `y3` using `xtset` data

```
xtvar y1 y2 y3
```

Same as above, but use three lags of the dependent variables instead of one lag

```
xtvar y1 y2 y3, lags(3)
```

Same as above, but limit the number of lags of the dependent variables used as instruments to two

```
xtvar y1 y2 y3, lags(3) maxldep(2)
```

Same as above, but include exogenous variables `x1` and `x2`

```
xtvar y1 y2 y3, lags(3) maxldep(2) exogenous(x1 x2)
```

Same as above, but use first differences instead of the default forward-orthogonal deviations

```
xtvar y1 y2 y3, lags(3) maxldep(2) exogenous(x1 x2) fd
```

Same as above, but use the one-step generalized method of moments (GMM) estimator instead of the default two-step estimator

```
xtvar y1 y2 y3, lags(3) maxldep(2) exogenous(x1 x2) fd onestep
```

Menu

Statistics > Longitudinal/panel data > Dynamic panel data (DPD) > Panel-data VAR estimation

Syntax

`xtvar` *depvarlist* [*if*] [*in*] [, *options*]

options	Description
Model	
`lags(#)`	specify the number of lags for the dependent variables; default is `lags(1)`
`fodeviation`	use forward-orthogonal deviations to remove fixed effects; the default
`fd`	use first differences to remove fixed effects
`minldep(#)`	specify minimum number of lags of dependent, endogenous, and predetermined variables to use as instruments; default is `minldep(1)`
`maxldep(#)`	specify maximum lags of dependent, endogenous, and predetermined variables to use as instruments; default is all lags
`collapse`	collapse moment conditions from all time periods within each panel
Additional regressors	
`exogenous(`*varlist*`)`	specify strictly exogenous regressors
`endogenous(`*varlist*`)`	specify endogenous variables
`predetermined(`*varlist*`)`	specify predetermined variables
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `wcrobust`, `robust`, `bootstrap`, or `jackknife`; default is `vce(wcrobust)` for the two-step estimator and `vce(robust)` for the one-step estimator
Reporting	
`level(#)`	set confidence level; default is `level(95)`
display_options	control columns and column formats, row spacing, and line width
GMM	
`winitial(xt` | `identity)`	specify initial weight matrix; default is `winitial(xt)`
`onestep`	use the one-step GMM estimator rather than the two-step GMM estimator
`coeflegend`	display legend instead of statistics

You must `xtset` your data before using `xtvar`; see [XT] **xtset**.

depvarlist and *varlist* may contain time-series operators; see [U] **11.4.4 Time-series varlists**.

`by`, `collect`, and `statsby` are allowed; see [U] **11.1.10 Prefix commands**.

`coeflegend` does not appear in the dialog box.

See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

lags(#) specifies the lags of the dependent variables to be included in the model. The default is lags(1).

fodeviation requests that the fixed effects be removed from the model by applying the forward-orthogonal-deviation (FOD) transformation to all variables in the model. This is the default.

fd requests that the fixed effects be removed from the model by taking first-difference (FD) transformation of all variables in the model.

minldep(#) specifies the minimum number of lags of the dependent, endogenous, and predetermined variables that need to be available for use as instruments for an observation to be included in the estimation sample.

maxldep(#) specifies the maximum number of lags of the dependent, endogenous, and predetermined variables to use as instruments. The default is to use all available lags of all of these variables as instruments. To request the default explicitly, specify maxldep(.).

The number of lags of the predetermined variables, $\mathbf{v}_{it}$, that will be used as instruments is actually $\#+1$. This is because for time t, $\mathbf{v}_{i,t-1}$ is a valid instrument, but the same is not true for the dependent variables or endogenous variables.

collapse requests that xtvar use a version of the moment conditions that sums across t within each panel. The standard GMM estimator can be severely biased when the number of moment conditions (which are based on the number of instruments) is large, as often happens with dynamic panel-data models. Collapsing the moment conditions ameliorates that bias at the expense of yielding a less-efficient estimator.

Additional regressors

exogenous(*varlist*) specifies a list of strictly exogenous variables to be included as regressors in the model.

endogenous(*varlist*) specifies a set of endogenous variables to be included in the model. By default, all available lags for these endogenous variables will be used as instruments; additionally, at least one lag of the endogenous variables must be available for an observation to be included in the estimation sample. You can use the maxldep() option to specify the maximum number of lags to use as instruments and the minldep() option to specify the minimum number of lags that need to be available to use as instruments for an observation to be included in the estimation sample.

predetermined(*varlist*) specifies a set of predetermined variables to be included in the model. A predetermined variable is a variable that at time t is affected by the error terms in previous time periods but not affected by the error terms in the current time period, t. By default, all available lags for these predetermined variables will be used as instruments; additionally, at least one lag of the predetermined variables must be available for an observation to be included in the estimation sample. You can use the maxldep() option to specify the maximum number of lags to use as instruments and the minldep() option to specify the minimum number of lags that need to be available to use as instruments for an observation to be included in the estimation sample.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (wcrobust, robust) and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] ***vce_options***.

vce(wcrobust), the default with the two-step estimator, requests the Windmeijer (2005) robust variance–covariance estimator, which corrects for the downward bias of the usual GMM cluster–robust variance–covariance estimator. This option is not available if you request the one-step estimator with the onestep option.

vce(robust), the default with the one-step estimator, requests the robust (sandwich) GMM variance–covariance estimator (VCE), which allows for intragroup correlation at the panel level. This is the default if you specify the onestep option. In small samples, this performs well with the one-step GMM estimator but is downward biased with the two-step estimator.

vce(bootstrap) and vce(jackknife) request VCEs based on the bootstrap or jackknife, respectively, where sampling is done at the panel level.

Reporting

level(#); see [R] **Estimation options**.

display_options: noci, nopvalues, vsquish, cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

GMM

winitial(xt | identity) specifies the initial weight matrix.

winitial(xt), the default, specifies an initial weight matrix based on the instruments specified with the model; additionally, this matrix assumes the idiosyncratic errors are homoskedastic. The exact form of the matrix depends on whether the FOD or FD transform is used to remove the model's fixed effects. See *Estimators* in *Methods and formulas* for more details.

winitial(identity) requests that the identity matrix be used as the initial weight matrix.

onestep specifies that the one-step estimator be calculated; the default is to use the two-step estimator.

The following option is available with xtvar but is not shown in the dialog box:

coeflegend; see [R] **Estimation options**.

Remarks and examples

Remarks are presented under the following headings:

Introduction

In a panel-data VAR model, each dependent variable is modeled as a function of its own lags, the lags of all other dependent variables, a panel-level fixed effect, and possibly other additional covariates. Therefore, panel-data VAR models combine elements of dynamic panel-data estimators and elements of time-series VAR models. To better understand the estimators implemented by `xtvar`, you should familiarize yourself with the Arellano and Bond (1991) estimator for dynamic panels (described in [XT] **xtabond**) as well as time-series VARs (described in [TS] **var**).

Below, we provide a short formal introduction to panel-data VAR models and their estimation; you can find more technical details in *Methods and formulas*. Then we provide a practical guide on how to fit a panel-data VAR model using `xtvar` and how to specify options that address common issues that arise with these models. We demonstrate how to check whether your model satisfies the moment conditions that are necessary for correct specification. We then examine the options available to reduce the number of moment conditions in the model and discuss in which cases you might want to use them. Additionally, we present the postestimation tools available after `xtvar` that can be used for diagnostics, testing, and interpretation. Many of these commands are ones you would use after fitting a VAR model using `var`. For instance, we discuss impulse–response functions (IRFs), which can be obtained by using the `irf` commands, to see the effects of a shock on an endogenous variable on itself or other endogenous variables; Granger causality tests, which can be performed by using `vargranger`; model selection that can be performed by using `xtvarsoc`; and lag-exclusion restrictions that can be tested by using `varwle`.

Panel-data VAR model formulation

Here we introduce basic aspects of a panel-data VAR model. As we mentioned above, this model combines elements of a dynamic panel-data model and elements of a VAR model. A benefit of a panel-data VAR model over a dynamic panel-data model is that we can model multiple dependent variables and their intertemporal relation. A benefit over a VAR model is that we can account for time-invariant heterogeneity. From both frameworks, we get the benefit of being able to explore dynamic behavior.

Panel-data VAR models express each of a set of K variables as a linear function of p of its own lags, p lags of the other $K-1$ variables, and, optionally, other covariates. The most general panel-data VAR model with p lags that `xtvar` accommodates is

$$\mathbf{y}_{it} = \mathbf{A}_1\mathbf{y}_{i,t-1} + \mathbf{A}_2\mathbf{y}_{i,t-2} + \cdots + \mathbf{A}_p\mathbf{y}_{i,t-p} + \mathbf{B}\mathbf{x}_{it} + \mathbf{C}\mathbf{w}_{it} + \mathbf{D}\mathbf{v}_{it} + \mathbf{u}_i + \boldsymbol{\epsilon}_{it} \tag{1}$$

where

$\mathbf{y}_{it}$ is a $K \times 1$ vector of dependent variables;

$\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_p$ are $K \times K$ matrices of parameters;

$\mathbf{x}_{it}$ is an $M_1 \times 1$ vector of strictly exogenous variables;

$\mathbf{B}$ is a $K \times M_1$ matrix of parameters;

$\mathbf{w}_{it}$ is an $M_2 \times 1$ vector of endogenous variables;

$\mathbf{C}$ is a $K \times M_2$ matrix of parameters;

$\mathbf{v}_{it}$ is an $M_3 \times 1$ vector of predetermined variables;

$\mathbf{D}$ is a $K \times M_3$ matrix of parameters;

$\mathbf{u}_i$ is a $K \times 1$ fixed-effects vector;

$\boldsymbol{\epsilon}_{it}$ is a $K \times 1$ vector of serially uncorrelated idiosyncratic errors;

$i = 1, \ldots, N$ denotes the ith panel; and

$t = 1, \ldots, T$ denotes the tth time period.

Note that (1) does not include a constant term because it cannot be identified together with the fixed effects $\mathbf{u}_i$.

Strictly exogenous variables, $\mathbf{x}_{it}$, are uncorrelated with past, present, or future realizations of the error term. Endogenous regressors, $\mathbf{w}_{it}$, on the other hand, may be correlated with the present and past realizations of the error term. Predetermined variables, $\mathbf{v}_{it}$, represent an intermediate case; they are not correlated with the present period's error term but may be correlated with past realizations of it. Both endogenous and predetermined variables are assumed to be uncorrelated with the future realizations of the error term. The sets of exogenous, endogenous, and predetermined variables determine the instrument set to be used and the complexity of your model.

To simplify our discussion, suppose that the only covariates in the model are the lags of the dependent variables. Thus, we have

$$\mathbf{y}_{it} = \mathbf{A}_1\mathbf{y}_{i,t-1} + \mathbf{A}_2\mathbf{y}_{i,t-2} + \cdots + \mathbf{A}_p\mathbf{y}_{i,t-p} + \mathbf{u}_i + \boldsymbol{\epsilon}_{it} \tag{2}$$

To fit this model, we first need to remove the fixed-effect term $\mathbf{u}_i$. `xtvar` provides two ways to do that: using either the FD transformation or the FOD transformation. See *Eliminating the fixed effect* in *Methods and formulas* for full descriptions of these transformations.

Applying the FD transformation to (2), we obtain

$$\left(\mathbf{y}_{it} - \mathbf{y}_{i,t-1}\right) = \mathbf{A}_1\left(\mathbf{y}_{i,t-1} - \mathbf{y}_{i,t-2}\right) + \cdots + \mathbf{A}_p\left(\mathbf{y}_{i,t-p} - \mathbf{y}_{i,t-p-1}\right) + \left(\boldsymbol{\epsilon}_{it} - \boldsymbol{\epsilon}_{i,t-1}\right)$$

Using tildes to represent transformed variables, we can write this more succinctly as

$$\tilde{\mathbf{y}}_{it} = \mathbf{A}_1\tilde{\mathbf{y}}_{i,t-1} + \mathbf{A}_2\tilde{\mathbf{y}}_{i,t-2} + \cdots + \mathbf{A}_p\tilde{\mathbf{y}}_{i,t-p} + \tilde{\boldsymbol{\epsilon}}_{it} \tag{3}$$

While we used the FD transformation, the discussion that follows also applies if $\tilde{\mathbf{y}}$ and $\tilde{\boldsymbol{\epsilon}}$ were produced by the FOD transformation.

Note that the first period at which we can evaluate (3) is $t = p + 2$.

We have removed the fixed-effect term, but the transformed regressor $\tilde{\mathbf{y}}_{i,t-1}$ is still endogenous because it is correlated with the transformed error $\tilde{\boldsymbol{\epsilon}}_{i,t-1}$ (through its dependence on $\boldsymbol{\epsilon}_{i,t-1}$). As shown by Nerlove (1967) and Nickell (1981) in the single-equation context, failing to account for this endogeneity may result in considerable bias, specially when T is small.

We must therefore find suitable instruments, $\mathbf{z}_{it}$, that are correlated with our endogenous regressors but not with our error term. We use the moment conditions $E\{\text{vec}(\mathbf{z}_{it}\tilde{\boldsymbol{\epsilon}}'_{it})\} = \mathbf{0}$ and GMM to estimate our parameters. The asymptotics of the GMM estimator rely on the number of panels going to infinity, so it is important that the number of panels in your data is not too small. For more background on GMM, Hayashi (2000), Cameron and Trivedi (2022), and Wooldridge (2010) provide textbook treatments. The monograph by Hall (2005) contains more detail. Also see [R] **gmm**. For more on the specific application of the GMM estimator in the context of panel-data VAR models, see *A concise representation of the GMM estimator* in *Methods and formulas*.

As in the dynamic panel literature (see Arellano and Bond [1991]), we can use the lags of the dependent variables as instruments for our endogenous regressors. Specifically, for endogenous regressor $\tilde{\mathbf{y}}_{i,t-1}$, we can use $\mathbf{y}_{i,t-2}, \mathbf{y}_{i,t-3}, \ldots, \mathbf{y}_{i,1}$, as instruments. Similarly, for endogenous regressor $\tilde{\mathbf{y}}_{i,t-2}$, we can use lags 3 and greater of $\mathbf{y}_{i,t}$ as instruments, and so on.

The total number of moment conditions in our GMM estimator grows rapidly with the T and K. In fact, the number of moment conditions in this simplified model with no additional covariates equals $(K^2/2)(T-p-1)(T+p-2)$. This multiplicity of moment conditions (and the multiplicity of instruments from which they are derived) poses serious challenges to estimation via GMM because some of the instruments may be only weakly relevant. See Han and Phillips (2006), Stock and Wright (2000), and Roodman (2009a).

xtvar has tools to ameliorate the issues arising from instrument proliferation. We can use the maxldep() and collapse options. The maxldep() option caps the total number of lags used as instruments for each endogenous regressor. So, for instance, specifying maxldep(2) requests that at most two lags be used as instruments for each endogenous regressor. The collapse option, on the other hand, reduces the number of moment conditions by taking averages of individual moment conditions and specifying the GMM estimator in terms of the averaged conditions instead of the individual ones. We will illustrate how to use the maxldep() and collapse options in the examples below.

So far, we have mentioned issues that are inherent to dynamic panel-data models. We also need to address the VAR dimension. We need to select the number of lags to include in our model, check that we have a stationary process, verify that the relationships between dependent variables are meaningful and have predictive power, and estimate the effects of shocks over time. We can do this with xtvar and its postestimation tools as we will demonstrate in examples below.

