

**metobit** — Multilevel mixed-effects tobit regression

[Description](#)  
[Options](#)  
[Also see](#)

[Quick start](#)  
[Remarks and examples](#)

[Menu](#)  
[Stored results](#)

[Syntax](#)  
[Methods and formulas](#)

## Description

`metobit` fits mixed-effects models for continuous responses where the outcome variable is censored. Censoring limits may be fixed for all observations or vary across observations.

## Quick start

### *Without weights*

Two-level tobit regression of  $y$  on  $x$  with random intercepts by `lev2` where  $y$  is censored at a lower limit of 5

```
metobit y x || lev2:, ll(5)
```

As above, but specify that left-censoring occurs at 5 and right-censoring occurs at 25

```
metobit y x || lev2:, ll(5) ul(25)
```

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

```
metobit y x || lev2:, ll(lower) ul(upper)
```

Mixed-effects model adding random coefficients for  $x$

```
metobit y x || lev2: x, ll(5)
```

Three-level random-intercept model of  $y$  on  $x$  with `lev2` nested within `lev3`

```
metobit y x || lev3: || lev2:, ll(5)
```

Crossed-effects model of  $y$  on  $x$  with two-way crossed random effects by factors `a` and `b`

```
metobit y x || _all:R.a || b:, ll(5)
```

### *With weights*

Two-level tobit regression of  $y$  on  $x$  with random intercepts by `lev2` and observation-level frequency weights `wvar1`

```
metobit y x [fweight=wvar1] || lev2:, ll(5)
```

Two-level random-intercept model from a two-stage sampling design with PSUs identified by `psu` using PSU-level and observation-level sampling weights `wvar2` and `wvar1`, respectively

```
metobit y x [pweight=wvar1] || psu:, pweight(wvar2) ll(5)
```

Same as above, but `svyset` data first

```
svyset psu, weight(wvar2) || _n, weight(wvar1)
svy: metobit y x || psu:, ll(5)
```

## Menu

Statistics > Multilevel mixed-effects models > Tobit regression

## Syntax

```
metobit depvar fe_equation [ || re_equation ] [ || re_equation ... ] [ , options ]
```

where the syntax of *fe\_equation* is

```
[ indepvars ] [ if ] [ in ] [ weight ] [ , fe_options ]
```

and the syntax of *re\_equation* is one of the following:

for random coefficients and intercepts

```
levelvar: [ varlist ] [ , re_options ]
```

for random effects among the values of a factor variable

```
levelvar: R.varname
```

*levelvar* is a variable identifying the group structure for the random effects at that level or is `_all` representing one group comprising all observations.

<i>fe_options</i>	Description
Model	
<code>noconstant</code>	suppress constant term from the fixed-effects equation
<code>offset(<i>varname</i>)</code>	include <i>varname</i> in model with coefficient constrained to 1

<i>re_options</i>	Description
Model	
<code>covariance(<i>vartype</i>)</code>	variance–covariance structure of the random effects
<code>noconstant</code>	suppress constant term from the random-effects equation
<code>fweight(<i>varname</i>)</code>	frequency weights at higher levels
<code>iweight(<i>varname</i>)</code>	importance weights at higher levels
<code>pweight(<i>varname</i>)</code>	sampling weights at higher levels

<i>options</i>	Description
Model	
<u>ll</u> [ ( <i>varname</i>   #) ]	left-censoring variable or limit
<u>ul</u> [ ( <i>varname</i>   #) ]	right-censoring variable or limit
<u>constraints</u> ( <i>constraints</i> )	apply specified linear constraints
<u>collinear</u>	keep collinear variables
SE/Robust	
<u>vce</u> ( <i>vcetype</i> )	<i>vcetype</i> may be <u>oim</u> , <u>robust</u> , or <u>cluster</u> <i>clustvar</i>
Reporting	
<u>level</u> (#)	set confidence level; default is <u>level</u> (95)
<u>nocnsreport</u>	do not display constraints
<u>notable</u>	suppress coefficient table
<u>noheader</u>	suppress output header
<u>nogroup</u>	suppress table summarizing groups
<u>display_options</u>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Integration	
<u>intmethod</u> ( <i>intmethod</i> )	integration method
<u>intpoints</u> (#)	set the number of integration (quadrature) points for all levels; default is <u>intpoints</u> (7)
Maximization	
<u>maximize_options</u>	control the maximization process; seldom used
<u>startvalues</u> ( <i>svmethod</i> )	method for obtaining starting values
<u>startgrid</u> [ ( <i>gridspec</i> ) ]	perform a grid search to improve starting values
<u>noestimate</u>	do not fit the model; show starting values instead
<u>dnnumerical</u>	use numerical derivative techniques
<u>coeflegend</u>	display legend instead of statistics
<i>vartype</i>	Description
<u>independent</u>	one unique variance parameter per random effect, all covariances 0; the default unless the R. notation is used
<u>exchangeable</u>	equal variances for random effects, and one common pairwise covariance
<u>identity</u>	equal variances for random effects, all covariances 0; the default if the R. notation is used
<u>unstructured</u>	all variances and covariances to be distinctly estimated
<u>fixed</u> ( <i>matname</i> )	user-selected variances and covariances constrained to specified values; the remaining variances and covariances unrestricted
<u>pattern</u> ( <i>matname</i> )	user-selected variances and covariances constrained to be equal; the remaining variances and covariances unrestricted

