STATA LONGITUDINAL-DATA/PANEL-DATA REFERENCE MANUAL

RELEASE 12

A Stata Press Publication
StataCorp LP
College Station, Texas

Version 12

Published by Stata Press, 4905 Lakeway Drive, College Station, Texas 77845
Typeset in TEX
Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN-10: 1-59718-096-3
ISBN-13: 978-1-59718-096-2

The suggested citation for this software is

StataCorp. 2011. *Stata: Release 12*. Statistical Software. College Station, TX: StataCorp LP.

Table of contents

Cross-referencing the documentation

When reading this manual, you will find references to other Stata manuals. For example,

[U] **26 Overview of Stata estimation commands**
[R] **regress**
[D] **reshape**

The first example is a reference to chapter 26, *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*
[D]	*Stata Data-Management Reference Manual*
[G]	*Stata Graphics Reference Manual*
[XT]	*Stata Longitudinal-Data/Panel-Data Reference Manual*
[MI]	*Stata Multiple-Imputation Reference Manual*
[MV]	*Stata Multivariate Statistics Reference Manual*
[P]	*Stata Programming Reference Manual*
[SEM]	*Stata Structural Equation Modeling Reference Manual*
[SVY]	*Stata Survey Data Reference Manual*
[ST]	*Stata Survival Analysis and Epidemiological Tables Reference Manual*
[TS]	*Stata Time-Series Reference Manual*
[I]	*Stata Quick Reference and Index*
[M]	*Mata Reference Manual*

Detailed information about each of these manuals may be found online at

http://www.stata-press.com/manuals/

Title

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

Description

This entry describes this manual and what has changed since Stata 11.

Remarks

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**	Fixed-, between-, and random-effects, and population-averaged linear models

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**.

What's new

This section is intended for previous Stata users. If you are new to Stata, you may as well skip it.

1. **MI support for panel-data and multilevel models** includes `xtcloglog`, `xtgee`, `xtlogit`, `xtmelogit`, `xtmepoisson`, `xtmixed`, `xtnbreg`, `xtpoisson`, `xtprobit`, `xtrc`, and `xtreg`. See [MI] **estimation**.

2. **Survey feature support for multilevel models**, `xtmixed`, including multilevel sampling weights and robust variance estimators. See [XT] **xtmixed**.

3. **Documentation for xtmixed, xtmelogit, and xtmepoisson has been modified to adopt the standard "level" terminology from the literature on hierarchical models.** For example, what in previous Stata versions was considered a one-level model is now called a two-level model with the observations now being counted as "level one"; see the *Introduction* section of *Remarks* in both [XT] **xtmixed** and [XT] **xtmelogit** for more details.

4. **Contrasts**, which is to say, tests of linear hypotheses involving factor variables and their interactions from the most recently fit model, and that model can be virtually any model that Stata can fit. Tests include ANOVA-style tests of main effects, simple effects, interactions, and nested effects. Effects can be decomposed into comparisons with reference categories, comparisons of adjacent levels, comparisons with the grand mean, and more. New commands `contrast` and `margins, contrast` are available after most xt estimation commands. See [R] **contrast** and [R] **margins, contrast**.

5. **Pairwise comparisons** of means, estimated cell means, estimated marginal means, predictive margins of linear and nonlinear responses, intercepts, and slopes. In addition to ANOVA-style comparisons, comparisons can be made of population averages. New commands `pwcompare` and `margins, pwcompare` are available after most xt estimation commands. See [R] **pwcompare** and [R] **margins, pwcompare**.

6. **Graphs of margins, marginal effects, contrasts, and pairwise comparisons**. Margins and effects can be obtained from linear or nonlinear (for example, probability) responses. New command `marginsplot` is available after all xt estimation commands; see [R] **marginsplot**.
7. **xtmixed now uses maximum likelihood (ML) as the default method of estimation**, where previously it used restricted maximum likelihood (REML). REML is still available with the `reml` option, and previous behavior is preserved under version control.
8. **Estimation output improved.**
   a. **Implied zero coefficients now shown**. When a coefficient is omitted, it is now shown as being zero and the reason it was omitted—collinearity, base, empty—is shown in the standard-error column. (The word "omitted" is shown if the coefficient was omitted because of collinearity.)
   b. **You can set displayed precision for all values in coefficient tables** using `set cformat`, `set pformat`, and `set sformat`. Or you may use options `cformat()`, `pformat()`, and `sformat()` now allowed on all estimation commands. See [R] **set cformat** and [R] **estimation options**.
   c. **Estimation commands now respect the width of the Results window.** This feature may be turned off by new display option `nolstretch`. See [R] **estimation options**.
   d. **You can now set whether base levels, empty cells, and omitted are shown** using `set showbaselevels`, `set showemptycells`, and `set showomitted`. See [R] **set showbaselevels**.
9. **Robust and cluster–robust SEs after fixed-effects xtpoisson.** See [XT] **xtpoisson**.
10. **New residual covariance structures for multilevel models** include exponential, banded, and Toeplitz. See [XT] **xtmixed**.
11. **Probability predictions now available.** `predict` after random-effects and population-averaged count-data models, such as `xtpoisson` and `xtgee`, can now predict the probability of any count or count range. See [XT] **xtpoisson postestimation**, [XT] **xtgee postestimation**, and [XT] **xtnbreg postestimation**.
12. **Option addplot() now places added graphs above or below.** Commands that allow option `addplot()` can now place the added plots above or below the command's plots. Affected is the command `xtline`; see [XT] **xtline**.

For a complete list of all the new features in Stata 12, see [U] **1.3 What's new**.

Also see

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

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

Title

xt — Introduction to xt commands

Syntax

`xt`*cmd* ...

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*, 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] **26.16 Models with time-series data** and the *Time-Series Reference Manual*.

Data management and exploration tools

`xtset`	Declare data to be panel data
`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`	Fixed-, between-, and random-effects, and population-averaged linear models
`xtregar`	Fixed- and random-effects linear models with an AR(1) disturbance
`xtmixed`	Multilevel mixed-effects linear regression
`xtgls`	Panel-data models by using GLS
`xtpcse`	Linear regression with panel-corrected standard errors
`xthtaylor`	Hausman–Taylor estimator for error-components models
`xtfrontier`	Stochastic frontier models for panel data
`xtrc`	Random-coefficients regression
`xtivreg`	Instrumental variables and two-stage least squares for panel-data models

Unit-root tests

xtunitroot	Panel-data unit-root 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

Censored-outcome estimators

xttobit	Random-effects tobit models
xtintreg	Random-effects interval-data regression models

Binary-outcome estimators

xtlogit	Fixed-effects, random-effects, and population-averaged logit models
xtmelogit	Multilevel mixed-effects logistic regression
xtprobit	Random-effects and population-averaged probit models
xtcloglog	Random-effects and population-averaged cloglog models

Count-data estimators

xtpoisson	Fixed-effects, random-effects, and population-averaged Poisson models
xtmepoisson	Multilevel mixed-effects Poisson regression
xtnbreg	Fixed-effects, random-effects, & population-averaged negative binomial models

Multilevel (hierarchical) mixed-effects estimators

xtmelogit	Multilevel mixed-effects logistic regression
xtmepoisson	Multilevel mixed-effects Poisson regression
xtmixed	Multilevel mixed-effects linear regression

Generalized estimating equations estimator

xtgee	Population-averaged panel-data models by using GEE

Remarks

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 (2008), Greene (2012, chap. 11), Hsiao (2003), and Wooldridge (2010). Cameron and Trivedi (2010) illustrate many of Stata's panel-data estimators.

▷ 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 $i = 1, \ldots, n$ and $t = 1, \ldots, T$, 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 (for Human Resource Research 1989) on young women aged 14–26 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 http://www.stata-press.com/data/r12/nlswork
(National Longitudinal Survey.  Young Women 14-26 years of age in 1968)

. describe

Contains data from http://www.stata-press.com/data/r12/nlswork.dta
  obs:        28,534                          National Longitudinal Survey.
                                                Young Women 14-26 years of age
                                                in 1968
 vars:            21                          7 Dec 2010 17:02
 size:       941,622
-------------------------------------------------------------------------------
              storage  display     value
variable 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                  1=white, 2=black, 3=other
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	28534	2601.284	1487.359	1	5159
year	28534	77.95865	6.383879	68	88
birth_yr	28534	48.08509	3.012837	41	54
age	28510	29.04511	6.700584	14	46
race	28534	1.303392	.4822773	1	3
msp	28518	.6029175	.4893019	0	1
nev_mar	28518	.2296795	.4206341	0	1
grade	28532	12.53259	2.323905	0	18
collgrad	28534	.1680451	.3739129	0	1
not_smsa	28526	.2824441	.4501961	0	1
c_city	28526	.357218	.4791882	0	1
south	28526	.4095562	.4917605	0	1
ind_code	28193	7.692973	2.994025	1	12
occ_code	28413	4.777672	3.065435	1	13
union	19238	.2344319	.4236542	0	1
wks_ue	22830	2.548095	7.294463	0	76
ttl_exp	28534	6.215316	4.652117	0	28.88461
tenure	28101	3.123836	3.751409	0	25.91667
hours	28467	36.55956	9.869623	1	168
wks_work	27831	53.98933	29.03232	0	104
ln_wage	28534	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 http://www.stata-press.com/data/r12/union
(NLS Women 14-24 in 1968)
. describe
Contains data from http://www.stata-press.com/data/r12/union.dta
  obs:        26,200                          NLS Women 14-24 in 1968
 vars:             8                          4 May 2011 13:54
 size:       235,800
-------------------------------------------------------------------------------
              storage  display     value
variable 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 |     26200    2611.582    1484.994         1       5159
        year |     26200    79.47137    5.965499        70         88
         age |     26200    30.43221    6.489056        16         46
       grade |     26200    12.76145    2.411715         0         18
    not_smsa |     26200    .2837023    .4508027         0          1
-------------+--------------------------------------------------------
       south |     26200    .4130153    .4923849         0          1
       union |     26200    .2217939    .4154611         0          1
       black |     26200     .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

`xtmixed`, `xtmelogit`, and `xtmepoisson` do not use the information pertaining to i and t that is stored by `xtset`. Unlike the other xt commands, these can handle multiple nested levels of groups and thus use their own syntax for specifying the group structure of the data.

❑

❑ 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^2_1 but is rather a 50:50 mixture of a χ^2_0 (point mass at zero) and a χ^2_1, denoted as $\overline{\chi}^2_{01}$. See Gutierrez, Carter, and Drukker (2001) for a full discussion, and see [XT] **xtmixed** for a generalization of the concept as applied to variance-component estimation in mixed models.

❑

References

Baltagi, B. H. 2008. *Econometric Analysis of Panel Data*. 4th ed. New York: Wiley.

Cameron, A. C., and P. K. Trivedi. 2010. *Microeconometrics Using Stata*. Rev. ed. College Station, TX: Stata Press.

Greene, W. H. 2012. *Econometric Analysis*. 7th ed. Upper Saddle River, NJ: Prentice Hall.

Gutierrez, R. G., S. 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. 2003. *Analysis of Panel Data*. 2nd ed. New York: Cambridge University Press.

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

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

Title

quadchk — Check sensitivity of quadrature approximation

Syntax

```
quadchk [#1 #2] [, nooutput nofrom]
```

Menu

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

Description

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

- `xtcloglog`
- `xtintreg`
- `xtlogit`
- `xtpoisson, re` with the `normal` option
- `xtprobit`
- `xttobit`

`quadchk` refits the model for different numbers of quadrature points and then compares the different solutions.

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

Most options supplied to the original model are respected by `quadchk`, but some are not. These are `or`, `vce()`, and the *maximize_options*.

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. Adaptive quadrature with `intmethod(aghermite)` is more sensitive to starting values than nonadaptive quadrature, `intmethod(ghermite)`, or the default method of adaptive quadrature, `intmethod(mvaghermite)`. Specifying the `nofrom` option can level the playing field in testing estimation results.

Remarks

Remarks are presented under the following headings:

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 http://www.stata-press.com/data/r12/quad1
. xtset id
       panel variable:  id (balanced)
. xtprobit z x1-x6
  (output omitted)
Random-effects probit regression              Number of obs      =      6000
Group variable: id                            Number of groups   =       300
Random effects u_i ~ Gaussian                 Obs per group: min =        20
                                                             avg =      20.0
                                                             max =        20
                                              Wald chi2(6)       =     29.24
Log likelihood  = -3347.1097                  Prob > chi2        =    0.0001
```

z	Coef.	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

```
Likelihood-ratio 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                         8.983e-15     -1.388e-06   Difference
                                 2.086e-12     -.00032222   Relative difference
-------------------------------------------------------------------------------
z:                .10007418      .10007418      .10007431
      x2                         2.540e-15      1.362e-07   Difference
                                 2.538e-14      1.361e-06   Relative difference
-------------------------------------------------------------------------------
z:                .15035391      .15035391      .15035406
      x3                         6.356e-15      1.520e-07   Difference
                                 4.227e-14      1.011e-06   Relative difference
-------------------------------------------------------------------------------
z:                .12301495      .12301495      .12301506
      x4                         4.149e-15      1.099e-07   Difference
                                 3.373e-14      8.931e-07   Relative difference
-------------------------------------------------------------------------------
z:                .13429881      .13429881      .13429896
      x5                         4.913e-15      1.471e-07   Difference
                                 3.658e-14      1.096e-06   Relative difference
-------------------------------------------------------------------------------
z:                .08799332      .08799332      .08799346
      x6                         3.358e-15      1.363e-07   Difference
                                 3.817e-14      1.549e-06   Relative difference
-------------------------------------------------------------------------------
z:                .07570675      .07570675      .07570423
   _cons                         1.962e-14     -2.516e-06   Difference
                                 2.592e-13     -.00003323   Relative difference
-------------------------------------------------------------------------------
lnsig2u:         -.03299164     -.03299164     -.03298184
   _cons                         7.268e-14      9.798e-06   Difference
                                -2.203e-12     -.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      =      6000
Group variable: id                            Number of groups   =       300
Random effects u_i ~ Gaussian                 Obs per group: min =        20
                                                             avg =      20.0
                                                             max =        20
                                              Wald chi2(6)       =     36.15
Log likelihood  = -3349.6926                  Prob > chi2        =    0.0000
```

z	Coef.	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

```
Likelihood-ratio 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 quadrature 12 points	Comparison quadrature 8 points	Comparison quadrature 16 points	
Log likelihood	-3349.6926	-3354.6372	-3348.3881	
		-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
-------------+----------------------------------------------------------------
lnsig2u:     -.07918212     -.18133821     -.05815644
   _cons                    -.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      =      6000
Group variable: id                              Number of groups   =       300

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

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

------------------------------------------------------------------------------
           z |      Coef.   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
------------------------------------------------------------------------------
Likelihood-ratio 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
-------------------------------------------------------------------------------
lnsig2u:       -.03298317     -.03298909     -.03298186
   _cons                     -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 http://www.stata-press.com/data/r12/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      =      6000
Group variable: id                            Number of groups   =       300
Random effects u_i ~ Gaussian                 Obs per group: min =        20
                                                             avg =      20.0
                                                             max =        20
                                              Wald chi2(6)       =     39.43
Log likelihood  = -4065.3144                  Prob > chi2        =    0.0000
```

z	Coef.	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

```
Likelihood-ratio 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      5.457e-12   Difference
                           5.578e-12     -1.342e-15   Relative difference
-------------------------------------------------------------------------------
z:           .02469427      .02469427      .02469427
      x1                  -3.645e-12     -8.007e-12   Difference
                          -1.476e-10     -3.242e-10   Relative difference
-------------------------------------------------------------------------------
z:           .13001229      .13001229      .13001229
      x2                  -1.566e-11     -6.879e-13   Difference
                          -1.204e-10     -5.291e-12   Relative difference
-------------------------------------------------------------------------------
z:           .11904089      .11904089      .11904089
      x3                  -6.457e-12     -3.030e-13   Difference
                          -5.425e-11     -2.545e-12   Relative difference
-------------------------------------------------------------------------------
z:           .13919697      .13919697      .13919697
      x4                   1.442e-12      1.693e-13   Difference
                           1.036e-11      1.216e-12   Relative difference
-------------------------------------------------------------------------------
z:           .07736398      .07736398      .07736398
      x5                  -5.801e-12     -4.556e-13   Difference
                          -7.499e-11     -5.890e-12   Relative difference
-------------------------------------------------------------------------------
z:           .08620282      .08620282      .08620282
      x6                   5.903e-12      3.191e-13   Difference
                           6.848e-11      3.702e-12   Relative difference
-------------------------------------------------------------------------------
z:           .09226527      .09226527      .09226527
   _cons                  -2.850e-12     -1.837e-11   Difference
                          -3.089e-11     -1.991e-10   Relative difference
-------------------------------------------------------------------------------
lnsig2u:    -2.3439389     -2.3439389     -2.3439389
   _cons                  -2.946e-09     -2.172e-10   Difference
                           1.257e-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.

◁

Methods and formulas

`quadchk` is implemented as an ado-file.

Title

vce_options — Variance estimators

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

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.

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] **20.20 Obtaining robust variance estimates**.

If the command allows `pweights` and you specify them, `vce(robust)` is implied; see [U] **20.22.3 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] **20.20 Obtaining robust variance estimates**.

`vce(bootstrap` [`,` *bootstrap_options*]`)` uses a 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 chi-squared (or generalized chi-squared) 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

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 http://www.stata-press.com/data/r12/clogitid
. xtlogit y x1 x2, fe vce(jackknife)
(running xtlogit on estimation sample)
Jackknife replications (66)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
................
Conditional fixed-effects logistic regression   Number of obs      =       369
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 |      Coef.   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
Group variable: id                              Number of groups   =        66
                                                Obs per group: min =         2
                                                               avg =       5.6
                                                               max =        10
                                                Wald chi2(2)       =      8.52
Log likelihood  = -123.41386                    Prob > chi2        =    0.0141
                                  (Replications based on 66 clusters in id)
------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Normal-based
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |    .653363   .3015317     2.17   0.030      .0623717   1.244354
          x2 |   .0659169   .0512331     1.29   0.198    -.0344981    .1663319
------------------------------------------------------------------------------
```

❑ 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 have to type

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

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] **20.20 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 & 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**

Title

xtabond — Arellano–Bond linear dynamic panel-data estimation

Syntax

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

options	Description
Model	
`noconstant`	suppress constant term
`diffvars(`*varlist*`)`	already-differenced exogenous variables
`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*, however, may not contain time-series operators.

`by`, `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.

Menu

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

Description

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.

This 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**.

Options

Model

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

`diffvars(`*varlist*`)` specifies a set of variables that already have been differenced to be included as strictly exogenous covariates.

`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* 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

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**.

▷ Example 1

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 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 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 $\mathtt{n}_{it}$ 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 http://www.stata-press.com/data/r12/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	Coef.	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 chi-squared 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
```

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

```
H0: no autocorrelation
```

When the idiosyncratic errors are independently 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

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 |      Coef.   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
```

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

```
H0: no autocorrelation
```

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

`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 chi-squared 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

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)
```

n	Coef.	WC-Robust 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
```

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

H0: no autocorrelation

◁

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

Thus far we have been specifying the `noconstant` option to keep to the standard Arellano–Bond estimator, which uses instruments only for the differenced 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 |      Coef.   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

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	Coef.	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

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 |      Coef.   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

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)
------------------------------------------------------------------------------
             |               WC-Robust
           n |      Coef.   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

`xtabond` handles data in which there are missing observations in the middle of the panels. In the following 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 dropped from div() because of collinearity
note: yr1981 dropped from div() because of collinearity
note: yr1982 dropped from div() because of collinearity
note: yr1980 dropped because of collinearity
note: yr1981 dropped because of collinearity
note: yr1982 dropped 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 |      Coef.   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 dropped 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 http://www.stata-press.com/data/r12/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 dropped from div() because of collinearity
note: yr1980 dropped 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 |      Coef.   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 dropped 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.

◁

Saved results

`xtabond` saves 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_max)`	largest group size
`e(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(t_max)`	maximum time in sample
`e(t_min)`	minimum 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(hascons)`	`hascons`, if specified
`e(transform)`	specified transform
`e(datasignature)`	checksum from `datasignature`
`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

Methods and formulas

`xtabond` is implemented as an ado-file.

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^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. 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.

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

——. 1982. Formulation and estimation of dynamic models using panel data. *Journal of Econometrics* 18: 47–82.

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.

Baltagi, B. H. 2008. *Econometric Analysis of Panel Data*. 4th ed. New York: 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.

Hansen, L. P. 1982. Large sample properties of generalized method of moments estimators. *Econometrica* 50: 1029–1054.

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

Kiviet, J. F. 1995. On bias, inconsistency, and efficiency of various estimators in dynamic panel data models. *Journal of Econometrics* 68: 53–78.

Layard, R., and S. J. Nickell. 1986. Unemployment in Britain. *Economica* 53: S121–S169.

Windmeijer, F. 2005. A finite sample correction for the variance of linear efficient two-step GMM estimators. *Journal of Econometrics* 126: 25–51.

Also see

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

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

[XT] **xtdpdsys** — Arellano–Bover/Blundell–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** — Fixed-, between-, and random-effects and population-averaged linear models

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

[U] **20 Estimation and postestimation commands**

Title

xtabond postestimation — Postestimation tools for xtabond

Description

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

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

For information about these commands, see below.

The following standard postestimation commands are also available:

Command	Description
`estat`	VCE and estimation sample summary
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`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 coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`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

See the corresponding entries in the *Base Reference Manual* for details.

Special-interest postestimation commands

`estat abond` reports the Arellano–Bond tests for serial correlation in the first-differenced errors.

`estat sargan` reports the Sargan test of the overidentifying restrictions.

Syntax for predict

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

Menu

Statistics > Postestimation > Predictions, residuals, etc.

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.

Syntax for estat abond

```
estat abond [ , artests(#) ]
```

Menu

Statistics > Postestimation > Reports and statistics

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 be refit only if the data have not changed.

Syntax for estat sargan

```
estat sargan
```

Menu

Statistics > Postestimation > Reports and statistics

Remarks

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

Methods and formulas

All postestimation commands listed above are implemented as ado-files.

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

Also see

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

[U] **20 Estimation and postestimation commands**

Title

xtcloglog — Random-effects and population-averaged cloglog models

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
collinear	keep collinear variables
SE	
vce(*vcetype*)	*vcetype* may be oim, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
noskip	perform overall model test as a likelihood-ratio test
eform	report exponentiated coefficients
nocnsreport	do not display constraints
display_options	control column formats, row spacing, line width, and display of omitted variables and base and empty cells
Integration	
intmethod(*intmethod*)	integration method; *intmethod* may be mvaghermite, aghermite, or ghermite; default is intmethod(mvaghermite)
intpoints(#)	use # quadrature points; default is intpoints(12)
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-group correlation structure; see table below
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 column formats, row spacing, line width, and display of omitted variables and base and empty cells
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, 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**.

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.

coeflegend does not appear in the dialog box.

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

Menu

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

Description

`xtcloglog` fits population-averaged and random-effects complementary log-log (cloglog) models. 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.

By default, the population-averaged model is an equal-correlation model; that is, `xtcloglog, pa` assumes `corr(exchangeable)`. See [XT] **xtgee** for information on fitting other population-averaged models.

See [R] **logistic** for a list of related 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*`)`, `collinear`; see [R] **estimation options**.

SE

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

Reporting

`level(#)`, `noskip`; see [R] **estimation options**.

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

`nocnsreport`; see [R] **estimation options**.

display_options: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `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 option is available with xtcloglog 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.

offset(*varname*); see [R] **estimation options**

Correlation

corr(*correlation*), force; see [R] **estimation options**.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory, that are robust to some kinds of misspecification, and that use bootstrap or jackknife methods; 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: noomitted, vsquish, noemptycells, baselevels, allbaselevels, 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.

nolog suppresses display of the iteration log.

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

`xtcloglog, pa` is a shortcut command for fitting the population-averaged model. 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 $i = 1, \ldots, n$ panels, where $t = 1, \ldots, n_i$, ν_i are i.i.d., $N(0, \sigma_\nu^2)$, and $P(z) = 1 - \exp\{-\exp(z)\}$.

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 http://www.stata-press.com/data/r12/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      =     26200
Group variable: idcode                          Number of groups   =      4434
Random effects u_i ~ Gaussian                   Obs per group: min =         1
                                                               avg =       5.9
                                                               max =        12
                                                Wald chi2(6)       =    248.58
Log likelihood  = -10535.928                    Prob > chi2        =    0.0000
```

union	Coef.	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

```
Likelihood-ratio 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      =     26200
Group variable:                      idcode     Number of groups   =      4434
Link:                               cloglog     Obs per group: min =         1
Family:                            binomial                    avg =       5.9
Correlation:                   exchangeable                     max =        12
                                                Wald chi2(6)       =    234.66
Scale parameter:                           1    Prob > chi2        =    0.0000
```

union	Coef.	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 (the random-effects estimator does not allow the `vce(robust)` specification), 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      =     26200
Group variable:                       idcode    Number of groups   =      4434
Link:                                cloglog    Obs per group: min =         1
Family:                             binomial                   avg =       5.9
Correlation:                   exchangeable                    max =        12
                                                Wald chi2(6)       =    157.24
Scale parameter:                           1    Prob > chi2        =    0.0000
                                 (Std. Err. adjusted for clustering on idcode)
```

union	Coef.	Robust 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.

❑

Saved results

xtcloglog, re saves the following in e():

Scalars

e(N)	number of observations
e(N_g)	number of groups
e(N_cd)	number of completely determined 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(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)	significance
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(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(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

`xtcloglog, pa` saves 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)`	significance
`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(Cns)`	constraints matrix
`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

Methods and formulas

`xtcloglog` is implemented as an ado-file.

`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,

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

where

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

The panel-level likelihood l_i is given by

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

$$\equiv \int_{-\infty}^{\infty} g(y_{it}, x_{it}, \nu_i) d\nu_i$$

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

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

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

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

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

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

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.

One can instead use the adaptive quadrature method of Liu and Pierce (1994), the `intmethod(aghermite)` option, which uses the mode and curvature of the mode as approximations for the mean and variance. We take the integrand

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

and find α_i the mode of $g(y_{it}, x_{it}, \nu_i)$. We calculate

$$\gamma_i = -\frac{\partial^2}{\partial \nu_i^2} \log\{g(y_{it}, x_{it}, \nu_i)\}\big|_{\nu_i = \alpha_i}$$

Then

$$\int_{-\infty}^{\infty} g(y_{it}, x_{it}, \nu_i) d\nu_i \approx \left(\frac{2}{\gamma_i}\right)^{1/2} \sum_{m=1}^{M} w_m^* \exp\{(a_m^*)^2\} g\left\{y_{it}, x_{it}, \left(\frac{2}{\gamma_i}\right)^{1/2} a_m^* + \alpha_i\right\}$$

This adaptation is performed on the first iteration only; that is, the α_i and γ_i are calculated once at the first iteration and then held constant throughout the subsequent iterations.

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

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

All three 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.

References

Liang, K.-Y., and S. L. Zeger. 1986. Longitudinal data analysis using generalized linear models. *Biometrika* 73: 13–22.

Liu, Q., and D. A. Pierce. 1994. A note on Gauss–Hermite quadrature. *Biometrika* 81: 624–629.

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, Series C* 31: 214–225.

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

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.

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.

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

Also see

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

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

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

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

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

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

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

[U] **20 Estimation and postestimation commands**

Title

xtcloglog postestimation — Postestimation tools for xtcloglog

Description

The following postestimation commands are available after `xtcloglog`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
*`estat`	AIC, BIC, VCE, and estimation sample summary
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`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 coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

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

See the corresponding entries in the *Base Reference Manual* for details.

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
`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 $\mathbf{x}_j\boldsymbol{\beta}$

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

Menu

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

`xb` calculates the linear prediction. This is the default for the random-effects model.

`pu0` calculates the probability of a positive outcome, assuming that the random effect for that observation's panel is zero ($\nu = 0$). 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, $u_j = \partial \ln L_j(\mathbf{x}_j\boldsymbol{\beta})/\partial(\mathbf{x}_j\boldsymbol{\beta})$.

`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 $\mathbf{x}_{it}\boldsymbol{\beta} + \text{offset}_{it}$.

Remarks

▷ Example 1

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

```
. use http://www.stata-press.com/data/r12/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   =      26200
Model VCE    : Conventional
Expression   : Pr(union != 0), predict()
dy/dx w.r.t. : age grade not_smsa 1.south year
```

	Delta-method dy/dx	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**

Title

xtdata — 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**.

Menu

Statistics > Longitudinal/panel data > Setup and utilities > 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.

Options

Main

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

`fe` specifies that the data are to 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

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 http://www.stata-press.com/data/r12/nlswork
(National Longitudinal Survey.  Young Women 14-26 years of age 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 =    4697
-------------+------------------------------           F( 10,  4686) =  450.23
       Model |  415.021613    10  41.5021613            Prob > F      =  0.0000
    Residual |  431.954995  4686  .092179896            R-squared     =  0.4900
-------------+------------------------------           Adj R-squared =  0.4889
       Total |  846.976608  4696  .180361288            Root MSE      =  .30361

------------------------------------------------------------------------------
     ln_wage |      Coef.   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 http://www.stata-press.com/data/r12/nlswork, clear
(National Longitudinal Survey.  Young Women 14-26 years of age 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		
Model	412.443881	8	51.5554852	Number of obs	= 28091
Residual	1976.12232	28082	.070369714	F(8, 28082)	= 732.64
				Prob > F	= 0.0000
Total	2388.5662	28090	.085032617	R-squared	= 0.1727
				Adj R-squared	= 0.1724
				Root MSE	= .26527

ln_wage	Coef.	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 $0.25790313/0.29069544 = 0.88719358$, 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 http://www.stata-press.com/data/r12/xtdatasmpl, clear
(National Longitudinal Survey.  Young Women 14-26 years of age in 1968)
. xtdata, clear re ratio(.88719358)
------------------- 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 =   28091
-------------+------------------------------           F( 11, 28080) =14303.11
       Model |  13272.3241    11  1206.57492            Prob > F      =  0.0000
    Residual |  2368.75918 28080  .084357521            R-squared     =  0.8486
-------------+------------------------------           Adj R-squared =  0.8485
       Total |  15641.0833 28091  .556800517            Root MSE      =  .29044

------------------------------------------------------------------------------
      ln_wage |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        grade |   .0646499   .0017811    36.30   0.000     .0611588     .068141
          age |   .0368059   .0031195    11.80   0.000     .0306915    .0429204
         age2 |  -.0007133     .00005   -14.27   0.000    -.0008113   -.0006153
  (output omitted)
        south |  -.0868927   .0073031   -11.90   0.000    -.1012072   -.0725781
    constant |    .238721   .0494688     4.83   0.000     .1417598    .3356822
------------------------------------------------------------------------------
```

Results are the same coefficients and standard errors that `xtreg, re` previously estimated. 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.

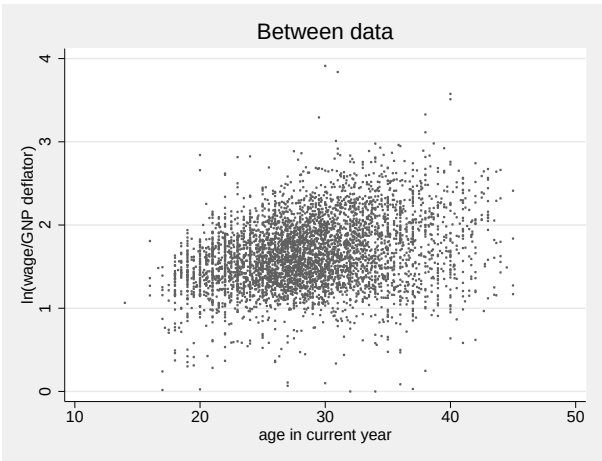
5. After converting the data, you may form linear, but not nonlinear, combinations of regressors; that is, if your data contained age, it would not be correct to convert the data and then form age squared. All nonlinear transformations should be done before conversion. (For `xtdata, be`, you can get away with forming nonlinear combinations ex post, but the results will not be exact.)

❑

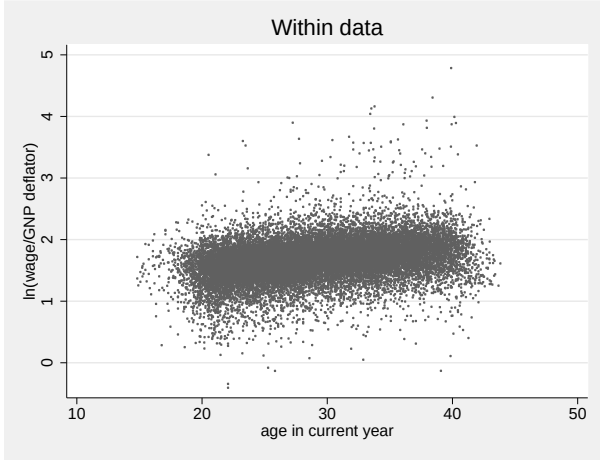
❑ Technical note

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

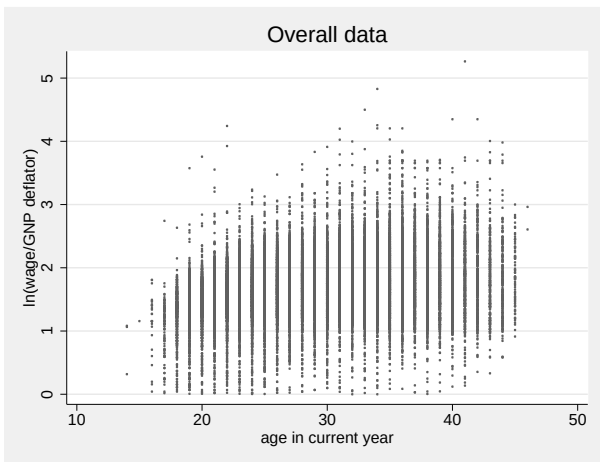
```
. use http://www.stata-press.com/data/r12/xtdatasmpl, clear
(National Longitudinal Survey.  Young Women 14-26 years of age in 1968)
. xtdata, be
. scatter ln_wage age, title(Between data) msymbol(o) msize(tiny)
```




```
. use http://www.stata-press.com/data/r12/xtdatasmpl, clear
(National Longitudinal Survey.  Young Women 14-26 years of age in 1968)
. xtdata, fe
. scatter ln_wage age, title(Within data) msymbol(o) msize(tiny)
```




```
. use http://www.stata-press.com/data/r12/xtdatasmpl, clear
(National Longitudinal Survey.  Young Women 14-26 years of age in 1968)
. scatter ln_wage age, title(Overall data) msymbol(o) msize(tiny)
```



Methods and formulas

`xtdata` is implemented as an ado-file.

(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

Title

xtdescribe — 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**.

Menu

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

Description

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

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

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 14–26 years in 1968. Here is a description of the data used in many of the [XT] **xt** examples:

```
. use http://www.stata-press.com/data/r12/nlswork
(National Longitudinal Survey.  Young Women 14-26 years of age 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 the [XT] **xt** examples:

```
. 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 http://www.stata-press.com/data/r12/xtdesxmpl
. xtdescribe
 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 |  11111111111111111111111111111111
         3     10.00   80.00 |  1111111111111111111111111111111..
         2      6.67   86.67 |  ..11111111111111111111111111111111
         2      6.67   93.33 |  .11111111111111111111111111111111
         2      6.67  100.00 |  1.11111111111111111111111111111111
------------------------------+-----------------------------------
        30    100.00          |  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

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.

◁

Methods and formulas

`xtdescribe` is implemented as an ado-file.

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

Title

xtdpd — Linear dynamic panel-data 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`, `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.

Menu

Statistics > Longitudinal/panel data > Dynamic panel data (DPD) > Linear DPD estimation

Description

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.

At the cost of a more complicated syntax, `xtdpd` can fit 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`; see [XT] **xtabond** and [XT] **xtdpdsys**.

Options

Model

`dgmmiv(`*varlist* [`, lagrange(`*flag* [*llag*]`)`]`)` specifies GMM-type instruments for the differenced 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 differenced 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 differenced 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 differenced equation. You may specify as many additional sets of standard instruments for the differenced 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.

`fodeviation` specifies that forward-orthogonal deviations are to 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

If you have not read [XT] **xtabond** and [XT] **xtdpdsys**, you should do so before continuing.

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 differenced 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 differenced 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

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 inside the United Kingdom 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`. To gain some insight into the syntax for `xtdpd`, we reproduce the first example from [XT] **xtabond** using `xtdpd`:

```
. use http://www.stata-press.com/data/r12/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	Coef.	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 differenced 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 differenced 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 differenced 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 differenced 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 differenced 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 differenced equation. Because there are 14 standard instruments, there must have been 27 GMM-type instruments, which matches our above calculation.

❑

▷ Example 2

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)
```

n	Coef.	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
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

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)
------------------------------------------------------------------------------
             |               WC-Robust
           n |      Coef.   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

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)
```

n	Coef.	Robust 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	-.9278879	-.5284301
L1.	.5602737	.1939617	2.89	0.004	.1801156	.9404317
L2.	-.0523028	.1487653	-0.35	0.725	-.3438775	.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	-93.75253	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

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 either the Arellano–Bond estimator or the Arellano–Bover/Blundell–Bond system estimator. For simplicity, we assume that the independent variables are strictly exogenous.

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 differenced 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	Coef.	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	Coef.	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 .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 differenced 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 differenced equation. An analogous argument works for higher lags.)

❑

Saved results

`xtdpd` saves 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(hascons)	hascons, if specified
e(transform)	specified transform
e(datasignature)	checksum from datasignature
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

Methods and formulas

xtdpd is implemented as an ado-file.

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).

$\mathbf{x}$ and $\mathbf{w}$ 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. Suppose that, for some i, the $t = 1$ observation was missing but was not missing for some other panels. `dgmmiv(y)` would then create the instrument matrix

$$\mathbf{Z}_{di} = \begin{pmatrix} 0 & 0 & 0 & y_{i2} & y_{i3} & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & y_{i2} & y_{i3} & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & y_{i2} & \dots & y_{iT-2} \end{pmatrix}$$

$\mathbf{Z}_{di}$ has $T_i - p - 1$ rows and $\sum_{m=p}^{\tau-2} m$ columns, where $\tau = \max_i \tau_i$ and τ_i is the number of nonmissing observations in panel i.

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

$$\widehat{\sigma}_1^2 = (1/(N-K)) \sum_i^N \widehat{\boldsymbol{\epsilon}}_{1i}^{*\prime} \widehat{\boldsymbol{\epsilon}}_{1i}^*$$

The GMM one-step VCE is then given by

$$\widehat{V}_{\mathrm{GMM}}[\widehat{\boldsymbol{\beta}}_1] = \widehat{\sigma}_1^2 \mathbf{W}_1^{-1}$$

The one-step level residuals are given by

$$\widehat{\boldsymbol{\epsilon}}_{1i}^{L} = \mathbf{y}_i^{L} - \widehat{\boldsymbol{\beta}}_1 \mathbf{X}_i^{L}$$

Stacking the residual vectors yields

$$\widehat{\boldsymbol{\epsilon}}_{1i} = \begin{pmatrix} \widehat{\boldsymbol{\epsilon}}_{1i}^{*} \\ \widehat{\boldsymbol{\epsilon}}_{1i}^{L} \end{pmatrix}$$

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 Center for Global Development, 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.

——. 1982. Formulation and estimation of dynamic models using panel data. *Journal of Econometrics* 18: 47–82.

Arellano, M. 2003. *Panel Data Econometrics*. Oxford: Oxford University Press.

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.

Arellano, M., and O. Bover. 1995. Another look at the instrumental variable estimation of error-components models. *Journal of Econometrics* 68: 29–51.

Baltagi, B. H. 2008. *Econometric Analysis of Panel Data*. 4th ed. New York: Wiley.

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

Blundell, R., and S. Bond. 1998. Initial conditions and moment restrictions in dynamic panel data models. *Journal of Econometrics* 87: 115–143.

Blundell, R., 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*, ed. B. H. Baltagi, 53–92. New York: Elsevier.

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.

Hansen, L. P. 1982. Large sample properties of generalized method of moments estimators. *Econometrica* 50: 1029–1054.

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

Layard, R., and S. J. Nickell. 1986. Unemployment in Britain. *Economica* 53: S121–S169.

Windmeijer, F. 2005. A finite sample correction for the variance of linear efficient two-step GMM estimators. *Journal of Econometrics* 126: 25–51.

Also see

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

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

[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** — Fixed-, between-, and random-effects and population-averaged linear models

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

[R] **gmm** — Generalized method of moments estimation

Title

xtdpd postestimation — Postestimation tools for xtdpd

Description

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

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

For information about these commands, see below.

The following standard postestimation commands are also available:

Command	Description
`estat`	VCE and estimation sample summary
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`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 coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`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

See the corresponding entries in the *Base Reference Manual* for details.

Special-interest postestimation commands

`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.

Syntax for predict

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

Menu

Statistics > Postestimation > Predictions, residuals, etc.

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.

Syntax for estat abond

```
estat abond [ , artests(#) ]
```

Menu

Statistics > Postestimation > Reports and statistics

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 be refit only if the data have not changed.

Syntax for estat sargan

```
estat sargan
```

Menu

Statistics > Postestimation > Reports and statistics

Remarks

Remarks are presented under the following headings:

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 independently 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 chi-squared 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

All postestimation commands listed above are implemented as ado-files.

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 $\widehat{\mathbf{u}}^{*}_{1i} = \mathrm{L}m.\widehat{\boldsymbol{\epsilon}}^{*}_{1i}$, 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,

$$\mathbf{Q}_{zu} = \left(\sum_i \mathbf{Z}'_i \mathbf{Q}_{\mathrm{fod}} \right)$$

where

$$\mathbf{Q}_{\mathrm{fod}} = \begin{pmatrix} -\sqrt{\frac{T_i+1}{T_i}} & 0 & \cdots & 0 \\ \sqrt{\frac{T_i-1}{T_i}} & \sqrt{\frac{T_i}{T_i-1}} & \cdots & 0 \\ 0 & . & . & \vdots \\ 0 & \cdots & \sqrt{\frac{1}{2}} & -\sqrt{\frac{2}{1}} \end{pmatrix} \widehat{\mathbf{u}}^{*}_{ji}$$

and * implies the first-differenced transform instead of the FOD transform.

For the Arellano–Bover/Blundell–Bond system estimator with the first-differenced transform,

$$\mathbf{Q}_{zu} = \left(\sum_i \mathbf{Z}'_i \widehat{\boldsymbol{\epsilon}}_{ji} \widehat{\boldsymbol{\epsilon}}^{*\prime}_{ji} \widehat{\mathbf{u}}^{*}_{ji} \right)$$

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.

Also see

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

Title

xtdpdsys — Arellano–Bover/Blundell–Bond linear dynamic panel-data 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`, `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.

Menu

Statistics > Longitudinal/panel data > Dynamic panel data (DPD) > Arellano-Bover/Blundell-Bond estimation

Description

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 this model. The Arellano and Bond estimator can perform poorly if the autoregressive parameters are too large or the ratio of the variance of the panel-level effect to the variance of idiosyncratic error is too large. Building on the work of Arellano and Bover (1995), Blundell and Bond (1998) developed a system estimator that uses additional moment conditions; `xtdpdsys` implements this 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.

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

If you have not read [XT] **xtabond**, you may want to do so before continuing.

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. 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 differenced 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. Because `xtdpdsys` extends `xtabond`, [XT] **xtabond** provides useful background.

▷ Example 1

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 http://www.stata-press.com/data/r12/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 |      Coef.   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	Coef.	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] **13.9 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 independently 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
```

Order	z	Prob > z
1	-4.6414	0.0000
2	-1.0572	0.2904

H0: no autocorrelation

The above output does not present evidence that the model is misspecified.

◁

▷ Example 2

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
```

n	Coef.	Robust 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	-.9278879	-.5284301
L1.	.5602737	.1939617	2.89	0.004	.1801156	.9404317
L2.	-.0523028	.1487653	-0.35	0.725	-.3438775	.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	-93.75253	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.

❑

Saved results

`xtdpdsys` saves 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(hascons)`	`hascons`, if specified
`e(transform)`	specified transform
`e(datasignature)`	checksum from `datasignature`
`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

Methods and formulas

`xtdpdsys` is implemented as an ado-file.

`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 Center for Global Development, 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.

——. 1982. Formulation and estimation of dynamic models using panel data. *Journal of Econometrics* 18: 47–82.

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.

Arellano, M., and O. Bover. 1995. Another look at the instrumental variable estimation of error-components models. *Journal of Econometrics* 68: 29–51.

Baltagi, B. H. 2008. *Econometric Analysis of Panel Data*. 4th ed. New York: Wiley.

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

Blundell, R., and S. Bond. 1998. Initial conditions and moment restrictions in dynamic panel data models. *Journal of Econometrics* 87: 115–143.

Blundell, R., 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*, ed. B. H. Baltagi, 53–92. New York: Elsevier.

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.

Hansen, L. P. 1982. Large sample properties of generalized method of moments estimators. *Econometrica* 50: 1029–1054.

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

Layard, R., and S. J. Nickell. 1986. Unemployment in Britain. *Economica* 53: S121–S169.

Windmeijer, F. 2005. A finite sample correction for the variance of linear efficient two-step GMM estimators. *Journal of Econometrics* 126: 25–51.

Also see

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

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

[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** — Fixed-, between-, and random-effects and population-averaged linear models

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

Title

xtdpdsys postestimation — Postestimation tools for xtdpdsys

Description

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

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

For information about these commands, see below.

The following standard postestimation commands are also available:

Command	Description
`estat`	VCE and estimation sample summary
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`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 coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`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

See the corresponding entries in the *Base Reference Manual* for details.

Special-interest postestimation commands

`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.

Syntax for predict

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

Menu

Statistics > Postestimation > Predictions, residuals, etc.

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.

Syntax for estat abond

```
estat abond [ , artests(#) ]
```

Menu

Statistics > Postestimation > Reports and statistics

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 be refit only if the data have not changed.

Syntax for estat sargan

```
estat sargan
```

Menu

Statistics > Postestimation > Reports and statistics

Remarks

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 a transform is required to remove the panel-level effects, but the transformed errors have a more complicated error structure than that of 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 independently 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 chi-squared 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 http://www.stata-press.com/data/r12/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.

Also see

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

Title

xtfrontier — Stochastic frontier models for panel data

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
`collinear`	keep collinear variables
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 column formats, row spacing, line width, and display of omitted variables and base and empty cells
Maximization	
maximize_options	control the maximization process; seldom used
`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
collinear	keep collinear variables
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 column formats, row spacing, line width, and display of omitted variables and base and empty cells
Maximization	
maximize_options	control the maximization process; seldom used
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 *indepvar*s may contain time-series operators; see [U] **11.4.4 Time-series varlists**.

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

`fweight`s and `iweight`s are allowed; 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.

Menu

Statistics > Longitudinal/panel data > Frontier models

Description

`xtfrontier` fits stochastic production or cost frontier models for panel data. More precisely, `xtfrontier` estimates the parameters of a linear model with a disturbance generated by specific mixture distributions.

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*. `xtfrontier` permits two different parameterizations of the inefficiency term: a time-invariant model and the Battese–Coelli (1992) parameterization of time effects. In the

time-invariant model, the inefficiency term is assumed to have a truncated-normal distribution. 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 both models, the idiosyncratic error term is assumed to have a normal distribution. The only panel-specific effect is the random inefficiency term.

See Kumbhakar and Lovell (2000) for a detailed introduction to frontier analysis.

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*`)`, `collinear`; see [R] **estimation options**.

SE

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

Reporting

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

`nocnsreport`; see [R] **estimation options**.

display_options: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `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 option is available with `xtfrontier` but is not shown in the dialog box:

`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*`)`, `collinear`; see [R] **estimation options**.

SE

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

Reporting

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

`nocnsreport`; see [R] **estimation options**.

display_options: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `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 option is available with `xtfrontier` but is not shown in the dialog box:

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

Remarks

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 $f(\mathbf{z}_{it}, \beta)$. In a world without error or inefficiency, in time t, the ith firm would produce

$$q_{it} = f(\mathbf{z}_{it}, \beta)$$

A fundamental element of stochastic frontier analysis is that each firm potentially produces less than it might because of a degree of inefficiency. Specifically,

$$q_{it} = f(\mathbf{z}_{it}, \beta)\xi_{it}$$

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 $f(\mathbf{z}_{it}, \beta)$. 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 $f(\mathbf{z}_{it}, \beta)$. 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

$$q_{it} = f(\mathbf{z}_{it}, \beta)\xi_{it}\exp(v_{it})$$

Taking the natural log of both sides yields

$$\ln(q_{it}) = \ln\{f(\mathbf{z}_{it}, \beta)\} + \ln(\xi_{it}) + v_{it}$$

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.

❑

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 independently and identically distributed.

Consider the simplest specification in which 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 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, $u_i \overset{\text{iid}}{\sim} N^+(\mu, \sigma_u^2)$, $v_{it} \overset{\text{iid}}{\sim} N(0, \sigma_v^2)$, and u_i and v_{it} are distributed independently of each other and the covariates in the model. Specifying the `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 $u_{it} = u_i$, $u_i \overset{\text{iid}}{\sim} N^+(\mu, \sigma_u^2)$, $v_{it} \overset{\text{iid}}{\sim} N(0, \sigma_v^2)$, and u_i and v_{it} are distributed independently of each other and the covariates in the model. 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 http://www.stata-press.com/data/r12/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	Coef.	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
/ilgtgamma	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`, `ilgtgamma`, `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 inverse logit of γ, and this estimate is reported as `ilgtgamma`. 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	Coef.	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
/ilgtgamma	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 Gould, Pitblado, and Poi (2010). 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:  ilgtgamma:         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	Coef.	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	.	.	.	.	.
/lnsigma2	1.422039	.2673128	5.32	0.000	.8981155	1.945962
/ilgtgamma	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

◁

Saved results

xtfrontier saves 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)	model significance
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(vcetype)	title used to label Std. Err.
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(V)	variance–covariance matrix of the estimators

Functions

e(sample)	marks estimation sample

Methods and formulas

`xtfrontier` is implemented as an ado-file.

`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

$$\begin{aligned} \ln L = &-\frac{1}{2}\left(\sum_{i=1}^{N} T_i\right)\{\ln(2\pi) + \ln(\sigma_S^2)\} - \frac{1}{2}\sum_{i=1}^{N}(T_i - 1)\ln(1-\gamma) \\ &- \frac{1}{2}\sum_{i=1}^{N}\ln\left\{1 + \left(\sum_{t=1}^{T_i}\eta_{it}^2 - 1\right)\gamma\right\} - N\ln\{1 - \Phi(-\widetilde{z})\} - \frac{1}{2}N\widetilde{z}^2 \\ &+ \sum_{i=1}^{N}\ln\{1 - \Phi(-z_i^*)\} + \frac{1}{2}\sum_{i=1}^{N} z_i^{*2} - \frac{1}{2}\sum_{i=1}^{N}\sum_{t=1}^{T_i}\frac{\epsilon_{it}^2}{(1-\gamma)\sigma_S^2} \end{aligned}$$

where $\sigma_S = (\sigma_u^2 + \sigma_v^2)^{1/2}$, $\gamma = \sigma_u^2/\sigma_S^2$, $\epsilon_{it} = y_{it} - \mathbf{x}_{it}\boldsymbol{\beta}$, $\eta_{it} = \exp\{-\eta(t - T_i)\}$, $\widetilde{z} = \mu/\left(\gamma\sigma_S^2\right)^{1/2}$, $\Phi()$ 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., C. A. K. Lovell, and P. Schmidt. 1977. Formulation and estimation of stochastic frontier production function models. *Journal of Econometrics* 6: 21–37.

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.

——. 1995. A model for technical inefficiency effects in a stochastic frontier production function for panel data. *Empirical Economics* 20: 325–332.

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.

Coelli, T. J. 1995. Estimators and hypothesis tests for a stochastic frontier function: A Monte Carlo analysis. *Journal of Productivity Analysis* 6: 247–268.

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.

Gould, W. W., J. S. Pitblado, and B. P. Poi. 2010. *Maximum Likelihood Estimation with Stata*. 4th ed. College Station, TX: Stata Press.

Kumbhakar, S. C., and C. A. K. Lovell. 2000. *Stochastic Frontier Analysis*. Cambridge: 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.

Zellner, A., and N. S. Revankar. 1969. Generalized production functions. *Review of Economic Studies* 36: 241–250.

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**

Title

xtfrontier postestimation — Postestimation tools for xtfrontier

Description

The following postestimation commands are available after `xtfrontier`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat`	AIC, BIC, VCE, and estimation sample summary
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`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 coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

See the corresponding entries in the *Base Reference Manual* for details.

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 $E\left(u_{it} \mid \epsilon_{it}\right)$
`m`	minus the natural log of the technical efficiency via $M\left(u_{it} \mid \epsilon_{it}\right)$
`te`	the technical efficiency via $E\left\{\exp(-su_{it}) \mid \epsilon_{it}\right\}$

where

$$s = \begin{cases} 1, & \text{for production functions} \\ -1, & \text{for cost functions} \end{cases}$$

Menu

Statistics > Postestimation > Predictions, residuals, etc.

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 $E\{\exp(-su_{it}) \mid \epsilon_{it}\}$.

Remarks

▷ 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 http://www.stata-press.com/data/r12/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	Coef.	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

All postestimation commands listed above are implemented as ado-files.

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 $f(u|\epsilon)$.

$$E\left(u_{it} \mid \epsilon_{it}\right) = \widetilde{\mu}_i + \widetilde{\sigma}_i \left\{ \frac{\phi\left(-\widetilde{\mu}_i/\widetilde{\sigma}_i\right)}{1-\Phi\left(-\widetilde{\mu}_i/\widetilde{\sigma}_i\right)} \right\}$$

$$M\left(u_{it} \mid \epsilon_{it}\right) = \begin{cases} -\widetilde{\mu}_i, & \text{if } \widetilde{\mu}_i >= 0 \\ 0, & \text{otherwise} \end{cases}$$

where

$$\widetilde{\mu}_i = \frac{\mu\sigma_v^2 - s\sum_{t=1}^{T_i}\eta_{it}\epsilon_{it}\sigma_u^2}{\sigma_v^2 + \sum_{t=1}^{T_i}\eta_{it}^2\sigma_u^2}$$

$$\widetilde{\sigma}_i^2 = \frac{\sigma_v^2\sigma_u^2}{\sigma_v^2 + \sum_{t=1}^{T_i}\eta_{it}^2\sigma_u^2}$$

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

$$E\left\{\exp(-su_{it}) \mid \epsilon_{it}\right\} = \left[\frac{1-\Phi\left\{s\eta_{it}\widetilde{\sigma}_i - (\widetilde{\mu}_i/\ \widetilde{\sigma}_i)\right\}}{1-\Phi\left(-\widetilde{\mu}_i/\ \widetilde{\sigma}_i\right)}\right] \exp\left(-s\eta_{it}\widetilde{\mu}_i + \frac{1}{2}\eta_{it}^2\widetilde{\sigma}_i^2\right)$$

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**

Title

xtgee — Fit population-averaged panel-data models by using GEE

Syntax

`xtgee` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, *options*]

options	Description
Model	
`family(`*family*`)`	distribution of *depvar*; see table below
`link(`*link*`)`	link function; see table below
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
`force`	estimate even if observations unequally spaced in time
Correlation	
`corr(`*correlation*`)`	within-group correlation structure; see table below
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 column formats, row spacing, line width, and display of omitted variables and base and empty cells
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 varlists**.

`by`, `fracpoly`, `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**.

`iweights`, `fweights`, and `pweights` 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 #

Menu

Statistics > Longitudinal/panel data > Generalized estimating equations (GEE) > Generalized estimating equations (GEE)

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.

See [R] **logistic** and [R] **regress** for lists of related estimation commands.

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.

`force`; see [R] **estimation options**.

Correlation

`corr(`*correlation*`)`; see [R] **estimation options**.

SE/Robust

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory, that are robust to some kinds of misspecification, and that use bootstrap or jackknife methods; 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: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `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.

`nolog` suppresses display of the iteration log.

`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

For a thorough introduction to GEE in the estimation of GLM, see Hardin and Hilbe (2003). 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 $g(\,)$ 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 $g(\,)$ 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

$$\mathbf{R} = \textit{matname}$$

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 `bprobit`; see [R] **probit** and [R] **glogit**.
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] **20.20 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 http://www.stata-press.com/data/r12/nlswork2
(National Longitudinal Survey.  Young Women 14-26 years of age in 1968)

. regress ln_w grade age c.age#c.age
```

Source	SS	df	MS	Number of obs =	16085
Model	597.54468	3	199.18156	F(3, 16081) =	1413.68
Residual	2265.74584	16081	.14089583	Prob > F =	0.0000
				R-squared =	0.2087
				Adj R-squared =	0.2085
Total	2863.29052	16084	.178021047	Root MSE =	.37536

ln_wage	Coef.	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 = 8.722e-13
```

GEE population-averaged model		Number of obs =	16085
Group variable:	idcode	Number of groups =	3913
Link:	identity	Obs per group: min =	1
Family:	Gaussian	avg =	4.1
Correlation:	independent	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	Coef.	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      =     16085
Group variable:                       idcode    Number of groups   =      3913
Link:                                identity   Obs per group: min =         1
Family:                              Gaussian                 avg =       4.1
Correlation:                    exchangeable                  max =         9
                                              Wald chi2(3)       =   2918.26
Scale parameter:                     .1416586  Prob > chi2        =    0.0000
```

ln_wage	Coef.	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 = -6035.2751
Iteration 1:   log likelihood = -5870.6718
Iteration 2:   log likelihood = -5858.9478
Iteration 3:   log likelihood = -5858.8244
Iteration 4:   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      =     16085
Group variable: idcode                          Number of groups   =      3913

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	Coef.	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

```
Likelihood-ratio 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      =     16085
Group variable: idcode                          Number of groups   =      3913

R-sq:  within  = 0.0983                         Obs per group: 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 |      Coef.   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      =     16085
Group variable:                      idcode     Number of groups   =      3913
Link:                              identity     Obs per group: min =         1
Family:                            Gaussian                       avg =       4.1
Correlation:                   exchangeable                       max =         9
                                                Wald chi2(3)       =   2031.28
Scale parameter:                   .1416586     Prob > chi2        =    0.0000
                                              (Std. Err. adjusted for clustering on idcode)
------------------------------------------------------------------------------
             |               Robust
     ln_wage |      Coef.   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.

◁

Saved results

xtgee saves 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)	significance
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

Methods and formulas

`xtgee` is implemented as an ado-file.

`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

$$
\begin{aligned}
h(\boldsymbol{\mu}_{i,j}) &= x_{i,j}^{\mathrm{T}}\boldsymbol{\beta} \\
\mathrm{Var}(y_{i,j}) &= g(\mu_{i,j})\phi \\
\boldsymbol{\mu}_i = E(\mathbf{y}_i) &= \{h^{-1}(x_{i,1}^{\mathrm{T}}\boldsymbol{\beta}), \ldots, h^{-1}(x_{i,n_i}^{\mathrm{T}}\boldsymbol{\beta})\}^{\mathrm{T}} \\
\mathbf{A}_i &= \mathrm{diag}\{g(\mu_{i,1}), \ldots, g(\mu_{i,n_i})\} \\
\mathrm{Cov}(\mathbf{y}_i) &= \phi\mathbf{A}_i \quad \text{for independent observations.}
\end{aligned}
$$

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

$$
\mathbf{S}_{\boldsymbol{\alpha}}(\beta, \boldsymbol{\alpha}) = \sum_{i=1}^{m} \left(\frac{\partial \boldsymbol{\eta}_i}{\partial \boldsymbol{\alpha}}\right)^{\mathrm{T}} \mathbf{H}_i^{-1}(\mathbf{W}_i - \boldsymbol{\eta}_i) = 0
$$

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,

$$
\mathbf{U}(\boldsymbol{\beta}) = \sum_{i=1}^{m} \left(\frac{\partial \boldsymbol{\mu}_i}{\partial \boldsymbol{\beta}}\right)^{\mathrm{T}} \mathbf{V}_i^{-1}(\boldsymbol{\alpha})(\mathbf{y}_i - \boldsymbol{\mu}_i) = 0
$$

$$
\text{where} \quad \mathbf{V}_i(\boldsymbol{\alpha}) = \mathbf{A}_i^{1/2}\mathbf{R}_i(\boldsymbol{\alpha})\mathbf{A}_i^{1/2}
$$

To solve this equation, we need only a crude approximation of the variance matrix, which we can obtain from a Taylor series expansion, where

$$\begin{aligned}\mathrm{Cov}(\mathbf{y}_i) &= \mathbf{L}_i\mathbf{Z}_i\mathbf{D}_i\mathbf{Z}_i^{\mathrm{T}}\mathbf{L}_i + \phi\mathbf{A}_i = \widetilde{\mathbf{V}}_i\\ \mathbf{L}_i &= \mathrm{diag}\{\partial h^{-1}(u)/\partial u, u = x_{i,j}^{\mathrm{T}}\boldsymbol{\beta}, j = 1,\ldots,n_i\}\end{aligned}$$

which allows that

$$\widehat{\mathbf{D}}_i \approx (\mathbf{Z}_i^{\mathrm{T}}\mathbf{Z}_i)^{-1}\mathbf{Z}_i\widehat{\mathbf{L}}_i^{-1}\left\{(\mathbf{y}_i-\widehat{\boldsymbol{\mu}}_i)(\mathbf{y}_i-\widehat{\boldsymbol{\mu}}_i)^{\mathrm{T}} - \widehat{\phi}\widehat{\mathbf{A}}_i\right\}\widehat{\mathbf{L}}_i^{-1}\mathbf{Z}_i^{\mathrm{T}}(\mathbf{Z}_i'\mathbf{Z}_i)^{-1}$$

$$\widehat{\phi} = \sum_{i=1}^{m}\sum_{j=1}^{n_i}\frac{(y_{i,j}-\widehat{\mu}_{i,j})^2 - (\widehat{\mathbf{L}}_{i,j})^2\mathbf{Z}_{i,j}^{\mathrm{T}}\widehat{\mathbf{D}}_i\mathbf{Z}_{i,j}}{g(\widehat{\mu}_{i,j})}$$

Calculating GEE for GLM

Using the notation from Liang and Zeger (1986), let $\mathbf{y}_i = (y_{i,1},\ldots,y_{i,n_i})^{\mathrm{T}}$ be the $n_i \times 1$ vector of outcome values, and let $\mathbf{X}_i = (x_{i,1},\ldots,x_{i,n_i})^{\mathrm{T}}$ be the $n_i \times p$ matrix of covariate values for the ith subject $i = 1,\ldots,m$. We assume that the marginal density for $y_{i,j}$ may be written in exponential family notation as

$$f(y_{i,j}) = \exp\left[\{y_{i,j}\theta_{i,j} - a(\theta_{i,j}) + b(y_{i,j})\}\,\phi\right]$$

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 $\mathbf{Z}_2$.

$$\begin{aligned}
\mathbf{Z}_1 &= \sum_{i=1}^{m} \mathbf{D}_i^{\mathrm{T}}(\widehat{\boldsymbol{\beta}}_j)\widetilde{\mathbf{V}}_i^{-1}(\widehat{\boldsymbol{\beta}}_j)\mathbf{D}_i(\widehat{\boldsymbol{\beta}}_j) = \sum_{i=1}^{m} \mathbf{X}_i^{\mathrm{T}}\boldsymbol{\Delta}_i^{\mathrm{T}}\mathbf{A}_i^{\mathrm{T}}\{\mathbf{A}_i^{1/2}\mathbf{R}(\boldsymbol{\alpha})\mathbf{A}_i^{1/2}\}^{-1}\mathbf{A}_i\boldsymbol{\Delta}_i\mathbf{X}_i \\
&= \sum_{i=1}^{m} \mathbf{X}_i^{\mathrm{T}} \operatorname{diag}\left\{\frac{\partial\theta_{i,j}}{\partial(\mathbf{X}\boldsymbol{\beta})}\right\} \operatorname{diag}\{a''(\theta_{i,j})\} \left[\operatorname{diag}\{a''(\theta_{i,j})\}^{1/2}\mathbf{R}(\boldsymbol{\alpha})\operatorname{diag}\{a''(\theta_{i,j})\}^{1/2}\right]^{-1} \\
&\qquad \operatorname{diag}\{a''(\theta_{i,j})\} \operatorname{diag}\left\{\frac{\partial\theta_{i,j}}{\partial(\mathbf{X}\boldsymbol{\beta})}\right\} \mathbf{X}_i \\
&= \sum_{i=1}^{m} \mathbf{X}_i^{\mathrm{T}}\mathbf{II}(\mathbf{III})^{-1}\mathbf{II}\mathbf{X}_i = \sum_{i=1}^{m} \mathbf{X}_i^{\mathrm{T}}\mathbf{X}_i = \mathbf{X}^{\mathrm{T}}\mathbf{X}
\end{aligned}$$

$$\begin{aligned}\mathbf{Z}_2 &= \sum_{i=1}^{m} \mathbf{D}_i^{\mathrm{T}}(\widehat{\boldsymbol{\beta}}_j)\widetilde{\mathbf{V}}_i^{-1}(\widehat{\boldsymbol{\beta}}_j)\mathbf{S}_i(\widehat{\boldsymbol{\beta}}_j) = \sum_{i=1}^{m} \mathbf{X}_i^{\mathrm{T}}\boldsymbol{\Delta}_i^{\mathrm{T}}\mathbf{A}_i^{\mathrm{T}}\{\mathbf{A}_i^{1/2}\mathbf{R}(\boldsymbol{\alpha})\mathbf{A}_i^{1/2}\}^{-1}\left(\mathbf{y}_i - \mathbf{X}_i\widehat{\boldsymbol{\beta}}_j\right) \\ &= \sum_{i=1}^{m} \mathbf{X}_i^{\mathrm{T}}\,\mathrm{diag}\left\{\frac{\partial\theta_{i,j}}{\partial(\mathbf{X}\boldsymbol{\beta})}\right\}\mathrm{diag}\left\{a''(\theta_{i,j})\right\}\left[\mathrm{diag}\left\{a''(\theta_{i,j})\right\}^{1/2}\mathbf{R}(\boldsymbol{\alpha})\,\mathrm{diag}\left\{a''(\theta_{i,j})\right\}^{1/2}\right]^{-1} \\ &\qquad\qquad \left(\mathbf{y}_i - \mathbf{X}_i\widehat{\boldsymbol{\beta}}_j\right) \\ &= \sum_{i=1}^{m} \mathbf{X}_i^{\mathrm{T}}\mathbf{II}(\mathbf{III})^{-1}(\mathbf{y}_i - \mathbf{X}_i\widehat{\boldsymbol{\beta}}_j) = \sum_{i=1}^{m} \mathbf{X}_i^{\mathrm{T}}(\mathbf{y}_i - \mathbf{X}_i\widehat{\boldsymbol{\beta}}_j) = \mathbf{X}^{\mathrm{T}}\widehat{s}_j\end{aligned}$$

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 saved 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).

$$\boldsymbol{\alpha} = \sum_{i=1}^{m} \left(\frac{\sum_{j=1}^{n_i} \widehat{r}_{i,j}^2}{n_i} \;,\; \frac{\sum_{j=1}^{n_i-1} \widehat{r}_{i,j}\widehat{r}_{i,j+1}}{n_i} \;,\; \cdots \;,\; \frac{\sum_{j=1}^{n_i-g} \widehat{r}_{i,j}\widehat{r}_{i,j+g}}{n_i} \right) \Bigg/ \left(\sum_{i=1}^{m} \frac{\sum_{j=1}^{n_i} \widehat{r}_{i,j}^2}{n_i} \right)$$

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

$$I(i,p,q) = \begin{cases} 1 & \text{if panel } i \text{ has valid observations at times p and q} \\ 0 & \text{otherwise} \end{cases}$$

where $N_{i,j} = \min(N_i, N_j)$, N_i = number of panels observed at time i, and $n = \max(n_1, n_2, \ldots, n_m)$.

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

Cui, J. 2007. QIC program and model selection in GEE analyses. *Stata Journal* 7: 209–220.

Hardin, J. W., and J. M. Hilbe. 2003. *Generalized Estimating Equations*. Boca Raton, FL: Chapman & Hall/CRC.

Hosmer, D. W., Jr., and S. Lemeshow. 2000. *Applied Logistic Regression*. 2nd ed. New York: Wiley.

Kleinbaum, D. G., and M. Klein. 2010. *Logistic Regression: A Self-Learning Text*. 3rd ed. New York: Springer.

Liang, K.-Y. 1987. Estimating functions and approximate conditional likelihood. *Biometrika* 4: 695–702.

Liang, K.-Y., and S. L. Zeger. 1986. Longitudinal data analysis using generalized linear models. *Biometrika* 73: 13–22.

Liang, K.-Y., S. L. Zeger, and B. Qaqish. 1992. Multivariate regression analyses for categorical data. *Journal of the Royal Statistical Society, Series B* 54: 3–40.

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

Nelder, J. A., and R. W. M. Wedderburn. 1972. Generalized linear models. *Journal of the Royal Statistical Society, Series A* 135: 370–384.

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.

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.

Rabe-Hesketh, S., A. Pickles, and C. Taylor. 2000. sg129: Generalized linear latent and mixed models. *Stata Technical Bulletin* 53: 47–57. Reprinted in *Stata Technical Bulletin Reprints*, vol. 9, pp. 293–307. College Station, TX: Stata Press.

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., 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. 2003. *Applied Longitudinal Data Analysis for Epidemiology: A Practical Guide*. Cambridge: Cambridge University Press.

Wedderburn, R. W. M. 1974. Quasi-likelihood functions, generalized linear models, and the Gauss–Newton method. *Biometrika* 61: 439–447.

Zeger, S. L., and K.-Y. Liang. 1986. Longitudinal data analysis for discrete and continuous outcomes. *Biometrics* 42: 121–130.

Zeger, S. L., K.-Y. Liang, and P. S. Albert. 1988. Models for longitudinal data: A generalized estimating equation approach. *Biometrics* 44: 1049–1060.

Zhao, L. P., and R. L. Prentice. 1990. Correlated binary regression using a quadratic exponential model. *Biometrika* 77: 642–648.

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** — Fixed-, between-, and random-effects and population-averaged linear models

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

[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**

Title

xtgee postestimation — Postestimation tools for xtgee

Description

The following postestimation command is of special interest after `xtgee`:

Command	Description
`estat wcorrelation`	estimated matrix of the within-group correlations

For information about `estat wcorrelation`, see below.

The following standard postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat`	VCE and estimation sample summary
`estimates`	cataloging estimation results
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`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 coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

See the corresponding entries in the *Base Reference Manual* for details.

Special-interest postestimation commands

`estat wcorrelation` displays the estimated matrix of the within-group correlations.

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 $\Pr(y_j = n)$ for `family(poisson) link(log)`
`pr(`*a*`,`*b*`)`	probability $\Pr(a \leq y_j \leq b)$ 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 $\mathbf{x}_j\boldsymbol{\beta}$

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

Menu

Statistics > Postestimation > Predictions, residuals, etc.

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 $\Pr(y_j = n)$ 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 $\Pr(a \leq y_j \leq b)$ 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 $\Pr(y_j \geq 20)$;
`pr(20,`*b*`)` calculates $\Pr(y_j \geq 20)$ in observations for which $b \geq .$ and calculates $\Pr(20 \leq y_j \leq b)$ 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, $u_j = \partial \ln L_j(\mathbf{x}_j\boldsymbol{\beta})/\partial(\mathbf{x}_j\boldsymbol{\beta})$.

`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`.

Syntax for estat wcorrelation