Fitting a panel-data VAR model with xtvar

▷ Example 1: Our first panel VAR

Blomquist and Dahlberg (1999) used a dataset consisting of 265 Swedish municipalities observed over 9 years. For each municipality, the dataset has variables recording expenditures, revenues, and grants from the central government. We want to fit a panel-data VAR to see how grants from the central government affect municipalities' expenditures and revenues over time. We first load in the dataset and describe its panel structure:

```
. use https://www.stata-press.com/data/r19/swedishgov
(1979-1987 Swedish municipality data)

. xtdescribe

  idcode:  114, 115, ..., 2584                              n =        265
    year:  1979, 1980, ..., 1987                            T =          9
           Delta(year) = 1 year
           Span(year)  = 9 periods
           (idcode*year uniquely identifies each observation)

Distribution of T_i:   min      5%     25%       50%       75%     95%     max
                         9       9       9         9         9       9       9

     Freq.  Percent    Cum. |  Pattern
 ---------------------------+-----------
      265    100.00  100.00 |  111111111
 ---------------------------+-----------
      265    100.00         |  XXXXXXXXX
```

The data have already been xtset to specify that idcode is the panel identifier and year is the time variable. The output from xtdescribe shows that we have nine years of data and that there are no gaps in our dataset. xtvar supports datasets with gaps, but we defer that discussion until later.

We fit a panel-data VAR with dependent variables `expenditures`, `revenue`, and `grants` as follows:

```
. xtvar expenditure revenues grants
Panel-data vector autoregression                  Number of obs    = 1,855
Group variable: idcode                            Number of groups =   265
Time variable: year                               Obs per group:
                                                               min =     7
Number of moment conditions = 252                              avg =   7.0
                                                               max =     7
Fixed-effects transform: FOD
Two-step results
                               (Std. err. adjusted for 265 clusters in idcode)
------------------------------------------------------------------------------
             |              WC robust
             | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
expenditures |
expenditures |
         L1. |   .2839341     .06484     4.38   0.000     .1568501    .4110182
             |
    revenues |
         L1. |  -.0451041   .0622281    -0.72   0.469     -.167069    .0768607
             |
      grants |
         L1. |   -1.68128   .2770326    -6.07   0.000    -2.224254   -1.138307
-------------+----------------------------------------------------------------
revenues     |
expenditures |
         L1. |   .2568554   .0781264     3.29   0.001     .1037304    .4099804
             |
    revenues |
         L1. |   .0598285   .0709236     0.84   0.399    -.0791791    .1988361
             |
      grants |
         L1. |   -2.24419   .2805223    -8.00   0.000    -2.794003   -1.694376
-------------+----------------------------------------------------------------
grants       |
expenditures |
         L1. |   .0164546   .0165141     1.00   0.319    -.0159124    .0488215
             |
    revenues |
         L1. |  -.0404274   .0143271    -2.82   0.005     -.068508   -.0123468
             |
      grants |
         L1. |   .3179538   .0506388     6.28   0.000     .2187037    .4172039
------------------------------------------------------------------------------
Hansen's test of overid. restrictions: chi2(243) = 264.16  Prob > chi2 = 0.168
GMM-type instruments: L(2/.).(expenditures revenues grants)
```

The first part of the output summarizes the panel structure of the estimation sample. The right-hand side shows that we have 265 groups (or panels) and 7 observations per group. The left-hand side shows that we used the FOD transformation and the default two-step GMM estimator with 252 moment conditions.

The coefficient table shows the estimated coefficients, standard errors, and related information. Our panel-data VAR includes three dependent variables, so there are three equations displayed in the table. We also have, by default, one lag of each dependent variable, so for each equation we see the coefficients for the first lag of each of the three variables.

Standard errors are labeled `WC robust`, meaning that they are based on the correction proposed in Windmeijer (2005) and thus robust to arbitrary within-panel correlation. We recommend using these standard errors with the two-step GMM estimator.

The footer of the table provides two valuable pieces of information: Hansen's J test of overidentifying restrictions and the instruments used to fit the model. A rejection of Hansen's test indicates that the moment conditions necessary for the correct specification of the model do not seem to hold. In this case, the test did not reject the null hypothesis that the moment conditions of our model hold.

◁

Modifying lags

While Hansen's J test did not indicate a problem with our model, we have reasons to believe that the effect of government grants on expenditures and revenues may last for more than one period. This would imply that our model with just one lag is misspecified. In the next examples, we explore models with more lags to address this issue, and we use postestimation tools to select the optimal number of lags and test their relevance in the model.

▷ Example 2: Adding lags and restricting the number used as instruments

To allow for dependence among dependent variables beyond the first lag, we can specify the `lags()` option; here we use two lags of the dependent variables by specifying `lags(2)`.

In example 1, we used all available lags of the dependent variables as instruments. This resulted in 252 moment conditions. As we mentioned before, having a proliferation of instruments and moment conditions makes our estimates less reliable. To reduce the number of moment conditions in the model, we specify the `maxldep(2)` option and fit our model using at most two lags as instruments for each endogenous regressor.

```
. xtvar expenditures revenues grants, lags(2) maxldep(2)
Panel-data vector autoregression                    Number of obs     = 1,590
Group variable: idcode                              Number of groups =   265
Time variable: year                                 Obs per group:
                                                                 min =     6
Number of moment conditions = 108                                avg =   6.0
                                                                 max =     6
Fixed-effects transform: FOD
Two-step results
                               (Std. err. adjusted for 265 clusters in idcode)
------------------------------------------------------------------------------
             |              WC robust
             | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
expenditures |
expenditures |
         L1. |   .1956019   .1147648     1.70   0.088    -.0293329    .4205367
         L2. |   .0017664    .100328     0.02   0.986    -.1948728    .1984056
             |
    revenues |
         L1. |   -.163357   .1162282    -1.41   0.160      -.39116    .0644459
         L2. |  -.3363544   .1003698    -3.35   0.001    -.5330756   -.1396332
             |
      grants |
         L1. |   -4.08135   .6900914    -5.91   0.000    -5.433904   -2.728796
         L2. |  -1.883438   .2732505    -6.89   0.000    -2.418999   -1.347877
-------------+----------------------------------------------------------------
revenues     |
expenditures |
         L1. |   .1709229   .1220747     1.40   0.161    -.0683391    .4101848
         L2. |   .0525276   .1051278     0.50   0.617    -.1535191    .2585742
             |
    revenues |
         L1. |   -.092228   .1237745    -0.75   0.456    -.3348216    .1503656
         L2. |    -.32843   .0984127    -3.34   0.001    -.5213154   -.1355446
             |
      grants |
         L1. |    -4.7028   .6957627    -6.76   0.000     -6.06647    -3.33913
         L2. |  -2.054873   .2618687    -7.85   0.000    -2.568126    -1.54162
-------------+----------------------------------------------------------------
grants       |
expenditures |
         L1. |   .0162825    .018789     0.87   0.386    -.0205433    .0531083
         L2. |   .0180168   .0164781     1.09   0.274    -.0142796    .0503132
             |
    revenues |
         L1. |  -.0281669   .0177173    -1.59   0.112    -.0628921    .0065583
         L2. |  -.0162105   .0161942    -1.00   0.317    -.0479506    .0155296
             |
      grants |
         L1. |   .2331196   .0762458     3.06   0.002     .0836807    .3825586
         L2. |   .1016583   .0487391     2.09   0.037     .0061315    .1971852
------------------------------------------------------------------------------
Hansen's test of overid. restrictions: chi2(90) = 228.48   Prob > chi2 = 0.000
GMM-type instruments: L(2/3).(expenditures revenues grants)
```

The bottom of the output now indicates that we have instrumented each of the endogenous regressors using lags two and three (`L(2/3)`) of the dependent variables when both are available. Otherwise, we instrumented the endogenous regressor using just lag two. The number of moment conditions is now 108 versus 252 in example 1. We have made progress in reducing the number of moment conditions. However, Hansen's J test provides strong evidence that our moment conditions are not valid. So we would hesitate to use these results without further investigation. You can verify that when we fit the model specifying `maxldep(1)` or `maxldep(3)`, we continue to reject the null hypothesis that our moment conditions are valid.

One possibility is that our lag specification of the model is still inadequate. If we do not include enough lags in the model, then the errors could be serially correlated. In that case, lags two and greater of the dependent variables are no longer valid instruments. Let's explore a model with four lags of the dependent variables as regressors and a maximum of two lags of the dependent variables to be used as instruments.

```
. xtvar expenditures revenues grants, lags(4) maxldep(2)

Panel-data vector autoregression                      Number of obs     = 1,060
Group variable: idcode                                Number of groups  =   265
Time variable: year                                   Obs per group:
                                                                    min =     4
Number of moment conditions = 72                                    avg =   4.0
                                                                    max =     4

Fixed-effects transform: FOD
Two-step results
                               (Std. err. adjusted for 265 clusters in idcode)
------------------------------------------------------------------------------
             |               WC robust
             | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
expenditures |
expenditures |
         L1. |   .3043156   .2596238     1.17   0.241    -.2045376    .8131689
         L2. |    .176059   .2198797     0.80   0.423    -.2548973    .6070153
         L3. |   .0807466   .2439179     0.33   0.741    -.3973237     .558817
         L4. |    .362827   .4003709     0.91   0.365    -.4218855    1.147539
             |
    revenues |
         L1. |  -.2788411   .2972961    -0.94   0.348    -.8615308    .3038486
         L2. |  -.2349415   .2596951    -0.90   0.366    -.7439345    .2740515
         L3. |  -.2040393   .2356327    -0.87   0.387    -.6658708    .2577923
         L4. |  -.5232187   .3816921    -1.37   0.170    -1.271321     .224884
             |
      grants |
         L1. |   1.012279   .9715139     1.04   0.297     -.891853    2.916412
         L2. |   .3069275   .4503541     0.68   0.496    -.5757503    1.189605
         L3. |   1.000217   .5569212     1.80   0.072    -.0913287    2.091762
         L4. |    .077072   1.548516     0.05   0.960    -2.957963    3.112107
```

revenues						
expenditures						
L1.	.5287675	.2030516	2.60	0.009	.1307937	.9267412
L2.	.3753373	.1738501	2.16	0.031	.0345973	.7160773
L3.	.2828897	.1838684	1.54	0.124	-.0774858	.6432651
L4.	.693354	.3521214	1.97	0.049	.0032087	1.383499
revenues						
L1.	-.4533828	.2293747	-1.98	0.048	-.902949	-.0038167
L2.	-.4150536	.207767	-2.00	0.046	-.8222694	-.0078378
L3.	-.4150709	.1865944	-2.22	0.026	-.7807891	-.0493527
L4.	-.7522478	.3355086	-2.24	0.025	-1.409833	-.0946629
grants						
L1.	.0457337	.7650976	0.06	0.952	-1.45383	1.545298
L2.	-.1807682	.3181763	-0.57	0.570	-.8043823	.442846
L3.	.5877744	.5370399	1.09	0.274	-.4648045	1.640353
L4.	-.5295232	1.330482	-0.40	0.691	-3.137221	2.078174
grants						
expenditures						
L1.	-.0747673	.0932016	-0.80	0.422	-.2574391	.1079044
L2.	-.039655	.0806699	-0.49	0.623	-.1977651	.118455
L3.	-.1083643	.0865062	-1.25	0.210	-.2779134	.0611848
L4.	-.0165543	.1265477	-0.13	0.896	-.2645833	.2314747
revenues						
L1.	.0551585	.1044118	0.53	0.597	-.1494848	.2598018
L2.	.0471306	.0934546	0.50	0.614	-.1360369	.2302982
L3.	.1051582	.0851519	1.23	0.217	-.0617365	.2720529
L4.	-.0509242	.1199238	-0.42	0.671	-.2859705	.184122
grants						
L1.	-.2410357	.2761211	-0.87	0.383	-.7822232	.3001517
L2.	-.0788769	.141957	-0.56	0.578	-.3571075	.1993537
L3.	.2370528	.2222193	1.07	0.286	-.1984889	.6725946
L4.	.2673538	.4635082	0.58	0.564	-.6411055	1.175813

```
Hansen's test of overid. restrictions: chi2(36) = 38.80    Prob > chi2 = 0.345
GMM-type instruments: L(2/3).(expenditures revenues grants)
```

As we can see from Hansen's J test, we cannot reject the null that the moment conditions in our model hold. We were previously concerned about both the lag structure and the number of instruments. Hansen's J test suggests that the lag structure was the more relevant concern in our case.

◁

Reducing moment conditions by collapsing the instrument matrix

Collapsing the instrument matrix (and thus collapsing the moment conditions) is another alternative to address the issues that come with instrument proliferation. We explore it in the next example.

▷ Example 3: Collapsing the instrument matrix

We refit our municipal finance model with two lags. This time we use all available lags as instruments but request the collapsed version of the instrument matrix by specifying the `collapse` option. This reduces the number of moment conditions because the GMM estimation now uses average moment conditions across time instead of individual moment conditions.

```
. xtvar expenditures revenues grants, lags(2) collapse

Panel-data vector autoregression                    Number of obs     = 1,590
Group variable: idcode                              Number of groups =    265
Time variable: year                                 Obs per group:
                                                                 min =      6
Number of moment conditions = 63 (collapsed)                     avg =    6.0
                                                                 max =      6

Fixed-effects transform: FOD
Two-step results
                               (Std. err. adjusted for 265 clusters in idcode)

                              WC robust
             | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
expenditures |
expenditures |
         L1. |   .1900148   .1513004     1.26   0.209    -.1065285    .4865581
         L2. |   .0327313    .147236     0.22   0.824    -.2558458    .3213085
             |
    revenues |
         L1. |  -.2920254   .1556137    -1.88   0.061    -.5970226    .0129719
         L2. |  -.4337591   .1341645    -3.23   0.001    -.6967167   -.1708014
             |
      grants |
         L1. |  -5.062357   .9626468    -5.26   0.000     -6.94911   -3.175604
         L2. |  -2.221608   .4108095    -5.41   0.000     -3.02678   -1.416436
-------------+----------------------------------------------------------------
revenues     |
expenditures |
         L1. |   .1919241   .1543727     1.24   0.214    -.1106408     .494489
         L2. |    .095686   .1481062     0.65   0.518    -.1945967    .3859688
             |
    revenues |
         L1. |  -.2275803   .1544383    -1.47   0.141    -.5302737    .0751132
         L2. |  -.4275827   .1291448    -3.31   0.001    -.6807018   -.1744635
             |
      grants |
         L1. |  -5.526232   .9128089    -6.05   0.000    -7.315304   -3.737159
         L2. |  -2.382141   .3992335    -5.97   0.000    -3.164624   -1.599658
-------------+----------------------------------------------------------------
grants       |
expenditures |
         L1. |   .0107939   .0220194     0.49   0.624    -.0323634    .0539512
         L2. |   .0176607   .0170148     1.04   0.299    -.0156878    .0510092
             |
    revenues |
         L1. |  -.0191644   .0198971    -0.96   0.335     -.058162    .0198332
         L2. |  -.0103149    .016148    -0.64   0.523    -.0419645    .0213346
             |
      grants |
         L1. |   .3128186   .0756299     4.14   0.000     .1645867    .4610504
         L2. |   .1347204   .0584898     2.30   0.021     .0200826    .2493582
------------------------------------------------------------------------------
Hansen's test of overid. restrictions: chi2(45) = 211.25   Prob > chi2 = 0.000
GMM-type instruments: L(2/.).(expenditures revenues grants)
```

The number of moment conditions after collapsing is 63, which is considerably smaller than with the previous examples. In the header of the output, the moment conditions are labeled "`(collapsed)`" to indicate that `xtvar` used a collapsed instrument matrix as we requested.

Despite the significant reduction in the number of moment conditions, we see that Hansen's J test still suggests our moment conditions are not valid. As we mentioned at the end of the previous example, the real solution is to increase the number of lags we include in the model.

