xtologit — Random-effects ordered logistic models

Syntax

xtologit 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
  nocoins report or 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.
xtologit — Random-effects ordered logistic models

Menu

Statistics > Longitudinal/panel data > Ordinal outcomes > Logistic regression (RE)

Description

xtologit fits random-effects ordered logistic 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 logistic 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 xtologit and the robust VCE estimator in Methods and formulas.

Reporting

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

or reports the estimated coefficients transformed to odds ratios, that is, \( e^b \) rather than \( b \). Standard errors and confidence intervals are similarly transformed. This option affects how results are displayed, not how they are estimated. or may be specified at estimation or when replaying previously estimated results.

noskip, nocnsreport; see [R] estimation options.

display_options: noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapping(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(#), itolerance(#), nrtolerance(#), nonrtolerance, and from(init_specs); see [R] maximize. These options are seldom used.
The following options are available with `xtologit` but are not shown in the dialog box:

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

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

`coeflegend`; see [R] estimation options.

### Remarks and examples

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

#### Example 1

We use the data from the “Television, School, and Family Smoking Prevention and Cessation Project” (Flay et al. 1988; Rabe-Hesketh and Skrondal 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] `meologit` for a model that incorporates the additional class-level variance component. The dependent variable is the tobacco and health knowledge score (`thk`) collapsed into four ordered categories. We regress the outcome on the treatment variables and their interaction and control for the pretreatment score.
. use http://www.stata-press.com/data/r13/tvsfpors
. xtset school
    panel variable: school (unbalanced)
. xtologit thk prethk cc##tv

Fitting comparison model:
Iteration 0:  log likelihood =  -2212.775
Iteration 1:  log likelihood = -2125.509
Iteration 2:  log likelihood = -2125.103
Iteration 3:  log likelihood = -2125.103

Refining starting values:
Grid node 0:  log likelihood = -2136.243

Fitting full model:
Iteration 0:  log likelihood = -2136.243 (not concave)
Iteration 1:  log likelihood = -2120.258
Iteration 2:  log likelihood = -2119.757
Iteration 3:  log likelihood = -2119.743
Iteration 4:  log likelihood = -2119.743

Random-effects ordered logistic regression

| thk    | Coef. | Std. Err. | z   | P>|z| | [95% Conf. Interval] |
|--------|-------|-----------|-----|-----|----------------------|
| prethk | .4032892 | .03886 | 10.38 | 0.000 | .327125 to .4794534 |
| 1.cc   | .9237904 | .204074 | 4.53 | 0.000 | .5238127 to 1.323768 |
| 1.tv   | .2749937 | .1977424 | 1.39 | 0.164 | -0.112574 to 0.6625618 |
| cc#tv  |      |         |     |     |                      |
| 1 1    | -.4659256 | .2845963 | -1.64 | 0.102 | -1.023724 to 0.0918728 |
| /cut1  |     |         |     |     |                      |
| /cut2  | 1.153364 | .165616 | 6.96 | 0.000 | .828762 to 1.477965 |
| /cut3  | 2.33195 | .1734199 | 13.45 | 0.000 | 1.992053 to 2.671846 |
| /sigma2_u | .0735112 | .0383106 | .0264695 | 0.2041551 |

LR test vs. ologit regression: chibar2(01) = 10.72 Prob>chibar2 = 0.0005

The estimation table reports the parameter estimates, the estimated cutpoints \((\kappa_1, \kappa_2, \kappa_3)\), and the estimated panel-level variance component labeled \(\sigma^2_u\). The parameter estimates can be interpreted just as the output from a standard ordered logistic regression would be interpreted; see [R] ologit. For example, we find that students with higher preintervention scores tend to have higher postintervention scores.

Underneath the parameter estimates and the cutpoints, the table shows the estimated variance component. The estimate of \(\sigma^2_u\) is 0.074 with standard error 0.038. The reported likelihood-ratio test shows that there is enough variability between schools to favor a random-effects ordered logistic regression over a standard ordered logistic regression.
Technical note

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

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

Stored results

xtologit stores the following in e():

Scalars

- e(N) number of observations
- e(N_g) number of groups
- e(k) number of parameters
- e(k_aux) number of auxiliary parameters
- e(k_eq) number of equations in e(b)
- e(k_eq_model) number of equations in overall model test
- e(k_dv) number of dependent variables
- e(k_cat) number of categories
- e(df_m) model degrees of freedom
- e(ll) log likelihood
- e(ll_0) log likelihood, constant-only model
- e(ll_c) log likelihood, comparison model
- e(chi2) $\chi^2$
- e(chi2_c) $\chi^2$ 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) xtologit
- 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 $\chi^2$ test
- e(vcetype) vcetype specified in vce()
xtologit fits via maximum likelihood the random-effects model

$$Pr(y_{it} > k | \kappa, x_{it}, \nu_i) = H(x_{it}\beta + \nu_i - \kappa_k)$$

for $i = 1, \ldots, n$ panels, where $t = 1, \ldots, n_i, \nu_i$ are independent and identically distributed $N(0, \sigma^2)$, and $\kappa$ is a set of cutpoints $\kappa_1, \kappa_2, \ldots, \kappa_{K-1}$, where $K$ is the number of possible outcomes; and $H(\cdot)$ is the logistic cumulative distribution function.