```
estat wcorrelation [ , compact format(%fmt) ]
```

Menu

Statistics > Postestimation > Reports and statistics

Options for estat wcorrelation

`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

▷ Example 1

`xtgee` can estimate rich correlation structures. In example 2 of [XT] **xtgee**, we fit the model

```
. use http://www.stata-press.com/data/r12/nlswork2
(National Longitudinal Survey.  Young Women 14-26 years of age in 1968)
. xtgee ln_w grade age c.age#c.age
  (output omitted)
```

After estimation, `estat wcorrelation` reports the working correlation matrix $\mathbf{R}$:

```
. 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 0.5140.) 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(unstr) nolog
GEE population-averaged model                   Number of obs      =     16085
Group and time vars:           idcode year      Number of groups   =      3913
Link:                             identity      Obs per group: min =         1
Family:                           Gaussian                     avg =       4.1
Correlation:                  unstructured                     max =         9
                                                Wald chi2(3)       =   2405.20
Scale parameter:                  .1418513      Prob > chi2        =    0.0000
```

ln_wage	Coef.	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 http://www.stata-press.com/data/r12/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      =     26200
Group variable:                       idcode    Number of groups   =      4434
Link:                                 probit    Obs per group: min =         1
Family:                             binomial                   avg =       5.9
Correlation:                   exchangeable                    max =        12
                                                Wald chi2(6)       =    242.57
Scale parameter:                           1    Prob > chi2        =    0.0000
```

union	Coef.	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(unstr) nolog

GEE population-averaged model                   Number of obs      =     26200
Group and time vars:           idcode year      Number of groups   =      4434
Link:                               probit      Obs per group: min =         1
Family:                           binomial                     avg =       5.9
Correlation:                 unstructured                      max =        12
                                                Wald chi2(6)       =    198.45
Scale parameter:                         1      Prob > chi2        =    0.0000
```

union	Coef.	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 http://www.stata-press.com/data/r12/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
Link:                                    log    Obs per group: min =         4
Family:                              Poisson                   avg =       4.0
Correlation:                     exchangeable                   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)				

```
. 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(unstr) nolog
GEE population-averaged model                   Number of obs      =        80
Group and time vars:           airline time     Number of groups   =        20
Link:                                    log    Obs per group: min =         4
Family:                              Poisson                   avg =       4.0
Correlation:                   unstructured                    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)				

```
. 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 http://www.stata-press.com/data/r12/airacc2, clear
. xtgee i_cnt inprog, family(poisson) eform offset(lnpm) nolog
GEE population-averaged model                   Number of obs      =      1600
Group variable:                      airline    Number of groups   =       400
Link:                                    log    Obs per group: min =         4
Family:                              Poisson                   avg =       4.0
Correlation:                    exchangeable                   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)				

```
. 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(unstr) eform offset(lnpm) nolog
GEE population-averaged model                   Number of obs      =      1600
Group and time vars:            airline time    Number of groups   =       400
Link:                                    log    Obs per group: min =         4
Family:                              Poisson                   avg =       4.0
Correlation:                    unstructured                   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)				

```
. 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.

◁

Methods and formulas

All postestimation commands listed above are implemented as ado-files.

Also see

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

[U] **20 Estimation and postestimation commands**

Title

xtgls — Fit panel-data models by using GLS

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 column formats, row spacing, line width, and display of omitted variables and base and empty cells
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` 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.

Menu

Statistics > Longitudinal/panel data > Contemporaneous correlation > GLS regression with correlated disturbances

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.

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`; see [R] **estimation options**.

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 (2012, 280) 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: noomitted, vsquish, noemptycells, baselevels, allbaselevels, 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.

nolog suppresses display of the iteration log.

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

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

Remarks

Remarks are presented under the following headings:

Introduction

Information on GLS can be found in Greene (2012), 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

$$\begin{aligned} E[\epsilon_{i,t}] &= 0 \\ \mathrm{Var}[\epsilon_{i,t}] &= \sigma^2 \\ \mathrm{Cov}[\epsilon_{i,t}, \epsilon_{j,s}] &= 0 \qquad \text{if } t \neq s \text{ or } i \neq j \end{aligned}$$

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 http://www.stata-press.com/data/r12/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	Coef.	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 |      Coef.   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 |      Coef.   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 Ω 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	Coef.	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	Coef.	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

◁

Saved results

`xtgls` saves the following in `e()`:

Scalars

`e(N)`	number of observations
`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_pear)`	degrees of freedom for Pearson χ^2
`e(ll)`	log likelihood
`e(chi2)`	χ^2
`e(df)`	degrees of freedom
`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

Methods and formulas

xtgls is implemented as an ado-file.

The GLS results are given by

$$\widehat{\boldsymbol{\beta}}_{\text{GLS}} = (\mathbf{X}'\widehat{\boldsymbol{\Omega}}^{-1}\mathbf{X})^{-1}\mathbf{X}'\widehat{\boldsymbol{\Omega}}^{-1}\mathbf{y}$$
$$\widehat{\text{Var}}(\widehat{\boldsymbol{\beta}}_{\text{GLS}}) = (\mathbf{X}'\widehat{\boldsymbol{\Omega}}^{-1}\mathbf{X})^{-1}$$

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., 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.

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.

Grunfeld, Y., and Z. Griliches. 1960. Is aggregation necessarily bad? *Review of Economics and Statistics* 42: 1–13.

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

[XT] **xtpcse** — Linear regression with panel-corrected standard errors

[XT] **xtreg** — Fixed-, between-, and random-effects and population-averaged linear models

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

[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**

Title

xtgls postestimation — Postestimation tools for xtgls

Description

The following postestimation commands are available after `xtgls`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat`[1]	AIC, BIC, VCE, and estimation sample summary
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`lrtest`[2]	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 coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

[1] AIC and BIC are available only if `igls` and `corr(independent)` were specified at estimation.

[2] Likelihood-ratio tests are available only if `igls` and `corr(independent)` were specified at estimation.

See the corresponding entries in the *Base Reference Manual* for details.

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.

Menu

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

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

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

Also see

[XT] **xtgls** — Fit panel-data models by using GLS

[U] **20 Estimation and postestimation commands**

Title

xthtaylor — Hausman–Taylor estimator for error-components models

Syntax

`xthtaylor` *depvar indepvars* [*if*] [*in*] [*weight*] , `endog(`*varlist*`)` [*options*]

options	Description
Main	
`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	
`vce(`*vcetype*`)`	*vcetype* may be `conventional`, `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`, `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.

Menu

Statistics > Longitudinal/panel data > Endogenous covariates > Hausman-Taylor regression (RE)

Description

`xthtaylor` fits panel-data random-effects models in which some of the covariates are correlated with the unobserved individual-level random effect. The estimators, originally proposed by Hausman and Taylor (1981) and by Amemiya and MaCurdy (1986), are based on instrumental variables. By default, `xthtaylor` uses the Hausman–Taylor estimator. When the `amacurdy` option is specified, `xthtaylor` uses the Amemiya–MaCurdy estimator.

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}.

Options

Main

`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*$_{\rm ti}$`)` 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*$_{\rm ti}$ 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*$_{\rm tv}$`)` 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*$_{\rm tv}$ 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 issues 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

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

`vce(conventional)`, the default, uses the conventionally derived variance estimator for this Hausman–Taylor model.

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

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 independently 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

$i = 1, \ldots, n$, where n is the number of panels in the sample and, for each i, $t = 1, \ldots, T_i$.

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 $\widehat{\boldsymbol{\delta}}_{1\mathrm{IV}}$ and $\widehat{\boldsymbol{\delta}}_{2\mathrm{IV}}$, 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 $\widehat{\boldsymbol{\delta}}_{1\mathrm{IV}}$ and $\widehat{\boldsymbol{\delta}}_{2\mathrm{IV}}$ 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

$$\mathrm{plim}_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \overline{\mathbf{X}}_{1i.}\mu_i = 0$$

and

$$\mathrm{plim}_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \mathbf{Z}_{1i}\mu_i = 0$$

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, $\text{plim}_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} \mathbf{X}_{1it}\mu_i = 0$ for $t = 1, \ldots, T$. 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.

▷ 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 http://www.stata-press.com/data/r12/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=4165)
```

	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      =      4165
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	Coef.	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      =      4165
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	Coef.	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 http://www.stata-press.com/data/r12/xthtaylor1
. correlate ui z1 z2 x1a x1b x2 eit
(obs=10000)
```

	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      =     10000
Group variable: id                           Number of groups   =      1000
                                             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	Coef.	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*$_{\text{ti}}$`)` 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 http://www.stata-press.com/data/r12/xthtaylor2
. xthtaylor yit x1a x1b x1c x2 z1 z2, endog(x2 z2)
Hausman-Taylor estimation                       Number of obs      =     10000
Group variable: id                              Number of groups   =      1000
                                                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	Coef.	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      =      8000
Group variable: id                              Number of groups   =      1000
                                                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	Coef.	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.

◁

Saved results

`xthtaylor` saves 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(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(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

Functions

`e(sample)`	marks estimation sample

Methods and formulas

`xthtaylor` is implemented as an ado-file.

Consider an error-components model of the form

$$y_{it} = \mathbf{X}_{1it}\boldsymbol{\beta}_1 + \mathbf{X}_{2it}\boldsymbol{\beta}_2 + \mathbf{Z}_{1i}\delta_1 + \mathbf{Z}_{2i}\delta_2 + \mu_i + \epsilon_{it} \tag{2}$$

for $i = 1, \ldots, n$ and, for each i, $t = 1, \ldots, T_i$, 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

$$\widetilde{w}_{it} = w_{it} - \overline{w}_{i.} \qquad \overline{w}_{i.} = \frac{1}{n}\sum_{t=1}^{T_i} w_{it}$$

Because the within estimator removes $\mathbf{Z}$, the within transformation reduces the model to

$$\widetilde{y}_{it} = \widetilde{\mathbf{X}}_{1it}\boldsymbol{\beta}_1 + \widetilde{\mathbf{X}}_{2it}\boldsymbol{\beta}_2 + \widetilde{\epsilon}_{it}$$

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 be 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

$$s^2 = \frac{1}{N}\sum_{i=1}^{n}\sum_{t=1}^{T_i}\left(\frac{1}{T_i}\sum_{t=1}^{T_i}\widehat{e}_i\right)^2$$

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,

$$\text{plim}_{n\to\infty} s^2 = \overline{T}\sigma_\mu^2 + \sigma_\epsilon^2$$

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

$$w_{it}^* = w_{it} - \widehat{\theta}_i \overline{w}_{i\cdot}$$

where $\overline{w}_{i\cdot}$ 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 $\widetilde{\mathbf{X}}_{it}$, $\overline{\mathbf{X}}_{1i\cdot}$, 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.

Baltagi, B. H. 2008. *Econometric Analysis of Panel Data*. 4th ed. New York: Wiley.

——. 2009. *A Companion to Econometric Analysis of Panel Data*. 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.

Hausman, J. A., and W. E. Taylor. 1981. Panel data and unobservable individual effects. *Econometrica* 49: 1377–1398.

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.

Also see

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

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

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

[XT] **xtreg** — Fixed-, between-, and random-effects and population-averaged linear models

[U] **20 Estimation and postestimation commands**

Title

xthtaylor postestimation — Postestimation tools for xthtaylor

Description

The following postestimation commands are available after `xthtaylor`:

Command	Description
`estat`	VCE and estimation sample summary
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`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 coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`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

See the corresponding entries in the *Base Reference Manual* for details.

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.

Menu

Statistics > Postestimation > Predictions, residuals, etc.

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 $\widehat{\epsilon}_{it}$.

Methods and formulas

All postestimation commands listed above are implemented as ado-files.

Also see

[XT] **xthtaylor** — Hausman–Taylor estimator for error-components models

[U] **20 Estimation and postestimation commands**

Title

xtintreg — Random-effects interval-data regression models

Syntax

`xtintreg` *depvar*$_{\text{lower}}$ *depvar*$_{\text{upper}}$ [*indepvars*] [*if*] [*in*] [*weight*] [`,` *options*]

options	Description
Model	
`noconstant`	suppress constant term
`offset(`*varname*`)`	include *varname* in model with coefficient constrained to 1
`constraints(`*constraints*`)`	apply specified linear constraints
`collinear`	keep collinear variables
SE	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`noskip`	perform overall model test as a likelihood-ratio test
`intreg`	perform likelihood-ratio test against pooled model
`nocnsreport`	do not display constraints
display_options	control column formats, row spacing, line width, and display of omitted variables and base and empty cells
Integration	
`intmethod(`*intmethod*`)`	integration method; *intmethod* may be `mvaghermite`, `aghermite`, or `ghermite`; default is `intmethod(mvaghermite)`
`intpoints(#)`	use # quadrature points; default is `intpoints(12)`
Maximization	
maximize_options	control the maximization process; see [R] **maximize**
`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` and `statsby` are allowed; see [U] **11.1.10 Prefix commands**.
`iweights` are allowed; 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.

Menu

Statistics > Longitudinal/panel data > Censored outcomes > Interval regression (RE)

Description

xtintreg fits a random-effects regression model whose dependent variable may be measured as point data, interval data, left-censored data, or right-censored data. *depvar*$_{\text{lower}}$ and *depvar*$_{\text{upper}}$ represent how the dependent variable was measured.

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	.

Options

Model

noconstant, **offset(***varname***)**, **constraints(***constraints***)**, **collinear**; see [R] **estimation options**.

SE

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

Reporting

level(#), **noskip**; 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: **noomitted**, **vsquish**, **noemptycells**, **baselevels**, **allbaselevels**, **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 option is available with **xtintreg** but is not shown in the dialog box:

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

Remarks

Consider the linear regression model with panel-level random effects

$$y_{it} = \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i + \epsilon_{it}$$

for $i = 1, \ldots, n$ panels, where $t = 1, \ldots, n_i$. 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 $y_{1it} \leq y_{it} \leq y_{2it}$, where y_{1it} is possibly $-\infty$ and y_{2it} is possibly $+\infty$.

▷ Example 1

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:

```
. use http://www.stata-press.com/data/r12/nlswork5
(National Longitudinal Survey.  Young Women 14-26 years of age in 1968)
. xtintreg ln_wage1 ln_wage2 union age grade south##c.year occ_code, intreg
  (output omitted)
Random-effects interval regression              Number of obs      =     19151
Group variable: idcode                          Number of groups   =      4140
Random effects u_i ~ Gaussian                   Obs per group: min =         1
                                                               avg =       4.6
                                                               max =        12
                                                Wald chi2(7)       =   2523.84
Log likelihood  = -23174.355                    Prob > chi2        =    0.0000
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
union	.1441844	.0094245	15.30	0.000	.1257128	.162656
age	.0104083	.0018804	5.54	0.000	.0067228	.0140939
grade	.0794958	.0023469	33.87	0.000	.074896	.0840955
1.south	-.3778103	.0979415	-3.86	0.000	-.5697722	-.1858485
year	.0013528	.0020176	0.67	0.503	-.0026016	.0053071
south#c.year						
1	.0034385	.0012105	2.84	0.005	.0010659	.005811
occ_code	-.0197912	.0014094	-14.04	0.000	-.0225535	-.0170289
_cons	.3791078	.1136641	3.34	0.001	.1563303	.6018853
/sigma_u	.2987074	.0052697	56.68	0.000	.2883789	.309036
/sigma_e	.3528109	.0030935	114.05	0.000	.3467478	.358874
rho	.4175266	.0102529			.3975474	.4377211

```
Likelihood-ratio test of sigma_u=0: chibar2(01)= 2516.85 Prob>=chibar2 = 0.000
  Observation summary:      4757  left-censored observations
                            4792     uncensored observations
                            4830 right-censored observations
                            4772       interval observations
```

The output 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 (intreg) 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. 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.

❑

Saved results

xtintreg saves 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(N_cd)	number of completely determined 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(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)	significance
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(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(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(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

Methods and formulas

xtintreg is implemented as an ado-file.

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

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

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

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

$$\equiv \int_{-\infty}^{\infty} g(y_{1it}, y_{2it}, x_{it}, \nu_i) d\nu_i$$

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

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

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.

One can instead use the adaptive quadrature method of Liu and Pierce (1994), the `intmethod(aghermite)` option, which uses the mode and curvature of the mode as approximations for the mean and variance. We take the integrand

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

and find α_i the mode of $g(y_{1it}, y_{2it}, x_{it}, \nu_i)$. We calculate

$$\gamma_i = -\frac{\partial^2}{\partial \nu^2} \log\{g(y_{1it}, y_{2it}, x_{it}, \nu_i)\}\big|_{\nu_i = \alpha_i}$$

Then

$$\int_{-\infty}^{\infty} g(y_{1it}, y_{2it}, x_{it}, \nu_i) d\nu_i \approx \left(\frac{2}{\gamma_i}\right)^{1/2} \sum_{m=1}^{M} w_m^* \exp\{(a_m^*)^2\} g\left\{y_{1it}, y_{2it}, x_{it}, \left(\frac{2}{\gamma_i}\right)^{1/2} a_m^* + \alpha_i\right\}$$

This adaptation is performed on the first iteration only; that is, the α_i and γ_i are calculated once at the first iteration and then held constant throughout the subsequent iterations.

The log likelihood can also be calculated by nonadaptive Gauss–Hermite quadrature, the `intmethod(ghermite)` option:

$$L = \sum_{i=1}^{n} w_i \log f\left\{(y_{1i1}, y_{2i1}), \ldots, (y_{1in_i}, y_{2in_i}) | \mathbf{x}_{1i}, \ldots, \mathbf{x}_{in_i}\right\}$$
$$\approx \sum_{i=1}^{n} w_i \log\left\{\frac{1}{\sqrt{\pi}} \sum_{m=1}^{M} w_m^* \prod_{t=1}^{n_i} F\left(y_{1it}, y_{2it}, \mathbf{x}_{it}\boldsymbol{\beta} + \sqrt{2}\sigma_\nu a_m^*\right)\right\}$$

All three 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

$$\prod_{t=1}^{n_i} F(y_{1it}, y_{2it}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i)$$

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

Liu, Q., and D. A. Pierce. 1994. A note on Gauss–Hermite quadrature. *Biometrika* 81: 624–629.

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, Series C* 31: 214–225.

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

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.

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

Also see

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

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

[XT] **xtreg** — Fixed-, between-, and random-effects and population-averaged linear models

[XT] **xttobit** — Random-effects tobit models

[R] **intreg** — Interval regression

[R] **tobit** — Tobit regression

[U] **20 Estimation and postestimation commands**

Title

xtintreg postestimation — Postestimation tools for xtintreg

Description

The following postestimation commands are available after `xtintreg`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat`	AIC, BIC, VCE, and estimation sample summary
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`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 coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

See the corresponding entries in the *Base Reference Manual* for details.

Syntax for predict

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

statistic	Description
Main	
`xb`	linear prediction assuming a zero random effect, the default
`stdp`	standard error of the linear prediction
`stdf`	standard error of the linear forecast
`pr0(`*a*`,`*b*`)`	$\Pr(a < y < b)$ assuming a zero random effect
`e0(`*a*`,`*b*`)`	$E(y \mid a < y < b)$ assuming a zero random effect
`ystar0(`*a*`,`*b*`)`	$E(y^*)$, $y^* = \max\{a, \min(y_j, b)\}$ assuming a zero 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**.

Menu

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

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

`stdp` calculates the standard error of the 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 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**.

`pr0(`*a*`,`*b*`)` calculates estimates of $\Pr(a < y < b|\mathbf{x} = \mathbf{x}_{it}, \nu_i = 0)$, which is the probability that y would be observed in the interval (a, b), given the current values of the predictors, $\mathbf{x}_{it}$, and given a zero random effect. 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;
`pr0(20,30)` calculates $\Pr(20 < y < 30)$;
`pr0(`*lb*`,`*ub*`)` calculates $\Pr(lb < y < ub)$; and
`pr0(20,`*ub*`)` calculates $\Pr(20 < y < ub)$.

a missing ($a \geq .$) means $-\infty$; `pr0(.,30)` calculates $\Pr(-\infty < y < 30)$;
`pr0(`*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 $+\infty$; `pr0(20,.)` calculates $\Pr(+\infty > y > 20)$;
`pr0(20,`*ub*`)` calculates $\Pr(+\infty > y > 20)$ in observations for which $ub \geq .$
(and calculates $\Pr(20 < y < ub)$ elsewhere).

`e0(`*a*`,`*b*`)` calculates estimates of $E(y \mid a < y < b, \mathbf{x} = \mathbf{x}_{it}, \nu_i = 0)$, 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 `pr0()`.

`ystar0(`*a*`,`*b*`)` calculates estimates of $E(y^*|\mathbf{x} = \mathbf{x}_{it}, \nu_i = 0)$, 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 `pr0()`.

`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 $\mathbf{x}_{it}\boldsymbol{\beta} + \text{offset}_{it}$.

Remarks

▷ Example 1

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 http://www.stata-press.com/data/r12/nlswork5
(National Longitudinal Survey.  Young Women 14-26 years of age in 1968)
. xtintreg ln_wage1 ln_wage2 i.union age grade south##c.year, intreg
  (output omitted)
. centile ln_wage, centile(20)
                                                      -- Binom. Interp. --
    Variable |      Obs  Percentile      Centile        [95% Conf. Interval]
-------------+-------------------------------------------------------------
     ln_wage |    28534         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, with the exception that they are college graduates (`grade=16`).

```
. margins, dydx(union) predict(pr0(.,1.30))
> at(age=30 south=1 year=88 grade=12 union=0)
> at(age=30 south=1 year=88 grade=16 union=0)
Conditional marginal effects                      Number of obs   =      19224
Model VCE    : OIM
Expression   : Pr(ln_wage1<1.30), predict(pr0(.,1.30))
dy/dx w.r.t. : 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]
-------------+----------------------------------------------------------------
1.union      |
         _at |
          1  |  -.0787117   .0060655   -12.98   0.000    -.0905999   -.0668235
          2  |  -.0378758   .0035595   -10.64   0.000    -.0448523   -.0308993
------------------------------------------------------------------------------
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 7.9 percentage points. Being a college graduate attenuates this effect to just under 3.8 percentage points.

◁

Methods and formulas

All postestimation commands listed above are implemented as ado-files.

Also see

[XT] **xtintreg** — Random-effects interval-data regression models

[U] **20 Estimation and postestimation commands**

Title

xtivreg — Instrumental variables and two-stage least squares for panel-data models

Syntax

GLS random-effects (RE) model

```
xtivreg depvar [varlist1] (varlist2 = varlistiv) [if] [in] [, re RE_options]
```

Between-effects (BE) model

```
xtivreg depvar [varlist1] (varlist2 = varlistiv) [if] [in], be [BE_options]
```

Fixed-effects (FE) model

```
xtivreg depvar [varlist1] (varlist2 = varlistiv) [if] [in], fe [FE_options]
```

First-differenced (FD) estimator

```
xtivreg depvar [varlist1] (varlist2 = varlistiv) [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	
`vce(`*vcetype*`)`	*vcetype* may be `conventional`, `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 column formats, row spacing, line width, and display of omitted variables and base and empty cells
`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	
`vce(`*vcetype*`)`	*vcetype* may be `conventional`, `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 column formats, row spacing, line width, and display of omitted variables and base and empty cells
`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	
`vce(`*vcetype*`)`	*vcetype* may be `conventional`, `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 column formats, row spacing, line width, and display of omitted variables and base and empty cells
`coeflegend`	display legend instead of statistics

FD_options	Description
Model	
noconstant	suppress constant term
fd	first-differenced estimator
regress	treat covariates as exogenous and ignore instrumental variables
SE	
vce(*vcetype*)	*vcetype* may be conventional, 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 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*$_{\text{iv}}$ may contain factor variables, except for the fd estimator; see [U] **11.4.3 Factor variables**.

depvar, *varlist*$_1$, *varlist*$_2$, and *varlist*$_{\text{iv}}$ may contain time-series operators; see [U] **11.4.4 Time-series varlists**.

by 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.

Menu

Statistics > Longitudinal/panel data > Endogenous covariates > Instrumental-variables regression (FE, RE, BE, FD)

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 (2008) 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).

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

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory and that use bootstrap or jackknife methods; 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**.

`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 chi-squared 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: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `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 are to be treated as exogenous and that the instrument list is to 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

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory and that use bootstrap or jackknife methods; 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.

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 chi-squared statistics.

display_options: noomitted, vsquish, noemptycells, baselevels, allbaselevels, 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 are to be treated as exogenous and that the instrument list is to 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

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory and that use bootstrap or jackknife methods; 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.

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 chi-squared statistics.

display_options: noomitted, vsquish, noemptycells, baselevels, allbaselevels, 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 are to be treated as exogenous and that the instrument list is to 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

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory and that use bootstrap or jackknife methods; 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**.

`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 chi-squared statistics.

display_options: `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

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

Consider an equation of the form

$$y_{it} = \mathbf{Y}_{it}\boldsymbol{\gamma} + \mathbf{X}_{1it}\boldsymbol{\beta} + \mu_i + \nu_{it} = \mathbf{Z}_{it}\boldsymbol{\delta} + \mu_i + \nu_{it} \tag{1}$$

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;

$\mathbf{Z}_{it} = [\mathbf{Y}_{it}\ \mathbf{X}_{it}]$;

$\boldsymbol{\gamma}$ 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, $N = \sum_{i=1}^{n} T_i$.

`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 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 (2003) 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 http://www.stata-press.com/data/r12/nlswork
(National Longitudinal Survey.  Young Women 14-26 years of age in 1968)

. xtreg ln_w age c.age#c.age tenure not_smsa union south, fe

Fixed-effects (within) regression               Number of obs      =     19007
Group variable: idcode                          Number of groups   =      4134

R-sq:  within  = 0.1333                         Obs per group: 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 |      Coef.   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      =        19007
Group variable: idcode                       Number of groups   =         4134

R-sq:  within  =      .                      Obs per group: 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 |      Coef.   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
------------------------------------------------------------------------------
Instrumented:   tenure
Instruments:    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      =     19007
Group variable: idcode                          Number of groups   =      4134

R-sq:  within  = 0.0664                         Obs per group: 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	Coef.	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
2.race	-.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)				

```
Instrumented:   tenure
Instruments:    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      =     19007
Group variable: idcode                          Number of groups   =      4134

R-sq:  within  = 0.0898                         Obs per group: 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	Coef.	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
2.race	-.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)				

```
Instrumented:   tenure
Instruments:    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      =     19007
Group variable: idcode                          Number of groups   =      4134

R-sq:  within  = 0.0664                         Obs per group: 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	Coef.	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
2.race	-.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)				

```
Instrumented:   tenure
Instruments:    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 different because Arellano and Bond are using robust standard errors.

```
. use http://www.stata-press.com/data/r12/abdata
. xtivreg n l2.n l(0/1).w l(0/2).(k ys) yr1981-yr1984 (l.n = l3.n), fd
First-differenced IV regression
Group variable: id                              Number of obs      =       471
Time variable: year                             Number of groups   =       140
R-sq:  within  = 0.0141                         Obs per group: min =         3
       between = 0.9165                                        avg =       3.4
       overall = 0.9892                                        max =         5
                                                Wald chi2(14)      =    122.53
corr(u_i, Xb)  = 0.9239                         Prob > chi2        =    0.0000
```

D.n	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
n						
LD.	1.422765	1.583053	0.90	0.369	-1.679962	4.525493
L2D.	-.1645517	.1647179	-1.00	0.318	-.4873928	.1582894
w						
D1.	-.7524675	.1765733	-4.26	0.000	-1.098545	-.4063902
LD.	.9627611	1.086506	0.89	0.376	-1.166752	3.092275
k						
D1.	.3221686	.1466086	2.20	0.028	.0348211	.6095161
LD.	-.3248778	.5800599	-0.56	0.575	-1.461774	.8120187
L2D.	-.0953947	.1960883	-0.49	0.627	-.4797207	.2889314
ys						
D1.	.7660906	.369694	2.07	0.038	.0415037	1.490678
LD.	-1.361881	1.156835	-1.18	0.239	-3.629237	.9054744
L2D.	.3212993	.5440403	0.59	0.555	-.745	1.387599
yr1981						
D1.	-.0574197	.0430158	-1.33	0.182	-.1417291	.0268896
yr1982						
D1.	-.0882952	.0706214	-1.25	0.211	-.2267106	.0501203
yr1983						
D1.	-.1063153	.10861	-0.98	0.328	-.319187	.1065563
yr1984						
D1.	-.1172108	.15196	-0.77	0.441	-.4150468	.1806253
_cons	.0161204	.0336264	0.48	0.632	-.0497861	.082027
sigma_u	.29069213					
sigma_e	.18855982					
rho	.70384993	(fraction of variance due to u_i)				

```
Instrumented:   L.n
Instruments:    L2.n w L.w k L.k L2.k ys L.ys L2.ys yr1981 yr1982 yr1983 yr1984
                L3.n
```

Saved results

`xtivreg, re` saves 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(sigma)`	ancillary parameter (`gamma`, `lnormal`)
`e(sigma_u)`	panel-level standard deviation
`e(sigma_e)`	standard deviation of ϵ_{it}
`e(r2_w)`	R-squared for within model
`e(r2_o)`	R-squared for overall model
`e(r2_b)`	R-squared 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(Tbar)`	harmonic mean of group sizes
`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(insts)`	instruments
`e(instd)`	instrumented variables
`e(model)`	`g2sls` or `ec2sls`
`e(small)`	`small`, if specified
`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

Functions

e(sample)	marks estimation sample

xtivreg, be saves 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-squared for within model
e(r2_o)	R-squared for overall model
e(r2_b)	R-squared for between model
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(insts)	instruments
e(instd)	instrumented variables
e(model)	be
e(small)	small, if specified
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

Functions

e(sample)	marks estimation sample

`xtivreg, fe` saves 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(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-squared for within model
`e(r2_o)`	R-squared for overall model
`e(r2_b)`	R-squared for between model
`e(chi2)`	model Wald (not `small`)
`e(df_b)`	degrees of freedom for χ^2 statistic
`e(chi2_p)`	p-value for model χ^2 statistic
`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(insts)`	instruments
`e(instd)`	instrumented variables
`e(model)`	`fe`
`e(small)`	`small`, if specified
`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

Functions

`e(sample)`	marks estimation sample

`xtivreg, fd` saves the following in `e()`:

Scalars

`e(N)`	number of observations
`e(N_g)`	number of groups
`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(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-squared for within model
`e(r2_o)`	R-squared for overall model
`e(r2_b)`	R-squared for between model
`e(chi2)`	model Wald (not `small`)
`e(df_b)`	degrees of freedom for the χ^2 statistic
`e(chi2_p)`	p-value for model χ^2 statistic
`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(insts)`	instruments
`e(instd)`	instrumented variables
`e(model)`	`fd`
`e(small)`	`small`, if specified
`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

Functions

`e(sample)`	marks estimation sample

Methods and formulas

`xtivreg` is implemented as an ado-file.

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

$$y_{it} - y_{it-1} = (\mathbf{Z}_{it} - \mathbf{Z}_{i,t-1})\,\delta + \nu_{it} - \nu_{i,t-1}$$

With the μ_i removed by differencing, we can obtain the estimated coefficients and their estimated variance–covariance matrix from a standard 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 $\left[\text{corr}\left\{(\mathbf{Z}_{it} - \overline{\mathbf{Z}}_i)\widehat{\boldsymbol{\delta}}, y_{it} - \overline{y}_i\right\}\right]^2$.

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

$$\overline{w}_{i.} = \frac{1}{n}\sum_{t=1}^{T_i} w_{it}$$

$$\overline{w} = \frac{1}{N}\sum_{i=1}^{n}\sum_{t=1}^{T_i} w_{it}$$

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.

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 statistic 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

$$\overline{w}_i = \frac{1}{T_i}\sum_{t=1}^{T_i} w_{it} \quad \text{for } w \in \{y, \mathbf{Z}, \nu\}$$

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 conventional 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 $\left[\text{corr}\left\{(\mathbf{Z}_{it}-\overline{\mathbf{Z}}_i)\widehat{\boldsymbol{\delta}}, y_{it}-\overline{y}_i\right\}\right]^2$.

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

$$u_{it}^w = \widetilde{y}_{it} - \widetilde{\mathbf{Z}}_{it}\widehat{\boldsymbol{\delta}}_w$$

be the combined residuals from the within estimator. Let $\widetilde{u}_{it}$ be the within-transformed u_{it}. Then

$$\widehat{\sigma}_\nu = \frac{\sum_{i=1}^{n}\sum_{t=1}^{T_i}\widetilde{u}_{it}^2}{N-n-K+1}$$

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 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 conventional 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 $\widetilde{\mathbf{X}}_i t$ 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 $\sqrt{\widehat{\sigma}_\mu^2 + \widehat{\sigma}_\nu^2}$.

R^2 between is reported as $\left\{\text{corr}(\overline{\mathbf{Z}}_i\widehat{\boldsymbol{\delta}}, \overline{y}_i)\right\}^2$.

R^2 within is reported as $\left[\text{corr}\left\{(\mathbf{Z}_{it} - \overline{\mathbf{Z}}_i)\widehat{\boldsymbol{\delta}}, y_{it} - \overline{y}_i\right\}\right]^2$.

R^2 overall is reported as $\left\{\text{corr}(\mathbf{Z}_{it}\widehat{\boldsymbol{\delta}}, y_{it})\right\}^2$.

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.

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

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.

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.

Baltagi, B. H. 2008. *Econometric Analysis of Panel Data*. 4th ed. New York: Wiley.

——. 2009. *A Companion to Econometric Analysis of Panel Data*. 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.

——. 2000. Simultaneous equations with incomplete panels. *Econometric Theory* 16: 269–279.

Baltagi, B. H., and Q. Li. 1992. A note on the estimation of simultaneous equations with error components. *Econometric Theory* 8: 113–119.

Hsiao, C. 2003. *Analysis of Panel Data*. 2nd ed. New York: Cambridge University Press.

Layard, R., and S. J. Nickell. 1986. Unemployment in Britain. *Economica* 53: S121–S169.

Mundlak, Y. 1978. On the pooling of time series and cross section data. *Econometrica* 46: 69–85.

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.

White, H. 2001. *Asymptotic Theory for Econometricians*. Rev. ed. New York: Academic Press.

Wooldridge, J. M. 1990. A note on the Lagrange multiplier and F-statistics for two stage least squares regressions. *Economics Letters* 34: 151–155.

Also see

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

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

[XT] **xtreg** — Fixed-, between-, and random-effects and population-averaged linear models

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

[XT] **xthtaylor** — Hausman–Taylor estimator for error-components models

[R] **ivregress** — Single-equation instrumental-variables regression

[U] **20 Estimation and postestimation commands**

Title

xtivreg postestimation — Postestimation tools for xtivreg

Description

The following postestimation commands are available after `xtivreg`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat`	VCE and estimation sample summary
`estimates`	cataloging estimation results
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`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 coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

See the corresponding entries in the *Base Reference Manual* for details.

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.

Menu

Statistics > Postestimation > Predictions, residuals, etc.

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 $\widehat{\mu}_i$, the estimated fixed or random effect. This is not available after the first-differenced model.

`e` calculates the prediction of $\widehat{\nu}_{it}$.

Also see

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

[U] **20 Estimation and postestimation commands**

Title

xtline — 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.

Menu

Statistics > Longitudinal/panel data > Line plots

Description

`xtline` draws line plots for panel data.

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

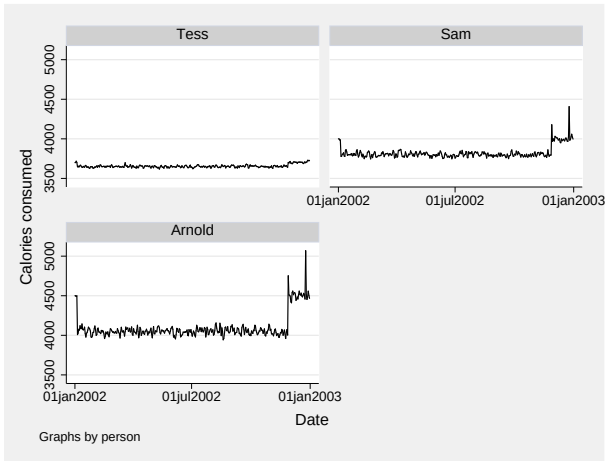
twoway_options are any of the options documented in [G-3] ***twoway_options***, excluding `by()`. These include options for titling the graph (see [G-3] ***title_options***) and for saving the graph to disk (see [G-3] ***saving_option***).

Remarks

▷ Example 1

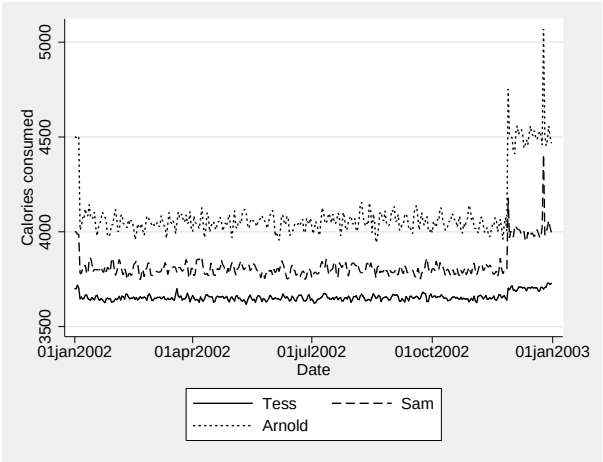
Suppose that Tess, Sam, and Arnold kept a calorie log for an entire calendar year. At the end of the year, if they pooled their data together, they would have a dataset (for example, `xtline1.dta`) that contains the number of calories each of them consumed for 365 days. They could then use `xtset` to 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 http://www.stata-press.com/data/r12/xtline1
. 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
```



Methods and formulas

`xtline` is implemented as an ado-file.

Also see

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

[G-2] **graph twoway** — Twoway graphs

[TS] **tsline** — Plot time-series data

Title

xtlogit — Fixed-effects, random-effects, and population-averaged logit models

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
`collinear`	keep collinear variables
SE	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`or`	report odds ratios
`noskip`	perform overall model test as a likelihood-ratio test
`nocnsreport`	do not display constraints
display_options	control column formats, row spacing, line width, and display of omitted variables and base and empty cells
Integration	
`intmethod(`*intmethod*`)`	integration method; *intmethod* may be `mvaghermite`, `aghermite`, or `ghermite`; default is `intmethod(mvaghermite)`
`intpoints(#)`	use # quadrature points; default is `intpoints(12)`
Maximization	
maximize_options	control the maximization process; seldom used
`nodisplay`	suppress display of header and coefficients
`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
`collinear`	keep collinear variables
SE	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`or`	report odds ratios
`noskip`	perform overall model test as a likelihood-ratio test
`nocnsreport`	do not display constraints
display_options	control column formats, row spacing, line width, and display of omitted variables and base and empty cells
Maximization	
maximize_options	control the maximization process; seldom used
`nodisplay`	suppress display of header and coefficients
`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-group 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 column formats, row spacing, line width, and display of omitted variables and base and empty cells
Optimization	
optimize_options	control the optimization process; seldom used
`nodisplay`	do not display the 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`, `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**.

`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` and `coeflegend` do not appear in the dialog box.

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

Menu

Statistics > Longitudinal/panel data > Binary outcomes > Logistic regression (FE, RE, PA)

Description

`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; `xtlogit, pa` assumes `corr(exchangeable)`. See [XT] **xtgee** for information on how to fit other population-averaged models.

See [R] **logistic** for a list of related 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*`)`, `collinear`; see [R] **estimation options**.

SE

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory and that use bootstrap or jackknife methods; 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.

`noskip`; see [R] **estimation options**.

`nocnsreport`; see [R] **estimation options**.

display_options: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `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.

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

Options for FE model

Model

`fe` requests the fixed-effects estimator.

`offset(`*varname*`)`, `constraints(`*constraints*`)`, `collinear`; see [R] **estimation options**.

SE

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory and that use bootstrap or jackknife methods; 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.

`noskip`; see [R] **estimation options**.

`nocnsreport`; see [R] **estimation options**.

display_options: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `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.

`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**.

Correlation

`corr(`*correlation*`)`, `force`; see [R] **estimation options**.

SE/Robust

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory, that are robust to some kinds of misspecification, and that use bootstrap or jackknife methods; 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: noomitted, vsquish, noemptycells, baselevels, allbaselevels, 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.

nolog suppresses display of the iteration log.

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

xtlogit is a convenience command if you want the population-averaged model. 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 http://www.stata-press.com/data/r12/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      =     26200
Group variable: idcode                          Number of groups   =      4434
Random effects u_i ~ Gaussian                   Obs per group: min =         1
                                                               avg =       5.9
                                                               max =        12
                                                Wald chi2(6)       =    227.46
Log likelihood  = -10540.274                    Prob > chi2        =    0.0000
```

union	Coef.	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

```
Likelihood-ratio 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 = .1487877
Iteration 2: tolerance = .00949342
Iteration 3: tolerance = .00040606
Iteration 4: tolerance = .00001602
Iteration 5: tolerance = 6.628e-07
GEE population-averaged model                 Number of obs      =     26200
Group variable:                      idcode   Number of groups   =      4434
Link:                                 logit   Obs per group: min =         1
Family:                            binomial                 avg =       5.9
Correlation:                   exchangeable                 max =        12
                                              Wald chi2(6)       =    235.08
Scale parameter:                         1    Prob > chi2        =    0.0000
```

union	Coef.	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      =     26200
Group variable:                      idcode   Number of groups   =      4434
Link:                                 logit   Obs per group: min =         1
Family:                            binomial                 avg =       5.9
Correlation:                   exchangeable                 max =        12
                                              Wald chi2(6)       =    154.88
Scale parameter:                         1    Prob > chi2        =    0.0000
                                (Std. Err. adjusted for clustering on idcode)
```

union	Coef.	Robust 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: 2744 groups (14165 obs) dropped 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      =     12035
Group variable: idcode                          Number of groups   =      1690
                                                Obs per group: min =         2
                                                               avg =       7.1
                                                               max =        12
                                                LR chi2(6)         =     78.60
Log likelihood  =  -4510.888                    Prob > chi2        =    0.0000
```

union	Coef.	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.

❑

Saved results

`xtlogit, re` saves the following in `e()`:

Scalars

`e(N)`	number of observations
`e(N_g)`	number of groups
`e(N_cd)`	number of completely determined 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(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)`	significance
`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(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(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

xtlogit, fe saves 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-squared
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)	significance
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(vcetype)	title used to label Std. Err.
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(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

xtlogit, pa saves 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)	significance
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

Functions

e(sample)	marks estimation sample

Methods and formulas

`xtlogit` is implemented as an ado-file.

`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

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

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.

One can instead use the adaptive quadrature method of Liu and Pierce (1994), the `intmethod(aghermite)` option, which uses the mode and curvature of the mode as approximations for the mean and variance. We take the integrand

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

and find α_i the mode of $g(y_{it}, x_{it}, \nu_i)$. We calculate

$$\gamma_i = -\frac{\partial^2}{\partial \nu_i^2} \log\{g(y_{it}, x_{it}, \nu_i)\}\big|_{\nu_i = \alpha_i}$$

Then

$$\int_{-\infty}^{\infty} g(y_{it}, x_{it}, \nu_i) d\nu_i \approx \left(\frac{2}{\gamma_i}\right)^{1/2} \sum_{m=1}^{M} w_m^* \exp\{(a_m^*)^2\} g\left\{y_{it}, x_{it}, \left(\frac{2}{\gamma_i}\right)^{1/2} a_m^* + \alpha_i\right\}$$

This adaptation is performed on the first iteration only; that is, the α_i and γ_i are calculated once at the first iteration and then held constant throughout the subsequent iterations.

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

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

All three 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.

References

Allison, P. D. 2009. *Fixed Effects Regression Models*. Newbury Park, CA: Sage.

Conway, M. R. 1990. A random effects model for binary data. *Biometrics* 46: 317–328.

Liang, K.-Y., and S. L. Zeger. 1986. Longitudinal data analysis using generalized linear models. *Biometrika* 73: 13–22.

Liu, Q., and D. A. Pierce. 1994. A note on Gauss–Hermite quadrature. *Biometrika* 81: 624–629.

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, Series C* 31: 214–225.

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

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.

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.

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

Twisk, J. W. R. 2003. *Applied Longitudinal Data Analysis for Epidemiology: A Practical Guide*. Cambridge: Cambridge University Press.

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** — Fit population-averaged panel-data models by using GEE

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

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

[R] **clogit** — Conditional (fixed-effects) logistic regression

[R] **logit** — Logistic regression, reporting coefficients

[R] **logistic** — Logistic regression, reporting odds ratios

[U] **20 Estimation and postestimation commands**

Title

xtlogit postestimation — Postestimation tools for xtlogit

Description

The following postestimation commands are available after `xtlogit`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat`[1]	AIC, BIC, VCE, and estimation sample summary
`estimates`	cataloging estimation results
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`lrtest`	likelihood-ratio test
`margins`[2]	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 coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

[1] `estat ic` is not appropriate after `xtlogit, pa`.

[2] The default prediction statistic for `xtlogit, fe`, `pu1`, cannot be correctly handled by `margins`; however, `margins` can be used after `xtlogit, fe` with the `predict(pu0)` option or the `predict(xb)` option.

See the corresponding entries in the *Base Reference Manual* for details.

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
`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

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 $\mathbf{x}_j\beta$

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

The predicted probability for the fixed-effects model is conditional on there being only one outcome per group. See [R] **clogit** for details.

Menu

Statistics > Postestimation > Predictions, residuals, etc.

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.

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

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

`score` calculates the equation-level score, $u_j = \partial \ln L_j(\mathbf{x}_j\boldsymbol{\beta})/\partial(\mathbf{x}_j\boldsymbol{\beta})$.

`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 $\mathbf{x}_{it}\boldsymbol{\beta} + \text{offset}_{it}$.

Remarks

▷ Example 1

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, save the estimation results for later use, and use `test` to conduct a Wald test of the joint significance of those two variables' parameters:

```
. use http://www.stata-press.com/data/r12/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                                 LR chi2(2)  =    146.55
(Assumption: . nested in fullmodel)                   Prob > chi2 =    0.0000
```

These results confirm our finding that living in the south affects union status.

◁

Methods and formulas

All postestimation commands listed above are implemented as ado-files.

Also see

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

[U] **20 Estimation and postestimation commands**

Title

xtmelogit — Multilevel mixed-effects logistic regression

Syntax

```
xtmelogit depvar fe_equation || re_equation [ || re_equation ... ] [ , options ]
```

where the syntax of *fe_equation* is

```
[ indepvars ] [ if ] [ in ] [ , fe_options ]
```

and the syntax of *re_equation* is one of the following:

for random coefficients and intercepts

```
levelvar: [ varlist ] [ , re_options ]
```

for random effects among the values of a factor variable

```
levelvar: R.varname [ , re_options ]
```

levelvar is a variable identifying the group structure for the random effects at that level or `_all` representing one group comprising all observations.

fe_options	Description
Model	
`noconstant`	suppress the constant term from the fixed-effects equation
`offset(`*varname*`)`	include *varname* in model with coefficient constrained to 1

re_options	Description
Model	
`covariance(`*vartype*`)`	variance–covariance structure of the random effects
`noconstant`	suppress constant term from the random-effects equation
`collinear`	keep collinear variables

options	Description
Model	
`binomial(`*varname*`|#)`	set binomial trials if data are in binomial form
Integration	
`laplace`	use Laplacian approximation; equivalent to `intpoints(1)`
`intpoints(# [# ...])`	set the number of integration (quadrature) points; default is 7

Reporting

level(#)	set confidence level; default is level(95)
or	report fixed-effects coefficients as odds ratios
variance	show random-effects parameter estimates as variances and covariances
noretable	suppress random-effects table
nofetable	suppress fixed-effects table
estmetric	show parameter estimates in the estimation metric
noheader	suppress output header
nogroup	suppress table summarizing groups
nolrtest	do not perform LR test comparing with logistic regression
display_options	control column formats, row spacing, line width, and display of omitted variables and base and empty cells

Maximization

maximize_options	control the maximization process during gradient-based optimization; seldom used
retolerance(#)	tolerance for random-effects estimates; default is retolerance(1e-8); seldom used
reiterate(#)	maximum number of iterations for random-effects estimation; default is reiterate(50); seldom used
matsqrt	parameterize variance components using matrix square roots; the default
matlog	parameterize variance components using matrix logarithms
refineopts(*maximize_options*)	control the maximization process during refinement of starting values
coeflegend	display legend instead of statistics

vartype	Description
independent	one unique variance parameter per random effect, all covariances zero; the default unless a factor variable is specified
exchangeable	equal variances for random effects, and one common pairwise covariance
identity	equal variances for random effects, all covariances zero; the default if factor variables are specified
unstructured	all variances–covariances distinctly estimated

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

indepvars and *varlist* may contain time-series operators; see [U] **11.4.4 Time-series varlists**.

bootstrap, by, jackknife, mi estimate, rolling, 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.

Menu

Statistics > Longitudinal/panel data > Multilevel mixed-effects models > Mixed-effects logistic regression

Description

`xtmelogit` fits mixed-effects models for binary/binomial responses. Mixed models contain both *fixed effects* and *random effects*. The fixed effects are analogous to standard regression coefficients and are estimated directly. The random effects are not directly estimated (although they may be obtained postestimation) but are summarized according to their estimated variances and covariances. Random effects may take the form of either random intercepts or random coefficients, and the grouping structure of the data may consist of multiple levels of nested groups. The distribution of the random effects is assumed to be Gaussian. The conditional distribution of the response given the random effects is assumed to be Bernoulli, with success probability determined by the logistic cumulative distribution function (c.d.f.). Because the log likelihood for this model has no closed form, it is approximated by adaptive Gaussian quadrature.

Options

Model

`noconstant` suppresses the constant (intercept) term and may be specified for the fixed-effects equation and for any or all the random-effects equations.

`offset(`*varname*`)` specifies that *varname* be included in the fixed-effects portion of the model with the coefficient constrained to be 1.

`covariance(`*vartype*`)`, where *vartype* is

`independent` | `exchangeable` | `identity` | `unstructured`

specifies the structure of the covariance matrix for the random effects and may be specified for each random-effects equation. An `independent` covariance structure allows for a distinct variance for each random effect within a random-effects equation and assumes that all covariances are zero. `exchangeable` structure specifies one common variance for all random effects and one common pairwise covariance. `identity` is short for "multiple of the identity"; that is, all variances are equal and all covariances are zero. `unstructured` allows for all variances and covariances to be distinct. If an equation consists of p random-effects terms, the unstructured covariance matrix will have $p(p+1)/2$ unique parameters.

`covariance(independent)` is the default, except when the random-effects equation is a factor-variable specification `R.`*varname*, in which case `covariance(identity)` is the default, and only `covariance(identity)` and `covariance(exchangeable)` are allowed.

`collinear` specifies that `xtmelogit` not omit collinear variables from the random-effects equation. Usually there is no reason to leave collinear variables in place, and in fact doing so usually causes the estimation to fail because of the matrix singularity caused by the collinearity. However, with certain models (for example, a random-effects model with a full set of contrasts), the variables may be collinear, yet the model is fully identified because of restrictions on the random-effects covariance structure. In such cases, using the `collinear` option allows the estimation to take place with the random-effects equation intact.

`binomial(`*varname* | *#*`)` specifies that the data are in binomial form; that is, *depvar* records the number of successes from a series of binomial trials. This number of trials is given either as *varname*, which allows this number to vary over the observations, or as the constant *#*. If `binomial()` is not specified (the default), *depvar* is treated as Bernoulli, with any nonzero, nonmissing values indicating positive responses.

Integration

`laplace` specifies that log likelihoods be calculated using the Laplacian approximation, equivalent to adaptive Gaussian quadrature with one integration point for each level in the model; `laplace` is equivalent to `intpoints(1)`. Computation time increases as a function of the number of quadrature points raised to a power equaling the dimension of the random-effects specification. The computational time saved by using `laplace` can thus be substantial, especially when you have many levels and/or random coefficients.

The Laplacian approximation has been known to produce biased parameter estimates, but the bias tends to be more prominent in the estimates of the variance components rather than in estimates of the fixed effects. If your interest lies primarily with the fixed-effects estimates, the Laplace approximation may be a viable faster alternative to adaptive quadrature with multiple integration points.

Specifying a factor variable, `R.`*varname*, increases the dimension of the random effects by the number of distinct values of *varname*, that is, the number of factor levels. Even when this number is small to moderate, it increases the total random-effects dimension to the point where estimation with more than one quadrature point is prohibitively intensive.

For this reason, when you have factor variables in your random-effects equations, the `laplace` option is assumed. You can override this behavior by using the `intpoints()` option.

`intpoints(`*#*`[`*# ...*`])` sets the number of integration points for adaptive Gaussian quadrature. The more points, the more accurate the approximation to the log likelihood. However, computation time increases with the number of quadrature points, and in models with many levels and/or many random coefficients, this increase can be substantial.

You may specify one number of integration points applying to all levels of random effects in the model, or you may specify distinct numbers of points for each level. `intpoints(7)` is the default; that is, by default seven quadrature points are used for each level.

Reporting

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

`or` reports the fixed-effects coefficients transformed to odds ratios, that is, $\exp(b)$ 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.

`variance` displays the random-effects parameter estimates as variances and covariances. The default is to display them as standard deviations and correlations.

`noretable` suppresses the table of random effects.

`nofetable` suppresses the table of fixed effects.

`estmetric` displays all parameter estimates in the estimation metric. Fixed-effects estimates are unchanged from those normally displayed, but random-effects parameter estimates are displayed as log-standard deviations and hyperbolic arctangents of correlations, with equation names that organize them by model level.

noheader suppresses the output header, either at estimation or upon replay.

nogroup suppresses the display of group summary information (number of groups, average group size, minimum, and maximum) from the output header.

nolrtest prevents xtmelogit from performing a likelihood-ratio test that compares the mixed-effects logistic model with standard (marginal) logistic regression. This option may also be specified upon replay to suppress this test from the output.

display_options: noomitted, vsquish, noemptycells, baselevels, allbaselevels, 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**. Those that require special mention for xtmelogit are listed below.

For the technique() option, the default is technique(nr). The bhhh algorithm may not be specified.

from(*init_specs*) is particularly useful when combined with refineopts(iterate(0)), which bypasses the initial optimization stage; see below.

retolerance(#) specifies the convergence tolerance for the estimated random effects used by adaptive Gaussian quadrature. Although not estimated as model parameters, random-effects estimators are used to adapt the quadrature points. Estimating these random effects is an iterative procedure, with convergence declared when the maximum relative change in the random effects is less than retolerance(). The default is retolerance(1e-8). You should seldom have to use this option.

reiterate(#) specifies the maximum number of iterations used when estimating the random effects to be used in adapting the Gaussian quadrature points; see the retolerance() option. The default is reiterate(50). You should seldom have to use this option.

matsqrt (the default), during optimization, parameterizes variance components by using the matrix square roots of the variance–covariance matrices formed by these components at each model level.

matlog, during optimization, parameterizes variance components by using the matrix logarithms of the variance–covariance matrices formed by these components at each model level.

The matsqrt parameterization ensures that variance–covariance matrices are positive semidefinite, while matlog ensures matrices that are positive definite. For most problems, the matrix square root is more stable near the boundary of the parameter space. However, if convergence is problematic, one option may be to try the alternate matlog parameterization. When convergence is not an issue, both parameterizations yield equivalent results.

refineopts(*maximize_options*) controls the maximization process during the refinement of starting values. Estimation in xtmelogit takes place in two stages. In the first stage, starting values are refined by holding the quadrature points fixed between iterations. During the second stage, quadrature points are adapted with each evaluation of the log likelihood. Maximization options specified within refineopts() control the first stage of optimization; that is, they control the refining of starting values.

maximize_options specified outside refineopts() control the second stage.

The one exception to the above rule is the nolog option, which when specified outside refineopts() applies globally.

from(*init_specs*) is not allowed within refineopts() and instead must be specified globally.

Refining starting values helps make the iterations of the second stage (those that lead toward the solution) more numerically stable. In this regard, of particular interest is `refineopts(iterate(#))`, with two iterations being the default. Should the maximization fail because of instability in the Hessian calculations, one possible solution may be to increase the number of iterations here.

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

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

Remarks

Remarks are presented under the following headings:

Introduction

Mixed-effects logistic regression is logistic regression containing both fixed effects and random effects. In longitudinal/panel data, random effects are useful for modeling intracluster correlation; that is, observations in the same cluster are correlated because they share common cluster-level random effects.

`xtmelogit` allows for not just one, but many levels of nested clusters of random effects. For example, in a three-level model you can specify random effects for schools and then random effects for classes nested within schools. In this model, the observations (presumably, the students) comprise the first level, the schools comprise the second level, and the classes comprise the third.

However, for simplicity, for now we consider the two-level model where, for a series of M independent clusters, and conditional on a set of *random effects* $\mathbf{u}_j$,

$$\Pr(y_{ij} = 1|\mathbf{u}_j) = H\left(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j\right) \tag{1}$$

for $j = 1, \ldots, M$ clusters, with cluster j consisting of $i = 1, \ldots, n_j$ observations. The responses are the binary-valued y_{ij}, and we follow the standard Stata convention of treating $y_{ij} = 1$ if $\textit{depvar}_{ij} \neq 0$, and $y_{ij} = 0$ otherwise. The $1 \times p$ row vector $\mathbf{x}_{ij}$ are the covariates for the fixed effects, analogous to the covariates you would find in a standard logistic regression model, with regression coefficients (fixed effects) $\boldsymbol{\beta}$.

The $1 \times q$ vector $\mathbf{z}_{ij}$ are the covariates corresponding to the random effects and can be used to represent both random intercepts and random coefficients. For example, in a random-intercept model, $\mathbf{z}_{ij}$ is simply the scalar 1. The random effects $\mathbf{u}_j$ are M realizations from a multivariate normal distribution with mean $\mathbf{0}$ and $q \times q$ variance matrix $\boldsymbol{\Sigma}$. The random effects are not directly estimated as model parameters but are instead summarized according to the unique elements of $\boldsymbol{\Sigma}$, known as *variance components*. One special case of (1) places $\mathbf{z}_{ij} = \mathbf{x}_{ij}$, so that all covariate effects are essentially random and distributed as multivariate normal with mean $\boldsymbol{\beta}$ and variance $\boldsymbol{\Sigma}$.

Finally, because this is logistic regression, $H(\cdot)$ is the logistic cumulative distribution function (c.d.f.). The logistic c.d.f. maps the linear predictor to the probability of a success ($y_{ij} = 1$), with $H(v) = \exp(v)/\{1 + \exp(v)\}$.

Model (1) may also be stated in terms of a latent linear response, where only $y_{ij} = I(y^*_{ij} > 0)$ is observed for the latent

$$y^*_{ij} = \mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j + \epsilon_{ij}$$

The errors ϵ_{ij} are distributed as logistic with mean zero and variance $\pi^2/3$ and are independent of $\mathbf{u}_j$.

Model (1) is an example of a generalized linear mixed model (GLMM), which generalizes the linear mixed-effects (LME) model to non-Gaussian responses. You can fit LMEs in Stata by using `xtmixed`. Because of the relationship between LMEs and GLMMs, there is insight to be gained through examination of the linear mixed model. This is especially true for Stata users because the terminology, syntax, options, and output for fitting these types of models are nearly identical. See [XT] **xtmixed** and the references therein, particularly in the *Introduction*, for more information.

Multilevel models with binary responses have been used extensively in the health and social sciences. As just one example, Leyland and Goldstein (2001, sec. 3.6) describe a study of equity of health care in Great Britain. Multilevel models with binary and other limited dependent responses also have a long history in econometrics; Rabe-Hesketh, Skrondal, and Pickles (2005) provide an excellent survey.

Log-likelihood calculations for fitting any mixed-effects model (LME, logistic, or otherwise) require integrating out the random effects. For LME, this integral has a closed-form solution, but this is not so with the logistic or any other GLMM. In dealing with this difficulty, early estimation methods avoided the integration altogether. Two such popular methods are the closely related penalized quasilikelihood (PQL) and marginal quasilikelihood (MQL) (Breslow and Clayton 1993). Both PQL and MQL use a combination of iterative reweighted least squares (see [R] **glm**) and standard estimation techniques for fitting LMEs. Efficient computational methods for fitting LMEs have existed for some time (Bates and Pinheiro 1998; Littell et al. 2006), and PQL and MQL inherit this computational efficiency. However, both these methods suffer from two key disadvantages. First, they have been shown to be biased, and this bias can be severe when clusters are small and/or intracluster correlation is high (Rodríguez and Goldman 1995; Lin and Breslow 1996). Second, because they are "quasilikelihood" methods and not true likelihood methods, their use prohibits comparing nested models via likelihood-ratio tests, blocking the main avenue of inference involving variance components.

The advent of modern computers has brought with it the development of more computationally intensive methods, such as bias-corrected PQL (Lin and Breslow 1996), Bayesian Markov-Chain Monte Carlo, and simulated maximum likelihood, just to name a few; see Ng et al. (2006) for a discussion of these alternate strategies (and more) for mixed-effects models for binary outcomes.

One widely used modern method is to directly estimate the integral required to calculate the log likelihood by Gauss–Hermite quadrature, or some variation thereof. Because the log likelihood itself is estimated, this method has the advantage of permitting LR tests for comparing nested models. Also, if done correctly, quadrature approximations can be quite accurate, thus minimizing bias.

In discussing quadrature, it is easiest to relate to the simplest form of (1)—the simplest model you can fit using `xtmelogit`—the two-level model with a random intercept,

$$\Pr(y_{ij} = 1) = H\left(\mathbf{x}_{ij}\boldsymbol{\beta} + u_j\right)$$

This model can also be fit using `xtlogit` with the `re` option. `xtlogit` supports three types of Gauss–Hermite quadrature; see [XT] **xtlogit**. The estimation method used by `xtmelogit` is a multicoefficient and multilevel extension of one of these quadrature types, namely, adaptive Gaussian quadrature (AGQ) based on conditional modes, with the multicoefficient extension from Pinheiro and Bates (1995) and the multilevel extension from Pinheiro and Chao (2006); see *Methods and formulas*.

Finally, using (1) and its multilevel extensions requires we state our convention of terminology. Model (1) is what we call a *two-level* model, with extensions to three, four, or any number of levels. In (1), the observation y_{ij} is for individual i within cluster j and the individuals comprise the first level and the clusters the second level of the model. In our hypothetical three-level model with classes nested within schools, the observations within schools (the students, presumably) would constitute the first level, the classes would constitute the second level, and the schools would constitute the third level. This differs from certain citations in the classical ANOVA literature and texts such as Pinheiro and Bates (2000) but is the standard in the vast literature on hierarchical models, for example, Skrondal and Rabe-Hesketh (2004).

Two-level models

We begin with a simple application of (1). We begin with a two-level model because a one-level model, in our terminology, is just standard logistic regression; see [R] **logistic**.

▷ Example 1

Ng et al. (2006) analyze a subsample of data from the 1989 Bangladesh fertility survey (Huq and Cleland 1990), which polled 1,934 Bangladeshi women on their use of contraception.