◁

Lag-order selection

So far, we have explored how we can use different options to modify the model we fit and what this implies regarding the number of parameters, the instruments in our model, and Hansen's J test. Rather than fitting many models manually and comparing results, we can use the `xtvarsoc` command to help select the number of lags to satisfy the moment conditions.

▷ Example 4: Reconsidering the lag order

After fitting a model with `xtvar`, we can use `xtvarsoc` to obtain model- and moment-selection criteria (MMSC) to help determine the correct lag length in our model. MMSC are an adaption of the Akaike (1973) information criterion (AIC), Schwarz's (1978) Bayesian information criterion (BIC), and Hannan and Quinn (1979) information criterion (HQIC) for panel-data VAR models. Panel-data VARs are fit using GMM, and no distributional assumptions are made; this precludes the computation of a likelihood function and the standard information criteria.

Andrews and Lu (2001) developed variants of the AIC, BIC, and HQIC for use with dynamic panel-data models. They refer to their statistics as MMSC (rather than using lag-order selection criteria terminology used with time-series VARs) because their statistics can be used not only to select a lag length but also to select a set of instruments conditional on lag length. We will refer to the three of Andrews and Lu's statistics implemented by `xtvarsoc` as MMSC-AIC, MMSC-BIC, and MMSC-HQIC. These statistics are based on Hansen's J test statistic rather than the maximized likelihood function value, and they apply penalty factors for the sample size and number of overidentifying restrictions. Here we use `xtvarsoc` to obtain the MMSC statistics for a range of panel-data VAR models based on the instrument specification of our model with two lags and a maximum of two lags used as instruments. We will request that `xtvarsoc` consider models with up to four lags. We type

```
. xtvar expenditures revenues grants, lags(2) maxldep(2)
  (output omitted)
. xtvarsoc, maxlag(4)
Model- and moment-selection criteria
```

Lag	N	MC	Hansen's J	df	p	MMSC-AIC	MMSC-BIC	MMSC-HQIC
1	1060	72	206.44	63	0.000	80.443	-232.42*	-38.129
2	1060	72	182.98	54	0.000	74.977	-193.19	-26.656
3	1060	72	103.72	45	0.000	13.72	-209.75	-70.974
4	1060	72	38.80	36	0.345	-33.199*	-211.98	-100.95*

```
* indicates minimum value within column.
Note: Maximum lag order specified exceeds lags used at estimation; using
      reduced sample.
GMM-type instruments: L(2/3).(expenditures revenues grants)
```

`xtvarsoc` computed the MMSC for models with from one to four lags. Your first inclination may be to look at the MMSC statistics to find the best model. Before you do that, though, you need to make sure that the numbers are comparable and relevant. In particular, you should look at the sample size, number of moment conditions, and Hansen's J test. Once those check out, then you can search for the best model on the basis of the MMSC statistics.

We first verify that the sample size is the same across models. The sample size reported in each row is 1,060. Our data have 265 panels, implying $1060/265 = 4$ observations per panel. There is a note near the bottom of the output indicating `xtvarsoc` used a reduced sample; we used 2 lags in our `xtvar` specification and are using `xtvarsoc` for models with up to 4 lags. For MMSC comparisons to be valid, the statistics must be computed on the same sample of data. `xtvarsoc` first fits a model using the highest number of lags requested and then uses that model's estimation sample for models with fewer lags. That generally results in a constant sample size across lags, though we cannot rule out the possibility that an odd pattern of missing data could cause the number of observations to change across different model lags. `xtvarsoc` will issue an error message if it detects that the sample size is not constant across estimates.

When comparing models to determine the optimal lag length, you must also make sure that the same set of instruments is being used across models. The final line of the output shows we are using lags 2 and 3 of our left-hand-side variables as instruments, just as we used when fitting our original model. The column labeled `MC`, which stands for moment conditions, indicates that each of the 4 candidate models here were fit using 72 moment conditions. Our original model used 108 moment conditions, while the models fit here used 72; the difference stems from the smaller sample size here because of our request to consider a model with 4 lags.

The number of moment conditions must be the same across models with different lag lengths when choosing the number of lags. In [XT] **xtvar postestimation**, we show how to use the MMSC statistics to select the optimal set of instruments instead of the optimal number of lags. There we will require the number of lags in different models to be the same. Either the number of lags or the number of moment conditions can change across models to make valid comparisons but not both at the same time. `xtvarsoc` will exit with an error message if it detects that comparisons will not be valid.

The column marked `Hansen's J` contains Hansen's J test statistic for each model, and the columns marked `df` and `p` contain, respectively, the corresponding degrees of freedom and p values. The degrees of freedom change across models because the number of estimated parameters changes; that this column's values change does not invalidate MMSC comparisons. In this example, we reject the null hypothesis that our moment conditions are valid for models with one, two, and three lags. Based on Hansen's J test, we are left with just one model: the one with four lags.

Having verified that the sample size and the number of moment conditions remain constant across models and having tested for instrument validity, now we are ready to consider the MMSC statistics. As with standard information criteria, a model with a lower value is to be preferred over a model with a higher value. Here the MMSC-AIC and MMSC-HQIC select a model with four lags, which is fortuitous because that model is the only one for which the moment conditions are valid based on Hansen's J test. The MMSC-BIC selects a model with one lag, though we would probably reject that model because of Hansen's J test. Andrews and Lu (2001) performed Monte Carlo analyses and recommend the MMSC-BIC as the best all-around criterion to use, assuming the orthogonality conditions are valid.

◁

Including endogenous covariates

▷ Example 5: Endogenous covariates

Suppose we are primarily interested in the relationship between grants and expenditures, but we suspect that revenues are also endogenously determined. In other words, there is a dynamic relationship between grants and expenditures, while revenues have a contemporaneous effect. An alternative to fitting a three-variable panel-data VAR model is to fit a two-variable model and include revenues as an additional endogenous covariate. Below, we fit this model using the endogenous() option and again requesting that at most two lags of each variable be used as instruments.

```
. xtvar expenditures grants, lags(2) maxldep(2) endogenous(revenues)

Panel-data vector autoregression                  Number of obs     = 1,590
Group variable: idcode                            Number of groups  =   265
Time variable: year                               Obs per group:
                                                                min =     6
Number of moment conditions = 72                                avg =   6.0
                                                                max =     6

Fixed-effects transform: FOD
Two-step results
                              (Std. err. adjusted for 265 clusters in idcode)
```

	Coefficient	WC robust std. err.	z	P>|z|	[95% conf.	interval]
expenditures						
expenditures						
L1.	-.036028	.0217366	-1.66	0.097	-.0786309	.0065749
L2.	-.0580125	.0204245	-2.84	0.005	-.0980438	-.0179813
grants						
L1.	.6735404	.2300296	2.93	0.003	.2226906	1.12439
L2.	.2240255	.1219378	1.84	0.066	-.0149682	.4630192
revenues	.9932527	.0296293	33.52	0.000	.9351804	1.051325
grants						
expenditures						
L1.	-.0068636	.0076561	-0.90	0.370	-.0218693	.008142
L2.	.003658	.0068504	0.53	0.593	-.0097685	.0170846
grants						
L1.	.3318416	.0841429	3.94	0.000	.1669245	.4967586
L2.	.16419	.045025	3.65	0.000	.0759426	.2524374
revenues	.0083887	.0098906	0.85	0.396	-.0109966	.027774

```
Hansen's test of overid. restrictions: chi2(62) = 142.48   Prob > chi2 = 0.000
Added endogenous: revenues
GMM-type instruments: L(2/3).(expenditures grants revenues)
```

The coefficients on revenues are 0.99 and 0.01 in the equations for expenditures and grants, respectively. When revenues are treated as endogenous covariates, the results of this model indicate that they have a positive effect on expenditures but are not relevant for grants.

◁

Lag exclusion tests

▷ Example 6: Wald lag exclusion tests

After we fit a panel-data VAR model, one hypothesis of interest is that all the endogenous variables at a given lag are jointly zero. `varwle` reports Wald tests of this hypothesis for each equation and for all equations jointly. We refit our panel-data VAR model using four lags of the dependent variables as covariates, which is the only specification we have fit so far in which the validity of the moment conditions has not been rejected by Hansen's J test.

```
. xtvar expenditures revenues grants, lags(4) maxldep(2)
  (output omitted)
. varwle
```

Equation: expenditures

lag	chi2	df	Prob > chi2
1	6.100499	3	0.107
2	4.914952	3	0.178
3	10.01386	3	0.018
4	33.88895	3	0.000

Equation: revenues

lag	chi2	df	Prob > chi2
1	10.6371	3	0.014
2	5.669409	3	0.129
3	14.89268	3	0.002
4	21.80127	3	0.000

Equation: grants

lag	chi2	df	Prob > chi2
1	4.001778	3	0.261
2	3.773334	3	0.287
3	2.394958	3	0.495
4	46.76529	3	0.000

Equation: All

lag	chi2	df	Prob > chi2
1	22.40912	9	0.008
2	14.22638	9	0.114
3	21.47419	9	0.011
4	89.77261	9	0.000

The first block of output refers to the equation for `expenditures`. Line 1 of this output reports a Wald test for the first lag of the 3 dependent variables. Here the χ^2 statistic is 6.10 ($p \approx 0.107$), so we cannot reject the null hypothesis that the 3 coefficients are jointly 0.

The first line in the first block is equivalent to typing

```
. test [expenditures]_b[L1.expenditures]   ///
       [expenditures]_b[L1.revenues]       ///
       [expenditures]_b[L1.grants]
```

but `varwle` does all the work for us and assembles the results into nice tables. The remaining lines in the first block test the same set of coefficients on the model variables at lags two through four. The next two blocks perform similar tests for the other two equations.

The final block is arguably more useful. It tests all the coefficients of all the variables for all equations at a particular lag. Referring back to (1), each line of the final block contains a test of the hypothesis that $\mathbf{A}_\ell = \mathbf{0}$ for $\ell = 1, 2, 3,$ or 4. If you fit a model with multiple lags and find that the last lag's coefficient matrix provides evidence to support the null hypothesis, you could consider refitting your model without that lag.

What is perhaps surprising about our results is that for `expenditures` the first two lags provide evidence to support the null and similarly for the first three lags of `grants`. However, for all three variables, the fourth lag's coefficients do not provide evidence to support the null hypothesis, perhaps suggesting a slow response. In terms of the $\mathbf{A}$ coefficient matrices, we find that our estimates $\widehat{\mathbf{A}}_1$, $\widehat{\mathbf{A}}_3$, and $\widehat{\mathbf{A}}_4$ do not provide evidence to support the null hypothesis.

◁

Granger causality test

Granger causality tests are a popular tool in the VAR analyst's toolkit. Granger causality tests, in the simplest case, test whether lags of one variable are useful in predicting the values of another variable. More formally, variable x Granger-causes variable y if for all $s > 0$ the mean squared error (MSE) of a forecast of y_{t+s} based on $(y_{t-1}, y_{t-2}, \ldots, x_{t-1}, x_{t-2}, \ldots)$ is lower than a forecast using only $(y_{t-1}, y_{t-2}, \ldots)$. See Hamilton (1994, chap. 11). The extension to variables x and z jointly Granger-causing variable y is immediate.

Just as with time-series VARs, we can conduct Granger causality tests after panel-data VARs by performing Wald tests of the joint significance of the parameters associated with the lags of one or more variables in an equation. In our example, to see if `revenues` Granger-causes `expenditures`, we would test the hypothesis that the coefficients on `L1.revenues`, `L2.revenues`, `L3.revenues`, and `L4.revenues` in the equation for `expenditures` are jointly equal to zero. `vargranger` makes that, and much more, very easy.

▷ Example 7: Granger causality

Typing `vargranger` after fitting our four-lag model, we obtain

```
. vargranger
```

Granger causality Wald tests

Equation	Excluded	chi2	df	Prob > chi2
expenditures	revenues	5.0979	4	0.277
expenditures	grants	5.0767	4	0.280
expenditures	ALL	34.47	8	0.000
revenues	expenditures	9.3224	4	0.054
revenues	grants	3.4496	4	0.486
revenues	ALL	20.979	8	0.007
grants	expenditures	5.5802	4	0.233
grants	revenues	7.5613	4	0.109
grants	ALL	59.35	8	0.000

Consider the first line of output, which refers to the equation for `expenditures`. Under column `Excluded`, we see the `revenues` variable. This line of output is therefore a test of whether `revenues` Granger-causes `expenditures`. In other words, by including `revenues` in the equation for `expenditures`, do we obtain a lower forecast MSE for `expenditures` than if we had not included `revenues`? Here the χ^2 statistic is 5.10 with 4 degrees of freedom. Because the corresponding p-value is large (0.277), we cannot reject the null hypothesis that `revenues` does not Granger-causes `expenditures`. If we were building a forecast model of `expenditures`, this line of output suggest that we would not lower our forecast MSE by including `revenues`.

Similarly, the second line of output provides no evidence that `grants` Granger-causes `expenditures`, either.

The results in the third line stand in contrast to the first two lines' results. Shown in column `Excluded` is the keyword `ALL`, which represents all variables in the equation for `expenditures` other than (lags of) `expenditures` itself. This line contains a Wald test that the coefficients on the lags of both `revenues` and `grants` are all jointly equal to zero. It is a test of whether `revenues` and `grants` jointly Granger-causes `expenditures`. Here we reject the null hypothesis of no Granger causality and conclude that `revenues` and `grants` do Granger-causes `expenditures`.

These results suggest neither `revenues` nor `grants` Granger-causes `expenditures`, conditional on the other variable being in the model. However, the two variables do jointly Granger-causes `expenditures`.

The last three lines of output, pertaining to `grants`, show a similar pattern to those for `expenditures`.

◁

Verifying the stability condition of the VAR

▷ Example 8: Checking stability

For the relationships implied by our estimates to be meaningful, we need to verify that we have a stationary process. To check whether our fitted VAR represents a stable dynamic process, we use

```
. varstable, graph
```
(*output omitted*)

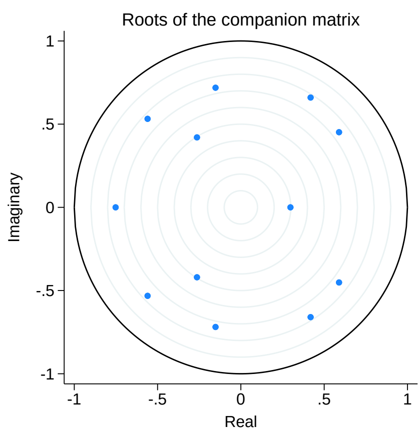


Figure 1. Checking the stability of our panel-data VAR model

Our panel-data VAR model contains 3 variables and 4 lags, so the companion matrix will have $3 \times 4 = 12$ eigenvalues, some of which are conjugate pairs. Looking at figure 1, we see that all the eigenvalues lie within the unit circle. This indicates that our panel-data VAR model does satisfy the stability condition. We could have alternatively displayed the values and modulus of the eigenvalues in a table by simply omitting the `graph` option after `varstable`.

Because our model represents a stable dynamic process, we can proceed to compute and analyze IRFs.

◁

IRFs

An IRF measures the effect across time of a shock to an endogenous variable on itself or another endogenous variable. After fitting a panel-data VAR using `xtvar`, you can obtain simple IRFs, cumulative IRFs, orthogonalized IRFs, and cumulative orthogonalized IRFs. A full description of IRFs and how you obtain them in Stata is in [TS] **irf**. Here we cover the highlights to get you started.

In Stata, an IRF set is a special dataset that contains one or more IRF results. An IRF result refers to the IRFs, cumulative IRFs, and other statistics as well as their standard errors and is obtained by calling `irf create` one time. Stata has several commands for manipulating IRF sets and results within IRF sets. For our purposes, though, we can accomplish everything we will do just by using `irf create` and specifying appropriate options.

