Title

stata.com

xtoprobit — Random-effects ordered probit models

Syntax | Menu | Description | Options
Remarks and examples | Stored results | Methods and formulas | References
Also see

Syntax

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

options	Description
Model	
`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`, `cluster` *clustvar*, `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, display of omitted variables and base and empty cells, and factor-variable labeling
Integration	
`intmethod(`*intmethod*`)`	integration method; *intmethod* may be `mvaghermite` (the default) or `ghermite`
`intpoints(`*#*`)`	use # quadrature points; default is `intpoints(12)`
Maximization	
maximize_options	control the maximization process; seldom used
`startgrid(`*numlist*`)`	improve starting value of the random-intercept parameter by performing a grid search
`nodisplay`	suppress display of header and coefficients
`coeflegend`	display legend instead of statistics

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

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

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

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

`startgrid()`, `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 > Ordinal outcomes > Probit regression (RE)

Description

`xtoprobit` fits random-effects ordered probit models. The actual values taken on by the dependent variable are irrelevant, although larger values are assumed to correspond to "higher" outcomes. The conditional distribution of the dependent variable given the random effects is assumed to be multinomial, with success probability determined by the standard normal cumulative distribution function.

Options

Model

`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 (`oim`), that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster` *clustvar*), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [XT] ***vce_options***.

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

Reporting

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

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

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

Integration

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

Maximization

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

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

`startgrid(`*numlist*`)` performs a grid search to improve the starting value of the random-intercept parameter. No grid search is performed by default unless the starting value is found to not be feasible; in this case, `xtoprobit` runs `startgrid(0.1 1 10)` and chooses the value that works best. You may already be using a default form of `startgrid()` without knowing it. If you see `xtoprobit` displaying Grid node 1, Grid node 2, ... following Grid node 0 in the iteration log, that is `xtoprobit` doing a default search because the original starting value was not feasible.

`nodisplay` is for programmers. It suppresses the display of the header and the coefficients.

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

Remarks and examples stata.com

`xtoprobit` fits random-effects ordered probit models. Ordered probit models are used to estimate relationships between an ordinal dependent variable and a set of independent variables. An *ordinal* variable is a variable that is categorical and ordered, for instance, “poor”, “good”, and “excellent”, which might indicate a person’s current health status or the repair record of a car. If there are only two outcomes, see [XT] **xtprobit**, [XT] **xtlogit**, and [XT] **xtcloglog**. This entry is concerned only with more than two outcomes.

▷ Example 1

We use the data from the “Television, School, and Family Smoking Prevention and Cessation Project” (Flay et al. 1988; Rabe-Hesketh and Skrondal 2012, chap. 11), where schools were randomly assigned into one of four groups defined by two treatment variables. Students within each school are nested in classes, and classes are nested in schools. In this example, we ignore the variability of classes within schools; see example 2 of [ME] **meoprobit** for a model that incorporates the additional class-level variance component. The dependent variable is the tobacco and health knowledge score (`thk`) collapsed into four ordered categories. We regress the outcome on the treatment variables and their interaction and control for the pretreatment score.

```
. use http://www.stata-press.com/data/r13/tvsfpors

. xtset school
       panel variable:  school (unbalanced)

. xtoprobit thk prethk cc##tv

Fitting comparison model:

Iteration 0:   log likelihood =  -2212.775
Iteration 1:   log likelihood = -2127.8111
Iteration 2:   log likelihood = -2127.7612
Iteration 3:   log likelihood = -2127.7612

Refining starting values:

Grid node 0:   log likelihood = -2149.7302

Fitting full model:

Iteration 0:   log likelihood = -2149.7302  (not concave)
Iteration 1:   log likelihood = -2129.6838  (not concave)
Iteration 2:   log likelihood = -2123.5143
Iteration 3:   log likelihood = -2122.2896
Iteration 4:   log likelihood = -2121.7949
Iteration 5:   log likelihood = -2121.7716
Iteration 6:   log likelihood = -2121.7715

Random-effects ordered probit regression        Number of obs      =      1600
Group variable: school                          Number of groups   =        28

Random effects u_i ~ Gaussian                   Obs per group: min =        18
                                                               avg =      57.1
                                                               max =       137

Integration method: mvaghermite                 Integration points =        12

                                                Wald chi2(4)       =    128.05
Log likelihood  = -2121.7715                    Prob > chi2        =    0.0000

------------------------------------------------------------------------------
         thk |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      prethk |   .2369804   .0227739    10.41   0.000     .1923444    .2816164
        1.cc |   .5490957   .1255108     4.37   0.000      .303099    .7950923
        1.tv |   .1695405   .1215889     1.39   0.163    -.0687693    .4078504
             |
       cc#tv |
        1 1  |  -.2951837   .1751969    -1.68   0.092    -.6385634    .0481959
-------------+----------------------------------------------------------------
       /cut1 |  -.0682011   .1003374    -0.68   0.497    -.2648587    .1284565
       /cut2 |     .67681   .1008836     6.71   0.000     .4790817    .8745382
       /cut3 |   1.390649   .1037494    13.40   0.000     1.187304    1.593995
-------------+----------------------------------------------------------------
   /sigma2_u |   .0288527   .0146201                      .0106874    .0778937
------------------------------------------------------------------------------
LR test vs. oprobit regression:  chibar2(01) =    11.98 Prob>=chibar2 = 0.0003
```

