STATA MULTILEVEL MIXED-EFFECTS REFERENCE MANUAL
RELEASE 13

A Stata Press Publication
StataCorp LP
College Station, Texas

Published by Stata Press, 4905 Lakeway Drive, College Station, Texas 77845
Typeset in TeX

ISBN-10: 1-59718-119-6
ISBN-13: 978-1-59718-119-8

The suggested citation for this software is

StataCorp. 2013. *Stata: Release 13*. Statistical Software. College Station, TX: StataCorp LP.

Contents

Cross-referencing the documentation

When reading this manual, you will find references to other Stata manuals. For example,

[U] **26 Overview of Stata estimation commands**
[R] **regress**
[D] **reshape**

The first example is a reference to chapter 26, *Overview of Stata estimation commands*, in the *User's Guide*; the second is a reference to the `regress` entry in the *Base Reference Manual*; and the third is a reference to the `reshape` entry in the *Data Management Reference Manual*.

All the manuals in the Stata Documentation have a shorthand notation:

[GSM]	*Getting Started with Stata for Mac*
[GSU]	*Getting Started with Stata for Unix*
[GSW]	*Getting Started with Stata for Windows*
[U]	*Stata User's Guide*
[R]	*Stata Base Reference Manual*
[D]	*Stata Data Management Reference Manual*
[G]	*Stata Graphics Reference Manual*
[XT]	*Stata Longitudinal-Data/Panel-Data Reference Manual*
[ME]	*Stata Multilevel Mixed-Effects Reference Manual*
[MI]	*Stata Multiple-Imputation Reference Manual*
[MV]	*Stata Multivariate Statistics Reference Manual*
[PSS]	*Stata Power and Sample-Size Reference Manual*
[P]	*Stata Programming Reference Manual*
[SEM]	*Stata Structural Equation Modeling Reference Manual*
[SVY]	*Stata Survey Data Reference Manual*
[ST]	*Stata Survival Analysis and Epidemiological Tables Reference Manual*
[TS]	*Stata Time-Series Reference Manual*
[TE]	*Stata Treatment-Effects Reference Manual: Potential Outcomes/Counterfactual Outcomes*
[I]	*Stata Glossary and Index*
[M]	*Mata Reference Manual*

Title

me — Introduction to multilevel mixed-effects models

Syntax by example Formal syntax Description Remarks and examples
Acknowledgments References Also see

Syntax by example

Linear mixed-effects models

Linear model of y on x with random intercepts by id

```
mixed y x || id:
```

Three-level linear model of y on x with random intercepts by doctor and patient

```
mixed y x || doctor: || patient:
```

Linear model of y on x with random intercepts and coefficients on x by id

```
mixed y x || id: x
```

Same model with covariance between the random slope and intercept

```
mixed y x || id: x, covariance(unstructured)
```

Linear model of y on x with crossed random effects for id and week

```
mixed y x || _all: R.id || _all: R.week
```

Same model specified to be more computationally efficient

```
mixed y x || _all: R.id || week:
```

Full factorial repeated-measures ANOVA of y on a and b with random effects by field

```
mixed y a##b || field:
```

Generalized linear mixed-effects models

Logistic model of y on x with random intercepts by id, reporting odds ratios

```
melogit y x || id: , or
```

Same model specified as a GLM

```
meglm y x || id:, family(bernoulli) link(logit)
```

Three-level ordered probit model of y on x with random intercepts by doctor and patient

```
meoprobit y x || doctor: || patient:
```

Formal syntax

Linear mixed-effects models

`mixed` *depvar* *fe_equation* [|| *re_equation*] [|| *re_equation* ...] [, *options*]

where the syntax of the fixed-effects equation, *fe_equation*, is

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

and the syntax of a random-effects equation, *re_equation*, is the same as below for a generalized linear mixed-effects model.

Generalized linear mixed-effects models

`me`*cmd* *depvar* *fe_equation* [|| *re_equation*] [|| *re_equation* ...] [, *options*]

where the syntax of the fixed-effects equation, *fe_equation*, is

[*indepvars*] [*if*] [*in*] [, *fe_options*]

and the syntax of a random-effects equation, *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.

Description

Mixed-effects models are characterized as containing both fixed effects and random effects. The fixed effects are analogous to standard regression coefficients and are estimated directly. The random effects are not directly estimated (although they may be obtained postestimation) but are summarized according to their estimated variances and covariances. Random effects may take the form of either random intercepts or random coefficients, and the grouping structure of the data may consist of multiple levels of nested groups. As such, mixed-effects models are also known in the literature as multilevel models and hierarchical models. Mixed-effects commands fit mixed-effects models for a variety of distributions of the response conditional on normally distributed random effects.

Mixed-effects linear regression

mixed	Multilevel mixed-effects linear regression

Mixed-effects generalized linear model

meglm	Multilevel mixed-effects generalized linear model

Mixed-effects binary regression

melogit	Multilevel mixed-effects logistic regression
meqrlogit	Multilevel mixed-effects logistic regression (QR decomposition)
meprobit	Multilevel mixed-effects probit regression
mecloglog	Multilevel mixed-effects complementary log-log regression

Mixed-effects ordinal regression

meologit	Multilevel mixed-effects ordered logistic regression
meoprobit	Multilevel mixed-effects ordered probit regression

Mixed-effects count-data regression

mepoisson	Multilevel mixed-effects Poisson regression
meqrpoisson	Multilevel mixed-effects Poisson regression (QR decomposition)
menbreg	Multilevel mixed-effects negative binomial regression

Mixed-effects multinomial regression

Although there is no memlogit command, multilevel mixed-effects multinomial logistic models can be fit using gsem; see [SEM] **example 41g**.

Remarks and examples

Remarks are presented under the following headings:

Introduction

Multilevel models have been used extensively in diverse fields, from the health and social sciences to econometrics. Mixed-effects models for binary outcomes have been used, for example, to analyze the effectiveness of toenail infection treatments (Lesaffre and Spiessens 2001) and to model union membership of young males (Vella and Verbeek 1998). Ordered outcomes have been studied by, for example, Tutz and Hennevogl (1996), who analyzed data on wine bitterness, and De Boeck and Wilson (2004), who studied verbal aggressiveness. For applications of mixed-effects models for count responses, see, for example, the study on police stops in New York City (Gelman and Hill 2007) and the analysis of the number of patents (Hall, Griliches, and Hausman 1986). Rabe-Hesketh and Skrondal (2012) provide more examples of linear and generalized linear mixed-effects models.

For a comprehensive treatment of mixed-effects models, see, for example, Searle, Casella, and McCulloch (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).

Using mixed-effects commands

Below we summarize general capabilities of the mixed-effects commands. We let *mecmd* stand for any mixed-effects command, such as `mixed`, `melogit`, or `meprobit`.

1. Fit a two-level random-intercept model with *levelvar* defining the second level:

   . *mecmd depvar* [*indepvars*] ... || *levelvar*:, ...

2. Fit a two-level random-coefficients model containing the random-effects covariates *revars* at the level *levelvar*:

   . *mecmd depvar* [*indepvars*] ... || *levelvar*: *revars*, ...

   This model assumes an independent covariance structure between the random effects; that is, all covariances are assumed to be 0. There is no statistical justification, however, for imposing any particular covariance structure between random effects at the onset of the analysis. In practice, models with an unstructured random-effects covariance matrix, which allows for distinct variances and covariances between all random-effects covariates (*revars*) at the same level, must be explored first; see *Other covariance structures* and example 3 in [ME] **meqrlogit** for details.

   Stata's commands use the default independent covariance structure for computational feasibility. Numerical methods for fitting mixed-effects models are computationally intensive—computation time increases significantly as the number of parameters increases; see *Computation time and the Laplacian approximation* for details. The unstructured covariance is the most general and contains many parameters, which may result in an unreasonable computation time even for relatively simple random-effects models. Whenever feasible, however, you should start your statistical analysis by fitting mixed-effects models with an unstructured covariance between random effects, as we show next.

3. Specify the unstructured covariance between the random effects in the above:

   . *mecmd depvar* [*indepvars*] ... || *levelvar*: *revars*, `covariance(unstructured)` ...

4. Fit a three-level nested model with *levelvar1* defining the third level and *levelvar2* defining the second level:

   . *mecmd depvar* [*indepvars*] ... || *levelvar1*: || *levelvar2*:, ...

5. Fit the above three-level nested model as a two-level model with exchangeable covariance structure at the second level (`mixed`, `meqrlogit`, and `meqrpoisson` only):

```
. mecmd depvar [indepvars] ... || levelvar1: R.levelvar2, cov(exchangeable) ...
```

See example 11 in [ME] **mixed** for details about this equivalent specification. This specification may be useful for a more efficient fitting of random-effects models with a mixture of crossed and nested effects.

6. Fit higher-level nested models:

```
. mecmd depvar [indepvars] ... || levelvar1: || levelvar2: || levelvar3: || ...
```

7. Fit a two-way crossed-effects model with the `_all:` notation for each random-effects equation:

```
. mecmd depvar [indepvars] ... || _all: R.factor1 || _all: R.factor2 ...
```

When you use the `_all:` notation for each random-effects equation, the total dimension of the random-effects design equals $r_1 + r_2$, where r_1 and r_2 are the numbers of levels in *factor1* and *factor2*, respectively. This specification may be infeasible for some mixed-effects models; see item 8 below for a more efficient specification of this model.

8. Fit a two-way crossed-effects model with the `_all:` notation for the first random-effects equation only:

```
. mecmd depvar [indepvars] ... || _all: R.factor1 || factor2:, ...
```

Compared with the specification in item 7, this specification requires only $r_1 + 1$ parameters and is thus more efficient; see *Crossed-effects models* for details.

9. Fit a two-way full-factorial random-effects model:

```
. mecmd depvar [indepvars] ... || _all: R.factor1 || factor2: || factor1: ...
```

10. Fit a two-level mixed-effects model with a blocked-diagonal covariance structure between *revars1* and *revars2*:

```
. mecmd depvar [indepvars] ... || levelvar: revars1, noconstant ///
                               || levelvar: revars2, noconstant ...
```

11. Fit a linear mixed-effects model where the correlation between the residual errors follows an autoregressive process of order 1:

```
. mixed depvar [indepvars] ... || levelvar:, residuals(ar 1, t(time)) ...
```

More residual error structures are available; see [ME] **mixed** for details.

12. Fit a two-level linear mixed-effects model accounting for sampling weights *expr1* at the first (residual) level and for sampling weights *expr2* at the level of *levelvar*:

```
. mixed depvar [indepvars] [pweight=expr1] ... || levelvar:, pweight(expr2) ...
```

Mixed-effects commands—with the exception of `mixed`, `meqrlogit`, and `meqrpoisson`—allow constraints on both fixed-effects and random-effects parameters. We provide several examples below of imposing constraints on variance components.

13. Fit a mixed-effects model with the variance of the random intercept on *levelvar* constrained to be 16:

```
. constraint 1 _b[var(_cons[levelvar]):_cons]=16
. mecmd depvar [indepvars] ... || levelvar:, constraints(1) ...
```

14. Fit a mixed-effects model with the variance of the random intercept on *levelvar* and the variance of the random slope on *revar* to be equal:

```
. constraint 1 _b[var(revar[levelvar]):_cons] = _b[var(_cons[levelvar]):_cons]
. mecmd depvar [indepvars] ... || levelvar: revar, constraints(1) ...
```

    Note that the constraints above are equivalent to imposing an identity covariance structure for the random-effects equation:

```
. mecmd depvar [indepvars] ... || levelvar: revar, cov(identity) ...
```

15. Assuming four random slopes *revars*, fit a mixed-effects model with the variance components at the level of *levelvar* constrained to have a banded structure:

```
. mat p = (1,.,.,. \ 2,1,.,. \ 3,2,1,. \ 4,3,2,1)
. mecmd depvar [indepvars] ... || levelvar: revars, noconstant ///
    covariance(pattern(p)) ...
```

16. Assuming four random slopes *revars*, fit a mixed-effects model with the variance components at the level of *levelvar* constrained to the specified numbers, and with all the covariances constrained to be 0:

```
. mat f = diag((1,2,3,4))
. mecmd depvar [indepvars] ... || levelvar: revars, noconstant ///
    covariance(fixed(f)) ...
```

The variance components in models in items 15 and 16 can also be constrained by using the `constraints()` option, but using `covariance(pattern())` or `covariance(fixed())` is more convenient.

Mixed-effects models

Linear mixed-effects models

Mixed-effects models for continuous responses, or linear mixed-effects (LME) models, are a generalization of linear regression allowing for the inclusion of random deviations (effects) other than those associated with the overall error term. In matrix notation,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + \boldsymbol{\epsilon} \tag{1}$$

where $\mathbf{y}$ is the $n \times 1$ vector of responses, $\mathbf{X}$ is an $n \times p$ design/covariate matrix for the fixed effects $\boldsymbol{\beta}$, and $\mathbf{Z}$ is the $n \times q$ design/covariate matrix for the random effects $\mathbf{u}$. The $n \times 1$ vector of errors $\boldsymbol{\epsilon}$ is assumed to be multivariate normal with mean 0 and variance matrix $\sigma^2_\epsilon \mathbf{R}$.