```
. use http://www.stata-press.com/data/r12/bangladesh
(Bangladesh Fertility Survey, 1989)
. describe
Contains data from http://www.stata-press.com/data/r12/bangladesh.dta
  obs:         1,934                          Bangladesh Fertility Survey, 1989
 vars:             7                          28 May 2011 20:27
 size:        19,340                          (_dta has notes)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
district        byte   %9.0g                  District
c_use           byte   %9.0g       yesno      Use contraception
urban           byte   %9.0g       urban      Urban or rural
age             float  %9.0g                  Age, mean centered
child1          byte   %9.0g                  1 child
child2          byte   %9.0g                  2 children
child3          byte   %9.0g                  3 or more children
-------------------------------------------------------------------------------
Sorted by:  district
```

The women sampled were from 60 districts, identified by variable `district`. Each district contained either urban or rural areas (variable `urban`) or both. Variable `c_use` is the binary response, with a value of one indicating contraceptive use. Other covariates include mean-centered `age` and three indicator variables recording number of children.

Consider a standard logistic regression model, amended to have random effects for each district. Defining $\pi_{ij} = \Pr(\texttt{c\_use}_{ij} = 1)$, we have

$$\text{logit}(\pi_{ij}) = \beta_0 + \beta_1 \texttt{urban}_{ij} + \beta_2 \texttt{age}_{ij} + \beta_3 \texttt{child1}_{ij} + \beta_4 \texttt{child2}_{ij} + \beta_5 \texttt{child3}_{ij} + u_j \quad (2)$$

for $j = 1, \ldots, 60$ districts, with $i = 1, \ldots, n_j$ women in district j.

```
. xtmelogit c_use urban age child* || district:

Refining starting values:

Iteration 0:   log likelihood = -1219.2682
Iteration 1:   log likelihood = -1209.3544
Iteration 2:   log likelihood = -1207.1919

Performing gradient-based optimization:

Iteration 0:   log likelihood = -1207.1919
Iteration 1:   log likelihood = -1206.8323
Iteration 2:   log likelihood = -1206.8322
Iteration 3:   log likelihood = -1206.8322

Mixed-effects logistic regression               Number of obs      =      1934
Group variable: district                        Number of groups   =        60

                                                Obs per group: min =         2
                                                               avg =      32.2
                                                               max =       118

Integration points =   7                        Wald chi2(5)       =    109.60
Log likelihood = -1206.8322                     Prob > chi2        =    0.0000
```

c_use	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
urban	.7322764	.1194857	6.13	0.000	.4980887	.9664641
age	-.0264982	.0078916	-3.36	0.001	-.0419654	-.0110309
child1	1.116002	.1580921	7.06	0.000	.8061466	1.425856
child2	1.365895	.1746691	7.82	0.000	1.02355	1.70824
child3	1.344031	.1796549	7.48	0.000	.991914	1.696148
_cons	-1.68929	.1477592	-11.43	0.000	-1.978892	-1.399687

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
district: Identity				
sd(_cons)	.4643477	.0789531	.3327464	.6479975

```
LR test vs. logistic regression: chibar2(01) =    43.39 Prob>=chibar2 = 0.0000
```

Those of you familiar with `xtmixed`, Stata's command for fitting linear mixed models, will recognize the syntax and output. Whether you are familiar with `xtmixed`, however, there are enough nuances in `xtmelogit` to warrant the guided tour:

1. By typing "`c_use urban age child*`", we specified the binary response, `c_use`, and the fixed portion of the model in the same way we would if we were using `logit` or any other estimation command. Our fixed effects are a constant term (intercept) and coefficients on `urban`, `age`, and the indicator variables `child1`, `child2`, and `child3`.
2. When we added "`|| district:`", we specified random effects at the level identified by group variable `district`. Because we wanted only a random intercept, that is all we had to type.
3. The estimation log consists of two parts:
   (a) A set of iterations aimed at refining starting values. These are designed to be relatively quick iterations aimed at getting the parameter estimates within a neighborhood of the eventual solution, making the iterations in (b) more numerically stable.
   (b) A set of "gradient-based" iterations. By default, these are Newton–Raphson iterations, but other methods are available by specifying the appropriate *maximize_options*; see [R] **maximize**.
4. Within the output header you will find a series of group (`district`) statistics. District sizes vary greatly, ranging the all way from $n_j = 2$ to $n_j = 118$.

5. Just above the reported log likelihood, the number of "`Integration Points`" is displayed as 7, the default. As stated previously in *Introduction*, log likelihoods are approximated using adaptive Gaussian quadrature, and the more integration points you use, the better the approximation; see *Methods and formulas*. You can specify an alternate number of integration points using the `intpoints()` option.

   In any case, refitting this model with more integration points would demonstrate that seven integration points is sufficient.

6. The first estimation table reports the fixed effects, and these can be interpreted just as you would the output from `logit`. You can also specify the `or` option at estimation or on replay to display the fixed effects as odds ratios instead.

   If you did display results as odds ratios, you would find urban women to have roughly double the odds of using contraception as that of their rural counterparts. Having any number of children will increase the odds from three- to fourfold, when compared with the base category of no children. Contraceptive use also decreases with age.

7. The second estimation table shows the estimated variance components. The first section of the table is labeled "`district: Identity`", meaning that these are random effects at the `district` level and that their variance–covariance matrix is a multiple of the identity matrix; that is, $\boldsymbol{\Sigma} = \sigma_u^2\mathbf{I}$. Because we have only one random effect at this level, `xtmelogit` knew that `Identity` is the only possible covariance structure. In any case, σ_u was estimated as 0.464 with standard error 0.079.

   If you prefer variance estimates, $\widehat{\sigma}_u^2$, to standard deviation estimates, $\widehat{\sigma}_u$, specify the `variance` option either at estimation or on replay.

8. A likelihood-ratio test comparing the model to ordinary logistic regression, (2) without u_j, is provided and is highly significant for these data.

9. Finally, because (2) is a simple random-intercept model, you can also fit it with `xtlogit`, specifying the `re` option.

   We now store our estimates for later use.

```
. estimates store r_int
```

◁

In what follows we will be extending (2), focusing on variable `urban`. Before we begin, to keep things short we restate (2) as

$$\text{logit}(\pi_{ij}) = \beta_0 + \beta_1\texttt{urban}_{ij} + \mathcal{F}_{ij} + u_j$$

where $\mathcal{F}_{ij}$ is merely shorthand for the portion of the fixed-effects specification having to do with age and children.

▷ Example 2

Extending (2) to allow for a random slope on the indicator variable `urban` yields the model

$$\text{logit}(\pi_{ij}) = \beta_0 + \beta_1\texttt{urban}_{ij} + \mathcal{F}_{ij} + u_j + v_j\texttt{urban}_{ij} \tag{3}$$

which we can fit by typing

```
. xtmelogit c_use urban age child* || district: urban
(output omitted)
. estimates store r_urban
```

Extending the model was as simple as adding `urban` to the random effects specification, so that the model now includes a random intercept *and* a random coefficient on `urban`. We dispense with the output because, although this is an improvement over the random-intercept model (2),

```
. lrtest r_int r_urban
Likelihood-ratio test                                 LR chi2(1)  =      3.66
(Assumption: r_int nested in r_urban)                 Prob > chi2 =    0.0558
Note: The reported degrees of freedom assumes the null hypothesis is not on
      the boundary of the parameter space.  If this is not true, then the
      reported test is conservative.
```

we find the default covariance structure for (u_j, v_j), `covariance(Independent)`,

$$\boldsymbol{\Sigma} = \text{Var}\begin{bmatrix} u_j \\ v_j \end{bmatrix} = \begin{bmatrix} \sigma_u^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix}$$

to be inadequate. We state that the random-coefficient model is an "improvement" over the random-intercept model because the null hypothesis of the LR comparison test ($H_0: \sigma_v^2 = 0$) is on the boundary of the parameter test. This makes the reported p-value, 5.6%, an upper bound on the actual p-value, which is actually half that; see [XT] **xtmixed** for more details on boundary tests.

We see below that we can reject this model in favor of one that allows correlation between u_j and v_j.

```
. xtmelogit c_use urban age child* || district: urban, covariance(unstructured)
> variance
```

(*output omitted*)

```
Mixed-effects logistic regression               Number of obs      =      1934
Group variable: district                        Number of groups   =        60

                                                Obs per group: min =         2
                                                               avg =      32.2
                                                               max =       118

Integration points =   7                        Wald chi2(5)       =     97.50
Log likelihood =  -1199.315                     Prob > chi2        =    0.0000
```

c_use	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
urban	.8157872	.171552	4.76	0.000	.4795516	1.152023
age	-.026415	.008023	-3.29	0.001	-.0421398	-.0106902
child1	1.13252	.1603285	7.06	0.000	.8182819	1.446758
child2	1.357739	.1770522	7.67	0.000	1.010723	1.704755
child3	1.353827	.1828801	7.40	0.000	.9953881	1.712265
_cons	-1.71165	.1605617	-10.66	0.000	-2.026345	-1.396954

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
district: Unstructured				
var(urban)	.6663222	.3224715	.258071	1.7204
var(_cons)	.3897434	.1292459	.2034723	.7465388
cov(urban,_cons)	-.4058846	.1755418	-.7499403	-.0618289

```
LR test vs. logistic regression:     chi2(3) =    58.42   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

```
. estimates store r_urban_corr
. lrtest r_urban r_urban_corr
Likelihood-ratio test                                 LR chi2(1)  =     11.38
(Assumption: r_urban nested in r_urban_corr)          Prob > chi2 =    0.0007
```

By specifying `covariance(unstructured)` above, we told `xtmelogit` to allow correlation between random effects at the "`district` level"; that is,

$$\boldsymbol{\Sigma} = \text{Var}\begin{bmatrix} u_j \\ v_j \end{bmatrix} = \begin{bmatrix} \sigma_u^2 & \sigma_{uv} \\ \sigma_{uv} & \sigma_v^2 \end{bmatrix}$$

The `variance` option is a display option that does not affect estimation but merely displays the variance components as variances and covariances instead of standard deviations and correlations. This feature will prove convenient in the discussion that follows.

◁

▷ Example 3

The purpose of introducing a random coefficient on the binary variable `urban` in (3) was to allow for separate random effects, within each district, for the urban and rural areas of that district. Hence, if we had the binary variable `rural` in our data such that $\texttt{rural}_{ij} = 1 - \texttt{urban}_{ij}$, then we can reformulate (3) as

$$\text{logit}(\pi_{ij}) = \beta_0\texttt{rural}_{ij} + (\beta_0 + \beta_1)\texttt{urban}_{ij} + \mathcal{F}_{ij} + u_j\texttt{rural}_{ij} + (u_j + v_j)\texttt{urban}_{ij} \qquad (3a)$$

where we have translated both the fixed portion and random portion to be in terms of `rural` rather than a random intercept. Translating the fixed portion is not necessary to make the point we make below, but we do so anyway for uniformity.

Translating the estimated random-effects parameters from the previous output to ones appropriate for (3a), we get $\text{Var}(u_j) = \widehat{\sigma}_u^2 = 0.390$,

$$\begin{aligned} \text{Var}(u_j + v_j) &= \widehat{\sigma}_u^2 + \widehat{\sigma}_v^2 + 2\widehat{\sigma}_{uv} \\ &= 0.390 + 0.666 - 2(0.406) = 0.244 \end{aligned}$$

and $\text{Cov}(u_j, u_j + v_j) = \widehat{\sigma}_u^2 + \widehat{\sigma}_{uv} = 0.390 - 0.406 = -0.016$.

An alternative that does not require remembering how to calculate variances and covariances involving sums—and one that also gives you standard errors—is to let Stata do the work for you:

```
. generate byte rural = 1 - urban
. xtmelogit c_use rural urban age child*, nocons || district: rural urban,
> nocons cov(unstr) var
  (output omitted)
Mixed-effects logistic regression               Number of obs      =      1934
Group variable: district                        Number of groups   =        60
                                                Obs per group: min =         2
                                                               avg =      32.2
                                                               max =       118
Integration points =   7                        Wald chi2(6)       =    120.24
Log likelihood =  -1199.315                     Prob > chi2        =    0.0000
```

c_use	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
rural	-1.71165	.1605618	-10.66	0.000	-2.026345	-1.396954
urban	-.8958623	.1704961	-5.25	0.000	-1.230028	-.5616962
age	-.026415	.008023	-3.29	0.001	-.0421398	-.0106902
child1	1.13252	.1603285	7.06	0.000	.818282	1.446758
child2	1.357739	.1770522	7.67	0.000	1.010724	1.704755
child3	1.353827	.1828801	7.40	0.000	.9953882	1.712265

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
district: Unstructured				
var(rural)	.3897439	.1292459	.2034726	.7465394
var(urban)	.2442965	.1450673	.0762886	.7823029
cov(rural,urban)	-.0161411	.1057469	-.2234011	.191119

```
LR test vs. logistic regression:     chi2(3) =    58.42   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

The above output demonstrates an equivalent fit to that we displayed for model (3) in example 2, with the added benefit of a more direct comparison of the parameters for rural and urban areas. ◁

❑ Technical note

We used the binary variables, `rural` and `urban`, instead of the factor notation `i.urban` because, although supported in the fixed-effects specification of the model, such notation is not supported in random-effected specifications. ❑

❑ Technical note

Our model fits for (3) and (3a) are equivalent only because we allowed for correlation in the random effects for both. Had we used the default "`Independent`" covariance structure, we would be fitting different models; in (3) we would be making the restriction that $\text{Cov}(u_j, v_j) = 0$, whereas in (3a) we would be assuming that $\text{Cov}(u_j, u_j + v_j) = 0$.

The moral here is that, although `xtmelogit` will do this by default, one should be cautious when imposing an independent covariance structure, because the correlation between random effects is not invariant to model translations that would otherwise yield equivalent results in standard regression models. In our example, we remapped an intercept and binary coefficient to two complementary binary coefficients, something we could do in standard logistic regression without consequence, but that here required more consideration.

Rabe-Hesketh and Skrondal (2008, 150–153) provide a nice discussion of this phenomenon in the related case of recentering a continuous covariate. ❑

Other covariance structures

In the above examples, we demonstrated the `Independent` and `Unstructured` covariance structures. Also available are `Identity` (seen previously in output but not directly specified), which restricts random effects to be uncorrelated and share a common variance, and `Exchangeable`, which assumes a common variance and a common pairwise covariance.

You can also specify multiple random-effects equations at the same level, in which case the above four covariance types can be combined to form more complex blocked-diagonal covariance structures. This could be used, for example, to impose an equality constraint on a subset of variance components or to otherwise group together a set of related random effects.

Continuing the previous example, typing

```
. xtmelogit c_use urban age child* || district: child*, cov(exchangeable) || district:
```

would fit a model with the same fixed effects as (3) but with random-effects structure

$$\text{logit}(\pi_{ij}) = \beta_0 + \cdots + u_{1j}\texttt{child1}_{ij} + u_{2j}\texttt{child2}_{ij} + u_{3j}\texttt{child3}_{ij} + v_j$$

That is, we have random coefficients on each indicator variable for children (the first `district:` specification) and an overall district random intercept (the second `district:` specification). The above syntax fits a model with overall covariance structure

$$\mathbf{\Sigma} = \text{Var}\begin{bmatrix} u_{1j} \\ u_{2j} \\ u_{3j} \\ v_j \end{bmatrix} = \begin{bmatrix} \sigma_u^2 & \sigma_c & \sigma_c & 0 \\ \sigma_c & \sigma_u^2 & \sigma_c & 0 \\ \sigma_c & \sigma_c & \sigma_u^2 & 0 \\ 0 & 0 & 0 & \sigma_v^2 \end{bmatrix}$$

reflecting the relationship among the random coefficients for children. We did not have to specify `noconstant` on the first `district:` specification. `xtmelogit` automatically avoids collinearity by including an intercept on only the final specification among repeated-level equations.

Of course, if we fit the above model we would heed our own advice from the previous technical note and make sure that not only our data but also our specification characterization of the random effects permitted the above structure. That is, we would check the above against a model that had an `Unstructured` covariance for all four random effects and then perhaps against a model that assumed an `Unstructured` covariance among the three random coefficients on children, coupled with independence with the random intercept. All comparisons can be made by storing estimates (command `estimates store`) and then using `lrtest`, as demonstrated previously.

Distribution theory for likelihood-ratio tests

A keen observer of the output for fitting the equivalent models (3) and (3a) may have noticed that, in the output for (3a), the covariance parameter does not appear at all significant. In fact, an LR test would confirm this. In the results for (3), however, all three variance components appear to be significant, and you would be hard pressed to prove otherwise. We thus have two entirely equivalent model fits, yet the first fit relies on all three variance components, whereas with the second you could presumably drop the covariance between the random coefficients. Whether generalizing from model (2) to model (3)/(3a) requires one or two additional parameters is unclear. Asked another way: do the models differ by 1 or 2 degrees of freedom?

Such paradoxical cases are at the core of the central issue concerning distribution theory for LR tests, where oftentimes significance levels cannot be exactly computed when models differ by (or appear to differ by) more than one variance component. We will not go into the details here but instead direct you to the section in [XT] **xtmixed** with the same name as this one. What is stated there applies equally to `xtmelogit`.

When significance levels cannot be computed exactly, both `xtmelogit` and `lrtest` will caution you, and you may have noticed the following message at the bottom of some of the `xtmelogit` output we have produced:

```
Note: LR test is conservative and provided only for reference.
```

In Stata, part of that message is blue, meaning that you can click on it for more details. If you are not interested in all the details, it suffices to know that by "conservative" we mean that the p-value displayed is an upper bound on the actual p-value. If you choose to reject the null hypothesis of a reduced model on the basis of the displayed p-value, you would also reject based on the actual p-value, because it would be even smaller.

The output of `lrtest` will produce a similar note in these situations. However, because `lrtest` can be used to compare all kinds of nested models, determining whether boundary conditions exist is up to the user.

Three-level models

The methods we have discussed so far extend from two-level models to three or more level models with nested random effects. By *nested* we mean that the random effects shared within lower-level subgroups are unique to the upper-level groups. For example, assuming that classroom effects would be nested within schools would be natural, because classrooms are unique to schools.

▷ Example 4

Rabe-Hesketh, Toulopoulou, and Murray (2001) analyzed data from a study measuring the cognitive ability of patients with schizophrenia, compared with their relatives and control subjects. Cognitive ability was measured as the successful completion of the "Tower of London", a computerized task, measured at three levels of difficulty. For all but one of the 226 subjects, there were three measurements (one for each difficulty level), and because patients' relatives were also tested, a family identifier, `family`, was also recorded.

```
. use http://www.stata-press.com/data/r12/towerlondon, clear
(Tower of London data)
. describe
Contains data from http://www.stata-press.com/data/r12/towerlondon.dta
  obs:           677                          Tower of London data
 vars:             5                          31 May 2011 10:41
 size:         4,739                          (_dta has notes)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
family          int    %8.0g                  Family ID
subject         int    %9.0g                  Subject ID
dtlm            byte   %9.0g                  1 = task completed
difficulty      byte   %9.0g                  Level of difficulty: -1, 0, or 1
group           byte   %8.0g                  1: controls; 2: relatives; 3:
                                                schizophrenics
-------------------------------------------------------------------------------
Sorted by:  family  subject
```

We fit a logistic model with response `dtlm`, the indicator of cognitive function, and with covariates `difficulty` and a set of indicator variables for `group`, with the controls (`group==1`) being the base category. We also allow for random effects due to families and due to subjects within families.

```
. xtmelogit dtlm difficulty i.group || family: || subject:
  (output omitted)
Mixed-effects logistic regression               Number of obs      =       677
```

Group Variable	No. of Groups	Observations per Group Minimum	Average	Maximum	Integration Points
family	118	2	5.7	27	7
subject	226	2	3.0	3	7

```
                                                Wald chi2(3)       =     74.89
Log likelihood = -305.12043                     Prob > chi2        =    0.0000
```

dtlm	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
difficulty	-1.648506	.1932139	-8.53	0.000	-2.027198	-1.269814
group						
2	-.24868	.3544065	-0.70	0.483	-.943304	.4459441
3	-1.0523	.3999896	-2.63	0.009	-1.836265	-.2683348
_cons	-1.485861	.2848469	-5.22	0.000	-2.04415	-.9275709

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
family: Identity				
sd(_cons)	.7544415	.3457249	.3072983	1.852213
subject: Identity				
sd(_cons)	1.066739	.3214235	.5909883	1.925472

```
LR test vs. logistic regression:     chi2(2) =    17.54   Prob > chi2 = 0.0002
Note: LR test is conservative and provided only for reference.
```

But we would prefer to see odds ratios and variances for the random-effects parameters:

```
. xtmelogit, or variance
Mixed-effects logistic regression               Number of obs      =       677
```

Group Variable	No. of Groups	Observations per Group Minimum	Average	Maximum	Integration Points
family	118	2	5.7	27	7
subject	226	2	3.0	3	7

```
                                                Wald chi2(3)       =     74.89
Log likelihood = -305.12043                     Prob > chi2        =    0.0000
```

dtlm	Odds Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
difficulty	.192337	.0371622	-8.53	0.000	.131704	.2808839
group						
2	.7798295	.2763767	-0.70	0.483	.3893393	1.561964
3	.3491338	.1396499	-2.63	0.009	.1594117	.7646518
_cons	.2263075	.064463	-5.22	0.000	.1294902	.3955133

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
family: Identity				
var(_cons)	.569182	.5216584	.0944322	3.430694
subject: Identity				
var(_cons)	1.137931	.6857498	.3492672	3.707442

LR test vs. logistic regression: chi2(2) = 17.54 Prob > chi2 = 0.0002

Note: LR test is conservative and provided only for reference.

Notes:

1. This is a three-level model with two random-effects equations, separated by `||`. The first is a random intercept (constant only) at the `family` level, and the second is a random intercept at the `subject` level. The order in which these are specified (from left to right) is important—`xtmelogit` assumes that `subject` is nested within `family`.
2. The information on groups is now displayed as a table, with one row for each upper level. Among other things, we see that we have 226 subjects from 118 families. Also the number of integration points for adaptive Gaussian quadrature is displayed within this table, because you can choose to have it vary by model level. As with two-level models, the default is seven points.

   You can suppress this table with the `nogroup` or the `noheader` option, which will suppress the rest of the header as well.
3. The variance-component estimates are now organized and labeled according to level.

After adjusting for the random-effects structure, the odds of successful completion of the Tower of London decrease dramatically as the level of difficulty increases. Also, schizophrenics (`group==3`) tended not to perform as well as the control subjects. Of course we would make similar conclusions from a standard logistic model fit to the same data, but the odds ratios would differ somewhat.

◁

❑ Technical note

In the previous example, the subjects are coded with unique values between 1 and 251 (with some gaps), but such coding is not necessary to produce nesting within families. Once we specified the nesting structure to `xtmelogit`, all that was important was the relative coding of `subject` within each unique value of `family`. We could have coded `subjects` as the numbers 1, 2, 3, and so on, restarting at 1 with each new family, and `xtmelogit` would have produced the same results.

Group identifiers may also be coded using string variables.

❑

The above extends to models with more than two levels of nesting in the obvious manner, by adding more random-effects equations, each separated by `||`. The order of nesting goes from left to right as the groups go from biggest (highest level) to smallest (lowest level).

Computation time and the Laplacian approximation

Like many programs that fit generalized linear mixed models, `xtmelogit` can be computationally intensive. This is particularly true for large datasets with many lowest-level clusters, models with many random coefficients, models with many estimable parameters (both fixed effects and variance components), or any combination thereof.

Computation time will also depend on hardware and other external factors but in general is (roughly) a function of $p^2\{M + M(N_Q)^{q_t}\}$, where p is the number of estimable parameters, M is the number of lowest-level (smallest) clusters, N_Q is the number of quadrature points, and q_t is the total dimension of the random effects, that is, the total number of random intercepts and coefficients at all levels.

For a given model and a given dataset, the only prevailing factor influencing computation time is $(N_Q)^{q_t}$. However, because this is a power function, this factor can get prohibitively large. Consider a model with one random intercept and three random coefficients, such as that discussed in *Other covariance structures*. For such a model, $(N_Q)^{q_t} = 7^4 = 2{,}401$ using the default number of quadrature points. Even a modest reduction to five quadrature points would reduce this factor by almost fourfold ($5^4 = 625$) which, depending on M and p, could drastically speed up estimation.

Ideally, you want to use enough quadrature points such that your estimates are stable and that adding more quadrature points would not change the estimates much. If you want accurate estimates, we recommend that you perform this check. We have tacitly followed this advice in all the models we have fit thus far. In each example, increasing the number of quadrature points from the default of seven did not make much of a difference.

However, we do not deny a tradeoff between speed and accuracy, and in that spirit we give you the option to choose a (possibly) less accurate solution in the interest of getting quicker results. Toward this end is the limiting case of $N_Q = 1$, otherwise known as the Laplacian approximation; see *Methods and formulas*. You can obtain this estimate either by using the `laplace` option or by directly setting `intpoints(1)`. The computational benefit is evident—one raised to any power equals one—and the Laplacian approximation has been shown to perform well in certain situations (Liu and Pierce 1994; Tierney and Kadane 1986).

In the previous section, we fit a three-level model to the Tower of London data using seven quadrature points. We refit the same model, this time via the Laplacian approximation:

```
. xtmelogit dtlm difficulty i.group || family: || subject:, laplace or variance
  (output omitted)
Mixed-effects logistic regression               Number of obs      =       677
```

Group Variable	No. of Groups	Observations per Group Minimum	Average	Maximum	Integration Points
family	118	2	5.7	27	1
subject	226	2	3.0	3	1

```
                                                Wald chi2(3)       =     76.09
Log likelihood = -306.51035                     Prob > chi2        =    0.0000
```

dtlm	Odds Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
difficulty	.2044132	.0377578	-8.60	0.000	.1423248	.2935872
group						
2	.7860452	.2625197	-0.72	0.471	.4084766	1.512613
3	.3575718	.1354592	-2.71	0.007	.1701774	.7513194
_cons	.2396663	.0639645	-5.35	0.000	.1420464	.4043746

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
family: Identity				
var(_cons)	.5229424	.4704255	.0896881	3.049108
subject: Identity				
var(_cons)	.790933	.5699271	.1926569	3.247094

LR test vs. logistic regression: chi2(2) = 14.76 Prob > chi2 = 0.0006

Note: LR test is conservative and provided only for reference.
Note: log-likelihood calculations are based on the Laplacian approximation.

Comparing these results to those previously obtained, we observe the following:

1. Odds ratios and their standard errors are well approximated by the Laplacian method. Therefore, if your interest lies primarily here, then `laplace` may be a viable alternative.
2. Estimates of variance components exhibit bias, particularly at the lower (`subject`) level.
3. The model log-likelihood and comparison LR test are in fair agreement.

Although this is by no means the rule, we find the above observations to be fairly typical based on our own experience. Pinheiro and Chao (2006) also make observations similar to points 1 and 2 on the basis of their simulation studies: bias due to Laplace (when present) tends to exhibit itself more in the estimated variance components than in the estimates of the fixed effects.

Item 3 is of particular interest, because it demonstrates that `laplace` can produce a decent estimate of the model log likelihood. Consequently, you can use `laplace` during the model building phase of your analysis, during which you are comparing competing models by using LR tests. Once you settle on a parsimonious model that fits well, you can then increase the number of quadrature points and obtain more accurate parameter estimates for further study.

We discuss such a scenario in *Other covariance structures*, where we posit a blocked-diagonal exchangeable/identity covariance structure and recommend comparing against more complex structures to verify our assumptions. The comparisons ruling out the more complex structures can be performed more quickly using `laplace`.

Of course, sometimes the Laplacian approximation will perform either better or worse than observed here. This behavior depends primarily on cluster size and intracluster correlation, but the relative influence of these factors is unclear. The idea behind the Laplacian approximation is to approximate the posterior density of the random effects given the response with a normal distribution; see *Methods and formulas*. Asymptotic theory dictates that this approximation improves with larger clusters. Of course, the key question, as always, is "How large is large enough?" Also, there are data situations where the Laplacian approximation performs well even with small clusters. Therefore, it is difficult to make a definitive call as to when you can expect `laplace` to yield accurate results across all aspects of the model.

In conclusion, consider our above advice as a rule of thumb based on empirical evidence.

Crossed-effects models

Not all mixed-effects models contain nested random effects.

▷ Example 5

Rabe-Hesketh and Skrondal (2008, 481ff) perform an analysis on school data from Fife, Scotland. The data, originally from Paterson (1991), are from a study measuring students' attainment as an integer score from 1 to 10, based on the Scottish school exit examination taken at age 16. The study comprises 3,435 students who first attended any one of 148 primary schools and then any one of 19 secondary schools.

```
. use http://www.stata-press.com/data/r12/fifeschool
(School data from Fife, Scotland)
. describe
Contains data from http://www.stata-press.com/data/r12/fifeschool.dta
  obs:         3,435                          School data from Fife, Scotland
 vars:             5                          28 May 2011 10:08
 size:        24,045                          (_dta has notes)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
pid              int    %9.0g                 Primary school ID
sid              byte   %9.0g                 Secondary school ID
attain           byte   %9.0g                 Attainment score at age 16
vrq              int    %9.0g                 Verbal-reasoning score from final
                                                year of primary school
sex              byte   %9.0g                 1: female; 0: male
-------------------------------------------------------------------------------
Sorted by:
. generate byte attain_gt_6 = attain > 6
```

To make the analysis relevant to our present discussion, we focus not on the attainment score itself but instead on whether the score is greater than 6. We wish to model this indicator as a function of the fixed effect `sex` and of random effects due to primary and secondary schools.

For this analysis, it would make sense to assume that the random effects are not nested, but instead *crossed*, meaning that the effect due to primary school is the same regardless of the secondary school attended. Our model is thus

$$\text{logit}\{\Pr(\texttt{attain}_{ijk} > 6)\} = \beta_0 + \beta_1 \texttt{sex}_{ijk} + u_j + v_k \tag{4}$$

for student i, $i = 1, \ldots, n_{jk}$, who attended primary school j, $j = 1, \ldots, 148$, and then secondary school k, $k = 1, \ldots, 19$.

Because there is no evident nesting, one solution would be to consider the data as a whole and fit a two-level, one-cluster model with random-effects structure

$$\mathbf{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_{148} \\ v_1 \\ \vdots \\ v_{19} \end{bmatrix} \sim N(\mathbf{0}, \boldsymbol{\Sigma}); \quad \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_u^2 \mathbf{I}_{148} & \mathbf{0} \\ \mathbf{0} & \sigma_v^2 \mathbf{I}_{19} \end{bmatrix}$$

We can fit such a model by using the group designation `_all:`, which tells `xtmelogit` to treat the whole dataset as one cluster, and the factor notation `R.`*varname*, which mimics the creation of indicator variables identifying schools:

```
. xtmelogit attain_gt_6 sex || _all:R.pid || _all:R.sid, or variance
```

But we do not recommend fitting this model this way, because of high total dimension ($148+19 = 167$) of the random effects. This would require working with matrices of column dimension 167, which is probably not a problem for most current hardware, but would be if this number got much larger.

An equivalent way to fit (4) that has a smaller dimension is to treat the clusters identified by primary schools as nested within the entire data, that is, as nested within the "`_all`" group.

```
. xtmelogit attain_gt_6 sex || _all:R.sid || pid:, or variance
Note: factor variables specified; option laplace assumed
  (output omitted)
Mixed-effects logistic regression               Number of obs      =      3435
```

Group Variable	No. of Groups	Observations per Group Minimum	Average	Maximum	Integration Points
_all	1	3435	3435.0	3435	1
pid	148	1	23.2	72	1

```
                                                Wald chi2(1)       =     14.28
Log likelihood = -2220.0035                     Prob > chi2        =    0.0002
```

attain_gt_6	Odds Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
sex	1.32512	.0986967	3.78	0.000	1.145135	1.533395
_cons	.5311498	.0622641	-5.40	0.000	.4221188	.6683427

Random-effects Parameters		Estimate	Std. Err.	[95% Conf.	Interval]
_all: Identity	var(R.sid)	.1239741	.0694743	.0413354	.3718255
pid: Identity	var(_cons)	.4520491	.0953864	.2989334	.6835916

```
LR test vs. logistic regression:     chi2(2) =   195.80   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
Note: log-likelihood calculations are based on the Laplacian approximation.
```

Choosing the primary schools as those to nest was no accident; because there are far fewer secondary schools than primary schools, the above required only 19 random coefficients for the secondary schools, and one random intercept at the primary school level, for a total dimension of 20. Our data also include a measurement of verbal reasoning, variable `vrq`. Adding a fixed effect due to `vrq` in (4) would negate the effect due to secondary school, a fact we leave to you to verify as an exercise. ◁

See [XT] **xtmixed** for a similar discussion of crossed effects in the context of linear mixed models. Also see Rabe-Hesketh and Skrondal (2008, chap. 11) for more examples of crossed-effects models, including models with random interactions, and for more techniques on how to avoid high-dimensional estimation.

❑ Technical note

The estimation in the previous example was performed using a Laplacian approximation, even though we did not specify this. Whenever factor variables are used in random-effects specifications (the `R.`*varname* notation), estimation reverts to the Laplacian method because of the high dimension induced by having factor variables.

In the above example, through some creative nesting we reduced the dimension of the random effects to 20, but this is still too large to permit estimation via adaptive Gaussian quadrature; see *Computation time and the Laplacian approximation*. Even with two quadrature points, our rough formula for computation time would contain within it a factor of $2^{20} = 1{,}048{,}576$.

The `laplace` option is therefore assumed when you use factor variables. If the number of distinct levels of your factors is small enough (say, five or fewer) to permit estimation via AGQ, you can override the imposition of `laplace` by specifying the `intpoints()` option.

❑

Saved results

`xtmelogit` saves the following in `e()`:

Scalars

`e(N)`	number of observations
`e(k)`	number of parameters
`e(k_f)`	number of FE parameters
`e(k_r)`	number of RE parameters
`e(k_rs)`	number of standard deviations
`e(k_rc)`	number of correlations
`e(df_m)`	model degrees of freedom
`e(ll)`	log likelihood
`e(chi2)`	χ^2
`e(p)`	p-value for χ^2
`e(ll_c)`	log likelihood, comparison model
`e(chi2_c)`	χ^2, comparison model
`e(df_c)`	degrees of freedom, comparison model
`e(p_c)`	p-value, comparison model
`e(rank)`	rank of `e(V)`
`e(reparm_rc)`	return code, final reparameterization
`e(rc)`	return code
`e(converged)`	`1` if converged, `0` otherwise

Macros

e(cmd)	xtmelogit
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(ivars)	grouping variables
e(model)	logistic
e(title)	title in estimation output
e(offset)	linear offset variable
e(binomial)	binomial number of trials
e(redim)	random-effects dimensions
e(vartypes)	variance-structure types
e(revars)	random-effects covariates
e(n_quad)	number of integration points
e(laplace)	laplace, if Laplace approximation
e(chi2type)	Wald, type of model χ^2
e(vce)	bootstrap or jackknife if defined
e(vcetype)	title used to label Std. Err.
e(method)	ML
e(opt)	type of optimization
e(ml_method)	type of ml method
e(technique)	maximization technique
e(datasignature)	the checksum
e(datasignaturevars)	variables used in checksum
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(Cns)	constraints matrix
e(N_g)	group counts
e(g_min)	group-size minimums
e(g_avg)	group-size averages
e(g_max)	group-size maximums
e(V)	variance–covariance matrix of the estimator

Functions

e(sample)	marks estimation sample

Methods and formulas

xtmelogit is implemented as an ado-file.

Model (1) assumes Bernoulli data, a special case of the binomial. Because binomial data are also supported by xtmelogit (option binomial()), the methods presented below are in terms of the more general binomial mixed-effects model.

For a two-level binomial model, consider the response y_{ij} as the number of successes from a series of r_{ij} Bernoulli trials (replications). For cluster j, $j = 1, \ldots, M$, the conditional distribution of $\mathbf{y}_j = (y_{j1}, \ldots, y_{jn_j})'$, given a set of cluster-level random effects $\mathbf{u}_j$, is

$$f(\mathbf{y}_j|\mathbf{u}_j) = \prod_{i=1}^{n_j}\left[\binom{r_{ij}}{y_{ij}}\left\{H\left(\mathbf{x}_{ij}\boldsymbol{\beta}+\mathbf{z}_{ij}\mathbf{u}_j\right)\right\}^{y_{ij}}\left\{1-H\left(\mathbf{x}_{ij}\boldsymbol{\beta}+\mathbf{z}_{ij}\mathbf{u}_j\right)\right\}^{r_{ij}-y_{ij}}\right]$$
$$= \exp\left(\sum_{i=1}^{n_j}\left[y_{ij}\left(\mathbf{x}_{ij}\boldsymbol{\beta}+\mathbf{z}_{ij}\mathbf{u}_j\right) - r_{ij}\log\left\{1+\exp\left(\mathbf{x}_{ij}\boldsymbol{\beta}+\mathbf{z}_{ij}\mathbf{u}_j\right)\right\} + \log\binom{r_{ij}}{y_{ij}}\right]\right)$$

for $H(v) = \exp(v)/\{1+\exp(v)\}$.

Defining $\mathbf{r}_j = (r_{j1}, \ldots, r_{jn_j})'$ and

$$c\left(\mathbf{y}_j, \mathbf{r}_j\right) = \sum_{i=1}^{n_j}\log\binom{r_{ij}}{y_{ij}}$$

where $c(\mathbf{y}_j, \mathbf{r}_j)$ does not depend on the model parameters, we can express the above compactly in matrix notation,

$$f(\mathbf{y}_j|\mathbf{u}_j) = \exp\left[\mathbf{y}_j'\left(\mathbf{X}_j\boldsymbol{\beta}+\mathbf{Z}_j\mathbf{u}_j\right) - \mathbf{r}_j'\log\left\{\mathbf{1}+\exp\left(\mathbf{X}_j\boldsymbol{\beta}+\mathbf{Z}_j\mathbf{u}_j\right)\right\} + c\left(\mathbf{y}_j, \mathbf{r}_j\right)\right]$$

where $\mathbf{X}_j$ is formed by stacking the row vectors $\mathbf{x}_{ij}$, $\mathbf{Z}_j$ is formed by stacking the row vectors $\mathbf{z}_{ij}$, and we extend the definitions of the functions $\log()$ and $\exp()$ to be vector functions where necessary.

Because the prior distribution of $\mathbf{u}_j$ is multivariate normal with mean $\mathbf{0}$ and $q \times q$ variance matrix $\boldsymbol{\Sigma}$, the likelihood contribution for the j cluster is obtained by integrating $\mathbf{u}_j$ out the joint density $f(\mathbf{y}_j, \mathbf{u}_j)$,

$$\begin{aligned}\mathcal{L}_i(\boldsymbol{\beta}, \boldsymbol{\Sigma}) &= (2\pi)^{-q/2}\left|\boldsymbol{\Sigma}\right|^{-1/2}\int f(\mathbf{y}_j|\mathbf{u}_j)\exp\left(-\mathbf{u}_j'\boldsymbol{\Sigma}^{-1}\mathbf{u}_j/2\right)d\mathbf{u}_j \\ &= \exp\left\{c\left(\mathbf{y}_j, \mathbf{r}_j\right)\right\}(2\pi)^{-q/2}\left|\boldsymbol{\Sigma}\right|^{-1/2}\int\exp\left\{g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right)\right\}d\mathbf{u}_j\end{aligned} \tag{5}$$

where

$$g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right) = \mathbf{y}_j'\left(\mathbf{X}_j\boldsymbol{\beta}+\mathbf{Z}_j\mathbf{u}_j\right) - \mathbf{r}_j'\log\left\{\mathbf{1}+\exp\left(\mathbf{X}_j\boldsymbol{\beta}+\mathbf{Z}_j\mathbf{u}_j\right)\right\} - \mathbf{u}_j'\boldsymbol{\Sigma}^{-1}\mathbf{u}_j/2$$

and for convenience, in the arguments of $g()$ we suppress the dependence on the observable data $(\mathbf{y}_j, \mathbf{r}_j, \mathbf{X}_j, \mathbf{Z}_j)$.

The integration in (5) has no closed form and thus must be approximated. The Laplacian approximation (Tierney and Kadane 1986; Pinheiro and Bates 1995) is based on a second-order Taylor expansion of $g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right)$ about the value of $\mathbf{u}_j$ that maximizes it. Taking first and second derivatives, we obtain

$$g'\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right) = \frac{\partial g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right)}{\partial \mathbf{u}_j} = \mathbf{Z}_j'\left\{\mathbf{y}_j - \mathbf{m}(\boldsymbol{\beta}, \mathbf{u}_j)\right\} - \boldsymbol{\Sigma}^{-1}\mathbf{u}_j$$

$$g''\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right) = \frac{\partial^2 g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right)}{\partial \mathbf{u}_j \partial \mathbf{u}_j'} = -\left\{\mathbf{Z}_j'\mathbf{V}(\boldsymbol{\beta}, \mathbf{u}_j)\mathbf{Z}_j + \boldsymbol{\Sigma}^{-1}\right\}$$

where $\mathbf{m}(\boldsymbol{\beta}, \mathbf{u}_j)$ is the vector function with ith element equal to the conditional mean of y_{ij} given $\mathbf{u}_j$, that is, $r_{ij}H(\mathbf{x}_{ij}\boldsymbol{\beta}+\mathbf{z}_{ij}\mathbf{u}_j)$. $\mathbf{V}(\boldsymbol{\beta}, \mathbf{u}_j)$ is the diagonal matrix whose diagonal entries v_{ij} are the conditional variances of y_{ij} given $\mathbf{u}_j$, namely,

$$v_{ij} = r_{ij}H\left(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j\right)\left\{1 - H\left(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j\right)\right\}$$

The maximizer of $g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right)$ is $\widehat{\mathbf{u}}_j$ such that $g'\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j\right) = \mathbf{0}$. The integrand in (5) is proportional to the posterior density $f(\mathbf{u}_j|\mathbf{y}_j)$, so $\widehat{\mathbf{u}}_j$ also represents the posterior mode, a plausible estimator of $\mathbf{u}_j$ in its own right.

Given the above derivatives, the second-order Taylor approximation then takes the form

$$g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right) \approx g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j\right) + \frac{1}{2}\left(\mathbf{u}_j - \widehat{\mathbf{u}}_j\right)' g''\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j\right)\left(\mathbf{u}_j - \widehat{\mathbf{u}}_j\right) \tag{6}$$

The first-derivative term vanishes because $g'\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j\right) = \mathbf{0}$. Therefore,

$$\begin{aligned}\int \exp\left\{g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right)\right\} d\mathbf{u}_j &\approx \exp\left\{g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j\right)\right\} \\ &\quad \times \int \exp\left[-\frac{1}{2}\left(\mathbf{u}_j - \widehat{\mathbf{u}}_j\right)'\left\{-g''\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j\right)\right\}\left(\mathbf{u}_j - \widehat{\mathbf{u}}_j\right)\right] d\mathbf{u}_j \\ &= \exp\left\{g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j\right)\right\}(2\pi)^{q/2}\left|-g''\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j\right)\right|^{-1/2}\end{aligned} \tag{7}$$

because the latter integrand can be recognized as the "kernel" of a multivariate normal density.

Combining the above with (5) (and taking logs) gives the Laplacian log-likelihood contribution of the jth cluster,

$$L_j^{\mathrm{Lap}}(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = -\frac{1}{2}\log|\boldsymbol{\Sigma}| - \log|\mathbf{R}_j| + g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j\right) + c(\mathbf{y}_j, \mathbf{r}_j)$$

where $\mathbf{R}_j$ is an upper-triangular matrix such that $-g''\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j\right) = \mathbf{R}_j\mathbf{R}_j'$. Pinheiro and Chao (2006) show that $\widehat{\mathbf{u}}_j$ and $\mathbf{R}_j$ can be efficiently computed as the iterative solution to a least-squares problem by using matrix decomposition methods similar to those used in fitting LME models (Bates and Pinheiro 1998; Pinheiro and Bates 2000; [XT] **xtmixed**).

The fidelity of the Laplacian approximation is determined wholly by the accuracy of the approximation in (6). An alternative that does not depend so heavily on this approximation is integration via adaptive Gaussian quadrature (AGQ; Naylor and Smith 1982; Liu and Pierce 1994).

The application of AGQ to this particular problem is from Pinheiro and Bates (1995). When we reexamine the integral in question, a transformation of integration variables yields

$$\begin{aligned}\int \exp\left\{g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right)\right\} d\mathbf{u}_j &= \left|\mathbf{R}_j\right|^{-1} \int \exp\left\{g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j + \mathbf{R}_j^{-1}\mathbf{t}\right)\right\} d\mathbf{t} \\ &= (2\pi)^{q/2}\left|\mathbf{R}_j\right|^{-1} \int \exp\left\{g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j + \mathbf{R}_j^{-1}\mathbf{t}\right) + \mathbf{t}'\mathbf{t}/2\right\}\phi(\mathbf{t})d\mathbf{t}\end{aligned} \tag{8}$$

where $\phi()$ is the standard multivariate normal density. Because the integrand is now expressed as some function multiplied by a normal density, it can be estimated by applying the rules of standard Gauss–Hermite quadrature. For a predetermined number of quadrature points N_Q, define $a_k = \sqrt{2}a_k^*$ and $w_k = w_k^*/\sqrt{\pi}$, for $k = 1, \ldots, N_Q$, where (a_k^*, w_k^*) are a set of abscissas and weights for Gauss–Hermite quadrature approximations of $\int \exp(-x^2)f(x)dx$, as obtained from Abramowitz and Stegun (1972, 924).

Define $\mathbf{a_k} = (a_{k_1}, a_{k_2}, \ldots, a_{k_q})'$; that is, $\mathbf{a_k}$ is a vector that spans the N_Q abscissas over the dimension q of the random effects. Applying quadrature rules to (8) yields the AGQ approximation,

$$\begin{aligned}
&\int \exp\{g(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j)\}\, d\mathbf{u}_j \\
&\qquad \approx (2\pi)^{q/2} |\mathbf{R}_j|^{-1} \sum_{k_1=1}^{N_Q} \cdots \sum_{k_q=1}^{N_Q} \left[\exp\left\{ g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j + \mathbf{R}_j^{-1}\mathbf{a_k}\right) + \mathbf{a_k}'\mathbf{a_k}/2 \right\} \prod_{p=1}^{q} w_{k_p} \right] \\
&\qquad \equiv (2\pi)^{q/2} \widehat{G}_j(\boldsymbol{\beta}, \boldsymbol{\Sigma})
\end{aligned}$$

resulting in the AGQ log-likelihood contribution of the ith cluster,

$$L_j^{\mathrm{AGQ}}(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = -\frac{1}{2}\log|\boldsymbol{\Sigma}| + \log\left\{\widehat{G}_j(\boldsymbol{\beta}, \boldsymbol{\Sigma})\right\} + c(\mathbf{y}_j, \mathbf{r}_j)$$

The "adaptive" part of adaptive Gaussian quadrature lies in the translation and rescaling of the integration variables in (8) by using $\widehat{\mathbf{u}}_j$ and $\mathbf{R}_j^{-1}$ respectively. This transformation of quadrature abscissas (centered at zero in standard form) is chosen to better capture the features of the integrand, which through (7) can be seen to resemble a multivariate normal distribution with mean $\widehat{\mathbf{u}}_j$ and variance $\mathbf{R}_j^{-1}\mathbf{R}_j^{-T}$. AGQ is therefore not as dependent as the Laplace method upon the approximation in (6). In AGQ, (6) serves merely to redirect the quadrature abscissas, with the AGQ approximation improving as the number of quadrature points, N_Q, increases. In fact, Pinheiro and Bates (1995) point out that AGQ with only one quadrature point ($a = 0$ and $w = 1$) reduces to the Laplacian approximation.

The log likelihood for the entire dataset is then simply the sum of the contributions of the M individual clusters, namely, $L(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = \sum_{j=1}^{M} L_j^{\mathrm{Lap}}(\boldsymbol{\beta}, \boldsymbol{\Sigma})$ for Laplace and $L(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = \sum_{j=1}^{M} L_j^{\mathrm{AGQ}}(\boldsymbol{\beta}, \boldsymbol{\Sigma})$ for adaptive Gaussian quadrature.

Maximization of $L(\boldsymbol{\beta}, \boldsymbol{\Sigma})$ is performed with respect to $(\boldsymbol{\beta}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a vector comprising the unique elements of the matrix square root of $\boldsymbol{\Sigma}$. This is done to ensure that $\boldsymbol{\Sigma}$ is always positive semidefinite. If the `matlog` option is specified, then $\boldsymbol{\theta}$ instead consists of the unique elements of the matrix logarithm of $\boldsymbol{\Sigma}$. For well-conditioned problems both methods produce equivalent results, yet our experience deems the former as more numerically stable near the boundary of the parameter space.

Once maximization is achieved, parameter estimates are mapped from $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\theta}})$ to $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\gamma}})$, where $\widehat{\boldsymbol{\gamma}}$ is a vector containing the unique (estimated) elements of $\boldsymbol{\Sigma}$, expressed as logarithms of standard deviations for the diagonal elements and hyperbolic arctangents of the correlations for off-diagonal elements. This last step is necessary to (a) obtain a parameterization under which parameter estimates can be displayed and interpreted individually, rather than as elements of a matrix square root (or logarithm), and (b) parameterize these elements such that their ranges each encompass the entire real line.

Parameter estimates are stored in `e(b)` as $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\gamma}})$, with the corresponding variance–covariance matrix stored in `e(V)`. Parameter estimates can be displayed in this metric by specifying the `estmetric` option. However, in `xtmelogit` output, variance components are most often displayed either as variances and covariances (the `variance` option) or as standard deviations and correlations (the default).

The approach outlined above can be extended from two-level models to higher-level models; see Pinheiro and Chao (2006) for details.

Acknowledgments

We are indebted to Sophia Rabe-Hesketh, University of California, Berkeley; Anders Skrondal, University of Oslo and Norwegian Institute of Public Health; and Andrew Pickles, University of Manchester, for their extensive body of work in Stata, both previous and ongoing, in this area.

References

Abramowitz, M., and I. A. Stegun, ed. 1972. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. 10th ed. Washington, DC: National Bureau of Standards.

Andrews, M., T. Schank, and R. Upward. 2006. Practical fixed-effects estimation methods for the three-way error-components model. *Stata Journal* 6: 461–481.

Bates, D. M., and J. C. Pinheiro. 1998. Computational methods for multilevel modelling. In *Technical Memorandum BL0112140-980226-01TM*. Murray Hill, NJ: Bell Labs, Lucent Technologies. http://stat.bell-labs.com/NLME/CompMulti.pdf.

Breslow, N. E., and D. G. Clayton. 1993. Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association* 88: 9–25.

Gutierrez, R. G., S. 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.

Harbord, R. M., and P. Whiting. 2009. metandi: Meta-analysis of diagnostic accuracy using hierarchical logistic regression. *Stata Journal* 9: 211–229.

Huq, N. M., and J. Cleland. 1990. *Bangladesh Fertility Survey 1989 (Main Report)*. National Institute of Population Research and Training.

Joe, H. 2008. Accuracy of Laplace approximation for discrete response mixed models. *Computational Statistics & Data Analysis* 52: 5066–5074.

Laird, N. M., and J. H. Ware. 1982. Random-effects models for longitudinal data. *Biometrics* 38: 963–974.

Leyland, A. H., and H. Goldstein, ed. 2001. *Multilevel Modelling of Health Statistics*. New York: Wiley.

Lin, X., and N. E. Breslow. 1996. Bias correction in generalized linear mixed models with multiple components of dispersion. *Journal of the American Statistical Association* 91: 1007–1016.

Littell, R. C., G. A. Milliken, W. W. Stroup, R. D. Wolfinger, and O. Schabenberger. 2006. *SAS System for Mixed Models*. 2nd ed. Cary, NC: SAS Institute.

Liu, Q., and D. A. Pierce. 1994. A note on Gauss–Hermite quadrature. *Biometrika* 81: 624–629.

Marchenko, Y. V. 2006. Estimating variance components in Stata. *Stata Journal* 6: 1–21.

McCulloch, C. E., S. R. Searle, and J. M. Neuhaus. 2008. *Generalized, Linear, and Mixed Models*. 2nd ed. Hoboken, NJ: Wiley.

McLachlan, G. J., and K. E. Basford. 1988. *Mixture Models*. New York: Dekker.

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, Series C* 31: 214–225.

Ng, E. S. W., J. R. Carpenter, H. Goldstein, and J. Rasbash. 2006. Estimation in generalised linear mixed models with binary outcomes by simulated maximum likelihood. *Statistical Modelling* 6: 23–42.

Paterson, L. 1991. Socio-economic status and educational attainment: A multidimensional and multilevel study. *Evaluation and Research in Education* 5: 97–121.

Pinheiro, J. C., and D. M. Bates. 1995. Approximations to the log-likelihood function in the nonlinear mixed-effects model. *Journal of Computational and Graphical Statistics* 4: 12–35.

——. 2000. *Mixed-Effects Models in S and S-PLUS*. New York: Springer.

Pinheiro, J. C., and E. C. Chao. 2006. Efficient Laplacian and adaptive Gaussian quadrature algorithms for multilevel generalized linear mixed models. *Journal of Computational and Graphical Statistics* 15: 58–81.

Rabe-Hesketh, S., and A. Skrondal. 2008. *Multilevel and Longitudinal Modeling Using Stata*. 2nd ed. College Station, TX: Stata Press.

Rabe-Hesketh, S., A. Skrondal, and A. Pickles. 2005. Maximum likelihood estimation of limited and discrete dependent variable models with nested random effects. *Journal of Econometrics* 128: 301–323.

Rabe-Hesketh, S., T. Toulopoulou, and R. M. Murray. 2001. Multilevel modeling of cognitive function in schizophrenic patients and their first degree relatives. *Multivariate Behavioral Research* 36: 279–298.

Rodríguez, G., and N. Goldman. 1995. An assessment of estimation procedures for multilevel models with binary responses. *Journal of the Royal Statistical Society, Series A* 158: 73–89.

Self, S. G., and K.-Y. Liang. 1987. Asymptotic properties of maximum likelihood estimators and likelihood ratio tests under nonstandard conditions. *Journal of the American Statistical Association* 82: 605–610.

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

Tierney, L., and J. B. Kadane. 1986. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association* 81: 82–86.

Also see

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

[XT] **xtmepoisson** — Multilevel mixed-effects Poisson regression

[XT] **xtmixed** — Multilevel mixed-effects linear regression

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

[XT] **xtreg** — Fixed-, between-, and random-effects and population-averaged linear models

[XT] **xtrc** — Random-coefficients model

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

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

[U] **20 Estimation and postestimation commands**

Title

xtmelogit postestimation — Postestimation tools for xtmelogit

The documentation for [XT] **xtmelogit postestimation** has been updated. To see the latest PDF of [XT] **xtmelogit postestimation**, click here.

Description

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

Command	Description
`estat group`	summarize the composition of the nested groups
`estat recovariance`	display the estimated random-effects covariance matrix (or matrices)

For information about these commands, see below.

The following standard postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat`	AIC, BIC, VCE, and estimation sample summary
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`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 coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

See the corresponding entries in the *Base Reference Manual* for details.

Special-interest postestimation commands

`estat group` reports number of groups and minimum, average, and maximum group sizes for each level of the model. Model levels are identified by the corresponding group variable in the data. Because groups are treated as nested, the information in this summary may differ from what you would get if you `tabulate`d each group variable individually.

`estat recovariance` displays the estimated variance–covariance matrix of the random effects for each level in the model. Random effects can be either random intercepts, in which case the corresponding rows and columns of the matrix are labeled as `_cons`, or random coefficients, in which case the label is the name of the associated variable in the data.

Syntax for predict

Syntax for obtaining estimated random effects or their standard errors

```
predict [type] {stub*|newvarlist} [if] [in], {reffects|reses}
        [level(levelvar)]
```

Syntax for obtaining other predictions

```
predict [type] newvar [if] [in] [, statistic fixedonly nooffset]
```

statistic	Description
Main	
mu	the predicted mean; the default
xb	linear prediction for the *fixed* portion of the model only
stdp	standard error of the fixed-portion linear prediction
pearson	Pearson residuals
deviance	deviance residuals
anscombe	Anscombe residuals

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

Menu

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

`reffects` calculates posterior modal estimates of the random effects. By default, estimates for all random effects in the model are calculated. However, if the `level(`*levelvar*`)` option is specified, then estimates for only level *levelvar* in the model are calculated. For example, if `classes` are nested within `schools`, then typing

```
. predict b*, reffects level(school)
```

would yield random-effects estimates at the school level. You must specify q new variables, where q is the number of random-effects terms in the model (or level). However, it is much easier to just specify *stub*`*` and let Stata name the variables *stub*`1`, *stub*`2`, ..., *stub*q for you.

`reses` calculates standard errors for the random-effects estimates obtained by using the `reffects` option. By default, standard errors for all random effects in the model are calculated. However, if the `level(`*levelvar*`)` option is specified, then standard errors for only level *levelvar* in the model are calculated. For example, if `classes` are nested within `schools`, then typing

```
. predict se*, reses level(school)
```

would yield standard errors at the school level. You must specify q new variables, where q is the number of random-effects terms in the model (or level). However, it is much easier to just specify *stub*`*` and let Stata name the variables *stub*`1`, *stub*`2`, ..., *stubq* for you.

The `reffects` and `reses` options often generate multiple new variables at once. When this occurs, the random effects (or standard errors) contained in the generated variables correspond to the order in which the variance components are listed in the output of `xtmelogit`. Still, examining the variable labels of the generated variables (using the `describe` command, for instance) can be useful in deciphering which variables correspond to which terms in the model.

`level(`*levelvar*`)` specifies the level in the model at which predictions for random effects and their standard errors are to be obtained. *levelvar* is the name of the model level and is either the name of the variable describing the grouping at that level or `_all`, a special designation for a group comprising all the estimation data.

`mu`, the default, calculates the predicted mean. By default, this is based on a linear predictor that includes *both* the fixed effects and the random effects, and the predicted mean is conditional on the values of the random effects. Use the `fixedonly` option (see below) if you want predictions that include only the fixed portion of the model, that is, if you want random effects set to zero.

`xb` calculates the linear prediction $\mathbf{x}\boldsymbol{\beta}$ based on 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 fixed-effects linear predictor $\mathbf{x}\boldsymbol{\beta}$.

`pearson` calculates Pearson residuals. Pearson residuals large in absolute value may indicate a lack of fit. By default, residuals include both the fixed portion and the random portion of the model. The `fixedonly` option modifies the calculation to include the fixed portion only.

`deviance` calculates deviance residuals. Deviance residuals are recommended by McCullagh and Nelder (1989) as having the best properties for examining the goodness of fit of a GLM. They are approximately normally distributed if the model is correctly specified. They may be plotted against the fitted values or against a covariate to inspect the model's fit. By default, residuals include both the fixed portion and the random portion of the model. The `fixedonly` option modifies the calculation to include the fixed portion only.

`anscombe` calculates Anscombe residuals, residuals that are designed to closely follow a normal distribution. By default, residuals include both the fixed portion and the random portion of the model. The `fixedonly` option modifies the calculation to include the fixed portion only.

`fixedonly` modifies predictions to include only the fixed portion of the model, equivalent to setting all random effects equal to zero; see above.

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

Syntax for estat group

```
estat group
```

Menu

Statistics > Postestimation > Reports and statistics

Syntax for estat recovariance

```
estat recovariance [ , level(levelvar) correlation matlist_options ]
```

Menu

Statistics > Postestimation > Reports and statistics

Options for estat recovariance

`level(`*levelvar*`)` specifies the level in the model for which the random-effects covariance matrix is to be displayed and returned in `r(cov)`. By default, the covariance matrices for all levels in the model are displayed. *levelvar* is the name of the model level and is either the name of variable describing the grouping at that level or `_all`, a special designation for a group comprising all the estimation data.

`correlation` displays the covariance matrix as a correlation matrix and returns the correlation matrix in `r(corr)`.

matlist_options are style and formatting options that control how the matrix (or matrices) are displayed; see [P] **matlist** for a list of what is available.

Remarks

Various predictions, statistics, and diagnostic measures are available after fitting a logistic mixed-effects model with `xtmelogit`. For the most part, calculation centers around obtaining estimates of the subject/group-specific random effects. Random effects are not provided as estimates when the model is fit but instead need to be predicted after estimation.

▷ Example 1

In example 3 of [XT] **xtmelogit**, we represented the probability of contraceptive use among Bangladeshi women by using the model (stated with slightly different notation here)

$$\begin{aligned}\text{logit}(\pi_{ij}) = \beta_0\texttt{rural}_{ij}+\beta_1\texttt{urban}_{ij} + \beta_2\texttt{age}_{ij}+ \\ \beta_3\texttt{child1}_{ij} + \beta_4\texttt{child2}_{ij} + \beta_5\texttt{child3}_{ij} + a_j\texttt{rural}_{ij} + b_j\texttt{urban}_{ij}\end{aligned}$$

where π_{ij} is the probability of contraceptive use, $j = 1, \ldots, 60$ districts, $i = 1, \ldots, n_j$ women within each district, and a_j and b_j are normally distributed with mean zero and variance–covariance matrix

$$\boldsymbol{\Sigma} = \text{Var}\begin{bmatrix} a_j \\ b_j \end{bmatrix} = \begin{bmatrix} \sigma_a^2 & \sigma_{ab} \\ \sigma_{ab} & \sigma_b^2 \end{bmatrix}$$

```
. use http://www.stata-press.com/data/r12/bangladesh
(Bangladesh Fertility Survey, 1989)
. generate byte rural = 1 - urban
. xtmelogit c_use rural urban age child*, nocons || district: rural urban,
> nocons cov(unstructured)
  (output omitted)
Mixed-effects logistic regression               Number of obs      =      1934
Group variable: district                        Number of groups   =        60
                                                Obs per group: min =         2
                                                               avg =      32.2
                                                               max =       118
Integration points =   7                        Wald chi2(6)       =    120.24
Log likelihood =  -1199.315                     Prob > chi2        =    0.0000
```

c_use	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
rural	-1.71165	.1605618	-10.66	0.000	-2.026345	-1.396954
urban	-.8958623	.1704961	-5.25	0.000	-1.230028	-.5616962
age	-.026415	.008023	-3.29	0.001	-.0421398	-.0106902
child1	1.13252	.1603285	7.06	0.000	.818282	1.446758
child2	1.357739	.1770522	7.67	0.000	1.010724	1.704755
child3	1.353827	.1828801	7.40	0.000	.9953882	1.712265

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
district: Unstructured				
sd(rural)	.6242947	.1035136	.4510794	.8640251
sd(urban)	.4942636	.146751	.2762039	.8844789
corr(rural,urban)	-.0523099	.3384599	-.6153876	.5461173

```
LR test vs. logistic regression:     chi2(3) =    58.42   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

Rather than see the estimated variance components listed as standard deviations and correlations as above, we can instead see them as variance–covariances in matrix form; that is, we can see $\widehat{\Sigma}$

```
. estat recovariance
Random-effects covariance matrix for level district
```

	rural	urban
rural	.3897439	
urban	-.0161411	.2442965

or we can see $\widehat{\Sigma}$ as a correlation matrix

```
. estat recovariance, correlation
Random-effects correlation matrix for level district
```

	rural	urban
rural	1	
urban	-.0523099	1

The purpose of using this particular model was to allow for district random effects that were specific to the rural and urban areas of that district and that could be interpreted as such. We can obtain predictions of these random effects

```
. predict re_rural re_urban, reffects
```

and their corresponding standard errors

```
. predict se_rural se_urban, reses
```

The order in which we specified the variables to be generated corresponds to the order in which the variance components are listed in `xtmelogit` output. If in doubt, a simple `describe` will show how these newly generated variables are labeled just to be sure.

Having generated estimated random effects and standard errors, we can now list them for the first 10 districts:

```
. by district, sort: generate tolist = (_n==1)
. list district re_rural se_rural re_urban se_urban if district <= 10 & tolist,
> sep(0)
```

	district	re_rural	se_rural	re_urban	se_urban
1.	1	-.9206641	.3129662	-.5551252	.2321872
118.	2	-.0307772	.3784629	.0012746	.4938357
138.	3	-.0149148	.6242095	.2257356	.4689535
140.	4	-.2684802	.3951617	.5760575	.3970433
170.	5	.0787537	.3078451	.004534	.4675104
209.	6	-.3842217	.2741989	.2727722	.4184852
274.	7	-.1742786	.4008164	.0072177	.493866
292.	8	.0447142	.315396	.2256406	.46799
329.	9	-.3561363	.3885605	.0733451	.4555067
352.	10	-.5368572	.4743089	.0222338	.4939776

◁

❑ Technical note

When these data were first introduced in [XT] **xtmelogit**, we noted that not all districts contained both urban and rural areas. This fact is somewhat demonstrated by the random effects that are nearly zero in the above. A closer examination of the data would reveal that district 3 has no rural areas, and districts 2, 7, and 10 have no urban areas.

The estimated random effects are not exactly zero in these cases is because of the correlation between urban and rural effects. For instance, if a district has no urban areas, it can still yield a nonzero (albeit small) random-effect estimate for a nonexistent urban area because of the correlation with its rural counterpart.

Had we imposed an independent covariance structure in our model, the estimated random effects in the cases in question would be exactly zero.

❑

❑ Technical note

The estimated standard errors produced above using the `reses` option are conditional on the values of the estimated model parameters: β and the components of Σ. Their interpretation is therefore not one of standard sample-to-sample variability but instead one that does not incorporate uncertainty in the estimated model parameters; see *Methods and formulas*.

That stated, conditional standard errors can still be used as a measure of relative precision, provided that you keep this caveat in mind.

