Title

> **melogit** — Multilevel mixed-effects logistic regression

Syntax

> melogit *depvar fe_equation* [|| *re_equation*] [|| *re_equation* ...] [, *options*]

where the syntax of *fe_equation* is

> [*indepvars*] [*if*] [*in*] [, *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.

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

options	Description
[Model]	
<u>bin</u>omial(*varname* │ *#*)	set binomial trials if data are in binomial form
constraints(*constraints*)	apply specified linear constraints
collinear	keep collinear variables
[SE/Robust]	
vce(*vcetype*)	*vcetype* may be oim, <u>r</u>obust, or <u>cl</u>uster *clustvar*
[Reporting]	
<u>l</u>evel(*#*)	set confidence level; default is level(95)
or	report fixed-effects coefficients as odds ratios
<u>nocns</u>report	do not display constraints
<u>notable</u>	suppress coefficient table
<u>nohe</u>ader	suppress output header
<u>nogr</u>oup	suppress table summarizing groups
nolrtest	do not perform likelihood-ratio test comparing with logistic regression
display_options	control 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
<u>intp</u>oints(*#*)	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
<u>coefl</u>egend	display legend instead of statistics

vartype	Description
independent	one unique variance parameter per random effect, 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, 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
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 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**.

by is allowed; see [U] **11.1.10 Prefix commands**.

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.

Menu

Statistics > Multilevel mixed-effects models > Logistic regression

Description

melogit fits mixed-effects models for binary and binomial responses. The conditional distribution of the response given the random effects is assumed to be Bernoulli, with success probability determined by the logistic cumulative distribution function.

melogit performs optimization using the original metric of variance components. When variance components are near the boundary of the parameter space, you may consider using the meqrlogit command, which provides alternative parameterizations of variance components; see [ME] **meqrlogit**.

Options

> [Model]

noconstant suppresses the constant (intercept) term and may be specified for the fixed-effects equation and for any or all of 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*$[k, l]$.

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*), 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(#); see [R] **estimation options**.

or reports estimated fixed-effects coefficients transformed to odds ratios, that is, $\exp(\beta)$ rather than β. Standard errors and confidence intervals are similarly transformed. This option affects how results are displayed, not how they are estimated. or 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.

nolrtest prevents melogit from performing a likelihood-ratio test that compares the mixed-effects logistic model with standard (marginal) logistic regression. This option may also be specified upon replay to suppress this test from the output.

display_options: noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvla-bel, 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 and variance adaptive Gauss–Hermite quadrature; mcaghermite
performs mode and 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 melogit 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 melogit 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

For a general introduction to me commands, see [ME] **me**.

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

Remarks are presented under the following headings:

Introduction

Mixed-effects logistic regression is logistic 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.

Comprehensive treatments of mixed models are provided by, for example, Searle, Casella, and Mc-Culloch (1992); Verbeke and Molenberghs (2000); Raudenbush and Bryk (2002); Demidenko (2004); Hedeker and Gibbons (2006); McCulloch, Searle, and Neuhaus (2008); and Rabe-Hesketh and Skrondal (2012). Guo and Zhao (2000) and Rabe-Hesketh and Skrondal (2012, chap. 10) are good introductory readings on applied multilevel modeling of binary data.

`melogit` allows for not just one, but many levels of nested clusters of random effects. For example, in a three-level model you can specify random effects for schools and then random effects for classes nested within schools. In this model, the observations (presumably, the students) comprise the first level, the classes comprise the second level, and the schools comprise the third level.

However, for simplicity, for now we consider the two-level model, where for a series of M independent clusters, and conditional on 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 logistic 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 logistic regression, $H(\cdot)$ is the logistic cumulative distribution function, which maps the linear predictor to the probability of a success ($y_{ij} = 1$) with $H(v) = \exp(v)/\{1+\exp(v)\}$.

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 distributed as logistic with mean 0 and variance $\pi^2/3$ and are independent of $\mathbf{u}_j$.

Model (1) is an example of a generalized linear mixed model (GLMM), which generalizes the linear mixed-effects (LME) model to non-Gaussian responses. You can fit LMEs in Stata by using `mixed` and fit GLMMs by using `meglm`. Because of the relationship between LMEs and GLMMs, there is insight to be gained through examination of the linear mixed model. This is especially true for Stata users because the terminology, syntax, options, and output for fitting these types of models are nearly identical. See [ME] **mixed** and the references therein, particularly in *Introduction*, for more information.