▷ Example 9: Simple IRFs

We first create the IRF results from our current four-lag model that is still in memory, and we store them under the name `four_lags`. We will store those results in an IRF set called `example1`, and we will create the IRFs for nine steps:

```
. irf create four_lags, set(example1) step(9)
(file example1.irf created)
(file example1.irf now active)
(file example1.irf updated)
```

If IRF set `example1` already contained results stored under the name `four_lags`, the previous command would have issued an error message. If that happens and you wish to replace the existing results with that name, you can specify `replace` within the `set()` option.

With an IRF result available, we can plot the IRFs:

```
. irf graph irf, irf(four_lags)
```

When we used `irf create`, we made `irf` set `example1` active, and Stata remembered that action. Therefore, when we use the `irf graph` command without the `set()` option, we use this currently active IRF set and obtain the following graph:

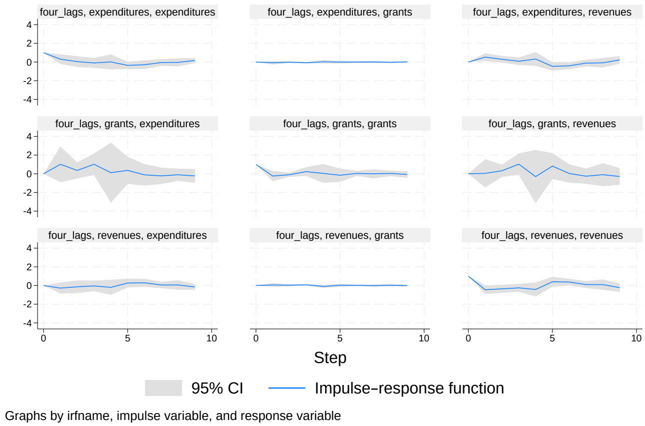


Figure 2. IRFs of our panel-data VAR model

The first row of graphs shows the effect of a shock to expenditures on expenditures themselves, grants, and revenues. In the second row, we see the effects of a shock to grants on each variable. And in the last row, we see the effects of a shock to revenues on each variable.

◁

Simple IRFs are often uninformative because they assume that a shock happens to a single variable in isolation. In the vast majority of applications, shocks affect multiple variables at the same time. One solution is to use orthogonalized IRFs, and the most common way to orthogonalize shocks is with the Cholesky decomposition of the error covariance matrix. See Hamilton (1994), Lütkepohl (2005), and Kilian and Lütkepohl (2017) for more information.

Getting Cholesky-orthogonalized IRFs after fitting a panel-data VAR requires no additional work: earlier when we called `irf create` before plotting simple IRFs, it automatically created the orthogonalized IRFs as well as the cumulative variants of both of those types of IRFs. To view the orthogonalized IRFs, we just tell `irf graph` or a similar command to plot the orthogonalized IRFs instead.

▷ Example 10: Orthogonalized IRFs

To create tables or graphs of orthogonalized IRFs, we simply change the *stat* from `irf` to `oirf` in the table or graph specification. Here we use the `irf cgraph` command to combine graphs of two of the orthogonalized IRFs of interest. We plot the effects of an orthogonalized shock to `grants` on `revenues` and `expenditures`:

```
. irf cgraph (four_lags grants revenues oirf)
>             (four_lags grants expenditures oirf)
```

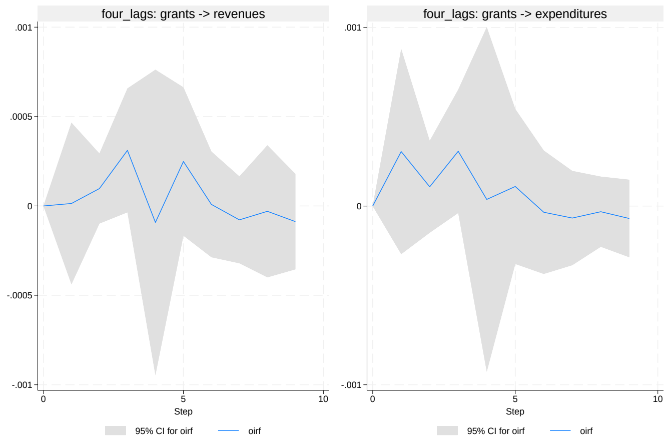


Figure 3. Orthogonalized IRFs of our panel-data VAR model

The orthogonalized IRF graphs both have wide confidence intervals that include 0. We do not have evidence that a random shock to `grants` has an effect on `revenues` or `expenditures`.

◁

Stored results

xtvar stores the following in e():

Scalars	
e(N)	number of observations
e(n_zc)	number of columns of instrument matrix
e(J)	Hansen's J statistic
e(p_J)	p-value for Hansen test
e(df_J)	degrees of freedom for Hansen test
e(rank_weight)	rank of final weight matrix
e(k)	number of parameters
e(k_eq)	number of equations
e(N_clust)	number of clusters
e(N_g)	number of groups
e(g_min)	smallest group size
e(g_max)	largest group size
e(g_avg)	average group size
e(tmin)	first period in estimation sample
e(tmax)	last period in estimation sample
e(maxldep_act)	largest lag order actually used for instruments
e(maxldep)	maximum number of instruments requested
e(minldep)	minimum number of instruments requested
e(lags)	number of lags in model
Macros	
e(cmd)	xtvar
e(cmdline)	command as typed
e(depvar)	names of dependent variables
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(winit_type)	type of initial weight matrix used
e(transform)	specified transform for removing fixed effects
e(collapse)	collapse, if specified
e(predetermined)	predetermined variables, if specified
e(endogenous)	additional endogenous variables, if specified
e(exogenous)	exogenous variables, if specified
e(onestep)	onestep, if specified
e(ivar)	group variable specified in xtset
e(tvar)	time variable specified in xtset
e(tsfmt)	display format for time variable
e(datasignaturevars)	variables used in calculation of checksum
e(datasignature)	checksum from datasignature
e(properties)	b V
e(predict)	program used to implement predict
e(eqnames)	names of equations
Matrices	
e(b)	coefficient vector
e(V)	variance–covariance matrix of the estimators
Functions	
e(sample)	marks estimation sample

In addition to the above, the following is stored in r():

Matrices	
r(table)	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

Methods and formulas are presented under the following headings:

Introduction

The panel-data VAR model with p lags, K regressands, M_1 strictly exogenous variables $\mathbf{x}_{it}$, M_2 additional endogenous variables $\mathbf{w}_{it}$, and M_3 predetermined variables $\mathbf{v}_{it}$ is

$$\mathbf{y}_{it} = \mathbf{A}_1\mathbf{y}_{i,t-1} + \mathbf{A}_2\mathbf{y}_{i,t-2} + \cdots + \mathbf{A}_p\mathbf{y}_{i,t-p} + \mathbf{B}\mathbf{x}_{it} + \mathbf{C}\mathbf{w}_{it} + \mathbf{D}\mathbf{v}_{it} + \mathbf{u}_i + \boldsymbol{\epsilon}_{it} \tag{4}$$

where i indexes panels 1 through N, t indexes time from 1 to T, $\mathbf{u}_i$ is a $K \times 1$ fixed-effect vector, and $\boldsymbol{\epsilon}_{it}$ is a $K \times 1$ vector of idiosyncratic i.i.d shocks. $\mathbf{A}_1, \ldots, \mathbf{A}_p$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ are parameter matrices to be estimated.

Fully exogenous variables are uncorrelated with past, present, or future realizations of the error term. Therefore, they satisfy the condition $E\{\text{vec}(\mathbf{x}_{it}\boldsymbol{\epsilon}'_{is})\} = \mathbf{0}$ for all s and t.

Endogenous variables, on the other hand, are not correlated with future realizations of the error term but may be correlated with present or past realizations of it. This means that $E\{\text{vec}(\mathbf{w}_{it}\boldsymbol{\epsilon}'_{is})\} = \mathbf{0}$ for $s > t$, but possibly $E\{\text{vec}(\mathbf{w}_{it}\boldsymbol{\epsilon}'_{is})\} \neq \mathbf{0}$ for $s \leq t$.

Predetermined variables represent an intermediate case. They are not correlated with future or present realizations of the error term but may be correlated with past realizations of it. Therefore, they satisfy $E\{\text{vec}(\mathbf{v}_{it}\boldsymbol{\epsilon}'_{is})\} = \mathbf{0}$ for $s \geq t$, but possibly $E\{\text{vec}(\mathbf{v}_{it}\boldsymbol{\epsilon}'_{is})\} \neq \mathbf{0}$ for $s < t$.

Eliminating the fixed effect

The first task in estimating the panel-data VAR model is to eliminate the fixed effect term $\mathbf{u}_i$ using either the FD transformation or the FOD transformation.

For an arbitrary variable h_{it}, the FD transformation is

$$h_{it}^{\text{FD}} = h_{it} - h_{i,t-1}$$

for $t = 2, \ldots, T$. Note that, if h_{it} is missing, then both h_{it}^{FD} and $h_{i,t+1}^{\text{FD}}$ will be missing.

The FOD transformation, introduced by Arellano and Bover (1995), is defined as follows. Let $\Omega(i,t)$ denote the set of periods for which $h_{i\tau}$ is nonmissing for $\tau = t+1,\dots,T$. Let $\#_{\Omega(i,t)}$ denote the number of elements in set $\Omega(i,t)$. Then

$$h_{it}^{\text{FOD}} = \sqrt{\frac{\#_{\Omega(i,t)}}{\#_{\Omega(i,t)}+1}}\left(h_{it} - \frac{1}{\#_{\Omega(i,t)}}\sum_{\tau\in\Omega(i,t)} h_{i\tau}\right)$$

and exists for $t = 1,\dots,T-1$. To make the formulas provided below work regardless of which transform is used, we use the common practice of storing the value h_{it}^{FOD} in period $t+1$.

We adopt the tilde diacritic to represent a variable that has undergone one of these transformations: $\widetilde{h}_{it} = h_{it}^{\text{FD}}$ or $\widetilde{h}_{it} = h_{it}^{\text{FOD}}$ depending on whether the `fd` or `fod` option was specified with `xtvar`. For vectors like $\mathbf{y}_{it}$ or $\mathbf{x}_{it}$, we apply the transform to each element within the vector individually.

Applying either the FD or the FOD transformation has the effect of eliminating $\mathbf{u}_i$ from (4) as follows:

$$\tilde{\mathbf{y}}_{it} = \mathbf{A}_1\tilde{\mathbf{y}}_{i,t-1} + \mathbf{A}_2\tilde{\mathbf{y}}_{i,t-2} + \cdots + \mathbf{A}_p\tilde{\mathbf{y}}_{i,t-p} + \mathbf{B}\tilde{\mathbf{x}}_{it} + \mathbf{C}\widetilde{\mathbf{w}}_{it} + \mathbf{D}\tilde{\mathbf{v}}_{it} + \tilde{\boldsymbol{\epsilon}}_{it}$$

The FOD transformation is often preferred over the FD transformation because it usually results in fewer observations being dropped because of gaps. The following data illustrate this:

	t	x	x_fd	x_fod
1.	1	1	.	.
2.	2	2	1	-1.5652476
3.	3	.	.	-.8660254
4.	4	4	.	.
5.	5	5	1	-.70710678

We have five periods `t`, and variable `x` runs from 1 to 5 but is missing when `t` = 3. Variable `x_fd` represents the FD transformation applied to `x`. Period 1 is missing because we do not have `x` at `t` = 0. `x_fd` is also missing for both periods `t` = 3 and `t` = 4 because `x` is missing at `t` = 3. Now consider variable `x_fod`, the FOD transform of `x`. Period `t` = 1 is again missing, but the gap in `x` at `t` = 3 causes us to lose only one observation for `x_fod` at `t` = 4.

Besides the greater number of observations, Hayakawa (2009) provides Monte Carlo evidence showing that the GMM estimator with the FOD transformation yields an estimate of the autoregression parameter with lower bias than the estimator with the FD transformation in the single equation case.

Constructing the instrument matrix

The second task in estimating the panel-data VAR model is to construct the instrument matrix Z_i for the transformed variables.

To simplify the exposition, suppose first that the only covariates in the model are the lags of the dependent variables. Therefore, after applying either the FOD or the FD transformation, we have

$$\tilde{\mathbf{y}}_{it} = \mathbf{A}_1\tilde{\mathbf{y}}_{i,t-1} + \mathbf{A}_2\tilde{\mathbf{y}}_{i,t-2} + \cdots + \mathbf{A}_p\tilde{\mathbf{y}}_{i,t-p} + \tilde{\boldsymbol{\epsilon}}_{it}$$

Note that each matrix of coefficients in the model is of dimension $K \times K$, and therefore there are pK^2 parameters to estimate in total.

To estimate these parameters consistently, we need to instrument the endogenous regressors with the lags of the dependent variables. Start with endogenous regressor $\tilde{\mathbf{y}}_{i,t-1}$. Given the assumptions in the model, we can instrument this endogenous regressor using $\mathbf{z}_{it} = [\mathbf{y}'_{i,t-2}, \mathbf{y}'_{i,t-3}, \ldots, \mathbf{y}'_{i1}]'$ as instruments. This implies that the moment conditions associated with this endogenous regressor are given by $E\{\text{vec}(\mathbf{z}_{it}\tilde{\boldsymbol{\epsilon}}'_{it})\} = \mathbf{0}$. Because vector $\mathbf{z}_{it}$ has $R_t = K(t-2)$ rows and the error vector $\tilde{\boldsymbol{\epsilon}}_{it}$ has K elements, we have a total of $KR_t = K^2(t-2)$ moment conditions associated with endogenous regressors $\tilde{\mathbf{y}}_{i,t-1}$. For the next endogenous regressor, $\tilde{\mathbf{y}}_{i,t-2}$, we proceed similarly. We use as instruments $\mathbf{z}_{i,t-1} = [\mathbf{y}'_{i,t-3}, \mathbf{y}'_{i,t-4}, \ldots, \mathbf{y}'_{i1}]'$ and thus obtain $KR_{t-1} = K^2(t-3)$ moment conditions. Proceeding in the same way for all the endogenous regressors, we obtain a total of $\sum_{t=p+2}^{T} KR_t = (K^2/2)(T-p-1)(T+p-2)$ moment conditions.

In tables 1a, 1b, and 1c, we show the number of parameters, instruments, and moment conditions for models with a varying number of variables, lags, and panel sizes. As you can see, even when the dataset has just 7 observations per panel, the number of moment conditions can become unwieldy, especially as the number of variables in the model increases. When the dataset has 10 or more observations per panel, tables 1a, 1b, and 1c lay bare the need to control the number of instruments.

Table 1a. Parameters, instruments, and moment conditions for $K = 2$

	$T=5$		$T=7$			$T=10$			$T=15$		
Lag order p	2	3	2	3	4	2	3	4	2	3	4
Parameters	8	12	8	12	16	8	12	16	8	12	16
Instruments	10	6	28	24	19	70	66	60	180	176	170
Moments	20	12	56	48	36	140	132	120	360	352	340

Table 1b. Parameters, instruments, and moment conditions for $K = 3$

	$T=5$		$T=7$			$T=10$			$T=15$		
Lag order p	2	3	2	3	4	2	3	4	2	3	4
Parameters	18	27	18	27	36	18	27	36	18	27	36
Instruments	15	9	42	36	27	105	99	90	270	264	255
Moments	45	27	126	108	81	315	297	270	810	792	765

Table 1c. Parameters, instruments, and moment conditions for $K = 4$

	$T=5$		$T=7$			$T=10$			$T=15$		
Lag order p	2	3	2	3	4	2	3	4	2	3	4
Parameters	32	48	32	48	64	32	48	64	32	48	64
Instruments	20	12	56	48	36	140	132	120	360	352	340
Moments	80	48	224	192	144	560	528	480	1,440	1,408	1,360

The panel-data VAR estimator is implemented at the panel level. Therefore, our next step is to consider the appropriate instrument matrix for an entire panel i.