❑

▷ Example 2

Continuing with example 1, we can obtain predicted probabilities, the default prediction:

```
. predict p
(option mu assumed; predicted means)
```

These predictions are based on a linear predictor that includes *both* the fixed effects and random effects due to district. Specifying the `fixedonly` option gives predictions that set the random effects to their prior mean of zero. Below, we compare both over the first 20 observations:

```
. predict p_fixed, fixedonly
(option mu assumed; predicted means)
. list c_use p p_fixed age child* in 1/20
```

	c_use	p	p_fixed	age	child1	child2	child3
1.	no	.3579543	.4927183	18.44	0	0	1
2.	no	.2134724	.3210403	-5.56	0	0	0
3.	no	.4672256	.6044016	1.44	0	1	0
4.	no	.4206505	.5584864	8.44	0	0	1
5.	no	.2510909	.3687281	-13.56	0	0	0
6.	no	.2412878	.3565185	-11.56	0	0	0
7.	no	.3579543	.4927183	18.44	0	0	1
8.	no	.4992191	.6345999	-3.56	0	0	1
9.	no	.4572049	.594723	-5.56	1	0	0
10.	no	.4662518	.6034657	1.44	0	0	1
11.	yes	.2412878	.3565185	-11.56	0	0	0
12.	no	.2004691	.3040173	-2.56	0	0	0
13.	no	.4506573	.5883407	-4.56	1	0	0
14.	no	.4400747	.5779263	5.44	0	0	1
15.	no	.4794194	.6160359	-0.56	0	0	1
16.	yes	.4465936	.5843561	4.44	0	0	1
17.	no	.2134724	.3210403	-5.56	0	0	0
18.	yes	.4794194	.6160359	-0.56	0	0	1
19.	yes	.4637673	.6010735	-6.56	1	0	0
20.	no	.5001973	.6355067	-3.56	0	1	0

◁

❑ Technical note

Out-of-sample predictions are permitted after `xtmelogit`, but if these predictions involve estimated random effects, the integrity of the estimation data must be preserved. If the estimation data have changed since the model was fit, `predict` will be unable to obtain predicted random effects that are appropriate for the fitted model and will give an error. Thus, to obtain out-of-sample predictions that contain random-effects terms, be sure that the data for these predictions are in observations that augment the estimation data.

❑

Saved results

`estat recovariance` saves the last-displayed random-effects covariance matrix in `r(cov)` or in `r(corr)` if it is displayed as a correlation matrix.

Methods and formulas

Continuing the discussion in *Methods and formulas* of [XT] **xtmelogit**, and using the definitions and formulas defined there, we begin by considering the "prediction" of the random effects $\mathbf{u}_j$ for the jth cluster in a two-level model.

Given a set of estimated `xtmelogit` parameters, $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\Sigma}})$, a profile likelihood in $\mathbf{u}_j$ is derived from the joint distribution $f(\mathbf{y}_j, \mathbf{u}_j)$ as

$$\mathcal{L}_j(\mathbf{u}_j) = \exp\left\{c\left(\mathbf{y}_j, \mathbf{r}_j\right)\right\}(2\pi)^{-q/2}|\widehat{\boldsymbol{\Sigma}}|^{-1/2}\exp\left\{g\left(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\Sigma}}, \mathbf{u}_j\right)\right\} \tag{1}$$

The conditional MLE of $\mathbf{u}_j$—conditional on fixed $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\Sigma}})$—is the maximizer of $\mathcal{L}_j(\mathbf{u}_j)$, or equivalently, the value of $\widehat{\mathbf{u}}_j$ that solves

$$\mathbf{0} = g'\left(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\Sigma}}, \widehat{\mathbf{u}}_j\right) = \mathbf{Z}_j'\left\{\mathbf{y}_j - \mathbf{m}(\widehat{\boldsymbol{\beta}}, \widehat{\mathbf{u}}_j)\right\} - \widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\mathbf{u}}_j$$

Because (1) is proportional to the conditional density $f(\mathbf{u}_j|\mathbf{y}_j)$, you can also refer to $\widehat{\mathbf{u}}_j$ as the *conditional mode* (or *posterior mode* if you lean toward Bayesian terminology). Regardless, you are referring to the same estimator.

Conditional standard errors for the estimated random effects are derived from standard theory of maximum likelihood, which dictates that the asymptotic variance matrix of $\widehat{\mathbf{u}}_j$ is the negative inverse of the Hessian, which is estimated as

$$g''\left(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\Sigma}}, \widehat{\mathbf{u}}_j\right) = -\left\{\mathbf{Z}_j'\mathbf{V}(\widehat{\boldsymbol{\beta}}, \widehat{\mathbf{u}}_j)\mathbf{Z}_j + \widehat{\boldsymbol{\Sigma}}^{-1}\right\}$$

Similar calculations extend to models with more than one level of random effects; see Pinheiro and Chao (2006).

For any i observation in the j cluster in a two-level model, define the linear predictor as

$$\widehat{\eta}_{ij} = \mathbf{x}_{ij}\widehat{\boldsymbol{\beta}} + \mathbf{z}_{ij}\widehat{\mathbf{u}}_j$$

In a three-level model, for the ith observation within the jth level-two cluster within the kth level-three cluster,

$$\widehat{\eta}_{ijk} = \mathbf{x}_{ijk}\widehat{\boldsymbol{\beta}} + \mathbf{z}^{(3)}_{ijk}\widehat{\mathbf{u}}^{(3)}_k + \mathbf{z}^{(2)}_{ijk}\widehat{\mathbf{u}}^{(2)}_{jk}$$

where the $\mathbf{z}^{(p)}$ and $\mathbf{u}^{(p)}$ refer to the level p design variables and random effects, respectively. For models with more than three levels, the definition of $\widehat{\eta}$ extends in the natural way, with only the notation becoming more complicated.

If the `fixedonly` option is specified, $\widehat{\eta}$ contains the linear predictor for only the fixed portion of the model, for example, in a two-level model $\widehat{\eta}_{ij} = \mathbf{x}_{ij}\widehat{\boldsymbol{\beta}}$. In what follows, we assume a two-level model, with the only necessary modification for multilevel models being the indexing.

The predicted mean, conditional on the random effects $\widehat{\mathbf{u}}_j$, is

$$\widehat{\mu}_{ij} = r_{ij}H(\widehat{\eta}_{ij})$$

Pearson residuals are calculated as

$$\nu_{ij}^{P} = \frac{y_{ij} - \widehat{\mu}_{ij}}{\{V(\widehat{\mu}_{ij})\}^{1/2}}$$

for $V(\widehat{\mu}_{ij}) = \widehat{\mu}_{ij}(1 - \widehat{\mu}_{ij}/r_{ij})$.

Deviance residuals are calculated as

$$\nu_{ij}^{D} = \text{sign}(y_{ij} - \widehat{\mu}_{ij})\sqrt{\widehat{d}_{ij}^{2}}$$

where

$$\widehat{d}_{ij}^{2} = \begin{cases} 2r_{ij}\log\left(\dfrac{r_{ij}}{r_{ij} - \widehat{\mu}_{ij}}\right) & \text{if } y_{ij} = 0 \\ 2y_{ij}\log\left(\dfrac{y_{ij}}{\widehat{\mu}_{ij}}\right) + 2(r_{ij} - y_{ij})\log\left(\dfrac{r_{ij} - y_{ij}}{r_{ij} - \widehat{\mu}_{ij}}\right) & \text{if } 0 < y_{ij} < r_{ij} \\ 2r_{ij}\log\left(\dfrac{r_{ij}}{\widehat{\mu}_{ij}}\right) & \text{if } y_{ij} = r_{ij} \end{cases}$$

Anscombe residuals are calculated as

$$\nu_{ij}^{A} = \frac{3\left\{y_{ij}^{2/3}\mathcal{H}(y_{ij}/r_{ij}) - \widehat{\mu}^{2/3}\mathcal{H}(\widehat{\mu}_{ij}/r_{ij})\right\}}{2\left(\widehat{\mu}_{ij} - \widehat{\mu}_{ij}^{2}/r_{ij}\right)^{1/6}}$$

where $\mathcal{H}(t)$ is a specific univariate case of the Hypergeometric2F1 function (Wolfram 1999, 771–772). For Anscombe residuals for binomial regression, the specific form of the Hypergeometric2F1 function that we require is $\mathcal{H}(t) = {}_2F_1(2/3, 1/3, 5/3, t)$.

For a discussion of the general properties of the above residuals, see Hardin and Hilbe (2007, chap. 4).

References

Hardin, J. W., and J. M. Hilbe. 2007. *Generalized Linear Models and Extensions*. 2nd ed. College Station, TX: Stata Press.

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

Pinheiro, J. C., and E. C. Chao. 2006. Efficient Laplacian and adaptive Gaussian quadrature algorithms for multilevel generalized linear mixed models. *Journal of Computational and Graphical Statistics* 15: 58–81.

Rabe-Hesketh, S., and A. Skrondal. 2008. *Multilevel and Longitudinal Modeling Using Stata*. 2nd ed. College Station, TX: Stata Press.

Wolfram, S. 1999. *The Mathematica Book*. 4th ed. Cambridge: Cambridge University Press.

Also see

[XT] **xtmelogit** — Multilevel mixed-effects logistic regression

[U] **20 Estimation and postestimation commands**

Title

xtmepoisson — Multilevel mixed-effects Poisson regression

Syntax

`xtmepoisson` *depvar fe_equation* `||` *re_equation* [`||` *re_equation* ...] [`,` *options*]

where the syntax of *fe_equation* is

[*indepvars*] [*if*] [*in*] [`,` *fe_options*]

and the syntax of *re_equation* is one of the following:

for random coefficients and intercepts

levelvar`:` [*varlist*] [`,` *re_options*]

for random effects among the values of a factor variable

levelvar`: R.`*varname* [`,` *re_options*]

levelvar is a variable identifying the group structure for the random effects at that level or `_all` representing one group comprising all observations.

fe_options	Description
Model	
`noconstant`	suppress constant term from the fixed-effects equation
`exposure(`*varname*$_e$`)`	include ln(*varname*$_e$) in model with coefficient constrained to 1
`offset(`*varname*$_o$`)`	include *varname*$_o$ in model with coefficient constrained to 1

re_options	Description
Model	
`covariance(`*vartype*`)`	variance–covariance structure of the random effects
`noconstant`	suppress the constant term from the random-effects equation
`collinear`	keep collinear variables

options	Description
Integration	
`laplace`	use Laplacian approximation; equivalent to `intpoints(1)`
`intpoints(#` [`#` ...]`)`	set the number of integration (quadrature) points; default is 7

Reporting

level(#)	set confidence level; default is level(95)
irr	report fixed-effects coefficients as incidence-rate ratios
variance	show random-effects parameter estimates as variances and covariances
noretable	suppress random-effects table
nofetable	suppress fixed-effects table
estmetric	show parameter estimates in the estimation metric
noheader	suppress output header
nogroup	suppress table summarizing groups
nolrtest	do not perform LR test comparing to Poisson regression
display_options	control column formats, row spacing, line width, and display of omitted variables and base and empty cells

Maximization

maximize_options	control the maximization process during gradient-based optimization; seldom used
retolerance(#)	tolerance for random-effects estimates; default is retolerance(1e-8); seldom used
reiterate(#)	maximum number of iterations for random-effects estimation; default is reiterate(50); seldom used
matsqrt	parameterize variance components using matrix square roots; the default
matlog	parameterize variance components using matrix logarithms
refineopts(*maximize_options*)	control the maximization process during refinement of starting values
coeflegend	display legend instead of statistics

vartype	Description
independent	one unique variance parameter per random effect, all covariances zero; the default unless a factor variable is specified
exchangeable	equal variances for random effects, and one common pairwise covariance
identity	equal variances for random effects, all covariances zero; the default if factor variables are specified
unstructured	all variances–covariances distinctly estimated

indepvars may contain factor variables; see [U] **11.4.3 Factor variables**.
indepvars and *varlist* may contain time-series operators; see [U] **11.4.4 Time-series varlists**.
bootstrap, by, jackknife, mi estimate, rolling, 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.

Menu

Statistics > Longitudinal/panel data > Multilevel mixed-effects models > Mixed-effects Poisson regression

Description

`xtmepoisson` fits mixed-effects models for count responses. Mixed models contain both *fixed effects* and *random effects*. The fixed effects are analogous to standard regression coefficients and are estimated directly. The random effects are not directly estimated (although they may be obtained postestimation) but are summarized according to their estimated variances and covariances. Random effects may take the form of either random intercepts or random coefficients, and the grouping structure of the data may consist of multiple levels of nested groups. The distribution of the random effects is assumed to be Gaussian. The conditional distribution of the response given the random effects is assumed to be Poisson. Because the log likelihood for this model has no closed form, it is approximated by adaptive Gaussian quadrature.

Options

Model

`noconstant` suppresses the constant (intercept) term and may be specified for the fixed-effects equation and for any or all the random-effects equations.

`exposure(`*varname*$_e$`)` specifies a variable that reflects the amount of exposure over which the *depvar* events were observed for each observation; ln(*varname*$_e$) is included in the fixed-effects portion of the model with the coefficient constrained to be 1.

`offset(`*varname*$_o$`)` specifies that *varname*$_o$ be included in the fixed-effects portion of the model with the coefficient constrained to be 1.

`covariance(`*vartype*`)`, where *vartype* is

`independent` | `exchangeable` | `identity` | `unstructured`

specifies the structure of the covariance matrix for the random effects and may be specified for each random-effects equation. An `independent` covariance structure allows for a distinct variance for each random effect within a random-effects equation and assumes that all covariances are zero. `exchangeable` structure specifies one common variance for all random effects and one common pairwise covariance. `identity` is short for "multiple of the identity"; that is, all variances are equal and all covariances are zero. `unstructured` allows for all variances and covariances to be distinct. If an equation consists of p random-effects terms, the unstructured covariance matrix will have $p(p+1)/2$ unique parameters.

`covariance(independent)` is the default, except when the random-effects equation is a factor-variable specification `R.`*varname*, in which case `covariance(identity)` is the default, and only `covariance(identity)` and `covariance(exchangeable)` are allowed.

`collinear` specifies that `xtmepoisson` not omit collinear variables from the random-effects equation. Usually there is no reason to leave collinear variables in place, and in fact doing so usually causes the estimation to fail because of the matrix singularity caused by the collinearity. However, with certain models (for example, a random-effects model with a full set of contrasts), the variables may be collinear, yet the model is fully identified because of restrictions on the random-effects covariance structure. In such cases, using the `collinear` option allows the estimation to take place with the random-effects equation intact.

Integration

`laplace` specifies that log likelihoods be calculated using the Laplacian approximation, equivalent to adaptive Gaussian quadrature with one integration point for each level in the model; `laplace` is equivalent to `intpoints(1)`. Computation time increases as a function of the number of quadrature points raised to a power equaling the dimension of the random-effects specification. The computational time saved by using `laplace` can thus be substantial, especially when you have many levels and/or random coefficients.

The Laplacian approximation has been known to produce biased parameter estimates, but the bias tends to be more prominent in the estimates of the variance components rather than in estimates of the fixed effects. If your interest lies primarily with the fixed-effects estimates, the Laplace approximation may be a viable faster alternative to adaptive quadrature with multiple integration points.

Specifying a factor variable, `R.`*varname*, increases the dimension of the random effects by the number of distinct values of *varname*, that is, the number of factor levels. Even when this number is small to moderate, it increases the total random-effects dimension to the point where estimation with more than one quadrature point is prohibitively intensive.

For this reason, when you have factor variables in your random-effects equations, the `laplace` option is assumed. You can override this behavior by using the `intpoints()` option, but doing so is not recommended.

`intpoints(#[# ...])` sets the number of integration points for adaptive Gaussian quadrature. The more points, the more accurate the approximation to the log likelihood. However, computation time increases with the number of quadrature points, and in models with many levels and/or many random coefficients, this increase can be substantial.

You may specify one number of integration points applying to all levels of random effects in the model, or you may specify distinct numbers of points for each level. `intpoints(7)` is the default; that is, by default seven quadrature points are used for each level.

Reporting

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

`irr` reports the fixed-effects coefficients transformed to incidence-rate ratios, that is, $\exp(b)$ rather than b. Standard errors and confidence intervals are similarly transformed. This option affects how results are displayed, not how they are estimated. `irr` may be specified at estimation or when replaying previously estimated results.

`variance` displays the random-effects parameter estimates as variances and covariances. The default is to display them as standard deviations and correlations.

`noretable` suppresses the table of random effects.

`nofetable` suppresses the table of fixed effects.

`estmetric` displays all parameter estimates in the estimation metric. Fixed-effects estimates are unchanged from those normally displayed, but random-effects parameter estimates are displayed as log-standard deviations and hyperbolic arctangents of correlations, with equation names that organize them by model level.

`noheader` suppresses the output header, either at estimation or upon replay.

`nogroup` suppresses the display of group summary information (number of groups, average group size, minimum, and maximum) from the output header.

nolrtest prevents xtmepoisson from performing a likelihood-ratio test that compares the mixed-effects Poisson model with standard (marginal) Poisson regression. This option may also be specified upon replay to suppress this test from the output.

display_options: noomitted, vsquish, noemptycells, baselevels, allbaselevels, 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**. Those that require special mention for xtmepoisson are listed below.

For the technique() option, the default is technique(nr). The bhhh algorithm may not be specified.

from(*init_specs*) is particularly useful when combined with refineopts(iterate(0)), which bypasses the initial optimization stage; see below.

retolerance(#) specifies the convergence tolerance for the estimated random effects used by adaptive Gaussian quadrature. Although not estimated as model parameters, random-effects estimators are used to adapt the quadrature points. Estimating these random effects is an iterative procedure, with convergence declared when the maximum relative change in the random effects is less than retolerance(). The default is retolerance(1e-8). You should seldom have to use this option.

reiterate(#) specifies the maximum number of iterations used when estimating the random effects to be used in adapting the Gaussian quadrature points; see the retolerance() option. The default is reiterate(50). You should seldom have to use this option.

matsqrt (the default), during optimization, parameterizes variance components by using the matrix square roots of the variance–covariance matrices formed by these components at each model level.

matlog, during optimization, parameterizes variance components by using the matrix logarithms of the variance–covariance matrices formed by these components at each model level.

The matsqrt parameterization ensures that variance–covariance matrices are positive semidefinite, while matlog ensures matrices that are positive definite. For most problems, the matrix square root is more stable near the boundary of the parameter space. However, if convergence is problematic, one option may be to try the alternate matlog parameterization. When convergence is not an issue, both parameterizations yield equivalent results.

refineopts(*maximize_options*) controls the maximization process during the refinement of starting values. Estimation in xtmepoisson takes place in two stages. In the first stage, starting values are refined by holding the quadrature points fixed between iterations. During the second stage, quadrature points are adapted with each evaluation of the log likelihood. Maximization options specified within refineopts() control the first stage of optimization; that is, they control the refining of starting values.

maximize_options specified outside refineopts() control the second stage.

The one exception to the above rule is the nolog option, which when specified outside refineopts() applies globally.

from(*init_specs*) is not allowed within refineopts() and instead must be specified globally.

Refining starting values helps make the iterations of the second stage (those that lead toward the solution) more numerically stable. In this regard, of particular interest is refineopts(iterate(#)), with two iterations being the default. Should the maximization fail because of instability in the Hessian calculations, one possible solution may be to increase the number of iterations here.

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

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

Remarks

Remarks are presented under the following headings:

Introduction

Mixed-effects Poisson regression is Poisson regression containing both fixed effects and random effects. In longitudinal/panel data, random effects are useful for modeling intracluster correlation; that is, observations in the same cluster are correlated because they share common cluster-level random effects.

`xtmepoisson` allows for not just one, but many levels of nested clusters. For example, in a three-level model you can specify random effects for schools and then random effects for classes nested within schools. The observations (students, presumably) would comprise level one of the model, the classes would comprise level two, and the schools would comprise level three.

However, for simplicity, for now we consider the two-level model where, for a series of M independent clusters and, conditional on a set of *random effects* $\mathbf{u}_j$,

$$\Pr(y_{ij} = y|\mathbf{u}_j) = \exp\left(-\mu_{ij}\right)\mu_{ij}^{y}/y! \tag{1}$$

for $\mu_{ij} = \exp(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j)$, $j = 1, \ldots, M$ clusters, and with cluster j consisting of $i = 1, \ldots, n_j$ observations. The responses are counts y_{ij}. The $1 \times p$ row vector $\mathbf{x}_{ij}$ are the covariates for the fixed effects, analogous to the covariates you would find in a standard Poisson regression model, with regression coefficients (fixed effects) $\boldsymbol{\beta}$.

The $1 \times q$ vector $\mathbf{z}_{ij}$ are the covariates corresponding to the random effects and can be used to represent both random intercepts and random coefficients. For example, in a random-intercept model, $\mathbf{z}_{ij}$ is simply the scalar 1. The random effects $\mathbf{u}_j$ are M realizations from a multivariate normal distribution with mean $\mathbf{0}$ and $q \times q$ variance matrix $\boldsymbol{\Sigma}$. The random effects are not directly estimated as model parameters but are instead summarized according to the unique elements of $\boldsymbol{\Sigma}$, known as *variance components*. One special case of (1) places $\mathbf{z}_{ij} = \mathbf{x}_{ij}$, so that all covariate effects are essentially random and distributed as multivariate normal with mean $\boldsymbol{\beta}$ and variance $\boldsymbol{\Sigma}$.

Model (1) is a member of the class of generalized linear mixed models (GLMMs), which generalize the linear mixed-effects (LME) model to non-Gaussian responses. In particular, model (1) deals with count responses. Stata also has the `xtmelogit` command for fitting another type of GLMM, the logistic model for binary and binomial responses.

From a general prospective, there is not much to distinguish `xtmepoisson` from `xtmelogit`, and most everything said about `xtmelogit` in [XT] **xtmelogit** applies to `xtmepoisson`. If you are anxious to get started applying `xtmepoisson` to your count data, continue reading this entry. Examples are provided below.

We encourage you to read [XT] **xtmelogit**, however. In addition to some history and guided tours of syntax and output, substantive issues are discussed, and these apply equally to Poisson data. These include conventions for multilevel terminology, specifying covariance structures for random effects, constructing complex blocked-diagonal covariance structures, distribution theory for likelihood-ratio tests, factors that affect computation time, the Laplacian approximation, advice on model building, and fitting crossed-effects models.

A two-level model

We begin with a simple application of (1). We begin with a two-level model because, in multilevel-model terminology, a one-level model is just standard Poisson regression; see [R] **poisson**.

▷ Example 1

Breslow and Clayton (1993) fit a mixed-effects Poisson model to data from a randomized trial of the drug progabide for the treatment of epilepsy.

```
. use http://www.stata-press.com/data/r12/epilepsy
(Epilepsy data; progabide drug treatment)
. describe
Contains data from http://localpress.stata.com/data/r12/epilepsy.dta
  obs:           236                          Epilepsy data; progabide drug
                                                treatment
 vars:             8                          31 May 2011 14:09
 size:         4,956                          (_dta has notes)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
subject         byte   %9.0g                  Subject ID: 1-59
seizures        int    %9.0g                  No. of seizures
treat           byte   %9.0g                  1: progabide; 0: placebo
visit           float  %9.0g                  Dr. visit; coded as (-.3, -.1,
                                                .1, .3)
lage            float  %9.0g                  log(age), mean-centered
lbas            float  %9.0g                  log(0.25*baseline seizures),
                                                mean-centered
lbas_trt        float  %9.0g                  lbas/treat interaction
v4              byte   %8.0g                  Fourth visit indicator
-------------------------------------------------------------------------------
Sorted by:  subject
```

Originally from Thall and Vail (1990), data were collected on 59 subjects (31 on progabide, 28 placebo). The number of epileptic seizures (`seizures`) was recorded during the two weeks prior to each of four doctor visits (`visit`). The treatment group is identified by the indicator variable `treat`. Data were also collected on the logarithm of age (`lage`) and the logarithm of one-quarter the number of seizures during the eight weeks prior to the study (`lbas`). Variable `lbas_trt` represents the interaction between `lbas` and treatment. `lage`, `lbas`, and `lbas_trt` are mean centered. Because the study originally noted a substantial decrease in seizures prior to the fourth doctor visit, an indicator, `v4`, for the fourth visit was also recorded.

Breslow and Clayton (1993) fit a random-effects Poisson model for the number of observed seizures

$$\log(\mu_{ij}) = \beta_0 + \beta_1 \texttt{treat}_{ij} + \beta_2 \texttt{lbas}_{ij} + \beta_3 \texttt{lbas\_trt}_{ij} + \beta_4 \texttt{lage}_{ij} + \beta_5 \texttt{v4}_{ij} + u_j$$

for $j = 1, \ldots, 59$ subjects and $i = 1, \ldots, 4$ visits. The random effects u_j are assumed to be normally distributed with mean zero and variance σ_u^2.

```
. xtmepoisson seizures treat lbas lbas_trt lage v4 || subject:
Refining starting values:
Iteration 0:   log likelihood = -680.40577  (not concave)
Iteration 1:   log likelihood = -668.60112
Iteration 2:   log likelihood = -666.37033
Performing gradient-based optimization:
Iteration 0:   log likelihood = -666.37033
Iteration 1:   log likelihood = -665.45248
Iteration 2:   log likelihood = -665.29074
Iteration 3:   log likelihood = -665.29068
Mixed-effects Poisson regression                Number of obs      =       236
Group variable: subject                         Number of groups   =        59
                                                Obs per group: min =         4
                                                               avg =       4.0
                                                               max =         4
Integration points =   7                        Wald chi2(5)       =    121.67
Log likelihood = -665.29068                     Prob > chi2        =    0.0000
```

seizures	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
treat	-.9330384	.4008345	-2.33	0.020	-1.71866	-.1474172
lbas	.8844329	.1312313	6.74	0.000	.6272243	1.141642
lbas_trt	.3382606	.2033384	1.66	0.096	-.0602754	.7367966
lage	.4842383	.3472776	1.39	0.163	-.1964134	1.16489
v4	-.1610871	.0545758	-2.95	0.003	-.2680536	-.0541206
_cons	2.154574	.2200426	9.79	0.000	1.723299	2.58585

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
subject: Identity				
sd(_cons)	.5028187	.0586255	.4000983	.6319115

```
LR test vs. Poisson regression:  chibar2(01) =   304.74 Prob>=chibar2 = 0.0000
```

The number of seizures before the fourth visit does exhibit a significant drop, and the patients on progabide demonstrate a decrease in frequency of seizures compared with the placebo group. The subject-specific random effects also appear significant, $\widehat{\sigma}_u = 0.503$ with standard error 0.059. The above results are also in good agreement with those of Breslow and Clayton (1993, table 4), who fit this model by the method of penalized quasilikelihood (PQL).

Because this is a simple random-intercept model, you can obtain equivalent results by using `xtpoisson` with the `re` and `normal` options.

See *Two-level models* in [XT] **xtmelogit** for a detailed description of syntax and of reading the resulting output.

◁

▷ Example 2

In their study of PQL, Breslow and Clayton (1993) also fit a model where they dropped the fixed effect on `v4` and replaced it with a random subject-specific linear trend over the four doctor visits. The model they fit is

$$\log(\mu_{ij}) = \beta_0 + \beta_1 \texttt{treat}_{ij} + \beta_2 \texttt{lbas}_{ij} + \beta_3 \texttt{lbas\_trt}_{ij} + \beta_4 \texttt{lage}_{ij} + \beta_5 \texttt{visit}_{ij} + u_j + v_j \texttt{visit}_{ij}$$

where (u_j, v_j) are bivariate normal with zero mean and variance–covariance matrix:

$$\mathbf{\Sigma} = \text{Var}\begin{bmatrix} u_j \\ v_j \end{bmatrix} = \begin{bmatrix} \sigma_u^2 & \sigma_{uv} \\ \sigma_{uv} & \sigma_v^2 \end{bmatrix}$$

```
. xtmepoisson seizures treat lbas lbas_trt lage visit || subject: visit,
> cov(unstructured) intpoints(9)
```

(*output omitted*)

```
Mixed-effects Poisson regression                Number of obs      =       236
Group variable: subject                         Number of groups   =        59

                                                Obs per group: min =         4
                                                               avg =       4.0
                                                               max =         4

Integration points =   9                        Wald chi2(5)       =    115.56
Log likelihood = -655.68103                     Prob > chi2        =    0.0000
```

seizures	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
treat	-.9286588	.4021639	-2.31	0.021	-1.716886	-.140432
lbas	.8849767	.1312519	6.74	0.000	.6277277	1.142226
lbas_trt	.3379757	.2044443	1.65	0.098	-.0627277	.7386791
lage	.4767192	.353622	1.35	0.178	-.2163673	1.169806
visit	-.2664098	.1647096	-1.62	0.106	-.5892347	.0564151
_cons	2.099555	.220371	9.53	0.000	1.667635	2.531474

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
subject: Unstructured				
sd(visit)	.7290273	.1573227	.477591	1.112837
sd(_cons)	.5014906	.0586145	.3988172	.6305967
corr(visit,_cons)	.0078543	.2426514	-.43639	.4490197

```
LR test vs. Poisson regression:      chi2(3) =   324.54   Prob > chi2 = 0.0000

Note: LR test is conservative and provided only for reference.
```

In the above, we specified the `cov(unstructured)` option to allow correlation between u_j and v_j, although on the basis of the above output it probably was not necessary—the default `Independent` structure would have sufficed. In the interest of getting more accurate estimates, we also increased the number of quadrature points to nine, although the estimates do not change much when compared with estimates based on the default seven quadrature points.

The essence of the above-fitted model is that, after adjusting for other covariates, the log trend in seizures is modeled as a random subject-specific line, with intercept distributed as $N(\beta_0, \sigma_u^2)$ and slope distributed as $N(\beta_5, \sigma_v^2)$. From the above output, $\widehat{\beta}_0 = 2.100$, $\widehat{\sigma}_u = 0.501$, $\widehat{\beta}_5 = -0.266$, and $\widehat{\sigma}_v = 0.729$.

You can predict the random effects u_j and v_j by using `predict` after `xtmepoisson`; see [XT] **xtmepoisson postestimation**. Better still, you can obtain a predicted number of seizures that takes these random effects into account.

`xtmepoisson` also offers a myriad of display options. Among the most useful are `variance` for displaying estimated variance components as variance and covariances, and `irr` for displaying fixed effects as incidence-rate ratios.

```
. xtmepoisson, variance irr
Mixed-effects Poisson regression                Number of obs      =       236
Group variable: subject                         Number of groups   =        59
                                                Obs per group: min =         4
                                                               avg =       4.0
                                                               max =         4
Integration points =   9                        Wald chi2(5)       =    115.56
Log likelihood = -655.68103                     Prob > chi2        =    0.0000
```

seizures	IRR	Std. Err.	z	P>|z|	[95% Conf.	Interval]
treat	.3950833	.1588884	-2.31	0.021	.1796246	.8689834
lbas	2.422928	.3180141	6.74	0.000	1.873349	3.133736
lbas_trt	1.402106	.2866529	1.65	0.098	.9391988	2.09317
lage	1.610781	.5696077	1.35	0.178	.8054394	3.221366
visit	.7661251	.1261882	-1.62	0.106	.5547517	1.058037
_cons	8.162533	1.798787	9.53	0.000	5.29962	12.57203

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
subject: Unstructured				
var(visit)	.5314808	.2293851	.2280931	1.238406
var(_cons)	.2514928	.0587892	.1590552	.3976522
cov(visit,_cons)	.0028715	.0887018	-.1709808	.1767238

```
LR test vs. Poisson regression:      chi2(3) =   324.54   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

◁

A three-level model

`xtmepoisson` can also fit higher-level models with multiple levels of nested random effects.

▷ Example 3

Rabe-Hesketh and Skrondal (2008, exercise 9.8) describe data from the *Atlas of Cancer Mortality in the European Economic Community* (EEC) (Smans, Mair, and Boyle 1993). The data were analyzed in Langford, Bentham, and McDonald (1998) and record the number of deaths among males because of malignant melanoma during 1971–1980.

```
. use http://www.stata-press.com/data/r12/melanoma
(Skin cancer (melanoma) data)
. describe
Contains data from melanoma.dta
  obs:           354                          Skin cancer (melanoma) data
 vars:             6                          30 May 2011 17:10
 size:         4,956                          (_dta has notes)
```

variable name	storage type	display format	value label	variable label
nation	byte	%11.0g	n	Nation ID
region	byte	%9.0g		Region ID: EEC level-I areas
county	int	%9.0g		County ID: EEC level-II/level-III areas
deaths	int	%9.0g		No. deaths during 1971-1980
expected	float	%9.0g		No. expected deaths
uv	float	%9.0g		UV dose, mean-centered

```
Sorted by:
```

Nine European nations (variable `nation`) are represented, and data were collected over geographical regions defined by EEC statistical services as level I areas (variable `region`), with deaths being recorded for each of 354 counties, which are level II or level III EEC-defined areas (variable `county`, which identifies the observations). Counties are nested within regions, and regions are nested within nations.

Variable `deaths` records the number of deaths for each county, and `expected` records the expected number of deaths (the exposure) on the basis of crude rates for the combined countries. Finally, variable `uv` is a measure of exposure to ultraviolet (UV) radiation.

In modeling the number of deaths, one possibility is to include dummy variables for the nine nations as fixed effects. Another is to treat these as random effects and fit the three-level random-intercept Poisson model,

$$\log(\mu_{ijk}) = \log(\texttt{expected}_{ijk}) + \beta_0 + \beta_1 \texttt{uv}_{ijk} + u_k + v_{jk}$$

for nation k, region j, and county i. The model includes an exposure term for expected deaths.

```
. xtmepoisson deaths uv, exposure(expected) || nation: || region:
  (output omitted)
Mixed-effects Poisson regression                Number of obs      =       354
```

Group Variable	No. of Groups	Observations per Group Minimum	Average	Maximum	Integration Points
nation	9	3	39.3	95	7
region	78	1	4.5	13	7

```
                                                Wald chi2(1)       =      6.12
Log likelihood =  -1097.714                     Prob > chi2        =    0.0134
```

deaths	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
uv	-.0281991	.0114027	-2.47	0.013	-.050548	-.0058503
_cons	-.0639477	.1335246	-0.48	0.632	-.3256511	.1977558
ln(expected)	1	(exposure)				

Random-effects Parameters		Estimate	Std. Err.	[95% Conf.	Interval]
nation: Identity	sd(_cons)	.3701812	.0976276	.2207637	.6207278
region: Identity	sd(_cons)	.2199668	.0248379	.1762959	.2744554

```
LR test vs. Poisson regression:      chi2(2) =  1252.12   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

By including an exposure variable that is an expected rate, we are in effect specifying a linear model for the log of the standardized mortality ratio (SMR), the ratio of observed deaths to expected deaths that is based on a reference population. Here the reference population is all nine nations.

We now add a random-intercept for counties nested within regions, making this a four-level model. Because counties also identify the observations, the corresponding variance component can be interpreted as a measure of overdispersion, variability above and beyond that allowed by standard Poisson; see [R] **nbreg**.

```
. xtmepoisson deaths uv, exposure(expected) || nation: || region: || county:,
> laplace
  (output omitted)
Mixed-effects Poisson regression                Number of obs      =       354
```

Group Variable	No. of Groups	Observations per Group Minimum	Average	Maximum	Integration Points
nation	9	3	39.3	95	1
region	78	1	4.5	13	1
county	354	1	1.0	1	1

```
                                                Wald chi2(1)       =      8.63
Log likelihood = -1086.7309                     Prob > chi2        =    0.0033
```

deaths	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
uv	-.0334682	.0113919	-2.94	0.003	-.0557959	-.0111404
_cons	-.086411	.1298712	-0.67	0.506	-.3409539	.1681319
ln(expected)	1	(exposure)				

Random-effects Parameters		Estimate	Std. Err.	[95% Conf.	Interval]
nation: Identity	sd(_cons)	.3588058	.0948823	.2136822	.6024909
region: Identity	sd(_cons)	.2014854	.026057	.1563732	.2596119
county: Identity	sd(_cons)	.1208414	.0210052	.085952	.169893

```
LR test vs. Poisson regression:      chi2(3) =  1274.08   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
Note: log-likelihood calculations are based on the Laplacian approximation.
```

In the above, we used a Laplacian approximation, which is not only faster but also produces estimates that closely agree with those obtained with the default seven quadrature points.

See *Computation time and the Laplacian approximation* in [XT] **xtmelogit** for a discussion comparing Laplacian approximation with adaptive quadrature.

◁

Saved results

xtmepoisson saves the following in e():

Scalars

e(N)	number of observations
e(k)	number of parameters
e(k_f)	number of FE parameters
e(k_r)	number of RE parameters
e(k_rs)	number of standard deviations
e(k_rc)	number of correlations
e(df_m)	model degrees of freedom
e(ll)	log likelihood
e(chi2)	χ^2
e(p)	p-value for χ^2
e(ll_c)	log likelihood, comparison model
e(chi2_c)	χ^2, comparison model
e(df_c)	degrees of freedom, comparison model
e(p_c)	p-value, comparison model
e(rank)	rank of e(V)
e(reparm_rc)	return code, final reparameterization
e(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros

e(cmd)	xtmepoisson
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(ivars)	grouping variables
e(exposurevar)	exposure variable
e(model)	Poisson
e(title)	title in estimation output
e(offset)	linear offset variable
e(redim)	random-effects dimensions
e(vartypes)	variance-structure types
e(revars)	random-effects covariates
e(n_quad)	number of integration points
e(laplace)	laplace, if Laplace approximation
e(chi2type)	Wald, type of model χ^2
e(vce)	bootstrap or jackknife if defined
e(vcetype)	title used to label Std. Err.
e(method)	ML
e(opt)	type of optimization
e(ml_method)	type of ml method
e(technique)	maximization technique
e(datasignature)	the checksum
e(datasignaturevars)	variables used in checksum
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(Cns)	constraints matrix
e(N_g)	group counts
e(g_min)	group-size minimums
e(g_avg)	group-size averages
e(g_max)	group-size maximums
e(V)	variance–covariance matrix of the estimator

Functions

e(sample)	marks estimation sample

Methods and formulas

`xtmepoisson` is implemented as an ado-file.

In a two-level Poisson model, for cluster j, $j = 1, \ldots, M$, the conditional distribution of $\mathbf{y}_j = (y_{j1}, \ldots, y_{jn_j})'$, given a set of cluster-level random effects $\mathbf{u}_j$, is

$$\begin{aligned} f(\mathbf{y}_j|\mathbf{u}_j) &= \prod_{i=1}^{n_j} \left[\left\{\exp\left(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j\right)\right\}^{y_{ij}} \exp\left\{-\exp\left(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j\right)\right\} / y_{ij}!\right] \\ &= \exp\left[\sum_{i=1}^{n_j} \left\{y_{ij}\left(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j\right) - \exp\left(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j\right) - \log(y_{ij}!)\right\}\right] \end{aligned}$$

Defining $c\left(\mathbf{y}_j\right) = \sum_{i=1}^{n_j} \log(y_{ij}!)$, where $c(\mathbf{y}_j)$ does not depend on the model parameters, we can express the above compactly in matrix notation,

$$f(\mathbf{y}_j|\mathbf{u}_j) = \exp\left\{\mathbf{y}_j'\left(\mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j\right) - \mathbf{1}'\exp\left(\mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j\right) - c\left(\mathbf{y}_j\right)\right\}$$

where $\mathbf{X}_j$ is formed by stacking the row vectors $\mathbf{x}_{ij}$, $\mathbf{Z}_j$ is formed by stacking the row vectors $\mathbf{z}_{ij}$, and we extend the definition of $\exp()$ to be a vector function where necessary.

Because the prior distribution of $\mathbf{u}_j$ is multivariate normal with mean $\mathbf{0}$ and $q \times q$ variance matrix $\boldsymbol{\Sigma}$, the likelihood contribution for the j cluster is obtained by integrating $\mathbf{u}_j$ out the joint density $f(\mathbf{y}_j, \mathbf{u}_j)$,

$$\begin{aligned} \mathcal{L}_j(\boldsymbol{\beta}, \boldsymbol{\Sigma}) &= (2\pi)^{-q/2}\left|\boldsymbol{\Sigma}\right|^{-1/2} \int f(\mathbf{y}_j|\mathbf{u}_j) \exp\left(-\mathbf{u}_j'\boldsymbol{\Sigma}^{-1}\mathbf{u}_j/2\right) d\mathbf{u}_j \\ &= \exp\left\{-c\left(\mathbf{y}_j\right)\right\}(2\pi)^{-q/2}\left|\boldsymbol{\Sigma}\right|^{-1/2} \int \exp\left\{g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right)\right\} d\mathbf{u}_j \end{aligned} \tag{2}$$

where

$$g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right) = \mathbf{y}_j'\left(\mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j\right) - \mathbf{1}'\exp\left(\mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j\right) - \mathbf{u}_j'\boldsymbol{\Sigma}^{-1}\mathbf{u}_j/2$$

and for convenience, in the arguments of $g()$ we suppress the dependence on the observable data $(\mathbf{y}_j, \mathbf{X}_j, \mathbf{Z}_j)$.

The integration in (2) has no closed form and thus must be approximated. The Laplacian approximation (Tierney and Kadane 1986; Pinheiro and Bates 1995) is based on a second-order Taylor expansion of $g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right)$ about the value of $\mathbf{u}_j$ that maximizes it. Taking first and second derivatives, we obtain

$$\begin{aligned} g'\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right) &= \frac{\partial g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right)}{\partial \mathbf{u}_j} = \mathbf{Z}_j'\left\{\mathbf{y}_j - \mathbf{m}(\boldsymbol{\beta}, \mathbf{u}_j)\right\} - \boldsymbol{\Sigma}^{-1}\mathbf{u}_j \\ g''\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right) &= \frac{\partial^2 g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right)}{\partial \mathbf{u}_j \partial \mathbf{u}_j'} = -\left\{\mathbf{Z}_j'\mathbf{V}(\boldsymbol{\beta}, \mathbf{u}_j)\mathbf{Z}_j + \boldsymbol{\Sigma}^{-1}\right\} \end{aligned}$$

where $\mathbf{m}(\boldsymbol{\beta}, \mathbf{u}_j)$ is the vector function with ith element equal to the conditional mean of y_{ij} given $\mathbf{u}_j$, that is, $\exp(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j)$. $\mathbf{V}(\boldsymbol{\beta}, \mathbf{u}_j)$ is the diagonal matrix whose diagonal entries v_{ij} are the conditional variances of y_{ij} given $\mathbf{u}_j$, namely,

$$v_{ij} = \exp\left(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j\right)$$

because equality of mean and variance is a characteristic of the Poisson distribution.

The maximizer of $g(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j)$ is $\widehat{\mathbf{u}}_j$ such that $g'(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j) = \mathbf{0}$. The integrand in (2) is proportional to the posterior density $f(\mathbf{u}_j|\mathbf{y}_j)$, so $\widehat{\mathbf{u}}_j$ also represents the posterior mode, a plausible estimator of $\mathbf{u}_j$ in its own right.

Given the above derivatives, the second-order Taylor approximation then takes the form

$$g(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j) \approx g(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j) + \frac{1}{2}(\mathbf{u}_j - \widehat{\mathbf{u}}_j)' g''(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j)(\mathbf{u}_j - \widehat{\mathbf{u}}_j) \tag{3}$$

The first-derivative term vanishes because $g'(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j) = \mathbf{0}$. Therefore,

$$\begin{aligned}\int \exp\{g(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j)\} d\mathbf{u}_j &\approx \exp\{g(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j)\} \\ &\quad \times \int \exp\left[-\frac{1}{2}(\mathbf{u}_j - \widehat{\mathbf{u}}_j)'\{-g''(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j)\}(\mathbf{u}_j - \widehat{\mathbf{u}}_j)\right] d\mathbf{u}_j \\ &= \exp\{g(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j)\}(2\pi)^{q/2} |-g''(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j)|^{-1/2}\end{aligned} \tag{4}$$

because the latter integrand can be recognized as the "kernel" of a multivariate normal density.

Combining the above with (2) (and taking logs) gives the Laplacian log-likelihood contribution of the jth cluster,

$$L_j^{\text{Lap}}(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = -\frac{1}{2}\log|\boldsymbol{\Sigma}| - \log|\mathbf{R}_j| + g(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j) - c(\mathbf{y}_j)$$

where $\mathbf{R}_j$ is an upper-triangular matrix such that $-g''(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j) = \mathbf{R}_j\mathbf{R}_j'$. Pinheiro and Chao (2006) show that $\widehat{\mathbf{u}}_j$ and $\mathbf{R}_j$ can be efficiently computed as the iterative solution to a least-squares problem by using matrix decomposition methods similar to those used in fitting LME models (Bates and Pinheiro 1998; Pinheiro and Bates 2000; [XT] **xtmixed**).

The fidelity of the Laplacian approximation is determined wholly by the accuracy of the approximation in (3). An alternative that does not depend so heavily on this approximation is integration via adaptive Gaussian quadrature (AGQ; Naylor and Smith 1982; Liu and Pierce 1994).

The application of AGQ to this particular problem is from Pinheiro and Bates (1995). When we reexamine the integral in question, a transformation of integration variables yields

$$\begin{aligned}\int \exp\{g(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j)\} d\mathbf{u}_j &= |\mathbf{R}_j|^{-1} \int \exp\left\{g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j + \mathbf{R}_j^{-1}\mathbf{t}\right)\right\} d\mathbf{t} \\ &= (2\pi)^{q/2} |\mathbf{R}_j|^{-1} \int \exp\left\{g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j + \mathbf{R}_j^{-1}\mathbf{t}\right) + \mathbf{t}'\mathbf{t}/2\right\} \phi(\mathbf{t}) d\mathbf{t}\end{aligned} \tag{5}$$

where $\phi()$ is the standard multivariate normal density. Because the integrand is now expressed as some function multiplied by a normal density, it can be estimated by applying the rules of standard Gauss–Hermite quadrature. For a predetermined number of quadrature points N_Q, define $a_k = \sqrt{2}a_k^*$ and $w_k = w_k^*/\sqrt{\pi}$, for $k = 1, \ldots, N_Q$, where (a_k^*, w_k^*) are a set of abscissas and weights for Gauss–Hermite quadrature approximations of $\int \exp(-x^2) f(x) dx$, as obtained from Abramowitz and Stegun (1972, 924).

Define $\mathbf{a_k} = (a_{k_1}, a_{k_2}, \ldots, a_{k_q})'$; that is, $\mathbf{a_k}$ is a vector that spans the N_Q abscissas over the dimension q of the random effects. Applying quadrature rules to (5) yields the AGQ approximation,

$$\begin{aligned}
&\int \exp\left\{g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right)\right\} d\mathbf{u}_j \\
&\quad \approx (2\pi)^{q/2} \left|\mathbf{R}_j\right|^{-1} \sum_{k_1=1}^{N_Q} \cdots \sum_{k_q=1}^{N_Q} \left[\exp\left\{g\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j + \mathbf{R}_j^{-1}\mathbf{a}_\mathbf{k}\right) + \mathbf{a}_\mathbf{k}'\mathbf{a}_\mathbf{k}/2\right\} \prod_{p=1}^{q} w_{k_p}\right] \\
&\quad \equiv (2\pi)^{q/2} \widehat{G}_j(\boldsymbol{\beta}, \boldsymbol{\Sigma})
\end{aligned}$$

resulting in the AGQ log-likelihood contribution of the jth cluster,

$$L_j^{\mathrm{AGQ}}(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = -\frac{1}{2}\log|\boldsymbol{\Sigma}| + \log\left\{\widehat{G}_j(\boldsymbol{\beta}, \boldsymbol{\Sigma})\right\} - c(\mathbf{y}_j)$$

The "adaptive" part of adaptive Gaussian quadrature lies in the translation and rescaling of the integration variables in (5) by using $\widehat{\mathbf{u}}_j$ and $\mathbf{R}_j^{-1}$ respectively. This transformation of quadrature abscissas (centered at zero in standard form) is chosen to better capture the features of the integrand, through which (4) can be seen to resemble a multivariate normal distribution with mean $\widehat{\mathbf{u}}_j$ and variance $\mathbf{R}_j^{-1}\mathbf{R}_j^{-T}$. AGQ is therefore not as dependent as the Laplace method upon the approximation in (3). In AGQ, (3) serves merely to redirect the quadrature abscissas, with the AGQ approximation improving as the number of quadrature points, N_Q, increases. In fact, Pinheiro and Bates (1995) point out that AGQ with only one quadrature point ($a = 0$ and $w = 1$) reduces to the Laplacian approximation.

The log likelihood for the entire dataset is then simply the sum of the contributions of the M individual clusters, namely, $L(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = \sum_{j=1}^{M} L_j^{\mathrm{Lap}}(\boldsymbol{\beta}, \boldsymbol{\Sigma})$ for Laplace and $L(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = \sum_{j=1}^{M} L_j^{\mathrm{AGQ}}(\boldsymbol{\beta}, \boldsymbol{\Sigma})$ for adaptive Gaussian quadrature.

Maximization of $L(\boldsymbol{\beta}, \boldsymbol{\Sigma})$ is performed with respect to $(\boldsymbol{\beta}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a vector comprising the unique elements of the matrix square root of $\boldsymbol{\Sigma}$. This is done to ensure that $\boldsymbol{\Sigma}$ is always positive semidefinite. If the `matlog` option is specified, then $\boldsymbol{\theta}$ instead consists of the unique elements of the matrix logarithm of $\boldsymbol{\Sigma}$. For well-conditioned problems both methods produce equivalent results, yet our experience deems the former as more numerically stable near the boundary of the parameter space.

Once maximization is achieved, parameter estimates are mapped from $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\theta}})$ to $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\gamma}})$, where $\widehat{\boldsymbol{\gamma}}$ is a vector containing the unique (estimated) elements of $\boldsymbol{\Sigma}$, expressed as logarithms of standard deviations for the diagonal elements and hyperbolic arctangents of the correlations for off-diagonal elements. This last step is necessary to (a) obtain a parameterization under which parameter estimates can be displayed and interpreted individually, rather than as elements of a matrix square root (or logarithm), and (b) parameterize these elements such that their ranges each encompass the entire real line.

Parameter estimates are stored in `e(b)` as $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\gamma}})$, with the corresponding variance–covariance matrix stored in `e(V)`. Parameter estimates can be displayed in this metric by specifying the `estmetric` option. However, in `xtmepoisson` output, variance components are most often displayed either as variances and covariances (option `variance`) or as standard deviations and correlations (the default).

The approach outlined above can be extended from two-level models to three- and higher-level models; see Pinheiro and Chao (2006) for details.

Acknowledgments

We are indebted to Sophia Rabe-Hesketh, University of California, Berkeley; Anders Skrondal, University of Oslo and Norwegian Institute of Public Health; and Andrew Pickles, University of Manchester, for their extensive body of work in Stata, both previous and ongoing, in this area.

References

Abramowitz, M., and I. A. Stegun, ed. 1972. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. 10th ed. Washington, DC: National Bureau of Standards.

Andrews, M., T. Schank, and R. Upward. 2006. Practical fixed-effects estimation methods for the three-way error-components model. *Stata Journal* 6: 461–481.

Bates, D. M., and J. C. Pinheiro. 1998. Computational methods for multilevel modelling. In *Technical Memorandum BL0112140-980226-01TM*. Murray Hill, NJ: Bell Labs, Lucent Technologies. http://stat.bell-labs.com/NLME/CompMulti.pdf.

Breslow, N. E., and D. G. Clayton. 1993. Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association* 88: 9–25.

Gutierrez, R. G., S. 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.

Joe, H. 2008. Accuracy of Laplace approximation for discrete response mixed models. *Computational Statistics & Data Analysis* 52: 5066–5074.

Laird, N. M., and J. H. Ware. 1982. Random-effects models for longitudinal data. *Biometrics* 38: 963–974.

Langford, I. H., G. Bentham, and A. McDonald. 1998. Multi-level modelling of geographically aggregated health data: A case study on malignant melanoma mortality and UV exposure in the European community. *Statistics in Medicine* 17: 41–57.

Leyland, A. H., and H. Goldstein, ed. 2001. *Multilevel Modelling of Health Statistics*. New York: Wiley.

Lin, X., and N. E. Breslow. 1996. Bias correction in generalized linear mixed models with multiple components of dispersion. *Journal of the American Statistical Association* 91: 1007–1016.

Liu, Q., and D. A. Pierce. 1994. A note on Gauss–Hermite quadrature. *Biometrika* 81: 624–629.

Marchenko, Y. V. 2006. Estimating variance components in Stata. *Stata Journal* 6: 1–21.

McCulloch, C. E., S. R. Searle, and J. M. Neuhaus. 2008. *Generalized, Linear, and Mixed Models*. 2nd ed. Hoboken, NJ: Wiley.

McLachlan, G. J., and K. E. Basford. 1988. *Mixture Models*. New York: Dekker.

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, Series C* 31: 214–225.

Pinheiro, J. C., and D. M. Bates. 1995. Approximations to the log-likelihood function in the nonlinear mixed-effects model. *Journal of Computational and Graphical Statistics* 4: 12–35.

——. 2000. *Mixed-Effects Models in S and S-PLUS*. New York: Springer.

Pinheiro, J. C., and E. C. Chao. 2006. Efficient Laplacian and adaptive Gaussian quadrature algorithms for multilevel generalized linear mixed models. *Journal of Computational and Graphical Statistics* 15: 58–81.

Rabe-Hesketh, S., and A. Skrondal. 2008. *Multilevel and Longitudinal Modeling Using Stata*. 2nd ed. College Station, TX: Stata Press.

Rabe-Hesketh, S., A. Skrondal, and A. Pickles. 2005. Maximum likelihood estimation of limited and discrete dependent variable models with nested random effects. *Journal of Econometrics* 128: 301–323.

Raudenbush, S. W., and A. S. Bryk. 2002. *Hierarchical Linear Models: Applications and Data Analysis Methods*. 2nd ed. Thousand Oaks, CA: Sage.

Self, S. G., and K.-Y. Liang. 1987. Asymptotic properties of maximum likelihood estimators and likelihood ratio tests under nonstandard conditions. *Journal of the American Statistical Association* 82: 605–610.

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

Smans, M., C. S. Mair, and P. Boyle. 1993. *Atlas of Cancer Mortality in the European Economic Community*. Lyon, France: IARC Scientific Publications.

Thall, P. F., and S. C. Vail. 1990. Some covariance models for longitudinal count data with overdispersion. *Biometrics* 46: 657–671.

Tierney, L., and J. B. Kadane. 1986. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association* 81: 82–86.

Also see

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

[XT] **xtmelogit** — Multilevel mixed-effects logistic regression

[XT] **xtmixed** — Multilevel mixed-effects linear regression

[XT] **xtpoisson** — Fixed-effects, random-effects, and population-averaged Poisson models

[XT] **xtrc** — Random-coefficients model

[XT] **xtreg** — Fixed-, between-, and random-effects and population-averaged linear models

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

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

[U] **20 Estimation and postestimation commands**

Title

xtmepoisson postestimation — Postestimation tools for xtmepoisson

Description

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

Command	Description
`estat group`	summarize the composition of the nested groups
`estat recovariance`	display the estimated random-effects covariance matrix (or matrices)

For information about these commands, see below.

The following standard postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat`	AIC, BIC, VCE, and estimation sample summary
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`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 coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

See the corresponding entries in the *Base Reference Manual* for details.

Special-interest postestimation commands

`estat group` reports number of groups and minimum, average, and maximum group sizes for each level of the model. Model levels are identified by the corresponding group variable in the data. Because groups are treated as nested, the information in this summary may differ from what you would get if you `tabulate`d each group variable individually.

`estat recovariance` displays the estimated variance–covariance matrix of the random effects for each level in the model. Random effects can be either random intercepts, in which case the corresponding rows and columns of the matrix are labeled as `_cons`, or random coefficients, in which case the label is the name of the associated variable in the data.

Syntax for predict

Syntax for obtaining estimated random effects or their standard errors

```
predict [type] {stub*|newvarlist} [if] [in], {reffects|reses}
        [level(levelvar)]
```

Syntax for obtaining other predictions

```
predict [type] newvar [if] [in] [, statistic fixedonly nooffset]
```

statistic	Description
Main	
mu	the predicted mean; the default
xb	linear prediction for the *fixed* portion of the model only
stdp	standard error of the fixed-portion linear prediction
pearson	Pearson residuals
deviance	deviance residuals
anscombe	Anscombe residuals

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

Menu

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

`reffects` calculates posterior modal estimates of the random effects. By default, estimates for all random effects in the model are calculated. However, if the `level(`*levelvar*`)` option is specified, then estimates for only level *levelvar* in the model are calculated. For example, if `classes` are nested within `schools`, then typing

```
. predict b*, reffects level(school)
```

would yield random-effects estimates at the school level. You must specify q new variables, where q is the number of random-effects terms in the model (or level). However, it is much easier to just specify *stub*`*` and let Stata name the variables *stub*`1`, *stub*`2`, ..., *stub*q for you.

`reses` calculates standard errors for the random-effects estimates obtained by using the `reffects` option. By default, standard errors for all random effects in the model are calculated. However, if the `level(`*levelvar*`)` option is specified, then standard errors for only level *levelvar* in the model are calculated. For example, if `classes` are nested within `schools`, then typing

```
. predict se*, reses level(school)
```

would yield standard errors at the school level. You must specify q new variables, where q is the number of random-effects terms in the model (or level). However, it is much easier to just specify *stub*`*` and let Stata name the variables *stub*`1`, *stub*`2`, ..., *stubq* for you.

The `reffects` and `reses` options often generate multiple new variables at once. When this occurs, the random effects (or standard errors) contained in the generated variables correspond to the order in which the variance components are listed in the output of `xtmepoisson`. Still, examining the variable labels of the generated variables (using the `describe` command, for instance) can be useful in deciphering which variables correspond to which terms in the model.

`level(`*levelvar*`)` specifies the level in the model at which predictions for random effects and their standard errors are to be obtained. *levelvar* is the name of the model level and is either the name of the variable describing the grouping at that level or `_all`, a special designation for a group comprising all the estimation data.

`mu`, the default, calculates the predicted mean, that is, the predicted count. By default, this is based on a linear predictor that includes *both* the fixed effects and the random effects, and the predicted mean is conditional on the values of the random effects. Use the `fixedonly` option (see below) if you want predictions that include only the fixed portion of the model, that is, if you want random effects set to zero.

`xb` calculates the linear prediction $\mathbf{x}\boldsymbol{\beta}$ based on 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 fixed-effects linear predictor $\mathbf{x}\boldsymbol{\beta}$.

`pearson` calculates Pearson residuals. Pearson residuals large in absolute value may indicate a lack of fit. By default, residuals include both the fixed portion and the random portion of the model. The `fixedonly` option modifies the calculation to include the fixed portion only.

`deviance` calculates deviance residuals. Deviance residuals are recommended by McCullagh and Nelder (1989) as having the best properties for examining the goodness of fit of a GLM. They are approximately normally distributed if the model is correctly specified. They may be plotted against the fitted values or against a covariate to inspect the model's fit. By default, residuals include both the fixed portion and the random portion of the model. The `fixedonly` option modifies the calculation to include the fixed portion only.

`anscombe` calculates Anscombe residuals, residuals that are designed to closely follow a normal distribution. By default, residuals include both the fixed portion and the random portion of the model. The `fixedonly` option modifies the calculation to include the fixed portion only.

`fixedonly` modifies predictions to include only the fixed portion of the model, equivalent to setting all random effects equal to zero; see the `mu` option.

`nooffset` is relevant only if you specified `offset(`$varname_o$`)` or `exposure(`$varname_e$`)` for `xtmepoisson`. It modifies the calculations made by `predict` so that they ignore the offset variable; the linear prediction is treated as $\mathbf{X}\beta + \mathbf{Zu}$ rather than $\mathbf{X}\beta + \mathbf{Zu} + \text{offset}$, or $\mathbf{X}\beta + \mathbf{Zu} + \ln(\text{exposure})$, whichever is relevant.

Syntax for estat group

```
estat group
```

Menu

Statistics > Postestimation > Reports and statistics

Syntax for estat recovariance

```
estat recovariance [ , level(levelvar) correlation matlist_options ]
```

Menu

Statistics > Postestimation > Reports and statistics

Options for estat recovariance

`level(`*levelvar*`)` specifies the level in the model for which the random-effects covariance matrix is to be displayed and returned in `r(cov)`. By default, the covariance matrices for all levels in the model are displayed. *levelvar* is the name of the model level and is either the name of variable describing the grouping at that level or `_all`, a special designation for a group comprising all the estimation data.

`correlation` displays the covariance matrix as a correlation matrix and returns the correlation matrix in `r(corr)`.

matlist_options are style and formatting options that control how the matrix (or matrices) are displayed; see [P] **matlist** for a list of what is available.

Remarks

Various predictions, statistics, and diagnostic measures are available after fitting a Poisson mixed-effects model with `xtmepoisson`. For the most part, calculation centers around obtaining estimates of the subject/group-specific random effects. Random effects are not estimated when the model is fit but instead need to be predicted after estimation.

▷ Example 1

In example 2 of [XT] **xtmepoisson**, we modeled the number of observed epileptic seizures as a function of treatment with the drug progabide and other covariates

$$\log(\mu_{ij}) = \beta_0 + \beta_1 \texttt{treat}_{ij} + \beta_2 \texttt{lbas}_{ij} + \beta_3 \texttt{lbas\_trt}_{ij} + \beta_4 \texttt{lage}_{ij} + \beta_5 \texttt{visit}_{ij} + u_j + v_j \texttt{visit}_{ij}$$

where (u_j, v_j) are bivariate normal with zero mean and variance–covariance matrix

$$\mathbf{\Sigma} = \text{Var}\begin{bmatrix} u_j \\ v_j \end{bmatrix} = \begin{bmatrix} \sigma_u^2 & \sigma_{uv} \\ \sigma_{uv} & \sigma_v^2 \end{bmatrix}$$

```
. use http://www.stata-press.com/data/r12/epilepsy
(Epilepsy data; progabide drug treatment)
. xtmepoisson seizures treat lbas lbas_trt lage visit || subject: visit,
> cov(unstructured) intpoints(9)
```

(*output omitted*)

```
Mixed-effects Poisson regression                Number of obs      =       236
Group variable: subject                         Number of groups   =        59
                                                Obs per group: min =         4
                                                               avg =       4.0
                                                               max =         4
Integration points =   9                        Wald chi2(5)       =    115.56
Log likelihood = -655.68103                     Prob > chi2        =    0.0000
```

seizures	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
treat	-.9286588	.4021643	-2.31	0.021	-1.716886	-.1404312
lbas	.8849767	.131252	6.74	0.000	.6277275	1.142226
lbas_trt	.3379757	.2044445	1.65	0.098	-.0627281	.7386795
lage	.4767192	.353622	1.35	0.178	-.2163673	1.169806
visit	-.2664098	.1647096	-1.62	0.106	-.5892347	.0564151
_cons	2.099555	.2203712	9.53	0.000	1.667635	2.531474

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
subject: Unstructured				
sd(visit)	.7290273	.1573227	.4775909	1.112837
sd(_cons)	.5014906	.0586145	.3988172	.6305967
corr(visit,_cons)	.0078542	.2426514	-.43639	.4490197

```
LR test vs. Poisson regression:      chi2(3) =   324.54   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

The purpose of this model was to allow subject-specific linear log trends over each subject's four doctor visits, after adjusting for the other covariates. The intercepts of these lines are distributed $N(\beta_0, \sigma_u^2)$, and the slopes $N(\beta_5, \sigma_v^2)$, based on the fixed effects and assumed distribution of the random effects.

We can use `predict` to obtain estimates of the random effects u_j and v_j and combine these with our estimates of β_0 and β_5 to obtain the intercepts and slopes of the linear log trends.

```
. predict re_visit re_cons, reffects
. generate b1 = _b[visit] + re_visit
. generate b0 = _b[_cons] + re_cons
. by subject, sort: generate tolist = _n==1
```

```
. list subject treat b1 b0 if tolist & (subject <=5 | subject >=55)
```

	subject	treat	b1	b0
1.	1	0	-.4284563	2.164691
5.	2	0	-.2727145	2.179111
9.	3	0	.0026486	2.450811
13.	4	0	-.3194157	2.268827
17.	5	0	.6063656	2.123723
217.	55	1	-.2304782	2.311493
221.	56	1	.2904741	3.211369
225.	57	1	-.4831492	1.457485
229.	58	1	-.252236	1.168154
233.	59	1	-.1266651	2.204869

We list these slopes (`b1`) and intercepts (`b0`) for five control subjects and five subjects on the treatment.

```
. count if tolist & treat
  31
. count if tolist & treat  & b1 < 0
  25
. count if tolist & !treat
  28
. count if tolist & !treat & b1 < 0
  20
```

We also find that 25 of the 31 subjects taking progabide were estimated to have a downward trend in seizures over their four doctor visits, compared with 20 of the 28 control subjects.

We also obtain predictions for number of seizures, and unless we specify the `fixedonly` option, these predictions will incorporate the estimated subject-specific random effects.

