xtprobit — Random-effects and population-averaged probit models

Description

`xtprobit` fits random-effects and population-averaged probit models for a binary dependent variable. The probability of a positive outcome is assumed to be determined by the standard normal cumulative distribution function.

Quick start

Random-effects probit model of `y` as a function of `x1`, `x2`, and indicators for levels of categorical variable `a` using `xtset` data

```
xtprobit y x1 x2 i.a
```

Population-averaged model with robust standard errors

```
xtprobit y x1 x2 i.a, pa vce(robust)
```

Same as above, but specify an autoregressive correlation structure of order 1

```
xtprobit y x1 x2 i.a, pa vce(robust) corr(ar 1)
```

Random-effects model with cluster–robust standard errors for panels nested within `cvar`

```
xtprobit y x1 x2 i.a, vce(cluster cvar)
```

Menu

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

Syntax

Random-effects (RE) model

xtprobit *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, re *RE_options*]

Population-averaged (PA) model

xtprobit *depvar* [*indepvars*] [*if*] [*in*] [*weight*] , pa [*PA_options*]

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

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

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

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

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

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

by, collect, mi estimate, and statsby are allowed; see [U] **11.1.10 Prefix commands**. bayes is allowed for the random-effects model. For more details, see [BAYES] **bayes: xtprobit**. fp is allowed for the random-effects model.

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

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

collinear and coeflegend do not appear in the dialog box.

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

Options for RE model

Model

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

re requests the random-effects estimator. re is the default if neither re nor pa is specified.

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

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

SE/Robust

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

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

Reporting

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

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

Integration

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

Maximization

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

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

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

Options for PA model

Model

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

pa requests the population-averaged estimator.

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

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

Correlation

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

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

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

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

SE/Robust

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

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

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

Reporting

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

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

Optimization

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

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

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

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

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

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

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

Remarks and examples

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

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

```
. xtprobit ..., pa ...
```

is equivalent to typing

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

See also [XT] **xtgee** for information about xtprobit.

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

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

for $i = 1, \dots, n$ panels, where $t = 1, \dots, n_i$, ν_i are i.i.d., $N(0, \sigma_\nu^2)$, 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 $\sigma_\epsilon^2 = 1$, independently of ν_i.

▷ Example 1: Random-effects model

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

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

. xtprobit union age grade i.not_smsa south##c.year

Fitting comparison model:

Iteration 0:  Log likelihood =  -13864.23
Iteration 1:  Log likelihood = -13545.541
Iteration 2:  Log likelihood = -13544.385
Iteration 3:  Log likelihood = -13544.385

Fitting full model:

rho =  0.0    Log likelihood = -13544.385
rho =  0.1    Log likelihood = -12237.655
rho =  0.2    Log likelihood = -11590.282
rho =  0.3    Log likelihood = -11211.185
rho =  0.4    Log likelihood = -10981.319
rho =  0.5    Log likelihood = -10852.793
rho =  0.6    Log likelihood = -10808.759
rho =  0.7    Log likelihood =  -10865.57

Iteration 0:  Log likelihood = -10807.712
Iteration 1:  Log likelihood = -10599.332
Iteration 2:  Log likelihood = -10552.287
Iteration 3:  Log likelihood = -10552.225
Iteration 4:  Log likelihood = -10552.225

Random-effects probit regression                     Number of obs     = 26,200
Group variable: idcode                               Number of groups =  4,434

Random effects u_i ~ Gaussian                        Obs per group:
                                                                  min =      1
                                                                  avg =    5.9
                                                                  max =     12

Integration method: mvaghermite                      Integration pts. =     12

                                                     Wald chi2(6)      = 220.91
Log likelihood = -10552.225                          Prob > chi2       = 0.0000