The fixed portion of (1), $\mathbf{X}\boldsymbol{\beta}$, is analogous to the linear predictor from a standard OLS regression model with $\boldsymbol{\beta}$ being the regression coefficients to be estimated. For the random portion of (1), $\mathbf{Z}\mathbf{u} + \boldsymbol{\epsilon}$, we assume that $\mathbf{u}$ has variance–covariance matrix $\mathbf{G}$ and that $\mathbf{u}$ is orthogonal to $\boldsymbol{\epsilon}$ so that

$$\text{Var}\begin{bmatrix}\mathbf{u}\\ \boldsymbol{\epsilon}\end{bmatrix} = \begin{bmatrix}\mathbf{G} & \mathbf{0}\\ \mathbf{0} & \sigma^2_\epsilon \mathbf{R}\end{bmatrix}$$

The random effects $\mathbf{u}$ are not directly estimated (although they may be predicted) but instead are characterized by the elements of $\mathbf{G}$, known as variance components, that are estimated along with the overall residual variance σ^2_ϵ and the residual-variance parameters that are contained within $\mathbf{R}$.

The general forms of the design matrices $\mathbf{X}$ and $\mathbf{Z}$ allow estimation for a broad class of linear models: blocked designs, split-plot designs, growth curves, multilevel or hierarchical designs, etc. They also allow a flexible method of modeling within-cluster correlation. Subjects within the same cluster can be correlated as a result of a shared random intercept, or through a shared random slope on age (for example), or both. The general specification of $\mathbf{G}$ also provides additional flexibility: the random intercept and random slope could themselves be modeled as independent, or correlated, or independent with equal variances, and so forth. The general structure of $\mathbf{R}$ also allows for residual errors to be heteroskedastic and correlated and allows flexibility in exactly how these characteristics can be modeled.

In clustered-data situations, it is convenient not to consider all n observations at once but instead to organize the mixed model as a series of M independent groups (or clusters)

$$\mathbf{y}_j = \mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j + \boldsymbol{\epsilon}_j \tag{2}$$

for $j = 1, \ldots, M$, with cluster j consisting of n_j observations. The response $\mathbf{y}_j$ comprises the rows of $\mathbf{y}$ corresponding with the jth cluster, with $\mathbf{X}_j$ and $\boldsymbol{\epsilon}_j$ defined analogously. The random effects $\mathbf{u}_j$ can now be thought of as M realizations of a $q \times 1$ vector that is normally distributed with mean $\mathbf{0}$ and $q \times q$ variance matrix $\boldsymbol{\Sigma}$. The matrix $\mathbf{Z}_j$ is the $n_j \times q$ design matrix for the jth cluster random effects. Relating this to (1),

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Z}_M \end{bmatrix}; \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_M \end{bmatrix}; \quad \mathbf{G} = \mathbf{I}_M \otimes \boldsymbol{\Sigma}; \quad \mathbf{R} = \mathbf{I}_M \otimes \boldsymbol{\Lambda} \tag{3}$$

where $\boldsymbol{\Lambda}$ denotes the variance matrix of the level-1 residuals and $\otimes$ is the Kronecker product.

The mixed-model formulation (2) is from Laird and Ware (1982) and offers two key advantages. First, it makes specifications of random-effects terms easier. If the clusters are schools, you can simply specify a random effect at the school level, as opposed to thinking of what a school-level random effect would mean when all the data are considered as a whole (if it helps, think Kronecker products). Second, representing a mixed-model with (2) generalizes easily to more than one set of random effects. For example, if classes are nested within schools, then (2) can be generalized to allow random effects at both the school and the class-within-school levels.

In Stata, you can use `mixed` to fit linear mixed-effects models; see [ME] **mixed** for a detailed discussion and examples. Various predictions, statistics, and diagnostic measures are available after fitting an LME model with `mixed`. For the most part, calculation centers around obtaining estimates of random effects; see [ME] **mixed postestimation** for a detailed discussion and examples.

Generalized linear mixed-effects models

Generalized linear mixed-effects (GLME) models, also known as generalized linear mixed models (GLMMs), are extensions of generalized linear models allowing for the inclusion of random deviations (effects). In matrix notation,

$$g\big\{E(\mathbf{y}|\mathbf{X},\mathbf{u})\big\} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u}, \qquad \mathbf{y} \sim F \tag{4}$$

where $\mathbf{y}$ is the $n \times 1$ vector of responses from the distributional family F, $\mathbf{X}$ is an $n \times p$ design/covariate matrix for the fixed effects $\boldsymbol{\beta}$, and $\mathbf{Z}$ is an $n \times q$ design/covariate matrix for the random effects $\mathbf{u}$. The $\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u}$ part is called the linear predictor and is often denoted as $\boldsymbol{\eta}$. $g(\cdot)$ is called the link function and is assumed to be invertible such that

$$E(\mathbf{y}|\mathbf{u}) = g^{-1}(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u}) = H(\boldsymbol{\eta}) = \boldsymbol{\mu}$$

For notational convenience here and throughout this manual entry, we suppress the dependence of $\mathbf{y}$ on $\mathbf{X}$. Substituting various definitions for $g(\cdot)$ and F results in a wide array of models. For instance, if $g(\cdot)$ is the logit function and $\mathbf{y}$ is distributed as Bernoulli, we have

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

or mixed-effects logistic regression. If $g(\cdot)$ is the natural log function and $\mathbf{y}$ is distributed as Poisson, we have

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

or mixed-effects Poisson regression.

In Stata, you can use `meglm` to fit mixed-effects models for nonlinear responses. Some combinations of families and links are so common that we implemented them as separate commands in terms of `meglm`.

Command	`meglm` equivalent
`melogit`	`family(bernoulli) link(logit)`
`meprobit`	`family(bernoulli) link(probit)`
`mecloglog`	`family(bernoulli) link(cloglog)`
`meologit`	`family(ordinal) link(logit)`
`meoprobit`	`family(ordinal) link(probit)`
`mepoisson`	`family(poisson) link(log)`
`menbreg`	`family(nbinomial) link(log)`

When no family–link combination is specified, `meglm` defaults to a Gaussian family with an identity link. Thus `meglm` can be used to fit linear mixed-effects models; however, for those models we recommend using the more specialized `mixed`, which, in addition to `meglm` capabilities, accepts frequency and sampling weights and allows for modeling of the structure of the residual errors; see [ME] **mixed** for details.

Various predictions, statistics, and diagnostic measures are available after fitting a GLME model with `meglm` and other `me` commands. For the most part, calculation centers around obtaining estimates of random effects; see [ME] **meglm postestimation** for a detailed discussion and examples.

For the random portion of (4), $\mathbf{Z}\mathbf{u}$, we assume that $\mathbf{u}$ has variance–covariance matrix $\mathbf{G}$ such that

$$\text{Var}(\mathbf{u}) = \mathbf{G}$$

The random effects $\mathbf{u}$ are not directly estimated (although they may be predicted) but instead are characterized by the elements of $\mathbf{G}$, known as variance components.

Analogously to (2), in clustered-data situations, we can write

$$E(\mathbf{y}_j|\mathbf{u}_j) = g^{-1}(\mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j), \qquad \mathbf{y}_j \sim F \tag{5}$$

with all the elements defined as before. In terms of the whole dataset, we now have

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Z}_M \end{bmatrix}; \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_M \end{bmatrix}; \quad \mathbf{G} = \mathbf{I}_M \otimes \boldsymbol{\Sigma} \tag{6}$$

Finally, we state our convention on counting and ordering model levels. Models (2) and (5) are what we call two-level models, with extensions to three, four, or any number of levels. The observation y_{ij} is for individual i within cluster j, and the individuals comprise the first level while the clusters comprise the second level of the model. In our hypothetical three-level model with classes nested within schools, the observations within classes (the students, presumably) would constitute the first level, the classes would constitute the second level, and the schools would constitute the third level. This differs from certain citations in the classical ANOVA literature and texts such as Pinheiro and Bates (2000) but is the standard in the vast literature on hierarchical models, for example, Skrondal and Rabe-Hesketh (2004).

Alternative mixed-effects model specification

In this section, we present a hierarchical or multistage formulation of mixed-effects models where each level is described by its own set of equations.

Consider a random-intercept model that we write here in general terms:

$$y_{ij} = \beta_0 + \beta_1 x_{ij} + u_j + \epsilon_{ij} \tag{7}$$

This single-equation specification contains both level-1 and level-2 effects. In the hierarchical form, we specify a separate equation for each level.

$$\begin{aligned} y_{ij} &= \gamma_{0j} + \beta_1 x_{ij} + \epsilon_{ij} \\ \gamma_{0j} &= \beta_{00} + u_{0j} \end{aligned} \tag{8}$$

The equation for the intercept γ_{0j} consists of the overall mean intercept β_{00} and a cluster-specific random intercept u_{0j}. To fit this model in Stata, we must translate the multiple-equation notation into a single-equation form. We substitute the second equation into the first one and rearrange terms.

$$\begin{aligned} y_{ij} &= \beta_{00} + u_{0j} + \beta_1 x_{ij} + \epsilon_{ij} \\ &= \beta_{00} + \beta_1 x_{ij} + u_{0j} + \epsilon_{ij} \end{aligned} \tag{9}$$

Note that model (9) is the same as model (7) with $\beta_{00} \equiv \beta_0$ and $u_{0j} \equiv u_j$. Thus the Stata syntax for our generic random-intercept model is

```
. mixed y x || id:
```

where `id` is the variable designating the clusters.

We can extend model (8) to include a random slope. We do so by specifying an additional equation for the slope on x_{ij}.

$$\begin{aligned} y_{ij} &= \gamma_{0j} + \gamma_{1j} x_{ij} + \epsilon_{ij} \\ \gamma_{0j} &= \beta_{00} + u_{0j} \\ \gamma_{1j} &= \beta_{10} + u_{1j} \end{aligned} \tag{10}$$

The additional equation for the slope γ_{1j} consists of the overall mean slope β_{10} and a cluster-specific random slope u_{1j}. We substitute the last two equations into the first one to obtain a reduced-form model.

$$
\begin{aligned}
y_{ij} &= (\beta_{00} + u_{0j}) + (\beta_{10} + u_{1j})x_{ij} + \epsilon_{ij} \\
&= \beta_{00} + \beta_{10}x_{ij} + u_{0j} + u_{1j}x_{ij} + \epsilon_{ij}
\end{aligned}
$$

The Stata syntax for this model becomes

```
. mixed y x || id: x, covariance(unstructured)
```

where we specified an unstructured covariance structure for the level-2 u terms.

Here we further extend the random-slope random-intercept model (10) by adding a level-2 covariate z_j into the level-2 equations.

$$
\begin{aligned}
y_{ij} &= \gamma_{0j} + \gamma_{1j}x_{ij} + \epsilon_{ij} \\
\gamma_{0j} &= \beta_{00} + \beta_{01}z_j + u_{0j} \\
\gamma_{1j} &= \beta_{10} + \beta_{11}z_j + u_{1j}
\end{aligned}
$$

We substitute as before to obtain a single-equation form:

$$
\begin{aligned}
y_{ij} &= (\beta_{00} + \beta_{01}z_j + u_{0j}) + (\beta_{10} + \beta_{11}z_j + u_{1j})x_{ij} + \epsilon_{ij} \\
&= \beta_{00} + \beta_{01}z_j + \beta_{10}x_{ij} + \beta_{11}z_jx_{ij} + u_{0j} + u_{1j}x_{ij} + \epsilon_{ij}
\end{aligned}
$$

Now the fixed-effects portion of the equation contains a constant and variables x, z, and their interaction. Assuming both x and z are continuous variables, we can use the following Stata syntax to fit this model:

```
. mixed y x z c.x#c.z  || id: x, covariance(unstructured)
```

We refer you to Raudenbush and Bryk (2002) and Rabe-Hesketh and Skrondal (2012) for a more thorough discussion and further examples of multistage mixed-model formulations, including three-level models.

Likelihood calculation

The key to fitting mixed models lies in estimating the variance components, and for that there exist many methods. Most of the early literature in LME models dealt with estimating variance components in ANOVA models. For simple models with balanced data, estimating variance components amounts to solving a system of equations obtained by setting expected mean-squares expressions equal to their observed counterparts. Much of the work in extending the ANOVA method to unbalanced data for general ANOVA designs is attributed to Henderson (1953).

