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 \( x_1, x_2, \) and indicators for levels of categorical variable \( a \) using _xtset_ data

\[
_xtprobit\ y\ x_1\ x_2\ i.a
\]

Population-averaged model with robust standard errors

\[
_xtprobit\ y\ x_1\ x_2\ i.a,\ pa\ vce(robust)
\]

As above, but specify an autoregressive correlation structure of order 1

\[
_xtprobit\ y\ x_1\ x_2\ i.a,\ pa\ vce(robust)\ corr(ar\ 1)
\]

Random-effects model with cluster–robust standard errors for panels nested within \( cvar \)

\[
_xtprobit\ y\ x_1\ x_2\ i.a,\ vce(cluster\ cvar)
\]

Menu

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

### Random-effects (RE) model

```bash
xtprobit depvar [ indepvars ] [ if ] [ in ] [ weight ] [ , re RE_options ]
```

### Population-averaged (PA) model

```bash
xtprobit depvar [ indepvars ] [ if ] [ in ] [ weight ] , pa [ PA_options ]
```

### RE_options

<table>
<thead>
<tr>
<th>Description</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>noconstant suppress constant term</td>
</tr>
<tr>
<td></td>
<td>offset(varname) include varname in model with coefficient constrained to 1</td>
</tr>
<tr>
<td></td>
<td>asis retain perfect predictor variables</td>
</tr>
<tr>
<td></td>
<td>level(#) set confidence level; default is level(95)</td>
</tr>
<tr>
<td></td>
<td>noconsreport do not display constraints</td>
</tr>
<tr>
<td></td>
<td>intmethod(intmethod) integration method; intmethod may be mvaghermite (the default) or ghermite</td>
</tr>
<tr>
<td></td>
<td>maximize_options control the maximization process; seldom used</td>
</tr>
<tr>
<td></td>
<td>coeflegend display legend instead of statistics</td>
</tr>
</tbody>
</table>
### Model options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>noconstant</td>
<td>suppress constant term</td>
</tr>
<tr>
<td>pa</td>
<td>use population-averaged estimator</td>
</tr>
<tr>
<td>offset(varname)</td>
<td>include varname in model with coefficient constrained to 1</td>
</tr>
<tr>
<td>asis</td>
<td>retain perfect predictor variables</td>
</tr>
</tbody>
</table>

### Correlation

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>corr(correlation)</td>
<td>within-panel correlation structure</td>
</tr>
<tr>
<td>force</td>
<td>estimate even if observations unequally spaced in time</td>
</tr>
</tbody>
</table>

### SE/Robust

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vce(vcetype)</td>
<td>vcetype may be conventional, robust, bootstrap, or jackknife</td>
</tr>
<tr>
<td>nmp</td>
<td>use divisor ( N - P ) instead of the default ( N )</td>
</tr>
<tr>
<td>scale(parm)</td>
<td>overrides the default scale parameter; ( parm ) may be ( x2 ), dev, phi, or #</td>
</tr>
</tbody>
</table>

### Reporting

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>level(#)</td>
<td>set confidence level; default is level(95)</td>
</tr>
<tr>
<td>display_options</td>
<td>control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling</td>
</tr>
</tbody>
</table>

### Optimization

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>optimize_options</td>
<td>control the optimization process; seldom used</td>
</tr>
<tr>
<td>coeflegend</td>
<td>display legend instead of statistics</td>
</tr>
</tbody>
</table>

### Correlation

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>exchangeable</td>
<td>exchangeable</td>
</tr>
<tr>
<td>independent</td>
<td>independent</td>
</tr>
<tr>
<td>unstructured</td>
<td>unstructured</td>
</tr>
<tr>
<td>fixed matname</td>
<td>user-specified</td>
</tr>
<tr>
<td>ar #</td>
<td>autoregressive of order #</td>
</tr>
<tr>
<td>stationary #</td>
<td>stationary of order #</td>
</tr>
<tr>
<td>nonstationary #</td>
<td>nonstationary of order #</td>
</tr>
</tbody>
</table>

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

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

depvar and indevars 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. 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.

iwights, 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

- `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, nolabel, 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

- `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.