```
. predict n
(option mu assumed; predicted means)
. list subject treat visit seizures n if subject <= 2 | subject >= 58, sep(0)
```

	subject	treat	visit	seizures	n
1.	1	0	-.3	5	3.887582
2.	1	0	-.1	3	3.568324
3.	1	0	.1	3	3.275285
4.	1	0	.3	3	3.00631
5.	2	0	-.3	3	3.705628
6.	2	0	-.1	5	3.508926
7.	2	0	.1	3	3.322664
8.	2	0	.3	3	3.14629
229.	58	1	-.3	0	.9972093
230.	58	1	-.1	0	.9481507
231.	58	1	.1	0	.9015056
232.	58	1	.3	0	.8571553
233.	59	1	-.3	1	2.487858
234.	59	1	-.1	4	2.425625
235.	59	1	.1	3	2.364948
236.	59	1	.3	2	2.305789

❑ Technical note

Out-of-sample predictions are permitted after `xtmepoisson`, but if these predictions involve estimated random effects, the integrity of the estimation data must be preserved. If the estimation data have changed since the model was fit, `predict` will be unable to obtain predicted random effects that are appropriate for the fitted model and will give an error. Thus, to obtain out-of-sample predictions that contain random-effects terms, be sure that the data for these predictions are in observations that augment the estimation data.

❑

Saved results

`estat recovariance` saves the last-displayed random-effects covariance matrix in `r(cov)` or in `r(corr)` if it is displayed as a correlation matrix.

Methods and formulas

Continuing the discussion in *Methods and formulas* of [XT] **xtmepoisson**, and using the definitions and formulas defined there, we begin by considering the "prediction" of the random effects $\mathbf{u}_j$ for the jth cluster in a two-level model.

Given a set of estimated `xtmepoisson` parameters, $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\Sigma}})$, a profile likelihood in $\mathbf{u}_j$ is derived from the joint distribution $f(\mathbf{y}_j, \mathbf{u}_j)$ as

$$\mathcal{L}_j(\mathbf{u}_j) = \exp\left\{-c\left(\mathbf{y}_j\right)\right\}(2\pi)^{-q/2}|\widehat{\boldsymbol{\Sigma}}|^{-1/2}\exp\left\{g\left(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\Sigma}}, \mathbf{u}_j\right)\right\} \tag{1}$$

The conditional MLE of $\mathbf{u}_j$—conditional on fixed $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\Sigma}})$—is the maximizer of $\mathcal{L}_j(\mathbf{u}_j)$, or equivalently, the value of $\widehat{\mathbf{u}}_j$ that solves

$$\mathbf{0} = g'\left(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\Sigma}}, \widehat{\mathbf{u}}_j\right) = \mathbf{Z}_j'\left\{\mathbf{y}_j - \mathbf{m}(\widehat{\boldsymbol{\beta}}, \widehat{\mathbf{u}}_j)\right\} - \widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\mathbf{u}}_j$$

Because (1) is proportional to the conditional density $f(\mathbf{u}_j|\mathbf{y}_j)$, you can also refer to $\widehat{\mathbf{u}}_j$ as the *conditional mode* (or *posterior mode* if you lean toward Bayesian terminology). Regardless, you are referring to the same estimator.

Conditional standard errors for the estimated random effects are derived from standard theory of maximum likelihood, which dictates that the asymptotic variance matrix of $\widehat{\mathbf{u}}_j$ is the negative inverse of the Hessian, which is estimated as

$$g''\left(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\Sigma}}, \widehat{\mathbf{u}}_j\right) = -\left\{\mathbf{Z}_j'\mathbf{V}(\widehat{\boldsymbol{\beta}}, \widehat{\mathbf{u}}_j)\mathbf{Z}_j + \widehat{\boldsymbol{\Sigma}}^{-1}\right\}$$

Similar calculations extend to models with more than one level of random effects; see Pinheiro and Chao (2006).

For any i observation in the jth cluster in a two-level model, define the linear predictor as

$$\widehat{\eta}_{ij} = \mathbf{x}_{ij}\widehat{\boldsymbol{\beta}} + \mathbf{z}_{ij}\widehat{\mathbf{u}}_j$$

In a three-level model, for the ith observation within the jth level-two cluster within the kth level-three cluster,

$$\widehat{\eta}_{ijk} = \mathbf{x}_{ijk}\widehat{\boldsymbol{\beta}} + \mathbf{z}^{(3)}_{ijk}\widehat{\mathbf{u}}^{(3)}_{k} + \mathbf{z}^{(2)}_{ijk}\widehat{\mathbf{u}}^{(2)}_{jk}$$

where the $\mathbf{z}^{(p)}$ and $\mathbf{u}^{(p)}$ refer to the level p design variables and random effects, respectively. For models with more than three levels, the definition of $\widehat{\eta}$ extends in the natural way, with only the notation becoming more complicated.

If the `fixedonly` option is specified, $\widehat{\eta}$ contains the linear predictor for only the fixed portion of the model, for example, in a two-level model $\widehat{\eta}_{ij} = \mathbf{x}_{ij}\widehat{\boldsymbol{\beta}}$. In what follows, we assume a two-level model, with the only necessary modification for multilevel models being the indexing.

The predicted mean, conditional on the random effects $\widehat{\mathbf{u}}_j$, is

$$\widehat{\mu}_{ij} = \exp(\widehat{\eta}_{ij})$$

Pearson residuals are calculated as

$$\nu^{P}_{ij} = \frac{y_{ij} - \widehat{\mu}_{ij}}{\{V(\widehat{\mu}_{ij})\}^{1/2}}$$

for $V(\widehat{\mu}_{ij}) = \widehat{\mu}_{ij}$.

Deviance residuals are calculated as

$$\nu^{D}_{ij} = \text{sign}(y_{ij} - \widehat{\mu}_{ij})\sqrt{\widehat{d}^{2}_{ij}}$$

where

$$\widehat{d}^{2}_{ij} = \begin{cases} 2\widehat{\mu}_{ij} & \text{if } y_{ij} = 0 \\ 2\left\{ y_{ij}\log\left(\dfrac{y_{ij}}{\widehat{\mu}_{ij}}\right) - (y_{ij} - \widehat{\mu}_{ij})\right\} & \text{otherwise} \end{cases}$$

Anscombe residuals are calculated as

$$\nu^{A}_{ij} = \frac{3\left(y^{2/3}_{ij} - \widehat{\mu}^{2/3}_{ij}\right)}{2\widehat{\mu}^{1/6}_{ij}}$$

For a discussion of the general properties of the above residuals, see Hardin and Hilbe (2007, chap. 4).

References

Hardin, J. W., and J. M. Hilbe. 2007. *Generalized Linear Models and Extensions*. 2nd ed. College Station, TX: Stata Press.

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

Pinheiro, J. C., and E. C. Chao. 2006. Efficient Laplacian and adaptive Gaussian quadrature algorithms for multilevel generalized linear mixed models. *Journal of Computational and Graphical Statistics* 15: 58–81.

Rabe-Hesketh, S., and A. Skrondal. 2008. *Multilevel and Longitudinal Modeling Using Stata*. 2nd ed. College Station, TX: Stata Press.

Also see

[XT] **xtmepoisson** — Multilevel mixed-effects Poisson regression

[U] **20 Estimation and postestimation commands**

Title

xtmixed — Multilevel mixed-effects linear regression

Syntax

`xtmixed` *depvar fe_equation* [`||` *re_equation*] [`||` *re_equation* ...] [`,` *options*]

where the syntax of *fe_equation* is

[*indepvars*] [*if*] [*in*] [*weight*] [`,` *fe_options*]

and the syntax of *re_equation* is one of the following:

for random coefficients and intercepts

levelvar`:` [*varlist*] [`,` *re_options*]

for random effects among the values of a factor variable

levelvar`: R.`*varname* [`,` *re_options*]

levelvar is a variable identifying the group structure for the random effects at that level or `_all` representing one group comprising all observations.

fe_options	Description
Model	
`noconstant`	suppress constant term from the fixed-effects equation

re_options	Description
Model	
`covariance(`*vartype*`)`	variance–covariance structure of the random effects
`noconstant`	suppress constant term from the random-effects equation
`collinear`	keep collinear variables
`fweight(`*exp*`)`	frequency weights at higher levels
`pweight(`*exp*`)`	sampling weights at higher levels

vartype	Description
`independent`	one unique variance parameter per random effect, all covariances zero; the default unless a factor variable is specified
`exchangeable`	equal variances for random effects, and one common pairwise covariance
`identity`	equal variances for random effects, all covariances zero
`unstructured`	all variances and covariances distinctly estimated

options	Description
Model	
mle	fit model via maximum likelihood; the default
reml	fit model via restricted maximum likelihood
pwscale(*scale_method*)	control scaling of sampling weights in two-level models
residuals(*rspec*)	structure of residual errors
SE/Robust	
vce(*vcetype*)	*vcetype* may be oim, robust, or cluster *clustvar*
Reporting	
level(#)	set confidence level; default is level(95)
variance	show random-effects parameter estimates as variances and covariances
noretable	suppress random-effects table
nofetable	suppress fixed-effects table
estmetric	show parameter estimates in the estimation metric
noheader	suppress output header
nogroup	suppress table summarizing groups
nostderr	do not estimate standard errors of random-effects parameters
nolrtest	do not perform LR test comparing to linear regression
display_options	control column formats, row spacing, line width, and display of omitted variables and base and empty cells
EM options	
emiterate(#)	number of EM iterations; default is 20
emtolerance(#)	EM convergence tolerance; default is 1e-10
emonly	fit model exclusively using EM
emlog	show EM iteration log
emdots	show EM iterations as dots
Maximization	
maximize_options	control the maximization process; seldom used
matsqrt	parameterize variance components using matrix square roots; the default
matlog	parameterize variance components using matrix logarithms
coeflegend	display legend instead of statistics

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

depvar, *indepvars*, and *varlist* may contain time-series operators; see [U] **11.4.4 Time-series varlists**.

bootstrap, by, jackknife, mi estimate, rolling, and statsby are allowed; see [U] **11.1.10 Prefix commands**.

pweights and fweights 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.

Menu

Statistics > Longitudinal/panel data > Multilevel mixed-effects models > Mixed-effects linear regression

Description

`xtmixed` fits linear mixed models. Mixed models are characterized as containing both *fixed effects* and *random effects*. The fixed effects are analogous to standard regression coefficients and are estimated directly. The random effects are not directly estimated but are summarized according to their estimated variances and covariances. Although random effects are not directly estimated, you can form best linear unbiased predictions (BLUPs) of them (and standard errors) by using `predict` after `xtmixed`; see [XT] **xtmixed postestimation**. Random effects may take the form of either random intercepts or random coefficients, and the grouping structure of the data may consist of multiple levels of nested groups. As such, mixed models are also known in the literature as *multilevel models* and *hierarchical linear models*. The overall error distribution of the linear mixed model is assumed to be Gaussian, and heteroskedasticity and correlations within lowest-level groups also may be modeled.

Options

Model

`noconstant` suppresses the constant (intercept) term and may be specified for the fixed-effects equation and for any or all the random-effects equations.

`covariance(`*vartype*`)`, where *vartype* is

`independent` | `exchangeable` | `identity` | `unstructured`

specifies the structure of the covariance matrix for the random effects and may be specified for each random-effects equation. An `independent` covariance structure allows for a distinct variance for each random effect within a random-effects equation and assumes that all covariances are zero. `exchangeable` structure specifies one common variance for all random effects and one common pairwise covariance. `identity` is short for "multiple of the identity"; that is, all variances are equal and all covariances are zero. `unstructured` allows for all variances and covariances to be distinct. If an equation consists of p random-effects terms, the unstructured covariance matrix will have $p(p+1)/2$ unique parameters.

`covariance(independent)` is the default, except when the random-effects equation is a factor-variable specification `R.`*varname*, in which case `covariance(identity)` is the default, and only `covariance(identity)` and `covariance(exchangeable)` are allowed.

`collinear` specifies that `xtmixed` not omit collinear variables from the random-effects equation. Usually there is no reason to leave collinear variables in place, and in fact doing so usually causes the estimation to fail because of the matrix singularity caused by the collinearity. However, with certain models (for example, a random-effects model with a full set of contrasts), the variables may be collinear, yet the model is fully identified because of restrictions on the random-effects covariance structure. In such cases, using the `collinear` option allows the estimation to take place with the random-effects equation intact.

`fweight(`*exp*`)` specifies frequency weights at higher levels in a multilevel model, whereas frequency weights at the first level (the observation level) are specified in the usual manner, for example, `[fw=`*fwtvar1*`]`. *exp* can be any valid Stata expression, and you can specify `fweight()` at levels two and higher of a multilevel model. For example, in the two-level model

```
. xtmixed fixed_portion [fw = wt1] || school: ..., fweight(wt2) ...
```

variable wt1 would hold the first-level (the observation-level) frequency weights, and wt2 would hold the second-level (the school-level) frequency weights.

pweight(*exp*) specifies sampling weights at higher levels in a multilevel model, whereas sampling weights at the first level (the observation level) are specified in the usual manner, for example, [pw=*pwtvar1*]. *exp* can be any valid Stata expression, and you can specify pweight() at levels two and higher of a multilevel model. For example, in the two-level model

```
. xtmixed fixed_portion [pw = wt1] || school: ..., pweight(wt2) ...
```

variable wt1 would hold the first-level (the observation-level) sampling weights, and wt2 would hold the second-level (the school-level) sampling weights.

See *Survey data* in *Remarks* below for more information regarding the use of sampling weights in multilevel models.

Weighted estimation, whether frequency or sampling, is not supported under restricted maximum-likelihood estimation (REML).

mle and reml specify the statistical method for fitting the model.

mle, the default, specifies that the model be fit using maximum likelihood (ML).

reml specifies that the model be fit using restricted maximum likelihood (REML), also known as residual maximum likelihood.

pwscale(*scale_method*), where *scale_method* is

size | effective | gk

controls how sampling weights (if specified) are scaled in two-level models.

scale_method size specifies that first-level (observation-level) weights be scaled so that they sum to the sample size of their corresponding second-level cluster. Second-level sampling weights are left unchanged.

scale_method effective specifies that first-level weights be scaled so that they sum to the effective sample size of their corresponding second-level cluster. Second-level sampling weights are left unchanged.

scale_method gk specifies the Graubard and Korn (1996) method. Under this method, second-level weights are set to the cluster averages of the products of the weights at both levels, and first-level weights are then set equal to one.

pwscale() is supported only with two-level models. See *Survey data* in *Remarks* below for more details on using pwscale().

residuals(*rspec*), where *rspec* is

restype [, *residual_options*]

specifies the structure of the residual errors within the lowest-level groups (the second level of a multilevel model with the observations comprising the first level) of the linear mixed model. For example, if you are modeling random effects for classes nested within schools, then residuals() refers to the residual variance–covariance structure of the observations within classes, the lowest-level groups.

restype is

independent | exchangeable | ar # | ma # | unstructured |
banded # | toeplitz # | exponential

By default, *restype* is `independent`, which means that all residuals are i.i.d. Gaussian with one common variance. When combined with `by(`*varname*`)`, independence is still assumed, but you estimate a distinct variance for each level of *varname*. Unlike with the structures described below, *varname* does not need to be constant within groups.

restype `exchangeable` estimates two parameters, one common within-group variance and one common pairwise covariance. When combined with `by(`*varname*`)`, these two parameters are distinctly estimated for each level of *varname*. Because you are modeling a within-group covariance, *varname* must be constant within lowest-level groups.

restype `ar` # assumes that within-group errors have an autoregressive (AR) structure of order #; `ar 1` is the default. The `t(`*varname*`)` option is required, where *varname* is an integer-valued time variable used to order the observations within groups and to determine the lags between successive observations. Any nonconsecutive time values will be treated as gaps. For this structure, # + 1 parameters are estimated (# AR coefficients and one overall error variance). *restype* `ar` may be combined with `by(`*varname*`)`, but *varname* must be constant within groups.

restype `ma` # assumes that within-group errors have a moving average (MA) structure of order #; `ma 1` is the default. The `t(`*varname*`)` option is required, where *varname* is an integer-valued time variable used to order the observations within groups and to determine the lags between successive observations. Any nonconsecutive time values will be treated as gaps. For this structure, # + 1 parameters are estimated (# MA coefficients and one overall error variance). *restype* `ma` may be combined with `by(`*varname*`)`, but *varname* must be constant within groups.

restype `unstructured` is the most general structure; it estimates distinct variances for each within-group error and distinct covariances for each within-group error pair. The `t(`*varname*`)` option is required, where *varname* is a nonnegative-integer–valued variable that identifies the observations within each group. The groups may be unbalanced in that not all levels of `t()` need to be observed within every group, but you may not have repeated `t()` values within any particular group. When you have p levels of `t()`, then $p(p+1)/2$ parameters are estimated. *restype* `unstructured` may be combined with `by(`*varname*`)`, but *varname* must be constant within groups.

restype `banded` # is a special case of `unstructured` that restricts estimation to the covariances within the first # off-diagonals and sets the covariances outside this band to zero. The `t(`*varname*`)` option is required, where *varname* is a nonnegative-integer–valued variable that identifies the observations within each group. # is an integer between zero and $p-1$, where p is the number of levels of `t()`. By default, # is $p-1$; that is, all elements of the covariance matrix are estimated. When # is zero, only the diagonal elements of the covariance matrix are estimated. *restype* `banded` may be combined with `by(`*varname*`)`, but *varname* must be constant within groups.

restype `toeplitz` # assumes that within-group errors have Toeplitz structure of order #, for which correlations are constant with respect to time lags less than or equal to # and are zero for lags greater than #. The `t(`*varname*`)` option is required, where *varname* is an integer-valued time variable used to order the observations within groups and to determine the lags between successive observations. # is an integer between one and the maximum observed lag (the default). Any nonconsecutive time values will be treated as gaps. For this structure, # + 1 parameters are estimated (# correlations and one overall error variance). *restype* `toeplitz` may be combined with `by(`*varname*`)`, but *varname* must be constant within groups.

restype `exponential` is a generalization of the autoregressive (AR) covariance model that allows for unequally spaced and noninteger time values. The `t(`*varname*`)` option is required, where *varname* is real-valued. For the exponential covariance model, the correlation between two errors is the parameter ρ, raised to a power equal to the absolute value of the difference between the `t()` values for those errors. For this structure, two parameters are estimated (the correlation parameter ρ and one overall error variance). *restype* `exponential` may be combined with `by(`*varname*`)`, but *varname* must be constant within groups.

residual_options are `by(`*varname*`)` and `t(`*varname*`)`.

`by(`*varname*`)` is for use within the `residuals()` option and specifies that a set of distinct residual-error parameters be estimated for each level of *varname*. In other words, you use `by()` to model heteroskedasticity.

`t(`*varname*`)` is for use within the `residuals()` option to specify a time variable for the `ar`, `ma`, `toeplitz`, and `exponential` structures, or to ID the observations when *restype* is `unstructured` or `banded`.

SE/Robust

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification and that allow for intragroup correlation; see [R] ***vce_option***. `vce(oim)` is the default. If `vce(robust)` is specified, robust variances are clustered at the highest level in the multilevel model.

`vce(robust)` and `vce(cluster` *clustvar*`)` are not supported with REML estimation.

Reporting

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

`variance` displays the random-effects and residual-error parameter estimates as variances and covariances. The default is to display them as standard deviations and correlations.

`noretable` suppresses the random-effects table from the output.

`nofetable` suppresses the fixed-effects table from the output.

`estmetric` displays all parameter estimates in the estimation metric. Fixed-effects estimates are unchanged from those normally displayed, but random-effects parameter estimates are displayed as log-standard deviations and hyperbolic arctangents of correlations, with equation names that organize them by model level. Residual-variance parameter estimates are also displayed in their original estimation metric.

`noheader` suppresses the output header, either at estimation or upon replay.

`nogroup` suppresses the display of group summary information (number of groups, average group size, minimum, and maximum) from the output header.

`nostderr` prevents `xtmixed` from calculating standard errors for the estimated random-effects parameters, although standard errors are still provided for the fixed-effects parameters. Specifying this option will speed up computation times. `nostderr` is available only when residuals are modeled as independent with constant variance.

`nolrtest` prevents `xtmixed` from fitting a reference linear regression model and using this model to calculate a likelihood-ratio test comparing the mixed model to ordinary regression. This option may also be specified on replay to suppress this test from the output.

display_options: noomitted, vsquish, noemptycells, baselevels, allbaselevels, cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **estimation options**.

EM options

These options control the EM (expectation-maximization) iterations that take place before estimation switches to a gradient-based method. When residuals are modeled as independent with constant variance, EM will either converge to the solution or bring parameter estimates close to the solution. For other residual structures or for weighted estimation, EM is used to obtain starting values.

emiterate(#) specifies the number of EM iterations to perform. The default is emiterate(20).

emtolerance(#) specifies the convergence tolerance for the EM algorithm. The default is emtolerance(1e-10). EM iterations will be halted once the log (restricted) likelihood changes by a relative amount less than #. At that point, optimization switches to a gradient-based method, unless emonly is specified, in which case maximization stops.

emonly specifies that the likelihood be maximized exclusively using EM. The advantage of specifying emonly is that EM iterations are typically much faster than those for gradient-based methods. The disadvantages are that EM iterations can be slow to converge (if at all) and that EM provides no facility for estimating standard errors for the random-effects parameters. emonly is available only with unweighted estimation and when residuals are modeled as independent with constant variance.

emlog specifies that the EM iteration log be shown. The EM iteration log is, by default, not displayed unless the emonly option is specified.

emdots specifies that the EM iterations be shown as dots. This option can be convenient because the EM algorithm may require many iterations to converge.

Maximization

maximize_options: difficult, technique(*algorithm_spec*), iterate(#), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), and nonrtolerance; see [R] **maximize**. Those that require special mention for xtmixed are listed below.

For the technique() option, the default is technique(nr). The bhhh algorithm may not be specified.

matsqrt (the default), during optimization, parameterizes variance components by using the matrix square roots of the variance–covariance matrices formed by these components at each model level.

matlog, during optimization, parameterizes variance components by using the matrix logarithms of the variance–covariance matrices formed by these components at each model level.

The matsqrt parameterization ensures that variance–covariance matrices are positive semidefinite, while matlog ensures matrices that are positive definite. For most problems, the matrix square root is more stable near the boundary of the parameter space. However, if convergence is problematic, one option may be to try the alternate matlog parameterization. When convergence is not an issue, both parameterizations yield equivalent results.

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

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

Remarks

Remarks are presented under the following headings:

Introduction

Linear mixed models are models containing both fixed effects and random effects. They are a generalization of linear regression allowing for the inclusion of random deviations (effects) other than those associated with the overall error term. In matrix notation,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + \boldsymbol{\epsilon} \tag{1}$$

where $\mathbf{y}$ is the $n \times 1$ vector of responses, $\mathbf{X}$ is an $n \times p$ design/covariate matrix for the fixed effects $\boldsymbol{\beta}$, and $\mathbf{Z}$ is the $n \times q$ design/covariate matrix for the random effects $\mathbf{u}$. The $n \times 1$ vector of errors, $\boldsymbol{\epsilon}$, is assumed to be multivariate normal with mean zero and variance matrix $\sigma^2_\epsilon\mathbf{R}$.

The fixed portion of (1), $\mathbf{X}\boldsymbol{\beta}$, is analogous to the linear predictor from a standard OLS regression model with $\boldsymbol{\beta}$ being the regression coefficients to be estimated. For the random portion of (1), $\mathbf{Z}\mathbf{u}+\boldsymbol{\epsilon}$, we assume that $\mathbf{u}$ has variance–covariance matrix $\mathbf{G}$ and that $\mathbf{u}$ is orthogonal to $\boldsymbol{\epsilon}$ so that

$$\text{Var}\begin{bmatrix}\mathbf{u}\\ \boldsymbol{\epsilon}\end{bmatrix} = \begin{bmatrix}\mathbf{G} & \mathbf{0}\\ \mathbf{0} & \sigma^2_\epsilon\mathbf{R}\end{bmatrix}$$

The random effects $\mathbf{u}$ are not directly estimated (although they may be predicted), but instead are characterized by the elements of $\mathbf{G}$, known as *variance components*, that are estimated along with the overall residual variance σ^2_ϵ and the residual-variance parameters that are contained within $\mathbf{R}$.

The general forms of the design matrices $\mathbf{X}$ and $\mathbf{Z}$ allow estimation for a broad class of linear models: blocked designs, split-plot designs, growth curves, multilevel or hierarchical designs, etc. They also allow a flexible method of modeling within-cluster correlation. Subjects within the same cluster can be correlated as a result of a shared random intercept, or through a shared random slope on (say) age, or both. The general specification of $\mathbf{G}$ also provides additional flexibility—the random intercept and random slope could themselves be modeled as independent, or correlated, or independent with equal variances, and so forth. The general structure of $\mathbf{R}$ also allows for residual errors to be heteroskedastic and correlated, and allows flexibility in exactly how these characteristics can be modeled.

Comprehensive treatments of mixed models are provided by, among others, Searle, Casella, and McCulloch (1992); McCulloch, Searle, and Neuhaus (2008); Verbeke and Molenberghs (2000); Raudenbush and Bryk (2002); Demidenko (2004); and Pinheiro and Bates (2000). In particular, chapter 2 of Searle, Casella, and McCulloch (1992) provides an excellent history.

The key to fitting mixed models lies in estimating the variance components, and for that there exist many methods. Most of the early literature in mixed models dealt with estimating variance components in ANOVA models. For simple models with balanced data, estimating variance components amounts to solving a system of equations obtained by setting expected mean-squares expressions equal to their observed counterparts. Much of the work in extending the "ANOVA method" to unbalanced data for general ANOVA designs is due to Henderson (1953).

The ANOVA method, however, has its shortcomings. Among these is a lack of uniqueness in that alternative, unbiased estimates of variance components could be derived using other quadratic forms of the data in place of observed mean squares (Searle, Casella, and McCulloch 1992, 38–39). As a result, ANOVA methods gave way to more modern methods, such as minimum norm quadratic unbiased estimation (MINQUE) and minimum variance quadratic unbiased estimation (MIVQUE); see Rao (1973) for MINQUE and LaMotte (1973) for MIVQUE. Both methods involve finding optimal quadratic forms of the data that are unbiased for the variance components.

The most popular methods, however, are maximum likelihood (ML) and restricted maximum-likelihood (REML), and these are the two methods that are supported by `xtmixed`. The ML estimates are based on the usual application of likelihood theory, given the distributional assumptions of the model. The basic idea behind REML (Thompson 1962) is that you can form a set of linear contrasts of the response that do not depend on the fixed effects, $\boldsymbol{\beta}$, but instead depend only on the variance components to be estimated. You then apply ML methods by using the distribution of the linear contrasts to form the likelihood.

Returning to (1): in clustered-data situations, it is convenient not to consider all n observations at once but instead to organize the mixed model as a series of M independent groups (or clusters)

$$\mathbf{y}_j = \mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j + \boldsymbol{\epsilon}_j \tag{2}$$

for $j = 1, \ldots, M$, with cluster j consisting of n_j observations. The response, $\mathbf{y}_j$, comprises the rows of $\mathbf{y}$ corresponding with the jth cluster, with $\mathbf{X}_j$ and $\boldsymbol{\epsilon}_j$ defined analogously. The random effects, $\mathbf{u}_j$, can now be thought of as M realizations of a $q \times 1$ vector that is normally distributed with mean $\mathbf{0}$ and $q \times q$ variance matrix $\boldsymbol{\Sigma}$. The matrix $\mathbf{Z}_i$ is the $n_j \times q$ design matrix for the jth cluster random effects. Relating this to (1), note that

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Z}_M \end{bmatrix}; \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_M \end{bmatrix}; \quad \mathbf{G} = \mathbf{I}_M \otimes \boldsymbol{\Sigma}; \quad \mathbf{R} = \mathbf{I}_M \otimes \boldsymbol{\Lambda} \tag{3}$$

The mixed-model formulation (2) is from Laird and Ware (1982) and offers two key advantages. First, it makes specifications of random-effects terms easier. If the clusters are schools, you can simply specify a random effect "at the school level", as opposed to thinking of what a school-level random effect would mean when all the data are considered as a whole (if it helps, think Kronecker products). Second, representing a mixed-model with (2) generalizes easily to more than one set of random effects. For example, if classes are nested within schools, then (2) can be generalized to allow random effects at both the school and the class-within-school levels. This we demonstrate later.

Finally, we state our convention on counting and ordering model levels. Model (2) is what we call a *two-level* model, with extensions to three, four, or any number of levels. The observation y_{ij} is for individual i within cluster j, and the individuals comprise the first level and the clusters comprise the second level of the model. In our hypothetical three-level model with classes nested within schools, the observations within schools (the students, presumably) would constitute the first level, the classes would constitute the second level, and the schools would constitute the third level.

This differs from certain citations in the classical ANOVA literature and texts such as Pinheiro and Bates (2000) but is the standard in the vast literature on hierarchical models, for example, Skrondal and Rabe-Hesketh (2004).

In the sections that follow, we assume that residuals are independent with constant variance; that is, in (3) we treat $\mathbf{\Lambda}$ equal to the identity matrix and limit ourselves to estimating one overall residual variance, σ^2_ϵ. Beginning in *Heteroskedastic residual errors*, we relax this assumption.

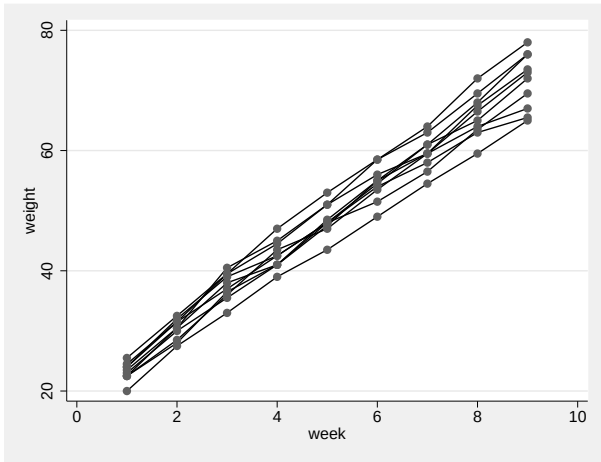
Two-level models

We begin with a simple application of (2). We begin with a two-level model because a one-level linear model, by our convention, is just standard OLS regression.

▷ Example 1

Consider a longitudinal dataset, used by both Ruppert, Wand, and Carroll (2003) and Diggle et al. (2002), consisting of `weight` measurements of 48 pigs on 9 successive `weeks`. Pigs are identified by variable `id`. Below is a plot of the growth curves for the first 10 pigs.

```
. use http://www.stata-press.com/data/r12/pig
(Longitudinal analysis of pig weights)
. twoway connected weight week if id<=10, connect(L)
```



It seems clear that each pig experiences a linear trend in growth and that overall weight measurements vary from pig to pig. Because we are not really interested in these particular 48 pigs per se, we instead treat them as a random sample from a larger population and model the between-pig variability as a random effect or, in the terminology of (2), as a random-intercept term at the pig level. We thus wish to fit the model

$$\texttt{weight}_{ij} = \beta_0 + \beta_1 \texttt{week}_{ij} + u_j + \epsilon_{ij} \tag{4}$$

for $i = 1, \ldots, 9$ weeks and $j = 1, \ldots, 48$ pigs. The fixed portion of the model, $\beta_0 + \beta_1 \texttt{week}_{ij}$, simply states that we want one overall regression line representing the population average. The random effect, u_j, serves to shift this regression line up or down according to each pig. Because the random effects occur at the pig level (`id`), we fit the model by typing

```
. xtmixed weight week || id:
Performing EM optimization:
Performing gradient-based optimization:
Iteration 0:   log likelihood = -1014.9268
Iteration 1:   log likelihood = -1014.9268
Computing standard errors:
Mixed-effects ML regression                     Number of obs      =       432
Group variable: id                              Number of groups   =        48
                                                Obs per group: min =         9
                                                               avg =       9.0
                                                               max =         9
                                                Wald chi2(1)       =  25337.49
Log likelihood = -1014.9268                     Prob > chi2        =    0.0000
```

weight	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
week	6.209896	.0390124	159.18	0.000	6.133433	6.286359
_cons	19.35561	.5974059	32.40	0.000	18.18472	20.52651

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
id: Identity				
sd(_cons)	3.849352	.4058119	3.130769	4.732866
sd(Residual)	2.093625	.0755472	1.95067	2.247056

```
LR test vs. linear regression: chibar2(01) =   472.65 Prob >= chibar2 = 0.0000
```

At this point, a guided tour of the model specification and output is in order:

1. By typing "`weight week`", we specified the response, `weight`, and the fixed portion of the model in the same way that we would if we were using `regress` or any other estimation command. Our fixed effects are a coefficient on `week` and a constant term.

2. When we added "`|| id:`", we specified random effects at the level identified by group variable `id`, that is, the pig level (level two). Because we wanted only a random intercept, that is all we had to type.

3. The estimation log consists of three parts:

   a. A set of expectation-maximization (EM) iterations used to refine starting values. By default, the iterations themselves are not displayed, but you can display them with the `emlog` option.

   b. A set of "gradient-based" iterations. By default, these are Newton–Raphson iterations, but other methods are available by specifying the appropriate `maximize` options; see [R] **maximize**.

   c. The message "Computing standard errors:". This is just to inform you that `xtmixed` has finished its iterative maximization and is now reparameterizing from a matrix-based parameterization (see *Methods and formulas*) to the natural metric of variance components and their estimated standard errors.

4. The output title, "Mixed-effects ML regression", informs us that our model was fit using ML, the default. For REML estimates, use the `reml` option.

   Because this model is a simple random-intercept model fit by ML, it would be equivalent to using `xtreg` with its `mle` option.

5. The first estimation table reports the fixed effects. We estimate $\beta_0 = 19.36$ and $\beta_1 = 6.21$.

6. The second estimation table shows the estimated variance components. The first section of the table is labeled "`id: Identity`", meaning that these are random effects at the `id` (pig) level and that their variance–covariance matrix is a multiple of the identity matrix; that is, $\mathbf{\Sigma} = \sigma_u^2\mathbf{I}$. Because we have only one random effect at this level, `xtmixed` knew that `Identity` is the only possible covariance structure. In any case, the standard deviation of the level-two errors, σ_u, is estimated as 3.85 with standard error 0.406.

   If you prefer variance estimates, $\widehat{\sigma}_u^2$, to standard deviation estimates, $\widehat{\sigma}_u$, then specify the `variance` option either at estimation or on replay.

7. The row labeled "`sd(Residual)`" displays the estimated standard deviation of the overall error term; that is, $\widehat{\sigma}_\epsilon = 2.09$. This is the standard deviation of the level-one errors, that is, the residuals.

8. Finally, a likelihood-ratio test comparing the model with one-level ordinary linear regression, model (4) without u_j, is provided and is highly significant for these data.

   We now store our estimates for later use:

```
. estimates store randint
```

◁

▷ Example 2

Extending (4) to allow for a random slope on `week` yields the model

$$\texttt{weight}_{ij} = \beta_0 + \beta_1\texttt{week}_{ij} + u_{0j} + u_{1j}\texttt{week}_{ij} + \epsilon_{ij} \tag{5}$$

fit using `xtmixed`:

```
. xtmixed weight week || id: week
Performing EM optimization:
Performing gradient-based optimization:
Iteration 0:   log likelihood = -869.03825
Iteration 1:   log likelihood = -869.03825
Computing standard errors:
Mixed-effects ML regression                     Number of obs      =       432
Group variable: id                              Number of groups   =        48
                                                Obs per group: min =         9
                                                               avg =       9.0
                                                               max =         9
                                                Wald chi2(1)       =   4689.51
Log likelihood = -869.03825                     Prob > chi2        =    0.0000
------------------------------------------------------------------------------
      weight |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        week |   6.209896   .0906819    68.48   0.000     6.032163    6.387629
       _cons |   19.35561   .3979159    48.64   0.000     18.57571    20.13551
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters |   Estimate   Std. Err.      [95% Conf. Interval]
----------------------------+-------------------------------------------------
id: Independent             |
                  sd(week) |   .6066851   .0660294      .4901417    .7509396
                 sd(_cons) |   2.599301   .2969073      2.077913    3.251515
----------------------------+-------------------------------------------------
              sd(Residual) |   1.264441   .0487958       1.17233    1.363789
------------------------------------------------------------------------------
LR test vs. linear regression:       chi2(2) =   764.42   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

```
. estimates store randslope
```

Because we did not specify a covariance structure for the random effects $(u_{0j}, u_{1j})'$, `xtmixed` used the default `Independent` structure; that is,

$$\mathbf{\Sigma} = \text{Var}\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} = \begin{bmatrix} \sigma^2_{u0} & 0 \\ 0 & \sigma^2_{u1} \end{bmatrix} \tag{6}$$

with $\widehat{\sigma}_{u0} = 2.60$ and $\widehat{\sigma}_{u1} = 0.61$. Our point estimates of the fixed effects are essentially identical to those from model (4), but note that this does not hold generally. Given the 95% confidence interval for $\widehat{\sigma}_{u1}$, it would seem that the random slope is significant, and we can use `lrtest` and our two saved estimation results to verify this fact:

```
. lrtest randslope randint
Likelihood-ratio test                                 LR chi2(1)  =     291.78
(Assumption: randint nested in randslope)             Prob > chi2 =     0.0000

Note: The reported degrees of freedom assumes the null hypothesis is not on
      the boundary of the parameter space.  If this is not true, then the
      reported test is conservative.
```

The near-zero significance level favors the model that allows for a random pig-specific regression line over the model that allows only for a pig-specific shift.

◁

❑ Technical note

At the bottom of the previous `xtmixed` output, there is a note stating that the likelihood ratio (LR) test comparing our model with standard linear regression is conservative. Also, our `lrtest` output warns us that our test comparing the random-slope model with the random-intercept model may be conservative if the null hypothesis is on the boundary. For the former, the null hypothesis is $H_0 : \sigma^2_{u0} = \sigma^2_{u1} = 0$. For the latter, the null hypothesis is $H_0 : \sigma^2_{u1} = 0$. Because variances are constrained to be positive, both null hypotheses are on the boundaries of their respective parameter spaces. `xtmixed` is capable of detecting this automatically because it compares with linear regression. `lrtest`, on the other hand, can be used to compare a wide variety of nested mixed models, making automatic detection of boundary conditions impractical. With `lrtest`, the onus is on the user to verify testing on the boundary.

By "conservative", we mean that when boundary conditions exist, the reported significance level is an upper bound on the actual significance; see *Distribution theory for likelihood-ratio tests* later in this entry for further details.

❑

❑ Technical note

LR tests with REML require identical fixed-effects specifications for both models. As stated in Ruppert, Wand, and Carroll (2003), "The reason for this is that restricted likelihood is the likelihood of the residuals after fitting the fixed effects and so is not appropriate when there is more than one fixed effects model under consideration." This is not an issue above because we used the default ML estimation, but had we fit the models using the `reml` option, we would have to confine our tests to models comparing different variance structures and not different βs.

In our example, the fixed-effects specifications for both models are identical ($\beta_0 + \beta_1$`week`), so either ML or REML would have produced valid LR tests.

Finally, `lrtest` is capable of detecting when you change fixed-effects structures under REML and will issue an error directing you to refit your models with ML. As such, there is no danger of making an inappropriate inference.

❑

Covariance structures

In example 2, we fit a model with the default `Independent` covariance given in (6). Within any random-effects level specification, we can override this default by specifying an alternative covariance structure via the `covariance()` option.

▷ Example 3

We generalize (6) to allow u_{0j} and u_{1j} to be correlated; that is,

$$\mathbf{\Sigma} = \text{Var}\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} = \begin{bmatrix} \sigma^2_{u0} & \sigma_{01} \\ \sigma_{01} & \sigma^2_{u1} \end{bmatrix}$$

```
. xtmixed weight week || id: week, covariance(unstructured) variance
  (output omitted)
Mixed-effects ML regression                     Number of obs      =       432
Group variable: id                              Number of groups   =        48
                                                Obs per group: min =         9
                                                               avg =       9.0
                                                               max =         9
                                                Wald chi2(1)       =   4649.17
Log likelihood = -868.96185                     Prob > chi2        =    0.0000
```

weight	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
week	6.209896	.0910745	68.18	0.000	6.031393	6.388399
_cons	19.35561	.3996387	48.43	0.000	18.57234	20.13889

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
id: Unstructured				
var(week)	.3715251	.0812958	.2419532	.570486
var(_cons)	6.823363	1.566194	4.351297	10.69986
cov(week,_cons)	-.0984378	.2545767	-.5973991	.4005234
var(Residual)	1.596829	.123198	1.372735	1.857505

```
LR test vs. linear regression:       chi2(3) =   764.58   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

But we do not find the correlation to be at all significant.

```
. lrtest . randslope
Likelihood-ratio test                                 LR chi2(1)  =      0.15
(Assumption: randslope nested in .)                   Prob > chi2 =    0.6959
```

In addition to specifying an alternate covariance structure, we specified the `variance` option to display variance components in the variance–covariance metric, rather than the default, which displays them as standard deviations and correlations.

◁

Instead, we could have also specified `covariance(identity)`, restricting u_{0j} and u_{1j} to not only be independent but also to have common variance, or we could have specified `covariance(exchangeable)`, which imposes a common variance but allows for a nonzero correlation.

Likelihood versus restricted likelihood

Thus far, all our examples have used maximum likelihood (ML) to estimate variance components. We could have just as easily asked for REML estimates. Refitting the model in example 2 by REML, we get

```
. xtmixed weight week || id: week, reml
  (output omitted)
Mixed-effects REML regression                   Number of obs      =       432
Group variable: id                              Number of groups   =        48
                                                Obs per group: min =         9
                                                               avg =       9.0
                                                               max =         9
                                                Wald chi2(1)       =   4592.10
Log restricted-likelihood = -870.51473          Prob > chi2        =    0.0000
```

weight	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
week	6.209896	.0916387	67.77	0.000	6.030287	6.389504
_cons	19.35561	.4021144	48.13	0.000	18.56748	20.14374

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
id: Independent				
sd(week)	.6135475	.0673971	.4947037	.7609413
sd(_cons)	2.630134	.3028832	2.09872	3.296107
sd(Residual)	1.26443	.0487971	1.172317	1.363781

```
LR test vs. linear regression:       chi2(2) =   765.92   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

Although ML estimators are based on the usual likelihood theory, the idea behind REML is to transform the response into a set of linear contrasts whose distribution is free of the fixed effects $\boldsymbol{\beta}$. The restricted likelihood is then formed by considering the distribution of the linear contrasts. Not only does this make the maximization problem free of $\boldsymbol{\beta}$, it also incorporates the degrees of freedom used to estimate $\boldsymbol{\beta}$ into the estimation of the variance components. This follows because, by necessity, the rank of the linear contrasts must be less than the number of observations.

As a simple example, consider a constant-only regression where $y_i \sim N(\mu, \sigma^2)$ for $i = 1, \ldots, n$. The ML estimate of σ^2 can be derived theoretically as the n-divided sample variance. The REML estimate can be derived by considering the first $n-1$ error contrasts, $y_i - \overline{y}$, whose joint distribution is free of μ. Applying maximum likelihood to this distribution results in an estimate of σ^2, that is, the $(n-1)$ divided sample variance, which is unbiased for σ^2.

The unbiasedness property of REML extends to all mixed models when the data are balanced, and thus REML would seem the clear choice in balanced-data problems, although in large samples the difference between ML and REML is negligible. One disadvantage of REML is that LR tests based on REML are inappropriate for comparing models with different fixed-effects specifications. ML is appropriate for such LR tests and has the advantage of being easy to explain and being the method of choice for other estimators.

Another factor to consider is that ML estimation under `xtmixed` is more feature-rich, allowing for weighted estimation and robust variance–covariance matrices, features not supported under REML. In the end, which method to use should be based both on your needs and on personal taste.

Examining the REML output, we find that the estimates of the variance components are slightly larger than the ML estimates. This is typical, because ML estimates, which do not incorporate the degrees of freedom used to estimate the fixed effects, tend to be biased downward.

Three-level models

The clustered-data representation of the mixed model given in (2) can be extended to two nested levels of clustering, creating a three-level model once the observations are considered. Formally,

$$\mathbf{y}_{jk} = \mathbf{X}_{jk}\boldsymbol{\beta} + \mathbf{Z}^{(3)}_{jk}\mathbf{u}^{(3)}_{k} + \mathbf{Z}^{(2)}_{jk}\mathbf{u}^{(2)}_{jk} + \boldsymbol{\epsilon}_{jk} \tag{7}$$

for $i = 1, \ldots, n_{jk}$ first-level observations nested within $j = 1, \ldots, M_k$ second-level groups, which are nested within $k = 1, \ldots, M$ third-level groups. Group j, k consists of n_{jk} observations, so $\mathbf{y}_{jk}$, $\mathbf{X}_{jk}$, and $\boldsymbol{\epsilon}_{jk}$ each have row dimension n_{jk}. $\mathbf{Z}^{(3)}_{jk}$ is the $n_{jk} \times q_3$ design matrix for the third-level random effects $\mathbf{u}^{(3)}_{k}$, and $\mathbf{Z}^{(2)}_{jk}$ is the $n_{jk} \times q_2$ design matrix for the second-level random effects $\mathbf{u}^{(2)}_{jk}$. Furthermore, assume that

$$\mathbf{u}^{(3)}_{k} \sim N(\mathbf{0}, \boldsymbol{\Sigma}_3); \quad \mathbf{u}^{(2)}_{jk} \sim N(\mathbf{0}, \boldsymbol{\Sigma}_2); \quad \boldsymbol{\epsilon}_{jk} \sim N(\mathbf{0}, \sigma^2_{\epsilon}\mathbf{I})$$

and that $\mathbf{u}^{(3)}_{k}$, $\mathbf{u}^{(2)}_{jk}$, and $\boldsymbol{\epsilon}_{jk}$ are independent.

Fitting a three-level model requires you to specify two random-effects "equations": one for level three, and then one for level two. The variable list for the first equation represents $\mathbf{Z}^{(3)}_{jk}$, and for the second equation represents $\mathbf{Z}^{(2)}_{jk}$; that is, you specify the levels top to bottom in `xtmixed`.

▷ Example 4

Baltagi, Song, and Jung (2001) estimate a Cobb–Douglas production function examining the productivity of public capital in each state's private output. Originally provided by Munnell (1990), the data were recorded over 1970–1986 for 48 states grouped into nine regions.

```
. use http://www.stata-press.com/data/r12/productivity
(Public Capital Productivity)
. describe
Contains data from http://www.stata-press.com/data/r12/productivity.dta
  obs:           816                          Public Capital Productivity
 vars:            11                          29 Mar 2011 10:57
 size:        29,376                          (_dta has notes)
```

variable name	storage type	display format	value label	variable label
state	byte	%9.0g		states 1-48
region	byte	%9.0g		regions 1-9
year	int	%9.0g		years 1970-1986
public	float	%9.0g		public capital stock
hwy	float	%9.0g		log(highway component of public)
water	float	%9.0g		log(water component of public)
other	float	%9.0g		log(bldg/other component of public)
private	float	%9.0g		log(private capital stock)
gsp	float	%9.0g		log(gross state product)
emp	float	%9.0g		log(non-agriculture payrolls)
unemp	float	%9.0g		state unemployment rate

```
Sorted by:
```

Because the states are nested within regions, we fit a three-level mixed model with random intercepts at both the region and the state-within-region levels. That is, we use (7) with both $\mathbf{Z}_{jk}^{(3)}$ and $\mathbf{Z}_{jk}^{(2)}$ set to the $n_{jk} \times 1$ column of ones, and $\mathbf{\Sigma}_3 = \sigma_3^2$ and $\mathbf{\Sigma}_2 = \sigma_2^2$ are both scalars.

```
. xtmixed gsp private emp hwy water other unemp || region: || state:
  (output omitted)
Mixed-effects ML regression                     Number of obs      =       816
```

Group Variable	No. of Groups	Observations per Group Minimum	Average	Maximum
region	9	51	90.7	136
state	48	17	17.0	17

```
                                                Wald chi2(6)       =  18829.06
Log likelihood =  1430.5017                     Prob > chi2        =    0.0000
```

gsp	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
private	.2671484	.0212591	12.57	0.000	.2254814	.3088154
emp	.754072	.0261868	28.80	0.000	.7027468	.8053973
hwy	.0709767	.023041	3.08	0.002	.0258172	.1161363
water	.0761187	.0139248	5.47	0.000	.0488266	.1034109
other	-.0999955	.0169366	-5.90	0.000	-.1331906	-.0668004
unemp	-.0058983	.0009031	-6.53	0.000	-.0076684	-.0041282
_cons	2.128823	.1543854	13.79	0.000	1.826233	2.431413

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
region: Identity				
sd(_cons)	.038087	.0170591	.0158316	.091628
state: Identity				
sd(_cons)	.0792193	.0093861	.0628027	.0999273
sd(Residual)	.0366893	.000939	.0348944	.0385766

LR test vs. linear regression: chi2(2) = 1154.73 Prob > chi2 = 0.0000

Note: LR test is conservative and provided only for reference.

Some items of note:

1. Our model now has two random-effects equations, separated by `||`. The first is a random intercept (constant only) at the `region` level (level three), and the second is a random intercept at the `state` level (level two). The order in which these are specified (from left to right) is significant—`xtmixed` assumes that `state` is nested within `region`.
2. The information on groups is now displayed as a table, with one row for each grouping. You can suppress this table with the `nogroup` or the `noheader` option, which will suppress the rest of the header, as well.
3. The variance-component estimates are now organized and labeled according to level.

After adjusting for the nested-level error structure, we find that the highway and water components of public capital had significant positive effects on private output, whereas the other public buildings component had a negative effect.

◁

❑ Technical note

In the previous example, the states are coded 1–48 and are nested within nine regions. `xtmixed` treated the states as nested within regions, regardless of whether the codes for each state are unique between regions. That is, even if codes for states were duplicated between regions, `xtmixed` would have enforced the nesting and produced the same results.

The group information at the top of `xtmixed` output and that produced by the postestimation command `estat group` (see [XT] **xtmixed postestimation**) take the nesting into account. The statistics are thus not necessarily what you would get if you instead `tabulated` each group variable individually.

❑

Model (7) extends in a straightforward manner to more than three levels, as does the specification of such models in `xtmixed`.

Blocked-diagonal covariance structures

Covariance matrices of random effects within an equation can be modeled either as a multiple of the identity matrix, diagonal (that is, `Independent`), exchangeable, or as general symmetric (`Unstructured`). These may also be combined to produce more complex block-diagonal covariance structures, effectively placing constraints on the variance components.

▷ Example 5

Returning to our productivity data, we now add random coefficients on `hwy` and `unemp` at the `region` level. This only slightly changes the estimates of the fixed effects, so we focus our attention on the variance components:

```
. xtmixed gsp private emp hwy water other unemp || region: hwy unemp || state:,
> nolog nogroup nofetable
Mixed-effects ML regression                     Number of obs      =       816
                                                Wald chi2(6)       =  17137.94
Log likelihood =  1447.6787                     Prob > chi2        =    0.0000
```

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
region: Independent				
sd(hwy)	.0045717	.0120663	.0000259	.8066567
sd(unemp)	.0048777	.0013807	.0028007	.0084948
sd(_cons)	.0550901	.0786743	.0033533	.9050571
state: Identity				
sd(_cons)	.0797859	.0097832	.0627412	.101461
sd(Residual)	.0353108	.0009104	.0335708	.037141

```
LR test vs. linear regression:       chi2(4) =  1189.08   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
. estimates store prodrc
```

This model is the same as that fit in example 4, except that $\mathbf{Z}_{jk}^{(3)}$ is now the $n_{jk} \times 3$ matrix with columns determined by the values of `hwy`, `unemp`, and an intercept term (one), in that order, and (because we used the default `Independent` structure) $\mathbf{\Sigma}_3$ is

$$\mathbf{\Sigma}_3 = \begin{array}{c} \begin{array}{ccc} \texttt{hwy} & \texttt{unemp} & \texttt{\_cons} \end{array} \\ \begin{pmatrix} \sigma_a^2 & 0 & 0 \\ 0 & \sigma_b^2 & 0 \\ 0 & 0 & \sigma_c^2 \end{pmatrix} \end{array}$$

The random-effects specification at the state level remains unchanged; that is, $\mathbf{\Sigma}_2$ is still treated as the scalar variance of the random intercepts at the state level.

An LR test comparing this model with that from example 4 favors the inclusion of the two random coefficients, a fact we leave to the interested reader to verify.

Examining the estimated variance components reveals that the variances of the random coefficients on `hwy` and `unemp` could be treated as equal. That is,

$$\mathbf{\Sigma}_3 = \begin{array}{c} \begin{array}{ccc} \texttt{hwy} & \texttt{unemp} & \texttt{\_cons} \end{array} \\ \begin{pmatrix} \sigma_a^2 & 0 & 0 \\ 0 & \sigma_a^2 & 0 \\ 0 & 0 & \sigma_c^2 \end{pmatrix} \end{array}$$

looks plausible. We can impose this equality constraint by treating $\mathbf{\Sigma}_3$ as block diagonal: the first block is a 2×2 multiple of the identity matrix, that is, $\sigma_a^2 \mathbf{I}_2$; the second is a scalar, equivalently, a 1×1 multiple of the identity.

We construct block-diagonal covariances by repeating level specifications:

```
. xtmixed gsp private emp hwy water other unemp || region: hwy unemp,
> cov(identity) || region: || state:, nolog nogroup nofetable
Mixed-effects ML regression                     Number of obs      =       816
                                                Wald chi2(6)       =  17136.65
Log likelihood =  1447.6784                     Prob > chi2        =    0.0000
```

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
region: Identity				
sd(hwy unemp)	.0048802	.001376	.0028082	.0084809
region: Identity				
sd(_cons)	.0530951	.0286555	.0184356	.1529149
state: Identity				
sd(_cons)	.0797369	.0095999	.0629766	.1009577
sd(Residual)	.0353111	.0009104	.0335712	.0371413

```
LR test vs. linear regression:       chi2(3) =  1189.08   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

We specified two equations for the `region` level: the first for the random coefficients on `hwy` and `unemp` with covariance set to `Identity` and the second for the random intercept `_cons`, whose covariance defaults to `Identity` because it is of dimension one. `xtmixed` labeled the estimate of σ_a as "`sd(hwy unemp)`" to designate that it is common to the random coefficients on both `hwy` and `unemp`.

An LR test shows that the constrained model fits equally well.

```
. lrtest . prodrc
Likelihood-ratio test                                 LR chi2(1)  =      0.00
(Assumption: . nested in prodrc)                      Prob > chi2 =    0.9784
Note: The reported degrees of freedom assumes the null hypothesis is not on
      the boundary of the parameter space.  If this is not true, then the
      reported test is conservative.
```

◁

Because the null hypothesis for this test is one of equality ($H_0 : \sigma_a^2 = \sigma_b^2$), it is not on the boundary of the parameter space. As such, we can take the reported significance as precise rather than a conservative estimate.

You can repeat level specifications as often as you like, defining successive blocks of a block-diagonal covariance matrix. However, repeated-level equations must be listed consecutively; otherwise, `xtmixed` will give an error.

❑ Technical note

In the previous estimation output, there was no constant term included in the first `region` equation, even though we did not use the `noconstant` option. When you specify repeated-level equations, `xtmixed` knows not to put constant terms in each equation because such a model would be unidentified. By default, it places the constant in the last repeated-level equation, but you can use `noconstant` creatively to override this.

❑

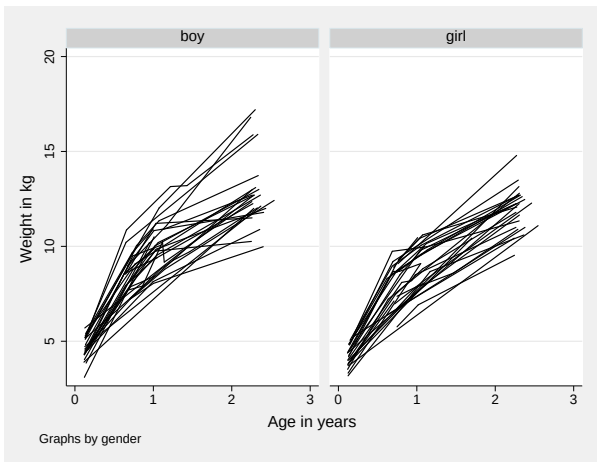
Heteroskedastic random effects

Blocked-diagonal covariance structures and repeated-level specifications of random effects can also be used to model heteroskedasticity among random effects at a given level.

▷ Example 6

Following Rabe-Hesketh and Skrondal (2008, sec. 5.10), we analyze data from Asian children in a British community who were weighed up to four times, roughly between the ages of 6 weeks and 27 months. The dataset is a random sample of data previously analyzed by Goldstein (1986) and Prosser, Rasbash, and Goldstein (1991).

```
. use http://www.stata-press.com/data/r12/childweight
(Weight data on Asian children)
. describe
Contains data from http://www.stata-press.com/data/r12/childweight.dta
  obs:           198                          Weight data on Asian children
 vars:             5                          23 May 2011 15:12
 size:         3,168                          (_dta has notes)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
id              int    %8.0g                  child identifier
age             float  %8.0g                  age in years
weight          float  %8.0g                  weight in Kg
brthwt          int    %8.0g                  Birth weight in g
girl            float  %9.0g       bg         gender
-------------------------------------------------------------------------------
Sorted by:  id  age
. graph twoway (line weight age, connect(ascending)), by(girl)
> xtitle(Age in years) ytitle(Weight in kg)
```



Ignoring gender effects for the moment, we begin with the following model for the ith measurement on the jth child:

$$\texttt{weight}_{ij} = \beta_0 + \beta_1 \texttt{age}_{ij} + \beta_2 \texttt{age}_{ij}^2 + u_{j0} + u_{j1}\texttt{age}_{ij} + \epsilon_{ij}$$

The above models overall mean growth as quadratic in age and allows for two child-specific random effects: a random intercept, u_{j0}, that represents each child's vertical shift from the overall mean (β_0), and a random age slope, u_{j1}, that represents each child's deviation in linear growth rate from the overall mean linear growth rate (β_1). For reasons of simplicity, we do not consider child-specific changes in the quadratic component of growth.

```
. xtmixed weight age c.age#c.age || id: age, nolog
Mixed-effects ML regression                     Number of obs      =       198
Group variable: id                              Number of groups   =        68
                                                Obs per group: min =         1
                                                               avg =       2.9
                                                               max =         5
                                                Wald chi2(2)       =   1863.46
Log likelihood = -258.51915                     Prob > chi2        =    0.0000
```

weight	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
age	7.693701	.2381076	32.31	0.000	7.227019	8.160384
c.age#c.age	-1.654542	.0874987	-18.91	0.000	-1.826037	-1.483048
_cons	3.497628	.1416914	24.68	0.000	3.219918	3.775338

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
id: Independent				
sd(age)	.5465535	.075708	.4166057	.7170347
sd(_cons)	.7087917	.0996506	.5380794	.9336647
sd(Residual)	.5561382	.0426951	.4784488	.6464426

```
LR test vs. linear regression:       chi2(2) =   114.70   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

◁

Because there is no reason to believe that the random effects are uncorrelated, it is always a good idea to first fit a model with the `covariance(unstructured)` option. We do not include the output for such a model because for these data the correlation between random effects is not significant, but we did check this before reverting to `xtmixed`'s default `Independent` structure.

Next we introduce gender effects into the fixed portion of the model by including a main gender effect and gender/age interaction for overall mean growth:

```
. xtmixed weight i.girl i.girl#c.age c.age#c.age || id: age, nolog
Mixed-effects ML regression                     Number of obs      =       198
Group variable: id                              Number of groups   =        68
                                                Obs per group: min =         1
                                                               avg =       2.9
                                                               max =         5
                                                Wald chi2(4)       =   1942.30
Log likelihood =   -253.182                     Prob > chi2        =    0.0000
```

weight	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
1.girl	-.5104676	.2145529	-2.38	0.017	-.9309835	-.0899516
girl#c.age						
0	7.806765	.2524583	30.92	0.000	7.311956	8.301574
1	7.577296	.2531318	29.93	0.000	7.081166	8.073425
c.age#c.age	-1.654323	.0871752	-18.98	0.000	-1.825183	-1.483463
_cons	3.754275	.1726404	21.75	0.000	3.415906	4.092644

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
id: Independent				
sd(age)	.5265782	.0730408	.4012307	.6910851
sd(_cons)	.6385054	.0969921	.4740922	.8599364
sd(Residual)	.5596163	.0426042	.4820449	.6496707

```
LR test vs. linear regression:       chi2(2) =   104.39   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
. estimates store homoskedastic
```

The main gender effect is significant at the 5% level, but the gender/age interaction is not:

```
. test 0.girl#c.age = 1.girl#c.age
 ( 1)  [weight]0b.girl#c.age - [weight]1.girl#c.age = 0
           chi2(  1) =    1.66
         Prob > chi2 =    0.1978
```

On average, boys are heavier than girls but their average linear growth rates are not significantly different.

In the above model, we introduced a gender effect on average growth, but we still assumed that the variability in child-specific deviations from this average was the same for boys and girls. To check this assumption, we introduce gender into the random component of the model. Because support for factor-variable notation is limited in specifications of random effects (see *Random-effects factor notation and crossed-effects models* below), we need to generate the interactions ourselves.

```
. gen boy = !girl
. gen boyXage = boy*age
. gen girlXage = girl*age
. xtmixed weight i.girl i.girl#c.age c.age#c.age || id: boy boyXage, noconstant
> || id: girl girlXage, noconstant nolog nofetable
Mixed-effects ML regression                     Number of obs      =       198
Group variable: id                              Number of groups   =        68
                                                Obs per group: min =         1
                                                               avg =       2.9
                                                               max =         5
                                                Wald chi2(4)       =   2358.11
Log likelihood = -248.94752                     Prob > chi2        =    0.0000
```

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
id: Independent				
sd(boy)	.5622358	.138546	.3468691	.9113211
sd(boyXage)	.6880757	.1144225	.4966919	.9532031
id: Independent				
sd(girl)	.7614904	.1286769	.5467994	1.060476
sd(girlXage)	.257805	.1073047	.1140251	.582884
sd(Residual)	.5548717	.0418872	.4785591	.6433534

```
LR test vs. linear regression:       chi2(4) =   112.86   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
. estimates store heteroskedastic
```

In the above, we suppress displaying the fixed portion of the model (the `nofetable` option) because it does not differ much from that of the previous model.

Our previous model had the random effects specification

```
|| id: age
```

which we have replaced with the dual repeated-level specification

```
|| id: boy boyXage, noconstant || id: girl girlXage, noconstant
```

The former models a random intercept and random slope on age, and does so treating all children as a random sample from one population. The latter also specifies a random intercept and random slope on age, but allows for the variability of the random intercepts and slopes to differ between boys and girls. In other words, it allows for heteroskedasticity in random effects due to gender. We use the `noconstant` option so that we can separate the overall random intercept (automatically provided by the former syntax) into one specific to boys and one specific to girls.

There seems to be a large gender effect in the variability of linear growth rates. We can compare both models with a likelihood-ratio test, recalling that we saved the previous estimation results under the name `homoskedastic`:

```
. lrtest homoskedastic heteroskedastic
Likelihood-ratio test                                 LR chi2(2)  =      8.47
(Assumption: homoskedastic nested in heteroskedas~c)  Prob > chi2 =    0.0145
Note: The reported degrees of freedom assumes the null hypothesis is not on
      the boundary of the parameter space.  If this is not true, then the
      reported test is conservative.
```

Because the null hypothesis here is one of equality of variances and not that variances are zero, the above does not test on the boundary, and thus we can treat the significance level as precise and not conservative. Either way, the results favor the new model with heteroskedastic random effects.

Heteroskedastic residual errors

Up to this point, we have assumed that the level-one residual errors—the ϵ's in the stated models—have been i.i.d. Gaussian with variance σ^2_ϵ. This is demonstrated in `xtmixed` output in the random-effects table, where up until now we have estimated a single residual-error standard deviation or variance, labeled as `sd(Residual)` or `var(Residual)`, respectively.

To relax the assumptions of homoskedasticity or independence of residual errors, use the `residuals()` option.

▷ Example 7

West, Welch, and Galecki (2007, chap. 7) analyze data studying the effect of ceramic dental veneer placement on gingival (gum) health. Data on 55 teeth located in the maxillary arches of 12 patients were considered.

```
. use http://www.stata-press.com/data/r12/veneer, clear
(Dental veneer data)
. describe
Contains data from http://www.stata-press.com/data/r12/veneer.dta
  obs:           110                          Dental veneer data
 vars:             7                          24 May 2011 12:11
 size:         1,100                          (_dta has notes)
------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
------------------------------------------------------------------------------
patient         byte   %8.0g                  Patient ID
tooth           byte   %8.0g                  Tooth number with patient
gcf             byte   %8.0g                  Gingival crevicular fluid (GCF)
age             byte   %8.0g                  Patient age
base_gcf        byte   %8.0g                  Baseline GCF
cda             float  %9.0g                  Average contour difference after
                                                veneer placement
followup        byte   %9.0g       t          Follow-up time: 3 or 6 months
------------------------------------------------------------------------------
Sorted by:
```

Veneers were placed to match the original contour of the tooth as closely as possible, and researchers were interested in how contour differences (variable `cda`) impacted gingival health. Gingival health was measured as the amount of gingival crevical fluid (GCF) at each tooth, measured at baseline (variable `base_gcf`) and at two posttreatment follow-ups at 3 and 6 months. Variable `gcf` records GCF at follow-up, and variable `followup` records the follow-up time.

Because two measurements were taken for each tooth and there exist multiple teeth per patient, we fit a three-level model with the following random effects: a random intercept and random slope on follow-up time at the patient level, and a random intercept at the tooth level. For the ith measurement of the jth tooth from the kth patient, we have

$$\begin{aligned}\texttt{gcf}_{ijk} = {} & \beta_0 + \beta_1\texttt{followup}_{ijk} + \beta_2\texttt{base\_gcf}_{ijk} + \beta_3\texttt{cda}_{ijk} + \beta_4\texttt{age}_{ijk} + \\ & u_{0k} + u_{1k}\texttt{followup}_{ijk} + v_{0jk} + \epsilon_{ijk}\end{aligned}$$

which we can fit using `xtmixed` as

```
. xtmixed gcf followup base_gcf cda age || patient: followup, cov(un) || tooth:,
> reml nolog
Mixed-effects REML regression                   Number of obs      =       110
```

Group Variable	No. of Groups	Observations per Group Minimum	Average	Maximum
patient	12	2	9.2	12
tooth	55	2	2.0	2

```
                                                Wald chi2(4)       =      7.48
Log restricted-likelihood = -420.92761          Prob > chi2        =    0.1128
```

gcf	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
followup	.3009815	1.936863	0.16	0.877	-3.4952	4.097163
base_gcf	-.0183127	.1433094	-0.13	0.898	-.299194	.2625685
cda	-.329303	.5292525	-0.62	0.534	-1.366619	.7080128
age	-.5773932	.2139656	-2.70	0.007	-.9967582	-.1580283
_cons	45.73862	12.55497	3.64	0.000	21.13133	70.34591

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
patient: Unstructured				
sd(followup)	6.472072	1.452392	4.168943	10.04756
sd(_cons)	22.91255	5.521438	14.28736	36.74472
corr(followup,_cons)	-.9469371	.0394744	-.9878843	-.7827271
tooth: Identity				
sd(_cons)	6.888932	1.207033	4.886635	9.711668
sd(Residual)	6.990496	.7513934	5.662578	8.629822

```
LR test vs. linear regression:       chi2(4) =    91.12   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

We used REML estimation above for no other reason than variety.

Among the other features of the model fit, we note that the residual standard deviation, σ_ϵ, was estimated as 6.99 and that our model assumed that the residuals were independent with constant variance (homoskedastic). Because it may be the case that the precision of `gcf` measurements could change over time, we modify the above to estimate two distinct error standard deviations: one for the 3-month follow-up and one for the 6-month follow-up.

To fit this model, we add the `residuals(independent, by(followup))` option, which maintains independence of residual errors but allows for heteroskedasticity with respect to follow-up time.

```
. xtmixed gcf followup base_gcf cda age || patient: followup, cov(un) || tooth:,
> residuals(independent, by(followup)) reml nolog
Mixed-effects REML regression                   Number of obs      =       110
```

Group Variable	No. of Groups	Observations per Group Minimum	Average	Maximum
patient	12	2	9.2	12
tooth	55	2	2.0	2

```
                                                Wald chi2(4)       =      7.51
Log restricted-likelihood =  -420.4576          Prob > chi2        =    0.1113
```

gcf	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
followup	.2703944	1.933096	0.14	0.889	-3.518405	4.059193
base_gcf	.0062144	.1419121	0.04	0.965	-.2719283	.284357
cda	-.2947235	.5245126	-0.56	0.574	-1.322749	.7333023
age	-.5743755	.2142249	-2.68	0.007	-.9942487	-.1545024
_cons	45.15089	12.51452	3.61	0.000	20.62288	69.6789

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
patient: Unstructured				
sd(followup)	6.461555	1.449333	4.163051	10.02911
sd(_cons)	22.69806	5.55039	14.0554	36.65509
corr(followup,_cons)	-.9480776	.0395764	-.9885662	-.7800707
tooth: Identity				
sd(_cons)	6.881798	1.198038	4.892355	9.680234
Residual: Independent, by followup				
3 months: sd(e)	7.833764	1.17371	5.840331	10.5076
6 months: sd(e)	6.035612	1.240554	4.034281	9.029765

```
LR test vs. linear regression:       chi2(5) =    92.06   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

Comparison of both models via a likelihood-ratio test reveals the difference in residual standard deviations as not significant, something we leave to you to verify as an exercise.