The ANOVA method, however, has its shortcomings. Among these is a lack of uniqueness in that alternative, unbiased estimates of variance components could be derived using other quadratic forms of the data in place of observed mean squares (Searle, Casella, and McCulloch 1992, 38–39). As a result, ANOVA methods gave way to more modern methods, such as minimum norm quadratic unbiased estimation (MINQUE) and minimum variance quadratic unbiased estimation (MIVQUE); see Rao (1973) for MINQUE and LaMotte (1973) for MIVQUE. Both methods involve finding optimal quadratic forms of the data that are unbiased for the variance components.

Stata uses maximum likelihood (ML) to fit LME and GLME models. The ML estimates are based on the usual application of likelihood theory, given the distributional assumptions of the model. In addition, for linear mixed-effects models, `mixed` offers the method of restricted maximum likelihood (REML). The basic idea behind REML (Thompson 1962) is that you can form a set of linear contrasts of the response that do not depend on the fixed effects $\boldsymbol{\beta}$ but instead depend only on the variance components to be estimated. You then apply ML methods by using the distribution of the linear contrasts to form the likelihood; see the *Methods and formulas* section of [ME] **mixed** for a detailed discussion of ML and REML methods in the context of linear mixed-effects models.

Log-likelihood calculations for fitting any LME or GLME model require integrating out the random effects. For LME models, this integral has a closed-form solution; for GLME models, it does not. In dealing with this difficulty, early estimation methods avoided the integration altogether. Two such popular methods are the closely related penalized quasi-likelihood (PQL) and marginal quasi-likelihood (MQL) (Breslow and Clayton 1993). Both PQL and MQL use a combination of iterative reweighted least squares (see [R] **glm**) and standard estimation techniques for fitting LME models. Efficient computational methods for fitting LME models have existed for some time (Bates and Pinheiro 1998; Littell et al. 2006), and PQL and MQL inherit this computational efficiency. However, both of these methods suffer from two key disadvantages. First, they have been shown to be biased, and this bias can be severe when clusters are small or intracluster correlation is high (Rodríguez and Goldman 1995; Lin and Breslow 1996). Second, because they are "quasi-likelihood" methods and not true likelihood methods, their use prohibits comparing nested models via likelihood-ratio (LR) tests, blocking the main avenue of inference involving variance components.

The advent of modern computers has brought with it the development of more computationally intensive methods, such as bias-corrected PQL (Lin and Breslow 1996), Bayesian Markov-Chain Monte Carlo, and simulated maximum likelihood, just to name a few; see Ng et al. (2006) for a discussion of these alternate strategies (and more) for mixed-effects models for binary outcomes.

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 itself is estimated, this method has the advantage of permitting LR tests for comparing nested models. Also, if done correctly, quadrature approximations can be quite accurate, thus minimizing bias. `meglm` and the other `me` commands support three types of Gauss–Hermite quadratures: mean–variance adaptive Gauss–Hermite quadrature (MVAGH), mode-curvature adaptive Gauss–Hermite quadrature (MCAGH), and nonadaptive Gauss–Hermite quadrature (GHQ); see *Methods and formulas* of [ME] **meglm** for a detailed discussion of these quadrature methods. A fourth method, the Laplacian approximation, that does not involve numerical integration is also offered; see *Computation time and the Laplacian approximation* below and *Methods and formulas* of [ME] **meglm** for a detailed discussion of the Laplacian approximation method.

Computation time and the Laplacian approximation

Like many programs that fit generalized linear mixed models, `me` commands can be computationally intensive. This is particularly true for large datasets with many lowest-level clusters, models with many random coefficients, models with many estimable parameters (both fixed effects and variance components), or any combination thereof.

Computation time will also depend on hardware and other external factors but in general is (roughly) a function of $p^2\{M + M(N_Q)^{q_t}\}$, where p is the number of estimable parameters, M is the number of lowest-level (smallest) clusters, N_Q is the number of quadrature points, and q_t is the total dimension of the random effects, that is, the total number of random intercepts and coefficients at all levels.

For a given model and a given dataset, the only prevailing factor influencing computation time is $(N_Q)^{q_t}$. However, because this is a power function, this factor can get prohibitively large. For example, using five quadrature points for a model with one random intercept and three random coefficients, we get $(N_Q)^{q_t} = 5^4 = 625$. Even a modest increase to seven quadrature points would increase this factor by almost fourfold ($7^4 = 2{,}401$), which, depending on M and p, could drastically slow down estimation. When fitting mixed-effects models, you should always assess whether the approximation is adequate by refitting the model with a larger number of quadrature points. If the results are essentially the same, the lower number of quadrature points can be used.

However, we do not deny a tradeoff between speed and accuracy, and in that spirit we give you the option to choose a (possibly) less accurate solution in the interest of getting quicker results. Toward this end is the limiting case of $N_Q = 1$, otherwise known as the Laplacian approximation; see *Methods and formulas* of [ME] **meglm**. The computational benefit is evident—1 raised to any power equals 1—and the Laplacian approximation has been shown to perform well in certain situations (Liu and Pierce 1994; Tierney and Kadane 1986). When using Laplacian approximation, keep the following in mind:

1. Fixed-effects parameters and their standard errors are well approximated by the Laplacian method. Therefore, if your interest lies primarily here, then the Laplacian approximation may be a viable alternative.

2. Estimates of variance components exhibit bias, particularly the variances.

3. The model log likelihood and comparison LR test are in fair agreement with statistics obtained via quadrature methods.

Although this is by no means the rule, we find the above observations to be fairly typical based on our own experience. Pinheiro and Chao (2006) also make observations similar to points 1 and 2 on the basis of their simulation studies: bias due to Laplace (when present) tends to exhibit itself more in the estimated variance components than in the estimates of the fixed effects as well as at the lower levels in higher-level models.

Item 3 is of particular interest, because it demonstrates that the Laplacian approximation can produce a decent estimate of the model log likelihood. Consequently, you can use the Laplacian approximation during the model building phase of your analysis, during which you are comparing competing models by using LR tests. Once you settle on a parsimonious model that fits well, you can then increase the number of quadrature points and obtain more accurate parameter estimates for further study.

Of course, sometimes the Laplacian approximation will perform either better or worse than observed here. This behavior depends primarily on cluster size and intracluster correlation, but the relative influence of these factors is unclear. The idea behind the Laplacian approximation is to approximate the posterior density of the random effects given the response with a normal distribution; see *Methods and formulas* of [ME] **meglm**. Asymptotic theory dictates that this approximation improves with larger clusters. Of course, the key question, as always, is "How large is large enough?" Also, there are data situations where the Laplacian approximation performs well even with small clusters. Therefore, it is difficult to make a definitive call as to when you can expect the Laplacian approximation to yield accurate results across all aspects of the model.

In conclusion, consider our above advice as a rule of thumb based on empirical evidence.

Diagnosing convergence problems

Given the flexibility of mixed-effects models, you will find that some models fail to converge when used with your data. The default gradient-based method used by mixed-effects commands is the Newton–Raphson algorithm, requiring the calculation of a gradient vector and Hessian (second-derivative) matrix; see [R] **ml**.

A failure to converge can take any one of three forms:

1. repeated nonconcave or backed-up iterations without convergence;
2. a Hessian (second-derivative) calculation that has become asymmetric, unstable, or has missing values; or
3. the message "standard-error calculation has failed" when computing standard errors.

All three situations essentially amount to the same thing: the Hessian calculation has become unstable, most likely because of a ridge in the likelihood function, a subsurface of the likelihood in which all points give the same value of the likelihood and for which there is no unique solution.

Such behavior is usually the result of one of the following two situations:

A. A model that is not identified given the data, for example, fitting the three-level nested random intercept model

$$y_{jk} = \mathbf{x}_{jk}\boldsymbol{\beta} + u_k^{(3)} + u_{jk}^{(2)} + \epsilon_{jk}$$

without any replicated measurements at the (j,k) level, that is, with only one i per (j,k) combination. This model is unidentified for such data because the random intercepts $u_{jk}^{(2)}$ are confounded with the overall errors ϵ_{jk}.

B. A model that contains a variance component whose estimate is really close to 0. When this occurs, a ridge is formed by an interval of values near 0, which produce the same likelihood and look equally good to the optimizer.

For LME models, one useful way to diagnose problems of nonconvergence is to rely on the expectation-maximization (EM) algorithm (Dempster, Laird, and Rubin 1977), normally used by `mixed` only as a means of refining starting values; see *Diagnosing convergence problems* of [ME] **mixed** for details.

If your data and model are nearly unidentified, as opposed to fully unidentified, you may be able to obtain convergence with standard errors by changing some of the settings of the gradient-based optimization. Adding the `difficult` option can be particularly helpful if you are seeing many "nonconcave" messages; you may also consider changing the `technique()` or using the `nonrtolerance` option; see [R] **maximize**.

Regardless of how the convergence problem revealed itself, you may try to obtain better starting values; see *Obtaining better starting values* in [ME] **meglm** for details.

Distribution theory for likelihood-ratio test

When determining the asymptotic distribution of an LR test comparing two nested mixed-effects models, issues concerning boundary problems imposed by estimating strictly positive quantities (that is, variances) can complicate the situation. For example, when performing LR tests involving linear mixed-effects models (whether comparing with linear regression within `mixed` or comparing two separate linear mixed-effects models with `lrtest`), you may thus sometimes see a test labeled as `chibar` rather than the usual `chi2`, or you may see a `chi2` test with a note attached stating that the test is conservative or possibly conservative depending on the hypothesis being tested.

At the heart of the issue is the number of variances being restricted to 0 in the reduced model. If there are none, the usual asymptotic theory holds, and the distribution of the test statistic is χ^2 with degrees of freedom equal to the difference in the number of estimated parameters between both models.

When there is only one variance being set to 0 in the reduced model, the asymptotic distribution of the LR test statistic is a 50:50 mixture of a χ^2_p and a χ^2_{p+1} distribution, where p is the number of other restricted parameters in the reduced model that are unaffected by boundary conditions. Stata labels such test statistics as `chibar` and adjusts the significance levels accordingly. See Self and Liang (1987) for the appropriate theory or Gutierrez, Carter, and Drukker (2001) for a Stata-specific discussion.

When more than one variance parameter is being set to 0 in the reduced model, however, the situation becomes more complicated. For example, consider a comparison test versus linear regression for a mixed model with two random coefficients and unstructured covariance matrix

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_0^2 & \sigma_{01} \\ \sigma_{01} & \sigma_1^2 \end{bmatrix}$$

Because the random component of the mixed model comprises three parameters $(\sigma_0^2, \sigma_{01}, \sigma_1^2)$, on the surface it would seem that the LR comparison test would be distributed as χ^2_3. However, two complications need to be considered. First, the variances σ_0^2 and σ_1^2 are restricted to be positive, and second, constraints such as $\sigma_1^2 = 0$ implicitly restrict the covariance σ_{01} to be 0 as well. From a technical standpoint, it is unclear how many parameters must be restricted to reduce the model to linear regression.

Because of these complications, appropriate and sufficiently general distribution theory for the more-than-one-variance case has yet to be developed. Theory (for example, Stram and Lee [1994]) and empirical studies (for example, McLachlan and Basford [1988]) have demonstrated that, whatever the distribution of the LR test statistic, its tail probabilities are bounded above by those of the χ^2 distribution with degrees of freedom equal to the full number of restricted parameters (three in the above example).

The `mixed` and `me` commands use this reference distribution, the χ^2 with full degrees of freedom, to produce a conservative test and place a note in the output labeling the test as such. Because the displayed significance level is an upper bound, rejection of the null hypothesis based on the reported level would imply rejection on the basis of the actual level.

Examples

Two-level models

▷ Example 1: Growth-curve model

Consider a longitudinal dataset, used by both Ruppert, Wand, and Carroll (2003) and Diggle et al. (2002), consisting of `weight` measurements of 48 pigs on 9 successive `weeks`. Pigs are identified by the variable `id`. Each pig experiences a linear trend in growth, but overall weight measurements vary from pig to pig. Because we are not really interested in these particular 48 pigs per se, we instead treat them as a random sample from a larger population and model the between-pig variability as a random effect, or in the terminology of (2), as a random-intercept term at the pig level. We thus wish to fit the model

$$\texttt{weight}_{ij} = \beta_0 + \beta_1 \texttt{week}_{ij} + u_j + \epsilon_{ij}$$

for $i = 1, \ldots, 9$ weeks and $j = 1, \ldots, 48$ pigs. The fixed portion of the model, $\beta_0 + \beta_1\texttt{week}_{ij}$, simply states that we want one overall regression line representing the population average. The random effect u_j serves to shift this regression line up or down according to each pig. Because the random effects occur at the pig level (`id`), we fit the model by typing

```
. use http://www.stata-press.com/data/r13/pig
(Longitudinal analysis of pig weights)
. mixed weight week || id:
Performing EM optimization:
Performing gradient-based optimization:
Iteration 0:   log likelihood = -1014.9268
Iteration 1:   log likelihood = -1014.9268
Computing standard errors:
Mixed-effects ML regression                     Number of obs      =       432
Group variable: id                              Number of groups   =        48
                                                Obs per group: min =         9
                                                               avg =       9.0
                                                               max =         9
                                                Wald chi2(1)       =  25337.49
Log likelihood = -1014.9268                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
      weight |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        week |   6.209896   .0390124   159.18   0.000     6.133433    6.286359
       _cons |   19.35561   .5974059    32.40   0.000     18.18472    20.52651
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
id: Identity                 |
                  var(_cons) |   14.81751   3.124226      9.801716    22.40002
-----------------------------+------------------------------------------------
               var(Residual) |   4.383264   .3163348      3.805112     5.04926
------------------------------------------------------------------------------
LR test vs. linear regression: chibar2(01) =   472.65 Prob >= chibar2 = 0.0000
```

We explain the output in detail in example 1 of [ME] **mixed**. Here we only highlight the most important points.