The estimation table reports the parameter estimates, the estimated cutpoints $(\kappa_1, \kappa_2, \kappa_3)$, and the estimated panel-level variance component labeled `sigma2_u`. The parameter estimates can be interpreted just as the output from a standard ordered probit regression would be interpreted; see [R] **oprobit**. For example, we find that students with higher preintervention scores tend to have higher postintervention scores.

Underneath the parameter estimates and the cutpoints, the table shows the estimated variance component. The estimate of σ_u^2 is 0.029 with standard error 0.015. The reported likelihood-ratio test shows that there is enough variability between schools to favor a random-effects ordered probit regression over a standard ordered probit regression.

◁

❑ Technical note

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

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

❑

Stored results

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

Scalars

`e(N)`	number of observations
`e(N_g)`	number of groups
`e(k)`	number of parameters
`e(k_aux)`	number of auxiliary parameters
`e(k_eq)`	number of equations in `e(b)`
`e(k_eq_model)`	number of equations in overall model test
`e(k_dv)`	number of dependent variables
`e(k_cat)`	number of categories
`e(df_m)`	model degrees of freedom
`e(ll)`	log likelihood
`e(ll_0)`	log likelihood, constant-only model
`e(ll_c)`	log likelihood, comparison model
`e(chi2)`	χ^2
`e(chi2_c)`	χ^2 for comparison test
`e(N_clust)`	number of clusters
`e(sigma_u)`	panel-level standard deviation
`e(n_quad)`	number of quadrature points
`e(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(g_max)`	largest group size
`e(p)`	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)`	`xtoprobit`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(covariates)`	list of covariates
`e(ivar)`	variable denoting groups
`e(title)`	title in estimation output
`e(clustvar)`	name of cluster variable
`e(offset)`	linear offset variable
`e(chi2type)`	`Wald` or `LR`; type of model χ^2 test
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. Err.

`e(intmethod)`	integration method
`e(distrib)`	`Gaussian`; the distribution of the random effect
`e(opt)`	type of optimization
`e(which)`	`max` or `min`; whether optimizer is to perform maximization or minimization
`e(ml_method)`	type of `ml` method
`e(user)`	name of likelihood-evaluator program
`e(technique)`	maximization technique
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(marginsok)`	predictions allowed by `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`

Matrices

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

Functions

`e(sample)`	marks estimation sample

Methods and formulas

`xtoprobit` fits via maximum likelihood the random-effects model

$$\Pr(y_{it} > k|\boldsymbol{\kappa}, \mathbf{x}_{it}, \nu_i) = \Phi(\mathbf{x}_{it}\boldsymbol{\beta} + \nu_i - \kappa_k)$$

for $i = 1, \dots, n$ panels, where $t = 1, \dots, n_i$, ν_i are independent and identically distributed $N(0, \sigma_\nu^2)$, and $\boldsymbol{\kappa}$ is a set of cutpoints κ_1, κ_2, ..., κ_{K-1}, where K is the number of possible outcomes; and $\Phi(\cdot)$ is the standard normal cumulative distribution function.

From the above, we can derive the probability of observing outcome k for response y_{it} as

$$\begin{aligned}
p_{itk} \equiv \Pr(y_{it} = k|\boldsymbol{\kappa}, \mathbf{x}_{it}, \nu_i) &= \Pr(\kappa_{k-1} < \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i + \epsilon_{it} \le \kappa_k) \\
&= \Pr(\kappa_{k-1} - \mathbf{x}_{it}\boldsymbol{\beta} - \nu_i < \epsilon_{it} \le \kappa_k - \mathbf{x}_{it}\boldsymbol{\beta} - \nu_i) \\
&= \Phi(\kappa_k - \mathbf{x}_{it}\boldsymbol{\beta} - \nu_i) - \Phi(\kappa_{k-1} - \mathbf{x}_{it}\boldsymbol{\beta} - \nu_i)
\end{aligned}$$

where κ_0 is taken as $-\infty$, and κ_K is taken as $+\infty$. Here $\mathbf{x}_{it}$ does not contain a constant term, because its effect is absorbed into the cutpoints.