<i>intmethod</i>	Description
<u>m</u> vaghermite	mean–variance adaptive Gauss–Hermite quadrature; the default unless a crossed random-effects model is fit
<u>m</u> caghermite	mode-curvature adaptive Gauss–Hermite quadrature
<u>g</u> hermite	nonadaptive Gauss–Hermite quadrature
<u>l</u> aplace	Laplacian approximation; the default for crossed random-effects models

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

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

*bayes*, *by*, and *svy* are allowed; see [U] 11.1.10 **Prefix commands**. For more details, see [BAYES] **bayes: metobit**.

*vce()* and weights are not allowed with the *svy* prefix; see [SVY] **svy**.

*fweights*, *iweights*, and *pweights* are allowed; see [U] 11.1.6 **weight**. Only one type of weight may be specified.

Weights are not supported under the Laplacian approximation or for crossed models.

*startvalues()*, *startgrid*, *noestimate*, *dnumerical*, 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* suppresses the constant (intercept) term and may be specified for the fixed-effects equation and for any of or all the random-effects equations.

*ll*[(*varname* | #)] and *ul*[(*varname* | #)] indicate the censoring points. You may specify one or both. *ll*() indicates the lower limit for left-censoring. *ul*() indicates the upper limit for right-censoring. Observations with *depvar* ≤ *ll*() are left-censored, observations with *depvar* ≥ *ul*() are right-censored, and remaining observations are not censored.

*offset*(*varname*) specifies that *varname* be included in the fixed-effects portion of the model with the coefficient constrained to be 1.

*covariance*(*vartype*) specifies the structure of the covariance matrix for the random effects and may be specified for each random-effects equation. *vartype* is one of the following: **independent**, **exchangeable**, **identity**, **unstructured**, **fixed**(*matname*), or **pattern**(*matname*).

*covariance*(**independent**) covariance structure allows for a distinct variance for each random effect within a random-effects equation and assumes that all covariances are 0. The default is *covariance*(**independent**) unless a crossed random-effects model is fit, in which case the default is *covariance*(**identity**).

*covariance*(**exchangeable**) structure specifies one common variance for all random effects and one common pairwise covariance.

*covariance*(**identity**) is short for “multiple of the identity”; that is, all variances are equal and all covariances are 0.

*covariance*(**unstructured**) allows for all variances and covariances to be distinct. If an equation consists of *p* random-effects terms, the unstructured covariance matrix will have  $p(p + 1)/2$  unique parameters.

*covariance*(**fixed**(*matname*)) and *covariance*(**pattern**(*matname*)) covariance structures provide a convenient way to impose constraints on variances and covariances of random effects. Each specification requires a *matname* that defines the restrictions placed on variances and

covariances. Only elements in the lower triangle of *matname* are used, and row and column names of *matname* are ignored. A missing value in *matname* means that a given element is unrestricted. In a `fixed(matname)` covariance structure, (co)variance  $(i, j)$  is constrained to equal the value specified in the  $i, j$ th entry of *matname*. In a `pattern(matname)` covariance structure, (co)variances  $(i, j)$  and  $(k, l)$  are constrained to be equal if  $matname[i, j] = matname[k, l]$ .

`fweight(varname)` specifies frequency weights at higher levels in a multilevel model, whereas frequency weights at the first level (the observation level) are specified in the usual manner, for example, `[fw=fwtvar1]`. *varname* can be any valid Stata variable name, and you can specify `fweight()` at levels two and higher of a multilevel model. For example, in the two-level model

```
. mecmd fixed_portion [fw = wt1] || school: ... , fweight(wt2) ...
```

the variable `wt1` would hold the first-level (the observation-level) frequency weights, and `wt2` would hold the second-level (the school-level) frequency weights.