1. The first estimation table reports the fixed effects. We estimate $\beta_0 = 19.36$ and $\beta_1 = 6.21$.
2. The second estimation table shows the estimated variance components. The first section of the table is labeled `id: Identity`, meaning that these are random effects at the `id` (pig) level and that their variance–covariance matrix is a multiple of the identity matrix; that is, $\boldsymbol{\Sigma} = \sigma_u^2 \mathbf{I}$. The estimate of $\widehat{\sigma}_u^2$ is 14.82 with standard error 3.12.
3. The row labeled `var(Residual)` displays the estimated standard deviation of the overall error term; that is, $\widehat{\sigma}_\epsilon^2 = 4.38$. This is the variance of the level-one errors, that is, the residuals.
4. An LR test comparing the model with one-level ordinary linear regression is provided and is highly significant for these data.

We can predict the random intercept u_j and list the predicted random intercept for the first 10 pigs by typing

```
. predict r_int, reffects
. egen byte tag = tag(id)
. list id r_int if id<=10 & tag
```

	id	r_int
1.	1	-1.683105
10.	2	.8987018
19.	3	-1.952043
28.	4	-1.79068
37.	5	-3.189159
46.	6	-3.780823
55.	7	-2.382344
64.	8	-1.952043
73.	9	-6.739143
82.	10	1.16764

In example 3 of [ME] **mixed**, we show how to fit a random-slope model for these data, and in example 1 of [ME] **mixed postestimation**, we show how to plot the estimated regression lines for each of the pigs.

◁

▷ Example 2: Split-plot design

Here we replicate the example of a split-plot design from Kuehl (2000, 477). The researchers investigate the effects of nitrogen in four different chemical forms and the effects of thatch accumulation on the quality of golf turf. The experimental plots were arranged in a randomized complete block design with two replications. After two years of nitrogen treatment, the second treatment factor, years of thatch accumulation, was added to the experiment. Each of the eight experimental plots was split into three subplots. Within each plot, the subplots were randomly assigned to accumulate thatch for a period of 2, 5, and 8 years.

```
. use http://www.stata-press.com/data/r13/clippings, clear
(Turfgrass experiment)

. describe

Contains data from http://www.stata-press.com/data/r13/clippings.dta
  obs:            24                          Turfgrass experiment
 vars:             4                          21 Feb 2013 14:57
 size:           168
--------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
--------------------------------------------------------------------------------
chlorophyll     float   %9.0g                 Chlorophyll content (mg/g) of
                                                grass clippings
thatch          byte    %9.0g                 Years of thatch accumulation
block           byte    %9.0g                 Replication
nitrogen        byte    %17.0g     nitrolab   Nitrogen fertilizer
--------------------------------------------------------------------------------
Sorted by:
```

Nitrogen treatment is stored in the variable `nitrogen`, and the chemicals used are urea, ammonium sulphate, isobutylidene diurea (IBDU), and sulphur-coated urea (urea SC). The length of thatch accumulation is stored in the variable `thatch`. The response is the chlorophyll content of grass clippings, recorded in mg/g and stored in the variable `chlorophyll`. The `block` variable identifies the replication group.

There are two sources of variation in this example corresponding to the whole-plot errors and the subplot errors. The subplot errors are the residual errors. The whole-plot errors represents variation in the chlorophyll content across nitrogen treatments and replications. We create the variable `wpunit` to represent the whole-plot units that correspond to the levels of the nitrogen treatment and block interaction.

```
. egen wpunit = group(nitrogen block)

. mixed chlorophyll ibn.nitrogen##ibn.thatch ibn.block, noomitted noconstant ||
> wpunit:, reml
note: 8.thatch omitted because of collinearity
note: 1.nitrogen#8.thatch omitted because of collinearity
note: 2.nitrogen#8.thatch omitted because of collinearity
note: 3.nitrogen#8.thatch omitted because of collinearity
note: 4.nitrogen#2.thatch omitted because of collinearity
note: 4.nitrogen#5.thatch omitted because of collinearity
note: 4.nitrogen#8.thatch omitted because of collinearity
note: 2.block omitted because of collinearity

Performing EM optimization:

Performing gradient-based optimization:

Iteration 0:   log restricted-likelihood = -13.212401
Iteration 1:   log restricted-likelihood = -13.203149
Iteration 2:   log restricted-likelihood = -13.203125
Iteration 3:   log restricted-likelihood = -13.203125
```

```
Computing standard errors:

Mixed-effects REML regression                   Number of obs      =        24
Group variable: wpunit                          Number of groups   =         8

                                                Obs per group: min =         3
                                                               avg =       3.0
                                                               max =         3

                                                Wald chi2(13)      =   2438.36
Log restricted-likelihood = -13.203125          Prob > chi2        =    0.0000

------------------------------------------------------------------------------
 chlorophyll |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    nitrogen |
       urea  |   5.245833   .3986014    13.16   0.000     4.464589    6.027078
 ammonium .. |   5.945833   .3986014    14.92   0.000     5.164589    6.727078
       IBDU  |   7.945834   .3986014    19.93   0.000     7.164589    8.727078
  Urea (SC)  |   8.595833   .3986014    21.56   0.000     7.814589    9.377078
             |
      thatch |
          2  |       -1.1   .4632314    -2.37   0.018    -2.007917   -.1920828
          5  |   .1500006   .4632314     0.32   0.746    -.7579163    1.057917
             |
    nitrogen#|
      thatch |
     urea#2  |  -.1500005   .6551081    -0.23   0.819    -1.433989    1.133988
     urea#5  |   .0999994   .6551081     0.15   0.879    -1.183989    1.383988
ammonium .. #|
          2  |   .8999996   .6551081     1.37   0.169    -.3839887    2.183988
ammonium .. #|
          5  |  -.1000006   .6551081    -0.15   0.879    -1.383989    1.183988
     IBDU#2  |  -.2000005   .6551081    -0.31   0.760    -1.483989    1.083988
     IBDU#5  |  -1.950001   .6551081    -2.98   0.003    -3.233989   -.6660124
             |
       block |
          1  |  -.2916666   .2643563    -1.10   0.270    -.8097955    .2264622
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
wpunit: Identity             |
                  var(_cons) |   .0682407   .1195933      .0021994    2.117344
-----------------------------+------------------------------------------------
               var(Residual) |   .2145833   .1072917       .080537    .5717376
------------------------------------------------------------------------------
LR test vs. linear regression: chibar2(01) =     0.53 Prob >= chibar2 = 0.2324
```

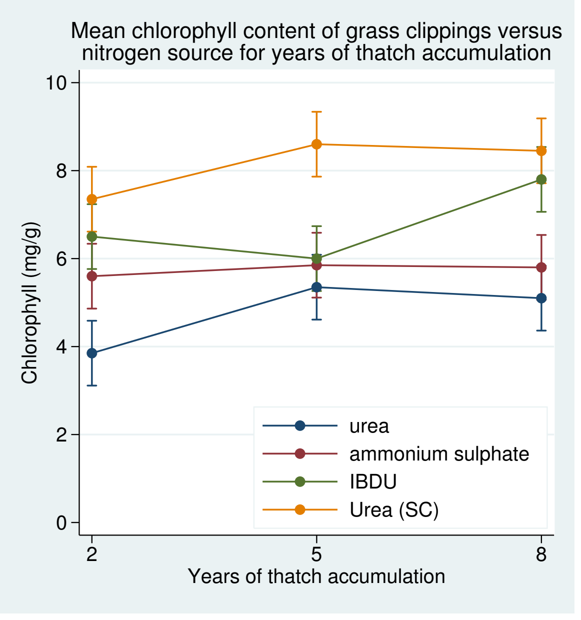
We can calculate the cell means for source of nitrogen and years of thatch accumulation by using `margins`.

```
. margins thatch#nitrogen

Predictive margins                              Number of obs     =         24

Expression   : Linear prediction, fixed portion, predict()
```

	Margin	Delta-method Std. Err.	z	P>|z|	[95% Conf. Interval]	
thatch# nitrogen						
2#urea	3.85	.3760479	10.24	0.000	3.11296	4.58704
2 # ammonium ..	5.6	.3760479	14.89	0.000	4.86296	6.33704
2#IBDU	6.5	.3760479	17.29	0.000	5.76296	7.23704
2#Urea (SC)	7.35	.3760479	19.55	0.000	6.61296	8.087041
5#urea	5.35	.3760479	14.23	0.000	4.61296	6.087041
5 # ammonium ..	5.85	.3760479	15.56	0.000	5.11296	6.58704
5#IBDU	6	.3760479	15.96	0.000	5.26296	6.73704
5#Urea (SC)	8.6	.3760479	22.87	0.000	7.86296	9.337041
8#urea	5.1	.3760479	13.56	0.000	4.36296	5.837041
8 # ammonium ..	5.8	.3760479	15.42	0.000	5.06296	6.53704
8#IBDU	7.8	.3760479	20.74	0.000	7.06296	8.537041
8#Urea (SC)	8.45	.3760479	22.47	0.000	7.712959	9.18704

It is easier to see the effect of the treatments if we plot the impact of the four nitrogen and the three thatch treatments. We can use `marginsplot` to plot the means of chlorophyll content versus years of thatch accumulation by nitrogen source.

```
. marginsplot, ytitle(Chlorophyll (mg/g)) title("")
> subtitle("Mean chlorophyll content of grass clippings versus"
> "nitrogen source for years of thatch accumulation") xsize(3) ysize(3.2)
> legend(cols(1) position(5) ring(0) region(lwidth(none)))
> ylabel(0(2)10, angle(0))
  Variables that uniquely identify margins: thatch nitrogen
```



We can see an increase in the mean chlorophyll content over the years of thatch accumulation for all but one nitrogen source.

The marginal means can be obtained by using `margins` on one variable at a time.

```
. margins thatch
Predictive margins                              Number of obs     =         24
Expression   : Linear prediction, fixed portion, predict()
```

	Margin	Delta-method Std. Err.	z	P>|z|	[95% Conf.	Interval]
thatch						
2	5.825	.188024	30.98	0.000	5.45648	6.19352
5	6.45	.188024	34.30	0.000	6.08148	6.81852
8	6.7875	.188024	36.10	0.000	6.41898	7.15602

```
. margins nitrogen
Predictive margins                              Number of obs     =         24
Expression   : Linear prediction, fixed portion, predict()

------------------------------------------------------------------------------
             |            Delta-method
             |     Margin   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    nitrogen |
       urea  |   4.766667   .2643563    18.03   0.000     4.248538    5.284796
  ammonium ..|       5.75   .2643563    21.75   0.000     5.231871    6.268129
       IBDU  |   6.766667   .2643563    25.60   0.000     6.248538    7.284796
   Urea (SC) |   8.133333   .2643563    30.77   0.000     7.615205    8.651462
------------------------------------------------------------------------------
```

Marchenko (2006) shows more examples of fitting other experimental designs using linear mixed-effects models.