Log-likelihood calculations for fitting any generalized mixed-effects model require integrating out the random effects. One widely used modern method is to directly estimate the integral required to

calculate the log likelihood by Gauss–Hermite quadrature or some variation thereof. Because the log likelihood is computed, this method has the advantage of permitting likelihood-ratio tests for comparing nested models. Also, if done correctly, quadrature approximations can be quite accurate, thus minimizing bias.

melogit supports three types of Gauss–Hermite quadrature and the Laplacian approximation method; see *Methods and formulas* of [ME] **meglm** for details. The simplest random-effects model you can fit using melogit is the two-level model with a random intercept,

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

This model can also be fit using xtlogit with the re option; see [XT] **xtlogit**.

Below we present two short examples of mixed-effects logit regression; refer to [ME] **me** and [ME] **meglm** for additional examples including crossed random-effects models.

Two-level models

We begin with a simple application of (1) as a two-level model, because a one-level model, in our terminology, is just standard logistic regression; see [R] **logistic**.

▷ Example 1

Ng et al. (2006) analyzed a subsample of data from the 1989 Bangladesh fertility survey (Huq and Cleland 1990), which polled 1,934 Bangladeshi women on their use of contraception.

```
. use http://www.stata-press.com/data/r13/bangladesh
(Bangladesh Fertility Survey, 1989)

. describe

Contains data from http://www.stata-press.com/data/r13/bangladesh.dta
  obs:         1,934                          Bangladesh Fertility Survey,
                                                1989
 vars:             7                          28 May 2013 20:27
 size:        19,340                          (_dta has notes)
```

variable name	storage type	display format	value label	variable label
district	byte	%9.0g		District
c_use	byte	%9.0g	yesno	Use contraception
urban	byte	%9.0g	urban	Urban or rural
age	float	%6.2f		Age, mean centered
child1	byte	%9.0g		1 child
child2	byte	%9.0g		2 children
child3	byte	%9.0g		3 or more children

```
Sorted by:  district
```

The women sampled were from 60 districts, identified by the variable district. Each district contained either urban or rural areas (variable urban) or both. The variable c_use is the binary response, with a value of 1 indicating contraceptive use. Other covariates include mean-centered age and three indicator variables recording number of children. Below we fit a standard logistic regression model amended to have random effects for each district.

```
. melogit c_use urban age child* || district:

Fitting fixed-effects model:

Iteration 0:   log likelihood = -1229.5485
Iteration 1:   log likelihood = -1228.5268
Iteration 2:   log likelihood = -1228.5263
Iteration 3:   log likelihood = -1228.5263

Refining starting values:

Grid node 0:   log likelihood = -1219.2681

Fitting full model:

Iteration 0:   log likelihood = -1219.2681  (not concave)
Iteration 1:   log likelihood = -1207.5978
Iteration 2:   log likelihood = -1206.8428
Iteration 3:   log likelihood = -1206.8322
Iteration 4:   log likelihood = -1206.8322
```

```
Mixed-effects logistic regression              Number of obs      =       1934
Group variable:         district               Number of groups   =         60

                                               Obs per group: min =          2
                                                              avg =       32.2
                                                              max =        118

Integration method: mvaghermite                Integration points =          7

                                               Wald chi2(5)       =     109.60
Log likelihood = -1206.8322                    Prob > chi2        =     0.0000
```

c_use	Coef.	Std. Err.	z	P>|z|	[95% Conf. Interval]	
urban	.7322765	.1194857	6.13	0.000	.4980888	.9664641
age	-.0264981	.0078916	-3.36	0.001	-.0419654	-.0110309
child1	1.116001	.1580921	7.06	0.000	.8061465	1.425856
child2	1.365895	.1746691	7.82	0.000	1.02355	1.70824
child3	1.344031	.1796549	7.48	0.000	.9919139	1.696148
_cons	-1.68929	.1477591	-11.43	0.000	-1.978892	-1.399687

district						
var(_cons)	.215618	.0733222			.1107208	.4198954

```
LR test vs. logistic regression: chibar2(01) =    43.39 Prob>=chibar2 = 0.0000
```

The estimation table reports the fixed effects and the estimated variance components. The fixed effects can be interpreted just as you would the output from logit. You can also specify the or option at estimation or on replay to display the fixed effects as odds ratios instead. If you did display results as odds ratios, you would find urban women to have roughly double the odds of using contraception as that of their rural counterparts. Having any number of children will increase the odds from three- to fourfold when compared with the base category of no children. Contraceptive use also decreases with age.

Underneath the fixed effect, the table shows the estimated variance components. The random-effects equation is labeled district, meaning that these are random effects at the district level. Because we have only one random effect at this level, the table shows only one variance component. The estimate of σ_u^2 is 0.22 with standard error 0.07.

A likelihood-ratio test comparing the model to ordinary logistic regression is provided and is highly significant for these data.

◁

Three-level models

Two-level models extend naturally to models with three or more levels with nested random effects. By "nested", we mean that the random effects shared within lower-level subgroups are unique to the upper-level groups. For example, assuming that classroom effects would be nested within schools would be natural, because classrooms are unique to schools.

