Title

stata.com

metobit — Multilevel mixed-effects tobit regression

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 in a crossed-effects model

```
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.

fe_options	Description
Model	
`noconstant`	suppress constant term from the fixed-effects equation
`offset(`*varname*`)`	include *varname* in model with coefficient constrained to 1

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

options	Description
Model	
`ll`[`(`*varname*`|`*#*`)`]	left-censoring variable or limit
`ul`[`(`*varname*`|`*#*`)`]	right-censoring variable or limit
`constraints(`*constraints*`)`	apply specified linear constraints
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `robust`, or `cluster` *clustvar*
Reporting	
`level(`*#*`)`	set confidence level; default is `level(95)`
`nocnsreport`	do not display constraints
`notable`	suppress coefficient table
`noheader`	suppress output header
`nogroup`	suppress table summarizing groups
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Integration	
`intmethod(`*intmethod*`)`	integration method
`intpoints(`*#*`)`	set the number of integration (quadrature) points for all levels; default is `intpoints(7)`
Maximization	
maximize_options	control the maximization process; seldom used
`startvalues(`*svmethod*`)`	method for obtaining starting values
`startgrid`[`(`*gridspec*`)`]	perform a grid search to improve starting values
`noestimate`	do not fit the model; show starting values instead
`dnumerical`	use numerical derivative techniques
`collinear`	keep collinear variables
`coeflegend`	display legend instead of statistics

vartype	Description
`independent`	one unique variance parameter per random effect and all covariances 0; the default unless the `R.` notation is used
`exchangeable`	equal variances for random effects and one common pairwise covariance
`identity`	equal variances for random effects and all covariances 0; the default if the `R.` notation is used
`unstructured`	all variances and covariances to be distinctly estimated
`fixed(`*matname*`)`	user-selected variances and covariances constrained to specified values; the remaining variances and covariances unrestricted
`pattern(`*matname*`)`	user-selected variances and covariances constrained to be equal; the remaining variances and covariances unrestricted

intmethod	Description
mvaghermite	mean–variance adaptive Gauss–Hermite quadrature; the default unless a crossed random-effects model is fit
mcaghermite	mode-curvature adaptive Gauss–Hermite quadrature
ghermite	nonadaptive Gauss–Hermite quadrature
laplace	Laplacian approximation; the default for crossed random-effects models

indepvars and *varlist* 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`, `collect`, 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`, `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` 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 lower and upper limits for censoring, respectively. Observations with *depvar* $\leq$ `ll()` are left-censored; observations with *depvar* $\geq$ `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*`)` 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, jth 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*`)`; 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**.

collinear, coeflegend; see [R] **Estimation options**.

Remarks and examples

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 ith observation within the jth 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, $\mathbf{y}$—is modeled using the following linear prediction:

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

$\mathbf{X}$ 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 24 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 https://www.stata-press.com/data/r17/nlswork
(National Longitudinal Survey of Young Women, 14-24 years old 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                  Number of obs     =     19,224
                                                  Uncensored      =     13,188
Limits: Lower = -inf                              Left-censored   =          0
        Upper = 1.96                              Right-censored  =      6,036

Group variable: idcode                          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
Log likelihood = -6818.5512                     Prob > chi2       =     0.0000
```

ln_wage	Coefficient	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 0.099 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

e(N)	number of observations
e(N_unc)	number of uncensored observations
e(N_lc)	number of left-censored observations
e(N_rc)	number of right-censored observations
e(k)	number of parameters
e(k_dv)	number of dependent variables
e(k_eq)	number of equations in e(b)
e(k_eq_model)	number of equations in overall model test
e(k_f)	number of fixed-effects parameters
e(k_r)	number of random-effects parameters
e(k_rs)	number of variances
e(k_rc)	number of covariances
e(df_m)	model degrees of freedom
e(ll)	log likelihood
e(N_clust)	number of clusters
e(chi2)	χ^2
e(p)	p-value for model test
e(ll_c)	log likelihood, comparison model
e(chi2_c)	χ^2, comparison test
e(df_c)	degrees of freedom, comparison test
e(p_c)	p-value for comparison test
e(rank)	rank of e(V)
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros

e(cmd)	meglm
e(cmd2)	metobit
e(cmdline)	command as typed
e(depvar)	names of dependent variables
e(llopt)	minimum of *depvar* or contents of ll()
e(ulopt)	maximum of *depvar* or contents of ul()
e(wtype)	weight type
e(wexp)	weight expression (first-level weights)
e(fweight*k*)	fweight variable for *k*th highest level, if specified
e(iweight*k*)	iweight variable for *k*th highest level, if specified
e(pweight*k*)	pweight variable for *k*th highest level, if specified
e(covariates)	list of covariates
e(ivars)	grouping variables
e(model)	tobit
e(title)	title in estimation output
e(link)	identity
e(family)	gaussian
e(clustvar)	name of cluster variable
e(offset)	offset
e(intmethod)	integration method
e(n_quad)	number of integration points
e(chi2type)	Wald; type of model χ^2

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

Functions

e(sample)	marks estimation sample

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

Matrices

r(table)	matrix containing the coefficients with their standard errors, test statistics, p-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

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

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

for $j = 1,\ldots,M$ clusters, with the jth cluster consisting of n_j observations, where, for the jth 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 η_{ij} are the ith individual elements of $\mathbf{y}_j$ and $\boldsymbol{\eta}_j$, $i = 1,\ldots,n_j$. 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

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

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

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

where $\Re$ denotes the set of values on the real line and $\Re^q$ 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