◁

▷ Example 3: Binomial counts

We use the data taken from Agresti (2013, 219) on graduate school applications to the 23 departments within the College of Liberal Arts and Sciences at the University of Florida during the 1997–1998 academic year. The dataset contains the department ID (`department`), the number of applications (`napplied`), and the number of students admitted (`nadmitted`) cross-classified by gender (`female`).

```
. use http://www.stata-press.com/data/r13/admissions, clear
(Graduate school admissions data)

. describe

Contains data from http://www.stata-press.com/data/r13/admissions.dta
  obs:            46                          Graduate school admissions data
 vars:             4                          25 Feb 2013 09:28
 size:           460                          (_dta has notes)
--------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
--------------------------------------------------------------------------------
department      long    %8.0g      dept       department id
nadmitted       byte    %8.0g                 number of admissions
napplied        float   %9.0g                 number of applications
female          byte    %8.0g                 =1 if female, =0 if male
--------------------------------------------------------------------------------
Sorted by:
```

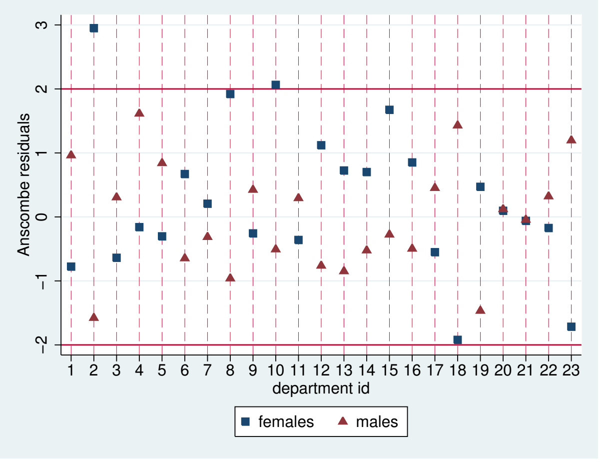
We wish to investigate whether admission decisions are independent of gender. Given department and gender, the probability of admission follows a binomial model, that is, $\Pr(Y_{ij} = y_{ij}) = \text{Binomial}(n_{ij}, \pi_{ij})$, where $i = \{0, 1\}$ and $j = 1, \ldots, 23$. We fit a mixed-effects binomial logistic model with a random intercept at the department level.

```
. melogit nadmitted female || department:, binomial(napplied) or
Fitting fixed-effects model:
Iteration 0:   log likelihood = -302.47786
Iteration 1:   log likelihood = -300.00004
Iteration 2:   log likelihood = -299.99934
Iteration 3:   log likelihood = -299.99934
Refining starting values:
Grid node 0:   log likelihood = -145.08843
Fitting full model:
Iteration 0:   log likelihood = -145.08843
Iteration 1:   log likelihood =  -140.8514
Iteration 2:   log likelihood = -140.61709
Iteration 3:   log likelihood = -140.61628
Iteration 4:   log likelihood = -140.61628
Mixed-effects logistic regression               Number of obs      =        46
Binomial variable:     napplied
Group variable:      department                 Number of groups   =        23
                                                Obs per group: min =         2
                                                               avg =       2.0
                                                               max =         2
Integration method: mvaghermite                 Integration points =         7
                                                Wald chi2(1)       =      2.14
Log likelihood = -140.61628                     Prob > chi2        =    0.1435

   nadmitted   Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]

      female     1.176898   .1310535     1.46   0.144     .9461357    1.463944
       _cons     .7907009   .2057191    -0.90   0.367     .4748457    1.316655

department
   var(_cons)    1.345383    .460702                      .6876497    2.632234

LR test vs. logistic regression: chibar2(01) =   318.77 Prob>=chibar2 = 0.0000
```

The odds of being admitted are higher for females than males but without statistical significance. The estimate of $\widehat{\sigma}_u^2$ is 1.35 with the standard error of 0.46. An LR test comparing the model with the one-level binomial regression model favors the random-intercept model, indicating that there is a significant variation in the number of admissions between departments.

We can further assess the model fit by performing a residual analysis. For example, here we predict and plot Anscombe residuals.

```
. predict anscres, anscombe
(predictions based on fixed effects and posterior means of random effects)
(using 7 quadrature points)
. twoway (scatter anscres department if female, msymbol(S))
> (scatter anscres department if !female, msymbol(T)),
> yline(-2 2) xline(1/23, lwidth(vvthin) lpattern(dash))
> xlabel(1/23) legend(label(1 "females") label(2 "males"))
```



Anscombe residuals are constructed to be approximately normally distributed, thus residuals that are above two in absolute value are usually considered outliers. In the graph above, the residual for female admissions in department 2 is a clear outlier, suggesting a poor fit for that particular observation; see [ME] **meglm postestimation** for more information about Anscombe residuals and other model diagnostics tools.

◁

Covariance structures

▷ Example 4: Growth-curve model with correlated random effects

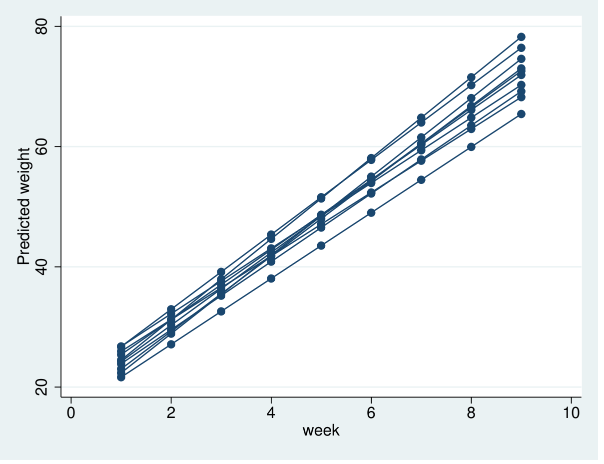
Here we extend the model from example 1 of [ME] **me** to allow for a random slope on `week` and an unstructured covariance structure between the random intercept and the random slope on `week`.

```
. use http://www.stata-press.com/data/r13/pig, clear
(Longitudinal analysis of pig weights)

. mixed weight week || id: week, covariance(unstructured)

Performing EM optimization:

Performing gradient-based optimization:

Iteration 0:   log likelihood = -868.96185
Iteration 1:   log likelihood = -868.96185

Computing standard errors:

Mixed-effects ML regression                     Number of obs      =       432
Group variable: id                              Number of groups   =        48

                                                Obs per group: min =         9
                                                               avg =       9.0
                                                               max =         9

                                                Wald chi2(1)       =   4649.17
Log likelihood = -868.96185                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
      weight |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        week |   6.209896   .0910745    68.18   0.000     6.031393    6.388399
       _cons |   19.35561   .3996387    48.43   0.000     18.57234    20.13889
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
id: Unstructured             |
                   var(week) |   .3715251   .0812958      .2419532     .570486
                  var(_cons) |   6.823363   1.566194      4.351297    10.69986
             cov(week,_cons) |  -.0984378   .2545767     -.5973991    .4005234
-----------------------------+------------------------------------------------
               var(Residual) |   1.596829    .123198      1.372735    1.857505
------------------------------------------------------------------------------
LR test vs. linear regression:       chi2(3) =   764.58   Prob > chi2 = 0.0000

Note: LR test is conservative and provided only for reference.
```

The `unstructured` covariance structure allows for correlation between the random effects. Other covariance structures supported by `mixed`, besides the default `independent`, include `identity` and `exchangeable`; see [ME] **mixed** for details. You can also specify multiple random-effects equations at the same level, in which case the covariance types can be combined to form more complex blocked-diagonal covariance structures; see example 5 below.

We can predict the fitted values and plot the estimated regression line for each of the pigs. The fitted values are based on both the fixed and the random effects.

```
. predict wgt_hat, fitted
. twoway connected wgt_hat week if id<=10, connect(L) ytitle("Predicted weight")
```



◁

▷ Example 5: Blocked-diagonal covariance structures

In this example, we fit a logistic mixed-effects model with a blocked-diagonal covariance structure of random effects.

We use the data from the 1989 Bangladesh fertility survey (Huq and Cleland 1990), which polled 1,934 Bangladeshi women on their use of contraception. 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 coefficients on each indicator variable for children and an overall district random intercept.

```
. use http://www.stata-press.com/data/r13/bangladesh, clear
(Bangladesh Fertility Survey, 1989)
. melogit c_use urban age child* || district: child*, cov(exchangeable)
> || district:, or
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 = -1234.3979
Fitting full model:
Iteration 0:   log likelihood = -1234.3979  (not concave)
Iteration 1:   log likelihood = -1208.0052
Iteration 2:   log likelihood = -1206.4497
Iteration 3:   log likelihood = -1206.2417
Iteration 4:   log likelihood = -1206.2397
Iteration 5:   log likelihood = -1206.2397
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)       =    100.01
Log likelihood = -1206.2397                     Prob > chi2        =    0.0000
 ( 1)  [var(child1[district])]_cons - [var(child3[district])]_cons = 0
 ( 2)  [cov(child2[district],child1[district])]_cons -
       [cov(child3[district],child2[district])]_cons = 0
 ( 3)  [cov(child3[district],child1[district])]_cons -
       [cov(child3[district],child2[district])]_cons = 0
 ( 4)  [var(child2[district])]_cons - [var(child3[district])]_cons = 0
```

c_use	Odds Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
urban	2.105163	.2546604	6.15	0.000	1.660796	2.668426
age	.9735765	.0077461	-3.37	0.001	.9585122	.9888775
child1	2.992596	.502149	6.53	0.000	2.153867	4.157931
child2	3.879345	.7094125	7.41	0.000	2.710815	5.551584
child3	3.774627	.7055812	7.11	0.000	2.616744	5.444863
_cons	.1859471	.0274813	-11.38	0.000	.1391841	.2484214
district						
var(child1)	.0841518	.0880698			.0108201	.654479
var(child2)	.0841518	.0880698			.0108201	.654479
var(child3)	.0841518	.0880698			.0108201	.654479
var(_cons)	.1870273	.0787274			.0819596	.426786
district						
cov(child2, child1)	.0616875	.0844681	0.73	0.465	-.1038669	.2272419
cov(child3, child1)	.0616875	.0844681	0.73	0.465	-.1038669	.2272419
cov(child3, child2)	.0616875	.0844681	0.73	0.465	-.1038669	.2272419

```
LR test vs. logistic regression:     chi2(3) =    44.57   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

The fixed effects can be interpreted just as you would the output from `logit`. Urban women have roughly double the odds of using contraception as compared with 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.

Because we specified `cov(exchangeable)`, the estimated variances on each indicator variable for children are constrained to be the same, and the estimated covariances on each indicator variable for children are constrained to be the same. More complex covariance structures with constraints can be specified using `covariance(pattern())` and `covariance(fixed())`; see example 6 below.

◁

▷ Example 6: Meta analysis

In this example, we present a mixed-effects model for meta analysis of clinical trials. The term "meta-analysis" refers to a statistical analysis that involves summary data from similar but independent studies.

Turner et al. (2000) performed a study of nine clinical trials examining the effect of taking diuretics during pregnancy on the risk of pre-eclampsia. The summary data consist of the log odds-ratio (variable `or`) estimated from each study, and the corresponding estimated variance (variable `varor`). The square root of the variance is stored in the variable `std` and the trial identifier is stored in the variable `trial`.

```
. use http://www.stata-press.com/data/r13/diuretics
(Meta analysis of clinical trials studying diuretics and pre-eclampsia)
. list
```

	trial	or	varor	std
1.	1	.04	.16	.4
2.	2	-.92	.12	.3464102
3.	3	-1.12	.18	.4242641
4.	4	-1.47	.3	.5477226
5.	5	-1.39	.11	.3316625
6.	6	-.3	.01	.1
7.	7	-.26	.12	.3464102
8.	8	1.09	.69	.8306624
9.	9	.14	.07	.2645751

In a random-effects modeling of summary data, the observed log odds-ratios are treated as a continuous outcome and assumed to be normally distributed, and the true treatment effect varies randomly among the trials. The random-effects model can be written as

$$y_i \sim N(\theta + \nu_i, \sigma_i^2)$$
$$\nu_i \sim N(0, \tau^2)$$

where y_i is the observed treatment effect corresponding to the ith study, $\theta + \nu_i$ is the true treatment effect, σ_i^2 is the variance of the observed treatment effect, and τ is the between-trial variance component. Our aim is to estimate θ, the global mean.

Notice that the responses y_i do not provide enough information to estimate this model, because we cannot estimate the group-level variance component from a dataset that contains one observation per group. However, we already have estimates for the σ_i's, therefore we can constrain each σ_i to be equal to its estimated value, which will allow us to estimate θ and τ. We use meglm to estimate this model because the mixed command does not support constraints.

In meglm, one way to constrain a group of individual variances to specific values is by using the fixed covariance structure (an alternative way is to define each constraint individually with the constraint command and specify them in the constraints() option). The covariance(fixed()) option requires a Stata matrix defining the constraints, thus we first create matrix f with the values of σ_i, stored in variable varor, on the main diagonal. We will use this matrix to constrain the variances.

```
. mkmat varor, mat(f)
. mat f = diag(f)
```

In the random-effects equation part, we need to specify nine random slopes, one for each trial. Because random-effects equations do not support factor variables (see **[U] 11.4.3 Factor variables**), we cannot use the i.trial notation. Instead, we tabulate the variable trial and use the generate() option to create nine dummy variables named tr1, tr2, ..., tr9. We can then fit the model. Because the model is computationally demanding, we use Laplacian approximation instead of the default mean-variance adaptive quadrature; see *Computation time and the Laplacian approximation* above for details.