▷ Example 2

Rabe-Hesketh, Toulopoulou, and Murray (2001) analyzed data from a study measuring 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.

```
. use http://www.stata-press.com/data/r13/towerlondon
(Tower of London data)

. describe

Contains data from http://www.stata-press.com/data/r13/towerlondon.dta
  obs:           677                          Tower of London data
 vars:             5                          31 May 2013 10:41
 size:         4,739                          (_dta has notes)

              storage   display    value
variable name   type    format     label      variable label

family          int     %8.0g                 Family ID
subject         int     %9.0g                 Subject ID
dtlm            byte    %9.0g                 1 = task completed
difficulty      byte    %9.0g                 Level of difficulty: -1, 0, or 1
group           byte    %8.0g                 1: controls; 2: relatives; 3:
                                                schizophrenics

Sorted by:  family  subject
```

We fit a logistic 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 allow for random effects due to families and due to subjects within families. We also request to display odds ratios for the fixed-effects parameters.

```
. melogit dtlm difficulty i.group || family: || subject: , or

Fitting fixed-effects model:

Iteration 0:   log likelihood = -317.35042
Iteration 1:   log likelihood = -313.90007
Iteration 2:   log likelihood = -313.89079
Iteration 3:   log likelihood = -313.89079

Refining starting values:

Grid node 0:   log likelihood = -310.28429

Fitting full model:

Iteration 0:   log likelihood = -310.28429
Iteration 1:   log likelihood = -307.36653
Iteration 2:   log likelihood = -305.19357
Iteration 3:   log likelihood = -305.12073
Iteration 4:   log likelihood = -305.12041
Iteration 5:   log likelihood = -305.12041
```

Mixed-effects logistic regression   Number of obs   =   677

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 points =   7

Wald chi2(3)   =   74.90
Log likelihood = -305.12041   Prob > chi2   =   0.0000

dtlm	Odds Ratio	Std. Err.	z	P>|z|	[95% Conf. Interval]	
difficulty	.1923372	.037161	-8.53	0.000	.1317057	.2808806
group						
2	.7798263	.2763763	-0.70	0.483	.3893369	1.561961
3	.3491318	.13965	-2.63	0.009	.15941	.764651
_cons	.226307	.0644625	-5.22	0.000	.1294902	.3955112
family						
var(_cons)	.5692105	.5215654			.0944757	3.429459
family> subject						
var(_cons)	1.137917	.6854853			.3494165	3.705762

LR test vs. logistic regression: chi2(2) =   17.54   Prob > chi2 = 0.0002
Note: LR test is conservative and provided only for reference.

Notes:

1. This is a three-level model with two random-effects equations, separated by ||. 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—melogit assumes that subject is nested within family.

2. The information on groups is now displayed as a table, with one row for each upper level. Among other things, we see that we have 226 subjects from 118 families. You can suppress this table with the `nogroup` or the `noheader` option, which will suppress the rest of the header as well.

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. Of course, we would make similar conclusions from a standard logistic model fit to the same data, but the odds ratios would differ somewhat.

◁

The above extends to models with more than two levels of nesting in the obvious manner, by adding more random-effects equations, each separated by `||`. The order of nesting goes from left to right as the groups go from biggest (highest level) to smallest (lowest level).

Stored results

`melogit` 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)	significance
e(ll_c)	log likelihood, comparison model
e(chi2_c)	χ^2, comparison model
e(df_c)	degrees of freedom, comparison model
e(p_c)	significance, comparison model
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)	melogit
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(covariates)	list of covariates
e(ivars)	grouping variables
e(model)	logistic
e(title)	title in estimation output
e(link)	logit
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

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 estimator
e(V_modelbased)	model-based variance

Functions
e(sample)	marks estimation sample

Methods and formulas

Model (1) assumes Bernoulli data, a special case of the binomial. Because binomial data are also supported by melogit (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

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

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

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

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

$$c(\mathbf{y}_j, \mathbf{r}_j) = \sum_{i=1}^{n_j} \log \binom{r_{ij}}{y_{ij}}$$

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

$$f(\mathbf{y}_j | \mathbf{u}_j) = \exp\left[\mathbf{y}_j'\boldsymbol{\eta}_j - \mathbf{r}_j' \log\left\{\mathbf{1} + \exp(\boldsymbol{\eta}_j)\right\} + c(\mathbf{y}_j, \mathbf{r}_j)\right]$$

where $\boldsymbol{\eta}_j$ is formed by stacking the row vectors $\boldsymbol{\eta}_{ij}$. We extend the definitions of the functions $\log(\cdot)$ 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 $f(\mathbf{y}_j, \mathbf{u}_j)$,

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

where

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

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. `melogit` offers four approximation methods: mean–variance adaptive Gauss–Hermite quadrature (default unless a crossed random-effects model is fit), mode-curvature adaptive Gauss–Hermite quadrature, nonadaptive Gauss–Hermite quadrature, and Laplacian approximation (default for crossed random-effects models).

