mecloglog — Multilevel mixed-effects complementary log–log regression

Description

mecloglog fits mixed-effects models for binary or binomial responses. The conditional distribution of the response given the random effects is assumed to be Bernoulli, with probability of success determined by the inverse complementary log–log function.

Quick start

Two-level complementary log–log model of y on x with random intercepts by lev2

    mecloglog y x || lev2:

Add binary variable a and random coefficients for a

    mecloglog y x a || lev2: a

Same as above, but allow the random effects to be correlated

    mecloglog y x a || lev2: a, covariance(unstructured)

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

    mecloglog y x || lev3: || lev2:

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

    mecloglog y x || _all:R.a || b:

Menu

Statistics > Multilevel mixed-effects models > Complementary log–log regression

Syntax

> mecloglog *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
asis	retain perfect predictor variables

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	
<u>bin</u>omial(*varname* | *#*)	set binomial trials if data are in binomial form
<u>con</u>straints(*constraints*)	apply specified linear constraints
SE/Robust	
vce(*vcetype*)	*vcetype* may be oim, opg, <u>r</u>obust, or <u>cl</u>uster *clustvar*
Reporting	
<u>l</u>evel(*#*)	set confidence level; default is level(95)
eform	report exponentiated coefficients
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	
<u>intm</u>ethod(*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
<u>ind</u>ependent	one unique variance parameter per random effect and all covariances 0; the default unless the R. notation is used
<u>exc</u>hangeable	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
<u>fix</u>ed(*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
<u>mva</u>ghermite	mean–variance adaptive Gauss–Hermite quadrature; the default unless a crossed random-effects model is fit
<u>mca</u>ghermite	mode-curvature adaptive Gauss–Hermite quadrature
<u>gh</u>ermite	nonadaptive Gauss–Hermite quadrature
<u>lap</u>lace	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: mecloglog**.

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.

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

asis forces retention of perfect predictor variables and their associated, perfectly predicted observations and may produce instabilities in maximization; see [R] **probit**.

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.

`binomial(`*varname* | #`)` specifies that the data are in binomial form; that is, *depvar* records the number of successes from a series of binomial trials. This number of trials is given either as *varname*, which allows this number to vary over the observations, or as the constant #. If `binomial()` is not specified (the default), *depvar* is treated as Bernoulli, with any nonzero, nonmissing values indicating positive responses.

`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`, `opg`), 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(`#`)`; see [R] **Estimation options**.

`eform` reports exponentiated coefficients and corresponding standard errors and confidence intervals. This option may be specified either at estimation or upon replay.

`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 mecloglog 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 mecloglog 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

Mixed-effects complementary log–log (cloglog) regression is cloglog 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.

mecloglog allows for many levels of random effects. However, for simplicity, we here consider the two-level model, where for a series of M independent clusters, and conditional on a set of fixed effects $\mathbf{x}_{ij}$ and a set of random effects $\mathbf{u}_j$,

$$\Pr(y_{ij} = 1|\mathbf{x}_{ij}, \mathbf{u}_j) = H(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j) \tag{1}$$

for $j = 1, \ldots, M$ clusters, with cluster j consisting of $i = 1, \ldots, n_j$ observations. The responses are the binary-valued y_{ij}, and we follow the standard Stata convention of treating $y_{ij} = 1$ if $depvar_{ij} \neq 0$ and treating $y_{ij} = 0$ otherwise. The $1 \times p$ row vector $\mathbf{x}_{ij}$ are the covariates for the fixed effects, analogous to the covariates you would find in a standard cloglog regression model, with regression coefficients (fixed effects) $\boldsymbol{\beta}$. For notational convenience here and throughout this manual entry, we suppress the dependence of y_{ij} on $\mathbf{x}_{ij}$.

The $1 \times q$ vector $\mathbf{z}_{ij}$ 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-intercept model, $\mathbf{z}_{ij}$ is simply the scalar 1. The random effects $\mathbf{u}_j$ are M 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 $\mathbf{z}_{ij} = \mathbf{x}_{ij}$, so that all covariate effects are essentially random and distributed as multivariate normal with mean $\boldsymbol{\beta}$ and variance $\boldsymbol{\Sigma}$.

Finally, because this is cloglog regression, $H(\cdot)$ is the inverse of the complementary log–log function that maps the linear predictor to the probability of a success ($y_{ij} = 1$) with $H(v) = 1 - \exp\{-\exp(v)\}$.