Define $\boldsymbol{\xi}_i \equiv [\widetilde{\boldsymbol{\epsilon}}_{i,p+2}, \widetilde{\boldsymbol{\epsilon}}_{i,p+3}, \ldots, \widetilde{\boldsymbol{\epsilon}}_{iT}]'$ as the $(T-p-1) \times K$ matrix of errors for panel i. The $T-p-1$ rows of the matrix correspond to the number of periods at which we can evaluate the model after transforming our variables. The K columns correspond to the errors for each of the K equations. Thus, we can write the moment conditions for an entire panel as $E\{\text{vec}(\mathbf{Z}_i'\boldsymbol{\xi}_i)\} = \mathbf{0}$, where we define $\mathbf{Z}_i$ as the block-diagonal matrix

$$\mathbf{Z}_i = \begin{pmatrix} \mathbf{z}'_{i,p+2} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{z}'_{i,p+3} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{z}'_{i,p+4} & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{z}'_{iT} \end{pmatrix} \tag{5}$$

and where $\mathbf{0}$ represents a vector of 0s, the length of which is implied by its position in the matrix. Definition (5) makes $\mathbf{Z}_i$ look relatively compact, but it conceals just how many columns it contains—and hence how many moment conditions our estimator will use. We denote the number of rows of $\mathbf{Z}$ as Z_r and the number of columns as Z_c.

To get an idea of how large $\mathbf{Z}_i$ can be, let us look at $\mathbf{Z}_i$ for a bivariate panel-data VAR(2) model where we use first differencing to remove the fixed effects and further assume that $T = 6$. The first period at which we can evaluate the model is $t = 4$, so the matrix $\mathbf{Z}_i$ will have $Z_r = (6-4+1) = 3$ rows. Recalling that we used R_t to denote the number of rows in $\mathbf{z}_{it}$, we will obtain $Z_c = R_4 + R_5 + R_6 = 18$ columns. Then

$$\mathbf{Z}_i = \begin{pmatrix} \mathbf{y}'_{i2} & \mathbf{y}'_{i1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{y}'_{i3} & \mathbf{y}'_{i2} & \mathbf{y}'_{i1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{y}'_{i4} & \mathbf{y}'_{i3} & \mathbf{y}'_{i2} & \mathbf{y}'_{i1} \end{pmatrix} \tag{6}$$

With these definitions, $E\{\text{vec}(\mathbf{Z}_i'\boldsymbol{\xi}_i)\} = \mathbf{0}$ is equivalent to $E\{\text{vec}(\mathbf{z}_{it}\widetilde{\boldsymbol{\epsilon}}'_{it})\} = \mathbf{0}$ for $t = 4$, 5, and 6. There are a total of $KZ_c = 2 \times 18 = 36$ scalar moment conditions.

Returning to the general case, the number of rows of $\mathbf{Z}_i$ is $Z_r = T - p - 1$, the number of periods at which we can evaluate the model. The number of columns depends on the panel dimension T, the model's lag order p, and K, the number of variables that make up $\mathbf{y}_{it}$. This works out to be $Z_c = \sum_{t=p+2}^{T} R_t = (K/2)(T-p-1)(T+p-2)$ columns. The total number of moment conditions therefore equals $KZ_c = (K^2/2)(T-p-1)(T+p-2)$. Notice that the number of instruments is quadratic in T. Adding more observations to each panel causes the number of instruments to grow rapidly.

`xtvar` provides two ways to control the number of instruments and hence moment conditions in a panel-data VAR model. One method controls the number of lags of the dependent variables that are used as instruments (see *Restricting instrument lags*). The other method, which has come to be known as "collapsing", results in a GMM estimator that replaces the moment conditions $E\{\text{vec}(\mathbf{z}_{it}\widetilde{\boldsymbol{\epsilon}}'_{it})\} = 0$ with an alternative that averages those moment conditions over time (see *Collapsing the instrument matrix*).

Dealing with gaps and missing data

Here we show how `xtvar` creates $\mathbf{Z}_i$ when there are gaps in the data. Gaps pose two problems: first, the regression equations cannot be evaluated when either left-hand-side or right-hand-side variables are missing; and second, gaps reduce the number of available instruments. Whether the gaps are caused by some periods being absent from the data or because some variables contain missing values, the solution in both cases is essentially a strategic placement of zeros in $\mathbf{Z}_i$.

Consider data in which we have three panels and six time periods. Suppose we want to fit a VAR(1) model with two variables, y1 and y2, using these data.

	id	t	y1	y2
1.	1	1	1	2
2.	1	2	2	3
3.	1	3	3	6
4.	1	4	4	4
5.	1	5	5	7
6.	1	6	6	9
7.	2	1	1	1
8.	2	2	2	6
9.	2	4	4	7
10.	2	5	5	6
11.	2	6	6	8
12.	3	1	1	3
13.	3	2	2	5
14.	3	3	3	5
15.	3	4	4	6
16.	3	5	5	5
17.	3	6	.a	.b

The first panel is balanced; we have valid data for all six time periods. The second panel has a gap: no data are recorded for $t = 3$. Finally, the third panel contains missing values for y1 and y2 at $t = 6$.

The maximum time period in our data is $T = 6$, and we are considering a panel VAR model with one lag. Therefore, each $\mathbf{Z}_i$ will have 4 rows, corresponding to $t = 3$, 4, 5, and 6. The number of columns is $(K/2)(T - p - 1)(T + p - 2) = 1(6 - 2)(6 - 1) = 20$. $\mathbf{Z}_1$ will be just as (5) shows,

$$\mathbf{Z}_1 = \begin{pmatrix} 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 3 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 6 & 2 & 3 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & 3 & 6 & 2 & 3 & 1 & 2 \end{pmatrix}$$

Turning to the second panel, that y1 and y2 are missing for $t = 3$ implies that we will have fewer available instruments in periods $t = 5$ and $t = 6$. The missing values also affect the time periods at which we can evaluate the regression equations, and if we cannot evaluate the regression equations for a certain period, we make the corresponding row in $\mathbf{Z}_i$ contain all zeros. After applying the FOD transformation to y1 and y2, our data for the second panel look like this:

	id	t	y1	y2	fod_y1	fod_y2
1.	2	1	1	1	.	.
2.	2	2	2	6	-4.011887	-6.326437
3.	2	4	4	7	.	.
4.	2	5	5	6	-2.236068	-1.341641
5.	2	6	6	8	-1.732051	-2.886751

After taking the FOD transform, we cannot evaluate the regression equations for $t = 3$, 4, or 5. We can at least evaluate them when $t = 6$, though in that period we cannot use the values of y1 and y2 at $t = 3$ as instruments, and we must put 0s where they would go if they were not missing. We have

$$\mathbf{Z}_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 7 & 0 & 0 & 2 & 6 & 1 & 1 \end{pmatrix}$$

Finally, when a gap occurs at the end of the panel, as in our third panel, we cannot evaluate the regression equations for the last period, so we put a vector of 0s in the row of $\mathbf{Z}_3$ corresponding to $t = 6$. Earlier periods, however, are not affected and so we have

$$\mathbf{Z}_3 = \begin{pmatrix} 1 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 5 & 1 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 5 & 2 & 5 & 1 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Missing values and gaps cause more problems for panel-data VARs than for many other estimators. The value at time t is used at a minimum for both the regression equation for time t and $t+1$, because we use lags as regressors; in short, a missing value for one time period could easily result in many observations being omitted from the regression equation. To make matters worse, the FOD and FD transformations can result in the loss of additional periods from our regression equation. Finally, when we have missing values and gaps, we need to consider how they affect the instruments available for later periods as well. xtvar does its best to accommodate gaps, but the nature of the estimator limits the tolerance of gaps.

Restricting instrument lags

One way to reduce the dimensionality of $\mathbf{Z}_i$ is to use a fixed number of lags to instrument each endogenous regressor, rather than using all the available lags. If we use a maximum of $l^{\max}$ lags of the dependent variables as instruments for each endogenous regressor, then the panel-level instrument matrix $\mathbf{Z}_i$ will still have $Z_r = T - p - 1$ rows. However, now the number of columns is at most $Z_c = Kl^{\max}(T - p - 1)$, which is linear in T: for each time period we add, the number of instruments increases by just $Kl^{\max}$.

Table 2a. Using two lags as instruments with $K = 2$

	$T=5$		$T=7$			$T=10$			$T=15$		
Lag order p	2	3	2	3	4	2	3	4	2	3	4
Parameters	8	12	8	12	16	8	12	16	8	12	16
Instruments	8	4	16	12	8	28	24	20	48	44	40
Moments	16	8	32	24	16	56	48	40	96	88	80

Table 2b. Using two lags as instruments with $K = 3$

	$T=5$		$T=7$			$T=10$			$T=15$		
Lag order p	2	3	2	3	4	2	3	4	2	3	4
Parameters	18	27	18	27	36	18	27	36	18	27	36
Instruments	12	6	24	18	12	42	36	30	72	66	60
Moments	36	18	72	54	36	126	108	90	216	198	180

Table 2c. Using two lags as instruments with $K = 4$

	$T=5$		$T=7$			$T=10$			$T=15$		
Lag order p	2	3	2	3	4	2	3	4	2	3	4
Parameters	32	48	32	48	64	32	48	64	32	48	64
Instruments	16	8	32	24	16	56	48	40	96	88	80
Moments	64	32	128	96	64	224	192	160	384	352	320

In tables 2a, 2b, and 2c, we show the number of parameters, instruments, and moment conditions for the same constellation of models as in tables 1a, 1b, and 1c, except that here we use just two lags of the dependent variables as instruments. We have certainly made progress limiting the proliferation of instruments. Models with three or four variables and just two or three lags still have many moment conditions in comparison with the number of parameters; but we have only one case in which the ratio exceeds 10. When $T = 5$, we can no longer fit panel-data VAR(3) models when we use $l^{\max} = 2$ because we do not have enough instruments. In these cases, though, even when using all available instruments, the models are just identified, so Hansen's J statistic would not be available to test the validity of the moment conditions. If we do limit ourselves to just two moment conditions, then the panel-data VAR(4) model is just identified when $T = 7$.

You use the `maxldep()` option with `xtvar` to limit the number of lags of $\mathbf{y}_{it}$ to use as instruments. For example, if you specify `maxldep(3)`, then `xtvar` will use $\mathbf{y}_{i,t-2}$, $\mathbf{y}_{i,t-3}$, and $\mathbf{y}_{i,t-4}$ as instruments.

Collapsing the instrument matrix

The second method provided by `xtvar` to control the number of moment conditions has come to be known as "collapsing" the instrument matrix, and it amounts to taking averages of individual moment conditions across time. You can request this method by specifying the `collapse` option with `xtvar`. Roodman (2009b) introduced the concept of collapsing in his community-contributed `xtabond2` command, and he mentions several other authors who have implemented this method of reducing the number of moment conditions.

Consider the $\mathbf{Z}_i$ matrix shown in (6) for a bivariate panel-data VAR(2) model with $T = 6$. When we collapse an instrument matrix, we first move the leading $\mathbf{0}$ vectors in each row to the end of the row, and then we remove columns consisting of all 0s. The matrix $\mathbf{Z}_i$ in that example becomes

$$\mathbf{Z}_i^C = \begin{pmatrix} \mathbf{y}'_{i2} & \mathbf{y}'_{i1} & \mathbf{0} & \mathbf{0} \\ \mathbf{y}'_{i3} & \mathbf{y}'_{i2} & \mathbf{y}'_{i1} & \mathbf{0} \\ \mathbf{y}'_{i4} & \mathbf{y}'_{i3} & \mathbf{y}'_{i2} & \mathbf{y}'_{i1} \end{pmatrix} \tag{7}$$

The number of rows still equals 3 because $t = 4$ is the first period in which we can evaluate a panel-data VAR(2) model and $T = 6$. However, now the number of columns equals $K(6-2) = 4K$ because at period $T = 6$ we have 4 available lags to use as instruments in that period.

When the "collapsing" procedure is applied to a general K-variable panel VAR(p) model with T observations per panel, the number of rows remains the same, but the number of columns reduces to $Z_c^C = K(T-p)$. As with the case where we limit the number of lags used as instruments, by collapsing we again have an instrument matrix for which the number of columns is linear in T rather than quadratic.

When we collapse the instrument matrix, we are changing the moment conditions used to identify the parameters. In the full panel-data VAR(p) model, we wrote the moment conditions as $E\{\text{vec}(\mathbf{z}_{it}\widetilde{\boldsymbol{\epsilon}}'_{it})\} = \mathbf{0}$ or, equivalently, $E(y_{i,\tau}^{(k)}\widetilde{\epsilon}_{it}^{(k')}) = 0$ for all k and k' from 1 to K and for all $\tau < t-1$. Each dependent variable lagged two periods or more is hypothesized to be uncorrelated with the time-t transformed error term.

Examining (7) in the context of the panel-level moment conditions $E(\mathbf{Z}_i^C\widetilde{\boldsymbol{\epsilon}}_t) = \mathbf{0}$ reveals that our collapsed instrument matrix implies something different. Performing the necessary multiplication, we see that the first K moment conditions can be written as $E(\sum_{t=p+2}^{T} y_{i,t-2}^{(k)}\epsilon_{i,t}^{(k')}) = 0$ for all k and k' from 1 to K. The total number of moment conditions when we use $\mathbf{Z}_i^C$ rather than $\mathbf{Z}_i$ is $KZ_c^C = K^2(T-2)$.

With the uncollapsed instrument matrix, the GMM estimator tries to make each term within that summation as close to zero as possible. Because $E(\widetilde{\boldsymbol{\epsilon}}_i) = \mathbf{0}$, our collapsed moment conditions still imply the terms within that summation have expectation zero. However, now we only ask the GMM estimator only to make sums (averages) across those terms as close to zero as possible. In that sense, the moment conditions with the collapsed instrument matrix are "weaker" because having an average of terms equal to zero is much easier than having each term equal to zero.

Adding other covariates

`xtvar` allows three types of additional covariates. Strictly exogenous regressors $\mathbf{x}_{it}$ are uncorrelated with past, present, and future realizations of the error term. Other regressors $\mathbf{w}_{it}$ may be endogenous, meaning that they are correlated with past and present realizations of the error. `xtvar` also accommodates an intermediate class of variables $\mathbf{v}_{it}$ that is predetermined, meaning that the current value of $\mathbf{v}_{it}$ is affected by errors from previous time periods but not errors from the current period.

Exogenous regressors

Letting the $M_1 \times 1$ vector $\mathbf{x}_{it}$ represent strictly exogenous variables, we can write our panel-data VAR model with such covariates as

$$\mathbf{y}_{it} = \mathbf{A}_1\mathbf{y}_{i,t-1} + \mathbf{A}_2\mathbf{y}_{i,t-2} + \cdots + \mathbf{A}_p\mathbf{y}_{i,t-p} + \mathbf{B}\mathbf{x}_{it} + \mathbf{u}_i + \boldsymbol{\epsilon}_{it} \tag{8}$$

Strictly exogenous variables are uncorrelated with past, present, or future realizations of the error term. They satisfy the condition $E\{\text{vec}(\mathbf{x}_{it}\boldsymbol{\epsilon}'_{is})\} = \mathbf{0}$ for all s and t. After applying the FD or FOD transform to remove the fixed-effect term from (8), we can use the transformed variant of $\mathbf{x}_{it}$ as its own instrument. The full-instrument matrix $\mathbf{Z}_i$ analogous to the one shown in (5) is now

$$\mathbf{Z}_i = \begin{pmatrix} \mathbf{z}'_{i,p+2} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \tilde{\mathbf{x}}'_{i,p+2} \\ \mathbf{0} & \mathbf{z}'_{i,p+3} & \mathbf{0} & \dots & \mathbf{0} & \tilde{\mathbf{x}}'_{i,p+3} \\ \mathbf{0} & \mathbf{0} & \mathbf{z}'_{i,p+4} & \dots & \mathbf{0} & \tilde{\mathbf{x}}'_{i,p+4} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{z}'_{iT} & \tilde{\mathbf{x}}'_{iT} \end{pmatrix} \tag{9}$$

Including $\mathbf{x}_{it}$ adds M_1 columns to $\mathbf{Z}_i$ regardless of K, p, or T.