The Laplacian approximation is based on a second-order Taylor expansion of $h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j)$ about the value of $\mathbf{u}_j$ that maximizes it; see *Methods and formulas* in [ME] **meglm** for details.

Gaussian quadrature relies on transforming the multivariate integral in (2) into a set of nested univariate integrals. Each univariate integral can then be evaluated using a form of Gaussian quadrature; see *Methods and formulas* in [ME] **meglm** for details.

The log likelihood for the entire dataset is simply the sum of the contributions of the M individual clusters, namely, $\mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = \sum_{j=1}^{M} \mathcal{L}_j(\boldsymbol{\beta}, \boldsymbol{\Sigma})$.

Maximization of $\mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\Sigma})$ is performed with respect to $(\boldsymbol{\beta}, \boldsymbol{\sigma}^2)$, where $\boldsymbol{\sigma}^2$ is a vector comprising the unique elements of $\boldsymbol{\Sigma}$. Parameter estimates are stored in `e(b)` as $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\sigma}}^2)$, with the corresponding variance–covariance matrix stored in `e(V)`.

References

Andrews, M. J., T. Schank, and R. Upward. 2006. Practical fixed-effects estimation methods for the three-way error-components model. *Stata Journal* 6: 461–481.

Demidenko, E. 2004. *Mixed Models: Theory and Applications*. Hoboken, NJ: Wiley.

Guo, G., and H. Zhao. 2000. Multilevel modeling of binary data. *Annual Review of Sociology* 26: 441–462.

Gutierrez, R. G., S. L. Carter, and D. M. Drukker. 2001. sg160: On boundary-value likelihood-ratio tests. *Stata Technical Bulletin* 60: 15–18. Reprinted in *Stata Technical Bulletin Reprints*, vol. 10, pp. 269–273. College Station, TX: Stata Press.

Harbord, R. M., and P. Whiting. 2009. metandi: Meta-analysis of diagnostic accuracy using hierarchical logistic regression. *Stata Journal* 9: 211–229.

Hedeker, D., and R. D. Gibbons. 2006. *Longitudinal Data Analysis*. Hoboken, NJ: Wiley.

Huq, N. M., and J. Cleland. 1990. *Bangladesh Fertility Survey 1989 (Main Report)*. National Institute of Population Research and Training.

Joe, H. 2008. Accuracy of Laplace approximation for discrete response mixed models. *Computational Statistics & Data Analysis* 52: 5066–5074.

Laird, N. M., and J. H. Ware. 1982. Random-effects models for longitudinal data. *Biometrics* 38: 963–974.

Lin, X., and N. E. Breslow. 1996. Bias correction in generalized linear mixed models with multiple components of dispersion. *Journal of the American Statistical Association* 91: 1007–1016.

Marchenko, Y. V. 2006. Estimating variance components in Stata. *Stata Journal* 6: 1–21.

McCulloch, C. E., S. R. Searle, and J. M. Neuhaus. 2008. *Generalized, Linear, and Mixed Models*. 2nd ed. Hoboken, NJ: Wiley.

McLachlan, G. J., and K. E. Basford. 1988. *Mixture Models*. New York: Dekker.

Ng, E. S.-W., J. R. Carpenter, H. Goldstein, and J. Rasbash. 2006. Estimation in generalised linear mixed models with binary outcomes by simulated maximum likelihood. *Statistical Modelling* 6: 23–42.

Rabe-Hesketh, S., and A. Skrondal. 2012. *Multilevel and Longitudinal Modeling Using Stata*. 3rd ed. College Station, TX: Stata Press.

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.

Raudenbush, S. W., and A. S. Bryk. 2002. *Hierarchical Linear Models: Applications and Data Analysis Methods*. 2nd ed. Thousand Oaks, CA: Sage.

Searle, S. R., G. Casella, and C. E. McCulloch. 1992. *Variance Components*. New York: Wiley.

Self, S. G., and K.-Y. Liang. 1987. Asymptotic properties of maximum likelihood estimators and likelihood ratio tests under nonstandard conditions. *Journal of the American Statistical Association* 82: 605–610.

Verbeke, G., and G. Molenberghs. 2000. *Linear Mixed Models for Longitudinal Data*. New York: Springer.

Also see

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

[ME] **mecloglog** — Multilevel mixed-effects complementary log-log regression

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

[ME] **meqrlogit** — Multilevel mixed-effects logistic regression (QR decomposition)

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

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

[XT] **xtlogit** — Fixed-effects, random-effects, and population-averaged logit models

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