------------------------------------------------------------------------------
       union | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
         age |   .0082967   .0084599     0.98   0.327    -.0082843    .0248778
       grade |   .0482731   .0099469     4.85   0.000     .0287776    .0677686
  1.not_smsa |   -.139657   .0460548    -3.03   0.002    -.2299227   -.0493913
     1.south |  -1.584394    .358473    -4.42   0.000    -2.286989   -.8818002
        year |  -.0039854   .0088399    -0.45   0.652    -.0213113    .0133406
             |
south#c.year |
           1 |   .0134017   .0044622     3.00   0.003     .0046559    .0221475
             |
       _cons |  -1.668202   .4751819    -3.51   0.000    -2.599542   -.7368628
-------------+----------------------------------------------------------------
    /lnsig2u |   .6103616   .0458783                      .5204418    .7002814
-------------+----------------------------------------------------------------
     sigma_u |    1.35687   .0311255                      1.297217    1.419267
         rho |   .6480233   .0104643                      .6272511    .6682502
------------------------------------------------------------------------------
LR test of rho=0: chibar2(01) = 5984.32                 Prob >= chibar2 = 0.000
```

The output includes the additional panel-level variance component, which is parameterized as the log of the variance $\ln(\sigma_\nu^2)$ (labeled lnsig2u in the output). The standard deviation σ_ν is also included in the output (labeled sigma_u) together with ρ (labeled rho), where

$$\rho = \frac{\sigma_\nu^2}{\sigma_\nu^2 + 1}$$

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

When rho is zero, the panel-level variance component is unimportant, and the panel estimator is not different from the pooled estimator. A likelihood-ratio test of this is included at the bottom of the output. This test formally compares the pooled estimator (probit) with the panel estimator.

◁

❑ Technical note

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

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

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

Log             -10552.225     -10554.496     -10552.399
likelihood                     -2.2712569     -.17396615   Difference
                                .00021524      .00001649   Relative difference

union:           .00829671      .00828745      .00831488
     age                       -9.265e-06      .00001817   Difference
                               -.0011167       .00218987   Relative difference

union:            .0482731      .04860277      .04826287
   grade                        .00032967     -.00001023   Difference
                                .00682917     -.00021188   Relative difference

union:          -.13965702     -.14057441     -.13953521
1.not_smsa                     -.00091739      .00012181   Difference
                                .00656891     -.00087218   Relative difference

union:          -1.5843944     -1.5909857     -1.5843375
 1.south                       -.00659135      .00005689   Difference
                                .00416017     -.00003591   Relative difference

union:          -.00398535     -.00397811     -.00400181
    year                        7.237e-06     -.00001646   Difference
                               -.00181578      .00412982   Relative difference

union:           .01340169      .01344457      .01340388
1.south#c.~r                    .00004288      2.193e-06   Difference
                                .00319946      .0001636    Relative difference

union:          -1.6682022     -1.6757524     -1.6665327
   _cons                       -.00755024      .00166948   Difference
                                .00452597     -.00100077   Relative difference

/:               .61036163      .61780789      .60974814
 lnsig2u                        .00744626     -.00061349   Difference
                                .01219976     -.00100513   Relative difference
```

The results obtained for 12 quadrature points were closer to the results for 16 points than to the results for eight points. Although the relative and absolute differences are a bit larger than we would like, they are not large. We can increase the number of quadrature points with the intpoints() option; if we choose intpoints(20) and do another quadchk we will get acceptable results, with relative differences around 0.01%.

This is not the case if we use nonadaptive quadrature. Then the results we obtain are

```
. xtprobit union age grade i.not_smsa south##c.year, intmethod(ghermite)
Fitting comparison model:
Iteration 0:  Log likelihood =  -13864.23
Iteration 1:  Log likelihood = -13545.541
Iteration 2:  Log likelihood = -13544.385
Iteration 3:  Log likelihood = -13544.385
Fitting full model:
rho =  0.0    Log likelihood = -13544.385
rho =  0.1    Log likelihood = -12237.655
rho =  0.2    Log likelihood = -11590.282
rho =  0.3    Log likelihood = -11211.185
rho =  0.4    Log likelihood = -10981.319
rho =  0.5    Log likelihood = -10852.793
rho =  0.6    Log likelihood = -10808.759
rho =  0.7    Log likelihood =  -10865.57
Iteration 0:  Log likelihood = -10808.759
Iteration 1:  Log likelihood = -10594.349
Iteration 2:  Log likelihood = -10560.913
Iteration 3:  Log likelihood = -10560.876
Iteration 4:  Log likelihood = -10560.876
Random-effects probit regression                     Number of obs    = 26,200
Group variable: idcode                               Number of groups =  4,434
Random effects u_i ~ Gaussian                        Obs per group:
                                                                  min =      1
                                                                  avg =    5.9
                                                                  max =     12
Integration method: ghermite                         Integration pts. =     12
                                                     Wald chi2(6)     = 218.99
Log likelihood = -10560.876                          Prob > chi2      = 0.0000
```