`iweight(varname)` specifies importance weights at higher levels in a multilevel model, whereas importance weights at the first level (the observation level) are specified in the usual manner, for example, `[iw=iwtvar1]`. *varname* can be any valid Stata variable name, and you can specify `iweight()` at levels two and higher of a multilevel model. For example, in the two-level model

```
. mecmd fixed_portion [iw = wt1] || school: ... , iweight(wt2) ...
```

the variable `wt1` would hold the first-level (the observation-level) importance weights, and `wt2` would hold the second-level (the school-level) importance weights.

`pweight(varname)` specifies sampling weights at higher levels in a multilevel model, whereas sampling weights at the first level (the observation level) are specified in the usual manner, for example, `[pw=pwtvar1]`. *varname* can be any valid Stata variable name, and you can specify `pweight()` at levels two and higher of a multilevel model. For example, in the two-level model

```
. mecmd fixed_portion [pw = wt1] || school: ... , pweight(wt2) ...
```

variable `wt1` would hold the first-level (the observation-level) sampling weights, and `wt2` would hold the second-level (the school-level) sampling weights.

`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`), and that allow for intragroup correlation (`cluster clustvar`); see [\[R\] vce\\_option](#). If `vce(robust)` is specified, robust variances are clustered at the highest level in the multilevel model.

#### Reporting

`level(#)`, `nocnsreport`; see [\[R\] estimation options](#).

`notable` suppresses the estimation table, either at estimation or upon replay.

`noheader` suppresses the output header, either at estimation or upon replay.

`nogroup` suppresses the display of group summary information (number of groups, average group size, minimum, and maximum) from the output header.

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

## Integration

`intmethod(intmethod)` specifies the integration method to be used for the random-effects model. `mvaghermite` performs mean–variance adaptive Gauss–Hermite quadrature; `mcaghermite` performs mode-curvature adaptive Gauss–Hermite quadrature; `ghermite` performs nonadaptive Gauss–Hermite quadrature; and `laplace` performs the Laplacian approximation, equivalent to mode-curvature adaptive Gaussian quadrature with one integration point.

The default integration method is `mvaghermite` unless a crossed random-effects model is fit, in which case the default integration method is `laplace`. The Laplacian approximation has been known to produce biased parameter estimates; however, the bias tends to be more prominent in the estimates of the variance components rather than in the estimates of the fixed effects.

For crossed random-effects models, estimation with more than one quadrature point may be prohibitively intensive even for a small number of levels. For this reason, the integration method defaults to the Laplacian approximation. You may override this behavior by specifying a different integration method.

`intpoints(#)` sets the number of integration points for quadrature. The default is `intpoints(7)`, which means that seven quadrature points are used for each level of random effects. This option is not allowed with `intmethod(laplace)`.

The more integration points, the more accurate the approximation to the log likelihood. However, computation time increases as a function of the number of quadrature points raised to a power equaling the dimension of the random-effects specification. In crossed random-effects models and in models with many levels or many random coefficients, this increase can be substantial.

## 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](#). Those that require special mention for `metobit` are listed below.

`from()` accepts a properly labeled vector of initial values or a list of coefficient names with values. A list of values is not allowed.

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

`startvalues(svmethod)`, `startgrid[(gridspec)]`, `noestimate`, and `dnumerical`; see [ME] [meglm](#).

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

## Remarks and examples

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

Mixed-effects tobit regression is tobit regression containing both fixed effects and random effects. In longitudinal data and panel data, random effects are useful for modeling intracluster correlation; that is, observations in the same cluster are correlated because they share common cluster-level random effects.

In a mixed-effects tobit regression, the values of the outcome variable may be observed, unobserved but known to fall below a given limit (left-censored data), or unobserved but known to fall above a given limit (right-censored data). That is, the observed data,  $y_{ij}^*$ , represent possibly censored versions of  $y_{ij}$  for the  $i$ th observation within the  $j$ th cluster.

The observed outcome is therefore defined as

$$y_{ij}^* = \begin{cases} y_{ij} & \text{if } a < y_{ij} < b \\ a & \text{if } y_{ij} \leq a \\ b & \text{if } y_{ij} \geq b \end{cases}$$

where  $a$  is the lower-censoring limit and  $b$  is the upper-censoring limit. If the data are uncensored,  $y_{ij}^* = y_{ij}$ , and the value is determined by the value of the outcome variable. If they are left-censored, all that is known is that  $y_{ij} \leq a$  and  $y_{ij}^*$  is determined by `ll()`. If they are right-censored, all that is known is that  $y_{ij} \geq b$  and  $y_{ij}^*$  is determined by `ul()`. The censoring limits specified in `ll()` and `ul()` can be the same for all observations or can vary from observation to observation.