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 xtprobbit 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

\[
   \Pr(y_{it} \neq 0| x_{it}) = \Phi(x_{it}\beta + \nu_i)
\]

for \(i = 1, \ldots, n\) panels, where \(t = 1, \ldots, n_i\), \(\nu_i\) are i.i.d., \(N(0, \sigma^2_\nu)\), and \(\Phi\) is the standard normal cumulative distribution function.

Underlying this model is the variance components model

\[
y_{it} \neq 0 \iff x_{it}\beta + \nu_i + \epsilon_{it} > 0
\]

where \(\epsilon_{it}\) are i.i.d. Gaussian distributed with mean zero and variance \(\sigma^2_\epsilon = 1\), independently of \(\nu_i\).

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/r16/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

|              | Coef.     | Std. Err. |      z    | P>|z|    | [95% Conf. Interval] |
|--------------|-----------|-----------|-----------|--------|---------------------|
| union        |           |           |           |        |                     |
| age          | .00858767 | .0084799 |  0.98     |  0.327 | -0.0082843          |  0.0248778 |
| grade        | .0482731  | .0092469  |  4.35     |  0.000 |  0.0287776          |  0.0677686 |
| 1.not_smsa   | -.139657  | .046548   | -3.03     |  0.002 | -.2299227           | -.0493931 |
| 1.south      | -.1584394 | .358473   | -4.42     |  0.000 | -2.286989           | -.8818002 |
| year         | -.0039854 | .0083999  | -0.45     |  0.652 | -.0213113           |  0.0135466 |
| south#c.year | 1         | 0.0134017 |  0.30     |  0.003 |  0.0046559          |  0.0221475 |
|              | _cons     | 1.668202  |  4.75     |  0.000 | -2.599542           | -.7368268 |
|              | /lnsig2u  | .6103616  | .0458783  |  5.204418 | .7002814        |
|              | sigma_u   | 1.35687   | .0311255  |  1.297217 |  1.419267        |
|              | rho       | 0.480323  | .0458783  |  2.672511 |  1.6682502        |
LR test of rho=0:  chibar2(01) =  5984.32  Prob >= chibar2 =   0.0000

The output includes the additional panel-level variance component, which is parameterized as the log of the variance ln(σν²) (labeled lnsig2u in the output). The standard deviation σν is also included in the output (labeled sigma_u) together with ρ (labeled rho), where

