

xttobit — Random-effects tobit models

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

Description

`xttobit` fits random-effects tobit models for panel data where the outcome variable is censored. Censoring limits may be fixed for all observations or vary across observations. The user can request that a likelihood-ratio test comparing the panel tobit model with the pooled tobit model be conducted at estimation time.

Quick start

Tobit model of y on x where y is censored at a lower limit of 5 using `xtset` data

```
xttobit y x, ll(5)
```

Add [indicators](#) for levels of categorical variable `a`

```
xttobit y x i.a, ll(5)
```

Same as above, but specify that censoring occurs at 5 and 25

```
xttobit y x i.a, ll(5) ul(25)
```

Same as above, but where `lower` and `upper` are variables containing the censoring limits

```
xttobit y x i.a, ll(lower) ul(upper)
```

Add likelihood-ratio test comparing the random-effects model with the pooled model

```
xttobit y x i.a, ll(lower) ul(upper) tobit
```

Menu

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

Syntax

```
xttobit depvar [indepvars] [if] [in] [weight] [, options]
```

<i>options</i>	Description
Model	
<code>noconstant</code>	suppress constant term
<code>ll</code> [<i>(varname #)</i>]	left-censoring variable or limit
<code>ul</code> [<i>(varname #)</i>]	right-censoring variable or limit
<code>offset</code> (<i>varname</i>)	include <i>varname</i> in model with coefficient constrained to 1
<code>constraints</code> (<i>constraints</i>)	apply specified linear constraints
SE	
<code>vce</code> (<i>vcetype</i>)	<i>vcetype</i> may be <code>oim</code> , <code>bootstrap</code> , or <code>jackknife</code>
Reporting	
<code>level</code> (#)	set confidence level; default is <code>level(95)</code>
<code>tobit</code>	perform likelihood-ratio test comparing against pooled tobit model
<code>lrmodel</code>	perform the likelihood-ratio model test instead of the default Wald test
<code>nocnsreport</code>	do not display constraints
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Integration	
<code>intmethod</code> (<i>intmethod</i>)	integration method; <i>intmethod</i> may be <code>mvaghermite</code> (the default) or <code>ghermite</code>
<code>intpoints</code> (#)	use # quadrature points; default is <code>intpoints(12)</code>
Maximization	
<code>maximize_options</code>	control the maximization process; seldom used
<code>collinear</code>	keep collinear variables
<code>coeflegend</code>	display legend instead of statistics

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

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

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

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

weights are allowed; 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

Model

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

`ll`[*(varname | #)*] and `ul`[*(varname | #)*] indicate the lower and upper limits for censoring, respectively. Observations with *depvar* ≤ `ll`() are left-censored; observations with *depvar* ≥ `ul`() are right-censored; and remaining observations are not censored. You do not have to specify the

censoring values. If you specify `ll`, the lower limit is the minimum of `depvar`. If you specify `ul`, the upper limit is the maximum of `depvar`.

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

SE

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

Reporting

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

`tobit` specifies that a likelihood-ratio test comparing the random-effects model with the pooled (tobit) model be included in the output.

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

`display_options`: `nocl`, `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 `xttobit` but are not shown in the dialog box:

`collinear`, `coeflegend`; see [R] [Estimation options](#).

Remarks and examples

[stata.com](http://www.stata.com)

`xttobit` fits random-effects tobit models. There is no command for a fixed-effects model, because there does not exist a sufficient statistic allowing the fixed effects to be conditioned out of the likelihood.

Consider the linear regression model with panel-level random effects

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

for $i = 1, \dots, n$ panels, where $t = 1, \dots, n_i$. The random effects, ν_i , are i.i.d., $N(0, \sigma_\nu^2)$, and ϵ_{it} are i.i.d. $N(0, \sigma_\epsilon^2)$ independently of ν_i .

The observed data, y_{it}^o , represent possibly censored versions of y_{it} . If they are left-censored, all that is known is that $y_{it} \leq y_{it}^o$. If they are right-censored, all that is known is that $y_{it} \geq y_{it}^o$. If they are uncensored, $y_{it} = y_{it}^o$. If they are left-censored, y_{it}^o is determined by `ll()`. If they are right-censored, y_{it}^o is determined by `ul()`. If they are uncensored, y_{it}^o is determined by `depvar`.