Endogenous regressors

Endogenous regressors affect $\mathbf{y}_{it}$, and your theory implies that those variables may be correlated with the present and past realizations of the error term. These variables are represented by $M_2 \times 1$ vector $\mathbf{w}_{it}$, and we have that $E\{\text{vec}(\mathbf{w}_{it}\boldsymbol{\epsilon}'_{is})\} \neq \mathbf{0}$ for $s \leq t$. Under this definition, $\mathbf{w}_{it}$ and $\boldsymbol{\epsilon}_{it}$ may be contemporaneously correlated, and past periods' errors may affect $\mathbf{w}_{it}$ as well. Adding these variables to our model, we have

$$\mathbf{y}_{it} = \mathbf{A}_1\mathbf{y}_{i,t-1} + \mathbf{A}_2\mathbf{y}_{i,t-2} + \cdots + \mathbf{A}_p\mathbf{y}_{i,t-p} + \mathbf{B}\mathbf{x}_{it} + \mathbf{C}\mathbf{w}_{it} + \mathbf{u}_i + \boldsymbol{\epsilon}_{it}$$

Predetermined regressors

Predetermined regressors are not correlated with the present period's error term but are correlated with past realizations of the error term. We represent these regressors with the $M_3 \times 1$ vector $\mathbf{v}_{it}$, and we have that $E\{\text{vec}(\mathbf{v}_{it}\boldsymbol{\epsilon}'_{is})\} \neq \mathbf{0}$ for $s < t$. Including these variables completes our model:

$$\mathbf{y}_{it} = \mathbf{A}_1\mathbf{y}_{i,t-1} + \mathbf{A}_2\mathbf{y}_{i,t-2} + \cdots + \mathbf{A}_p\mathbf{y}_{i,t-p} + \mathbf{B}\mathbf{x}_{it} + \mathbf{C}\mathbf{w}_{it} + \mathbf{D}\mathbf{v}_{it} + \mathbf{u}_i + \boldsymbol{\epsilon}_{it}$$

By taking the first difference of this equation, you will see that the transformed regressor $\mathbf{v}_{it} - \mathbf{v}_{i,t-1}$ will be correlated with the transformed error term $\tilde{\boldsymbol{\epsilon}}_{it} = \boldsymbol{\epsilon}_{it} - \tilde{\boldsymbol{\epsilon}}_{i,t-1}$. However, $\mathbf{v}_{i,t-1}$ is a valid instrument for the transformed regressor because $\mathbf{v}_{i,t-1}$ is not correlated with $\boldsymbol{\epsilon}_{i,t-1}$. In addition, $\mathbf{v}_{i,t-2}, \mathbf{v}_{i,t-3}, \dots$ are valid instruments. We can continue to use the definition of $\mathbf{Z}_i$ as in (5) or (9) if we again redefine $\mathbf{z}_{it}$. Now we have

$$\mathbf{z}_{it} = \left[\mathbf{y}'_{i,t-2}, \mathbf{y}'_{i,t-3}, \dots \mathbf{y}'_{i1}, \dots \mathbf{w}'_{i,t-2}, \mathbf{w}'_{i,t-3}, \dots \mathbf{w}'_{i1}, \mathbf{v}'_{i,t-1}, \mathbf{v}'_{i,t-2}, \dots \mathbf{v}'_{i1}\right]'$$

We have more lags to use as instruments for predetermined variables versus endogenous ones; each additional predetermined variable adds $(1/2)(T-p-1)(T+p)$ columns to $\mathbf{Z}_i$.

When you specify the `maxldep(#)` option to limit the number of lags of the dependent variables to use as instruments, the number of lags of predetermined variables used as instruments will equal # + 1 because $\mathbf{v}_{i,t-1}$ is a valid instrument while $\mathbf{y}_{i,t-1}$ and $\mathbf{w}_{i,t-1}$ are not.

The number of instruments revisited

Earlier in *Constructing the instrument matrix*, we showed that for a panel-data VAR with K dependent variables and p lags and no additional covariates, the number of columns in $\mathbf{Z}_i$ was given by

$$Z_c = \text{cols}\left(\mathbf{Z}_i\right) = \frac{K}{2}\left(T-p-1\right)\left(T+p-2\right)$$

unless you use `collapse`, `maxldep()`, or `minldep()`.

Taking stock of the full model with additional covariates, we see $\mathbf{Z}_i$ can have many columns, as shown below,

$$\begin{aligned}
\mathbf{Z}_c &= \underbrace{\frac{K}{2}\left(T-p-1\right)\left(T+p-2\right)}_{\text{Lags of } \mathbf{y}_{\text{it}}} + \underbrace{M_1}_{\text{cols}(\mathbf{x}_{\text{it}})} \\
&\quad + \underbrace{\frac{M_2}{2}\left(T-p-1\right)\left(T+p-2\right)}_{\text{Lags of } \mathbf{w}_{\text{it}}} + \underbrace{\frac{M_3}{2}\left(T-p-1\right)\left(T+p\right)}_{\text{Lags of } \mathbf{v}_{\text{it}}} \\
&= \frac{M_2+K}{2}\left(T-p-1\right)\left(T+p-2\right) + \frac{M_3}{2}\left(T-p-1\right)\left(T+p\right) + M_1 \qquad (10)
\end{aligned}$$

and the number of moment conditions is KZ_c. When a model includes additional endogenous or predetermined variables, you likely need to specify the `maxldep()` or `collapse` option to keep the number of instruments and moment conditions manageable. If you specify the `maxldep()` option, then the number of columns of $\mathbf{Z}_i$ is limited to

$$Z_c = M_1 + \left\{l^{\max}\left(K+M_2+M_3\right)+M_3\right\}\left(T-p-1\right)$$

The `maxldep()` option reduces the growth rate of instruments from quadratic to linear, which is often the difference between being able to fit a model and not being able to fit it.

A concise representation of the GMM estimator

We now write our model at the panel level as

$$\boldsymbol{y}_i = \boldsymbol{X}_i\boldsymbol{\theta} + \boldsymbol{e}_i \qquad (11)$$

for regressand vector $\boldsymbol{y}_i$, regressor matrix $\boldsymbol{X}_i$, parameter vector $\boldsymbol{\theta}$, and residual vector $\boldsymbol{e}_i$. Additionally, we write the instrument matrix of the model as $\boldsymbol{Z}_i$. The definition of these matrices is given below. The model in (11) does not contain the panel-level fixed effects because all the matrices in the equation are defined in terms of transformed variables.

Writing the model as such simplifies deriving our estimator because, after all, (11) is just a linear regression equation we are to fit using GMM. Therefore, all the usual single-equation results regarding the consistency and asymptotic normality of GMM estimators apply in our case as long as our VAR process is stationary.

Before we can define the matrices used in (11), we need to create some other matrices. For each matrix composed of variables, the rows of the matrix will represent the periods in which we can evaluate our regression equation.

Define the $(T-p-1)\times K$ matrix

$$\widetilde{\boldsymbol{Y}}_i = [\tilde{\mathbf{y}}_{i,p+2}, \tilde{\mathbf{y}}_{i,p+2}, \ldots, \tilde{\mathbf{y}}_{i,T}]$$

Earlier, we had defined

$$\boldsymbol{\xi}_i \equiv [\tilde{\boldsymbol{\epsilon}}_{i,p+2}, \tilde{\boldsymbol{\epsilon}}_{i,p+3}, \ldots, \tilde{\boldsymbol{\epsilon}}_{iT}]'$$

as the $(T-p-1)\times K$ matrix of residuals, and we retain that definition here.

Define

$$\widetilde{\mathbf{X}}_i = \begin{pmatrix} \tilde{\mathbf{y}}'_{i,p+1} & \widetilde{\mathbf{y}'}_{i,p} & \cdots & \widetilde{\mathbf{y}'}_{i,2} & \widetilde{\mathbf{x}'}_{i,p+2} & \widetilde{\mathbf{w}'}_{i,p+2} & \tilde{\mathbf{v}}'_{i,p+2} \\ \tilde{\mathbf{y}}'_{i,p+2} & \widetilde{\mathbf{y}'}_{i,p+1} & \cdots & \widetilde{\mathbf{y}'}_{i,3} & \widetilde{\mathbf{x}'}_{i,p+3} & \widetilde{\mathbf{w}'}_{i,p+3} & \tilde{\mathbf{v}}'_{i,p+3} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \tilde{\mathbf{y}}'_{i,T-1} & \widetilde{\mathbf{y}'}_{i,T-2} & \cdots & \widetilde{\mathbf{y}'}_{i,T-p} & \widetilde{\mathbf{x}'}_{i,p+3} & \widetilde{\mathbf{w}'}_{i,p+3} & \tilde{\mathbf{v}}'_{i,p+3} \end{pmatrix}$$

with dimensions $(T-p-1)\times(pK+M)$, where $M = M1 + M2 + M3$.

By concatenating the individual parameter matrices horizontally, define

$$\boldsymbol{\Theta} = \left[\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_p, \mathbf{B}, \mathbf{C}, \mathbf{D}\right]$$

Matrix $\boldsymbol{\Theta}$ has dimensions $K\times(pK+M)$.

Finally, the instrument matrix $\mathbf{Z}_i$ is defined in (9) using the $\mathbf{z}_{it}$ defined in (12); its dimensions are $(T-p-1)\times Z_c$, where Z_c is the long expression in (10).

Now we can define the matrices in (11) as follows: $\boldsymbol{y}_i = \text{vec}(\widetilde{\boldsymbol{Y}}_i)$ is the $K(T-p-1)\times 1$ regressand vector; $\boldsymbol{X}_i = \widetilde{\boldsymbol{X}}_i \otimes I_K$ is the $K(T-p-1)\times K(pK+M)$ regressor matrix; $\boldsymbol{\theta} = \text{vec}(\boldsymbol{\Theta})$ is the $K(pK+M)\times 1$ vector of parameters; and $\boldsymbol{e}_i = \text{vec}(\boldsymbol{\xi}_i)$ is the $K(T-p-1)\times 1$ vector of errors. Additionally, the $K(T-p-1)\times KZ_c$ instrument matrix is given by $\boldsymbol{Z}_i = \mathbf{Z}_i \otimes I_K$.

Finally, we define

$$\boldsymbol{Q}_{Z'X} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{Z}_i'\boldsymbol{X}_i$$

and

$$\boldsymbol{Q}_{Z'y} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{Z}_i'\boldsymbol{y}_i$$

Estimators

`xtvar` estimates the parameters of model (4) using the GMM. By default, `xtvar` uses the two-step GMM estimator unless you specify the `onestep` option, in which case the one-step GMM estimator is used. The two-step estimator makes use of results from the one-step estimator, so we describe the one-step estimator first.

One-step estimator

Equation (11) is a single-equation linear regression model with endogenous covariate $\boldsymbol{X}_i$ and corresponding instrument matrix $\mathbf{Z}_i$, so the one-step estimator $\widehat{\boldsymbol{\theta}}_1$ is

$$\widehat{\boldsymbol{\theta}}_1 = \left(\boldsymbol{Q}'_{Z'X}\mathbf{W}_0\boldsymbol{Q}_{Z'X}\right)^{-1}\boldsymbol{Q}'_{Z'X}\mathbf{W}_0\boldsymbol{Q}_{Z'y}$$

where $\mathbf{W}_0$ is an initial weight matrix with dimensions $KZ_c \times KZ_c$, which we describe next.

You control the initial weight matrix $\mathbf{W}_0$ with the `winitial()` option. If you specify `winitial(identity)` then `xtvar` sets $\mathbf{W}_0$ to the identity matrix of appropriate size. We generally do not recommend using the identity matrix as the initial weight matrix for the GMM estimator, as better alternatives exist; we include it as an option here for compatibility reasons.

Alternatively, you may specify `winitial(xt)`, which creates an initial weight matrix appropriate for dynamic panel models. We have

$$\mathbf{W}_0 = N\left(\sum_i \mathbf{Z}_i'\boldsymbol{\Omega}\mathbf{Z}_i\right)^{-1} \otimes \mathbf{I}_K$$

If `fod` is specified, then

$$\boldsymbol{\Omega} = \mathbf{I}_{T-p-1}$$

If `fd` is specified, then

$$\boldsymbol{\Omega} = \begin{pmatrix} -2 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & \ddots & 0 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 \end{pmatrix}$$

These initial weight matrices assume that for residual vector $\boldsymbol{\epsilon}_{it}$ elements ϵ_{itj} and ϵ_{itk} are independently and identically distributed for $j \neq k$. Matrix $\boldsymbol{\Omega}$ is proportional to the covariance matrix of the transformed residual $\widetilde{\epsilon}_{it}$. (The constant of proportionality cancels out of all the formulas we use involving $\boldsymbol{\Omega}$.) The form of $\boldsymbol{\Omega}$ in the case of using first differences to remove the fixed effects follows from the fact that even though we assume $\boldsymbol{\epsilon}_{it}$ is independent across t (and i), by taking first differences, the transformed residual will follow a first-order moving average process with coefficient $\theta = -(1/2)$.

If you specify the `onestep` option, then `xtvar` reports a cluster–robust variance–covariance matrix of $\widehat{\boldsymbol{\theta}}_1$ that allows for arbitrary within-cluster correlation. We have

$$\text{Var}(\widehat{\boldsymbol{\theta}}_1) = \left(\boldsymbol{Q}'_{Z'X}\mathbf{W}_0\boldsymbol{Q}_{Z'X}\right)^{-1}\boldsymbol{Q}'_{Z'X}\mathbf{W}_0\widehat{\mathbf{S}_1}\mathbf{W}_0\boldsymbol{Q}_{Z'X}\left(\boldsymbol{Q}'_{Z'X}\mathbf{W}_0\boldsymbol{Q}_{Z'X}\right)^{-1}$$

where

$$\widehat{\mathbf{S}_1} = \frac{1}{N}\sum_i \mathbf{Z}_i'\widehat{\mathbf{e}}_{1i}\widehat{\mathbf{e}}'_{1i}\mathbf{Z}_i \tag{12}$$

and $\widehat{\mathbf{e}}_{1i}$ are the residuals from the first step.