We may also express this model in terms of a latent linear response, where observed ordinal responses y_{it} are generated from the latent continuous responses, such that

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

and

$$y_{it} = \begin{cases} 1 & \text{if} \quad y_{it}^* \le \kappa_1 \\ 2 & \text{if} \quad \kappa_1 < y_{it}^* \le \kappa_2 \\ \vdots & \\ K & \text{if} \quad \kappa_{K-1} < y_{it}^* \end{cases}$$

The errors ϵ_{it} are distributed as standard normal with mean zero and variance one and are independent of ν_i.

Given a set of panel-level random effects ν_i, we can define the conditional distribution for response y_{it} as

$$\begin{aligned} f(y_{it}, \boldsymbol{\kappa}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i) &= \prod_{k=1}^{K} p_{itk}^{I_k(y_{it})} \\ &= \exp \sum_{k=1}^{K} \left\{ I_k(y_{it}) \, \log(p_{itk}) \right\} \end{aligned}$$

where

$$I_k(y_{it}) = \begin{cases} 1 & \text{if } y_{it} = k \\ 0 & \text{otherwise} \end{cases}$$

For panel i, $i = 1, \ldots, M$, the conditional distribution of $\mathbf{y}_i = (y_{i1}, \ldots, y_{in_i})'$ is

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

and the panel-level likelihood l_i is given by

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

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

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

This is equivalent to

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

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

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

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

where $\widehat{\sigma}_i$ and $\widehat{\mu}_i$ are the adaptive parameters for panel i. The method of calculating the posterior mean and variance and using those parameters for $\widehat{\mu}_i$ and $\widehat{\sigma}_i$ is described in detail in Naylor and Smith (1982) and Skrondal and Rabe-Hesketh (2004). We start with $\widehat{\sigma}_{i,0} = 1$ and $\widehat{\mu}_{i,0} = 0$, and the posterior means and variances are updated in the jth iteration. That is, at the jth iteration of the optimization for l_i, we use

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

Letting

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

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

and

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

This is repeated until $\widehat{\mu}_{i,j}$ and $\widehat{\sigma}_{i,j}$ have converged for this iteration of the maximization algorithm. This adaptation is applied on every iteration.

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

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

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

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

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

xtoprobit and the robust VCE estimator

Specifying `vce(robust)` or `vce(cluster` *clustvar*`)` causes the Huber/White/sandwich VCE estimator to be calculated for the coefficients estimated in this regression. See [P] **_robust**, particularly *Introduction* and *Methods and formulas*. Wooldridge (2013) and Arellano (2003) discuss this application of the Huber/White/sandwich VCE estimator. As discussed by Wooldridge (2013), Stock and Watson (2008), and Arellano (2003), specifying `vce(robust)` is equivalent to specifying `vce(cluster` *panelvar*`)`, where *panelvar* is the variable that identifies the panels.

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

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

References

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

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

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

Flay, B. R., B. R. Brannon, C. A. Johnson, W. B. Hansen, A. L. Ulene, D. A. Whitney-Saltiel, L. R. Gleason, S. Sussman, M. D. Gavin, K. M. Glowacz, D. F. Sobol, and D. C. Spiegel. 1988. The television, school, and family smoking cessation and prevention project: I. Theoretical basis and program development. *Preventive Medicine* 17: 585–607.

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

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.

Rabe-Hesketh, S., and A. Skrondal. 2012. *Multilevel and Longitudinal Modeling Using Stata*. 3rd ed. College Station, TX: Stata Press.

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

Stock, J. H., and M. W. Watson. 2008. Heteroskedasticity-robust standard errors for fixed effects panel data regression. *Econometrica* 76: 155–174.

Twisk, J. W. R. 2013. *Applied Longitudinal Data Analysis for Epidemiology: A Practical Guide*. 2nd ed. Cambridge: Cambridge University Press.

Wooldridge, J. M. 2013. *Introductory Econometrics: A Modern Approach*. 5th ed. Mason, OH: South-Western.

Also see

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

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

[XT] **xtologit** — Random-effects ordered logistic models

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

[ME] **meoprobit** — Multilevel mixed-effects ordered probit regression

[R] **probit** — Probit regression

[U] 20 Estimation and postestimation commands