Regardless of the type of censoring, the expected value of the underlying dependent variable—say,  $y$ —is modeled using the following linear prediction:

$$E(y|\mathbf{X}, \mathbf{u}) = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} \quad (1)$$

$\mathbf{X}$  is an  $n \times p$  design/covariate matrix, analogous to the covariates you would find in a standard linear regression model, with regression coefficients (fixed effects)  $\boldsymbol{\beta}$ .  $\mathbf{Z}$  is the  $n \times q$  design/covariate matrix for the random effects  $\mathbf{u}$ . This linear prediction also contains the offset when `offset()` is specified.

The columns of matrix  $\mathbf{Z}$  are the covariates corresponding to the random effects and can be used to represent both random intercepts and random coefficients. For example, in a random-intercepts model,  $\mathbf{Z}$  is simply the scalar 1. The random effects  $\mathbf{u}$  are realizations from a multivariate normal distribution with mean  $\mathbf{0}$  and  $q \times q$  variance matrix  $\boldsymbol{\Sigma}$ . The random effects are not directly estimated as model parameters but are instead summarized according to the unique elements of  $\boldsymbol{\Sigma}$ , known as variance components. One special case of (1) places  $\mathbf{Z} = \mathbf{X}$  so that all covariate effects are essentially random and distributed as multivariate normal with mean  $\boldsymbol{\beta}$  and variance  $\boldsymbol{\Sigma}$ .

Below we present a short example of mixed-effects tobit regression; refer to [\[ME\] me](#) and [\[ME\] meglm](#) for additional examples of random-effects models. A two-level tobit model can also be fit using `xttobit`; see [\[XT\] xttobit](#). In the absence of random effects, mixed-effects tobit regression reduces to standard tobit regression; see [\[R\] tobit](#).

### ► Example 1: Random-intercept model

We have wage data on young women who were between ages 14 and 26 in 1968 and who were surveyed over the period 1968–1988; see [\[XT\] xt](#) for a more detailed discussion of the data. We are interested in the effect of completed years of schooling, current age, union membership, and residence in the South on wages.

```
. use http://www.stata-press.com/data/r15/nlswork
(National Longitudinal Survey. Young Women 14-26 years of age in 1968)
```

We fit a mixed-effects tobit model of the log of inflation-adjusted wages (`ln_wage`). For illustration purposes, we use the `ul()` option to impose an artificial upper limit at 1.96, the 75th percentile of the recorded log wages.

```

. metobit ln_wage i.union age south##c.grade || idcode:, ul(1.96)
Fitting fixed-effects model:
Iteration 0:   log likelihood = -11628.188
Iteration 1:   log likelihood = -10617.455
Iteration 2:   log likelihood = -10555.304
Iteration 3:   log likelihood = -10554.78
Iteration 4:   log likelihood = -10554.78
Refining starting values:
Grid node 0:   log likelihood = -10225.917
Fitting full model:
Iteration 0:   log likelihood = -10225.917 (not concave)
Iteration 1:   log likelihood = -8728.9674 (not concave)
Iteration 2:   log likelihood = -7827.6894 (not concave)
Iteration 3:   log likelihood = -7112.0272
Iteration 4:   log likelihood = -6894.0253
Iteration 5:   log likelihood = -6821.7055
Iteration 6:   log likelihood = -6818.5592
Iteration 7:   log likelihood = -6818.5512
Iteration 8:   log likelihood = -6818.5512
Mixed-effects tobit regression
Limits: lower = -inf
        upper = 1.96
Group variable:      idcode
Number of obs       =      19,224
Uncensored          =      13,188
Left-censored       =           0
Right-censored      =       6,036
Number of groups    =       4,148
Obs per group:
    min =           1
    avg =           4.6
    max =           12
Integration method: mvaghermite
Integration pts.    =           7
Wald chi2(5)       =      2812.43
Prob > chi2        =       0.0000
Log likelihood = -6818.5512