Model (1) may also be stated in terms of a latent linear response, where only $y_{ij} = I(y_{ij}^* > 0)$ is observed for the latent

$$y_{ij}^* = \mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j + \epsilon_{ij}$$

The errors ϵ_{ij} are independent and identically extreme-value (Gumbel) distributed with the mean equal to Euler's constant and variance $\sigma_\epsilon^2 = \pi^2/6$, independently of $\mathbf{u}_j$. This nonsymmetric error distribution is an alternative to the symmetric error distribution underlying logistic and probit analysis and is usually used when the positive (or negative) outcome is rare.

Below we present two short examples of mixed-effects cloglog regression; refer to [ME] **me** and [ME] **meglm** for examples of other random-effects models. A two-level cloglog model can also be fit using xtcloglog with the re option; see [XT] **xtcloglog**. In the absence of random effects, mixed-effects cloglog regression reduces to standard cloglog regression; see [R] **cloglog**.

▷ Example 1: Two-level random-intercept model

In example 1 of [XT] **xtcloglog**, we analyze unionization of women in the United States over the period 1970–1988. The women are identified by the variable idcode. Here we refit that model with mecloglog. Because the original example used 12 integration points by default, we request 12 integration points as well.

```
. use https://www.stata-press.com/data/r19/union
(NLS Women 14-24 in 1968)
```

```
. mecloglog union age grade not_smsa south##c.year || idcode:, intpoints(12)

Fitting fixed-effects model:

Iteration 0:   Log likelihood = -14237.139
Iteration 1:   Log likelihood = -13546.159
Iteration 2:   Log likelihood = -13540.611
Iteration 3:   Log likelihood = -13540.607
Iteration 4:   Log likelihood = -13540.607

Refining starting values:

Grid node 0:   Log likelihood = -11104.448

Fitting full model:

Iteration 0:   Log likelihood = -11104.448
Iteration 1:   Log likelihood = -10617.891
Iteration 2:   Log likelihood = -10537.919
Iteration 3:   Log likelihood = -10535.946
Iteration 4:   Log likelihood = -10535.941
Iteration 5:   Log likelihood = -10535.941
```

```
Mixed-effects cloglog regression              Number of obs    =      26,200
Group variable: idcode                        Number of groups =       4,434

                                              Obs per group:
                                                            min =           1
                                                            avg =         5.9
                                                            max =          12

Integration method: mvaghermite               Integration pts. =          12

                                              Wald chi2(6)     =      248.12
Log likelihood = -10535.941                   Prob > chi2      =      0.0000
```

union	Coefficient	Std. err.	z	P>|z|	[95% conf. interval]	
age	.0128542	.0119441	1.08	0.282	-.0105559	.0362642
grade	.0699965	.0138551	5.05	0.000	.0428409	.097152
not_smsa	-.1982009	.0649258	-3.05	0.002	-.3254531	-.0709488
1.south	-2.049901	.4892644	-4.19	0.000	-3.008842	-1.090961
year	-.0006158	.0123999	-0.05	0.960	-.0249191	.0236875
south#c.year						
1	.0164457	.0060685	2.71	0.007	.0045516	.0283399
_cons	-3.277375	.6610552	-4.96	0.000	-4.57302	-1.981731
idcode						
var(_cons)	3.489803	.1630921			3.184351	3.824555

```
LR test vs. cloglog model: chibar2(01) = 6009.33      Prob >= chibar2 = 0.0000
```

The estimates are practically the same. `xtcloglog` reports the estimated variance component as a standard deviation, $\hat{\sigma}_{\mathbf{u}} = 1.86$. `mecloglog` reports $\hat{\sigma}_{\mathbf{u}}^2 = 3.49$, the square root of which is 1.87. We find that age and education each have a positive effect on union membership, although the former is not statistically significant. Women who live outside of metropolitan areas are less likely to unionize.

The estimated variance of the random intercept at the individual level, $\hat{\sigma}^2$, is 3.49 with standard error 0.16. The reported likelihood-ratio test shows that there is enough variability between women to favor a mixed-effects cloglog regression over an ordinary cloglog regression; see *Distribution theory for likelihood-ratio test* in [ME] **me** for a discussion of likelihood-ratio testing of variance components.