union	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
age	.0093488	.0083385	1.12	0.262	-.0069945	.025692
grade	.0488014	.0101168	4.82	0.000	.0289728	.06863
1.not_smsa	-.1364862	.0462831	-2.95	0.003	-.2271995	-.045773
1.south	-1.592711	.3576715	-4.45	0.000	-2.293734	-.8916877
year	-.0053723	.0087219	-0.62	0.538	-.0224668	.0117223
south#c.year						
1	.0136764	.0044532	3.07	0.002	.0049482	.0224046
_cons	-1.575539	.4639881	-3.40	0.001	-2.484939	-.6661388
/lnsig2u	.5615976	.0432021			.476923	.6462722
sigma_u	1.324187	.0286038			1.269295	1.381453
rho	.6368221	.0099918			.617021	.6561699

```
LR test of rho=0: chibar2(01) = 5967.02                Prob >= chibar2 = 0.000
```

We now check the stability of the quadrature technique for this nonadaptive quadrature model. We expect it to be less stable.

```
. quadchk, nooutput
Refitting model intpoints() =  8
Refitting model intpoints() = 16
                                 Quadrature check
                          Fitted      Comparison     Comparison
                        quadrature    quadrature     quadrature
                        12 points      8 points      16 points

Log                     -10560.876    -10574.239     -10555.792
likelihood                            -13.362535      5.0839579   Difference
                                       .00126529     -.0004814    Relative difference

union:                   .00934876     .01264615      .00731888
     age                               .0032974      -.00202987   Difference
                                       .35270966     -.21712744   Relative difference

union:                   .04880139     .05710089      .04432417
   grade                               .00829951     -.00447722   Difference
                                       .17006703     -.09174372   Relative difference

union:                  -.13648624    -.13327724     -.14094541
1.not_smsa                              .003209      -.00445917   Difference
                                      -.0235115       .03267123   Relative difference

union:                  -1.592711     -1.5275627     -1.6059143
 1.south                               .06514823     -.01320331   Difference
                                      -.04090399      .00828983   Relative difference

union:                  -.00537226    -.00867673     -.00307042
    year                              -.00330447      .00230184   Difference
                                       .61509968     -.4284678    Relative difference

union:                   .01367641     .01278071      .01369009
1.south#c.~r                          -.0008957       .00001368   Difference
                                      -.06549266      .00100054   Relative difference

union:                  -1.5755388    -1.4888646     -1.6505526
   _cons                               .08667418     -.0750138    Difference
                                      -.0550124       .04761152   Relative difference

/:                       .56159763     .49290978      .58068904
 lnsig2u                              -.06868786      .0190914    Difference
                                      -.12230795      .03399481   Relative difference
```

Once again, the results obtained for 12 quadrature points were closer to the results for 16 points than to the results for eight points. However, here the convergence point seems to be sensitive to the number of quadrature points, so we should not trust these results. We should increase the number of quadrature points with the `intpoints()` option and then use `quadchk` again. We should not use the results of a random-effects specification when there is evidence that the numeric technique for calculating the model is not stable (as shown by `quadchk`).

Generally, the relative differences in the coefficients should not change by more than 1% if the quadrature technique is stable. See [XT] **quadchk** for details. Increasing the number of quadrature points can often improve the stability, and for models with high `rho` we may need many. We can also switch between adaptive and nonadaptive quadrature. As a rule, adaptive quadrature, which is the default integration method, is much more flexible and robust.