```

ln_wage	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
1.union	.1418088	.0068398	20.73	0.000	.1284029	.1552146
age	.0107585	.0004068	26.45	0.000	.0099612	.0115559
1.south	-.2373995	.048346	-4.91	0.000	-.3321559	-.1426431
grade	.0763865	.0029104	26.25	0.000	.0706822	.0820909
south#						
c.grade						
1	.0099306	.0037452	2.65	0.008	.0025902	.0172709
_cons	.4146363	.0396691	10.45	0.000	.3368864	.4923862
idcode						
var(_cons)	.0985482	.003018			.0928071	.1046444
var(e.ln_w~e)	.0619327	.000876			.0602394	.0636736

LR test vs. tobit model: chibar2(01) = 7472.46      Prob >= chibar2 = 0.0000

The estimation table reports the fixed effects, which are interpreted just as you would the output from `tobit`, and the estimated variance components. Because the dependent variable is log transformed, the fixed-effects coefficients can be interpreted in terms of a percent change. For example, we see that on average, union members make 14.2% more than nonunion members and that each additional year of age is associated with a 1.1% increase in wages.



The random-effects equation is labeled `idcode`. The estimated variance of the subject-specific random intercept is 0.098 with standard error 0.003. A likelihood-ratio test comparing the model with a tobit model without random effects is provided under the table and indicates that the two-level tobit model is preferred.



## Stored results

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

### Scalars

<code>e(N)</code>	number of observations
<code>e(N_unc)</code>	number of uncensored observations
<code>e(N_lrc)</code>	number of left-censored observations
<code>e(N_rrc)</code>	number of right-censored observations
<code>e(k)</code>	number of parameters
<code>e(k_dv)</code>	number of dependent variables
<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_f)</code>	number of fixed-effects parameters
<code>e(k_r)</code>	number of random-effects parameters
<code>e(k_rsr)</code>	number of variances
<code>e(k_rc)</code>	number of covariances
<code>e(df_m)</code>	model degrees of freedom
<code>e(ll)</code>	log likelihood
<code>e(N_clust)</code>	number of clusters
<code>e(chi2)</code>	$\chi^2$
<code>e(p)</code>	$p$ -value for model test
<code>e(ll_c)</code>	log likelihood, comparison model
<code>e(chi2_c)</code>	$\chi^2$ , comparison test
<code>e(df_c)</code>	degrees of freedom, comparison test
<code>e(p_c)</code>	$p$ -value for comparison test
<code>e(rank)</code>	rank of <code>e(V)</code>
<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>meglm</code>
<code>e(cmd2)</code>	<code>metobit</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	names of dependent variables
<code>e(llopt)</code>	minimum of <code>depvar</code> or contents of <code>ll()</code>
<code>e(ulopt)</code>	maximum of <code>depvar</code> or contents of <code>ul()</code>
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression (first-level weights)
<code>e(fweight<math>k</math>)</code>	<code>fweight</code> variable for $k$ th highest level, if specified
<code>e(iweight<math>k</math>)</code>	<code>iweight</code> variable for $k$ th highest level, if specified
<code>e(pweight<math>k</math>)</code>	<code>pweight</code> variable for $k$ th highest level, if specified
<code>e(covariates)</code>	list of covariates
<code>e(ivars)</code>	grouping variables
<code>e(model)</code>	<code>tobit</code>
<code>e(title)</code>	title in estimation output
<code>e(link)</code>	<code>identity</code>
<code>e(family)</code>	<code>gaussian</code>
<code>e(clustvar)</code>	name of cluster variable
<code>e(offset)</code>	<code>offset</code>
<code>e(intmethod)</code>	integration method
<code>e(n_quad)</code>	number of integration points
<code>e(chi2type)</code>	Wald; type of model $\chi^2$

<code>e(vce)</code>	<code>vcetype</code> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. Err.
<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(datasignature)</code>	the checksum
<code>e(datasignaturevars)</code>	variables used in calculation of checksum
<code>e(properties)</code>	b V
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(marginswtype)</code>	weight type for <code>margins</code>
<code>e(marginswexp)</code>	weight expression for <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 (up to 20 iterations)
<code>e(gradient)</code>	gradient vector
<code>e(N_g)</code>	group counts
<code>e(g_min)</code>	group-size minimums
<code>e(g_avg)</code>	group-size averages
<code>e(g_max)</code>	group-size maximums
<code>e(V)</code>	variance-covariance matrix of the estimators
<code>e(V_modelbased)</code>	model-based variance