◁

> ## Example 2: Three-level random-intercept model

Rabe-Hesketh, Toulopoulou, and Murray (2001) analyzed data from a study that measured the cognitive ability of patients with schizophrenia compared with their relatives and control subjects. Cognitive ability was measured as the successful completion of the "Tower of London", a computerized task, measured at three levels of difficulty. For all but one of the 226 subjects, there were three measurements (one for each difficulty level). Because patients' relatives were also tested, a family identifier, family, was also recorded.

We fit a cloglog model with response dtlm, the indicator of cognitive function, and with covariates difficulty and a set of indicator variables for group, with the controls (group==1) being the base category. We also allow for random effects due to families and due to subjects within families. The first is a random intercept (constant only) at the family level, and the second is a random intercept at the subject level. The order in which these are specified (from left to right) is significant—mecloglog assumes that subject is nested within family. The equations are separated by ||.

```
. use https://www.stata-press.com/data/r19/towerlondon
(Tower of London data)

. mecloglog dtlm difficulty i.group || family: || subject:

Fitting fixed-effects model:

Iteration 0:   Log likelihood = -337.21921
Iteration 1:   Log likelihood = -313.79023
Iteration 2:   Log likelihood = -313.56906
Iteration 3:   Log likelihood = -313.56888
Iteration 4:   Log likelihood = -313.56888

Refining starting values:

Grid node 0:   Log likelihood = -314.57061

Fitting full model:

Iteration 0:   Log likelihood = -314.57061   (not concave)
Iteration 1:   Log likelihood = -308.82101
Iteration 2:   Log likelihood = -305.71841
Iteration 3:   Log likelihood = -305.26804
Iteration 4:   Log likelihood = -305.26516
Iteration 5:   Log likelihood = -305.26516
```

Mixed-effects cloglog regression   Number of obs   =   677

Grouping information

Group variable	No. of groups	Observations per group Minimum	Average	Maximum
family	118	2	5.7	27
subject	226	2	3.0	3

Integration method: mvaghermite   Integration pts. =   7

Log likelihood = -305.26516

Wald chi2(3)   =   83.32
Prob > chi2   =   0.0000

dtlm	Coefficient	Std. err.	z	P>|z|	[95% conf. interval]	
difficulty	-1.342844	.1501508	-8.94	0.000	-1.637135	-1.048554
group						
2	-.1331007	.269389	-0.49	0.621	-.6610935	.3948922
3	-.7714314	.3097099	-2.49	0.013	-1.378452	-.164411
_cons	-1.6718	.2290325	-7.30	0.000	-2.120695	-1.222905
family						
var(_cons)	.2353453	.2924064			.0206122	2.687117
family> subject						
var(_cons)	.7737687	.4260653			.2629714	2.276742

LR test vs. cloglog model: chi2(2) = 16.61   Prob > chi2 = 0.0002
Note: LR test is conservative and provided only for reference.

After adjusting for the random-effects structure, the probability of successful completion of the Tower of London decreases dramatically as the level of difficulty increases. Also, schizophrenics (group==3) tended not to perform as well as the control subjects.

◁

The above extends to models with more than two levels of nesting by adding more random-effects equations, each separated by ||.

Stored results

mecloglog stores the following in e():

Scalars

e(N)	number of 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)	mecloglog
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(wtype)	weight type
e(wexp)	weight expression (first-level weights)
e(fweight$k$)	fweight variable for kth highest level, if specified
e(iweight$k$)	iweight variable for kth highest level, if specified
e(pweight$k$)	pweight variable for kth highest level, if specified
e(covariates)	list of covariates
e(ivars)	grouping variables
e(model)	cloglog
e(title)	title in estimation output
e(link)	cloglog
e(family)	bernoulli or binomial
e(clustvar)	name of cluster variable
e(offset)	offset
e(binomial)	binomial number of trials
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

mecloglog is a convenience command for meglm with a cloglog link and a bernoulli or binomial family; see [ME] **meglm**.

Model (1) assumes Bernoulli data, a special case of the binomial. Because binomial data are also supported by mecloglog (option binomial()), the methods presented below are in terms of the more general binomial mixed-effects model.