```
. qui tabulate trial, gen(tr)
. meglm or || _all: tr1-tr9, nocons cov(fixed(f)) intm(laplace) nocnsreport
Fitting fixed-effects model:
Iteration 0:   log likelihood = -10.643432
Iteration 1:   log likelihood = -10.643432
Refining starting values:
Grid node 0:   log likelihood = -10.205455
Fitting full model:
Iteration 0:   log likelihood = -10.205455
Iteration 1:   log likelihood = -9.4851561  (backed up)
Iteration 2:   log likelihood = -9.4587068
Iteration 3:   log likelihood = -9.4552982
Iteration 4:   log likelihood = -9.4552759
Iteration 5:   log likelihood = -9.4552759
Mixed-effects GLM                               Number of obs      =          9
Family:                 Gaussian
Link:                   identity
Group variable:             _all                Number of groups   =          1
                                                Obs per group: min =          9
                                                               avg =        9.0
                                                               max =          9
Integration method:      laplace
                                                Wald chi2(0)       =          .
Log likelihood = -9.4552759                     Prob > chi2        =          .

------------------------------------------------------------------------------
          or |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _cons |  -.5166151   .2059448    -2.51   0.012    -.9202594   -.1129707
-------------+----------------------------------------------------------------
_all         |
    var(tr1) |        .16  (constrained)
    var(tr2) |        .12  (constrained)
    var(tr3) |        .18  (constrained)
    var(tr4) |         .3  (constrained)
    var(tr5) |        .11  (constrained)
    var(tr6) |        .01  (constrained)
    var(tr7) |        .12  (constrained)
    var(tr8) |        .69  (constrained)
    var(tr9) |        .07  (constrained)
-------------+----------------------------------------------------------------
   var(e.or) |   .2377469   .1950926                      .0476023    1.187413
------------------------------------------------------------------------------
```

We estimate $\widehat{\theta} = -0.52$, which agrees with the estimate reported by Turner et al. (2000).

We can fit the above model in a more efficient way. We can consider the trials as nine independent random variables, each with variance unity, and each being multiplied by a different standard error. To accomplish this, we treat `trial` as a random-effects level, use the standard deviations of the log odds-ratios as a random covariate at the `trial` level, and constrain the variance component of `trial` to unity.

```
. constraint 1 _b[var(std[trial]):_cons] = 1
. meglm or || trial: std, nocons constraints(1)
Fitting fixed-effects model:
Iteration 0:   log likelihood = -10.643432
Iteration 1:   log likelihood = -10.643432
Refining starting values:
Grid node 0:   log likelihood = -10.205455
Fitting full model:
Iteration 0:   log likelihood = -10.205455
Iteration 1:   log likelihood = -9.4851164  (backed up)
Iteration 2:   log likelihood =   -9.45869
Iteration 3:   log likelihood = -9.4552794
Iteration 4:   log likelihood = -9.4552759
Iteration 5:   log likelihood = -9.4552759
Mixed-effects GLM                               Number of obs      =          9
Family:                  Gaussian
Link:                    identity
Group variable:             trial               Number of groups   =          9
                                                Obs per group: min =          1
                                                               avg =        1.0
                                                               max =          1
Integration method: mvaghermite                 Integration points =          7
                                                Wald chi2(0)       =          .
Log likelihood = -9.4552759                     Prob > chi2        =          .
 ( 1)  [var(std[trial])]_cons = 1
```

or	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
_cons	-.5166151	.2059448	-2.51	0.012	-.9202594	-.1129708
trial						
var(std)	1	(constrained)				
var(e.or)	.2377469	.1950926			.0476023	1.187413

The results are the same, but this model took a fraction of the time compared with the less efficient specification.

◁

Three-level models

The methods we have discussed so far extend from two-level models 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. Below we illustrate a three-level mixed-effects ordered probit model.

▷ Example 7: Three-level ordinal response model

In this example, we fit a three-level ordered probit model. The data are from the Television, School, and Family Smoking Prevention and Cessation Project (Flay et al. 1988; Rabe-Hesketh and Skrondal 2012, chap. 11), where schools were randomly assigned into one of four groups defined by two treatment variables. Students within each school are nested in classes, and classes are nested in schools. The dependent variable is the tobacco and health knowledge (THK) scale score collapsed into four ordered categories. We regress the outcome on the treatment variables and their interaction and control for the pretreatment score.

```
. use http://www.stata-press.com/data/r13/tvsfpors, clear
. meoprobit thk prethk cc##tv || school: || class:
Fitting fixed-effects model:
Iteration 0:   log likelihood =  -2212.775
Iteration 1:   log likelihood = -2127.8111
Iteration 2:   log likelihood = -2127.7612
Iteration 3:   log likelihood = -2127.7612
Refining starting values:
Grid node 0:   log likelihood = -2195.6424
Fitting full model:
Iteration 0:   log likelihood = -2195.6424  (not concave)
Iteration 1:   log likelihood = -2167.9576  (not concave)
Iteration 2:   log likelihood = -2140.2644  (not concave)
Iteration 3:   log likelihood = -2128.6948  (not concave)
Iteration 4:   log likelihood = -2119.9225
Iteration 5:   log likelihood = -2117.0947
Iteration 6:   log likelihood = -2116.7004
Iteration 7:   log likelihood = -2116.6981
Iteration 8:   log likelihood = -2116.6981
Mixed-effects oprobit regression                Number of obs      =      1600

-----------------------------------------------------------
                |   No. of       Observations per Group
 Group Variable |   Groups    Minimum    Average    Maximum
----------------+------------------------------------------
         school |       28         18       57.1        137
          class |      135          1       11.9         28
-----------------------------------------------------------

Integration method: mvaghermite                 Integration points =         7

                                                Wald chi2(4)       =    124.20
Log likelihood = -2116.6981                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
         thk |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      prethk |    .238841   .0231446    10.32   0.000     .1934784    .2842036
        1.cc |   .5254813   .1285816     4.09   0.000     .2734659    .7774967
        1.tv |   .1455573   .1255827     1.16   0.246    -.1005803    .3916949
             |
       cc#tv |
         1 1 |  -.2426203   .1811999    -1.34   0.181    -.5977656    .1125251
-------------+----------------------------------------------------------------
       /cut1 |   -.074617   .1029791    -0.72   0.469    -.2764523    .1272184
       /cut2 |   .6863046   .1034813     6.63   0.000     .4834849    .8891242
       /cut3 |   1.413686   .1064889    13.28   0.000     1.204972    1.622401
-------------+----------------------------------------------------------------
school       |
   var(_cons)|   .0186456   .0160226                      .0034604    .1004695
-------------+----------------------------------------------------------------
school>class |
   var(_cons)|   .0519974   .0224014                      .0223496    .1209745
------------------------------------------------------------------------------
LR test vs. oprobit regression:      chi2(2) =    22.13   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

Notes:

1. Our model now has two random-effects equations, separated by `||`. The first is a random intercept (constant only) at the `school` level (level three), and the second is a random intercept at the `class` level (level two). The order in which these are specified (from left to right) is significant—`meoprobit` assumes that `class` is nested within `school`.
2. The information on groups is now displayed as a table, with one row for each grouping. You can suppress this table with the `nogroup` or the `noheader` option, which will also suppress the rest of the header.
3. The variance-component estimates are now organized and labeled according to level. The variance component for `class` is labeled `school>class` to emphasize that classes are nested within schools.

◁

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).

Crossed-effects models

Not all mixed-effects models contain nested levels of random effects.

▷ Example 8: Crossed random effects

Returning to our longitudinal analysis of pig weights, suppose that we wish to fit

$$\texttt{weight}_{ij} = \beta_0 + \beta_1 \texttt{week}_{ij} + u_i + v_j + \epsilon_{ij} \tag{11}$$

for the $i = 1, \ldots, 9$ weeks and $j = 1, \ldots, 48$ pigs and

$$u_i \sim N(0, \sigma_u^2); \quad v_j \sim N(0, \sigma_v^2); \quad \epsilon_{ij} \sim N(0, \sigma_\epsilon^2)$$

all independently. That is, we assume an overall population-average growth curve $\beta_0 + \beta_1 \texttt{week}$ and a random pig-specific shift. In other words, the effect due to `week`, u_i, is systematic to that week and common to all pigs. The rationale behind (11) could be that, assuming that the pigs were measured contemporaneously, we might be concerned that week-specific random factors such as weather and feeding patterns had significant systematic effects on all pigs.

Model (11) is an example of a two-way crossed-effects model, with the pig effects v_j being crossed with the week effects u_i. One way to fit such models is to consider all the data as one big cluster, and treat u_i and v_j as a series of $9 + 48 = 57$ random coefficients on indicator variables for `week` and `pig`. The random effects $\mathbf{u}$ and the variance components $\mathbf{G}$ are now represented as

$$\mathbf{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_9 \\ v_1 \\ \vdots \\ v_{48} \end{bmatrix} \sim N(\mathbf{0}, \mathbf{G}); \quad \mathbf{G} = \begin{bmatrix} \sigma_u^2 \mathbf{I}_9 & \mathbf{0} \\ \mathbf{0} & \sigma_v^2 \mathbf{I}_{48} \end{bmatrix}$$

Because $\mathbf{G}$ is block diagonal, it can be represented as repeated-level equations. All we need is an ID variable to identify all the observations as one big group and a way to tell mixed-effects commands to treat `week` and `pig` as factor variables (or equivalently, as two sets of overparameterized indicator variables identifying weeks and pigs, respectively). The mixed-effects commands support the special group designation `_all` for the former and the `R.`*varname* notation for the latter.

```
. use http://www.stata-press.com/data/r13/pig
(Longitudinal analysis of pig weights)

. mixed weight week || _all: R.id || _all: R.week

Performing EM optimization:

Performing gradient-based optimization:

Iteration 0:   log likelihood =  -1013.824
Iteration 1:   log likelihood =  -1013.824

Computing standard errors:

Mixed-effects ML regression                     Number of obs      =       432
Group variable: _all                            Number of groups   =         1

                                                Obs per group: min =       432
                                                               avg =     432.0
                                                               max =       432

                                                Wald chi2(1)       =  13258.28
Log likelihood =  -1013.824                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
      weight |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        week |   6.209896   .0539313   115.14   0.000     6.104192    6.315599
       _cons |   19.35561   .6333982    30.56   0.000     18.11418    20.59705
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
_all: Identity               |
                   var(R.id) |   14.83623   3.126142      9.816733    22.42231
-----------------------------+------------------------------------------------
_all: Identity               |
                 var(R.week) |   .0849874   .0868856      .0114588    .6303302
-----------------------------+------------------------------------------------
               var(Residual) |   4.297328   .3134404      3.724888    4.957741
------------------------------------------------------------------------------
LR test vs. linear regression:       chi2(2) =   474.85   Prob > chi2 = 0.0000

Note: LR test is conservative and provided only for reference.
```

We estimate $\widehat{\sigma}_u^2 = 0.08$ and $\widehat{\sigma}_v^2 = 14.84$.

The `R.`*varname* notation is equivalent to giving a list of overparameterized (none dropped) indicator variables for use in a random-effects specification. When you use `R.`*varname*, mixed-effects commands handle the calculations internally rather than creating the indicators in the data. Because the set of indicators is overparameterized, `R.`*varname* implies `noconstant`.

Note that the column dimension of our random-effects design is 57. Computation time and memory requirements grow (roughly) quadratically with the dimension of the random effects. As a result, fitting such crossed-effects models is feasible only when the total column dimension is small to moderate. For this reason, mixed-effects commands use the Laplacian approximation as the default estimation method for crossed-effects models; see *Computation time and the Laplacian approximation* above for more details.

It is often possible to rewrite a mixed-effects model in a way that is more computationally efficient. For example, we can treat pigs as nested within the `_all` group, yielding the equivalent and more efficient (total column dimension 10) way to fit (11):

```
. mixed weight week || _all: R.week || id:
```

The results of both estimations are identical, but the latter specification, organized at the cluster (pig) level with random-effects dimension 1 (a random intercept) is much more computationally efficient. Whereas with the first form we are limited in how many pigs we can analyze, there is no such limitation with the second form.

All the mixed-effects commands—except `mixed`, `meqrlogit`, and `meqrpoisson`—automatically attempt to recast the less efficient model specification into a more efficient one. However, this automatic conversion may not be sufficient for some complicated mixed-effects specifications, especially if both crossed and nested effects are involved. Therefore, we strongly encourage you to always specify the more efficient syntax; see Rabe-Hesketh and Skrondal (2012) and Marchenko (2006) for additional techniques to make calculations more efficient in more complex mixed-effects models.

◁

Acknowledgments

We are indebted to Sophia Rabe-Hesketh of the University of California, Berkeley; Anders Skrondal of the University of Oslo and the Norwegian Institute of Public Health; and Andrew Pickles of the University of Manchester for their extensive body of work in Stata, both previous and ongoing, in this area.

References

Agresti, A. 2013. *Categorical Data Analysis*. 3rd ed. Hoboken, NJ: Wiley.

Bates, D. M., and J. C. Pinheiro. 1998. Computational methods for multilevel modelling. In *Technical Memorandum BL0112140-980226-01TM*. Murray Hill, NJ: Bell Labs, Lucent Technologies. http://stat.bell-labs.com/NLME/CompMulti.pdf.