Two-step estimator

To compute the two-step estimator, we first obtain the one-step estimated parameter vector $\widehat{\boldsymbol{\theta}}_1$, use it to compute the residuals $\widehat{\mathbf{e}}_{1i}$, and use those to compute $\widehat{\mathbf{S}}_1$ as shown in (12). Hansen (1982) shows that to obtain the optimal GMM estimator, we should use as a weight matrix the inverse of the VCE of the moment conditions. $\widehat{\mathbf{S}}_1$ is a consistent estimator of that VCE, so we use as our second-step weight matrix $\widehat{\mathbf{W}}_1 = \widehat{\mathbf{S}}_1^{-1}$. The two-step estimator is then

$$\widehat{\boldsymbol{\theta}}_2 = \left(\boldsymbol{Q}'_{Z'X}\widehat{\mathbf{W}}_1\boldsymbol{Q}_{Z'X}\right)^{-1}\boldsymbol{Q}'_{Z'X}\widehat{\mathbf{W}}_1\boldsymbol{Q}_{Z'y}$$

The default VCE is the Windmeijer (2005) robust VCE given by

$$\mathrm{Var}(\widehat{\boldsymbol{\theta}}_2) = \frac{1}{N}V_2 + \frac{1}{N}(DV_2 + V_2D') + D\,\mathrm{Var}(\widehat{\boldsymbol{\theta}}_1)D'$$

where

$$\begin{aligned}
V_2 &= \left(\boldsymbol{Q}'_{Z'X}\widehat{\mathbf{W}}_1\boldsymbol{Q}_{Z'X}\right)^{-1}\\
\frac{\partial S(\theta)}{\partial\theta_j} &= -\frac{1}{N}\sum_i\left(\boldsymbol{Q}^{[j]}_{Z'X}\widehat{\mathbf{e}}'_{1i}\mathbf{Z}_i + \widehat{\mathbf{e}}_{1i}\mathbf{Z}'_i\boldsymbol{Q}^{[j]\prime}_{Z'X}\right)\\
D^{[j]} &= -V_2\boldsymbol{Q}'_{Z'X}\widehat{\mathbf{W}}_1\frac{\partial S(\theta)}{\partial\theta_j}\widehat{\mathbf{W}}_1\bar{\mathbf{g}}\left(\widehat{\boldsymbol{\theta}}_2\right)\\
\bar{\mathbf{g}}\left(\widehat{\boldsymbol{\theta}}_2\right) &= \frac{1}{N}\sum_i\mathbf{Z}'_i\widehat{\mathbf{e}}_{2i}
\end{aligned}$$

and $[j]$ corresponds to the jth column of a matrix and $\widehat{\mathbf{e}}_{2i}$ are the residuals of the two-step estimator.

Hansen's J statistic

Hansen's J statistic is N times the GMM objective function as in

$$J = N\left(\frac{1}{N}\sum_i\mathbf{Z}'_i\widehat{\mathbf{e}}_{1i}\right)'\widehat{\mathbf{W}}\left(\frac{1}{N}\sum_i\mathbf{Z}'_i\widehat{\mathbf{e}}_{1i}\right)$$

where $\widehat{\mathbf{W}}$ is the weight matrix used, either the one-step or two-step weight matrix.

Acknowledgment

We thank Brian Poi of Poi Consulting LLC for his advice on and review of the xtvar command.

References

Akaike, H. 1973. "Information theory and an extension of the maximum likelihood principle". In *Second International Symposium on Information Theory*, edited by B. N. Petrov and F. Csaki, 267–281. Budapest: Akailseoniai–Kiudo.

Andrews, D. W. K., and B. Lu. 2001. Consistent model and moment selection procedures for GMM estimation with application to dynamic panel data models. *Journal of Econometrics* 101: 123–164. https://doi.org/10.1016/S0304-4076(00)00077-4.

Arellano, M., and S. Bond. 1991. Some tests of specification for panel data: Monte Carlo evidence and an application to employment equations. *Review of Economic Studies* 58: 277–297. https://doi.org/10.2307/2297968.

Arellano, M., and O. Bover. 1995. Another look at the instrumental variable estimation of error-components models. *Journal of Econometrics* 68: 29–51. https://doi.org/10.1016/0304-4076(94)01642-D.

Blomquist, S., and M. Dahlberg. 1999. Small sample properties of LIML and jackknife IV estimators: Experiments with weak instruments. *Journal of Applied Econometrics* 14: 69–88. https://doi.org/10.1002/(SICI)1099-1255(199901/02)14:1<C69::AID-JAE521>3.0.CO;2-7.

Cameron, A. C., and P. K. Trivedi. 2022. *Microeconometrics Using Stata*. 2nd ed. College Station, TX: Stata Press.

Hall, A. R. 2005. *Generalized Method of Moments*. Oxford: Oxford University Press.

Hamilton, J. D. 1994. *Time Series Analysis*. Princeton, NJ: Princeton University Press. https://doi.org/10.2307/j.ctv14jx6sm.

Han, C., and P. C. B. Phillips. 2006. GMM with many moment conditions. *Econometrica* 74: 147–192. https://doi.org/10.1111/j.1468-0262.2006.00652.x.

Hannan, E. J., and B. G. Quinn. 1979. The determination of the order of an autoregression. *Journal of the Royal Statistical Society*, B ser., 41: 190–195. https://doi.org/10.1111/j.2517-6161.1979.tb01072.x.

Hansen, L. P. 1982. Large sample properties of generalized method of moments estimators. *Econometrica* 50: 1029–1054. https://doi.org/10.2307/1912775.

Hayakawa, K. 2009. First difference or forward orthogonal deviation—which transformation should be used in dynamic panel data models?: A simulation study. *Economics Bulletin* 29: 2008–2017.

Hayashi, F. 2000. *Econometrics*. Princeton, NJ: Princeton University Press.

Kilian, L., and H. Lütkepohl. 2017. *Structural Vector Autoregressive Analysis*. Cambridge: Cambridge University Press. https://doi.org/10.1017/9781108164818.

Lütkepohl, H. 2005. *New Introduction to Multiple Time Series Analysis*. New York: Springer.

Nerlove, M. 1967. Experimental evidence on the estimation of dynamic economic relations from a time series of cross-section. *Economics Studies Quarterly* 18: 42–74. https://doi.org/10.11398/economics1950.18.3_42.

Nickell, S. J. 1981. Biases in dynamic models with fixed effects. *Econometrica* 49: 1417–1426. https://doi.org/10.2307/1911408.

Roodman, D. 2009a. A note on the theme of too many instruments. *Oxford Bulletin of Economics and Statistics* 71: 135–158. https://doi.org/10.1111/j.1468-0084.2008.00542.x.

———. 2009b. How to do xtabond2: An introduction to difference and system GMM in Stata. *Stata Journal* 9: 86–136.

Schwarz, G. 1978. Estimating the dimension of a model. *Annals of Statistics* 6: 461–464. https://doi.org/10.1214/aos/1176344136.

Stock, J. H., and J. H. Wright. 2000. GMM with weak identification. *Econometrica* 68: 1055–1096. https://doi.org/10.1111/1468-0262.00151.

Windmeijer, F. 2005. A finite sample correction for the variance of linear efficient two-step GMM estimators. *Journal of Econometrics* 126: 25–51. https://doi.org/10.1016/j.jeconom.2004.02.005.

Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

Also see

[XT] **xtvar postestimation** — Postestimation tools for xtvar

[XT] **xtset** — Declare data to be panel data

[TS] **forecast** — Econometric model forecasting

[TS] **var intro** — Introduction to vector autoregressive models

[TS] **var** — Vector autoregressive models

[U] 20 Estimation and postestimation commands

Postestimation commands

The following postestimation commands are of special interest after `xtvar`:

Command	Description
`irf`	create and analyze IRFs
`vargranger`	Granger causality tests
`xtvarsoc`	model- and moment-selection criteria (MMSC)
`varstable`	check stability condition of estimates
`varwle`	Wald lag-exclusion statistics

The following standard postestimation commands are also available:

Command	Description
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
`forecast`	dynamic forecasts and simulations
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	fitted values, error components
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

predict

Description for predict

predict creates a new variable containing predictions such as fitted values and panel-level error components.

Menu for predict

Statistics > Postestimation

Syntax for predict

predict [*type*] *newvar* [*if*] [*in*] [, *statistic* equation(*eqno* | *eqname*)]

statistic	Description
Main	
xb	fitted values; the default
* xbu	fitted values including panel effect
* u	u_i, the fixed error component
* e	e_{it}, the overall error component

Unstarred statistics are available both in and out of sample; type predict ... if e(sample) ... if wanted only for the estimation sample. Starred statistics are calculated only for the estimation sample, even when if e(sample) is not specified.

Options for predict

Main

xb calculates the fitted values, the linear prediction, for the specified equation. This is the default.

xbu calculates the fitted values including the panel-level fixed effect for the specified equation.

u calculates the panel-level fixed effect for the specified equation.

e calculates the overall error component, the residual, for the specified equation.

equation(*eqno* | *eqname*) specifies the equation for which the prediction is desired.

You need to specify either one equation number (*eqno*) or equation name (*eqname*) with equation(). For example, equation(#1) indicates that the calculation be made for the first equation, equation(#2) would refer to the second equation, and so on. You could also refer to the equation by its name, which is the same as the corresponding variable name; thus, equation(lnwage) would refer to the equation for the dependent variable named lnwage.

If you do not specify equation(), the results are the same as if you specified equation(#1).

For more information on using predict after multiple-equation estimation commands, see [R] **predict**.

xtvarsoc

Description for xtvarsoc

xtvarsoc calculates MMSC for panel-data vector autoregressive (VAR) models previously fit by xtvar. xtvarsoc can be used to select the appropriate lag conditional on a specification for the instrumental variables, and it can be used to select the appropriate specification for the instrumental-variables conditional on the lag order chosen for the model.

Menu for xtvarsoc

Statistics > Postestimation

Syntax for xtvarsoc

Lag-order selection syntax

xtvarsoc [, *lag_options*]

Instrument selection syntax

xtvarsoc *namelist* [, separator(#)]

namelist specifies the names of previously stored sets of xtvar estimates.

lag_options	Description
minlag(#)	minimum lag order to consider; default is minlag(1)
maxlag(#)	maximum lag order to consider; default is based on previous model
estimates(*estname*)	use previously stored results *estname*; default is to use active results
separator(#)	draw a separator line after every # rows; default is separator(0)

collect is allowed; see **[U] 11.1.10 Prefix commands**.

Options for xtvarsoc

minlag(#) specifies the minimum lag order to consider when computing the MMSCs. The default is minlag(1).

maxlag(#) specifies the maximum lag order to consider when computing the MMSCs. By default, xtvarsoc uses the lag order of the currently active estimation results or the lag order of the model specified in the estimates() option as the maximum.

estimates(*estname*) specifies the name of a previously stored set of xtvar estimates. By default, xtvarsoc uses the currently active estimation results. See [R] **estimates** for information on manipulating estimation results.

separator(#) specifies how often separator lines should be drawn between rows. By default, separator lines do not appear. For example, separator(1) would draw a line between each row, separator(2) would draw a line between every other row, and so on.

Remarks and examples

Remarks are presented under the following headings:

Model stability and hypothesis testing

The following commands are available after `xtvar` to check the model's stability and perform certain hypothesis tests commonly used after fitting a VAR model:

`vargranger` (see [TS] **vargranger**);

`varstable` (see [TS] **varstable**); and

`varwle` (see [TS] **varwle**).

See example 6, example 7, and example 8 of [XT] **xtvar** for illustrations and discussions of each of these commands following `xtvar`.

IRFs

The full suite of commands listed in [TS] **irf** is available after `xtvar`. `irf create` estimates the following IRFs after `xtvar`:

1. Simple IRFs
2. Orthogonalized IRFs
3. Cumulative IRFs
4. Cumulative orthogonalized IRFs

For each of these IRFs, standard errors are by default computed using asymptotic formulas based on the delta method and the sampling variances of the estimated parameters. Optionally, you may specify the `bs` option to request that the standard errors instead be computed using the bootstrap. After `xtvar`, `irf create` uses a cluster bootstrap, sampling with replacement from panels included in the estimation sample. At each bootstrap replication, `irf create` refits your model using the same instrument specification that you used at estimation time. By default, if `xtvar` reports that it encountered a singular matrix, the bootstrap replication is ignored.

We show how to create and analyze IRFs after `xtvar` in example 9 and example 10 of [XT] **xtvar**.

MMSC

When working with time-series VAR models, analysts often use various information criteria (IC) to determine the optimal lag length of a VAR model. These include the IC of the Akaike (1973) information criterion (AIC), Schwarz's (1978) Bayesian information criterion (BIC), and the Hannan and Quinn (1979) information criterion (HQIC). Each of those ICs is a function of the maximized log likelihood of a model and includes a penalty term that is increasing in the number of parameters. A model with a lower, say, HQIC is to be preferred to one with a higher HQIC.

`xtvar` fits VARs with panel data using the generalized method of moments (GMM), which does not require us to place any strong distributional assumptions on the error terms, such as normality. There is no likelihood function that we can compute, so we cannot obtain the traditional ICs used with time-series VARs.

Andrews and Lu (2001) developed what they coined "model- and moment-selection criteria" for use with GMM models that are akin to the AIC, BIC, and HQIC for maximum likelihood models. Rather than use the maximized log likelihood, these MMSC are based on Hansen's (1982) J statistic, which is normally used to test the validity of the overidentifying restrictions of a GMM model. Like the traditional ICs, Andrews and Lu's MMSC apply a penalty term that favors more parsimonious models. Also like traditional ICs, a model with a lower MMSC is preferable to one with a higher MMSC. There are three variants of Andrews and Lu's (2001) MMSC, corresponding to the AIC, BIC, and HQIC. We therefore refer to these variants as the MMSC-AIC, MMSC-BIC, and MMSC-HQIC.

Much like with `varsoc` after fitting a time-series VAR model, you use `xtvarsoc` after fitting one or more models with `xtvar`; `xtvarsoc` reports MMSC rather than ICs after `xtvar`. `xtvarsoc`, in contrast to `varsoc`, does not provide results before fitting a model because panel-data VAR models require you to provide an appropriate specification of the instrumental variables, and you do that when you fit the model with `xtvar`.

Andrews and Lu's (2001) MMSC can be used for multiple purposes, including selecting a lag length as well as selecting an instrument specification. `xtvarsoc` therefore has two syntaxes, one for selecting a lag length and one for selecting instruments. For the lag-order selection syntax, you call `xtvarsoc` and optionally specify the maximum and minimum number of lags to consider. `xtvarsoc` will use the instrument specification you specified when fitting your candidate `xtvar` model, and it is careful to maintain the same estimation sample across all candidate models, with varying numbers of lags, that it fits. The key is that `xtvarsoc` will fit variants of your model with different lag lengths.

We illustrated how to use `xtvarsoc` for lag-order selection in example 4 of [XT] **xtvar**. We reiterate a few key points to bear in mind when using `xtvar` for lag-order selection. For model comparisons to be valid, the number of observations should remain constant across all candidate models. Moreover, when we are choosing the lag length, the number of moment conditions in each model should remain constant as well. If the number of moment conditions changes from model to model, then we are comparing not only models with different lag lengths but also models with different instrument specifications. If our goal is to choose an optimal lag length, we should compare only similar models. Finally, before comparing MMSC across models, you should first examine Hansen's J statistic so that you are choosing only between the set of models with valid moment conditions.

To use `xtvarsoc` for instrument selection, you need to specify a list of previously stored estimation results, and `xtvarsoc` will report Hansen's J statistic and the MMSC for the models you specified.

▷ Example 1: Choosing an instrument specification

In [XT] **xtvar**, we fit a three-variable panel-data VAR model using Dahlberg and Johansson's (2000) Swedish municipality data. In example 4, we used `xtvarsoc` to decide upon a model with four lags of the left-hand-side variables, and we specified the `maxldep(2)` option with `xtvar` so that two lags of those variables were used as instruments. Here we reexamine the `maxldep(2)` option. Should we have used three lags as instruments? Four? We can answer that question with `xtvarsoc`.