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

❑

▷ Example 2: Equal-correlation model

As an alternative to the random-effects specification, we can fit an equal-correlation probit model:

```
. xtprobit union age grade i.not_smsa south##c.year, pa
Iteration 1:  Tolerance = .12544249
Iteration 2:  Tolerance = .0034686
Iteration 3:  Tolerance = .00017448
Iteration 4:  Tolerance = 8.382e-06
Iteration 5:  Tolerance = 3.997e-07
GEE population-averaged model                    Number of obs     = 26,200
Group variable: idcode                           Number of groups  =  4,434
Family: Binomial                                 Obs per group:
Link:   Probit                                                min =      1
Correlation: exchangeable                                     avg =    5.9
                                                              max =     12
                                                 Wald chi2(6)      = 242.57
Scale parameter = 1                              Prob > chi2       = 0.0000

------------------------------------------------------------------------------
       union | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
         age |   .0089699   .0053208     1.69   0.092    -.0014586    .0193985
       grade |   .0333174   .0062352     5.34   0.000     .0210966    .0455382
  1.not_smsa |  -.0715717    .027543    -2.60   0.009    -.1255551   -.0175884
     1.south |  -1.017368    .207931    -4.89   0.000    -1.424905   -.6098308
        year |  -.0062708   .0055314    -1.13   0.257    -.0171122    .0045706
             |
south#c.year |
           1 |   .0086294     .00258     3.34   0.001     .0035727     .013686
             |
       _cons |  -.8670997    .294771    -2.94   0.003     -1.44484   -.2893592
------------------------------------------------------------------------------
```

◁

▷ Example 3: Population-averaged model

In example 3 of [R] **probit**, we showed the above results and compared them with `probit, vce(cluster id)`. `xtprobit` with the `pa` option allows a `vce(robust)` option, so we can obtain the population-averaged probit estimator with the robust variance calculation by typing

```
. xtprobit union age grade i.not_smsa south##c.year, pa vce(robust) nolog
GEE population-averaged model                     Number of obs     = 26,200
Group variable: idcode                            Number of groups =   4,434
Family: Binomial                                  Obs per group:
Link:   Probit                                                  min =      1
Correlation: exchangeable                                       avg =    5.9
                                                                max =     12
                                                  Wald chi2(6)      = 156.33
Scale parameter = 1                               Prob > chi2       = 0.0000
                                  (Std. err. adjusted for clustering on idcode)
```

union	Coefficient	Semirobust std. err.	z	P>|z|	[95% conf. interval]	
age	.0089699	.0051169	1.75	0.080	-.001059	.0189988
grade	.0333174	.0076425	4.36	0.000	.0183383	.0482965
1.not_smsa	-.0715717	.0348659	-2.05	0.040	-.1399076	-.0032359
1.south	-1.017368	.3026981	-3.36	0.001	-1.610645	-.4240906
year	-.0062708	.0055745	-1.12	0.261	-.0171965	.0046549
south#c.year						
1	.0086294	.0037866	2.28	0.023	.0012078	.0160509
_cons	-.8670997	.3243959	-2.67	0.008	-1.502904	-.2312955

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

◁

▷ Example 4: Random-effects model with stable quadrature

In a previous example, we showed how quadchk indicated that the quadrature technique was numerically unstable. Here we present an example in which the quadrature is stable.

In this example, we have (synthetic) data on whether workers complain to managers at fast-food restaurants. The covariates are age (in years of the worker), grade (years of schooling completed by the worker), south (equal to 1 if the restaurant is located in the South), tenure (the number of years spent on the job by the worker), gender (of the worker), race (of the worker), income (in thousands of dollars by the restaurant), genderm (gender of the manager), burger (equal to 1 if the restaurant specializes in hamburgers), and chicken (equal to 1 if the restaurant specializes in chicken). The model is given by

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

. xtprobit complain age grade south tenure gender race income genderm burger
> chicken, nolog

Random-effects probit regression                     Number of obs    =  2,763
Group variable: restaurant                           Number of groups =    500