ρ = \frac{σν²}{σν² + 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

<table>
<thead>
<tr>
<th></th>
<th>Fitted quadrature</th>
<th>Comparison quadrature</th>
<th>Comparison quadrature</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12 points</td>
<td>8 points</td>
<td>16 points</td>
</tr>
<tr>
<td>Log likelihood</td>
<td>-10552.225</td>
<td>-10554.496</td>
<td>-10552.399</td>
</tr>
<tr>
<td>union:</td>
<td>.00829671</td>
<td>.00828745</td>
<td>.00831488</td>
</tr>
<tr>
<td>age</td>
<td>-9.265e-06</td>
<td>.00011167</td>
<td>.00218987</td>
</tr>
<tr>
<td>union:</td>
<td>.0482731</td>
<td>.04860277</td>
<td>.04826287</td>
</tr>
<tr>
<td>grade</td>
<td>.00329677</td>
<td>-.00001023</td>
<td>.04826917</td>
</tr>
<tr>
<td>union:</td>
<td>-.13965702</td>
<td>-.14057441</td>
<td>-.13953521</td>
</tr>
<tr>
<td>1.not_smsa</td>
<td>-.00091739</td>
<td>.00012181</td>
<td>-.00087218</td>
</tr>
<tr>
<td>union:</td>
<td>-.15843944</td>
<td>-.15909857</td>
<td>-.15843735</td>
</tr>
<tr>
<td>1.south</td>
<td>-.00659135</td>
<td>.00005689</td>
<td>.0416017</td>
</tr>
<tr>
<td>union:</td>
<td>-.00398535</td>
<td>-.00397811</td>
<td>-.00400181</td>
</tr>
<tr>
<td>year</td>
<td>7.237e-06</td>
<td>-.00001646</td>
<td>.00412982</td>
</tr>
<tr>
<td>union:</td>
<td>.01340169</td>
<td>.01344457</td>
<td>.01340388</td>
</tr>
<tr>
<td>1.south#c.r</td>
<td>.00004288</td>
<td>2.193e-06</td>
<td>.00319946</td>
</tr>
<tr>
<td>union:</td>
<td>-.16682022</td>
<td>-.16757524</td>
<td>-.16665327</td>
</tr>
<tr>
<td>_cons</td>
<td>-.00755024</td>
<td>.00166948</td>
<td>-.00100077</td>
</tr>
<tr>
<td>/: lnsig2u</td>
<td>.61036163</td>
<td>.61780789</td>
<td>.60974814</td>
</tr>
<tr>
<td></td>
<td>.00744626</td>
<td>-.00061349</td>
<td>.01219976</td>
</tr>
</tbody>
</table>
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 1.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
```

We now check the stability of the quadrature technique for this nonadaptive quadrature model. We expect it to be less stable.
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 \texttt{xtprobit}, \texttt{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.

\section*{Example 2: Equal-correlation model}

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

\begin{verbatim}
. xtprobit union age grade i.not_smsa south#c.year, pa

display

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
Link: probit
Family: binomial
Correlation: exchangeable

Wald chi2(6) = 242.57
Scale parameter: 1
Prob > chi2 = 0.0000

| Coef.  | Std. Err. | z     | P>|z| | 95% Conf. Interval |
|--------|-----------|-------|------|---------------------|
| union  | .0089699  | .0053208 | 1.69 | .092 | -.0014586, .0193985 |
| age    | .033174   | .0062352 | 5.34 | .000 | .0210966, .0455382 |
| grade  | -.0715717 | .027543 | -2.60 | .009 | -.1255551, -.0175884 |
| 1.not_smsa | -.1.017368  | .207931 | -4.89 | .000 | -.1.424905, -.6098308 |
| 1.south | -.0062708 | .0055314 | -1.13 | .257 | -.0171122, .0045706 |
| year   | .0086294  | .00258  | 3.34 | .001 | .0035727, .013686 |
| south#c.year | .8670997  | .294771 | -2.94 | .003 | -.1.44484, -.2893592 |

\end{verbatim}

\section*{Example 3: Population-averaged model}

In example 3 of \texttt{[R] probit}, we showed the above results and compared them with \texttt{probit, vce(cluster id)}. \texttt{xtprobit} with the \texttt{pa} option allows a \texttt{vce(robust)} option, so we can obtain the population-averaged probit estimator with the robust variance calculation by typing