We first load the dataset and fit three variants of the model, changing the argument in `maxldep()` each time. We fit these models with the `quietly` prefix to save space, but you are welcome to omit that prefix.

```
. use https://www.stata-press.com/data/r19/swedishgov
(1979-1987 Swedish municipality data)

. quietly xtvar expenditures revenues grants, lags(4) maxldep(2)

. estimates title: "lags(4), maxldep(2)"

. estimates store ldep_2
```

```
. quietly xtvar expenditures revenues grants, lags(4) maxldep(3)
. estimates title: "lags(4), maxldep(3)"
. estimates store ldep_3
. quietly xtvar expenditures revenues grants, lags(4) maxldep(4)
. estimates title: "lags(4), maxldep(4)"
. estimates store ldep_4
```

We used `estimates title` to add a label to each of our estimation results before storing them with `estimates store`. This will help avoid ambiguity when we call `xtvarsoc`, which we do now:

```
. xtvarsoc ldep_2 ldep_3 ldep_4
Model- and moment-selection criteria
```

Est. #	N	MC	Hansen's J	df	p	MMSC-AIC	MMSC-BIC	MMSC-HQIC
1	1060	72	38.80	36	0.345	-33.199	-211.98	-100.95
2	1060	108	101.02	72	0.014	-42.976	-400.53	-178.49
3	1060	144	147.46	108	0.007	-68.544*	-604.87*	-271.81*

`* indicates minimum value within column.`

Est. #	Name	Title
1	ldep_2	lags(4), maxldep(2)
2	ldep_3	lags(4), maxldep(3)
3	ldep_4	lags(4), maxldep(4)

In our call to `xtvarsoc`, we simply specified the names of our stored results. Looking at the output, we see that all three estimates have the same sample size, so we are comparing models with the same estimation sample. Unlike when we use `xtvarsoc` to select the lag length, here we are not concerned that the number of moment conditions in each model—listed in the column marked `MC`—differs. After all, we are looking at different specifications for the instruments, and the instrument specification determines the number of moment conditions.

What caught our eye is the fact that we reject Hansen's J test of overidentifying restrictions for the models where we used more than two lags of the dependent variables as instruments. All three of the MMSC would lead us to pick the model with four lags as instruments, but because we reject the validity of the overidentifying restrictions for that model, we are inclined to stick with our original model that uses two lags.

◁

The previous example illustrates a common theme we have noticed when fitting panel-data VARs: in many cases, it is not so much a matter of selecting the "optimal" lag length or instrument specification as it is a matter of finding a model where we can find a valid set of instruments.

Stored results

`xtvarsoc` stores the following in `r()`:

Matrices

`r(results)`	sample size, J statistics, and MMSCs

Methods and formulas

Methods and formulas are presented under the following headings:

predict

The complete panel-data VAR model is

$$\mathbf{y}_{it} = \mathbf{A}_1\mathbf{y}_{i,t-1} + \cdots + \mathbf{A}_p\mathbf{y}_{i,t-p} + \mathbf{B}\mathbf{x}_{it} + \mathbf{C}\mathbf{w}_{it} + \mathbf{D}\mathbf{v}_{it} + \mathbf{u}_i + \boldsymbol{\epsilon}_{it}$$

The fitted values, obtained by specifying the `xb` option with `predict`, are

$$\widehat{\mathbf{y}}_{it} = \widehat{\mathbf{A}}_1\mathbf{y}_{i,t-1} + \cdots + \widehat{\mathbf{A}}_p\mathbf{y}_{i,t-p} + \widehat{\mathbf{B}}\mathbf{x}_{it} + \widehat{\mathbf{C}}\mathbf{w}_{it} + \widehat{\mathbf{D}}\mathbf{v}_{it}$$

The overall error component, $\boldsymbol{\nu}_{it} = \mathbf{u}_i + \boldsymbol{\epsilon}_{it}$, consists of the fixed-effect term $\mathbf{u}_i$ and the residuals $\boldsymbol{\epsilon}_{it}$. Let $\widehat{\boldsymbol{\nu}}_{it} = \mathbf{y}_{it} - \widehat{\mathbf{y}}_{it}$. We estimate the fixed-effect term as $\widehat{\mathbf{u}}_i = \sum_t \widehat{\boldsymbol{\nu}}_{it}/T$, with the obvious adaptation for panels with missing time periods. We estimate the sample residuals as $\widehat{\boldsymbol{\epsilon}}_{it} = \widehat{\boldsymbol{\nu}}_{it} - \widehat{\mathbf{u}}_i$. Finally, we estimate the fitted values including the fixed-effect term as $\widehat{\mathbf{y}}_{it} + \widehat{\mathbf{u}}_i$.

irf create

The formulas for the IRFs, orthogonalized IRFs, their cumulative variants, and their standard errors after `xtvar` are identical to those used after `var`. See *Impulse–response function formulas for VAR models* in [TS] **irf create**.

xtvarsoc

Let N denote sample size, J denote Hansen's J test statistic of overidentifying restrictions, and d represent the corresponding degrees of freedom. The MMSC are computed as follows:

$$\begin{aligned}
\text{MMSC}_{\text{AIC}} &= J - 2d \\
\text{MMSC}_{\text{BIC}} &= J - d\ln N \\
\text{MMSC}_{\text{HQIC}} &= J - 2d\ln\ln N
\end{aligned}$$

The MMSC will not be available if the final weight matrix of the underlying model is not of full rank or if the model is just identified. When the weight matrix is not full rank is true, we report each MMSC as `NFR` and set the value to `.r` in the `r(results)` matrix result. When the model is just identified, we report `JI` for each MMSC and set the value to `.j` in `r(results)`.

References

Akaike, H. 1973. "Information theory and an extension of the maximum likelihood principle". In *Second International Symposium on Information Theory*, edited by B. N. Petrov and F. Csaki, 267–281. Budapest: Akailseoniai–Kiudo.

Andrews, D. W. K., and B. Lu. 2001. Consistent model and moment selection procedures for GMM estimation with application to dynamic panel data models. *Journal of Econometrics* 101: 123–164. https://doi.org/10.1016/S0304-4076(00)00077-4.

Dahlberg, M., and E. Johansson. 2000. An examination of the dynamic behaviour of local governments using GMM bootstrapping methods. *Journal of Applied Econometrics* 15: 401–416. https://doi.org/10.1002/1099-1255(200007/08)15:4<401::AID-JAE564>3.0.CO;2-G.

Hannan, E. J., and B. G. Quinn. 1979. The determination of the order of an autoregression. *Journal of the Royal Statistical Society*, B ser., 41: 190–195. https://doi.org/10.1111/j.2517-6161.1979.tb01072.x.

Hansen, L. P. 1982. Large sample properties of generalized method of moments estimators. *Econometrica* 50: 1029–1054. https://doi.org/10.2307/1912775.

Schwarz, G. 1978. Estimating the dimension of a model. *Annals of Statistics* 6: 461–464. https://doi.org/10.1214/aos/1176344136.

Also see

[XT] **xtvar** — Panel-data vector autoregressive models

[U] 20 Estimation and postestimation commands

Glossary

Arellano–Bond estimator. The Arellano–Bond estimator is a generalized method of moments (GMM) estimator for linear dynamic panel-data models that uses lagged levels of the endogenous variables as well as first differences of the exogenous variables as instruments. The Arellano–Bond estimator removes the panel-specific heterogeneity by first-differencing the regression equation.

autoregressive process. In autoregressive processes, the current value of a variable is a linear function of its own past values and a white-noise error term. For panel data, a first-order autoregressive process, denoted as an AR(1) process, is $y_{it} = \rho y_{i,t-1} + \epsilon_{it}$, where i denotes panels, t denotes time, and ϵ_{it} is white noise.

balanced data. A longitudinal or panel dataset is said to be balanced if each panel has the same number of observations. See also *weakly balanced* and *strongly balanced*.

between estimator. The between estimator is a panel-data estimator that obtains its estimates by running OLS on the panel-level means of the variables. This estimator uses only the between-panel variation in the data to identify the parameters, ignoring any within-panel variation. For it to be consistent, the between estimator requires that the panel-level means of the regressors be uncorrelated with the panel-specific heterogeneity terms.

canonical link. Corresponding to each family of distributions in a generalized linear model is a canonical link function for which there is a sufficient statistic with the same dimension as the number of parameters in the linear predictor. The use of canonical link functions provides the GLM with desirable statistical properties, especially when the sample size is small.

conditional fixed-effects model. In general, including panel-specific dummies to control for fixed effects in nonlinear models results in inconsistent estimates. For some nonlinear models, the fixed-effect term can be removed from the likelihood function by conditioning on a sufficient statistic. For example, the conditional fixed-effect logit model conditions on the number of positive outcomes within each panel.

conditional hazard function. In the context of random-effects survival models, the conditional hazard function is the hazard function computed conditionally on the random effects. Even within the same covariate pattern, the conditional hazard function varies among individuals who belong to different random-effects clusters.

conditional hazard ratio. In the context of random-effects survival models, the conditional hazard ratio is the ratio of two conditional hazard functions evaluated at different values of the covariates. Unless stated differently, the denominator corresponds to the conditional hazard function at baseline, that is, with all the covariates set to zero.

correlation structure. A correlation structure is a set of assumptions imposed on the within-panel variance–covariance matrix of the errors in a panel-data model. See [XT] **xtgee** for examples of different correlation structures.

cross-sectional data. Cross-sectional data refers to data collected over a set of individuals, such as households, firms, or countries sampled from a population at a given point in time.

cross-sectional time-series data. Cross-sectional time-series data is another name for panel data. The term *cross-sectional time-series data* is sometimes reserved for datasets in which a relatively small number of panels were observed over many periods. See also *panel data*.

disturbance term. The disturbance term encompasses any shocks that occur to the dependent variable that cannot be explained by the conditional (or deterministic) portion of the model.

dynamic model. A dynamic model is one in which prior values of the dependent variable or disturbance term affect the current value of the dependent variable.

endogenous variable. An endogenous variable is a regressor that is correlated with the unobservable error term. Equivalently, an endogenous variable is one whose values are determined by the equilibrium or outcome of a structural model.

error-components model. The error-components model is another name for the random-effects model. See also *random-effects model*.

exogenous variable. An exogenous variable is a regressor that is not correlated with any of the unobservable error terms in the model. Equivalently, an exogenous variable is one whose values change independently of the other variables in a structural model.

fixed-effects model. The fixed-effects model is a model for panel data in which the panel-specific errors are treated as fixed parameters. These parameters are panel-specific intercepts and therefore allow the conditional mean of the dependent variable to vary across panels. The linear fixed-effects estimator is consistent, even if the regressors are correlated with the fixed effects. See also *random-effects model*.

generalized estimating equations (GEE). The method of generalized estimating equations is used to fit population-averaged panel-data models. GEE extends the GLM method by allowing the user to specify a variety of different within-panel correlation structures.

generalized linear model. The generalized linear model is an estimation framework in which the user specifies a distributional family for the dependent variable and a link function that relates the dependent variable to a linear combination of the regressors. The distribution must be a member of the exponential family of distributions. The generalized linear model encompasses many common models, including linear, probit, and Poisson regression.

idiosyncratic error term. In longitudinal or panel-data models, the idiosyncratic error term refers to the observation-specific zero-mean random-error term. It is analogous to the random-error term of cross-sectional regression analysis.

instrumental variables. Instrumental variables are exogenous variables that are correlated with one or more of the endogenous variables in a structural model. The term *instrumental variable* is often reserved for those exogenous variables that are not included as regressors in the model.

instrumental-variables (IV) estimator. An instrumental variables estimator uses instrumental variables to produce consistent parameter estimates in models that contain endogenous variables. IV estimators can also be used to control for measurement error.

interval data. Interval data are data in which the true value of the dependent variable is not observed. Instead, all that is known is that the value lies within a given interval.

link function. In a generalized linear model, the link function relates a linear combination of predictors to the expected value of the dependent variable. In a linear regression model, the link function is simply the identity function.

longitudinal data. Longitudinal data is another term for panel data. See also *panel data*.

negative binomial regression model. The negative binomial regression model is for applications in which the dependent variable represents the number of times an event occurs. The negative binomial regression model is an alternative to the Poisson model for use when the dependent variable is overdispersed, meaning that the variance of the dependent variable is greater than its mean.

overidentifying restrictions. The order condition for model identification requires that the number of exogenous variables excluded from the model be at least as great as the number of endogenous regressors. When the number of excluded exogenous variables exceeds the number of endogenous regressors, the model is overidentified, and the validity of the instruments can then be checked via a test of overidentifying restrictions.

panel data. Panel data are data in which the same units were observed over multiple periods. The units, called panels, are often firms, households, or patients who were observed at several points in time. In a typical panel dataset, the number of panels is large, and the number of observations per panel is relatively small.

panel-corrected standard errors (PCSEs). The term *panel-corrected standard errors* refers to a class of estimators for the variance–covariance matrix of the OLS estimator when there are relatively few panels with many observations per panel. PCSEs account for heteroskedasticity, autocorrelation, or cross-sectional correlation.

Poisson regression model. The Poisson regression model is used when the dependent variable represents the number of times an event occurs. In the Poisson model, the variance of the dependent variable is equal to the conditional mean.

pooled estimator. A pooled estimator ignores the longitudinal or panel aspect of a dataset and treats the observations as if they were cross-sectional.

population-averaged model. A population-averaged model is used for panel data in which the parameters measure the effects of the regressors on the outcome for the average individual in the population. The panel-specific errors are treated as uncorrelated random variables drawn from a population with zero mean and constant variance, and the parameters measure the effects of the regressors on the dependent variable after integrating over the distribution of the random effects.

predetermined variable. A predetermined variable is a regressor in which its contemporaneous and future values are not correlated with the unobservable error term but past values are correlated with the error term.

prewhiten. To prewhiten is to apply a transformation to a time series so that it becomes white noise.

production function. A production function describes the maximum amount of a good that can be produced, given specified levels of the inputs.

quadrature. Quadrature is a set of numerical methods to evaluate a definite integral.

random-coefficients model. A random-coefficients model is a panel-data model in which group-specific heterogeneity is introduced by assuming that each group has its own parameter vector, which is drawn from a population common to all panels.

random-effects model. A random-effects model for panel data treats the panel-specific errors as uncorrelated random variables drawn from a population with zero mean and constant variance. The regressors must be uncorrelated with the random effects for the estimates to be consistent. See also *fixed-effects model*.

regressand. The regressand is the variable that is being explained or predicted in a regression model. Synonyms include dependent variable, left-hand-side variable, and endogenous variable.

regressor. Regressors are variables in a regression model used to predict the regressand. Synonyms include independent variable, right-hand-side variable, explanatory variable, predictor variable, and exogenous variable.

robust standard errors. Robust standard errors, also known as Huber/White or Taylor linearization standard errors, are based on the sandwich estimator of variance. Robust standard errors can be interpreted as representing the sample-to-sample variability of the parameter estimates, even when the model is misspecified. See also *semirobust standard errors*.

semirobust standard errors. Semirobust standard errors are closely related to robust standard errors and can be interpreted as representing the sample-to-sample variability of the parameter estimates, even when the model is misspecified, as long as the mean structure of the model is specified correctly. See also *robust standard errors*.

sequential limit theory. The sequential limit theory is a method of determining asymptotic properties of a panel-data statistic in which one index, say, N, the number of panels, is held fixed, while T, the number of time periods, goes to infinity, providing an intermediate limit. Then one obtains a final limit by studying the behavior of this intermediate limit as the other index (N here) goes to infinity.

strongly balanced. A longitudinal or panel dataset is said to be strongly balanced if each panel has the same number of observations and the observations for different panels were all made at the same times.

unbalanced data. A longitudinal or panel dataset is said to be unbalanced if each panel does not have the same number of observations. See also *weakly balanced* and *strongly balanced*.

weakly balanced. A longitudinal or panel dataset is said to be weakly balanced if each panel has the same number of observations but the observations for different panels were not all made at the same times.

white noise. A variable u_t represents a white-noise process if the mean of u_t is zero, the variance of u_t is σ^2, and the covariance between u_t and u_s is zero for all $s \neq t$.

within estimator. The within estimator is a panel-data estimator that removes the panel-specific heterogeneity by subtracting the panel-level means from each variable and then performing ordinary least squares on the demeaned data. The within estimator is used in fitting the linear fixed-effects model.

Subject and author index

See the combined subject index and the combined author index in the *Stata Index*.