◁

The default residual-variance structure is `independent`, and when specified without `by()` is equivalent to the default behavior of `xtmixed`: estimating one overall residual standard deviation/variance for the entire model.

Other residual-error structures

Besides the default `independent` residual-error structure, `xtmixed` supports four other structures that allow for correlation between residual errors within the lowest-level (smallest/level two) groups. For purposes of notation, in what follows we assume a two-level model, with the obvious extension to higher-level models.

The `exchangeable` structure assumes one overall variance and one common pairwise covariance; that is,

$$\text{Var}(\boldsymbol{\epsilon}_j) = \text{Var}\begin{bmatrix}\epsilon_{j1}\\ \epsilon_{j2}\\ \vdots\\ \epsilon_{jn_j}\end{bmatrix} = \begin{bmatrix}\sigma_\epsilon^2 & \sigma_1 & \cdots & \sigma_1\\ \sigma_1 & \sigma_\epsilon^2 & \cdots & \sigma_1\\ \vdots & \vdots & \ddots & \vdots\\ \sigma_1 & \sigma_1 & \sigma_1 & \sigma_\epsilon^2\end{bmatrix}$$

By default, `xtmixed` will report estimates of the two parameters as estimates of the common standard deviation, σ_ϵ, and of pairwise correlation. If the `variance` option is specified, you obtain estimates of σ_ϵ^2 and the covariance σ_1. When the `by(`*varname*`)` option is also specified, these two parameters are estimated for each level *varname*.

The `ar` p structure assumes that the errors have an autoregressive structure of order p. That is,

$$\epsilon_{ij} = \phi_1\epsilon_{i-1,j} + \cdots + \phi_p\epsilon_{i-p,j} + u_{ij}$$

where u_{ij} are i.i.d. Gaussian with mean zero and variance σ_u^2. `xtmixed` reports estimates of $\phi_1, \ldots, \phi_p$ and the overall error standard deviation σ_ϵ (or variance if the `variance` option is specified), which can be derived from the above expression. The `t(`*varname*`)` option is required, where *varname* is a time variable used to order the observations within lowest-level groups and to determine any gaps between observations. When the `by(`*varname*`)` option is also specified, the set of $p+1$ parameters is estimated for each level of *varname*. If $p = 1$, then the estimate of ϕ_1 is reported as "`rho`", because in this case it represents the correlation between successive error terms.

The `ma` q structure assumes that the errors are a moving average process of order q. That is,

$$\epsilon_{ij} = u_{ij} + \theta_1 u_{i-1,j} + \cdots + \theta_q u_{i-q,j}$$

where u_{ij} are i.i.d. Gaussian with mean zero and variance σ_u^2. `xtmixed` reports estimates of $\theta_1, \ldots, \theta_q$ and the overall error standard deviation σ_ϵ (or variance if the `variance` option is specified), which can be derived from the above expression. The `t(`*varname*`)` option is required, where *varname* is a time variable used to order the observations within lowest level groups and to determine any gaps between observations. When the `by(`*varname* `option)` is also specified, the set of $q+1$ parameters is estimated for each level of *varname*.

The `unstructured` structure is the most general and estimates unique variances and unique pairwise covariances for all residuals within the lowest level grouping. Because the data may be unbalanced and the ordering of the observations is arbitrary, the `t(`*varname*`)` option is required, where *varname* is an ID variable that matches error terms in different groups. If *varname* has n distinct levels, then $n(n+1)/2$ parameters are estimated. Not all n levels need to be observed within each group, but duplicated levels of *varname* within a given group are not allowed because they would cause a singularity in the estimated error variance matrix for that group. When the `by(`*varname*`)` option is also specified, the set of $n(n+1)/2$ parameters is estimated for each level of *varname*.

The `banded` q structure is a special case of `unstructured` that confines estimation to within the first q off-diagonal elements of the residual variance–covariance matrix and sets the covariances outside this band to zero. As is the case with `unstructured`, the `t(`*varname*`)` is required, where *varname* is an ID variable that matches error terms in different groups. However, with `banded` variance structures, the ordering of the values in *varname* is significant because it determines which covariances are to be estimated and which are to be set to zero. For example, if *varname* has $n = 5$ distinct values $t = 1, 2, 3, 4, 5$, then a banded variance–covariance structure of order $q = 2$ would estimate the following:

$$\text{Var}(\boldsymbol{\epsilon}_j) = \text{Var}\begin{bmatrix}\epsilon_{1j}\\ \epsilon_{2j}\\ \epsilon_{3j}\\ \epsilon_{4j}\\ \epsilon_{5j}\end{bmatrix} = \begin{bmatrix}\sigma_1^2 & \sigma_{12} & \sigma_{13} & 0 & 0\\ \sigma_{12} & \sigma_2^2 & \sigma_{23} & \sigma_{24} & 0\\ \sigma_{13} & \sigma_{23} & \sigma_3^2 & \sigma_{34} & \sigma_{35}\\ 0 & \sigma_{24} & \sigma_{34} & \sigma_4^2 & \sigma_{45}\\ 0 & 0 & \sigma_{35} & \sigma_{45} & \sigma_5^2\end{bmatrix}$$

In other words, you would have an unstructured variance matrix that constrains $\sigma_{14} = \sigma_{15} = \sigma_{25} = 0$. If *varname* has n distinct levels, then $(q+1)(2n-q)/2$ parameters are estimated. Not all n levels need to be observed within each group, but duplicated levels of *varname* within a given group are not allowed because they would cause a singularity in the estimated error variance matrix for that group. When the `by(`*varname*`)` option is also specified, the set of parameters is estimated for each level of *varname*. If q is left unspecified, then `banded` is equivalent to `unstructured`; that is, all variances and covariances are estimated. When $q = 0$, $\text{Var}(\boldsymbol{\epsilon}_j)$ is treated as diagonal and can thus be used to model uncorrelated, yet heteroskedastic residual errors.

The `toeplitz` q structure assumes that the residual errors are homoskedastic and that the correlation between two errors is determined by the time lag between the two. That is, $\text{Var}(\epsilon_{ij}) = \sigma_\epsilon^2$ and

$$\text{Corr}(\epsilon_{ij}, \epsilon_{i+k,j}) = \rho_k$$

If the lag k is less than or equal to q, then the pairwise correlation ρ_k is estimated; if the lag is greater than q, then ρ_k is assumed to be zero. If q is left unspecified, then ρ_k is estimated for each observed lag k. The `t(`*varname*`)` option is required, where *varname* is a time variable t used to determine the lags between pairs of residual errors. As such, `t()` must be integer-valued. $q+1$ parameters are estimated, one overall variance σ_ϵ^2 and q correlations. When the `by(`*varname*`)` option is also specified, the set of $q+1$ parameters is estimated for each level of *varname*.

The `exponential` structure is a generalization of the AR structure that allows for noninteger and irregularly spaced time lags. That is, $\text{Var}(\epsilon_{ij}) = \sigma_\epsilon^2$ and

$$\text{Corr}(\epsilon_{ij}, \epsilon_{kj}) = \rho^{|i-k|}$$

for $0 \leq \rho \leq 1$, with i and k not required to be integers. The `t(`*varname*`)` option is required, where *varname* is a time variable used to determine i and k for each residual-error pair. `t()` is real-valued. `xtmixed` reports estimates of σ_ϵ^2 and ρ. When the `by(`*varname*`)` option is also specified, these two parameters are estimated for each level of *varname*.

▷ Example 8

Pinheiro and Bates (2000, chap. 5) analyze data from a study of the estrus cycles of mares. Originally analyzed in Pierson and Ginther (1987), the data record the number of ovarian follicles larger than 10mm, daily over a period ranging from three days before ovulation to three days after the subsequent ovulation.

```
. use http://www.stata-press.com/data/r12/ovary
(Ovarian follicles in mares)
. describe
Contains data from http://www.stata-press.com/data/r12/ovary.dta
  obs:           308                          Ovarian follicles in mares
 vars:             6                          20 May 2011 13:49
 size:         5,544                          (_dta has notes)
```

variable name	storage type	display format	value label	variable label
mare	byte	%9.0g		mare ID
stime	float	%9.0g		Scaled time
follicles	byte	%9.0g		Number of ovarian follicles > 10 mm in diameter
sin1	float	%9.0g		sine(2*pi*stime)
cos1	float	%9.0g		cosine(2*pi*stime)
time	float	%9.0g		time order within mare

```
Sorted by:  mare  stime
```

The `stime` variable is time that has been scaled so that ovulation occurs at scaled times 0 and 1, and the `time` variable records the time ordering within mares. Because graphical evidence suggests a periodic behavior, the analysis includes the `sin1` and `cos1` variables, which are sine and cosine transformations of scaled time, respectively.

We consider the following model for the ith measurement on the jth mare:

$$\texttt{follicles}_{ij} = \beta_0 + \beta_1 \texttt{sin1}_{ij} + \beta_2 \texttt{cos1}_{ij} + u_j + \epsilon_{ij}$$

The above model incorporates the cyclical nature of the data as affecting the overall average number of follicles and includes mare-specific random effects u_j. Because we believe successive measurements within each mare are probably correlated (even after controlling for the periodicity in the average), we also model the within-mare errors as being autoregressive of order 2.

```
. xtmixed follicles sin1 cos1 || mare:, residuals(ar 2, t(time)) reml nolog
Mixed-effects REML regression                   Number of obs      =       308
Group variable: mare                            Number of groups   =        11
                                                Obs per group: min =        25
                                                               avg =      28.0
                                                               max =        31
                                                Wald chi2(2)       =     34.72
Log restricted-likelihood = -772.59855          Prob > chi2        =    0.0000
```

follicles	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
sin1	-2.899228	.5110786	-5.67	0.000	-3.900923	-1.897532
cos1	-.8652936	.5432926	-1.59	0.111	-1.930127	.1995402
_cons	12.14455	.9473617	12.82	0.000	10.28775	14.00134

```
------------------------------------------------------------------------------
  Random-effects Parameters |   Estimate   Std. Err.      [95% Conf. Interval]
----------------------------+-------------------------------------------------
mare: Identity              |
                  sd(_cons) |   2.663158   .8264476      1.449596    4.892683
----------------------------+-------------------------------------------------
Residual: AR(2)             |
                       phi1 |    .5386104   .0624899      .4161325    .6610883
                       phi2 |     .144671    .063204      .0207934    .2685486
                     sd(e)  |   3.775055   .3225437      3.192979    4.463244
------------------------------------------------------------------------------
LR test vs. linear regression:       chi2(3) =   251.67   Prob > chi2 = 0.0000

Note: LR test is conservative and provided only for reference.
```

We picked an order of 2 as a guess, but we could have used likelihood-ratio tests of competing AR models to determine the optimal order, because models of smaller order are nested within those of larger order.

◁

▷ Example 9

Fitzmaurice, Laird, and Ware (2004, chap. 7) analyzed data on 37 subjects who participated in an exercise therapy trial.

```
. use http://www.stata-press.com/data/r12/exercise
(Exercise Therapy Trial)

. describe

Contains data from http://www.stata-press.com/data/r12/exercise.dta
  obs:           259                          Exercise Therapy Trial
 vars:             4                          24 Jun 2010 18:35
 size:         1,036                          (_dta has notes)
------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
------------------------------------------------------------------------------
id              byte   %9.0g                  Person ID
day             byte   %9.0g                  Day of measurement
program         byte   %9.0g                  1 = reps increase; 2 = weights
                                                increase
strength        byte   %9.0g                  Strength measurement
------------------------------------------------------------------------------
Sorted by:  id  day
```

Subjects (variable `id`) were placed on either an increased-repetition regimen (`program==1`) or a program that kept the repetitions constant but increased weight (`program==2`). Muscle-strength measurements (variable `strength`) were taken at baseline (`day==0`) and then at every two days over the next twelve days.

Following Fitzmaurice, Laird, and Ware (2004, chap. 7), and to demonstrate fitting residual-error structures to data collected at uneven time points, we confine our analysis to those data collected at baseline (day 0) and at days 4, 6, 8, and 12. We fit a full two-way factorial model of strength on program and day, with an unstructured residual-error covariance matrix over those repeated measurements taken on the same subject:

```
. keep if inlist(day, 0, 4, 6, 8, 12)
(74 observations deleted)
. xtmixed strength i.program##i.day || id:,
> noconstant residuals(unstructured, t(day)) nolog
Mixed-effects ML regression                     Number of obs      =       173
Group variable: id                              Number of groups   =        37
                                                Obs per group: min =         3
                                                               avg =       4.7
                                                               max =         5
                                                Wald chi2(9)       =     45.85
Log likelihood = -296.58215                     Prob > chi2        =    0.0000
```

strength	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
2.program	1.360119	1.003549	1.36	0.175	-.6068016	3.32704
day						
4	1.125	.3322583	3.39	0.001	.4737858	1.776214
6	1.360127	.3766894	3.61	0.000	.6218298	2.098425
8	1.583563	.4905876	3.23	0.001	.6220287	2.545097
12	1.623576	.5372947	3.02	0.003	.5704977	2.676654
program#day						
2 4	-.169034	.4423472	-0.38	0.702	-1.036019	.6979505
2 6	.2113012	.4982385	0.42	0.671	-.7652283	1.187831
2 8	-.1299763	.6524813	-0.20	0.842	-1.408816	1.148864
2 12	.3212829	.7306782	0.44	0.660	-1.11082	1.753386
_cons	79.6875	.7560448	105.40	0.000	78.20568	81.16932

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
id: (empty)				
Residual: Unstructured				
sd(e0)	3.024179	.3515413	2.408024	3.797993
sd(e4)	3.445452	.4007049	2.743164	4.327536
sd(e6)	3.17265	.3701737	2.524102	3.987837
sd(e8)	3.636569	.42814	2.88721	4.580421
sd(e12)	3.628924	.4364031	2.866916	4.593468
corr(e0,e4)	.9237568	.0241659	.8594221	.9592936
corr(e0,e6)	.8847673	.0360256	.7902873	.9381525
corr(e0,e8)	.8438552	.0481853	.7193946	.915815
corr(e0,e12)	.8107881	.0591609	.6589166	.899148
corr(e4,e6)	.9598061	.0131155	.9242041	.9788692
corr(e4,e8)	.949579	.016828	.9036835	.9739048
corr(e4,e12)	.9024383	.0333957	.81189	.9505891
corr(e6,e8)	.957802	.0157897	.9127914	.9798265
corr(e6,e12)	.9120406	.0293324	.8329488	.9546129
corr(e8,e12)	.9403092	.0213539	.8808047	.9705727

```
LR test vs. linear regression:      chi2(14) =   314.67   Prob > chi2 = 0.0000
Note: The reported degrees of freedom assumes the null hypothesis is not on
      the boundary of the parameter space.  If this is not true, then the
      reported test is conservative.
```

Because we are using variable `id` only to group the repeated measurements and not to introduce random effects at the subject level, we use the `noconstant` option to omit any subject-level effects. The unstructured covariance matrix is the most general and contains many parameters. In this example,

we estimate a distinct residual standard error for each day and a distinct correlation for each pair of days.

That there is very high positive correlation between all pairs of measurements is evident, but what is not as evident is whether the pairwise correlation may be more parsimoniously represented. One option would be to explore whether the correlation diminishes as the time gap between strength measurements increases and whether it diminishes systematically. Given the irregularity of the time intervals, an exponential structure would be more appropriate than, say, an autoregressive or moving-average structure.

```
. estimates store unstructured

. xtmixed strength i.program##i.day || id:, noconstant
> residuals(exponential, t(day)) nolog nofetable
Mixed-effects ML regression                     Number of obs      =       173
Group variable: id                              Number of groups   =        37
                                                Obs per group: min =         3
                                                               avg =       4.7
                                                               max =         5
                                                Wald chi2(9)       =     36.77
Log likelihood = -307.83324                     Prob > chi2        =    0.0000
```

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
id: (empty)				
Residual: Exponential				
rho	.9786462	.0051238	.9659207	.9866854
sd(e)	3.350148	.3489952	2.73144	4.109001

```
LR test vs. linear regression:       chi2(1) =   292.17   Prob > chi2 = 0.0000

Note: The reported degrees of freedom assumes the null hypothesis is not on
      the boundary of the parameter space.  If this is not true, then the
      reported test is conservative.
```

In the above example, we suppressed displaying the main regression parameters because they did not differ much from those of the previous model. While the unstructured model estimated fifteen variance–covariance parameters, the exponential model claims to get the job done with just two, a fact that is not disputed by an LR test comparing the two nested models (at least not at the 0.01 level).

```
. lrtest unstructured .

Likelihood-ratio test                                 LR chi2(13) =     22.50
(Assumption: . nested in unstructured)                Prob > chi2 =    0.0481

Note: The reported degrees of freedom assumes the null hypothesis is not on
      the boundary of the parameter space.  If this is not true, then the
      reported test is conservative.
```

◁

Random-effects factor notation and crossed-effects models

Not all mixed models contain nested levels of random effects.

▷ Example 10

Returning to our longitudinal analysis of pig weights, suppose that instead of (5) we wish to fit

$$\texttt{weight}_{ij} = \beta_0 + \beta_1 \texttt{week}_{ij} + u_i + v_j + \epsilon_{ij} \tag{8}$$

for the $i = 1, \ldots, 9$ weeks and $j = 1, \ldots, 48$ pigs and

$$u_i \sim N(0, \sigma_u^2); \quad v_j \sim N(0, \sigma_v^2); \quad \epsilon_{ij} \sim N(0, \sigma_\epsilon^2)$$

all independently. Both (5) and (8) assume an overall population-average growth curve $\beta_0 + \beta_1 \texttt{week}$ and a random pig-specific shift.

The models differ in how `week` enters into the random part of the model. In (5), we assume that the effect due to `week` is linear and pig specific (a random slope); in (8), we assume that the effect due to `week`, u_i, is systematic to that week and common to all pigs. The rationale behind (8) could be that, assuming that the pigs were measured contemporaneously, we might be concerned that week-specific random factors such as weather and feeding patterns had significant systematic effects on all pigs.

Model (8) is an example of a two-way *crossed-effects* model, with the pig effects, v_j, being crossed with the week effects, u_i. One way to fit such models is to consider all the data as one big cluster and treat the u_i and v_j as a series of $9 + 48 = 57$ random coefficients on indicator variables for `week` and `pig`. In the notation of (2),

$$\mathbf{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_9 \\ v_1 \\ \vdots \\ v_{48} \end{bmatrix} \sim N(\mathbf{0}, \mathbf{G}); \quad \mathbf{G} = \begin{bmatrix} \sigma_u^2 \mathbf{I}_9 & \mathbf{0} \\ \mathbf{0} & \sigma_v^2 \mathbf{I}_{48} \end{bmatrix}$$

Because $\mathbf{G}$ is block diagonal, it can be represented in `xtmixed` as repeated-level equations. All we need is an ID variable to identify all the observations as one big group and a way to tell `xtmixed` to treat `week` and `pig` as factor variables (or equivalently, as two sets of overparameterized indicator variables identifying weeks and pigs, respectively). `xtmixed` supports the special group designation `_all` for the former and the factor notation `R.`*varname* for the latter.

```
. use http://www.stata-press.com/data/r12/pig, clear
(Longitudinal analysis of pig weights)
. xtmixed weight week || _all: R.week || _all: R.id
Performing EM optimization:
Performing gradient-based optimization:
Iteration 0:   log likelihood =  -1013.824
Iteration 1:   log likelihood =  -1013.824
Computing standard errors:
Mixed-effects ML regression                     Number of obs      =       432
Group variable: _all                            Number of groups   =         1
                                                Obs per group: min =       432
                                                               avg =     432.0
                                                               max =       432
                                                Wald chi2(1)       =  13258.28
Log likelihood =  -1013.824                     Prob > chi2        =    0.0000
```

weight	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
week	6.209896	.0539313	115.14	0.000	6.104192	6.315599
_cons	19.35561	.6333982	30.56	0.000	18.11418	20.59705

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
_all: Identity sd(R.week)	.2915259	.1490187	.107046	.7939333
_all: Identity sd(R.id)	3.851783	.4058045	3.133167	4.73522
sd(Residual)	2.073	.0756007	1.929997	2.226598

```
LR test vs. linear regression:       chi2(2) =   474.85   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
. estimates store crossed
```

Thus we estimate $\widehat{\sigma}_u = 0.29$ and $\widehat{\sigma}_v = 3.85$. Both (5) and (8) estimate a total of five parameters, two fixed effects and three variance components. The models, however, are not nested within each other, which precludes the use of an LR test to compare both models. Refitting model (5) and looking at the AIC values by using `estimates stats`,

```
. quietly xtmixed weight week || id:week
. estimates stats crossed .
```

Model	Obs	ll(null)	ll(model)	df	AIC	BIC
crossed	432	.	-1013.824	5	2037.648	2057.99
.	432	.	-869.0383	5	1748.077	1768.419

Note: N=Obs used in calculating BIC; see **[R] BIC note**

definitely favors model (5). This finding is not surprising, given that our rationale behind (8) was somewhat fictitious. In our `estimates stats` output, the values of `ll(null)` are missing. `xtmixed` does not fit a constant-only model as part of its usual estimation of the full model, but you can use `xtmixed` to fit a constant-only model directly, if you wish.

◁

The R.*varname* notation is equivalent to giving a list of overparameterized (none dropped) indicator variables for use in a random-effects specification. When you use R.*varname*, xtmixed handles the calculations internally rather than creating the indicators in the data. Because the set of indicators is overparameterized, R.*varname* implies noconstant. You can include factor variables in the fixed-effects specification by using standard methods; see [U] **11.4.3 Factor variables**. However, random-effects equations support only the R.*varname* factor specification. For more complex factor specifications (such as interactions) in random-effects equations, use generate to form the variables manually, as we demonstrated in example 6.

❑ Technical note

Although we were able to fit the crossed-effects model (8), it came at the expense of increasing the column dimension of our random-effects design from two in model (5) to 57 in model (8). Computation time and memory requirements grow (roughly) quadratically with the dimension of the random effects. As a result, fitting such crossed-effects models is feasible only when the total column dimension is small to moderate.

Reexamining model (8), we note that if we drop u_i, we end up with a model equivalent to (4), meaning that we could have fit (4) by typing

```
. xtmixed weight week || _all: R.id
```

instead of

```
. xtmixed weight week || id:
```

as we did when we originally fit the model. The results of both estimations are identical, but the latter specification, organized at the cluster (pig) level with random-effects dimension one (a random intercept) is much more computationally efficient. Whereas with the first form we are limited in how many pigs we can analyze, there is no such limitation with the second form.

Furthermore, we fit model (8) by using

```
. xtmixed weight week || _all: R.week || _all: R.id
```

as a direct way to demonstrate factor notation. However, we can technically treat pigs as nested within the "_all" group, yielding the equivalent and more efficient (total column dimension 10) way to fit (8):

```
. xtmixed weight week || _all: R.week || id:
```

We leave it to you to verify that both produce identical results. See Rabe-Hesketh and Skrondal (2008, chap. 11) for more techniques for making calculations more efficient in more complex models.

❑

▷ Example 11

As another example of how the same model may be fit in different ways by using xtmixed (and as a way to demonstrate covariance(exchangeable)), consider the three-level model used in example 4:

$$\mathbf{y}_{jk} = \mathbf{X}_{jk}\boldsymbol{\beta} + u_k^{(3)} + u_{jk}^{(2)} + \boldsymbol{\epsilon}_{jk}$$

where $\mathbf{y}_{jk}$ represents the logarithms of gross state products for the $n_{jk} = 17$ observations from state j in region k, $\mathbf{X}_{jk}$ is a set of regressors, $u_k^{(3)}$ is a random intercept at the region level, and $u_{jk}^{(2)}$ is a random intercept at the state (nested within region) level. We assume that $u_k^{(3)} \sim N(0, \sigma_3^2)$ and $u_{jk}^{(2)} \sim N(0, \sigma_2^2)$ independently. Define

$$\mathbf{v}_k = \begin{bmatrix} u_k^{(3)} + u_{1k}^{(2)} \\ u_k^{(3)} + u_{2k}^{(2)} \\ \vdots \\ u_k^{(3)} + u_{M_k,k}^{(2)} \end{bmatrix}$$

where M_k is the number of states in region k. Making this substitution, we can stack the observations for all the states within region k to get

$$\mathbf{y}_k = \mathbf{X}_k\boldsymbol{\beta} + \mathbf{Z}_k\mathbf{v}_k + \boldsymbol{\epsilon}_k$$

where $\mathbf{Z}_k$ is a set of indicators identifying the states within each region; that is,

$$\mathbf{Z}_k = \mathbf{I}_{M_k} \otimes \mathbf{J}_{17}$$

for a k-column vector of ones $\mathbf{J}_k$, and

$$\boldsymbol{\Sigma} = \mathrm{Var}(\mathbf{v}_k) = \begin{bmatrix} \sigma_3^2 + \sigma_2^2 & \sigma_3^2 & \cdots & \sigma_3^2 \\ \sigma_3^2 & \sigma_3^2 + \sigma_2^2 & \cdots & \sigma_3^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_3^2 & \sigma_3^2 & \sigma_3^2 & \sigma_3^2 + \sigma_2^2 \end{bmatrix}_{M_k \times M_k}$$

Because $\boldsymbol{\Sigma}$ is an exchangeable matrix, we can fit this alternative form of the model by specifying the `exchangeable` covariance structure.

```
. use http://www.stata-press.com/data/r12/productivity
(Public Capital Productivity)
. xtmixed gsp private emp hwy water other unemp || region: R.state,
> cov(exchangeable) variance
```

(*output omitted*)

```
Mixed-effects ML regression                     Number of obs      =       816
Group variable: region                          Number of groups   =         9
                                                Obs per group: min =        51
                                                               avg =      90.7
                                                               max =       136
                                                Wald chi2(6)       =  18829.06
Log likelihood =  1430.5017                     Prob > chi2        =    0.0000
```

gsp	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
private	.2671484	.0212591	12.57	0.000	.2254813	.3088154
emp	.7540721	.0261868	28.80	0.000	.7027468	.8053973
hwy	.0709767	.023041	3.08	0.002	.0258172	.1161363
water	.0761187	.0139248	5.47	0.000	.0488266	.1034109
other	-.0999955	.0169366	-5.90	0.000	-.1331907	-.0668004
unemp	-.0058983	.0009031	-6.53	0.000	-.0076684	-.0041282
_cons	2.128823	.1543855	13.79	0.000	1.826233	2.431413

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
region: Exchangeable				
var(R.state)	.0077263	.0017926	.0049032	.0121749
cov(R.state)	.0014506	.0012995	-.0010963	.0039975
var(Residual)	.0013461	.0000689	.0012176	.0014882

```
LR test vs. linear regression:       chi2(2) =  1154.73   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

The estimates of the fixed effects and their standard errors are equivalent to those from example 4, and remapping the variance components from $(\sigma_3^2 + \sigma_2^2, \sigma_3^2, \sigma_\epsilon^2)$, as displayed here, to $(\sigma_3, \sigma_2, \sigma_\epsilon)$, as displayed in example 4, will show that they are equivalent as well.

Of course, given the discussion in the previous technical note, it is more efficient to fit this model as we did originally, as a three-level model.

◁

Diagnosing convergence problems

Given the flexibility of the class of linear mixed models, you will find that some models "fail to converge" when used with your data. The default gradient-based method used by `xtmixed` is the Newton–Raphson algorithm, requiring the calculation of a gradient vector and Hessian (second-derivative) matrix; see [R] **ml**.

A failure to converge can take any one of three forms:

1. repeated "nonconcave" or "backed-up" iterations without convergence;
2. a Hessian (second-derivative) calculation that has become asymmetric, unstable, or has missing values; or

3. the message "standard-error calculation has failed" when computing standard errors.

All three situations essentially amount to the same thing: the Hessian calculation has become unstable, most likely because of a ridge in the likelihood function, a subsurface of the likelihood in which all points give the same value of the likelihood and for which there is no unique solution.

Such behavior is usually the result of either

A. a model that is not identified given the data, for example, fitting the three-level nested random intercept model

$$y_{jk} = \mathbf{x}_{jk}\boldsymbol{\beta} + u_k^{(3)} + u_{jk}^{(2)} + \epsilon_{jk}$$

without any replicated measurements at the (j, k) level, that is, with only one "i" per (j, k) combination. This model is unidentified for such data because the random intercepts $u_{jk}^{(2)}$ are confounded with the overall errors ϵ_{jk}; or

B. a model that contains a variance component whose estimate is really close to zero. When this occurs, a ridge is formed by an interval of values near zero, which produce the same likelihood and look equally good to the optimizer.

In unweighted models with independent and homoskedastic residuals, one useful way to diagnose problems of nonconvergence is to rely on the expectation-maximization (EM) algorithm (Dempster, Laird, and Rubin 1977), normally used by `xtmixed` only as a means of refining starting values. The advantages of EM are that it does not require a Hessian calculation, each successive EM iteration will result in a larger likelihood, iterations can be calculated quickly, and iterations will quickly bring parameter estimates into a neighborhood of the solution. The disadvantages of EM are that, once in a neighborhood of the solution, it can be slow to converge, if at all, and EM provides no facility for estimating standard errors of the estimated variance components. One useful property of EM is that it is always willing to provide a solution if you allow it to iterate enough times, if you are satisfied with being in a neighborhood of the optimum rather than right on the optimum, and if standard errors of variance components are not crucial to your analysis. If you encounter a nonconvergent model, try using the `emonly` option to bypass gradient-based optimization. Use `emiterate(#)` to specify the maximum number of EM iterations, which you will usually want to set much higher than the default of 20. If your EM solution shows an estimated variance component that is near zero, this provides evidence that reason B is the cause of the nonconvergence of the gradient-based method, in which case the solution would be to drop the offending variance component from the model. If no estimated variance components are near zero, reason A could be the culprit.

If your data and model are nearly unidentified, as opposed to fully unidentified, you may be able to obtain convergence with standard errors by changing some of the settings of the gradient-based optimization. Adding the `difficult` option can be particularly helpful if you are seeing many "nonconcave" messages; you may also consider changing the `technique()` or using the `nonrtolerance` option; see [R] **maximize**.

Distribution theory for likelihood-ratio tests

When determining the asymptotic distribution of a likelihood-ratio (LR) test comparing two nested models fit by `xtmixed`, issues concerning boundary problems imposed by estimating strictly positive quantities (that is, variances) can complicate the situation. When performing LR tests involving mixed models (whether comparing with linear regression within `xtmixed` or comparing two separate mixed models with `lrtest`), you may thus sometimes see a test labeled as "`chibar`" rather than the usual "`chi2`", or you may see a `chi2` test with a note attached stating that the test is conservative or possibly conservative depending on the hypothesis being tested.

At the heart of the issue is the number of variances being restricted to zero in the reduced model. If there are none, the usual asymptotic theory holds, and the distribution of the test statistic is χ^2 with degrees of freedom equal to the difference in the number of estimated parameters between both models.

When there is only one variance being set to zero in the reduced model, the asymptotic distribution of the LR test statistic is a 50:50 mixture of a χ^2_p and a χ^2_{p+1} distribution, where p is the number of other restricted parameters in the reduced model that are unaffected by boundary conditions. Stata labels such test statistics as `chibar` and adjusts the significance levels accordingly. See Self and Liang (1987) for the appropriate theory or Gutierrez, Carter, and Drukker (2001) for a Stata-specific discussion.

When more than one variance parameter is being set to zero in the reduced model, however, the situation becomes more complicated. For example, consider a comparison test versus linear regression for a mixed model with two random coefficients and unstructured covariance matrix

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_0^2 & \sigma_{01} \\ \sigma_{01} & \sigma_1^2 \end{bmatrix}$$

Because the random component of the mixed model comprises three parameters $(\sigma_0^2, \sigma_{01}, \sigma_1^2)$, on the surface it would seem that the LR comparison test would be distributed as χ^2_3. However, two complications need to be considered. First, the variances σ_0^2 and σ_1^2 are restricted to be positive, and second, constraints such as $\sigma_1^2 = 0$ implicitly restrict the covariance σ_{01} to be zero as well. From a technical standpoint, it is unclear how many parameters must be restricted to reduce the model to linear regression.

Because of these complications, appropriate and sufficiently general distribution theory for the more-than-one-variance case has yet to be developed. Theory (for example, Stram and Lee [1994]) and empirical studies (for example, McLachlan and Basford [1988]) have demonstrated that, whatever the distribution of the LR test statistic, its tail probabilities are bounded above by those of the χ^2 distribution with degrees of freedom equal to the full number of restricted parameters (three in the above example).

`xtmixed` uses this reference distribution, the χ^2 with full degrees of freedom, to produce a conservative test and places a note in the output labeling the test as such. Because the displayed significance level is an upper bound, rejection of the null hypothesis based on the reported level would imply rejection on the basis of the actual level.

❑ Technical note

It may seem that `xtmixed` does not follow Stata's standard syntax for multiple-equation models, but it does. In example 2, we typed

```
. xtmixed weight week || id:
```

but we could have used the standard multiequation syntax:

```
. xtmixed (weight week) (id:)
```

`xtmixed` will understand either and produce the same results. We prefer the syntax using `||` because it better emphasizes the nested structure of the levels.

❑

Survey data

Multilevel modeling of survey data is a little different than standard modeling in that weighted sampling can take place at multiple levels in the model, resulting in multiple sampling weights. Most survey datasets, regardless of the design, contain one overall inclusion weight for each observation in the data. This weight reflects the inverse of the probability of ultimate selection, and by "ultimate" we mean that it factors in all levels of clustered sampling, corrections for noninclusion and oversampling, poststratification, etc.

For simplicity, in what follows assume a simple two-stage sampling design where groups are randomly sampled and then individuals within groups are sampled. Also assume that no additional weight corrections are performed; that is, sampling weights are simply the inverse of the probability of selection. The sampling weight for observation i in cluster j in our two-level sample is then $w_{ij} = 1/\pi_{ij}$ where π_{ij} is the probability that observation i, j is selected. If you were performing a standard analysis such as OLS regression with `regress`, you would simply use a variable holding w_{ij} as your `pweight` variable, and the fact that it came from two levels of sampling would not concern you. Perhaps you would type `vce(cluster` *groupvar*`)` where *groupvar* identifies the top-level groups to get standard errors that control for correlation within these groups, but you would still use only a single weight variable.

Now take these same data and fit a two-level model with `xtmixed`. As seen in (14) in *Methods and formulas* later in this entry, it is not sufficient to use the single sampling weight w_{ij}, because weights enter into the log likelihood at both the group level and the individual level. Instead, what is required for a two-level model under this sampling design is w_j, the inverse of the probability that group j is selected in the first stage, and $w_{i|j}$, the inverse of the probability that individual i from group j is selected at the second stage *conditional on group j already being selected*. It simply will not do to just use w_{ij} without making any assumptions about w_j.

Given the rules of conditional probability, $w_{ij} = w_j w_{i|j}$. If your dataset has only w_{ij}, then you will need to either assume equal probability sampling at the first stage ($w_j = 1$ for all j) or find some way to recover w_j from other variables in your data; see Rabe-Hesketh and Skrondal (2006) and the references therein for some suggestions on how to do this, but realize that there is little yet known about how well these approximations perform in practice.

What you really need to fit your two-level model are data that contain w_j in addition to either w_{ij} or $w_{i|j}$. If you have w_{ij}—that is, the unconditional inclusion weight for observation i, j—then you need to either divide w_{ij} by w_j to obtain $w_{i|j}$ or rescale w_{ij} so that its dependence on w_j disappears. If you already have $w_{i|j}$, then rescaling becomes optional (but still an important decision to make!).

Weight rescaling is not an exact science, because the scale of the level-one weights is at issue regardless of whether they represent w_{ij} or $w_{i|j}$. The reason it is an issue is that because w_{ij} is unique to group j, the group-to-group magnitudes of these weights need to be normalized so that they are "consistent" from group to group. This is in stark contrast to a standard analysis, where the scale of sampling weights does not factor into estimation, instead only affecting the estimate of the total population size.

`xtmixed` offers three methods for standardizing weights in a two-level model, and you can specify which method you want via the `pwscale()` option. If you specify `pwscale(size)`, then the $w_{i|j}$ (or w_{ij}, it does not matter) are scaled to sum to the cluster size n_j. Method `pwscale(effective)` adds in a dependence on the sum of the squared weights so that level-one weights sum to the "effective" sample size. Just like `pwscale(size)`, `pwscale(effective)` also behaves the same whether you have $w_{i|j}$ or w_{ij}, and so it can be used with either.

Although both `pwscale(size)` and `pwscale(effective)` leave w_j untouched, the `pwscale(gk)` method is a little different in that 1) it changes the weights at both levels and 2) it does assume you have $w_{i|j}$ for level-one weights and not w_{ij} (if you have the latter, then first divide by w_j). Using the method of Graubard and Korn (1996), it sets the weights at the group level (level two) to the cluster averages of the products of both level weights (this product being w_{ij}). It then sets the individual weights to one everywhere; see *Methods and formulas* for the computational details of all three methods.

Determining which method is "best" is a tough call and depends on cluster size (the smaller the clusters, the greater the sensitivity to scale), whether the sampling is informative (that is, the sampling weights are correlated with the residuals), whether you are interested primarily in regression coefficients or in variance components, whether you have a simple random-intercept model or a more complex random-coefficients model, and other factors; see Rabe-Hesketh and Skrondal (2006), Carle (2009), and Pfeffermann et al. (1998) for some detailed advice. At the very least, you want to compare estimates across all three scaling methods (four, if you add no scaling) and perform a sensitivity analysis.

If you choose to rescale level-one weights, it does not matter if you have $w_{i|j}$ or w_{ij}. For the `pwscale(size)` and `pwscale(effective)` methods, you get identical results, and even though `pwscale(gk)` assumes $w_{i|j}$, you can obtain this as $w_{i|j} = w_{ij}/w_j$ before proceeding.

If you do not specify `pwscale()`, then no scaling takes place, and thus at a minimum, you need to make sure you have $w_{i|j}$ in your data and not w_{ij}.

▷ Example 12

Rabe-Hesketh and Skrondal (2006) analyzed their data from the 2000 Programme for International Student Assessment (PISA) study on reading proficiency among 15-year-old American students, as performed by the Organisation for Economic Co-operation and Development (OECD). The original study was a three-stage cluster sample where geographic areas were sampled at the first stage, schools at the second, and students at the third. Our version of the data does not contain the geographic-areas variable, so we treat this as a two-stage sample where schools are sampled at the first stage and students at the second.

```
. use http://www.stata-press.com/data/r12/pisa2000
(Programme for International Student Assessment (PISA) 2000 data)
. describe
Contains data from http://www.stata-press.com/data/r12/pisa2000.dta
  obs:         2,069                          Programme for International Student
                                                Assessment (PISA) 2000 data
 vars:            11                          12 Jun 2011 10:08
 size:        37,242                          (_dta has notes)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
female          byte   %8.0g                  1 if female
isei            byte   %8.0g                  International socio-economic
                                                index
w_fstuwt        float  %9.0g                  Student-level weight
wnrschbw        float  %9.0g                  School-level weight
high_school     byte   %8.0g                  1 if highest level by either
                                                parent is high school
college         byte   %8.0g                  1 if highest level by either
                                                parent is college
one_for         byte   %8.0g                  1 if one parent foreign born
both_for        byte   %8.0g                  1 if both parents are foreign
                                                born
test_lang       byte   %8.0g                  1 if English (the test language)
                                                is spoken at home
pass_read       byte   %8.0g                  1 if passed reading proficiency
                                                threshold
id_school       int    %8.0g                  School ID
-------------------------------------------------------------------------------
Sorted by:
```

For student i in school j, where variable `id_school` identifies the schools, variable `w_fstuwt` is a student-level overall inclusion weight (w_{ij}, not $w_{i|j}$) adjusted for noninclusion and nonparticipation of students, and variable `wnrschbw` is the school-level weight w_j adjusted for oversampling of schools with more minority students. The weight adjustments do not interfere with the methods prescribed above, and thus we can treat the weight variables simply as w_{ij} and w_j, respectively.

Rabe-Hesketh and Skrondal (2006) fit a two-level logistic model for passing a reading proficiency threshold. We fit a two-level linear random-intercept model for socioeconomic index. Because we have w_{ij} and not $w_{i|j}$, we rescale using `pwscale(size)` and thus obtain results as if we had $w_{i|j}$.

```
. xtmixed isei female high_school college one_for both_for test_lang
> [pw=w_fstuwt] || id_school:, pweight(wnrschbw) pwscale(size)
```

(*output omitted*)

```
Mixed-effects regression                        Number of obs      =      2069
Group variable: id_school                       Number of groups   =       148
                                                Obs per group: min =         1
                                                               avg =      14.0
                                                               max =        28
                                                Wald chi2(6)       =    187.23
Log pseudolikelihood = -1443093.9               Prob > chi2        =    0.0000
                          (Std. Err. adjusted for 148 clusters in id_school)
```

isei	Coef.	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
female	.59379	.8732886	0.68	0.497	-1.117824	2.305404
high_school	6.410618	1.500337	4.27	0.000	3.470011	9.351224
college	19.39494	2.121145	9.14	0.000	15.23757	23.55231
one_for	-.9584613	1.789947	-0.54	0.592	-4.466692	2.54977
both_for	-.2021101	2.32633	-0.09	0.931	-4.761633	4.357413
test_lang	2.519539	2.393165	1.05	0.292	-2.170978	7.210056
_cons	28.10788	2.435712	11.54	0.000	23.33397	32.88179

Random-effects Parameters	Estimate	Robust Std. Err.	[95% Conf.	Interval]
id_school: Identity				
sd(_cons)	5.890139	.7279	4.623113	7.50441
sd(Residual)	14.7898	.3793531	14.06466	15.55232

Some notes:

1. We specified the level-one weights using standard Stata weight syntax, that is, `[pw=w_fstuwt]`.
2. We specified the level-two weights via the `pweight(wnrschbw)` option as part of the random-effects specification for the `id_school` level. As such, it is treated as a school-level weight. Accordingly, `wnrschbw` needs to be constant within schools, and `xtmixed` did check for that before estimating.
3. Because our level-one weights are unconditional, we specified `pwscale(size)` to rescale them.
4. As is the case with other estimation commands in Stata, standard errors in the presence of sampling weights are robust.
5. Robust standard errors are clustered at the top level of the model, and this will always be true unless you specify `vce(cluster` *clustvar*`)`, where *clustvar* identifies an even higher level of grouping.

As a form of sensitivity analysis, we compare the above with scaling via `pwscale(gk)`. Because `pwscale(gk)` assumes $w_{i|j}$, you want to first divide w_{ij} by w_j. But you can handle that within the weight specification itself.

```
. xtmixed isei female high_school college one_for both_for test_lang
> [pw=w_fstuwt/wnrschbw] || id_school:, pweight(wnrschbw) pwscale(gk)
  (output omitted)
Mixed-effects regression                        Number of obs      =      2069
Group variable: id_school                       Number of groups   =       148

                                                Obs per group: min =         1
                                                               avg =      14.0
                                                               max =        28

                                                Wald chi2(6)       =    291.37
Log pseudolikelihood = -7270505.6               Prob > chi2        =    0.0000

                          (Std. Err. adjusted for 148 clusters in id_school)
```

isei	Coef.	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
female	-.3519458	.7436334	-0.47	0.636	-1.80944	1.105549
high_school	7.074911	1.139777	6.21	0.000	4.84099	9.308833
college	19.27285	1.286029	14.99	0.000	16.75228	21.79342
one_for	-.9142879	1.783091	-0.51	0.608	-4.409082	2.580506
both_for	1.214151	1.611885	0.75	0.451	-1.945085	4.373388
test_lang	2.661866	1.556491	1.71	0.087	-.3887996	5.712532
_cons	31.20145	1.907413	16.36	0.000	27.46299	34.93991

Random-effects Parameters	Estimate	Robust Std. Err.	[95% Conf.	Interval]
id_school: Identity				
sd(_cons)	5.628074	.6034248	4.561384	6.944213
sd(Residual)	15.04137	.2709432	14.5196	15.5819

The results are somewhat similar to before, which is good news from a sensitivity standpoint. Note that we specified `[pw=w_fstwtw/wnrschbw]` and thus did the conversion from w_{ij} to $w_{i|j}$ within our call to `xtmixed`.

◁

We close this section with a bit of bad news. Although weight rescaling and the issues that arise have been well studied for two-level models, as pointed out by Carle (2009), "... a best practice for scaling weights across multiple levels has yet to be advanced." As such, `pwscale()` is currently supported only for two-level models. If you are fitting a higher-level model with survey data, you need to make sure your sampling weights are conditional on selection at the previous stage and not overall inclusion weights, because there is currently no rescaling option to fall back on if you do not.

Saved results

xtmixed saves the following in e():

Scalars

e(N)	number of observations
e(k)	number of parameters
e(k_f)	number of FE parameters
e(k_r)	number of RE parameters
e(k_rs)	number of standard deviations
e(k_rc)	number of correlations
e(k_res)	number of residual-error parameters
e(N_clust)	number of clusters
e(nrgroups)	number of residual-error by() groups
e(ar_p)	AR order of residual errors, if specified
e(ma_q)	MA order of residual errors, if specified
e(res_order)	order of residual-error structure, if appropriate
e(df_m)	model degrees of freedom
e(ll)	log (restricted) likelihood
e(chi2)	χ^2
e(p)	p-value for χ^2
e(ll_c)	log likelihood, comparison model
e(chi2_c)	χ^2, comparison model
e(df_c)	degrees of freedom, comparison model
e(p_c)	p-value, comparison model
e(rank)	rank of e(V)
e(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros

e(cmd)	xtmixed
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(wtype)	weight type (first-level weights)
e(wexp)	weight expression (first-level weights)
e(fweight$k$)	fweight expression for kth highest level, if specified
e(pweight$k$)	pweight expression for kth highest level, if specified
e(ivars)	grouping variables
e(title)	title in estimation output
e(redim)	random-effects dimensions
e(vartypes)	variance-structure types
e(revars)	random-effects covariates
e(resopt)	residuals() specification, as typed
e(rstructure)	residual-error structure
e(rstructlab)	residual-error structure output label
e(rbyvar)	residual-error by() variable, if specified
e(rglabels)	residual-error by() groups labels
e(pwscale)	sampling-weight scaling method
e(timevar)	residual-error t() variable, if specified
e(chi2type)	Wald; type of model χ^2 test
e(clustvar)	name of cluster variable
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. Err.
e(method)	ML or REML
e(opt)	type of optimization
e(optmetric)	matsqrt or matlog; random-effects matrix parameterization
e(emonly)	emonly, if specified
e(ml_method)	type of ml method
e(technique)	maximization technique
e(properties)	b V
e(estat_cmd)	program used to implement estat
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(N_g)	group counts
e(g_min)	group-size minimums
e(g_avg)	group-size averages
e(g_max)	group-size maximums
e(tmap)	ID mapping for unstructured residual errors
e(V)	variance–covariance matrix of the estimator
e(V_modelbased)	model-based variance

Functions

e(sample)	marks estimation sample

Methods and formulas

xtmixed is implemented as an ado-file that uses Mata.

As given by (1), in the absence of weights we have the linear mixed model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + \boldsymbol{\epsilon}$$

where $\mathbf{y}$ is the $n \times 1$ vector of responses, $\mathbf{X}$ is an $n \times p$ design/covariate matrix for the fixed effects $\boldsymbol{\beta}$, and $\mathbf{Z}$ is the $n \times q$ design/covariate matrix for the random effects $\mathbf{u}$. The $n \times 1$ vector of errors, $\boldsymbol{\epsilon}$, is for now assumed to be multivariate normal with mean zero and variance matrix $\sigma_\epsilon^2\mathbf{I}_n$. We also assume that $\mathbf{u}$ has variance–covariance matrix $\mathbf{G}$ and that $\mathbf{u}$ is orthogonal to $\boldsymbol{\epsilon}$ so that

$$\mathrm{Var}\begin{bmatrix}\mathbf{u}\\ \boldsymbol{\epsilon}\end{bmatrix} = \begin{bmatrix}\mathbf{G} & \mathbf{0}\\ \mathbf{0} & \sigma_\epsilon^2\mathbf{I}_n\end{bmatrix}$$

Considering the combined error term $\mathbf{Z}\mathbf{u} + \boldsymbol{\epsilon}$, we see that $\mathbf{y}$ is multivariate normal with mean $\mathbf{X}\boldsymbol{\beta}$ and $n \times n$ variance–covariance matrix

$$\mathbf{V} = \mathbf{Z}\mathbf{G}\mathbf{Z}' + \sigma_\epsilon^2\mathbf{I}_n$$

Defining $\boldsymbol{\theta}$ as the vector of unique elements of $\mathbf{G}$ results in the log likelihood

$$L(\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma_\epsilon^2) = -\frac{1}{2}\left\{n\log(2\pi) + \log|\mathbf{V}| + (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'\mathbf{V}^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\right\} \tag{9}$$

which is maximized as a function of $\boldsymbol{\beta}$, $\boldsymbol{\theta}$, and σ_ϵ^2. As explained in chapter 6 of Searle, Casella, and McCulloch (1992), considering instead the likelihood of a set of linear contrasts, $\mathbf{K}\mathbf{y}$, that do not depend on $\boldsymbol{\beta}$ results in the restricted log likelihood

$$L_R(\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma_\epsilon^2) = L(\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma_\epsilon^2) - \frac{1}{2}\log\left|\mathbf{X}'\mathbf{V}^{-1}\mathbf{X}\right| \tag{10}$$

Given the high dimension of $\mathbf{V}$, however, the log-likelihood and restricted log-likelihood criteria are not usually computed by brute-force application of the above expressions. Instead, you can simplify the problem by subdividing the data into independent clusters (and subclusters if possible) and using matrix decomposition methods on the smaller matrices that result from treating each cluster one at a time.

Consider the two-level model described previously in (2)

$$\mathbf{y}_j = \mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j + \boldsymbol{\epsilon}_j$$

for $j = 1, \ldots, M$ clusters with cluster j containing n_j observations, with $\mathrm{Var}(\mathbf{u}_j) = \boldsymbol{\Sigma}$, a $q \times q$ matrix.

Efficient methods for computing (9) and (10) are given in chapter 2 of Pinheiro and Bates (2000). Namely, for the two-level model, define $\mathbf{\Delta}$ to be the Cholesky factor of $\sigma_\epsilon^2\mathbf{\Sigma}^{-1}$, such that $\sigma_\epsilon^2\mathbf{\Sigma}^{-1} = \mathbf{\Delta}'\mathbf{\Delta}$. For $j = 1, \ldots, M$, decompose

$$\begin{bmatrix} \mathbf{Z}_j \\ \mathbf{\Delta} \end{bmatrix} = \mathbf{Q}_j \begin{bmatrix} \mathbf{R}_{11j} \\ \mathbf{0} \end{bmatrix}$$

using an orthogonal-triangular (QR) decomposition, with $\mathbf{Q}_j$ a $(n_j + q)$-square matrix and $\mathbf{R}_{11j}$ a q-square matrix. We then apply $\mathbf{Q}_j$ as follows:

$$\begin{bmatrix} \mathbf{R}_{10j} \\ \mathbf{R}_{00j} \end{bmatrix} = \mathbf{Q}_j' \begin{bmatrix} \mathbf{X}_j \\ \mathbf{0} \end{bmatrix}; \qquad \begin{bmatrix} \mathbf{c}_{1j} \\ \mathbf{c}_{0j} \end{bmatrix} = \mathbf{Q}_j' \begin{bmatrix} \mathbf{y}_j \\ \mathbf{0} \end{bmatrix}$$

Stack the $\mathbf{R}_{00j}$ and $\mathbf{c}_{0j}$ matrices, and perform the additional QR decomposition

$$\begin{bmatrix} \mathbf{R}_{001} & \mathbf{c}_{01} \\ \vdots & \vdots \\ \mathbf{R}_{00M} & \mathbf{c}_{0M} \end{bmatrix} = \mathbf{Q}_0 \begin{bmatrix} \mathbf{R}_{00} & \mathbf{c}_0 \\ \mathbf{0} & \mathbf{c}_1 \end{bmatrix}$$

Pinheiro and Bates (2000) show that ML estimates of $\boldsymbol{\beta}$, σ_ϵ^2, and $\mathbf{\Delta}$ (the unique elements of $\mathbf{\Delta}$, that is) are obtained by maximizing the profile log likelihood (profiled in $\mathbf{\Delta}$)

$$L(\mathbf{\Delta}) = \frac{n}{2}\{\log n - \log(2\pi) - 1\} - n\log||\mathbf{c}_1|| + \sum_{j=1}^{M} \log\left|\frac{\det(\mathbf{\Delta})}{\det(\mathbf{R}_{11j})}\right| \tag{11}$$

where $||\cdot||$ denotes the 2-norm, and following this maximization with

$$\widehat{\boldsymbol{\beta}} = \mathbf{R}_{00}^{-1}\mathbf{c}_0; \quad \widehat{\sigma}_\epsilon^2 = n^{-1}||\mathbf{c}_1||^2 \tag{12}$$

REML estimates are obtained by maximizing

$$\begin{aligned} L_R(\mathbf{\Delta}) = \frac{n-p}{2}\{\log(n-p) - \log(2\pi) - 1\} - (n-p)\log||\mathbf{c}_1|| \\ - \log|\det(\mathbf{R}_{00})| + \sum_{j=1}^{M} \log\left|\frac{\det(\mathbf{\Delta})}{\det(\mathbf{R}_{11j})}\right| \end{aligned} \tag{13}$$

followed by

$$\widehat{\boldsymbol{\beta}} = \mathbf{R}_{00}^{-1}\mathbf{c}_0; \quad \widehat{\sigma}_\epsilon^2 = (n-p)^{-1}||\mathbf{c}_1||^2$$

For numerical stability, maximization of (11) and (13) is not performed with respect to the unique elements of $\mathbf{\Delta}$ but instead with respect to the unique elements of the matrix square root (or matrix logarithm if the `matlog` option is specified) of $\mathbf{\Sigma}/\sigma_\epsilon^2$; define $\boldsymbol{\gamma}$ to be the vector containing these elements.

Once maximization with respect to $\boldsymbol{\gamma}$ is completed, $(\boldsymbol{\gamma}, \sigma_\epsilon^2)$ is reparameterized to $\{\boldsymbol{\alpha}, \log(\sigma_\epsilon)\}$, where $\boldsymbol{\alpha}$ is a vector containing the unique elements of $\mathbf{\Sigma}$, expressed as logarithms of standard deviations for the diagonal elements and hyperbolic arctangents of the correlations for off-diagonal elements. This last step is necessary to (a) obtain a joint variance–covariance estimate of the elements of $\mathbf{\Sigma}$ and σ_ϵ^2; (b) obtain a parameterization under which parameter estimates can be interpreted individually, rather than as elements of a matrix square root (or logarithm); and (c) parameterize these elements such that their ranges each encompass the entire real line.

Obtaining a joint variance–covariance matrix for the estimated $\{\boldsymbol{\alpha}, \log(\sigma_\epsilon)\}$ requires the evaluation of the log likelihood (or log-restricted likelihood) with only $\boldsymbol{\beta}$ profiled out. For ML, we have

$$\begin{aligned} L^*\{\boldsymbol{\alpha}, \log(\sigma_\epsilon)\} &= L\{\boldsymbol{\Delta}(\boldsymbol{\alpha}, \sigma_\epsilon^2), \sigma_\epsilon^2\} \\ &= -\frac{n}{2}\log(2\pi\sigma_\epsilon^2) - \frac{||\mathbf{c}_1||^2}{2\sigma_\epsilon^2} + \sum_{j=1}^{M} \log\left|\frac{\det(\boldsymbol{\Delta})}{\det(\mathbf{R}_{11j})}\right| \end{aligned}$$

with the analogous expression for REML.

The variance–covariance matrix of $\widehat{\boldsymbol{\beta}}$ is estimated as

$$\widehat{\mathrm{Var}}(\widehat{\boldsymbol{\beta}}) = \widehat{\sigma}_\epsilon^2 \mathbf{R}_{00}^{-1} \left(\mathbf{R}_{00}^{-1}\right)'$$

but this does not mean that $\widehat{\mathrm{Var}}(\widehat{\boldsymbol{\beta}})$ is identical under both ML and REML because $\mathbf{R}_{00}$ depends on $\boldsymbol{\Delta}$. Because $\widehat{\boldsymbol{\beta}}$ is asymptotically uncorrelated with $\{\widehat{\boldsymbol{\alpha}}, \log(\widehat{\sigma}_\epsilon)\}$, the covariance of $\widehat{\boldsymbol{\beta}}$ with the other estimated parameters is treated as zero.

Parameter estimates are stored in `e(b)` as $\{\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\alpha}}, \log(\widehat{\sigma}_\epsilon)\}$, with the corresponding (block-diagonal) variance–covariance matrix stored in `e(V)`. Parameter estimates can be displayed in this metric by specifying the `estmetric` option. However, in `xtmixed` output, variance components are most often displayed either as variances and covariances or as standard deviations and correlations.

EM iterations are derived by considering the $\mathbf{u}_j$ in (2) as missing data. Here we describe the procedure for maximizing the log likelihood via EM; the procedure for maximizing the restricted log likelihood is similar. The log likelihood for the full data $(\mathbf{y}, \mathbf{u})$ is

$$L_F(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \sigma_\epsilon^2) = \sum_{j=1}^{M} \left\{\log f_1(\mathbf{y}_j|\mathbf{u}_j, \boldsymbol{\beta}, \sigma_\epsilon^2) + \log f_2(\mathbf{u}_j|\boldsymbol{\Sigma})\right\}$$

where $f_1()$ is the density function for multivariate normal with mean $\mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j$ and variance $\sigma_\epsilon^2\mathbf{I}_{n_j}$, and $f_2()$ is the density for multivariate normal with mean $\mathbf{0}$ and $q \times q$ covariance matrix $\boldsymbol{\Sigma}$. As before, we can profile $\boldsymbol{\beta}$ and σ_ϵ^2 out of the optimization, yielding the following EM iterative procedure:

1. For the current iterated value of $\boldsymbol{\Sigma}^{(t)}$, fix $\widehat{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}(\boldsymbol{\Sigma}^{(t)})$ and $\widehat{\sigma}_\epsilon^2 = \widehat{\sigma}_\epsilon^2(\boldsymbol{\Sigma}^{(t)})$ according to (12).
2. Expectation step: Calculate

$$\begin{aligned} D(\boldsymbol{\Sigma}) &\equiv E\left\{L_F(\widehat{\boldsymbol{\beta}}, \boldsymbol{\Sigma}, \widehat{\sigma}_\epsilon^2)|\mathbf{y}\right\} \\ &= C - \frac{M}{2}\log\det(\boldsymbol{\Sigma}) - \frac{1}{2}\sum_{j=1}^{M} E\left(\mathbf{u}_j'\boldsymbol{\Sigma}^{-1}\mathbf{u}_j|\mathbf{y}\right) \end{aligned}$$

   where C is a constant that does not depend on $\boldsymbol{\Sigma}$, and the expected value of the quadratic form $\mathbf{u}_j'\boldsymbol{\Sigma}^{-1}\mathbf{u}_j$ is taken with respect to the conditional density $f(\mathbf{u}_j|\mathbf{y}, \widehat{\boldsymbol{\beta}}, \boldsymbol{\Sigma}^{(t)}, \widehat{\sigma}_\epsilon^2)$.
3. Maximization step: Maximize $D(\boldsymbol{\Sigma})$ to produce $\boldsymbol{\Sigma}^{(t+1)}$.

For general, symmetric $\boldsymbol{\Sigma}$, the maximizer of $D(\boldsymbol{\Sigma})$ can be derived explicitly, making EM iterations quite fast.

For general residual-error structures,

$$\text{Var}(\boldsymbol{\epsilon}_j) = \sigma_\epsilon^2 \boldsymbol{\Lambda}_j$$

where the subscript j merely represents that $\boldsymbol{\epsilon}_j$ and $\boldsymbol{\Lambda}_j$ vary in dimension in unbalanced data, the data are first transformed according to

$$\mathbf{y}_j^* = \widehat{\boldsymbol{\Lambda}}_j^{-1/2}\mathbf{y}_j; \qquad \mathbf{X}_j^* = \widehat{\boldsymbol{\Lambda}}_j^{-1/2}\mathbf{X}_j; \qquad \mathbf{Z}_j^* = \widehat{\boldsymbol{\Lambda}}_j^{-1/2}\mathbf{Z}_j;$$

and the likelihood-evaluation techniques described above are applied to $\mathbf{y}_j^*$, $\mathbf{X}_j^*$, and $\mathbf{Z}_j^*$ instead. The unique elements of $\boldsymbol{\Lambda}$, $\boldsymbol{\rho}$, are estimated along with the fixed effects and variance components. Because σ_ϵ^2 is always estimated and multiplies the entire $\boldsymbol{\Lambda}_j$ matrix, $\widehat{\boldsymbol{\rho}}$ is parameterized to take this into account.

In the presence of sampling weights, following Rabe-Hesketh and Skrondal (2006), the weighted log pseudolikelihood for a two-level model is given as

$$L(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \sigma_\epsilon^2) = \sum_{j=1}^{M} w_j \log\left[\int \exp\left\{\sum_{i=1}^{n_j} w_{i|j} \log f_1(y_{ij}|\mathbf{u}_j, \boldsymbol{\beta}, \sigma_\epsilon^2)\right\} f_2(\mathbf{u}_j|\boldsymbol{\Sigma}) d\mathbf{u}_j\right] \tag{14}$$

where w_j is the inverse of the probability of selection for the jth cluster, $w_{i|j}$ is the inverse of the conditional probability of selection of individual i given the selection of cluster j, and $f_1()$ and $f_2()$ are the multivariate normal densities previously defined.

Weighted estimation is achieved through incorporating w_j and $w_{i|j}$ into the matrix decomposition methods detailed above so as to reflect replicated clusters for w_j and replicated observations within clusters for $w_{i|j}$. Because this estimation is based on replicated clusters and observations, frequency weights are handled similarly.

Rescaling of sampling weights can take one of three available forms:

Under `pwscale(size)`,

$$w_{i|j}^* = n_j w_{i|j}^* \left\{\sum_{i=1}^{n_j} w_{i|j}\right\}^{-1}$$

Under `pwscale(effective)`,

$$w_{i|j}^* = w_{i|j}^* \left\{\sum_{i=1}^{n_j} w_{i|j}\right\} \left\{\sum_{i=1}^{n_j} w_{i|j}^2\right\}^{-1}$$

Under both the above, w_j remains unchanged. For method `pwscale(gk)`, however, both weights are modified:

$$w_j^* = n_j^{-1} \sum_{i=1}^{n_j} w_{i|j} w_j; \quad w_{i|j}^* = 1$$

Under ML estimation, robust standard errors are obtained in the usual way (see [P] **_robust**) with the one distinction being that in multilevel models, robust variances are, at a minimum, clustered at the highest level. This is because given the form of the log likelihood, scores aggregate at the top-level clusters. For a two-level model, scores are obtained as the partial derivatives of $L_j(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \sigma_\epsilon^2)$ with respect to $\{\boldsymbol{\beta}, \boldsymbol{\alpha}, \log(\sigma_\epsilon)\}$, where L_j is the log likelihood for cluster j and $L = \sum_{j=1}^{M} L_j$. Robust variances are not supported under REML estimation because the form of the log restricted likelihood does not lend itself to separation by highest-level clusters.

EM iterations always assume equal weighting and an independent, homoskedastic error structure. As such, with weighted data or when error structures are more complex, EM is used only to obtain starting values.

For extensions to three-level models and higher, see Bates and Pinheiro (1998) and Rabe-Hesketh and Skrondal (2006).

Charles Roy Henderson (1911–1989) was born in Iowa and grew up on the family farm. His education in animal husbandry, animal nutrition, and statistics at Iowa State was interspersed with jobs in the Iowa Extension Service, Ohio University, and the U.S. Army. After completing his PhD, Henderson joined the Animal Science faculty at Cornell. He developed and applied statistical methods in the improvement of farm livestock productivity through genetic selection, with particular focus on dairy cattle. His methods are general and have been used worldwide in livestock breeding and beyond agriculture. Henderson's work on variance components and best linear unbiased predictions has proved to be one of the main roots of current mixed-model methods.

Acknowledgments

We thank Badi Baltagi, Department of Economics, Syracuse University, and Ray Carroll, Department of Statistics, Texas A&M University, for each providing us with a dataset used in this entry.

References

Andrews, M., T. Schank, and R. Upward. 2006. Practical fixed-effects estimation methods for the three-way error-components model. *Stata Journal* 6: 461–481.

Baltagi, B. H., S. H. Song, and B. C. Jung. 2001. The unbalanced nested error component regression model. *Journal of Econometrics* 101: 357–381.

Bates, D. M., and J. C. Pinheiro. 1998. Computational methods for multilevel modelling. In *Technical Memorandum BL0112140-980226-01TM*. Murray Hill, NJ: Bell Labs, Lucent Technologies. http://stat.bell-labs.com/NLME/CompMulti.pdf.

Cameron, A. C., and P. K. Trivedi. 2010. *Microeconometrics Using Stata*. Rev. ed. College Station, TX: Stata Press.

Canette, I. 2011. Including covariates in crossed-effects models. The Stata Blog: Not Elsewhere Classified. http://blog.stata.com/2010/12/22/including-covariates-in-crossed-effects-models/

Carle, A. C. 2009. Fitting multilevel models in complex survey data with design weights: Recommendations. *BMC Medical Research Methodology* 9: 49.

Demidenko, E. 2004. *Mixed Models: Theory and Applications*. Hoboken, NJ: Wiley.

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B* 39: 1–38.

Diggle, P. J., P. J. Heagerty, K.-Y. Liang, and S. L. Zeger. 2002. *Analysis of Longitudinal Data*. 2nd ed. Oxford: Oxford University Press.

Fitzmaurice, G. M., N. M. Laird, and J. H. Ware. 2004. *Applied Longitudinal Analysis*. Hoboken, NJ: Wiley.

Goldstein, H. 1986. Efficient statistical modelling of longitudinal data. *Annals of Human Biology* 13: 129–141.

Graubard, B. I., and E. L. Korn. 1996. Modelling the sampling design in the analysis of health surveys. *Statistical Methods in Medical Research* 5: 263–281.

Gutierrez, R. G., S. 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.

Harville, D. A. 1977. Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association* 72: 320–338.

Henderson, C. R. 1953. Estimation of variance and covariance components. *Biometrics* 9: 226–252.

Hocking, R. R. 1985. *The Analysis of Linear Models*. Monterey, CA: Brooks/Cole.

Horton, N. J. 2011. Stata tip 95: Estimation of error covariances in a linear model. *Stata Journal* 11: 145–148.

Laird, N. M., and J. H. Ware. 1982. Random-effects models for longitudinal data. *Biometrics* 38: 963–974.

LaMotte, L. R. 1973. Quadratic estimation of variance components. *Biometrics* 29: 311–330.

Marchenko, Y. V. 2006. Estimating variance components in Stata. *Stata Journal* 6: 1–21.

McCulloch, C. E., S. R. Searle, and J. M. Neuhaus. 2008. *Generalized, Linear, and Mixed Models*. 2nd ed. Hoboken, NJ: Wiley.

McLachlan, G. J., and K. E. Basford. 1988. *Mixture Models*. New York: Dekker.

Munnell, A. 1990. Why has productivity growth declined? Productivity and public investment. *New England Economic Review* Jan./Feb.: 3–22.

Nichols, A. 2007. Causal inference with observational data. *Stata Journal* 7: 507–541.

Pantazis, N., and G. Touloumi. 2010. Analyzing longitudinal data in the presence of informative drop-out: The jmre1 command. *Stata Journal* 10: 226–251.

Pfeffermann, D., C. J. Skinner, D. J. Holmes, H. Goldstein, and J. Rasbash. 1998. Weighting for unequal selection probabilities in multilevel models. *Journal of the Royal Statistical Society, Series B* 60: 23–40.

Pierson, R. A., and O. J. Ginther. 1987. Follicular population dynamics during the estrous cycle of the mare. *Animal Reproduction Science* 14: 219–231.

Pinheiro, J. C., and D. M. Bates. 2000. *Mixed-Effects Models in S and S-PLUS*. New York: Springer.

Prosser, R., J. Rasbash, and H. Goldstein. 1991. *ML3 Software for 3-Level Analysis: User's Guide for V. 2*. London: Institute of Education, University of London.

Rabe-Hesketh, S., and A. Skrondal. 2006. Multilevel modelling of complex survey data. *Journal of the Royal Statistical Society, Series A* 169: 805–827.

——. 2008. *Multilevel and Longitudinal Modeling Using Stata*. 2nd ed. College Station, TX: Stata Press.

Rao, C. R. 1973. *Linear Statistical Inference and Its Applications*. 2nd ed. New York: Wiley.

Raudenbush, S. W., and A. S. Bryk. 2002. *Hierarchical Linear Models: Applications and Data Analysis Methods*. 2nd ed. Thousand Oaks, CA: Sage.

Ruppert, D., M. P. Wand, and R. J. Carroll. 2003. *Semiparametric Regression*. Cambridge: Cambridge University Press.

Searle, S. R. 1989. Obituary: Charles Roy Henderson 1911–1989. *Biometrics* 45: 1333–1335.

Searle, S. R., G. Casella, and C. E. McCulloch. 1992. *Variance Components*. New York: Wiley.

Self, S. G., and K.-Y. Liang. 1987. Asymptotic properties of maximum likelihood estimators and likelihood ratio tests under nonstandard conditions. *Journal of the American Statistical Association* 82: 605–610.

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

Stram, D. O., and J. W. Lee. 1994. Variance components testing in the longitudinal mixed effects model. *Biometrics* 50: 1171–1177.

Thompson, W. A., Jr. 1962. The problem of negative estimates of variance components. *Annals of Mathematical Statistics* 33: 273–289.

Verbeke, G., and G. Molenberghs. 2000. *Linear Mixed Models for Longitudinal Data*. New York: Springer.

West, B. T., K. B. Welch, and A. T. Galecki. 2007. *Linear Mixed Models: A Practical Guide Using Statistical Software*. Boca Raton, FL: Chapman & Hall/CRC.

Also see

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

[XT] **xtmelogit** — Multilevel mixed-effects logistic regression

[XT] **xtmepoisson** — Multilevel mixed-effects Poisson regression

[XT] **xtreg** — Fixed-, between-, and random-effects and population-averaged linear models

[XT] **xtrc** — Random-coefficients model

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

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

[U] **20 Estimation and postestimation commands**

Title

xtmixed postestimation — Postestimation tools for xtmixed

The documentation for [XT] **xtmixed postestimation** has been updated. To see the latest PDF of [XT] **xtmixed postestimation**, click here.

Description

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

Command	Description
`estat group`	summarize the composition of the nested groups
`estat recovariance`	display the estimated random-effects covariance matrix (or matrices)

For information about these commands, see below.

The following standard postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat`	AIC, BIC, VCE, and estimation sample summary
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`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 coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

See the corresponding entries in the *Base Reference Manual* for details.

Special-interest postestimation commands

`estat group` reports number of groups and minimum, average, and maximum group sizes for each level of the model. Model levels are identified by the corresponding group variable in the data. Because groups are treated as nested, the information in this summary may differ from what you would get if you `tabulate`d each group variable individually.

`estat recovariance` displays the estimated variance–covariance matrix of the random effects for each level in the model. Random effects can be either random intercepts, in which case the corresponding rows and columns of the matrix are labeled as `_cons`, or random coefficients, in which case the label is the name of the associated variable in the data.

Syntax for predict

Syntax for obtaining best linear unbiased predictions (BLUPs) of random effects, or the BLUPs' standard errors