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/r16/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  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     .1012102   .0707196   1.43  0.152  -.037408   .2398079
tenure    -.0440079   .0987099  -0.45  0.656  -.2374758   .1498617
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  -.1437656   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 \( \rho \) and the small size of the panels (between 3 and 8), we should find that the quadrature technique is numerically stable.
Refitting model intpoints() = 8
Refitting model intpoints() = 16

<table>
<thead>
<tr>
<th>Quadrature check</th>
<th>Fitted quadrature</th>
<th>Comparison quadrature</th>
<th>Comparison quadrature</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12 points</td>
<td>8 points</td>
<td>16 points</td>
</tr>
<tr>
<td>Log likelihood</td>
<td>-1318.2088</td>
<td>-1318.2088</td>
<td>-1318.2088</td>
</tr>
<tr>
<td></td>
<td>-2.002e-06</td>
<td>-1.194e-09</td>
<td>1.519e-09</td>
</tr>
<tr>
<td></td>
<td>1.519e-09</td>
<td>9.061e-13</td>
<td>9.061e-13</td>
</tr>
<tr>
<td>complain: age</td>
<td>-.04304086</td>
<td>-.04304086</td>
<td>-.04304086</td>
</tr>
<tr>
<td></td>
<td>-3.896e-10</td>
<td>-2.625e-12</td>
<td>9.051e-09</td>
</tr>
<tr>
<td></td>
<td>9.051e-09</td>
<td>6.100e-11</td>
<td>6.100e-11</td>
</tr>
<tr>
<td>complain: grade</td>
<td>.0330934</td>
<td>.0330934</td>
<td>.0330934</td>
</tr>
<tr>
<td></td>
<td>2.208e-11</td>
<td>1.867e-12</td>
<td>6.673e-10</td>
</tr>
<tr>
<td></td>
<td>6.673e-10</td>
<td>5.643e-11</td>
<td>5.643e-11</td>
</tr>
<tr>
<td>complain: south</td>
<td>.101199999</td>
<td>.101199999</td>
<td>.101199999</td>
</tr>
<tr>
<td></td>
<td>2.369e-09</td>
<td>3.957e-11</td>
<td>2.341e-08</td>
</tr>
<tr>
<td></td>
<td>2.341e-08</td>
<td>3.910e-10</td>
<td>3.910e-10</td>
</tr>
<tr>
<td>complain: tenure</td>
<td>-.04400789</td>
<td>-.04400789</td>
<td>-.04400789</td>
</tr>
<tr>
<td></td>
<td>-3.362e-09</td>
<td>-2.250e-11</td>
<td>7.640e-08</td>
</tr>
<tr>
<td></td>
<td>7.640e-08</td>
<td>5.114e-10</td>
<td>5.114e-10</td>
</tr>
<tr>
<td>complain: gender</td>
<td>.33184986</td>
<td>.33184986</td>
<td>.33184986</td>
</tr>
<tr>
<td></td>
<td>3.190e-09</td>
<td>2.546e-11</td>
<td>9.612e-09</td>
</tr>
<tr>
<td></td>
<td>9.612e-09</td>
<td>7.673e-11</td>
<td>7.673e-11</td>
</tr>
<tr>
<td>complain: race</td>
<td>.34179006</td>
<td>.34179007</td>
<td>.34179006</td>
</tr>
<tr>
<td></td>
<td>3.801e-09</td>
<td>2.990e-11</td>
<td>1.112e-08</td>
</tr>
<tr>
<td></td>
<td>1.112e-08</td>
<td>8.749e-11</td>
<td>8.749e-11</td>
</tr>
<tr>
<td>complain: income</td>
<td>-.00227021</td>
<td>-.00227021</td>
<td>-.00227021</td>
</tr>
<tr>
<td></td>
<td>-4.468e-11</td>
<td>-9.252e-13</td>
<td>1.968e-08</td>
</tr>
<tr>
<td></td>
<td>1.968e-08</td>
<td>4.075e-10</td>
<td>4.075e-10</td>
</tr>
<tr>
<td>complain: genderm</td>
<td>.05245769</td>
<td>.05245769</td>
<td>.05245769</td>
</tr>
<tr>
<td></td>
<td>1.963e-09</td>
<td>4.481e-11</td>
<td>3.742e-08</td>
</tr>
<tr>
<td></td>
<td>3.742e-08</td>
<td>8.542e-10</td>
<td>8.542e-10</td>
</tr>
<tr>
<td>complain: burger</td>
<td>.04489311</td>
<td>.04489311</td>
<td>.04489311</td>
</tr>
<tr>
<td></td>
<td>4.173e-10</td>
<td>6.628e-12</td>
<td>9.296e-09</td>
</tr>
<tr>
<td></td>
<td>9.296e-09</td>
<td>1.476e-10</td>
<td>1.476e-10</td>
</tr>
<tr>
<td>complain: chicken</td>
<td>.19047138</td>
<td>.19047139</td>
<td>.19047138</td>