Random effects u_i ~ Gaussian                        Obs per group:
                                                                  min =      3
                                                                  avg =    5.5
                                                                  max =      8

Integration method: mvaghermite                      Integration pts. =     12

                                                     Wald chi2(10)    = 126.59
Log likelihood = -1318.2088                          Prob > chi2      = 0.0000

------------------------------------------------------------------------------
    complain | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
         age |  -.0430409   .0130211    -3.31   0.001    -.0685617    -.01752
       grade |   .0330934   .0264572     1.25   0.211    -.0187618    .0849486
       south |      .1012   .0707196     1.43   0.152     -.037408    .2398079
      tenure |  -.0440079   .0987099    -0.45   0.656    -.2374758      .14946
      gender |   .3318499   .0601382     5.52   0.000     .2139812    .4497185
        race |   .3417901   .0382251     8.94   0.000     .2668703    .4167098
      income |  -.0022702   .0008885    -2.56   0.011    -.0040117   -.0005288
     genderm |   .0524577   .0706585     0.74   0.458    -.0860305    .1909459
      burger |   .0448931   .0956151     0.47   0.639    -.1425091    .2322953
     chicken |   .1904714   .0953067     2.00   0.046     .0036737    .3772691
       _cons |  -.2145311   .6240549    -0.34   0.731    -1.437656    1.008594
-------------+----------------------------------------------------------------
    /lnsig2u |  -1.704494   .2502057                     -2.194888   -1.214099
-------------+----------------------------------------------------------------
     sigma_u |   .4264557   .0533508                       .333723    .5449563
         rho |   .1538793   .0325769                      .1002105    .2289765
------------------------------------------------------------------------------
LR test of rho=0: chibar2(01) = 29.91                  Prob >= chibar2 = 0.000
```

Again we would like to check the stability of the quadrature technique of the model before interpreting the results. Given the estimate of ρ and the small size of the panels (between 3 and 8), we should find that the quadrature technique is numerically stable.

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

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

Log             -1318.2088    -1318.2088    -1318.2088
likelihood                    -2.002e-06    -1.194e-09   Difference
                               1.519e-09     9.061e-13   Relative difference

complain:       -.04304086    -.04304086    -.04304086
     age                      -3.896e-10    -2.625e-12   Difference
                               9.051e-09     6.100e-11   Relative difference

complain:         .0330934      .0330934      .0330934
   grade                       2.208e-11     1.867e-12   Difference
                               6.673e-10     5.642e-11   Relative difference

complain:        .10119998     .10119999     .10119998
   south                       2.369e-09     3.957e-11   Difference
                               2.341e-08     3.910e-10   Relative difference

complain:       -.04400789     -.0440079    -.04400789
  tenure                      -3.362e-09    -2.250e-11   Difference
                               7.640e-08     5.114e-10   Relative difference

complain:        .33184986     .33184986     .33184986
  gender                       3.190e-09     2.546e-11   Difference
                               9.612e-09     7.673e-11   Relative difference

complain:        .34179006     .34179007     .34179006
    race                       3.801e-09     2.990e-11   Difference
                               1.112e-08     8.749e-11   Relative difference

complain:       -.00227021    -.00227021    -.00227021
  income                      -4.468e-11    -9.252e-13   Difference
                               1.968e-08     4.075e-10   Relative difference

complain:        .05245769     .05245769     .05245769
 genderm                       1.963e-09     4.481e-11   Difference
                               3.742e-08     8.542e-10   Relative difference

complain:        .04489311     .04489311     .04489311
  burger                       4.173e-10     6.628e-12   Difference
                               9.296e-09     1.476e-10   Relative difference

complain:        .19047138     .19047139     .19047138
 chicken                       3.096e-09     4.916e-11   Difference
                               1.625e-08     2.581e-10   Relative difference

complain:       -.21453112    -.21453111    -.21453112
   _cons                       1.281e-08     2.682e-10   Difference
                              -5.972e-08    -1.250e-09   Relative difference

/:              -1.7044935    -1.7044934    -1.7044935
 lnsig2u                       1.255e-07    -4.135e-10   Difference
                              -7.365e-08     2.426e-10   Relative difference
```

The relative and absolute differences are all small between the default 12 quadrature points and the result with 16 points. We do not have any coefficients that have a large difference between the default 12 quadrature points and eight quadrature points.