For a two-level binomial model, consider the response y_{ij} as the number of successes from a series of r_{ij} Bernoulli trials (replications). For cluster j, $j = 1, \ldots, M$, the conditional distribution of $\mathbf{y}_j = (y_{j1}, \ldots, y_{jn_j})'$, given a set of cluster-level random effects $\mathbf{u}_j$, is

$$f(\mathbf{y}_j|\mathbf{u}_j) = \prod_{i=1}^{n_j} \left[\binom{r_{ij}}{y_{ij}} \left\{ H(\boldsymbol{\eta}_{ij}) \right\}^{y_{ij}} \left\{ 1 - H(\boldsymbol{\eta}_{ij}) \right\}^{r_{ij}-y_{ij}} \right]$$

$$= \exp\left(\sum_{i=1}^{n_j} \left[y_{ij} \log\left\{ H(\boldsymbol{\eta}_{ij}) \right\} - (r_{ij} - y_{ij}) \exp(\boldsymbol{\eta}_{ij}) + \log\binom{r_{ij}}{y_{ij}} \right] \right)$$

for $\boldsymbol{\eta}_{ij} = \mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j + \text{offset}_{ij}$ and $H(v) = 1 - \exp\{-\exp(v)\}$.

Defining $\mathbf{r}_j = (r_{j1}, \ldots, r_{jn_j})'$ and

$$
c\left(\mathbf{y}_j, \mathbf{r}_j\right) = \sum_{i=1}^{n_j} \log \begin{pmatrix} r_{ij} \\ y_{ij} \end{pmatrix}
$$

where $c(\mathbf{y}_j, \mathbf{r}_j)$ does not depend on the model parameters, we can express the above compactly in matrix notation,

$$
f(\mathbf{y}_j | \mathbf{u}_j) = \exp\left[\mathbf{y}_j' \log \left\{ H(\boldsymbol{\eta}_j) \right\} - (\mathbf{r}_j - \mathbf{y}_j)' \exp(\boldsymbol{\eta}_j) + c\left(\mathbf{y}_j, \mathbf{r}_j\right) \right]
$$

where $\boldsymbol{\eta}_j$ is formed by stacking the row vectors $\boldsymbol{\eta}_{ij}$. We extend the definitions of the functions $H(\cdot)$, $\log(\cdot)$, and $\exp(\cdot)$ to be vector functions where necessary.

Because the prior distribution of $\mathbf{u}_j$ is multivariate normal with mean $\mathbf{0}$ and $q \times q$ variance matrix $\boldsymbol{\Sigma}$, the likelihood contribution for the jth cluster is obtained by integrating $\mathbf{u}_j$ out of the joint density $f(\mathbf{y}_j, \mathbf{u}_j)$,

$$
\begin{aligned}
\mathcal{L}_j(\boldsymbol{\beta}, \boldsymbol{\Sigma}) &= (2\pi)^{-q/2} \left| \boldsymbol{\Sigma} \right|^{-1/2} \int f(\mathbf{y}_j | \mathbf{u}_j) \exp\left(-\mathbf{u}_j' \boldsymbol{\Sigma}^{-1} \mathbf{u}_j / 2 \right) d\mathbf{u}_j \\
&= \exp\left\{ c\left(\mathbf{y}_j, \mathbf{r}_j\right) \right\} (2\pi)^{-q/2} \left| \boldsymbol{\Sigma} \right|^{-1/2} \int \exp\left\{ h\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right) \right\} d\mathbf{u}_j
\end{aligned}
\tag{2}
$$

where

$$
h\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right) = \mathbf{y}_j' \log \left\{ H(\boldsymbol{\eta}_j) \right\} - (\mathbf{r}_j - \mathbf{y}_j)' \exp(\boldsymbol{\eta}_j) - \mathbf{u}_j' \boldsymbol{\Sigma}^{-1} \mathbf{u}_j / 2
$$

and for convenience, in the arguments of $h(\cdot)$ we suppress the dependence on the observable data $(\mathbf{y}_j, \mathbf{r}_j, \mathbf{X}_j, \mathbf{Z}_j)$.

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

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

Reference

Rabe-Hesketh, S., T. Toulopoulou, and R. M. Murray. 2001. Multilevel modeling of cognitive function in schizophrenic patients and their first degree relatives. *Multivariate Behavioral Research* 36: 279–298. https://doi.org/10.1207/S15327906MBR3602_07.

Also see

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

[ME] **melogit** — Multilevel mixed-effects logistic regression

[ME] **meprobit** — Multilevel mixed-effects probit regression

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

[BAYES] **bayes: mecloglog** — Bayesian multilevel complementary log–log regression

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

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

[XT] **xtcloglog** — Random-effects and population-averaged cloglog models

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

For suggested citations, see the FAQ on citing Stata documentation.