</tr>
<tr>
<td></td>
<td>3.096e-09</td>
<td>4.916e-11</td>
<td>1.625e-08</td>
</tr>
<tr>
<td></td>
<td>1.625e-08</td>
<td>2.581e-10</td>
<td>2.581e-10</td>
</tr>
<tr>
<td>complain: _cons</td>
<td>-.21453112</td>
<td>-.21453111</td>
<td>-.21453112</td>
</tr>
<tr>
<td></td>
<td>1.281e-08</td>
<td>2.682e-10</td>
<td>5.972e-08</td>
</tr>
<tr>
<td></td>
<td>5.972e-08</td>
<td>1.250e-09</td>
<td>1.250e-09</td>
</tr>
<tr>
<td>/: lnsig2u</td>
<td>-1.7044935</td>
<td>-1.7044934</td>
<td>-1.7044935</td>
</tr>
<tr>
<td></td>
<td>1.255e-07</td>
<td>-4.135e-10</td>
<td>-7.365e-08</td>
</tr>
<tr>
<td></td>
<td>1.255e-07</td>
<td>-4.135e-10</td>
<td>-7.365e-08</td>
</tr>
</tbody>
</table>
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) : \( \chi^2 \)
-   e(chi2_c) : \( \chi^2 \) for comparison test
-   e(N_clust) : number of clusters
-   e(rho) : \( \rho \)
-   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 \( \chi^2 \) test
-   e(chi2_ct) : Wald or LR; type of model \( \chi^2 \) 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

<table>
<thead>
<tr>
<th>Matrix</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>e(b)</em></td>
<td>coefficient vector</td>
</tr>
<tr>
<td><em>e(Cns)</em></td>
<td>constraints matrix</td>
</tr>
<tr>
<td><em>e(ilog)</em></td>
<td>iteration log</td>
</tr>
<tr>
<td><em>e(gradient)</em></td>
<td>gradient vector</td>
</tr>
<tr>
<td><em>e(V)</em></td>
<td>variance–covariance matrix of the estimators</td>
</tr>
<tr>
<td><em>e(V_modelbased)</em></td>
<td>model-based variance</td>
</tr>
</tbody>
</table>

#### Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>e(sample)</em></td>
<td>marks estimation sample</td>
</tr>
</tbody>
</table>

### xtprobit, pa stores the following in _e_():

#### Scalars

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

#### Macros

<table>
<thead>
<tr>
<th>Macro</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>e(cmd)</em></td>
<td><em>xtgee</em></td>
</tr>
<tr>
<td><em>e(cmd2)</em></td>
<td><em>xtprobit</em></td>
</tr>
<tr>
<td><em>e(cmdline)</em></td>
<td>command as typed</td>
</tr>
<tr>
<td><em>e(depvar)</em></td>
<td>name of dependent variable</td>
</tr>
<tr>
<td><em>e(ivar)</em></td>
<td>variable denoting groups</td>
</tr>
<tr>
<td><em>e(tvar)</em></td>
<td>variable denoting time within groups</td>
</tr>
<tr>
<td><em>e(model)</em></td>
<td><em>pa</em></td>
</tr>
<tr>
<td><em>e(family)</em></td>
<td><em>binomial</em></td>
</tr>
<tr>
<td><em>e(link)</em></td>
<td><em>probit</em>; link function</td>
</tr>
<tr>
<td><em>e(corr)</em></td>
<td>correlation structure</td>
</tr>
<tr>
<td><em>e(scale)</em></td>
<td>_x2, dev, phi, or <em>#</em>; scale parameter</td>
</tr>
<tr>
<td><em>e(vtype)</em></td>
<td>weight type</td>
</tr>
<tr>
<td><em>e(wexp)</em></td>
<td>weight expression</td>
</tr>
<tr>
<td><em>e(offset)</em></td>
<td>linear offset variable</td>
</tr>
<tr>
<td><em>e(chi2type)</em></td>
<td><em>Wald</em>; type of model <em>χ^2</em> test</td>
</tr>
<tr>
<td><em>e(vce)</em></td>
<td><em>vcetype</em> specified in <em>vce</em>()</td>
</tr>
<tr>
<td><em>e(vcetype)</em></td>
<td>title used to label Std. Err.</td>
</tr>
<tr>
<td><em>e(nmp)</em></td>
<td><em>nmp</em>, if specified</td>
</tr>
<tr>
<td><em>e(properties)</em></td>
<td><em>b</em> <em>V</em></td>
</tr>
<tr>
<td><em>e(predict)</em></td>
<td>program used to implement <em>predict</em></td>
</tr>
</tbody>
</table>
xtprobit — Random-effects and population-averaged probit models

Predictions disallowed by margins