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

$$p_{itk} \equiv Pr(y_{it} = k | \kappa, x_{it}, \nu_i) = Pr(\kappa_{k-1} < x_{it}\beta + \nu_i + \epsilon_{it} \leq \kappa_k)$$

$$= Pr(\kappa_{k-1} - x_{it}\beta - \nu_i < \epsilon_{it} \leq \kappa_k - x_{it}\beta - \nu_i)$$

$$= H(\kappa_k - x_{it}\beta - \nu_i) - H(\kappa_{k-1} - x_{it}\beta - \nu_i)$$

$$= \frac{1}{1 + \exp(-\kappa_k + x_{it}\beta + \nu_i)} - \frac{1}{1 + \exp(-\kappa_{k-1} + x_{it}\beta + \nu_i)}$$

where $\kappa_0$ is taken as $-\infty$ and $\kappa_K$ is taken as $+\infty$. Here $x_{it}$ 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

$$y_{it}^* = x_{it}\beta + \nu_i + \epsilon_{it}$$

and

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

The errors $\epsilon_{it}$ are distributed as logistic with mean zero and variance $\pi^2/3$ and are independent of $\nu_i$. 
Given a set of panel-level random effects $\nu_i$, we can define the conditional distribution for response $y_{it}$ as

$$f(y_{it}, \kappa, x_{it}\beta + \nu_i) = \prod_{k=1}^{K} p_{itk}^{I_k(y_{it})}$$

$$= \exp \sum_{k=1}^{K} \left\{ I_k(y_{it}) \log(p_{itk}) \right\}$$

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 $y_i = (y_{i1}, \ldots, y_{in_i})'$ is

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

and the panel-level likelihood $l_i$ is given by

$$l_i(\beta, \kappa, \sigma^2_\nu) = \int_{-\infty}^{\infty} e^{-\nu_i^2/2\sigma^2_\nu} \left\{ \prod_{t=1}^{n_i} f(y_{it}, \kappa, x_{it}\beta + \nu_i) \right\} d\nu_i$$

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

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

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

This is equivalent to

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

where the $w_m^*$ denote the quadrature weights and the $a_m^*$ denote the quadrature absicssas. 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

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

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

$$l_{i,j} \approx \sum_{m=1}^{M} \sqrt{2\sigma_{i,j-1}} w_m^* \exp \left\{ (a_m^*)^2 \right\} g(y_{it}, \kappa, x_{it}, \sqrt{2\sigma_{i,j-1}}a_m^* + \hat{\mu}_{i,j-1})$$
Letting
\[ \tau_{i,m,j-1} = \sqrt{2} \tilde{\sigma}_{i,j-1} a_m^* + \tilde{\mu}_{i,j-1} \]

\[ \hat{\mu}_{i,j} = \sum_{m=1}^{M} \left( \tau_{i,m,j-1} \right) \frac{\sqrt{2} \tilde{\sigma}_{i,j-1} w_m^* \exp\left\{ (a_m^*)^2 \right\} g(y_{it}, \kappa, x_{it}, \tau_{i,m,j-1})}{l_{i,j}} \]

and
\[ \hat{\sigma}_{i,j} = \sum_{m=1}^{M} \left( \tau_{i,m,j-1} \right)^2 \frac{\sqrt{2} \tilde{\sigma}_{i,j-1} w_m^* \exp\left\{ (a_m^*)^2 \right\} g(y_{it}, \kappa, x_{it}, \tau_{i,m,j-1})}{l_{i,j}} - (\hat{\mu}_{i,j})^2 \]

This is repeated until \( \hat{\mu}_{i,j} \) and \( \hat{\sigma}_{i,j} \) 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 \( \rho = \sigma^2_\nu / (\sigma^2_\nu + 1) \):

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

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
\[
\prod_{t=1}^{n_i} f(y_{it}, \kappa, x_{it}\beta + \nu_i)
\]
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.

**xtologit and the robust VCE estimator**

Specifying vce(robust) or vce(cluster clustvar) causes the Huber/White/sandwich VCE estimator to be calculated for the coefficients estimated in this regression. See [P] _robust, particularly Introduction and Methods and formulas. Wooldridge (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 \( \epsilon_{it} \).

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] xtologit postestimation — Postestimation tools for xtologit

[XT] quadchk — Check sensitivity of quadrature approximation

[XT] xtprobit — Random-effects ordered probit models

[XT] xtset — Declare data to be panel data

[ME] meologit — Multilevel mixed-effects ordered logistic regression

[R] logistic — Logistic regression, reporting odds ratios

[R] logit — Logistic regression, reporting coefficients

[U] 20 Estimation and postestimation commands