We conclude that the quadrature technique is stable. Because the differences here are so small, we would plan on using and interpreting these results rather than trying to rerun with more quadrature points.

Stored results

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

Scalars

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

Macros

`e(cmd)`	`xtprobit`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(ivar)`	variable denoting groups
`e(model)`	`re`
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(title)`	title in estimation output
`e(clustvar)`	name of cluster variable
`e(offset)`	linear offset variable
`e(chi2type)`	`Wald` or `LR`; type of model χ^2 test
`e(chi2_ct)`	`Wald` or `LR`; type of model χ^2 test corresponding to `e(chi2_c)`
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. err.
`e(intmethod)`	integration method
`e(distrib)`	`Gaussian`; the distribution of the random effect

e(opt)	type of optimization
e(which)	max or min; whether optimizer is to perform maximization or minimization
e(ml_method)	type of ml method
e(user)	name of likelihood-evaluator program
e(technique)	maximization technique
e(properties)	b V
e(predict)	program used to implement predict
e(marginsdefault)	default predict() specification for margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved

Matrices

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

Functions

e(sample)	marks estimation sample

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

Matrices

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

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

xtprobit, pa stores the following in e():

Scalars

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

Macros

e(cmd)	xtgee
e(cmd2)	xtprobit
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(ivar)	variable denoting groups
e(tvar)	variable denoting time within groups
e(model)	pa

e(family)	binomial
e(link)	probit; link function
e(corr)	correlation structure
e(scale)	x2, dev, phi, or #; scale parameter
e(wtype)	weight type
e(wexp)	weight expression
e(offset)	linear offset variable
e(chi2type)	Wald; type of model χ^2 test
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(nmp)	nmp, if specified
e(properties)	b V
e(predict)	program used to implement predict
e(marginsnotok)	predictions disallowed by margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved
Matrices	
e(b)	coefficient vector
e(R)	estimated working correlation matrix
e(V)	variance–covariance matrix of the estimators
e(V_modelbased)	model-based variance
Functions	
e(sample)	marks estimation sample

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

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

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

Methods and formulas

xtprobit reports the population-averaged results obtained by using xtgee, family(binomial) link(probit) to obtain estimates.

Assuming a normal distribution, $N(0, \sigma_\nu^2)$, for the random effects ν_i

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

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

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

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

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

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

Letting

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

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

and

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

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

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

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

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

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

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

xtprobit, re and the robust VCE estimator

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

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

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

References

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

Baltagi, B. H. 2009. *A Companion to Econometric Analysis of Panel Data*. Chichester, UK: Wiley.

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

Conway, M. R. 1990. A random effects model for binary data. *Biometrics* 46: 317–328. https://doi.org/10.2307/2531437.

Cruz-Gonzalez, M., I. Fernández-Val, and M. Weidner. 2017. Bias corrections for probit and logit models with two-way fixed effects. *Stata Journal* 17: 517–545.

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

Guilkey, D. K., and J. L. Murphy. 1993. Estimation and testing in the random effects probit model. *Journal of Econometrics* 59: 301–317. https://doi.org/10.1016/0304-4076(93)90028-4.

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

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

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

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

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

Plum, A. 2016. bireprob: An estimator for bivariate random-effects probit models. *Stata Journal* 16: 96–111.

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

Stewart, M. B. 2006. Maximum simulated likelihood estimation of random-effects dynamic probit models with autocorrelated errors. *Stata Journal* 6: 256–272.

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

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

Also see

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

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

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

[XT] **xteprobit** — Extended random-effects probit regression

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

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

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

[BAYES] **bayes: xtprobit** — Bayesian random-effects probit model

[ME] **meprobit** — Multilevel mixed-effects probit regression

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

[R] **probit** — Probit regression

[U] 20 Estimation and postestimation commands