▷ Example 1: Random-effects tobit regression

Using the `nlswork` data described in [XT] `xt`, we fit a random-effects tobit model of adjusted (log) wages. We use the `ul()` option to impose an upper limit on the recorded log of wages.

```
. use https://www.stata-press.com/data/r18/nlswork3
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)
. xttobit ln_wage i.union age grade not_smsa south##c.year, ul(1.9) tobit
(output omitted)

Random-effects tobit regression                Number of obs   = 19,224
                                                Uncensored     = 12,334
Limits: Lower = -inf                        Left-censored  = 0
        Upper = 1.90                        Right-censored = 6,890
Group variable: idcode                       Number of groups = 4,148
Random effects u_i ~ Gaussian                Obs per group:
                                                min = 1
                                                avg = 4.6
                                                max = 12

Integration method: mvaghermite              Integration pts. = 12
                                                Wald chi2(7)    = 2925.68
Log likelihood = -6814.4606                  Prob > chi2     = 0.0000
```

ln_wage	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
1.union	.1430527	.0069718	20.52	0.000	.1293883	.1567171
age	.0099132	.0017516	5.66	0.000	.0064801	.0133464
grade	.0784855	.0022764	34.48	0.000	.0740239	.0829472
not_smsa	-.1339978	.009206	-14.56	0.000	-.1520413	-.1159544
1.south	-.3507188	.0695554	-5.04	0.000	-.4870449	-.2143928
year	-.0008285	.0018371	-0.45	0.652	-.0044292	.0027721
south#c.year						
1	.0031938	.0008606	3.71	0.000	.0015071	.0048805
_cons	.5101956	.1006646	5.07	0.000	.3128966	.7074946
/sigma_u	.3045992	.0048344	63.01	0.000	.2951239	.3140745
/sigma_e	.2488678	.0018254	136.34	0.000	.24529	.2524455
rho	.5996844	.0084095			.583118	.6160734

```
LR test of sigma_u=0: chibar2(01) = 6650.63          Prob >= chibar2 = 0.000
```

The results from a tobit regression can be interpreted as we would those from a linear regression. Because the dependent variable is log transformed, the coefficients can be interpreted in terms of a percentage change. We see, for example, that on average, union members make 14.3% more than nonunion members.

The output also includes the overall and panel-level variance components (labeled `sigma_e` and `sigma_u`, respectively) together with ρ (labeled `rho`)

$$\rho = \frac{\sigma_u^2}{\sigma_e^2 + \sigma_u^2}$$

which is the percent contribution to the total variance of the panel-level variance component.

When `rho` is zero, the panel-level variance component is unimportant, and the panel estimator is not different from the pooled estimator. A likelihood-ratio test of this is included at the bottom of the output. This test formally compares the pooled estimator (tobit) with the panel estimator. In this case, we reject the null hypothesis that there are no panel-level effects.