```
predict [type] {stub*|newvarlist} [if] [in], {reffects|reses}
  [level(levelvar)]
```

Syntax for obtaining scores after ML estimation

```
predict [type] {stub*|newvarlist} [if] [in], scores
```

Syntax for obtaining other predictions

```
predict [type] newvar [if] [in] [, statistic level(levelvar)]
```

statistic	Description
Main	
xb	linear prediction for the *fixed* portion of the model only; the default
stdp	standard error of the fixed-portion linear prediction
fitted	fitted values, fixed-portion linear prediction plus contributions based on predicted random effects
residuals	residuals, response minus fitted values
* rstandard	standardized residuals

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.

Menu

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

xb, the default, calculates the linear prediction $\mathbf{x}\beta$ based on the estimated fixed effects (coefficients) in the model. This is equivalent to fixing all random effects in the model to their theoretical mean value of zero.

stdp calculates the standard error of the linear predictor $\mathbf{x}\beta$.

level(*levelvar*) specifies the level in the model at which predictions involving random effects are to be obtained; see the options below for the specifics. *levelvar* is the name of the model level and is either the name of the variable describing the grouping at that level or _all, a special designation for a group comprising all the estimation data.

reffects calculates best linear unbiased predictions (BLUPs) of the random effects. By default, BLUPs for all random effects in the model are calculated. However, if the level(*levelvar*) option is specified, then BLUPs for only level *levelvar* in the model are calculated. For example, if classes are nested within schools, then typing

```
. predict b*, reffects level(school)
```

would produce BLUPs at the school level. You must specify q new variables, where q is the number of random-effects terms in the model (or level). However, it is much easier to just specify *stub*`*` and let Stata name the variables *stub*`1` ... *stubq* for you.

`reses` calculates the standard errors of the best linear unbiased predictions (BLUPs) of the random effects. By default, standard errors for all BLUPs in the model are calculated. However, if the `level(`*levelvar*`)` option is specified, then standard errors for only level *levelvar* in the model are calculated; see the `reffects` option. You must specify q new variables, where q is the number of random-effects terms in the model (or level). However, it is much easier to just specify *stub*`*` and let Stata name the variables *stub*`1` ... *stubq* for you.

The `reffects` and `reses` options often generate multiple new variables at once. When this occurs, the random effects (or standard errors) contained in the generated variables correspond to the order in which the variance components are listed in the output of `xtmixed`. Still, examining the variable labels of the generated variables (using the `describe` command, for instance) can be useful in deciphering which variables correspond to which terms in the model.

`scores` calculates the parameter-level scores, one for each parameter in the model including regression coefficients and variance components. The score for a parameter is the first derivative of the log likelihood (or log pseudolikelihood) with respect to that parameter. One score per highest-level group is calculated, and it is placed on the last record within that group. Scores are calculated in the estimation metric as stored in `e(b)`.

`scores` is not available after restricted maximum-likelihood (REML) estimation.

`fitted` calculates fitted values, which are equal to the fixed-portion linear predictor *plus* contributions based on predicted random effects, or in mixed-model notation, $\mathbf{x}\boldsymbol{\beta} + \mathbf{Zu}$. By default, the fitted values take into account random effects from all levels in the model; however, if the `level(`*levelvar*`)` option is specified, the fitted values are fit beginning with the topmost level down to and including level *levelvar*. For example, if `class`es are nested within `school`s, then typing

```
. predict yhat_school, fitted level(school)
```

would produce school-level predictions. That is, the predictions would incorporate school-specific random effects but not those for each class nested within each school.

`residuals` calculates residuals, equal to the responses minus fitted values. By default, the fitted values take into account random effects from all levels in the model; however, if the `level(`*levelvar*`)` option is specified, the fitted values are fit beginning at the topmost level down to and including level *levelvar*.

`rstandard` calculates standardized residuals, equal to the residuals multiplied by the inverse square root of the estimated error covariance matrix.

Syntax for estat group

```
estat group
```

Menu

Statistics > Postestimation > Reports and statistics

Syntax for estat recovariance

```
estat recovariance [ , level(levelvar) correlation matlist_options ]
```

Menu

Statistics > Postestimation > Reports and statistics

Options for estat recovariance

`level(`*levelvar*`)` specifies the level in the model for which the random-effects covariance matrix is to be displayed and returned in `r(cov)`. By default, the covariance matrices for all levels in the model are displayed. *levelvar* is the name of the model level and is either the name of variable describing the grouping at that level or `_all`, a special designation for a group comprising all the estimation data.

`correlation` displays the covariance matrix as a correlation matrix and returns the correlation matrix in `r(corr)`.

matlist_options are style and formatting options that control how the matrix (or matrices) are displayed; see [P] **matlist** for a list of what is available.

Remarks

Various predictions, statistics, and diagnostic measures are available after fitting a mixed model using `xtmixed`. For the most part, calculation centers around obtaining best linear unbiased predictors (BLUPs) of the random effects. Random effects are not estimated when the model is fit but instead need to be predicted after estimation.

▷ Example 1

In example 3 of [XT] **xtmixed**, we modeled the weights of 48 pigs measured on nine successive weeks as

$$\texttt{weight}_{ij} = \beta_0 + \beta_1 \texttt{week}_{ij} + u_{0j} + u_{1j}\texttt{week}_{ij} + \epsilon_{ij} \tag{1}$$

for $i = 1, \ldots, 9$, $j = 1, \ldots, 48$, $\epsilon_{ij} \sim N(0, \sigma^2_\epsilon)$, and u_{0j} and u_{1j} normally distributed with mean zero and variance–covariance matrix

$$\mathbf{\Sigma} = \text{Var}\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} = \begin{bmatrix} \sigma^2_{u0} & \sigma_{01} \\ \sigma_{01} & \sigma^2_{u1} \end{bmatrix}$$

```
. use http://www.stata-press.com/data/r12/pig
(Longitudinal analysis of pig weights)
. xtmixed weight week || id: week, covariance(unstructured) variance
  (output omitted)
Mixed-effects ML regression                     Number of obs      =       432
Group variable: id                              Number of groups   =        48
                                                Obs per group: min =         9
                                                               avg =       9.0
                                                               max =         9
                                                Wald chi2(1)       =   4649.17
Log likelihood = -868.96185                     Prob > chi2        =    0.0000
```

weight	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
week	6.209896	.0910745	68.18	0.000	6.031393	6.388399
_cons	19.35561	.3996387	48.43	0.000	18.57234	20.13889

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
id: Unstructured				
var(week)	.3715251	.0812958	.2419532	.570486
var(_cons)	6.823363	1.566194	4.351297	10.69986
cov(week,_cons)	-.0984378	.2545767	-.5973991	.4005234
var(Residual)	1.596829	.123198	1.372735	1.857505

```
LR test vs. linear regression:       chi2(3) =   764.58   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

Rather than see the estimated variance components listed as above, we can instead see them in matrix form; that is, we can see $\widehat{\Sigma}$

```
. estat recovariance
Random-effects covariance matrix for level id
```

	week	_cons
week	.3715251	
_cons	-.0984378	6.823363

or we can see $\widehat{\Sigma}$ as a correlation matrix

```
. estat recovariance, correlation
Random-effects correlation matrix for level id
```

	week	_cons
week	1	
_cons	-.0618257	1

We can also obtain BLUPs of the pig-level random effects (u_{0j} and u_{1j}). We need to specify the variables to be created in the order `u1 u0` because that is the order in which the corresponding variance components are listed in the output (`week _cons`). We obtain the predictions and list them for the first 10 pigs.

```
. predict u1 u0, reffects
. by id, sort: generate tolist = (_n==1)
. list id u0 u1 if id <=10 & tolist
```

	id	u0	u1
1.	1	.2369444	-.3957636
10.	2	-1.584127	.510038
19.	3	-3.526551	.3200372
28.	4	1.964378	-.7719702
37.	5	1.299236	-.9241479
46.	6	-1.147302	-.5448151
55.	7	-2.590529	.0394454
64.	8	-1.137067	-.1696566
73.	9	-3.189545	-.7365507
82.	10	1.160324	.0030772

If you forget how to order your variables in `predict`, or if you use `predict` *stub*`*`, remember that `predict` labels the generated variables for you to avoid confusion.

```
. describe u0 u1
```

variable name	storage type	display format	value label	variable label
u0	float	%9.0g		BLUP r.e. for id: _cons
u1	float	%9.0g		BLUP r.e. for id: week

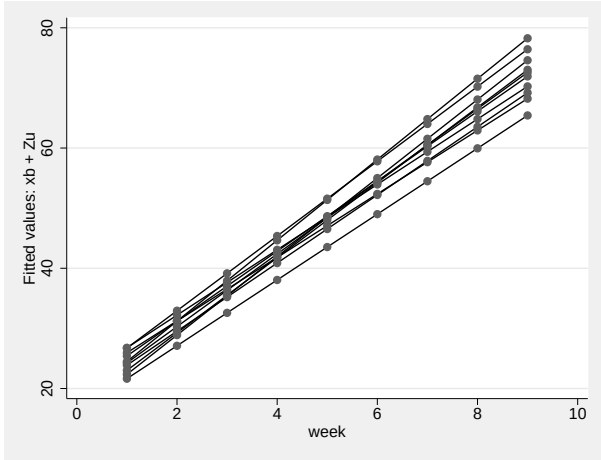
Examining (1), we see that, within each pig, the successive weight measurements are modeled as simple linear regression with intercept $\beta_0 + u_{j0}$ and slope $\beta_1 + u_{j1}$. We can generate estimates of the pig-level intercepts and slopes with

```
. generate intercept = _b[_cons] + u0
. generate slope = _b[week] + u1
. list id intercept slope if id<=10 & tolist
```

	id	interc~t	slope
1.	1	19.59256	5.814132
10.	2	17.77149	6.719934
19.	3	15.82906	6.529933
28.	4	21.31999	5.437926
37.	5	20.65485	5.285748
46.	6	18.20831	5.665081
55.	7	16.76509	6.249341
64.	8	18.21855	6.040239
73.	9	16.16607	5.473345
82.	10	20.51594	6.212973

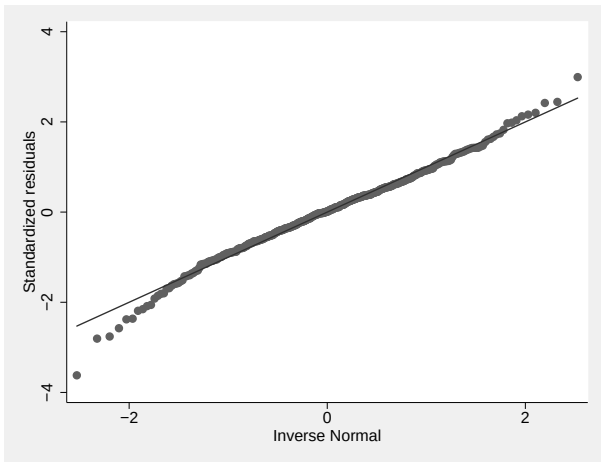
Thus we can plot estimated regression lines for each of the pigs. Equivalently, we can just plot the fitted values because they are based on both the fixed and random effects:

```
. predict fitweight, fitted
. twoway connected fitweight week if id<=10, connect(L)
```



We can also generate standardized residuals and see if they follow a standard normal distribution, as they should in any good-fitting model:

```
. predict rs, rstandard
. summarize rs
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
          rs |        432    1.01e-09    .8929356  -3.621446   3.000929
. qnorm rs
```



◁

Example 2

In example 4 of [XT] **xtmixed**, we estimated a Cobb–Douglas production function with random intercepts at the region level and at the state-within-region level:

$$\mathbf{y}_{jk} = \mathbf{X}_{jk}\boldsymbol{\beta} + u_k^{(3)} + u_{jk}^{(2)} + \boldsymbol{\epsilon}_{jk}$$

```
. use http://www.stata-press.com/data/r12/productivity, clear
(Public Capital Productivity)
. xtmixed gsp private emp hwy water other unemp || region: || state:
  (output omitted)
```

We can use `estat group` to see how the data are broken down by state and region

```
. estat group
```

Group Variable	No. of Groups	Observations per Group Minimum	Average	Maximum
region	9	51	90.7	136
state	48	17	17.0	17

and we are reminded that we have balanced productivity data for 17 years for each state.

We can use `predict, fitted` to get the fitted values

$$\widehat{\mathbf{y}}_{jk} = \mathbf{X}_{jk}\widehat{\boldsymbol{\beta}} + \widehat{u}_k^{(3)} + \widehat{u}_{jk}^{(2)}$$

but if we instead want fitted values at the region level, that is,

$$\widehat{\mathbf{y}}_{jk} = \mathbf{X}_{jk}\widehat{\boldsymbol{\beta}} + \widehat{u}_k^{(3)}$$

we need to use the `level()` option;

```
. predict gsp_region, fitted level(region)
. list gsp gsp_region in 1/10
```

	gsp	gsp_re~n
1.	10.25478	10.40529
2.	10.2879	10.42336
3.	10.35147	10.47343
4.	10.41721	10.52648
5.	10.42671	10.54947
6.	10.4224	10.53537
7.	10.4847	10.60781
8.	10.53111	10.64727
9.	10.59573	10.70503
10.	10.62082	10.72794

❑ Technical note

Out-of-sample predictions are permitted after `xtmixed`, but if these predictions involve BLUPs of random effects, the integrity of the estimation data must be preserved. If the estimation data have changed since the mixed model was fit, `predict` will be unable to obtain predicted random effects that are appropriate for the fitted model and will give an error. Thus, to obtain out-of-sample predictions that contain random-effects terms, be sure that the data for these predictions are in observations that augment the estimation data.

❑

Saved results

`estat recovariance` saves the last-displayed random-effects covariance matrix in `r(cov)` or in `r(corr)` if it is displayed as a correlation matrix.

Methods and formulas

Following the notation defined throughout [XT] **xtmixed**, best linear unbiased predictions (BLUPs) of random effects $\mathbf{u}$ are obtained as

$$\widetilde{\mathbf{u}} = \widetilde{\mathbf{G}}\mathbf{Z}'\widetilde{\mathbf{V}}^{-1}\left(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}\right)$$

where $\widetilde{\mathbf{G}}$ and $\widetilde{\mathbf{V}}$ are $\mathbf{G}$ and $\mathbf{V} = \mathbf{Z}\mathbf{G}\mathbf{Z}' + \sigma^2_\epsilon\mathbf{R}$ with ML or REML estimates of the variance components plugged in. Standard errors for BLUPs are calculated based on the iterative technique of Bates and Pinheiro (1998, sec. 3.3) for estimating the BLUPs themselves. If estimation is done by REML, these standard errors account for uncertainty in the estimate of $\boldsymbol{\beta}$, while for ML the standard errors treat $\boldsymbol{\beta}$ as known. As such, standard errors of REML-based BLUPs will usually be larger.

Fitted values are given by $\mathbf{X}\widehat{\boldsymbol{\beta}} + \mathbf{Z}\widetilde{\mathbf{u}}$, residuals as $\widehat{\boldsymbol{\epsilon}} = \mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}} - \mathbf{Z}\widetilde{\mathbf{u}}$, and standardized residuals as

$$\widehat{\boldsymbol{\epsilon}}_* = \widehat{\sigma}_\epsilon^{-1}\widehat{\mathbf{R}}^{-1/2}\widehat{\boldsymbol{\epsilon}}$$

If the `level(`*levelvar*`)` option is specified, fitted values, residuals, and standardized residuals consider only those random-effects terms up to and including level *levelvar* in the model.

For details concerning the calculation of scores, see *Methods and formulas* in [XT] **xtmixed**.

Reference

Bates, D. M., and J. C. Pinheiro. 1998. Computational methods for multilevel modelling. In *Technical Memorandum BL0112140-980226-01TM*. Murray Hill, NJ: Bell Labs, Lucent Technologies. http://stat.bell-labs.com/NLME/CompMulti.pdf.

Also see

[XT] **xtmixed** — Multilevel mixed-effects linear regression

[U] **20 Estimation and postestimation commands**

Title

xtnbreg — Fixed-effects, random-effects, & population-averaged negative binomial models

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
collinear	keep collinear variables
SE	
vce(*vcetype*)	*vcetype* may be oim, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
irr	report incidence-rate ratios
noskip	perform overall model test as a likelihood-ratio test
nocnsreport	do not display constraints
display_options	control column formats, row spacing, line width, and display of omitted variables and base and empty cells
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
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-group 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 column formats, row spacing, line width, and display of omitted variables and base and empty cells
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, 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**.

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.

coeflegend does not appear in the dialog box.

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

Menu

Statistics > Longitudinal/panel data > Count outcomes > Negative binomial regression (FE, RE, PA)

Description

`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 $\mathbf{x}\beta$ 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)`. See [XT] **xtgee** for information on how to fit other population-averaged models.

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*`)`, `collinear`; see [R] **estimation options**.

SE

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory and that use bootstrap or jackknife methods; 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.

`noskip`; see [R] **estimation options**.

`nocnsreport`; see [R] **estimation options**.

display_options: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `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 option is available with xtnbreg 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.

exposure(*varname*), offset(*varname*); see [R] **estimation options**.

Correlation

corr(*correlation*), force; see [R] **estimation options**.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory, that are robust to some kinds of misspecification, and that use bootstrap or jackknife methods; 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: noomitted, vsquish, noemptycells, baselevels, allbaselevels, 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.

nolog suppresses display of the iteration log.

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

`xtnbreg` is a convenience command if you want the population-averaged model. 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 http://www.stata-press.com/data/r12/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

```
Likelihood-ratio 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)				

◁

▷ 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
Link:                                    log    Obs per group: min =         4
Family:              negative binomial(k=1)                     avg =       4.0
Correlation:                    exchangeable                    max =         4
                                                Wald chi2(1)       =      1.28
Scale parameter:                            1   Prob > chi2        =    0.2571
                                (Std. Err. adjusted for clustering on airline)
```

i_cnt	IRR	Robust 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)				

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
Dispersion           = mean                     Wald chi2(1)    =       0.60
Log pseudolikelihood = -274.55077               Prob > chi2     =     0.4369
                                (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
------------------------------------------------------------------------------
```

Saved results

`xtnbreg, re` saves 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)`	significance
`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(vcetype)`	title used to label Std. Err.
`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

`xtnbreg, fe` saves 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(r2_p)`	pseudo R-squared
`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)`	significance
`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(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

xtnbreg, pa saves 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)	significance
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

Methods and formulas

`xtnbreg` is implemented as an ado-file.

`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 $y_{it} \mid \gamma_{it} \sim \text{Poisson}(\gamma_{it})$, where $\gamma_{it} \mid \delta_i \sim \text{gamma}(\lambda_{it}, \delta_i)$ 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-group 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 $\mathbf{X}_i = (\mathbf{x}_{i1}, \ldots, \mathbf{x}_{in_i})$ and where f is the probability density function for δ_i. The resulting log likelihood is

$$\begin{aligned}\ln L = \sum_{i=1}^{n} w_i \Bigg[&\ln\Gamma(r+s) + \ln\Gamma\Big(r + \sum_{k=1}^{n_i}\lambda_{ik}\Big) + \ln\Gamma\Big(s + \sum_{k=1}^{n_i} y_{ik}\Big) - \ln\Gamma(r) - \ln\Gamma(s) \\ &- \ln\Gamma\Big(r+s+\sum_{k=1}^{n_i}\lambda_{ik}+\sum_{k=1}^{n_i}y_{ik}\Big) + \sum_{t=1}^{n_i}\Big\{\ln\Gamma(\lambda_{it}+y_{it}) - \ln\Gamma(\lambda_{it}) - \ln\Gamma(y_{it}+1)\Big\}\Bigg]\end{aligned}$$

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

$$\begin{aligned}&\Pr\Big(Y_{i1} = y_{i1}, \ldots, Y_{in_i} = y_{in_i} \;\Big|\; \mathbf{X}_i, \textstyle\sum_{t=1}^{n_i} Y_{it} = \sum_{t=1}^{n_i} y_{it}\Big) \\ &\qquad = \frac{\Gamma(\sum_{t=1}^{n_i}\lambda_{it})\Gamma(\sum_{t=1}^{n_i}y_{it}+1)}{\Gamma(\sum_{t=1}^{n_i}\lambda_{it}+\sum_{t=1}^{n_i}y_{it})} \prod_{t=1}^{n_i}\frac{\Gamma(\lambda_{it}+y_{it})}{\Gamma(\lambda_{it})\Gamma(y_{it}+1)}\end{aligned}$$

The conditional log likelihood is

$$\ln L = \sum_{i=1}^{n} w_i \left[\ln\Gamma\left(\sum_{t=1}^{n_i} \lambda_{it}\right) + \ln\Gamma\left(\sum_{t=1}^{n_i} y_{it} + 1\right) - \ln\Gamma\left(\sum_{t=1}^{n_i} \lambda_{it} + \sum_{t=1}^{n_i} y_{it}\right) + \sum_{t=1}^{n_i} \left\{ \ln\Gamma(\lambda_{it} + y_{it}) - \ln\Gamma(\lambda_{it}) - \ln\Gamma(y_{it} + 1) \right\} \right]$$

See Hausman, Hall, and Griliches (1984) for a more thorough development of the random-effects and fixed-effects models. Also see Cameron and Trivedi (1998) for a good textbook treatment of this model.

References

Cameron, A. C., and P. K. Trivedi. 1998. *Regression Analysis of Count Data*. Cambridge: 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.

Liang, K.-Y., and S. L. Zeger. 1986. Longitudinal data analysis using generalized linear models. *Biometrika* 73: 13–22.

Also see

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

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

[XT] **xtpoisson** — Fixed-effects, random-effects, and population-averaged Poisson models

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

[R] **nbreg** — Negative binomial regression

[U] **20 Estimation and postestimation commands**

Title

xtnbreg postestimation — Postestimation tools for xtnbreg

Description

The following postestimation commands are available after `xtnbreg`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
*`estat`	AIC, BIC, VCE, and estimation sample summary
`estimates`	cataloging estimation results
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`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 coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

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

See the corresponding entries in the *Base Reference Manual* for details.

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 $\Pr(y_j = n)$ assuming the random effect is zero; only allowed after `xtnbreg, re`
`pr0(`*a*`,`*b*`)`	probability $\Pr(a \leq y_j \leq b)$ 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 $\mathbf{x}_j\boldsymbol{\beta}$

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

Menu

Statistics > Postestimation > Predictions, residuals, etc.

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 $\Pr(y_j = n)$ 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 $\Pr(a \leq y_j \leq b)$ 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 $\Pr(y_j \geq 20)$;
`pr0(20,`*b*`)` calculates $\Pr(y_j \geq 20)$ in observations for which $b \geq .$ and calculates $\Pr(20 \leq y_j \leq b)$ 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, $u_j = \partial \ln L_j(\mathbf{x}_j\boldsymbol{\beta})/\partial(\mathbf{x}_j\boldsymbol{\beta})$.

`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 $\mathbf{x}_{it}\boldsymbol{\beta} + \text{offset}_{it}$.

Methods and formulas

All postestimation commands listed above are implemented as ado-files.

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**

Title

xtpcse — Linear regression with panel-corrected standard errors

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 column formats, row spacing, line width, and display of omitted variables and base and empty cells
`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` 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.

Menu

Statistics > Longitudinal/panel data > Contemporaneous correlation > Regression with panel-corrected standard errors (PCSE)

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.

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 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 (2012, 280) 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: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `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

`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 (2012), 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

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 http://www.stata-press.com/data/r12/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: min =        20
Autocorrelation:  no autocorrelation                           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
```

invest	Panel-corrected Coef.	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

`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	Coef.	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: min =        20
Autocorrelation:  common AR(1)                               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	Coef.	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

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: min =        20
Autocorrelation:  panel-specific AR(1)                         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					
	Coef.	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

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: min =        20
Autocorrelation:  common AR(1)                                 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					
	Coef.	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.

◁

Saved results

xtpcse saves 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-squared
e(chi2)	χ^2
e(p)	significance
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(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved

Matrices

e(b)	coefficient vector
e(Sigma)	$\widehat{\Sigma}$ 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

xtpcse is implemented as an ado-file.

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 option correlation(psar1)), 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 option correlation(ar1)), 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

$$\mathrm{Var}(\boldsymbol{\beta}) = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\Omega}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}$$

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*.

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, Department of Politics, New York University; Jonathan Katz, Division of the Humanities and Social Science, California Institute of Technology; and Robert John Franzese Jr., Center for Political Studies, Institute for Social Research, University of Michigan.

References

Beck, N., 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.

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.

Grunfeld, Y., and Z. Griliches. 1960. Is aggregation necessarily bad? *Review of Economics and Statistics* 42: 1–13.

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.

Also see

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

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

[XT] **xtgls** — Fit panel-data models by using GLS

[XT] **xtreg** — Fixed-, between-, and random-effects and population-averaged linear models

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

[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**

Title

xtpcse postestimation — Postestimation tools for xtpcse

Description

The following postestimation commands are available after `xtpcse`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat`	VCE and estimation sample summary
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`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 coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

See the corresponding entries in the *Base Reference Manual* for details.

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.

Menu

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

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

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

Methods and formulas

All postestimation commands listed above are implemented as ado-files.

Also see

[XT] **xtpcse** — Linear regression with panel-corrected standard errors

[U] **20 Estimation and postestimation commands**

Title

xtpoisson — Fixed-effects, random-effects, and population-averaged Poisson models

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
collinear	keep collinear variables
SE/Robust	
vce(*vcetype*)	*vcetype* may be oim, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
irr	report incidence-rate ratios
noskip	fit constant-only model and perform likelihood-ratio test
nocnsreport	do not display constraints
display_options	control column formats, row spacing, line width, and display of omitted variables and base and empty cells
Integration	
intmethod(*intmethod*)	integration method; *intmethod* may be mvaghermite, aghermite, or ghermite; default is intmethod(mvaghermite)
intpoints(#)	use # quadrature points; default is intpoints(12)
Maximization	
maximize_options	control the maximization process; seldom used
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
`collinear`	keep collinear variables
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 column formats, row spacing, line width, and display of omitted variables and base and empty cells
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
`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-group correlation structure
`force`	estimate 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 column formats, row spacing, line width, and display of omitted variables and base and empty cells

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
`stationary` #	stationary
`nonstationary` #	nonstationary

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`, `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**.

`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.

`coeflegend` does not appear in the dialog box.

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

Menu

Statistics > Longitudinal/panel data > Count outcomes > Poisson regression (FE, RE, PA)

Description

`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.

By default, the population-averaged model is an equal-correlation model; `xtpoisson, pa` assumes `corr(exchangeable)`. See [XT] **xtgee** for information on how to fit other population-averaged models.

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*`)`, `collinear`; see [R] **estimation options**.

SE/Robust

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory and that use bootstrap or jackknife methods; 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.

`noskip`; see [R] **estimation options**.

`nocnsreport`; see [R] **estimation options**.

display_options: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `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 option is available with `xtpoisson` but is not shown in the dialog box:

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

Options for FE model

Model

`fe` requests the fixed-effects estimator.

`exposure(`*varname*`)`, `offset(`*varname*`)`, `constraints(`*constraints*`)`, `collinear`; see [R] **estimation options**.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory, that are robust to some kinds of misspecification, and that use bootstrap or jackknife methods; 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: noomitted, vsquish, noemptycells, baselevels, allbaselevels, 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 option is available with xtpoisson 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.

exposure(*varname*), offset(*varname*); see [R] **estimation options**.

Correlation

corr(*correlation*), force; see [R] **estimation options**.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory, that are robust to some kinds of misspecification, and that use bootstrap or jackknife methods; 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: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `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.

`nolog` suppresses display of the iteration log.

`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

`xtpoisson` is a convenience command if you want the population-averaged model. 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 $i = 1, \ldots, n$ panels, where $t = 1, \ldots, n_i$, 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 http://www.stata-press.com/data/r12/ships
. xtpoisson accident op_75_79 co_65_69 co_70_74 co_75_79, exposure(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
------------------------------------------------------------------------------
Likelihood-ratio 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	2.008003	.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

                                                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	-2.74	0.006	-4.034745	-.6689918
sigma_u	.3085306	.1324562			.1330045	.7156988

```
Likelihood-ratio test of sigma_u=0: chibar2(01) =    10.67 Pr>=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
Link:                                    log    Obs per group: min =         6
Family:                              Poisson                   avg =       6.8
Correlation:                   exchangeable                    max =         7
                                               Wald chi2(4)       =    252.94
Scale parameter:                            1  Prob > chi2        =    0.0000
                                  (Std. Err. adjusted for clustering on ship)
```

accident	IRR	Robust 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)				

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)
------------------------------------------------------------------------------
```

◁

❑ 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.

❑

Saved results

`xtpoisson, re` saves the following in `e()`:

Scalars

`e(N)`	number of observations
`e(N_g)`	number of groups
`e(N_cd)`	number of completely determined 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(sigma_u)`	panel-level standard deviation
`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)`	significance
`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(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

Functions

e(sample)	marks estimation sample

xtpoisson, re normal saves the following in e():

Scalars

e(N)	number of observations
e(N_g)	number of groups
e(N_cd)	number of completely determined 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(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)	significance
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(offset)	linear offset variable
e(offset1)	ln(*varname*), where *varname* is variable from exposure()
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(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

xtpoisson, fe saves 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(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)	significance
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

xtpoisson, pa saves 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)	significance
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

Methods and formulas

xtpoisson is implemented as an ado-file.

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 (1998) 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

$$\Pr(y_{i1},\ldots,y_{in_i}|\alpha_i,\mathbf{x}_{i1},\ldots,\mathbf{x}_{in_i}) = \left(\prod_{t=1}^{n_i}\frac{\lambda_{it}^{y_{it}}}{y_{it}!}\right)\exp\left\{-\exp(\alpha_i)\sum_{t=1}^{n_i}\lambda_{it}\right\}\exp\left(\alpha_i\sum_{t=1}^{n_i}y_{it}\right)$$

where $\lambda_{it} = \exp(\mathbf{x}_{it}\boldsymbol{\beta})$. We may rewrite the above as [defining $\epsilon_i = \exp(\alpha_i)$]

$$\begin{aligned}\Pr(y_{i1},\ldots,y_{in_i}|\epsilon_i,\mathbf{x}_{i1},\ldots,\mathbf{x}_{in_i}) &= \left\{\prod_{t=1}^{n_i}\frac{(\lambda_{it}\epsilon_i)^{y_{it}}}{y_{it}!}\right\}\exp\left(-\sum_{t=1}^{n_i}\lambda_{it}\epsilon_i\right)\\ &= \left(\prod_{t=1}^{n_i}\frac{\lambda_{it}^{y_{it}}}{y_{it}!}\right)\exp\left(-\epsilon_i\sum_{t=1}^{n_i}\lambda_{it}\right)\epsilon_i^{\sum_{t=1}^{n_i}y_{it}}\end{aligned}$$

We now assume that ϵ_i follows a gamma distribution with mean one and variance $1/\theta$ so that unconditional on ϵ_i

$$\begin{aligned}\Pr(y_{i1},\ldots,y_{in_i}|\mathbf{X}_i) &= \frac{\theta^\theta}{\Gamma(\theta)}\left(\prod_{t=1}^{n_i}\frac{\lambda_{it}^{y_{it}}}{y_{it}!}\right)\int_0^\infty\exp\left(-\epsilon_i\sum_{t=1}^{n_i}\lambda_{it}\right)\epsilon_i^{\sum_{t=1}^{n_i}y_{it}}\epsilon_i^{\theta-1}\exp(-\theta\epsilon_i)d\epsilon_i\\ &= \frac{\theta^\theta}{\Gamma(\theta)}\left(\prod_{t=1}^{n_i}\frac{\lambda_{it}^{y_{it}}}{y_{it}!}\right)\int_0^\infty\exp\left\{-\epsilon_i\left(\theta+\sum_{t=1}^{n_i}\lambda_{it}\right)\right\}\epsilon_i^{\theta+\sum_{t=1}^{n_i}y_{it}-1}d\epsilon_i\\ &= \left(\prod_{t=1}^{n_i}\frac{\lambda_{it}^{y_{it}}}{y_{it}!}\right)\frac{\Gamma\left(\theta+\sum_{t=1}^{n_i}y_{it}\right)}{\Gamma(\theta)}\left(\frac{\theta}{\theta+\sum_{t=1}^{n_i}\lambda_{it}}\right)^\theta\left(\frac{1}{\theta+\sum_{t=1}^{n_i}\lambda_{it}}\right)^{\sum_{t=1}^{n_i}y_{it}}\end{aligned}$$

for $\mathbf{X}_i = (\mathbf{x}_{i1},\ldots,\mathbf{x}_{in_i})$.

The log likelihood (assuming gamma heterogeneity) is then derived using

$$u_i = \frac{\theta}{\theta+\sum_{t=1}^{n_i}\lambda_{it}} \qquad \lambda_{it} = \exp(\mathbf{x}_{it}\boldsymbol{\beta})$$

$$\Pr(Y_{i1}=y_{i1},\ldots,Y_{in_i}=y_{in_i}|\mathbf{X}_i) = \frac{\prod_{t=1}^{n_i}\lambda_{it}^{y_{it}}\Gamma\left(\theta+\sum_{t=1}^{n_i}y_{it}\right)}{\prod_{t=1}^{n_i}y_{it}!\Gamma(\theta)\left(\sum_{t=1}^{n_i}\lambda_{it}\right)^{\sum_{t=1}^{n_i}y_{it}}}u_i^\theta(1-u_i)^{\sum_{t=1}^{n_i}y_{it}}$$

such that the log likelihood may be written as

$$\begin{aligned}L = \sum_{i=1}^{n}w_i\Bigg\{&\log\Gamma\left(\theta+\sum_{t=1}^{n_i}y_{it}\right)-\sum_{t=1}^{n_i}\log\Gamma\left(1+y_{it}\right)-\log\Gamma(\theta)+\theta\log u_i\\ &+\log(1-u_i)\sum_{t=1}^{n_i}y_{it}+\sum_{t=1}^{n_i}y_{it}(\mathbf{x}_{it}\boldsymbol{\beta})-\left(\sum_{t=1}^{n_i}y_{it}\right)\log\left(\sum_{t=1}^{n_i}\lambda_{it}\right)\Bigg\}\end{aligned}$$

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

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

where

$$F(y, z) = \exp\Big\{ -\exp(z) + yz - \log(y!) \Big\}.$$

The panel-level likelihood l_i is given by

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

$$\equiv \int_{-\infty}^{\infty} g(y_{it}, x_{it}, \nu_i) d\nu_i$$

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

$$L \approx \sum_{i=1}^{n} w_i \log\left[\sqrt{2}\widehat{\sigma}_i \sum_{m=1}^{M} w_m^* \exp\big\{(a_m^*)^2\big\} \frac{\exp\big\{-(\sqrt{2}\widehat{\sigma}_i a_m^* + \widehat{\mu}_i)^2/2\sigma_\nu^2\big\}}{\sqrt{2\pi}\sigma_\nu} \right.$$

$$\left. \prod_{t=1}^{n_i} F(y_{it}, x_{it}\boldsymbol{\beta} + \sqrt{2}\widehat{\sigma}_i a_m^* + \widehat{\mu}_i) \right]$$

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.

One can instead use the adaptive quadrature method of Liu and Pierce (1994), the `intmethod(aghermite)` option, which uses the mode and curvature of the mode as approximations for the mean and variance. We take the integrand

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

and find α_i the mode of $g(y_{it}, x_{it}, \nu_i)$. We calculate

$$\gamma_i = -\frac{\partial^2}{\partial \nu_i^2} \log\{g(y_{it}, x_{it}, \nu_i)\}\big|_{\nu_i = \alpha_i}$$

Then

$$\int_{-\infty}^{\infty} g(y_{it}, x_{it}, \nu_i) d\nu_i \approx \left(\frac{2}{\gamma_i}\right)^{1/2} \sum_{m=1}^{M} w_m^* \exp\{(a_m^*)^2\} g\left\{ y_{it}, x_{it}, \left(\frac{2}{\gamma_i}\right)^{1/2} a_m^* + \alpha_i \right\}$$

This adaptation is performed on the first iteration only; that is, the α_i and γ_i are calculated once at the first iteration and then held constant throughout the subsequent iterations.

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

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

All three 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

$$
\begin{aligned}
&\Pr\left(Y_{i1} = y_{i1},\ldots,Y_{in_i} = y_{in_i}|\mathbf{X}_i\right) \\
&\qquad = \prod_{t=1}^{n_i}\frac{1}{y_{it}!}\exp\{-\exp(\alpha_i)\exp(\mathbf{x}_{it}\boldsymbol{\beta}) + \alpha_i y_{it}\}\exp(\mathbf{x}_{it}\boldsymbol{\beta})^{y_{it}} \\
&\qquad = \left(\prod_{t=1}^{n_i}\frac{\exp(\mathbf{x}_{it}\boldsymbol{\beta})^{y_{it}}}{y_{it}!}\right)\exp\left\{-\exp(\alpha_i)\sum_t \exp(\mathbf{x}_{it}\boldsymbol{\beta}) + \alpha_i\sum_t y_{it}\right\}
\end{aligned}
$$

and we also know that the sum of n_i Poisson independent random variables, each with parameter λ_{it} for $t = 1,\ldots,n_i$, is distributed as Poisson with parameter $\sum_t \lambda_{it}$. Thus

$$\Pr\left(\sum_t Y_{it} = \sum_t y_{it} \Big| \mathbf{X}_i\right) =$$

$$\frac{1}{(\sum_t y_{it})!}\exp\left\{-\exp(\alpha_i)\sum_t \exp(\mathbf{x}_{it}\boldsymbol{\beta}) + \alpha_i \sum_t y_{it}\right\}\left\{\sum_t \exp(\mathbf{x}_{it}\boldsymbol{\beta})\right\}^{\sum_t y_{it}}$$

So, the conditional likelihood is conditioned on the sum of the outcomes in the set (panel). The appropriate function is given by

$$\Pr\Big(Y_{i1} = y_{i1}, \ldots, Y_{in_i} = y_{in_i} \Big| \mathbf{X}_i, \sum_t Y_{it} = \sum_t y_{it}\Big) =$$

$$\left[\left(\prod_{t=1}^{n_i} \frac{\exp(\mathbf{x}_{it}\boldsymbol{\beta})^{y_{it}}}{y_{it}!}\right)\exp\left\{-\exp(\alpha_i)\sum_t \exp(\mathbf{x}_{it}\boldsymbol{\beta}) + \alpha_i \sum_t y_{it}\right\}\right] \Big/$$

$$\left[\frac{1}{(\sum_t y_{it})!}\exp\left\{-\exp(\alpha_i)\sum_t \exp(\mathbf{x}_{it}\boldsymbol{\beta}) + \alpha_i \sum_t y_{it}\right\}\left\{\sum_t \exp(\mathbf{x}_{it}\boldsymbol{\beta})\right\}^{\sum_t y_{it}}\right]$$

$$= \left(\sum_t y_{it}\right)! \prod_{t=1}^{n_i} \frac{\exp(\mathbf{x}_{it}\boldsymbol{\beta})^{y_{it}}}{y_{it}!\left\{\sum_k \exp(\mathbf{x}_{ik}\boldsymbol{\beta})\right\}^{y_{it}}}$$

which is free of α_i.

The conditional log likelihood is given by

$$L = \log \prod_{i=1}^{n}\left[\left(\sum_{t=1}^{n_i} y_{it}\right)! \prod_{t=1}^{n_i} \frac{\exp(\mathbf{x}_{it}\boldsymbol{\beta})^{y_{it}}}{y_{it}!\left\{\sum_{\ell=1}^{n_\ell} \exp(\mathbf{x}_{i\ell}\boldsymbol{\beta})\right\}^{y_{it}}}\right]^{w_i}$$

$$= \log \prod_{i=1}^{n}\left\{\frac{(\sum_t y_{it})!}{\prod_{t=1}^{n_i} y_{it}!}\prod_{t=1}^{n_i} p_{it}^{y_{it}}\right\}^{w_i}$$

$$= \sum_{i=1}^{n} w_i \left\{\log\Gamma\left(\sum_{t=1}^{n_i} y_{it} + 1\right) - \sum_{t=1}^{n_i} \log\Gamma(y_{it}+1) + \sum_{t=1}^{n_i} y_{it}\log p_{it}\right\}$$

where

$$p_{it} = e^{\mathbf{x}_{it}\boldsymbol{\beta}} \Big/ \sum_\ell e^{\mathbf{x}_{i\ell}\boldsymbol{\beta}}$$

References

Allison, P. D. 2009. *Fixed Effects Regression Models*. Newbury Park, CA: Sage.

Baltagi, B. H. 2008. *Econometric Analysis of Panel Data*. 4th ed. New York: Wiley.

——. 2009. *A Companion to Econometric Analysis of Panel Data*. Chichester, UK: Wiley.

Cameron, A. C., and P. K. Trivedi. 1998. *Regression Analysis of Count Data*. Cambridge: Cambridge University Press.

Greene, W. H. 2012. *Econometric Analysis*. 7th ed. Upper Saddle River, NJ: Prentice Hall.

Hardin, J. W., and J. M. Hilbe. 2007. *Generalized Linear Models and Extensions*. 2nd 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.

Liang, K.-Y., and S. L. Zeger. 1986. Longitudinal data analysis using generalized linear models. *Biometrika* 73: 13–22.

Liu, Q., and D. A. Pierce. 1994. A note on Gauss–Hermite quadrature. *Biometrika* 81: 624–629.

McCullagh, P., and J. A. Nelder. 1989. *Generalized Linear Models*. 2nd ed. London: Chapman & 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, Series C* 31: 214–225.

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

Wooldridge, J. M. 1999. Distribution-free estimation of some nonlinear panel data models. *Journal of Econometrics* 90: 77–97.

Also see

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

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

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

[XT] **xtnbreg** — Fixed-effects, random-effects, & population-averaged negative binomial models

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

[R] **poisson** — Poisson regression

[U] **20 Estimation and postestimation commands**

Title

xtpoisson postestimation — Postestimation tools for xtpoisson

Description

The following postestimation commands are available after `xtpoisson`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
*`estat`	AIC, BIC, VCE, and estimation sample summary
`estimates`	cataloging estimation results
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`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 coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

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

See the corresponding entries in the *Base Reference Manual* for details.

Syntax for predict

Random-effects (RE) and fixed-effects (FE) 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 $\Pr(y_j = n)$ assuming the random effect is zero; only allowed after `xtpoisson, re`
`pr0(`*a*`,`*b*`)`	probability $\Pr(a \leq y_j \leq b)$ assuming the random effect is zero; only allowed after `xtpoisson, re`

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 $\Pr(y_j = n)$
`pr(`*a*`,`*b*`)`	probability $\Pr(a \leq y_j \leq b)$
`stdp`	standard error of the linear prediction
`score`	first derivative of the log likelihood with respect to $\mathbf{x}_j\boldsymbol{\beta}$

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

Menu

Statistics > Postestimation > Predictions, residuals, etc.

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 $\Pr(y_j = n)$ 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 $\Pr(a \leq y_j \leq b)$ assuming the random effect is zero, where a and b are nonnegative integers that may be specified as numbers or variables (only allowed after `xtpoisson, re`);

b missing ($b \geq .$) means $+\infty$;
`pr0(20,.)` calculates $\Pr(y_j \geq 20)$;
`pr0(20,b)` calculates $\Pr(y_j \geq 20)$ in observations for which $b \geq .$ and calculates $\Pr(20 \leq y_j \leq b)$ 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 $\Pr(y_j = n)$, 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 $\Pr(a \leq y_j \leq b)$ (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, $u_j = \partial \ln L_j(\mathbf{x}_j\boldsymbol{\beta})/\partial(\mathbf{x}_j\boldsymbol{\beta})$.

`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 $\mathbf{x}_{it}\boldsymbol{\beta} + \text{offset}_{it}$.

Remarks

▷ Example 1

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 http://www.stata-press.com/data/r12/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

All postestimation commands listed above are implemented as ado-files.

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 $\mu_{it}=\exp(\mathbf{x}_{it}\boldsymbol{\beta}+\text{offset}_{it})$. 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**

Title

xtprobit — Random-effects and population-averaged probit models

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
collinear	keep collinear variables
SE	
vce(*vcetype*)	*vcetype* may be oim, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
noskip	perform overall model test as a likelihood-ratio test
nocnsreport	do not display constraints
display_options	control column formats, row spacing, line width, and display of omitted variables and base and empty cells
Integration	
intmethod(*intmethod*)	integration method; *intmethod* may be mvaghermite, aghermite, or ghermite; default is intmethod(mvaghermite)
intpoints(#)	use # quadrature points; default is intpoints(12)
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-group 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 column formats, row spacing, line width, and display of omitted variables and base and empty cells
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, 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**.

iweights, fweights, and pweights are allowed for the population-averaged model, and iweights are allowed in the 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.

Menu

Statistics > Longitudinal/panel data > Binary outcomes > Probit regression (RE, PA)

Description

`xtprobit` fits random-effects and population-averaged probit models. 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.

By default, the population-averaged model is an equal-correlation model; `xtprobit, pa` assumes `corr(exchangeable)`. See [XT] **xtgee** for information about how to fit other population-averaged models.

See [R] **logistic** for a list of related 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*`)`, `collinear`; see [R] **estimation options**.

SE

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

Reporting

`level(#)`, `noskip`; see [R] **estimation options**.

`nocnsreport`; see [R] **estimation options**.

display_options: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `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 option is available with `xtprobit` 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.

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

Correlation

`corr(`*correlation*`)`, `force`; see [R] **estimation options**.

SE/Robust

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory, that are robust to some kinds of misspecification, and that use bootstrap or jackknife methods; 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: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `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.

`nolog` suppresses display of the iteration log.

`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

`xtprobit` is a convenience command for obtaining the population-averaged model. 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

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 http://www.stata-press.com/data/r12/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      =     26200
Group variable: idcode                          Number of groups   =      4434
Random effects u_i ~ Gaussian                   Obs per group: min =         1
                                                               avg =       5.9
                                                               max =        12
                                                Wald chi2(6)       =    220.91
Log likelihood  = -10552.225                    Prob > chi2        =    0.0000
```

union	Coef.	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

```
Likelihood-ratio 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 quadrature 12 points	Comparison quadrature 8 points	Comparison quadrature 16 points	
Log likelihood	-10552.225	-10554.496	-10552.399	
		-2.2712569	-.17396615	Difference
		.00021524	.00001649	Relative difference
union: age	.00829671	.00828745	.00831488	
		-9.265e-06	.00001817	Difference
		-.0011167	.00218987	Relative difference
union: grade	.0482731	.04860277	.04826287	
		.00032967	-.00001023	Difference
		.00682917	-.00021188	Relative difference
union: 1.not_smsa	-.13965702	-.14057441	-.13953521	
		-.00091739	.00012181	Difference
		.00656891	-.00087218	Relative difference
union: 1.south	-1.5843944	-1.5909857	-1.5843375	
		-.00659135	.00005689	Difference
		.00416017	-.00003591	Relative difference
union: year	-.00398535	-.00397811	-.00400181	
		7.237e-06	-.00001646	Difference
		-.00181578	.00412982	Relative difference
union: 1.south#c.year	.01340169	.01344457	.01340388	
		.00004288	2.193e-06	Difference
		.00319946	.0001636	Relative difference
union: _cons	-1.6682022	-1.6757524	-1.6665327	
		-.00755024	.00166948	Difference
		.00452597	-.00100077	Relative difference
lnsig2u: _cons	.61036163	.61780789	.60974814	
		.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      =     26200
Group variable: idcode                        Number of groups   =      4434
Random effects u_i ~ Gaussian                 Obs per group: min =         1
                                                             avg =       5.9
                                                             max =        12
                                              Wald chi2(6)       =    218.99
Log likelihood  = -10560.876                  Prob > chi2        =    0.0000
```

union	Coef.	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

```
Likelihood-ratio 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.year            -.0008957      .00001368   Difference
                          -.06549266      .00100054   Relative difference
-------------------------------------------------------------------
union:     -1.5755388     -1.4888646     -1.6505526
   _cons                   .08667418      -.0750138   Difference
                           -.0550124      .04761152   Relative difference
-------------------------------------------------------------------
lnsig2u:    .56159763      .49290978      .58068904
   _cons                  -.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

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      =     26200
Group variable:                       idcode    Number of groups   =      4434
Link:                                 probit    Obs per group: min =         1
Family:                             binomial                   avg =       5.9
Correlation:                    exchangeable                   max =        12
                                                Wald chi2(6)       =    242.57
Scale parameter:                           1    Prob > chi2        =    0.0000
```

union	Coef.	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

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 (the random-effects estimator does not allow the `vce(robust)` specification), 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      =     26200
Group variable:                       idcode    Number of groups   =      4434
Link:                                 probit    Obs per group: min =         1
Family:                             binomial                   avg =       5.9
Correlation:                   exchangeable                    max =        12
                                                Wald chi2(6)       =    156.33
Scale parameter:                           1    Prob > chi2        =    0.0000
                                (Std. Err. adjusted for clustering on idcode)
```

union	Coef.	Robust 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

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 http://www.stata-press.com/data/r12/chicken
. xtprobit complain age grade south tenure gender race income genderm burger
> chicken, nolog
Random-effects probit regression                Number of obs      =      2763
Group variable: restaurant                      Number of groups   =       500
Random effects u_i ~ Gaussian                   Obs per group: min =         3
                                                               avg =       5.5
                                                               max =         8
                                                Wald chi2(10)      =    126.59
Log likelihood  = -1318.2088                    Prob > chi2        =    0.0000
```

complain	Coef.	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

```
Likelihood-ratio 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 quadrature 12 points	Comparison quadrature 8 points	Comparison quadrature 16 points	
Log likelihood	-1318.2088	-1318.2088	-1318.2088	
		-2.002e-06	-1.194e-09	Difference
		1.519e-09	9.061e-13	Relative difference
complain: age	-.04304086	-.04304086	-.04304086	
		-3.896e-10	-2.625e-12	Difference
		9.051e-09	6.100e-11	Relative difference
complain: grade	.0330934	.0330934	.0330934	
		2.208e-11	1.867e-12	Difference
		6.673e-10	5.643e-11	Relative difference
complain: south	.10119998	.10119999	.10119998	
		2.369e-09	3.957e-11	Difference
		2.341e-08	3.910e-10	Relative difference
complain: tenure	-.04400789	-.0440079	-.04400789	
		-3.362e-09	-2.250e-11	Difference
		7.640e-08	5.114e-10	Relative difference
complain: gender	.33184986	.33184986	.33184986	
		3.190e-09	2.546e-11	Difference
		9.612e-09	7.673e-11	Relative difference
complain: race	.34179006	.34179007	.34179006	
		3.801e-09	2.990e-11	Difference
		1.112e-08	8.749e-11	Relative difference
complain: income	-.00227021	-.00227021	-.00227021	
		-4.468e-11	-9.252e-13	Difference
		1.968e-08	4.075e-10	Relative difference
complain: genderm	.05245769	.05245769	.05245769	
		1.963e-09	4.481e-11	Difference
		3.742e-08	8.542e-10	Relative difference
complain: burger	.04489311	.04489311	.04489311	
		4.173e-10	6.628e-12	Difference
		9.296e-09	1.476e-10	Relative difference
complain: chicken	.19047138	.19047139	.19047138	
		3.096e-09	4.916e-11	Difference
		1.625e-08	2.581e-10	Relative difference
complain: _cons	-.21453112	-.21453111	-.21453112	
		1.281e-08	2.682e-10	Difference
		-5.972e-08	-1.250e-09	Relative difference
lnsig2u: _cons	-1.7044935	-1.7044934	-1.7044935	
		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.

◁

Saved results

`xtprobit, re` saves the following in `e()`:

Scalars

`e(N)`	number of observations
`e(N_g)`	number of groups
`e(N_cd)`	number of completely determined 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(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)`	significance
`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(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(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(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

`xtprobit, pa` saves 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)`	significance
`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(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(Cns)`	constraints matrix
`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

Methods and formulas

`xtprobit` is implemented as an ado-file.

`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

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

where

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

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

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

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

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

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.

One can instead use the adaptive quadrature method of Liu and Pierce (1994), the `intmethod(aghermite)` option, which uses the mode and curvature of the mode as approximations for the mean and variance. We take the integrand

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

and find α_i the mode of $g(y_{it}, x_{it}, \nu_i)$. We calculate

$$\gamma_i = -\frac{\partial^2}{\partial \nu_i^2} \log\{g(y_{it}, x_{it}, \nu_i)\}\big|_{\nu_i = \alpha_i}$$

Then

$$\int_{-\infty}^{\infty} g(y_{it}, x_{it}, \nu_i) d\nu_i \approx \left(\frac{2}{\gamma_i}\right)^{1/2} \sum_{m=1}^{M} w_m^* \exp\{(a_m^*)^2\} g\left\{y_{it}, x_{it}, \left(\frac{2}{\gamma_i}\right)^{1/2} a_m^* + \alpha_i\right\}$$

This adaptation is performed on the first iteration only; that is, the α_i and γ_i are calculated once at the first iteration and then held constant throughout the subsequent iterations.

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

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

All three 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.

References

Baltagi, B. H. 2008. *Econometric Analysis of Panel Data*. 4th ed. New York: Wiley.

——. 2009. *A Companion to Econometric Analysis of Panel Data*. Chichester, UK: Wiley.

Conway, M. R. 1990. A random effects model for binary data. *Biometrics* 46: 317–328.

Frechette, G. R. 2001a. sg158: Random-effects ordered probit. *Stata Technical Bulletin* 59: 23–27. Reprinted in *Stata Technical Bulletin Reprints*, vol. 10, pp. 261–266. College Station, TX: Stata Press.

——. 2001b. sg158.1: Update to random-effects ordered probit. *Stata Technical Bulletin* 61: 12. Reprinted in *Stata Technical Bulletin Reprints*, vol. 10, pp. 266–267. College Station, TX: Stata Press.

Guilkey, D. K., and J. L. Murphy. 1993. Estimation and testing in the random effects probit model. *Journal of Econometrics* 59: 301–317.

Liang, K.-Y., and S. L. Zeger. 1986. Longitudinal data analysis using generalized linear models. *Biometrika* 73: 13–22.

Liu, Q., and D. A. Pierce. 1994. A note on Gauss–Hermite quadrature. *Biometrika* 81: 624–629.

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, Series C* 31: 214–225.

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

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.

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.

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

Stewart, M. B. 2006. Maximum simulated likelihood estimation of random-effects dynamic probit models with autocorrelated errors. *Stata Journal* 6: 256–272.

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] **xtgee** — Fit population-averaged panel-data models by using GEE

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

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

[R] **probit** — Probit regression

[U] **20 Estimation and postestimation commands**

Title

xtprobit postestimation — Postestimation tools for xtprobit

Description

The following postestimation commands are available after `xtprobit`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
* `estat`	AIC, BIC, VCE, and estimation sample summary
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`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 coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

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

See the corresponding entries in the *Base Reference Manual* for details.

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
`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 $\mathbf{x}_j\boldsymbol{\beta}$

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

Menu

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

`xb` calculates the linear prediction. This is the default for the random-effects model.

`pu0` calculates the probability of a positive outcome, assuming that the random effect for that observation's panel is zero ($\nu = 0$). This probability 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()`, 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.

`score` calculates the equation-level score, $u_j = \partial \ln L_j(\mathbf{x}_j\boldsymbol{\beta})/\partial(\mathbf{x}_j\boldsymbol{\beta})$.

`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 $\mathbf{x}_{it}\boldsymbol{\beta} + \text{offset}_{it}$.

Remarks

▷ Example 1

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 http://www.stata-press.com/data/r12/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   =      26200
Model VCE    : Conventional
Expression   : Pr(union != 0), predict()
dy/dx w.r.t. : age grade 1.not_smsa 1.south year
```

	Delta-method dy/dx	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` = 0) lowers the probability of being in a union by about two percentage points.

◁

Methods and formulas

All postestimation commands listed above are implemented as ado-files.

Also see

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

[U] **20 Estimation and postestimation commands**

Title

xtrc — Random-coefficients model

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 column formats, row spacing, line width, and display of omitted variables and base and empty cells
`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`, `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.

Menu

Statistics > Longitudinal/panel data > Random-coefficients regression by GLS

Description

`xtrc` fits the Swamy (1970) random-coefficients linear regression model.

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 and that use bootstrap or jackknife methods; 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: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `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

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 http://www.stata-press.com/data/r12/invest2
. reshape wide invest market stock, i(time) j(company)
(note: j = 1 2 3 4 5)
Data                               long   ->   wide
-----------------------------------------------------------------------------
Number of obs.                      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  Parms        RMSE    "R-sq"       chi2        P
------------------------------------------------------------------------------
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
------------------------------------------------------------------------------
```

	Coef.	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 http://www.stata-press.com/data/r12/invest2
. xtrc invest market stock
Random-coefficients regression                  Number of obs      =       100
Group variable: company                         Number of groups   =         5
                                                Obs per group: min =        20
                                                               avg =      20.0
                                                               max =        20
                                                Wald chi2(2)       =     17.55
                                                Prob > chi2        =    0.0002
```

invest	Coef.	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).

◁

Saved results

`xtrc` saves 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(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(Sigma)`	$\widehat{\boldsymbol{\Sigma}}$ 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

Methods and formulas

`xtrc` is implemented as an ado-file.

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

$$\boldsymbol{\beta}_i = \boldsymbol{\beta} + \boldsymbol{\nu}_i \qquad E(\boldsymbol{\nu}_i) = \mathbf{0} \qquad E(\boldsymbol{\nu}_i\boldsymbol{\nu}_i') = \boldsymbol{\Sigma}$$

Our goal is to find $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\Sigma}}$.

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 $\boldsymbol{\Sigma}$,

$$\mathbf{y}_i = \mathbf{X}_i\boldsymbol{\beta}_i + \boldsymbol{\epsilon}_i = \mathbf{X}_i(\boldsymbol{\beta} + \boldsymbol{\nu}_i) + \boldsymbol{\epsilon}_i = \mathbf{X}_i\boldsymbol{\beta} + (\mathbf{X}_i\boldsymbol{\nu}_i + \boldsymbol{\epsilon}_i) = \mathbf{X}_i\boldsymbol{\beta} + \boldsymbol{\omega}_i$$

where $E(\boldsymbol{\omega}_i) = \mathbf{0}$ and

$$E(\boldsymbol{\omega}_i\boldsymbol{\omega}_i') = E\Big\{(\mathbf{X}_i\boldsymbol{\nu}_i + \boldsymbol{\epsilon}_i)(\mathbf{X}_i\boldsymbol{\nu}_i + \boldsymbol{\epsilon}_i)'\Big\} = E(\boldsymbol{\epsilon}_i\boldsymbol{\epsilon}_i') + \mathbf{X}_i E(\boldsymbol{\nu}_i\boldsymbol{\nu}_i')\mathbf{X}_i' = \sigma_i^2\mathbf{I} + \mathbf{X}_i\boldsymbol{\Sigma}\,\mathbf{X}_i' = \boldsymbol{\Pi}_i$$

Stacking the m equations, we have

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\omega}$$

where $\boldsymbol{\Pi} \equiv E(\boldsymbol{\omega}\boldsymbol{\omega}')$ is a block diagonal matrix with $\boldsymbol{\Pi}_i$, $i = 1...m$, along the main diagonal and zeros elsewhere. The GLS estimator of $\widehat{\boldsymbol{\beta}}$ is then

$$\widehat{\boldsymbol{\beta}} = \left(\sum_i \mathbf{X}_i'\boldsymbol{\Pi}_i^{-1}\mathbf{X}_i\right)^{-1}\sum_i \mathbf{X}_i'\boldsymbol{\Pi}_i^{-1}\mathbf{y}_i = \sum_{i=1}^{m}\mathbf{W}_i\mathbf{b}_i$$

where

$$\mathbf{W}_i = \left\{\sum_{i=1}^{m}(\boldsymbol{\Sigma} + \mathbf{V}_i)^{-1}\right\}^{-1}(\boldsymbol{\Sigma} + \mathbf{V}_i)^{-1}$$

$\mathbf{b}_i = (\mathbf{X}_i'\mathbf{X}_i)^{-1}\mathbf{X}_i'\mathbf{y}_i$ 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

$$\mathrm{Var}(\widehat{\boldsymbol{\beta}}) = \sum_{i=1}^{m}(\boldsymbol{\Sigma} + \mathbf{V}_i)^{-1}$$

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

$$\mathrm{Var}(\widehat{\boldsymbol{\beta}}_i) = \mathrm{Var}(\widehat{\boldsymbol{\beta}}) + (\mathbf{I} - \mathbf{A}_i)\left\{\widehat{\mathbf{V}}_i - \mathrm{Var}(\widehat{\boldsymbol{\beta}})\right\}(\mathbf{I} - \mathbf{A}_i)'$$

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

$$\chi^2_{k(m-1)} = \sum_{i=1}^{m}(\mathbf{b}_i - \overline{\boldsymbol{\beta}}^*)'\widehat{\mathbf{V}}_i^{-1}(\mathbf{b}_i - \overline{\boldsymbol{\beta}}^*) \qquad \text{where} \qquad \overline{\boldsymbol{\beta}}^* = \left(\sum_{i=1}^{m}\widehat{\mathbf{V}}_i^{-1}\right)^{-1}\sum_{i=1}^{m}\widehat{\mathbf{V}}_i^{-1}\mathbf{b}_i$$

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 $\boldsymbol{\Pi}_i$.

References

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.

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.

——. 1971. *Statistical Inference in Random Coefficient Regression Models*. New York: Springer.

Also see

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

[XT] **xtmixed** — Multilevel mixed-effects linear regression

[XT] **xtreg** — Fixed-, between-, and random-effects and population-averaged linear models

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

[U] **20 Estimation and postestimation commands**

Title

xtrc postestimation — Postestimation tools for xtrc

Description

The following postestimation commands are available after `xtrc`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat`	VCE and estimation sample summary
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combination of coefficients
`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 coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

See the corresponding entries in the *Base Reference Manual* for details.