**Functions**

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

## Methods and formulas

Without a loss of generality, consider a two-level regression model

$$E(\mathbf{y}_j | \mathbf{X}_j, \mathbf{u}_j) = \mathbf{X}_j \boldsymbol{\beta} + \mathbf{Z}_j \mathbf{u}_j \quad \mathbf{y} \sim \text{normal}$$

for  $j = 1, \dots, M$  clusters, with the  $j$ th cluster consisting of  $n_j$  observations, where, for the  $j$ th cluster,  $\mathbf{y}_j$  is the  $n_j \times 1$  censored response vector,  $\mathbf{X}_j$  is the  $n_j \times p$  matrix of fixed predictors,  $\mathbf{Z}_j$  is the  $n_j \times q$  matrix of random predictors,  $\mathbf{u}_j$  is the  $q \times 1$  vector of random effects, and  $\boldsymbol{\beta}$  is the  $p \times 1$  vector of regression coefficients on the fixed predictors. The random effects,  $\mathbf{u}_j$ , are assumed to be multivariate normal with mean  $\mathbf{0}$  and variance  $\boldsymbol{\Sigma}$ .

Let  $\boldsymbol{\eta}_j$  be the linear predictor,  $\boldsymbol{\eta}_j = \mathbf{X}_j \boldsymbol{\beta} + \mathbf{Z}_j \mathbf{u}_j$ , that also includes the offset variable when `offset()` is specified.  $y_{ij}$  and  $\eta_{ij}$  are the  $i$ th individual elements of  $\mathbf{y}_j$  and  $\boldsymbol{\eta}_j$ ,  $i = 1, \dots, n_j$ .  $a_{ij}$  refers to the lower limit for observation  $ij$ , and  $b_{ij}$  refers to the upper limit for observation  $ij$ . The conditional density function for the response at observation  $ij$  is then

$$f(y_{ij}^* | \eta_{ij}) = \begin{cases} (\sqrt{2\pi}\sigma_\epsilon)^{-1} \exp^{-(y_{ij} - \eta_{ij})^2 / (2\sigma_\epsilon^2)} & \text{if } y_{ij} = y_{ij}^* \\ \Phi\left(\frac{a_{ij} - \eta_{ij}}{\sigma_\epsilon}\right) & \text{if } y_{ij} \leq y_{ij}^* \\ 1 - \Phi\left(\frac{b_{ij} - \eta_{ij}}{\sigma_\epsilon}\right) & \text{if } y_{ij} \geq y_{ij}^* \end{cases}$$

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

Because the observations are assumed to be conditionally independent, the conditional log density function for cluster  $j$  is

$$\log f(\mathbf{y}_j^* | \boldsymbol{\eta}_j) = \sum_{j=1}^{n_i} \log f(y_{ij}^* | \eta_{ij})$$

and the likelihood function for cluster  $j$  is given by

$$\begin{aligned} \mathcal{L}_j(\boldsymbol{\beta}, \boldsymbol{\Sigma}) &= (2\pi)^{-q/2} |\boldsymbol{\Sigma}|^{-1/2} \int_{\mathfrak{R}^q} f(\mathbf{y}_j^* | \boldsymbol{\eta}_j) \exp\left(-\frac{1}{2} \mathbf{u}_j' \boldsymbol{\Sigma}^{-1} \mathbf{u}_j\right) d\mathbf{u}_j \\ &= (2\pi)^{-q/2} |\boldsymbol{\Sigma}|^{-1/2} \int_{\mathfrak{R}^q} \exp\left\{\log f(\mathbf{y}_j^* | \boldsymbol{\eta}_j) - \frac{1}{2} \mathbf{u}_j' \boldsymbol{\Sigma}^{-1} \mathbf{u}_j\right\} d\mathbf{u}_j \end{aligned} \quad (1)$$

where  $\mathfrak{R}$  denotes the set of values on the real line and  $\mathfrak{R}^q$  is the analog in  $q$ -dimensional space.

The integration in (2) has no closed form and thus must be approximated; see *Methods and formulas* in [ME] **meglm** for details.

`metobit` supports multilevel weights and survey data; see *Methods and formulas* in [ME] **meglm** for details.

## Also see

[ME] **metobit postestimation** — Postestimation tools for metobit

[ME] **meintreg** — Multilevel mixed-effects interval regression

[ME] **me** — Introduction to multilevel mixed-effects models

[BAYES] **bayes: metobit** — Bayesian multilevel tobit regression

[R] **tobit** — Tobit regression

[SEM] **intro 5** — Tour of models (*Multilevel mixed-effects models*)

[SVY] **svy estimation** — Estimation commands for survey data

[XT] **xttobit** — Random-effects tobit models

[U] **20 Estimation and postestimation commands**