Breslow, N. E., and D. G. Clayton. 1993. Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association* 88: 9–25.

De Boeck, P., and M. Wilson, ed. 2004. *Explanatory Item Response Models: A Generalized Linear and Nonlinear Approach*. New York: Springer.

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

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B* 39: 1–38.

Diggle, P. J., P. J. Heagerty, K.-Y. Liang, and S. L. Zeger. 2002. *Analysis of Longitudinal Data*. 2nd ed. Oxford: Oxford University Press.

Flay, B. R., B. R. Brannon, C. A. Johnson, W. B. Hansen, A. L. Ulene, D. A. Whitney-Saltiel, L. R. Gleason, S. Sussman, M. D. Gavin, K. M. Glowacz, D. F. Sobol, and D. C. Spiegel. 1988. The television, school, and family smoking cessation and prevention project: I. Theoretical basis and program development. *Preventive Medicine* 17: 585–607.

Gelman, A., and J. Hill. 2007. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge: Cambridge University Press.

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.

Hall, B. H., Z. Griliches, and J. A. Hausman. 1986. Patents and R and D: Is there a lag? *International Economic Review* 27: 265–283.

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

Henderson, C. R. 1953. Estimation of variance and covariance components. *Biometrics* 9: 226–252.

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

Kuehl, R. O. 2000. *Design of Experiments: Statistical Principles of Research Design and Analysis*. 2nd ed. Belmont, CA: Duxbury.

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

LaMotte, L. R. 1973. Quadratic estimation of variance components. *Biometrics* 29: 311–330.

Lesaffre, E., and B. Spiessens. 2001. On the effect of the number of quadrature points in a logistic random-effects model: An example. *Journal of the Royal Statistical Society, Series C* 50: 325–335.

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.

Littell, R. C., G. A. Milliken, W. W. Stroup, R. D. Wolfinger, and O. Schabenberger. 2006. *SAS System for Mixed Models*. 2nd ed. Cary, NC: SAS Institute.

Liu, Q., and D. A. Pierce. 1994. A note on Gauss–Hermite quadrature. *Biometrika* 81: 624–629.

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.

Pinheiro, J. C., and D. M. Bates. 2000. *Mixed-Effects Models in S and S-PLUS*. New York: Springer.

Pinheiro, J. C., and E. C. Chao. 2006. Efficient Laplacian and adaptive Gaussian quadrature algorithms for multilevel generalized linear mixed models. *Journal of Computational and Graphical Statistics* 15: 58–81.

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

Rao, C. R. 1973. *Linear Statistical Inference and Its Applications*. 2nd ed. New York: Wiley.

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

Rodríguez, G., and N. Goldman. 1995. An assessment of estimation procedures for multilevel models with binary responses. *Journal of the Royal Statistical Society, Series A* 158: 73–89.

Ruppert, D., M. P. Wand, and R. J. Carroll. 2003. *Semiparametric Regression*. Cambridge: Cambridge University Press.

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.

Skrondal, A., and S. Rabe-Hesketh. 2004. *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models*. Boca Raton, FL: Chapman & Hall/CRC.

Stram, D. O., and J. W. Lee. 1994. Variance components testing in the longitudinal mixed effects model. *Biometrics* 50: 1171–1177.

Thompson, W. A., Jr. 1962. The problem of negative estimates of variance components. *Annals of Mathematical Statistics* 33: 273–289.

Tierney, L., and J. B. Kadane. 1986. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association* 81: 82–86.

Turner, R. M., R. Z. Omar, M. Yang, H. Goldstein, and S. G. Thompson. 2000. A multilevel model framework for meta-analysis of clinical trials with binary outcomes. *Statistics in Medicine* 19: 3417–3432.

Tutz, G., and W. Hennevogl. 1996. Random effects in ordinal regression models. *Computational Statistics & Data Analysis* 22: 537–557.

Vella, F., and M. Verbeek. 1998. Whose wages do unions raise? A dynamic model of unionism and wage rate determination for young men. *Journal of Applied Econometrics* 13: 163–183.

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

Also see

[ME] **Glossary**

Title

mecloglog — Multilevel mixed-effects complementary log-log regression

Syntax

`mecloglog` *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	
binomial(*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, robust, or cluster *clustvar*
Reporting	
level(*#*)	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
nolrtest	do not perform likelihood-ratio test comparing with complementary log-log regression
display_options	control 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
coeflegend	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
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 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 > 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.

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**.

`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.

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

display_options: `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 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 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**.

coeflegend; see [R] **estimation options**.

Remarks and examples

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

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

Remarks are presented under the following headings:

Introduction

Mixed-effects complementary log-log regression is complementary log-log 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 McCulloch (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.

`mecloglog` 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.

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 $j = 1, \dots, M$ clusters, with cluster j consisting of $i = 1, \dots, n_j$ 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 $\sigma^2_\epsilon = \pi^2/6$, 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.

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 itself is estimated, 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.

mecloglog 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 mecloglog 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 xtcloglog with the re option; see [XT] **xtcloglog**.

Below we present two short examples of mixed-effects cloglog regression; refer to [ME] **melogit** for additional examples including crossed-effects models and to [ME] **me** and [ME] **meglm** for examples of other 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 cloglog regression; see [R] **cloglog**.

▷ Example 1

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 http://www.stata-press.com/data/r13/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      =     26200
Group variable:           idcode                Number of groups   =      4434

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

Integration method: mvaghermite                 Integration points =        12

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

------------------------------------------------------------------------------
       union |      Coef.   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 regression:  chibar2(01) =  6009.33 Prob>=chibar2 = 0.0000
```

The estimates are practically the same. `xtcloglog` reports the estimated variance component as a standard deviation, $\widehat{\sigma}_{\mathbf{u}} = 1.86$. `mecloglog` reports $\widehat{\sigma}^2_{\mathbf{u}} = 3.49$, 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, $\widehat{\boldsymbol{\sigma}}^2$, 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.

◁

Three-level models

Two-level models extend naturally to models with three or more levels with nested random effects. Below we analyze the data from example 2 of [ME] **melogit** with `mecloglog`.

▷ Example 2

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.

```
. use http://www.stata-press.com/data/r13/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

-------------------------------------------------------------
                |   No. of       Observations per Group
 Group Variable |   Groups    Minimum    Average    Maximum
----------------+--------------------------------------------
         family |      118          2        5.7         27
        subject |      226          2        3.0          3
-------------------------------------------------------------

Integration method: mvaghermite                 Integration points =         7

                                                Wald chi2(3)       =     83.32
Log likelihood = -305.26516                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
        dtlm |      Coef.   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 regression:      chi2(2) =    16.61   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—`mecloglog` 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.

◁

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

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)	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)`	`mecloglog`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`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`
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 `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

$$
\begin{aligned}
f(\mathbf{y}_j|\mathbf{u}_j) &= \prod_{i=1}^{n_j} \left[\binom{r_{ij}}{y_{ij}} \left\{H(\boldsymbol{\eta}_{ij})\right\}^{y_{ij}} \left\{1 - H(\boldsymbol{\eta}_{ij})\right\}^{r_{ij}-y_{ij}} \right] \\
&= \exp\left(\sum_{i=1}^{n_j} \left[y_{ij} \log\left\{H(\boldsymbol{\eta}_{ij})\right\} - (r_{ij} - y_{ij}) \exp(\boldsymbol{\eta}_{ij}) + \log\binom{r_{ij}}{y_{ij}} \right] \right)
\end{aligned}
$$

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

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

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

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

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

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

where

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

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. `mecloglog` offers four approximation methods: mean–variance adaptive Gauss–Hermite quadrature (default), mode-curvature adaptive Gauss–Hermite quadrature, nonadaptive Gauss–Hermite quadrature, and Laplacian approximation.

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

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.

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

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

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.

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

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

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

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

[U] 20 Estimation and postestimation commands

Title

mecloglog postestimation — Postestimation tools for mecloglog

Description

The following postestimation command is of special interest after `mecloglog`:

Command	Description
`estat group`	summarize the composition of the nested groups

The following standard postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat ic`	Akaike's and Schwarz's Bayesian information criteria (AIC and BIC)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

Special-interest postestimation commands

`estat group` reports the number of groups and minimum, average, and maximum group sizes for each level of the model. Model levels are identified by the corresponding group variable in the data. Because groups are treated as nested, the information in this summary may differ from what you would get if you used the `tabulate` command on each group variable individually.

Syntax for predict

Syntax for obtaining predictions of random effects and their standard errors

predict [*type*] *newvarsspec* [*if*] [*in*], { remeans | remodes } [reses(*newvarsspec*)]

Syntax for obtaining other predictions

predict [*type*] *newvarsspec* [*if*] [*in*] [, *statistic* *options*]

newvarsspec is *stub** or *newvarlist*.

statistic	Description
Main	
mu	predicted mean; the default
fitted	fitted linear predictor
xb	linear predictor for the fixed portion of the model only
stdp	standard error of the fixed-portion linear prediction
pearson	Pearson residuals
deviance	deviance residuals
anscombe	Anscombe residuals

These statistics are available both in and out of sample; type predict ... if e(sample) ... if wanted only for the estimation sample.

options	Description
Main	
means	compute *statistic* using empirical Bayes means; the default
modes	compute *statistic* using empirical Bayes modes
nooffset	ignore the offset variable in calculating predictions; relevant only if you specified offset() when you fit the model
fixedonly	prediction for the fixed portion of the model only
Integration	
intpoints(#)	use # quadrature points to compute empirical Bayes means
iterate(#)	set maximum number of iterations in computing statistics involving empirical Bayes estimators
tolerance(#)	set convergence tolerance for computing statistics involving empirical Bayes estimators

Menu for predict

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

remeans, remodes, reses(); see [ME] **meglm postestimation**.

mu, the default, calculates the predicted mean (the probability of a positive outcome), that is, the inverse link function applied to the linear prediction. By default, this is based on a linear predictor that includes both the fixed effects and the random effects, and the predicted mean is conditional on the values of the random effects. Use the fixedonly option if you want predictions that include only the fixed portion of the model, that is, if you want random effects set to 0.

fitted, xb, stdp, pearson, deviance, anscombe, means, modes, nooffset, fixedonly; see [ME] **meglm postestimation**.

By default or if the means option is specified, statistics mu, fitted, xb, stdp, pearson, deviance, and anscombe are based on the posterior mean estimates of random effects. If the modes option is specified, these statistics are based on the posterior mode estimates of random effects.

Integration

intpoints(), iterate(), tolerance(); see [ME] **meglm postestimation**.

Syntax for estat group

```
estat group
```

Menu for estat

Statistics > Postestimation > Reports and statistics

Remarks and examples

Various predictions, statistics, and diagnostic measures are available after fitting a mixed-effects complementary log-log model with mecloglog. Here we show a short example of predicted probabilities and predicted random effects; refer to [ME] **meglm postestimation** for additional examples.

▷ Example 1

In example 2 of [ME] **mecloglog**, we analyzed the cognitive ability (dtlm) of patients with schizophrenia compared with their relatives and control subjects. We used a three-level complementary log-log model with random effects at the family and subject levels. Cognitive ability was measured as the successful completion of the "Tower of London", a computerized task, measured at three levels of difficulty.

```
. use http://www.stata-press.com/data/r13/towerlondon
(Tower of London data)
. mecloglog dtlm difficulty i.group || family: || subject:
Fitting fixed-effects model:
  (output omitted )
Mixed-effects cloglog regression                Number of obs      =       677

------------------------------------------------------------
                |   No. of       Observations per Group
 Group Variable |   Groups    Minimum    Average    Maximum
----------------+-------------------------------------------
         family |      118          2        5.7         27
        subject |      226          2        3.0          3
------------------------------------------------------------

Integration method: mvaghermite                 Integration points =         7

                                                Wald chi2(3)       =     83.32
Log likelihood = -305.26516                     Prob > chi2        =    0.0000
------------------------------------------------------------------------------
        dtlm |      Coef.   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 regression:       chi2(2) =    16.61   Prob > chi2 = 0.0002
Note: LR test is conservative and provided only for reference.
```

We obtain predicted probabilities based on the contribution of both fixed effects and random effects by typing

```
. predict pr
(predictions based on fixed effects and posterior means of random effects)
(option mu assumed)
(using 7 quadrature points)
```

As the note says, the predicted values are based on the posterior means of random effects. You can use the `modes` option to obtain predictions based on the posterior modes of random effects.

We obtain predictions of the posterior means themselves by typing

```
. predict re*, remeans
(calculating posterior means of random effects)
(using 7 quadrature points)
```

Because we have one random effect at the family level and another random effect at the subject level, Stata saved the predicted posterior means in the variables `re1` and `re2`, respectively. If you are not sure which prediction corresponds to which level, you can use the `describe` command to show the variable labels.