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.

Menu

Statistics > Postestimation > Predictions, residuals, etc.

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 $\mathbf{x}_{it}\mathbf{b} + \text{offset}_{it}$.

Methods and formulas

All postestimation commands listed above are implemented as ado-files.

Also see

[XT] **xtrc** — Random-coefficients model

[U] **20 Estimation and postestimation commands**

Title

xtreg — Fixed-, between-, and random-effects and population-averaged linear models

Syntax

GLS random-effects (RE) model

`xtreg` *depvar* [*indepvars*] [*if*] [*in*] [`, re` *RE_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*]

ML random-effects (MLE) model

`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
display_options	control column formats, row spacing, line width, and display of omitted variables and base and empty cells
`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)`
display_options	control column formats, row spacing, line width, and display of omitted variables and base and empty cells
`coeflegend`	display legend instead of statistics

FE_options	Description
Model	
`fe`	use fixed-effects estimator
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `conventional`, `robust`, `cluster` *clustvar*, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
display_options	control column formats, row spacing, line width, and display of omitted variables and base and empty cells
`coeflegend`	display legend instead of statistics

MLE_options	Description
Model	
noconstant	suppress constant term
mle	use ML random-effects estimator
SE	
vce(*vcetype*)	*vcetype* may be oim, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
display_options	control column formats, row spacing, line width, and display of omitted variables and base and empty cells
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-group 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)
display_options	control column formats, row spacing, line width, and display of omitted variables and base and empty cells
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`, `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**.

`aweights`, `fweights`, and `pweights` are allowed for the fixed-effects model. `iweights`, `fweights`, and `pweights` are allowed for the population-averaged model. `iweights` are allowed for the maximum-likelihood random-effects (MLE) 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.

Menu

Statistics > Longitudinal/panel data > Linear models > Linear regression (FE, RE, PA, BE)

Description

`xtreg` fits regression models to panel data. In particular, `xtreg` with the `be` option fits random-effects models by using the between regression estimator; with the `fe` option, it fits fixed-effects models (by using the within regression estimator); and with the `re` option, it fits random-effects models by using the GLS estimator (producing a matrix-weighted average of the between and within results). See [XT] **xtdata** for a faster way to fit fixed- and random-effects models.

Options for RE model

Model

`re`, the default, requests the GLS random-effects estimator.

`sa` specifies that the small-sample Swamy–Arora estimator 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, that are robust to some kinds of misspecification, that allow for intragroup correlation, and that use bootstrap or jackknife methods; 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`, used with `xtreg, re` only, 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: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `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 and that use bootstrap or jackknife methods; 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**.

display_options: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `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.

SE/Robust

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

Reporting

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

display_options: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `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 MLE model

Model

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

`mle` requests the maximum-likelihood random-effects estimator.

SE

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

Reporting

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

display_options: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `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*`)`, `force`; see [R] **estimation options**.

SE/Robust

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

display_options: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `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.

`nolog` suppresses display of the iteration log.

`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

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

See Baltagi (2008, chap. 2) and Wooldridge (2009, 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 residual that we have little interest in; we want estimates of $\boldsymbol{\beta}$. ν_i is the unit-specific residual; 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 (FEV) year after year and so would have a negative ν_i.

ϵ_{it} is the "usual" residual 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 standard residual, 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

$$(y_{it} - \theta\overline{y}_i) = (1-\theta)\alpha + (\mathbf{x}_{it} - \theta\overline{\mathbf{x}}_i)\boldsymbol{\beta} + \{(1-\theta)\nu_i + (\epsilon_{it} - \theta\overline{\epsilon}_i)\} \tag{4}$$

where θ is a function of σ_ν^2 and σ_ϵ^2. If $\sigma_\nu^2 = 0$, meaning that ν_i is always 0, $\theta = 0$ and (1) can be estimated by OLS directly. Alternatively, if $\sigma_\epsilon^2 = 0$, meaning that ϵ_{it} is 0, $\theta = 1$ and the within estimator returns all the information available (which will, in fact, be a regression with an R^2 of 1).

For more reasonable cases, 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 a residual. 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.

All 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 FEV); 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.

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 $\mathrm{Var}(\widehat{y})/\mathrm{Var}(y)$. 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^2s, 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^2s are merely correlations squared, or, if you prefer, R^2s 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^2s are correlations squared, or, if you prefer, R^2s 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^2s 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^2s are correlations squared, or, if you prefer, from second-round regressions.

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 http://www.stata-press.com/data/r12/nlswork
(National Longitudinal Survey.  Young Women 14-26 years of age 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      =     28091
Group variable: idcode                          Number of groups   =      4697
R-sq:  within  = 0.1591                         Obs per group: 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	Coef.	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
2.race	-.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 $\nu_i + \bar{\epsilon}_i$, 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      =     28091
Group variable: idcode                          Number of groups   =      4697

R-sq:  within  = 0.1727                         Obs per group: 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	Coef.	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
2.race	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) =     5.13         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^2s 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).

`xtreg, fe` can estimate σ_ν and σ_ϵ, although how you interpret these estimates depends on whether you are using `xtreg` to fit a fixed-effects model or random-effects model. To clarify this fine point, in the fixed-effects model, ν_i are formally fixed—they have no distribution. If you subscribe to this view, think of the reported $\widehat{\sigma}_\nu$ as merely an arithmetic way to describe the range of the estimated but fixed ν_i. If, however, you are using the fixed-effects estimator of the random-effects model, 0.355622 is an estimate of σ_ν or would be if there were no omitted variables.

Here both grade and 2.race were omitted from the model because they do not vary over time. Because grade and 2.race are time invariant, our estimate u_i is an estimate of ν_i plus the effects of grade and 2.race, so our estimate of the standard deviation is based on the variation in ν_i, grade, and 2.race. 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) 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      =     28091
Group variable: idcode                          Number of groups   =      4697

R-sq:  within  = 0.1727                         Obs per group: 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 4697 clusters in idcode)
------------------------------------------------------------------------------
             |               Robust
     ln_wage |      Coef.   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
             |
       2.race |          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 (2009) 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      =     28091
Group variable: idcode                          Number of groups   =      4697

R-sq:  within  = 0.1715                         Obs per group: 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	Coef.	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
2.race	-.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^2s, 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 $\mathbf{x}$ 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 ML

We could also have fit this random-effects model with the maximum likelihood 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, mle

Fitting constant-only model:
Iteration 0:   log likelihood = -13690.161
Iteration 1:   log likelihood = -12819.317
Iteration 2:   log likelihood = -12662.039
Iteration 3:   log likelihood = -12649.744
Iteration 4:   log likelihood = -12649.614
Iteration 5:   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      =     28091
Group variable: idcode                          Number of groups   =      4697

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	Coef.	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
2.race	-.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

```
Likelihood-ratio 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` (ML)	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      =     28091
Group variable:                      idcode     Number of groups   =      4697
Link:                              identity     Obs per group: min =         1
Family:                            Gaussian                    avg =       6.0
Correlation:                   exchangeable                    max =        15
                                                Wald chi2(10)      =   9598.89
Scale parameter:                    .1436709    Prob > chi2        =    0.0000
```

ln_wage	Coef.	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
2.race	-.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`	(ML)	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.

◁

Saved results

`xtreg, re` saves 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)`	ancillary parameter (`gamma`, `lnormal`)
`e(sigma_u)`	panel-level standard deviation
`e(sigma_e)`	standard deviation of ϵ_{it}
`e(r2_w)`	R-squared for within model
`e(r2_o)`	R-squared for overall model
`e(r2_b)`	R-squared for between model
`e(N_clust)`	number of clusters
`e(chi2)`	χ^2
`e(p)`	significance
`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)`	Swamy–Arora estimator of the variance components (`sa` only)
`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

`xtreg, be` saves the following in `e()`:

Scalars

`e(N)`	number of observations
`e(N_g)`	number of groups
`e(typ)`	`WLS`, if `wls` specified
`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-squared
`e(r2_a)`	adjusted R-squared
`e(r2_w)`	R-squared for within model
`e(r2_o)`	R-squared for overall model
`e(r2_b)`	R-squared for between model
`e(F)`	F statistic
`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(title)`	title in estimation output
`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

Functions

`e(sample)`	marks estimation sample

`xtreg, fe` saves 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(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)`	R-squared
`e(r2_a)`	adjusted R-squared
`e(r2_w)`	R-squared for within model
`e(r2_o)`	R-squared for overall model
`e(r2_b)`	R-squared for between model
`e(ll)`	log likelihood
`e(ll_0)`	log likelihood, constant-only model
`e(N_clust)`	number of clusters
`e(rho)`	ρ
`e(F)`	F statistic
`e(F_f)`	F for u_i=0
`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)`	`xtreg`
`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(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(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(V)`	variance–covariance matrix of the estimators

Functions

`e(sample)`	marks estimation sample

`xtreg, mle` saves 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(chi2)`	χ^2
`e(chi2_c)`	χ^2 for comparison 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(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 RE
`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

xtreg, pa saves 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)	significance
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

Methods and formulas

xtreg is implemented as an ado-file.

The model to be fit is

$$y_{it} = \alpha + \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i + \epsilon_{it}$$

for $i = 1, \ldots, n$ and, for each i, $t = 1, \ldots, T$, 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**, in particular, in *Introduction* and *Methods and formulas*. Wooldridge (2009) and Arellano (2003) discuss this application of the Huber/White/sandwich VCE estimator. As discussed by Wooldridge (2009), 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.

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 $\text{corr}(\overline{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}, \overline{y}_i)^2$.

Reported as R^2 overall is $\text{corr}(\mathbf{x}_{it}\widehat{\boldsymbol{\beta}}, y_{it})^2$.

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, $\widehat{\sigma}^2_e$, and the individual component, $\widehat{\sigma}^2_u$, the GLS transform of a variable z for the random-effects model is

$$z^*_{it} = z_{it} - \widehat{\theta}_i \overline{z}_i$$

where $\overline{z}_i = \frac{1}{T_i}\sum_t^{T_i} z_{it}$ and

$$\widehat{\theta}_i = 1 - \sqrt{\frac{\widehat{\sigma}^2_e}{T_i\widehat{\sigma}^2_u + \widehat{\sigma}^2_e}}$$

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, in *Introduction* and *Methods and formulas*. Wooldridge (2009) and Arellano (2003) discuss this application of the Huber/White/sandwich VCE estimator. As discussed by Wooldridge (2009), 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^2_e. However, the two methods differ in their estimator of σ^2_u in unbalanced panels. We call the first $\widehat{\sigma}^2_{u\overline{T}}$ and the second $\widehat{\sigma}^2_{\text{uSA}}$. Both estimators are consistent; however, $\widehat{\sigma}^2_{\text{uSA}}$ has a more elaborate adjustment for small samples than $\widehat{\sigma}^2_{u\overline{T}}$. (See Baltagi [2008], 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,

$$\widehat{\sigma}^2_e = \frac{\sum_i^n \sum_t^{T_i} e^2_{it}}{N - n - K + 1}$$

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 $N = \sum_i^n T_i$. After passing the within residuals through the within transform, only the idiosyncratic errors are left.

The default method for estimating σ_u^2 is

$$\widehat{\sigma}^2_{u\overline{T}} = \max\left\{0, \frac{SSR_b}{n-K} - \frac{\widehat{\sigma}^2_e}{\overline{T}}\right\}$$

where

$$SSR_b = \sum_i^n \left(\overline{y}_i - \widehat{\alpha}_b - \overline{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}_b\right)^2$$

$\widehat{\alpha}_b$ and $\widehat{\boldsymbol{\beta}}_b$ are coefficient estimates from the between regression and $\overline{T}$ is the harmonic mean of T_i:

$$\overline{T} = \frac{n}{\sum_i^n \frac{1}{T_i}}$$

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,

$$\widehat{\sigma}^2_{\text{uSA}} = \max\left\{0, \frac{SSR_b - (n-K)\widehat{\sigma}^2_e}{N-tr}\right\}$$

where

$$tr = \text{trace}\left\{(\mathbf{X}'\mathbf{P}\mathbf{X})^{-1}\mathbf{X}'\mathbf{Z}\mathbf{Z}'\mathbf{X}\right\}$$

$$\mathbf{P} = \text{diag}\left\{\left(\frac{1}{T_i}\right)\boldsymbol{\iota}_{T_i}\boldsymbol{\iota}'_{T_i}\right\}$$

$$\mathbf{Z} = \text{diag}\left[\boldsymbol{\iota}_{T_i}\right]$$

$\mathbf{X}$ 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 $(\widehat{\alpha}_r, \widehat{\boldsymbol{\beta}}_r)$ and their covariance matrix $\mathbf{V}_r$ 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 $\text{corr}\left\{(\mathbf{x}_{it} - \overline{\mathbf{x}}_i)\widehat{\boldsymbol{\beta}}, y_{it} - \overline{y}_i\right\}^2$.

Reported as R^2 overall is $\text{corr}(\mathbf{x}_{it}\widehat{\boldsymbol{\beta}}, y_{it})^2$.

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.

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, and Badi Baltagi of Syracuse University and Manuelita Ureta of Texas A&M University, who assisted us in working our way through the literature.

References

Allison, P. D. 2009. *Fixed Effects Regression Models*. Newbury Park, CA: Sage.

Andrews, M., 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.

——. 2003. *Panel Data Econometrics*. Oxford: Oxford University Press.

Baltagi, B. H. 1985. Pooling cross-sections with unequal time-series lengths. *Economics Letters* 18: 133–136.

——. 2008. *Econometric Analysis of Panel Data*. 4th ed. New York: Wiley.

——. 2009. *A Companion to Econometric Analysis of Panel Data*. 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.

Baum, C. F. 2001. Residual diagnostics for cross-section time series regression models. *Stata Journal* 1: 101–104.

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.

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*, ed. J. H. Dwyer, M. Feinleib, P. Lippert, and H. Hoffmeister, 3–48. New York: Oxford University Press.

Greene, W. H. 1983. Simultaneous estimation of factor substitution, economies of scale, and non-neutral technical change. In *Econometric Analyses of Productivity*, ed. A. Dogramaci, 121–144. Boston: Kluwer.

——. 2012. *Econometric Analysis*. 7th ed. Upper Saddle River, NJ: Prentice Hall.

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.

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, Armidale, Australia: University of New England.

Nichols, A. 2007. Causal inference with observational data. *Stata Journal* 7: 507–541.

Rabe-Hesketh, S., A. Pickles, and C. Taylor. 2000. sg129: Generalized linear latent and mixed models. *Stata Technical Bulletin* 53: 47–57. Reprinted in *Stata Technical Bulletin Reprints*, vol. 9, pp. 293–307. College Station, TX: Stata Press.

Sosa-Escudero, W., and A. K. Bera. 2001. sg164: Specification tests for linear panel data models. *Stata Technical Bulletin* 61: 18–21. Reprinted in *Stata Technical Bulletin Reprints*, vol. 10, pp. 307–311. 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.

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.

Taub, A. J. 1979. Prediction in the context of the variance-components model. *Journal of Econometrics* 10: 103–107.

Twisk, J. W. R. 2003. *Applied Longitudinal Data Analysis for Epidemiology: A Practical Guide*. Cambridge: Cambridge University Press.

Wooldridge, J. M. 2009. *Introductory Econometrics: A Modern Approach*. 4th ed. Cincinnati, OH: South-Western.

Also see

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

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

[XT] **xtgls** — Fit panel-data models by using GLS

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

[XT] **xtmixed** — Multilevel mixed-effects linear regression

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

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

[R] **areg** — Linear regression with a large dummy-variable set

[R] **regress** — Linear regression

[TS] **prais** — Prais–Winsten and Cochrane–Orcutt regression

[U] **20 Estimation and postestimation commands**

Title

xtreg postestimation — Postestimation tools for xtreg

Description

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

Command	Description
`xttest0`	Breusch and Pagan LM test for random effects

For information about this command, see below.

The following standard postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
* `estat`	AIC, BIC, VCE, and estimation sample summary
`estimates`	cataloging estimation results
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`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 coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

* `estat ic` is not appropriate after `xtreg` with the `be`, `pa`, or `re` option.

See the corresponding entries in the *Base Reference Manual* for details.

Special-interest postestimation commands

`xttest0`, for use after `xtreg, re`, presents the Breusch and Pagan (1980) Lagrange multiplier test for random effects, a test that $\mathrm{Var}(\nu_i) = 0$.

Syntax for predict

For all but the population-averaged model

```
predict [type] newvar [if] [in] [, statistic nooffset]
```

Population-averaged model

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

statistic	Description
Main	
`xb`	$\mathbf{x}_j\mathbf{b}$, fitted values; the default
`stdp`	standard error of the fitted values
`ue`	$u_i + e_{it}$, the combined residual
*`xbu`	$\mathbf{x}_j\mathbf{b} + u_i$, prediction including effect
*`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.

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 $\mathbf{x}_j\boldsymbol{\beta}$

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

Menu

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

`xb` calculates the linear prediction, that is, $a + \mathbf{b}\mathbf{x}_{it}$. This is the default for all except the population-averaged 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 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.

`ue` calculates the prediction of $u_i + e_{it}$.

`xbu` calculates the prediction of $a + \mathbf{b}\mathbf{x}_{it} + u_i$, the prediction including the fixed or random component.

`u` calculates the prediction of u_i, the estimated fixed or random effect.

`e` calculates the prediction of e_{it}.

`score` calculates the equation-level score, $u_j = \partial \ln L_j(\mathbf{x}_j\boldsymbol{\beta})/\partial(\mathbf{x}_j\boldsymbol{\beta})$.

`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 $\mathbf{x}_{it}\mathbf{b}$ rather than $\mathbf{x}_{it}\mathbf{b} + \text{offset}_{it}$.

Syntax for xttest0

```
xttest0
```

Menu

Statistics > Longitudinal/panel data > Linear models > Lagrange multiplier test for random effects

Remarks

▷ Example 1

Continuing with our `xtreg, re` estimation example (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 http://www.stata-press.com/data/r12/nlswork
(National Longitudinal Survey.  Young Women 14-26 years of age 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

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	S.E.
age	.0359987	.0368059	-.0008073	.0013177
c.age#c.age	-.000723	-.0007133	-9.68e-06	.0000184
ttl_exp	.0334668	.0290208	.0044459	.001711
c.ttl_exp#~p	.0002163	.0003049	-.0000886	.000053
tenure	.0357539	.0392519	-.003498	.0005797
c.tenure#c~e	-.0019701	-.0020035	.0000334	.0000373
not_smsa	-.0890108	-.1308252	.0418144	.0062745
south	-.0606309	-.0868922	.0262613	.0081345

```
                b = consistent under Ho and Ha; obtained from xtreg
     B = inconsistent under Ha, efficient under Ho; obtained from xtreg

  Test:  Ho:  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.

What does this mean? 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(idcode)
. generate devnsma = not_smsa -avgnsmsa
(8 missing values generated)
. egen avgsouth = mean(south), by(idcode)
. 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      =     28091
Group variable: idcode                          Number of groups   =      4697
R-sq:  within  = 0.1723                         Obs per group: 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	Coef.	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
2.race	-.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	S.E.
age	.0359987	.0375196	-.0015209	.0013198
c.age#c.age	-.000723	-.0007248	1.84e-06	.0000184
ttl_exp	.0334668	.0286543	.0048124	.0017127
c.ttl_exp#~p	.0002163	.0003222	-.0001059	.0000531
tenure	.0357539	.0394423	-.0036884	.0005839
c.tenure#c~e	-.0019701	-.0020081	.000038	.0000377
devnsma	-.0890108	-.0887596	-.0002512	.000683
devsouth	-.0606309	-.0598538	-.0007771	.0007618

```
                b = consistent under Ho and Ha; obtained from xtreg
B = inconsistent under Ha, efficient under Ho; obtained from xtreg

    Test:  Ho:  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.

❑

Methods and formulas

All postestimation commands listed above are implemented as ado-files.

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

$$y_{it} = \alpha + \mathbf{x}_{it}\boldsymbol{\beta} + \nu_{it}$$

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

$$\lambda_{\text{LM}} = \begin{cases} \frac{nT}{2(T-1)}\left\{\frac{\sum_i(\sum_t v_{it})^2}{\sum_i\sum_t v_{it}^2} - 1\right\}^2, & \widehat{\sigma}_u^2 \geq 0 \\ 0 \qquad\qquad\qquad\qquad\qquad\quad , & \widehat{\sigma}_u^2 < 0 \end{cases}$$

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 $\chi^2(1)$.

References

Baltagi, B. H., and Q. Li. 1990. A Lagrange multiplier test for the error components model with incomplete panels. *Econometric Reviews* 9: 103–107.

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.

Hausman, J. A. 1978. Specification tests in econometrics. *Econometrica* 46: 1251–1271.

Sosa-Escudero, W., and A. K. Bera. 2008. Tests for unbalanced error-components models under local misspecification. *Stata Journal* 8: 68–78.

Also see

[XT] **xtreg** — Fixed-, between-, and random-effects and population-averaged linear models

[U] **20 Estimation and postestimation commands**

Title

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

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; see *Options* for details; 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 column formats, row spacing, line width, and display of omitted variables and base and empty cells
`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` and `statsby` are allowed; see [U] **11.1.10 Prefix commands**.

`fweights` and `aweights` 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.

Menu

Statistics > Longitudinal/panel data > Linear models > Linear regression with AR(1) disturbance (FE, RE)

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. Consider the model

$$y_{it} = \alpha + \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i + \epsilon_{it} \qquad i = 1, \ldots, N; \quad t = 1, \ldots, T_i \tag{1}$$

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. If ν_i are assumed to be fixed parameters, the model is a fixed-effects model. If ν_i are assumed to be realizations of an i.i.d. process with mean 0 and variance σ^2_ν, it is a random-effects model. Whereas in the fixed-effects model, the ν_i may be correlated with the covariates $\mathbf{x}_{it}$, in the random-effects model the ν_i are assumed to be independent of the $\mathbf{x}_{it}$. On the other hand, any $\mathbf{x}_{it}$ that do not vary over t are collinear with the ν_i and will be dropped from the fixed-effects model. In contrast, the random-effects model can accommodate covariates that are constant over time.

`xtregar` can accommodate unbalanced panels whose observations are unequally spaced over time. `xtregar` implements the methods derived in Baltagi and Wu (1999).

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, 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: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `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

Remarks are presented under the following headings:

Introduction

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

Consider a linear panel-data model described by (1) and (2). In the fixed-effects model, the ν_i are a set of fixed parameters to be estimated. Alternatively, the ν_i may be random and correlated with the other covariates, with inference conditional on the ν_i in the sample; see Mundlak (1978) and Hsiao (2003). In the random-effects model, also known as the variance-components model, the ν_i are assumed to be realizations of an i.i.d. process with mean 0 and variance σ^2_ν. `xtregar` offers a within estimator for the fixed-effect model and the Baltagi–Wu (1999) GLS estimator of the random-effects model. 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. Both these estimators offer several estimators of ρ.

The data can be unbalanced and unequally spaced. Specifically, the dataset contains observations on individual i at times t_{ij} for $j = 1, \ldots, n_i$. The difference $t_{ij} - t_{i,j-1}$ plays an integral role in the estimation techniques used by `xtregar`. For this reason, you must `xtset` your data before using `xtregar`. For instance, if you have quarterly data, the "time" difference between the third and fourth quarter must be 1 month, not 3.

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

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 http://www.stata-press.com/data/r12/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-sq:  within  = 0.5927                         Obs per group: 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 |      Coef.   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 due to 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-sq:  within  = 0.6583                         Obs per group: 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 |      Coef.   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 due to 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

`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-sq:  within  = 0.5927                         Obs per group: 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	Coef.	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 due to 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

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 !=1934 & year !=1944, fe lbi
FE (within) regression with AR(1) disturbances  Number of obs      =       180
Group variable: company                         Number of groups   =        10

R-sq:  within  = 0.5954                         Obs per group: min =        18
       between = 0.7952                                        avg =      18.0
       overall = 0.7889                                        max =        18

                                                F(2,168)           =    123.63
corr(u_i, Xb)  = -0.0516                        Prob > F           =    0.0000

------------------------------------------------------------------------------
      invest |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      mvalue |   .0941122   .0090926    10.35   0.000     .0761617    .1120627
      kstock |   .3535872   .0303562    11.65   0.000     .2936584    .4135161
       _cons |  -64.82534   5.946885   -10.90   0.000    -76.56559   -53.08509
-------------+----------------------------------------------------------------
      rho_ar |   .6697198
     sigma_u |  93.320452
     sigma_e |  41.580712
     rho_fov |   .83435413   (fraction of variance due to u_i)
------------------------------------------------------------------------------
F test that all u_i=0:     F(9,168) =    11.55                 Prob > F = 0.0000
modified Bhargava et al. Durbin-Watson = .71380994
Baltagi-Wu LBI = 1.0134522
```

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

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 !=1934 & year !=1944, re lbi
RE GLS regression with AR(1) disturbances       Number of obs      =       190
Group variable: company                         Number of groups   =        10
R-sq:  within  = 0.7707                         Obs per group: min =        19
       between = 0.8039                                        avg =      19.0
       overall = 0.7958                                        max =        19
                                                Wald chi2(3)       =    351.37
corr(u_i, Xb)      = 0 (assumed)                Prob > chi2        =    0.0000
```

invest	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
mvalue	.0947714	.0083691	11.32	0.000	.0783683	.1111746
kstock	.3223932	.0263226	12.25	0.000	.2708019	.3739845
_cons	-45.21427	27.12492	-1.67	0.096	-98.37814	7.949603
rho_ar	.6697198	(estimated autocorrelation coefficient)				
sigma_u	74.662876					
sigma_e	42.253042					
rho_fov	.75742494	(fraction of variance due to u_i)				
theta	.66973313					

```
modified Bhargava et al. Durbin-Watson = .71380994
Baltagi-Wu LBI = 1.0134522
```

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.

◁

Saved results

`xtregar, re` saves 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-squared for within model
`e(r2_o)`	R-squared for overall model
`e(r2_b)`	R-squared 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 $\rho_{\rm ar}$
`e(dw)`	LBI, if requested
`e(chi2type)`	`Wald`; type of model χ^2 test
`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(V)`	VCE for random-effects model

Functions

`e(sample)`	marks estimation sample

`xtregar, fe` saves 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-squared
`e(r2_w)`	R-squared for within model
`e(r2_o)`	R-squared for overall model
`e(r2_b)`	R-squared 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 $\rho_{\rm ar}$
`e(dw)`	LBI, if requested
`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(V)`	variance–covariance matrix of the estimators

Functions

`e(sample)`	marks estimation sample

Methods and formulas

`xtregar` is implemented as an ado-file.

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 $j = 1, \ldots, n_i$.

Methods and formulas are presented under the following headings:

Estimating ρ

The estimate of ρ is always obtained after removing the group means. Let $\widetilde{y}_{it} = y_{it} - \overline{y}_i$, let $\widetilde{\mathbf{x}}_{it} = \mathbf{x}_{it} - \overline{\mathbf{x}}_i$, and let $\widetilde{\epsilon}_{it} = \epsilon_{it} - \overline{\epsilon}_i$.

Then, except for the `onestep` method, all the estimates of ρ are obtained by running Stata's `prais` on

$$\widetilde{y}_{it} = \widetilde{x}_{it}\boldsymbol{\beta} + \widetilde{\epsilon}_{it}$$

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 $\widetilde{\epsilon}_{it_{ij}}$. 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

$$y^*_{it_{ij}} = \begin{cases} (1-\rho^2)^{1/2} y_{it_{ij}} & \text{if } t_{ij} = 1 \\ (1-\rho^2)^{1/2}\left[y_{i,t_{ij}} \left\{ \frac{1}{1-\rho^{2(t_{ij}-t_{i,j-1})}} \right\}^{1/2} - y_{i,t_{i,j-1}} \left\{ \frac{\rho^{2(t_{ij}-t_{i,j-1})}}{1-\rho^{2(t_{i,j}-t_{i,j-1})}} \right\}^{1/2} \right] & \text{if } t_{ij} > 1 \end{cases}$$

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 dropped. 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:

$$\widehat{\sigma}^2_{\omega} = \sum_{i=1}^{N} \frac{(\mu_i^{*\prime} g_i)^2}{(g_i^{\prime} g_i)}$$

$$\widehat{\sigma}^2_{\epsilon} = \frac{\left[\sum_{i=1}^{N}(\mu_i^{*\prime}\mu_i^{*}) - \sum_{i=1}^{N}\left\{\frac{(\mu_i^{*\prime} g_i)^2}{(g_i^{\prime} g_i)}\right\}\right]}{\sum_{i=1}^{N}(n_i - 1)}$$

$$\widehat{\sigma}^2_{\mu} = \frac{\left[\sum_{i=1}^{N}\left\{\frac{(\mu_i^{*\prime} g_i)^2}{(g_i^{\prime} g_i)}\right\} - N\widehat{\sigma}^2_{\epsilon}\right]}{\sum_{i=1}^{N}(g_i^{\prime} g_i)}$$

where

$$g_i = \left[1,\ \frac{\left\{1-\rho^{(t_{i,2}-t_{i,1})}\right\}}{\left\{1-\rho^{2(t_{i,2}-t_{i,1})}\right\}^{\frac{1}{2}}}, \ldots, \frac{\left\{1-\rho^{(t_{i,n_i}-t_{i,n_i-1})}\right\}}{\left\{1-\rho^{2(t_{i,n_i}-t_{i,n_i-1})}\right\}^{\frac{1}{2}}}\right]^{\prime}$$

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

$$\widehat{\omega}^2_i = g_i^{\prime} g_i \widehat{\sigma}^2_{\mu} + \widehat{\sigma}^2_{\epsilon}$$

With these estimates in hand, we can transform the data via

$$z^{**}_{it_{ij}} = z^{*}_{it_{ij}} - \widehat{\theta}_i g_{ij} \frac{\sum_{s=1}^{n_i} g_{is} z^{*}_{it_{is}}}{\sum_{s=1}^{n_i} g^2_{is}}$$

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

$$d_1 = \frac{\sum_{i=1}^{N}\sum_{j=1}^{n_i}\{\widetilde{z}_{it_{i,j-1}} - \widetilde{z}_{it_{ij}} I(t_{ij} - t_{i,j-1} = 1)\}^2}{\sum_{i=1}^{N}\sum_{j=1}^{n_i}\widetilde{z}^2_{it_{ij}}}$$

$$d_2 = \frac{\sum_{i=1}^{N}\sum_{j=1}^{n_i-1}\widetilde{z}^2_{it_{i,j-1}}\{1 - I(t_{ij} - t_{i,j-1} = 1)\}^2}{\sum_{i=1}^{N}\sum_{j=1}^{n_i}\widetilde{z}^2_{it_{ij}}}$$

$$d_3 = \frac{\sum_{i=1}^{N}\widetilde{z}^2_{it_{i1}}}{\sum_{i=1}^{N}\sum_{j=1}^{n_i}\widetilde{z}^2_{it_{ij}}}$$

$$d_4 = \frac{\sum_{i=1}^{N}\widetilde{z}^2_{it_{in_i}}}{\sum_{i=1}^{N}\sum_{j=1}^{n_i}\widetilde{z}^2_{it_{ij}}}$$

$I()$ is the indicator function that takes the value of 1 if the condition is true and 0 otherwise. The $\widetilde{z}_{it_{i,j-1}}$ 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, Department of Economics, Syracuse University, for his helpful comments.

References

Baltagi, B. H. 2008. *Econometric Analysis of Panel Data*. 4th ed. New York: Wiley.

——. 2009. *A Companion to Econometric Analysis of Panel Data*. 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.

Baltagi, B. H., and P. X. Wu. 1999. Unequally spaced panel data regressions with AR(1) disturbances. *Econometric Theory* 15: 814–823.

Bhargava, A., L. Franzini, and W. Narendranathan. 1982. Serial correlation and the fixed effects model. *Review of Economic Studies* 49: 533–549.

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.

Hsiao, C. 2003. *Analysis of Panel Data*. 2nd ed. New York: Cambridge University Press.

Mundlak, Y. 1978. On the pooling of time series and cross section data. *Econometrica* 46: 69–85.

Sosa-Escudero, W., and A. K. Bera. 2008. Tests for unbalanced error-components models under local misspecification. *Stata Journal* 8: 68–78.

Also see

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

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

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

[XT] **xtgls** — Fit panel-data models by using GLS

[XT] **xtreg** — Fixed-, between-, and random-effects and population-averaged linear models

[TS] **newey** — Regression with Newey–West standard errors

[TS] **prais** — Prais–Winsten and Cochrane–Orcutt regression

[U] **20 Estimation and postestimation commands**

Title

xtregar postestimation — Postestimation tools for xtregar

Description

The following postestimation commands are available after `xtregar`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
*`estat`	AIC, BIC, VCE, and estimation sample summary
`estimates`	cataloging estimation results
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`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 coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

*`estat ic` is not appropriate after `xtregar, re`.

See the corresponding entries in the *Base Reference Manual* for details.

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.

Menu

Statistics > Postestimation > Predictions, residuals, etc.

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}.

Methods and formulas

All postestimation commands listed above are implemented as ado-files.

Also see

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

[U] **20 Estimation and postestimation commands**

Title

xtset — Declare data 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 periodicity of *timevar*
`noquery`	suppress summary calculations and output

`noquery` is not shown in the dialog box.

unitoptions	Description
(default)	*timevar*'s units to be obtained 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  secs  sec
minutes  mins  min
hours    hour
days     day
weeks    week
```

and for all other %t *timevars* are

```
days   day
weeks  week
```

Menu

Statistics > Longitudinal/panel data > Setup and utilities > Declare dataset to be panel data

Description

xtset declares the data in memory to be a panel. You must xtset your data before you can use the other xt commands. If you save your data after xtset, the data will be remembered to be a panel and you will not have to xtset again.

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.

xtset without arguments—xtset—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 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.

Options

unitoptions `clocktime`, `daily`, `weekly`, `monthly`, `quarterly`, `halfyearly`, `yearly`, `generic`, and `format(%`*fmt*`)` specify the units in which *timevar* is recorded, if *timevar* is specified.

timevar will often simply 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 still 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)`

Understand that *timevar* is not required to be a `%t` variable; it can be any variable of your own concocting so long as it takes on 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 periodicity of *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 periodicity. The *unitoptions* set the units. `delta()` sets the periodicity. 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

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.

❑ 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 xtsetting your data is sufficient to allow you to use the ts commands and tssetting 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 *panelvar*.

❑

▷ Example 1

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 U.S. 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
varlist:  state:  string variable not allowed
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

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 http://www.stata-press.com/data/r12/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)
```

	id	year	wage	L.wage
1.	1	1977	13.1516	.
2.	1	1978	12.3018	13.1516
		(*output omitted*)		
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
		(*output omitted*)		
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

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 http://www.stata-press.com/data/r12/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 + tm(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 room reservations 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 translation** 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, however, 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 now 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 periodicity 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 periodicity 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 periodicity, and `xtset` will complain.

By the way, it is important only that clock times (`%tc` and `%tC` variables) be stored as `double`s. The other date values `%td`, `%tw`, `%tm`, `%tq`, `%th`, and `%ty` are small enough that they can safely be stored as `float`s, although forgetting and storing them as `double`s 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 pm), which is then followed by the usual 00:00:00 (12:00:00 am). 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.

❑

Saved results

xtset saves 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

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; a set of panels are strongly balanced if they all have the same time values, otherwise balanced if same number of time values, otherwise unbalanced

Methods and formulas

xtset is implemented as an ado-file.

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

Title

xtsum — 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` is allowed; see [D] **by**.

Menu

Statistics > Longitudinal/panel data > Setup and utilities > Summarize xt data

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.

Remarks

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 http://www.stata-press.com/data/r12/nlswork
(National Longitudinal Survey.  Young Women 14-26 years of age in 1968)
. summarize hours
    Variable |      Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
       hours |    28467    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

Methods and formulas

xtsum is implemented as an ado-file.

Also see

[XT] **xtdescribe** — Describe pattern of xt data

[XT] **xttab** — Tabulate xt data

Title

xttab — Tabulate xt data

Syntax

`xttab` *varname* [*if*]

`xttrans` *varname* [*if*] [, `freq`]

A panel variable must be specified; use `xtset`; see [XT] **xtset**.
`by` is allowed with `xttab` and `xttrans`; see [D] **by**.

Menu

xttab

Statistics > Longitudinal/panel data > Setup and utilities > Tabulate xt data

xttrans

Statistics > Longitudinal/panel data > Setup and utilities > Report transition probabilities

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).

Option

Main

`freq`, allowed with `xttrans` only, specifies that frequencies as well as transition probabilities be displayed.

Remarks

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 http://www.stata-press.com/data/r12/nlswork
(National Longitudinal Survey.  Young Women 14-26 years of age in 1968)
. xttab msp
```

msp	Overall Freq.	Overall Percent	Between Freq.	Between Percent	Within 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 69.75% 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
1	20180	70.72	3329	70.66	100.00
2	8051	28.22	1325	28.13	100.00
3	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	1,866	9,563
	80.49	19.51	100.00
1	1,133	13,100	14,233
	7.96	92.04	100.00
Total	8,830	14,966	23,796
	37.11	62.89	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)
```

❑

Saved results

`xttab` saves the following in `r()`:

Scalars

`r(n)`	number of panels

Matrices

`r(results)`	results matrix

Methods and formulas

`xttab` and `xttrans` are implemented as ado-files.

Also see

[XT] **xtdescribe** — Describe pattern of xt data

[XT] **xtsum** — Summarize xt data

Title

xttobit — Random-effects tobit models

Syntax

```
xttobit depvar [indepvars] [if] [in] [weight] [, options]
```

options	Description
Model	
noconstant	suppress constant term
ll(*varname* | #)	left-censoring variable/limit
ul(*varname* | #)	right-censoring variable/limit
offset(*varname*)	include *varname* in model with coefficient constrained to 1
constraints(*constraints*)	apply specified linear constraints
collinear	keep collinear variables
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
noskip	perform overall model test as a likelihood-ratio test
nocnsreport	do not display constraints
display_options	control column formats, row spacing, line width, and display of omitted variables and base and empty cells
Integration	
intmethod(*intmethod*)	integration method; *intmethod* may be mvaghermite, aghermite, or ghermite; default is intmethod(mvaghermite)
intpoints(#)	use # quadrature points; default is intpoints(12)
Maximization	
maximize_options	control the maximization process; seldom used
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 and statsby are allowed; see [U] **11.1.10 Prefix commands**.

iweights are allowed; 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.

Menu

Statistics > Longitudinal/panel data > Censored outcomes > Tobit regression (RE)

Description

`xttobit` fits random-effects tobit models. There is no command for a parametric conditional fixed-effects model, as there does not exist a sufficient statistic allowing the fixed effects to be conditioned out of the likelihood. Honoré (1992) has developed a semiparametric estimator for fixed-effect tobit models. Unconditional fixed-effects tobit models may be fit with the `tobit` command with indicator variables for the panels; the indicators can be created with the factor-variable syntax described in [U] **11.4.3 Factor variables**. However, unconditional fixed-effects estimates are biased.

Options

Model

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

`ll(`*varname*|*#*`)` and `ul(`*varname*|*#*`)` indicate the censoring points. You may specify one or both. `ll()` indicates the lower limit for left-censoring. Observations with *depvar* $\leq$ `ll()` are left-censored, observations with *depvar* $\geq$ `ul()` are right-censored, and remaining observations are not censored.

`offset(`*varname*`)`, `constraints(`*constraints*`)`, `collinear`; see [R] **estimation options**.

SE

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory and that use bootstrap or jackknife methods; 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.

`noskip`; see [R] **estimation options**.

`nocnsreport`; see [R] **estimation options**.

display_options: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `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 option is available with `xttobit` but is not shown in the dialog box:

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

Remarks

Consider the linear regression model with panel-level random effects

$$y_{it} = \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i + \epsilon_{it}$$

for $i = 1, \ldots, n$ panels, where $t = 1, \ldots, n_i$. 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 $y_{it} \leq y_{it}^o$. If they are right-censored, all that is known is that $y_{it} \geq y_{it}^o$. 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

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. We use the `intpoints(25)` option to increase the number of integration points to 25 from 12, which aids convergence of this model.

```
. use http://www.stata-press.com/data/r12/nlswork3
(National Longitudinal Survey.  Young Women 14-26 years of age in 1968)
. xttobit ln_wage union age grade not_smsa south##c.year, ul(1.9) intpoints(25)
> tobit
```

(*output omitted*)

```
Random-effects tobit regression                 Number of obs      =     19224
Group variable: idcode                          Number of groups   =      4148
Random effects u_i ~ Gaussian                   Obs per group: min =         1
                                                               avg =       4.6
                                                               max =        12
                                                Wald chi2(7)       =   2924.91
Log likelihood  = -6814.4638                    Prob > chi2        =    0.0000
```

ln_wage	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
union	.1430525	.0069719	20.52	0.000	.1293878	.1567172
age	.009913	.0017517	5.66	0.000	.0064797	.0133463
grade	.0784843	.0022767	34.47	0.000	.074022	.0829466
not_smsa	-.1339973	.0092061	-14.56	0.000	-.1520409	-.1159536
1.south	-.3507181	.0695557	-5.04	0.000	-.4870447	-.2143915
year	-.0008283	.0018372	-0.45	0.652	-.0044291	.0027725
south#c.year						
1	.0031938	.0008606	3.71	0.000	.0015071	.0048805
_cons	.5101968	.1006681	5.07	0.000	.312891	.7075025
/sigma_u	.3045995	.0048346	63.00	0.000	.2951239	.314075
/sigma_e	.2488682	.0018254	136.34	0.000	.2452904	.2524459
rho	.599684	.0084097			.5831174	.6160733

```
Likelihood-ratio test of sigma_u=0: chibar2(01)= 6650.63 Prob>=chibar2 = 0.000

  Observation summary:         0  left-censored observations
                           12334     uncensored observations
                            6890 right-censored observations
```

The output 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.

◁

❑ 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.

❑

Saved results

xttobit saves 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(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)	significance
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(), if specified
e(ulopt)	contents of ul(), if specified
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(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(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 estimator

Functions

`e(sample)`	marks estimation sample

Methods and formulas

`xttobit` is implemented as an ado-file.

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

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

where

$$F(y^o_{it}, \Delta_{it}) = \begin{cases} \left(\sqrt{2\pi}\sigma_\epsilon\right)^{-1} e^{-(y^o_{it}-\Delta_{it})^2/(2\sigma_\epsilon^2)} & \text{if } y^o_{it} \in C \\ \Phi\left(\frac{y^o_{it}-\Delta_{it}}{\sigma_\epsilon}\right) & \text{if } y^o_{it} \in L \\ 1 - \Phi\left(\frac{y^o_{it}-\Delta_{it}}{\sigma_\epsilon}\right) & \text{if } y^o_{it} \in R \end{cases}$$

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

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

$$\equiv \int_{-\infty}^{\infty} g(y^o_{it}, x_{it}, \nu_i) d\nu_i$$

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

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

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

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

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

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

Letting

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

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

and

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

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.

One can instead use the adaptive quadrature method of Liu and Pierce (1994), the `intmethod(aghermite)` option, which uses the mode and curvature of the mode as approximations for the mean and variance. We take the integrand

$$g(y_{it}^o, x_{it}, \nu_i) = \frac{e^{-\nu_i^2/2\sigma_\nu^2}}{\sqrt{2\pi}\sigma_\nu} \left\{\prod_{t=1}^{n_i} F(y_{it}^o, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i)\right\}$$

and find α_i the mode of $g(y^o_{it}, x_{it}, \nu_i)$. We calculate

$$\gamma_i = -\frac{\partial^2}{\partial \nu_i^2} \log\{g(y^o_{it}, x_{it}, \nu_i)\}\big|_{\nu_i=\alpha_i}$$

Then

$$\int_{-\infty}^{\infty} g(y^o_{it}, x_{it}, \nu_i) d\nu_i \approx \left(\frac{2}{\gamma_i}\right)^{1/2} \sum_{m=1}^{M} w^*_m \exp\{(a^*_m)^2\} g\left\{y^o_{it}, x_{it}, \left(\frac{2}{\gamma_i}\right)^{1/2} a^*_m + \alpha_i\right\}$$

This adaptation is performed on the first iteration only; that is, the α_i and γ_i are calculated once at the first iteration and then held constant throughout later iterations.

The log likelihood can also be calculated by nonadaptive Gauss–Hermite quadrature, the `intmethod(ghermite)` option:

$$\begin{aligned} L &= \sum_{i=1}^{n} w_i \log\Big\{\Pr(y_{i1}, \ldots, y_{in_i} | \mathbf{x}_{i1}, \ldots, \mathbf{x}_{in_i})\Big\} \\ &\approx \sum_{i=1}^{n} w_i \log\left[\frac{1}{\sqrt{\pi}} \sum_{m=1}^{M} w^*_m \prod_{t=1}^{n_i} F\left\{y^o_{it}, \mathbf{x}_{it}\boldsymbol{\beta} + \sqrt{2}\sigma_\nu a^*_m\right\}\right] \end{aligned}$$

All three 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

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

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.

References

Honoré, B. E. 1992. Trimmed LAD and least squares estimation of truncated and censored regression models with fixed effects. *Econometrica* 60: 533–565.

Liu, Q., and D. A. Pierce. 1994. A note on Gauss–Hermite quadrature. *Biometrika* 81: 624–629.

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, Series C* 31: 214–225.

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.

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

Also see

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

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

[XT] **xtintreg** — Random-effects interval-data regression models

[XT] **xtreg** — Fixed-, between-, and random-effects and population-averaged linear models

[R] **tobit** — Tobit regression

[U] **20 Estimation and postestimation commands**

Title

xttobit postestimation — Postestimation tools for xttobit

Description

The following postestimation commands are available after `xttobit`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat`	AIC, BIC, VCE, and estimation sample summary
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`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 coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

See the corresponding entries in the *Base Reference Manual* for details.

Syntax for predict

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

statistic	Description
Main	
`xb`	linear prediction assuming $\nu_i = 0$, the default
`stdp`	standard error of the linear prediction
`stdf`	standard error of the linear forecast
`pr0(`*a*`,`*b*`)`	$\Pr(a < y < b)$ assuming $\nu_i = 0$
`e0(`*a*`,`*b*`)`	$E(y \mid a < y < b)$ assuming $\nu_i = 0$
`ystar0(`*a*`,`*b*`)`	$E(y^*)$, $y^* = \max\{a, \min(y, b)\}$ assuming $\nu_i = 0$

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**.

Menu

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

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

`stdp` calculates the standard error of the 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 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**.

`pr0(`*a*`,`*b*`)` calculates estimates of $\Pr(a < y < b \mid \mathbf{x} = \mathbf{x}_{it}, \nu_i = 0)$, which is the probability that y would be observed in the interval (a, b), given the current values of the predictors, $\mathbf{x}_{it}$, and given a zero random effect. 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;
`pr0(20,30)` calculates $\Pr(20 < y < 30)$;
`pr0(`*lb*`,`*ub*`)` calculates $\Pr(lb < y < ub)$; and
`pr0(20,`*ub*`)` calculates $\Pr(20 < y < ub)$.

a missing ($a \geq .$) means $-\infty$; `pr0(.,30)` calculates $\Pr(-\infty < y < 30)$;
`pr0(`*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 $+\infty$; `pr0(20,.)` calculates $\Pr(+\infty > y > 20)$;
`pr0(20,`*ub*`)` calculates $\Pr(+\infty > y > 20)$ in observations for which $ub \geq .$
(and calculates $\Pr(20 < y < ub)$ elsewhere).

`e0(`*a*`,`*b*`)` calculates estimates of $E(y \mid a < y < b, \mathbf{x} = \mathbf{x}_{it}, \nu_i = 0)$, 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 `pr0()`.

`ystar0(`*a*`,`*b*`)` calculates estimates of $E(y^* \mid \mathbf{x} = \mathbf{x}_{it}, \nu_i = 0)$, 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 `pr0()`.

`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 $\mathbf{x}_{it}\boldsymbol{\beta} + \text{offset}_{it}$.

Methods and formulas

All postestimation commands listed above are implemented as ado-files.

Also see

[XT] **xttobit** — Random-effects tobit models

[U] **20 Estimation and postestimation commands**

Title

xtunitroot — 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**.

Menu

Statistics > Longitudinal/panel data > Unit-root tests

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.

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.

`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:

$$maxlags = \text{int}\left(3.21T^{1/3}\right)$$

where T is the number of observations per panel. This is the default.

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.

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.

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.

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.

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.

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.

robust requests a variant of the test that is robust to cross-sectional dependence.

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.

IPS_options

trend includes a linear time trend in the model that describes the process by which the series is generated.

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 $\widetilde{t}\text{-}bar$ and $Z_{\widetilde{t}\text{-}bar}$ 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 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.

`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.

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.

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**.

`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.

Hadri_options

`trend` includes a linear time trend in the model that describes the process by which the series is generated.

`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* $\left[\#\right]$. 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.

`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.

Remarks

Remarks are presented under the following headings:

Overview

We consider a simple panel-data model with a first-order autoregressive component:

$$y_{it} = \rho_i y_{i,t-1} + \mathbf{z}'_{it}\gamma_i + \epsilon_{it} \tag{1}$$

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 $\mathbf{z}'_{it}\gamma_i$ represents panel-specific means (fixed effects). If `trend` is specified, $\mathbf{z}'_{it} = (1, t)$ so that $\mathbf{z}'_{it}\gamma_i$ represents panel-specific means and linear time trends. For tests that allow it, specifying `noconstant` omits the $\mathbf{z}'_{it}\gamma_i$ term. The Im–Pesaran–Shin (`xtunitroot ips`), Fisher-type (`xtunitroot fisher`), and Hadri LM (`xtunitroot hadri`) 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 $H_0\colon \rho_i = 1$ for all i versus the alternative $H_a\colon \rho_i < 1$. 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

$$\Delta y_{it} = \phi_i y_{i,t-1} + \mathbf{z}'_{it}\gamma_i + \epsilon_{it} \tag{1′}$$

so that the null hypothesis is then $H_0\colon \phi_i = 0$ for all i versus the alternative $H_a\colon \phi_i < 0$.

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 (2008, 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 $(T, N) \rightarrow_{\text{seq}} \infty$. 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:

$$\Delta y_{it} = \phi y_{i,t-1} + \mathbf{z}'_{it}\gamma_i + \sum_{j=1}^{p} \theta_{ij} \Delta y_{i,t-j} + u_{it} \tag{2}$$

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 (2008, 280) mentions that the requirement $N/T \to 0$ 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 tradeable 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 tradeable 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 Organization for Economic Cooperation 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 http://www.stata-press.com/data/r12/pennxrate
. xtunitroot llc lnrxrate if g7, lags(aic 10)
Levin-Lin-Chu unit-root test for lnrxrate
------------------------------------------

Ho: 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
------------------------------------------
Ho: 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

$$y_{it} = \rho y_{i,t-1} + \mathbf{z}_{it}'\gamma_i + \epsilon_{it} \tag{4}$$

where the term $\mathbf{z}_{it}'\gamma_i$ 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 $H_0\colon \rho = 1$ 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
------------------------------------------------------------------------------
Ho: 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 $N/T \to 0$, 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
------------------------------------

Ho: 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

$$\Delta y_{it} = \phi_i y_{i,t-1} + \mathbf{z}'_{it}\gamma_i + \epsilon_{it} \tag{5}$$

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
------------------------------------------
Ho: 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 $\widetilde{t\text{-}bar}_{NT}$ 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, $Z_{\widetilde{t\text{-}bar}}$, 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 $Z_{\widetilde{t\text{-}bar}}$ statistic corresponds to $\widetilde{t}\text{-}bar_{NT}$, 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 saved results as `r(zt)`.

❑

When serial correlation is present, we augment the Dickey–Fuller regression with further lags of the dependent variable:

$$\Delta y_{it} = \phi_i y_{i,t-1} + \mathbf{z}'_{it}\gamma_i + \sum_{j=1}^{p} \Delta y_{i,t-j} + \epsilon_{it} \tag{6}$$

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 $T \to \infty$ followed by $N \to \infty$. 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
------------------------------------------------
Ho: 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 in order 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
------------------------------------------
Ho: 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 trade-off 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 $T \to \infty$ followed by $N \to \infty$, 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 $T \to \infty$ followed by $N \to \infty$. 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
------------------------------

Ho: 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.

◁

Saved results

`xtunitroot llc` saves 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 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` saves 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` saves 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` saves 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` saves 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` saves 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:

`xtunitroot` is implemented as an ado-file.

We consider a simple panel-data model with a first-order autoregressive component:

$$y_{it} = \rho_i y_{i,t-1} + \mathbf{z}'_{it}\gamma_i + \epsilon_{it}$$

where $i = 1, \ldots, N$ indexes panels and $t = 1, \ldots, T$ indexes time. For the IPS, Fisher-type, and Hadri LM tests, we instead have $t = 1, \ldots, T_i$, 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 $\mathbf{z}'_{it}\gamma_i$ represents panel-specific means (fixed effects). If `noconstant` is specified, $\mathbf{z}'_{it}\gamma_i$ vanishes. If `trend` is specified, $\mathbf{z}'_{it} = (1, t)$ so that $\mathbf{z}'_{it}\gamma_i$ represents panel-specific means and linear time trends.

Levin–Lin–Chu test

The starting point for the LLC test is the regression model

$$\Delta y_{it} = \phi y_{i,t-1} + \mathbf{z}'_{it}\gamma_i + \sum_{j=1}^{p_i} \theta_{ij}\Delta y_{i,t-j} + u_{it} \tag{7}$$

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 $i = 1, \ldots, N$. Otherwise, we fit (7) for each panel individually for lags $1 \ldots p^{\max}$ 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

$$\widehat{e}_{it} = \Delta y_{it} - \sum_{j=1}^{p_i} \widehat{\theta}_{ij}\Delta y_{ij} - \mathbf{z}'_{it}\widehat{\gamma}_i \tag{8}$$

and

$$\widehat{v}_{i,t-1} = y_{i,t-1} - \sum_{j=1}^{p_i} \widetilde{\theta}_{ij}\Delta y_{ij} - \mathbf{z}_{it}\widetilde{\gamma}_i \tag{9}$$

To control for panel-level heterogeneity, compute

$$\widetilde{e}_{it} = \widehat{e}_{it}/\widehat{\sigma}_{\epsilon i} \qquad \text{and} \qquad \widetilde{v}_{i,t-1} = \widehat{v}_{i,t-1}/\widehat{\sigma}_{\epsilon i}$$

where

$$\widehat{\sigma}^2_{\epsilon i} = \frac{1}{T - p_i - 1}\sum_{t=p_i}^{T}\left(\widehat{e}_{it} - \widehat{\delta}_i\widehat{v}_{i,t-1}\right)^2$$

and $\widehat{\delta}_i$ is the OLS coefficient from a regression of $\widehat{e}_{it}$ on $\widehat{v}_{i,t-1}$. 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

$$\widehat{\sigma}^2_{yi} = \frac{1}{T-1}\sum_{t=2}^{T}\Delta y_{it}^2 + \frac{2}{T-1}\sum_{j=1}^{m}K(j,m)\left(\sum_{t=j+2}^{T}\Delta y_{it}\Delta y_{i,t-j}\right)$$

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 $\widehat{\sigma}^2_{yi}$ 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 $\widehat{s}_i = \widehat{\sigma}_{yi}/\widehat{\sigma}_{\epsilon i}$, and denote $\widehat{S}_N = N^{-1}\sum_i \widehat{s}_i$.

In the third step, we run the OLS regression

$$\widetilde{e}_{it} = \delta\widetilde{v}_{i,t-1} + \widetilde{\epsilon}_{it}$$

Called the "Basic test statistic" in the output of `xtunitroot llc` is the standard t statistic for δ computed as

$$t_\delta = \widehat{\delta}/\text{se}_{\widehat{(\delta)}}$$

where

$$\text{se}_{(\widehat{\delta})} = \widehat{\sigma}_{\widetilde{\epsilon}} \left(\sum_{i=1}^{N} \sum_{t=p_i+2}^{T} \widetilde{v}_{i,t-1}^2 \right)^{-1/2}$$

$$\widehat{\sigma}_{\widetilde{\epsilon}}^2 = \frac{1}{N\widetilde{T}} \sum_{i=1}^{N} \sum_{t=p_i+2} \left(\widetilde{e}_{it} - \delta \widetilde{v}_{i,t-1} \right)^2$$

and $\widetilde{T} = T - \overline{p} - 1$ with $\overline{p}$ the average of $p_1, \ldots, p_N$.

The adjusted test statistic is then computed as

$$t_\delta^* = \frac{t_\delta - N\widetilde{T}\widehat{S}_N \text{se}_{(\widehat{\delta})} \mu_{\widetilde{T}}^*}{\sigma_{\widetilde{T}}^*}$$

where $\mu_{\widetilde{T}}^*$ and $\sigma_{\widetilde{T}}^*$ 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 $\widehat{\rho}$ the least-squares estimate of ρ.

HT show that $\sqrt{N}(\widehat{\rho} - \mu) \xrightarrow{D} N(0, \sigma^2)$ 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

$$y_{it} = \mathbf{z}_{it}'\gamma_i + x_{it}$$

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 $\mathbf{u}^{\ell}_i$ 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 $s = 1, \ldots, T-p-1$. 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

$$\Delta y_{it} = \phi_i y_{i,t-1} + \mathbf{z}_{it}' \gamma_i + \epsilon_{it}$$

where ϵ_{it} is independently distributed normal for all i and t with panel-specific variance σ_i^2. Denote $\Delta \mathbf{y}_i = (\Delta y_{i2}, \ldots, \Delta y_{iT})'$ and $\mathbf{y}_{i,-1} = (y_{i1}, \ldots, y_{i,T-1})'$. 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

$$\widetilde{t}\text{-}bar_{NT} = \frac{1}{N} \sum_{i=1}^{N} \widetilde{t}_{iT}$$

where

$$\widetilde{t}_{iT} = \frac{\Delta \mathbf{y}_i' \mathbf{M}_\tau \mathbf{y}_{i,-1}}{\widetilde{\sigma}_{iT} \left(\mathbf{y}_{i,-1}' \mathbf{M}_\tau \mathbf{y}_{i,-1}\right)^{1/2}}$$

and

$$\widetilde{\sigma}_{iT}^2 = \frac{\Delta \mathbf{y}_i' \mathbf{M}_\tau \Delta \mathbf{y}_i}{T-1}$$

Also

$$t\text{-}bar_{NT} = \frac{1}{N}\sum_{i=1}^{N} t_{iT}$$

where

$$t_{iT} = \frac{\Delta\mathbf{y}_i'\mathbf{M}_\tau\mathbf{y}_{i,-1}}{\widehat{\sigma}_{iT}\left(\mathbf{y}_{i,-1}'\mathbf{M}_\tau\mathbf{y}_{i,-1}\right)^{1/2}}$$

and

$$\widetilde{\sigma}_{iT}^2 = \frac{\Delta\mathbf{y}_i'\mathbf{M}_{X_i}\Delta\mathbf{y}_i}{T-1}$$

Now

$$Z_{\widetilde{t}\text{-}bar} = \frac{\sqrt{N}\left\{\widetilde{t}\text{-}bar_{NT} - N^{-1}\sum_{i=1}^{N} E(\widetilde{t}_{T_i})\right\}}{\sqrt{N^{-1}\sum_i \mathrm{Var}(\widetilde{t}_{T_i})}}$$

where $E(\widetilde{t}_{T_i})$ and $\mathrm{Var}(\widetilde{t}_{T_i})$ are obtained by linearly interpolating the values shown in IPS (2003, table 1). $Z_{\widetilde{t}\text{-}bar}$ has a standard normal limiting distribution for fixed T and $N \to \infty$; very negative values cast doubt on H_0. Similarly,

$$Z_{t\text{-}bar} = \frac{\sqrt{N}\left\{t\text{-}bar_{NT} - N^{-1}\sum_i E(t_{T_i})\right\}}{\sqrt{N^{-1}\sum_i \mathrm{Var}(t_{T_i})}}$$

If the `lags()` option is specified, then we fit the ADF regressions

$$\Delta y_{it} = \phi_i y_{i,t-1} + \mathbf{z}_{it}'\gamma_i + \sum_{j=1}^{p_i}\rho_{ij}\Delta y_{i,t-j} + \epsilon_i t$$

In matrix form, we can write this more compactly as

$$\Delta\mathbf{y}_i = \phi_i\mathbf{y}_{i,-1} + \mathbf{Q}_i\theta_i + \epsilon_i$$

where $\mathbf{Q}_i = (\tau_t, \Delta\mathbf{y}_{i,-1}, \ldots, \Delta\mathbf{y}_{i,-p_i})$ and $\theta_i = (\alpha_i, \rho_{i1}, \ldots, \rho_{ip_i})'$. Then

$$t\text{-}bar_{NT} = \frac{1}{N}\sum_{i=1}^{N} t_{iT}(p_i)$$

where

$$t_{iT}(p_i) = \frac{\sqrt{T-p_i-2}(\mathbf{y}_{i,-1}'\mathbf{M}_{Q_i}\Delta\mathbf{y}_i)}{(\mathbf{y}_{i,-1}'\mathbf{M}_{Q_i}\mathbf{y}_{i,-1})^{1/2}(\Delta\mathbf{y}_{i,-1}'\mathbf{M}_{Q_i}\Delta\mathbf{y}_{i,-1})^{1/2}}$$

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,

$$\mathbf{W}_{t\text{-}bar}(p) = \frac{\sqrt{N}\left[t\text{-}bar_{NT} - N^{-1}\sum_{i=1}^{N} E\left\{t_{iT}(p_i)\right\}\right]}{\sqrt{N^{-1}\sum_{i=1}^{N}\mathrm{Var}\left\{t_{iT}(p_i)\right\}}}$$

where $E\left\{t_{iT}(p_i)\right\}$ and $\mathrm{Var}\left\{t_{iT}(p_i)\right\}$ 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 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

$$\widehat{\mathrm{LM}} = \frac{\frac{1}{N}\sum_i \frac{1}{T^2}\sum_t S_{it}^2}{\widehat{\sigma}_\epsilon^2} \tag{10}$$

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

$$Z = \frac{\sqrt{N}\left(\widehat{\mathrm{LM}} - \mu\right)}{\sigma}$$

where $\mu = 1/15$ and $\sigma = 11/6300$ if `trend` is specified and $\mu = 1/6$ and $\sigma = 1/45$ 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:

$$\widehat{\sigma}_{\epsilon,i}^2 = \sum_{t=1}^{T}\widehat{\epsilon}_{it}^2$$

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 and Christopher F. Baum for the `ipshin` and `levinlin` commands, Christopher Baum for the `hadrilm` command, and Scott Merryman for the `xtfisher` command. We also gratefully acknowledge Jörg Breitung for useful discussions when implementing his tests. These contributions were beneficial during the development of `xtunitroot`.

References

Baltagi, B. H. 2008. *Econometric Analysis of Panel Data*. 4th ed. New York: Wiley.

Banerjee, A., M. Marcellino, and C. Osbat. 2005. Testing for PPP: Should we use panel methods? *Empirical Economics* 30: 77–91.

Breitung, J. 2000. The local power of some unit root tests for panel data. In *Advances in Econometrics, Volume 15: Nonstationary Panels, Panel Cointegration, and Dynamic Panels*, ed. B. H. Baltagi, 161–178. Amsterdam: JAI Press.

Breitung, J., and S. Das. 2005. Panel unit root tests under cross-sectional dependence. *Statistica Neerlandica* 59: 414–433.

Choi, I. 2001. Unit root tests for panel data. *Journal of International Money and Finance* 20: 249–272.

Hadri, K. 2000. Testing for stationarity in heterogeneous panel data. *Econometrics Journal* 3: 148–161.

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.

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. http://pwt.econ.upenn.edu/php_site/pwt_index.php.

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.

Im, K. S., M. H. Pesaran, and Y. Shin. 2003. Testing for unit roots in heterogeneous panels. *Journal of Econometrics* 115: 53–74.

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.

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.

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.

Newey, W. K., and K. D. West. 1994. Automatic lag selection in covariance matrix estimation. *Review of Economic Studies* 61: 631–653.

Nickell, S. J. 1981. Biases in dynamic models with fixed effects. *Econometrica* 49: 1417–1426.

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.

Oh, K.-Y. 1996. Purchasing power parity and unit root tests using panel data. *Journal of International Money and Finance* 15: 405–418.

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.

——. 2000. Nonstationary panel data analysis: An overview of some recent developments. *Econometric Reviews* 19: 263–286.

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

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

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

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

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.

BLUPs. BLUPs are best linear unbiased predictions of either random effects or linear combinations of random effects. In linear models containing random effects, these effects are not estimated directly but instead are integrated out of the estimation. Once the fixed effects and variance components have been estimated, you can use these estimates to predict group-specific random effects. These predictions are called BLUPs because they are unbiased and have minimal mean squared error among all linear functions of the response.

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.

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.

crossed-effects model. A crossed-effects model is a mixed model in which the levels of random effects are not nested. A simple crossed-effects model for cross-sectional time-series data would contain a random effect to control for panel-specific variation and a second random effect to control for time-specific random variation. Rather than being nested within panel, in this model a random effect due to a given time is the same for all panels.

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 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 (GLM). 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. GLM encompasses many common models, including linear, probit, and Poisson regression.

hierarchical model. A hierarchical model is one in which successively more narrowly defined groups are nested within larger groups. For example, in a hierarchical model, patients may be nested within doctors who are in turn nested within the hospital at which they practice.

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 GLM, 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*.

mixed model. A mixed model contains both fixed and random effects. The fixed effects are estimated directly, whereas the random effects are summarized according to their (co)variances. Mixed models are used primarily to perform estimation and inference on the regression coefficients in the presence of complicated within-panel correlation structures induced by multiple levels of grouping.

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.

one-level model. A one-level mixed model is a mixed model with one level of random variation. Suppose that you have a panel dataset consisting of patients at hospitals; a one-level model would contain a set of random effects "at the hospital level" to control for hospital-specific random variation.

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-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.

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.

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 an integral. Two types of quadrature commonly used in fitting panel-data models are Gaussian and Gauss–Hermite quadrature.

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.

REML (restricted maximum likelihood). REML is a method of fitting linear mixed models that involves transforming out the fixed effects so as to focus solely on variance-component estimation.

restricted maximum likelihood. See *REML*.

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.

two-level model. A two-level mixed model is a mixed model with two levels of random variation. Suppose that you have a dataset consisting of patients overseen by doctors at hospitals, and each doctor practices at one hospital. Then a two-level model would contain a set of random effects to control for hospital-specific variation and a second set of random effects to control for doctor-specific random variation.

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*.

variance components. In a mixed model, the variance components refer to the variances and covariances of the various random effects.

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

This is the subject and author index for the *Longitudinal-Data/Panel-Data Reference Manual*. Readers interested in topics other than cross-sectional time-series should see the combined subject index (and the combined author index) in the *Quick Reference and Index*.

Semicolons set off the most important entries from the rest. Sometimes no entry will be set off with semicolons, meaning that all entries are equally important.

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z