□ Technical note

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

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

□

Stored results

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

Scalars

<code>e(N)</code>	number of observations
<code>e(N_g)</code>	number of groups
<code>e(N_unc)</code>	number of uncensored observations
<code>e(N_lc)</code>	number of left-censored observations
<code>e(N_rc)</code>	number of right-censored observations
<code>e(N_cd)</code>	number of completely determined observations
<code>e(k)</code>	number of parameters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_eq_model)</code>	number of equations in overall model test
<code>e(k_dv)</code>	number of dependent variables
<code>e(df_m)</code>	model degrees of freedom
<code>e(ll)</code>	log likelihood
<code>e(ll_0)</code>	log likelihood, constant-only model
<code>e(ll_c)</code>	log likelihood, comparison model
<code>e(chi2)</code>	χ^2
<code>e(chi2_c)</code>	χ^2 for comparison test
<code>e(rho)</code>	ρ
<code>e(sigma_u)</code>	panel-level standard deviation
<code>e(sigma_e)</code>	standard deviation of ϵ_{it}
<code>e(n_quad)</code>	number of quadrature points
<code>e(g_min)</code>	smallest group size
<code>e(g_avg)</code>	average group size
<code>e(g_max)</code>	largest group size
<code>e(p)</code>	p -value for model test
<code>e(rank)</code>	rank of <code>e(V)</code>
<code>e(rank0)</code>	rank of <code>e(V)</code> for constant-only model
<code>e(ic)</code>	number of iterations
<code>e(rc)</code>	return code
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	<code>xttobit</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	names of dependent variables
<code>e(ivar)</code>	variable denoting groups
<code>e(llopt)</code>	contents of <code>ll()</code>
<code>e(ulopt)</code>	contents of <code>ul()</code>
<code>e(k_aux)</code>	number of auxiliary parameters
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(offset1)</code>	offset
<code>e(chi2type)</code>	Wald or LR; type of model χ^2 test
<code>e(chi2_ct)</code>	Wald or LR; type of model χ^2 test corresponding to <code>e(chi2_c)</code>
<code>e(vce)</code>	<i>vce</i> type specified in <code>vce()</code>
<code>e(intmethod)</code>	integration method
<code>e(distrib)</code>	Gaussian; the distribution of the random effect
<code>e(opt)</code>	type of optimization
<code>e(which)</code>	max or min; whether optimizer is to perform maximization or minimization
<code>e(ml_method)</code>	type of ml method
<code>e(user)</code>	name of likelihood-evaluator program
<code>e(technique)</code>	maximization technique
<code>e(properties)</code>	b V
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsok)</code>	predictions allowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(Cns)</code>	constraints matrix
<code>e(ilog)</code>	iteration log
<code>e(gradient)</code>	gradient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

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

Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, <i>p</i> -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

Assuming a normal distribution, $N(0, \sigma_\nu^2)$, for the random effects ν_i , we have the joint (unconditional of ν_i) density of the observed data from the *i*th panel

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

where

$$F(y_{it}^o, \Delta_{it}) = \begin{cases} (\sqrt{2\pi}\sigma_\epsilon)^{-1} e^{-(y_{it}^o - \Delta_{it})^2 / (2\sigma_\epsilon^2)} & \text{if } y_{it}^o \in C \\ \Phi\left(\frac{y_{it}^o - \Delta_{it}}{\sigma_\epsilon}\right) & \text{if } y_{it}^o \in L \\ 1 - \Phi\left(\frac{y_{it}^o - \Delta_{it}}{\sigma_\epsilon}\right) & \text{if } y_{it}^o \in R \end{cases}$$

where C is the set of noncensored observations, L is the set of left-censored observations, R is the set of right-censored observations, and $\Phi(\cdot)$ is the cumulative normal distribution.

The panel level likelihood l_i is given by

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

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

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

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

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

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

where $\hat{\sigma}_i$ and $\hat{\mu}_i$ are the adaptive parameters for panel i . Therefore, with the definition of $g(y_{it}^o, x_{it}, \nu_i)$, the total log likelihood is approximated by

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

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

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

Letting

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

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

and

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

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 $1e-6$; after this, the quadrature parameters are fixed.

The log likelihood can also be calculated by nonadaptive Gauss–Hermite quadrature if the `intmethod(ghermite)` option is specified. For nonadaptive Gauss–Hermite quadrature, the following formula for the log likelihood is used in place of (1).

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

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

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

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

References

- 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. <https://doi.org/10.2307/2347995>.
- 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>.
- Skrondal, A., and S. Rabe-Hesketh. 2004. *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models*. Boca Raton, FL: Chapman and Hall/CRC.

Also see

- [XT] **xttobit postestimation** — Postestimation tools for xttobit
- [XT] **quadchk** — Check sensitivity of quadrature approximation
- [XT] **xteintreg** — Extended random-effects interval regression
- [XT] **xtintreg** — Random-effects interval-data regression models
- [XT] **xtreg** — Fixed-, between-, and random-effects and population-averaged linear models
- [XT] **xtset** — Declare data to be panel data
- [ME] **metobit** — Multilevel mixed-effects tobit regression
- [R] **tobit** — Tobit regression
- [U] **20 Estimation and postestimation commands**

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2023 StataCorp LLC, College Station, TX, USA. All rights reserved.