factor variables fvset as asbalanced
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

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

Assuming a normal distribution, \( N(0, \sigma^2) \), for the random effects \( \nu_i \)

\[
\Pr(y_{i1}, \ldots, y_{in_i}|x_{i1}, \ldots, x_{in_i}) = \int_{-\infty}^{\infty} \frac{e^{-\nu^2/2\sigma^2}}{\sqrt{2\pi}\sigma} \left\{ \prod_{t=1}^{n_i} F(y_{it}, x_{it}\beta + \nu_i) \right\} d\nu_i
\]

where

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

where \( \Phi \) is the cumulative normal distribution.

The panel-level likelihood \( l_i \) is given by

\[
l_i = \int_{-\infty}^{\infty} \frac{e^{-\nu^2/2\sigma^2}}{\sqrt{2\pi}\sigma} \left\{ \prod_{t=1}^{n_i} F(y_{it}, x_{it}\beta + \nu_i) \right\} d\nu_i = \int_{-\infty}^{\infty} g(y_{it}, x_{it}, \nu_i) d\nu_i
\]

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

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

where \( \tilde{\sigma}_i \) and \( \tilde{\mu}_i \) 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
\[ L \approx \sum_{i=1}^{n} w_i \log \left[ \sqrt{2\hat{\sigma}_i} \sum_{m=1}^{M} w^*_m \exp \left\{ (a^*_m)^2 \right\} \frac{\exp \left\{ -(\sqrt{2\hat{\sigma}_i}a^*_m + \hat{\mu}_i)^2 / 2\sigma^2_\nu \right\}}{\sqrt{2\pi}\sigma_\nu} \right] \]

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 \( \hat{\mu}_i \) and \( \hat{\sigma}_i \) by following the method of Naylor and Smith (1982), further discussed in Skrondal and Rabe-Hesketh (2004). We start with \( \hat{\sigma}_i,0 = 1 \) and \( \hat{\mu}_i,0 = 0 \), and the posterior means and variances are updated in the \( k \)th iteration. That is, at the \( k \)th iteration of the optimization for \( l_i \), we use

\[
l_{i,k} \approx \sum_{m=1}^{M} \sqrt{2\hat{\sigma}_{i,k-1}} w^*_m \exp \left\{ (a^*_m)^2 \right\} g(y_{it}, x_{it}, \sqrt{2\hat{\sigma}_{i,k-1}} a^*_m + \hat{\mu}_{i,k-1})
\]

Letting

\[
\tau_{i,m,k-1} = \sqrt{2\hat{\sigma}_{i,k-1}} a^*_m + \hat{\mu}_{i,k-1}
\]

and

\[
\hat{\mu}_{i,k} = \sum_{m=1}^{M} \left( \tau_{i,m,k-1} \right) \frac{\sqrt{2\hat{\sigma}_{i,k-1}} w^*_m \exp \left\{ (a^*_m)^2 \right\} g(y_{it}, x_{it}, \tau_{i,m,k-1})}{l_{i,k}}
\]

and this is repeated until \( \hat{\mu}_{i,k} \) and \( \hat{\sigma}_{i,k} \) 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 \( 1 \times 10^{-6} \); after this, the quadrature parameters are fixed.

The log likelihood can also be calculated by nonadaptive Gauss–Hermite quadrature, the int-method(ghermite) option, where \( \rho = \sigma^2_\nu / (\sigma^2_\nu + 1) \):

\[
L = \sum_{i=1}^{n} w_i \log \left\{ \Pr(y_{i1}, \ldots, y_{in_i} | x_{i1}, \ldots, x_{in_i}) \right\}
\]

\[
\approx \sum_{i=1}^{n} w_i \log \left[ \frac{1}{\sqrt{\pi}} \sum_{m=1}^{M} w^*_m \prod_{t=1}^{n_i} F \left\{ y_{it}, x_{it} \beta + a^*_m \left( \frac{2\rho}{1-\rho} \right)^{1/2} \right\} \right]
\]

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

\[
\prod_{t=1}^{n_i} F(y_{it}, x_{it} \beta + \nu_i)
\]
is well approximated by a polynomial. As panel size and $\rho$ increase, the quadrature approximation can become less accurate. For large $\rho$, 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 $\epsilon_{it}$.

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**


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

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

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

[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