Here we list the data for family 16:

```
. list family subject dtlm pr re1 re2 if family==16, sepby(subject)
```

	family	subject	dtlm	pr	re1	re2
208.	16	5	1	.486453	.4184933	.2760492
209.	16	5	0	.1597047	.4184933	.2760492
210.	16	5	0	.0444156	.4184933	.2760492
211.	16	34	1	.9659582	.4184933	1.261488
212.	16	34	1	.5862808	.4184933	1.261488
213.	16	34	1	.205816	.4184933	1.261488
214.	16	35	0	.5571261	.4184933	-.1616545
215.	16	35	1	.1915688	.4184933	-.1616545
216.	16	35	0	.0540124	.4184933	-.1616545

We can see that the predicted random effects (`re1`) at the family level are the same for all members of the family. Similarly, the predicted random effects (`re2`) at the individual level are constant within each individual. Based on a cutoff of 0.5, the predicted probabilities (`pr`) for this family do not match the observed outcomes (`dtlm`) as well as the predicted probabilities from the logistic example; see example 1 in [ME] **melogit postestimation**.

◁

Methods and formulas

Methods and formulas for predicting random effects and other statistics are given in *Methods and formulas* of [ME] **meglm postestimation**.

Also see

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

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

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

Title

meglm — Multilevel mixed-effects generalized linear model

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

Syntax

`meglm` *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 the constant term from the fixed-effects equation
`exposure(`$varname_e$`)`	include $\ln(varname_e)$ in model with coefficient constrained to 1
`offset(`$varname_o$`)`	include $varname_o$ 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	
family(*family*)	distribution of *depvar*; default is family(gaussian)
link(*link*)	link function; default varies per family
constraints(*constraints*)	apply specified linear constraints
collinear	keep collinear variables
SE/Robust	
vce(*vcetype*)	*vcetype* may be oim, robust, or cluster *clustvar*
Reporting	
level(#)	set confidence level; default is level(95)
eform	report exponentiated fixed-effects coefficients
irr	report fixed-effects coefficients as incidence-rate ratios
or	report fixed-effects coefficients as odds ratios
nocnsreport	do not display constraints
notable	suppress coefficient table
noheader	suppress output header
nogroup	suppress table summarizing groups
nolrtest	do not perform likelihood-ratio test comparing with reference model
display_options	control 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
coeflegend	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
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

family	Description
gaussian	Gaussian (normal); the default
bernoulli	Bernoulli
binomial [# | *varname*]	binomial; default number of binomial trials is 1
gamma	gamma
nbinomial [mean | constant]	negative binomial; default dispersion is mean
ordinal	ordinal
poisson	Poisson

link	Description
identity	identity
log	log
logit	logit
probit	probit
cloglog	complementary log-log

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 > Generalized linear models (GLMs)

Description

meglm fits multilevel mixed-effects generalized linear models. meglm allows a variety of distributions for the response conditional on normally distributed random effects.

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.

exposure(*varname*$_e$) specifies a variable that reflects the amount of exposure over which the *depvar* events were observed for each observation; ln(*varname*$_e$) is included in the fixed-effects portion of the model with the coefficient constrained to be 1.

offset(*varname*$_o$) specifies that *varname*$_o$ 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*$[k,l]$.

family(*family*) specifies the distribution of *depvar*; family(gaussian) is the default.

link(*link*) specifies the link function; the default is the canonical link for the family() specified except for the gamma and negative binomial families.

If you specify both `family()` and `link()`, not all combinations make sense. You may choose from the following combinations:

	identity	log	logit	probit	cloglog
Gaussian	D	x			
Bernoulli			D	x	x
binomial			D	x	x
gamma		D			
negative binomial		D			
ordinal			D	x	x
Poisson		D			

D denotes the default.

`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**.

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

`irr` reports estimated fixed-effects coefficients transformed to incidence-rate 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 stored. `irr` may be specified either at estimation or upon replay. This option is allowed for count models only.

`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 at estimation or upon replay. This option is allowed for logistic models only.

`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 `meglm` from fitting a reference linear regression model and using this model to calculate a likelihood-ratio test comparing the mixed model with ordinary regression. This option may also be specified upon replay to suppress this test from the output.

display_options: `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 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 meglm 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 meglm but are not shown in the dialog box:

startvalues(*svmethod*) specifies how starting values are to be computed. Starting values specified in from() override the computed starting values.

startvalues(zero) specifies that starting values be set to 0.

startvalues(constantonly) builds on startvalues(zero) by fitting a constant-only model to obtain estimates of the intercept and auxiliary parameters, and it substitutes 1 for the variances of random effects.

startvalues(fixedonly) builds on startvalues(constantonly) by fitting a full fixed-effects model to obtain estimates of coefficients along with intercept and auxiliary parameters, and it continues to use 1 for the variances of random effects. This is the default behavior.

startvalues(iv) builds on startvalues(fixedonly) by using instrumental-variable methods with generalized residuals to obtain variances of random effects.

startgrid[(*gridspec*)] performs a grid search on variance components of random effects to improve starting values. No grid search is performed by default unless the starting values are found to be not feasible, in which case meglm runs startgrid() to perform a "minimal" search involving q^3 likelihood evaluations, where q is the number of random effects. Sometimes this resolves the problem. Usually, however, there is no problem and `startgrid()` is not run by default. There can be benefits from running `startgrid()` to get better starting values even when starting values are feasible.

`startgrid()` is a brute-force approach that tries various values for variances and covariances and chooses the ones that work best. You may already be using a default form of `startgrid()` without knowing it. If you see `meglm` displaying Grid node 1, Grid node 2, ... following Grid node 0 in the iteration log, that is `meglm` doing a default search because the original starting values were not feasible. The default form tries 0.1, 1, and 10 for all variances of all random effects.

`startgrid(`*numlist*`)` specifies values to try for variances of random effects.

`startgrid(`*covspec*`)` specifies the particular variances and covariances in which grid searches are to be performed. *covspec* is *name*`[`*level*`]` for variances and *name1*`[`*level*`]*`*name2*`[`*level*`]` for covariances. For example, the variance of the random intercept at level `id` is specified as `_cons[id]`, and the variance of the random slope on variable `week` at the same level is specified as `week[id]`. The residual variance for the linear mixed-effects model is specified as `e.`*depvar*, where *depvar* is the name of the dependent variable. The covariance between the random slope and the random intercept above is specified as `_cons[id]*week[id]`.

`startgrid(`*numlist covspec*`)` combines the two syntaxes. You may also specify `startgrid()` multiple times so that you can search the different ranges for different variances and covariances.

`noestimate` specifies that the model is not to be fit. Instead, starting values are to be shown (as modified by the above options if modifications were made), and they are to be shown using the `coeflegend` style of output.

`dnumerical` specifies that during optimization, the gradient vector and Hessian matrix be computed using numerical techniques instead of analytical formulas. By default, analytical formulas for computing the gradient and Hessian are used for all integration methods except `intmethod(laplace)`.

`coeflegend`; see [R] **estimation options**.

Remarks and examples

For a general introduction to `me` commands, see [ME] **me**. For additional examples of mixed-effects models for binary and binomial outcomes, see [ME] **melogit**, [ME] **meprobit**, and [ME] **mecloglog**. For additional examples of mixed-effects models for ordinal responses, see [ME] **meologit** and [ME] **meoprobit**. For additional examples of mixed-effects models for multinomial outcomes, see [SEM] **example 41g**. For additional examples of mixed-effects models for count outcomes, see [ME] **mepoisson** and [ME] **menbreg**.

Remarks are presented under the following headings:

Introduction

meglm fits multilevel mixed-effects generalized linear models of the form

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

where $\mathbf{y}$ is the $n\times 1$ vector of responses from the distributional family F, $\mathbf{X}$ is an $n\times p$ design/covariate matrix for the fixed effects $\boldsymbol{\beta}$, and $\mathbf{Z}$ is the $n\times q$ design/covariate matrix for the random effects $\mathbf{u}$. The $\mathbf{X}\boldsymbol{\beta}+\mathbf{Z}\mathbf{u}$ part is called the linear predictor, and it is often denoted as $\boldsymbol{\eta}$. The linear predictor also contains the offset or exposure variable when `offset()` or `exposure()` is specified. $g(\cdot)$ is called the link function and is assumed to be invertible such that

$$E(\mathbf{y}|\mathbf{X},\mathbf{u}) = g^{-1}(\mathbf{X}\boldsymbol{\beta}+\mathbf{Z}\mathbf{u}) = H(\boldsymbol{\eta}) = \boldsymbol{\mu}$$

For notational convenience here and throughout this manual entry, we suppress the dependence of $\mathbf{y}$ on $\mathbf{X}$. Substituting various definitions for $g(\cdot)$ and F results in a wide array of models. For instance, if $\mathbf{y}$ is distributed as Gaussian (normal) and $g(\cdot)$ is the identity function, we have

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

or mixed-effects linear regression. If $g(\cdot)$ is the logit function and $\mathbf{y}$ is distributed as Bernoulli, we have

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

or mixed-effects logistic regression. If $g(\cdot)$ is the natural log function and $\mathbf{y}$ is distributed as Poisson, we have

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

or mixed-effects Poisson regression. In fact, some combinations of families and links are so common that we implemented them as separate commands in terms of meglm.

Command	meglm equivalent
melogit	family(bernoulli) link(logit)
meprobit	family(bernoulli) link(probit)
mecloglog	family(bernoulli) link(cloglog)
meologit	family(ordinal) link(logit)
meoprobit	family(ordinal) link(probit)
mepoisson	family(poisson) link(log)
menbreg	family(nbinomial) link(log)

When no family–link combination is specified, meglm defaults to a Gaussian family with an identity link. Thus meglm can be used to fit linear mixed-effects models; however, for those models we recommend using the more specialized mixed, which, in addition to meglm capabilities, accepts frequency and sampling weights and allows for modeling of the structure of the residual errors; see [ME] **mixed** for details.

The random effects $\mathbf{u}$ are assumed to be distributed as multivariate normal with mean $\mathbf{0}$ and $q\times q$ variance matrix $\boldsymbol{\Sigma}$. The random effects are not directly estimated (although they may be predicted), but instead are characterized by the variance components, the elements of $\mathbf{G} = \text{Var}(\mathbf{u})$.

The general forms of the design matrices $\mathbf{X}$ and $\mathbf{Z}$ allow estimation for a broad class of generalized mixed-effects models: blocked designs, split-plot designs, growth curves, multilevel or hierarchical designs, etc. They also allow a flexible method of modeling within-cluster correlation. Subjects within the same cluster can be correlated as a result of a shared random intercept, or through a shared random slope on a covariate, or both. The general specification of variance components also provides additional flexibility—the random intercept and random slope could themselves be modeled as independent, or correlated, or independent with equal variances, and so forth.

Comprehensive treatments of mixed models are provided by, for example, Searle, Casella, and McCulloch (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).

The key to fitting mixed models lies in estimating the variance components, and for that there exist many methods; see, for example, Breslow and Clayton (1993); Lin and Breslow (1996); Bates and Pinheiro (1998); and Ng et al. (2006). `meglm` uses maximum likelihood (ML) to estimate model parameters. The ML estimates are based on the usual application of likelihood theory, given the distributional assumptions of the model.

Returning to (1): in clustered-data situations, it is convenient not to consider all n observations at once but instead to organize the mixed model as a series of M independent groups (or clusters)

$$g\{E(\mathbf{y}_j)\} = \mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j \tag{2}$$

for $j = 1, \dots, M$, with cluster j consisting of n_j observations. The response $\mathbf{y}_j$ comprises the rows of $\mathbf{y}$ corresponding with the jth cluster, with $\mathbf{X}_j$ defined analogously. The random effects $\mathbf{u}_j$ can now be thought of as M realizations of a $q \times 1$ vector that is normally distributed with mean $\mathbf{0}$ and $q \times q$ variance matrix $\boldsymbol{\Sigma}$. The matrix $\mathbf{Z}_i$ is the $n_j \times q$ design matrix for the jth cluster random effects. Relating this to (1), note that

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Z}_M \end{bmatrix}; \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_M \end{bmatrix}; \quad \mathbf{G} = \mathbf{I}_M \otimes \boldsymbol{\Sigma}$$

where $\mathbf{I}_M$ is the $M \times M$ identity matrix and $\otimes$ is the Kronecker product.

The mixed-model formula (2) is from Laird and Ware (1982) and offers two key advantages. First, it makes specifications of random-effects terms easier. If the clusters are schools, you can simply specify a random effect at the school level, as opposed to thinking of what a school-level random effect would mean when all the data are considered as a whole (if it helps, think Kronecker products). Second, representing a mixed-model with (2) generalizes easily to more than one set of random effects. For example, if classes are nested within schools, then (2) can be generalized to allow random effects at both the school and the class-within-school levels.

Two-level models for continuous responses

We begin with a simple application of (2).

▷ Example 1

Consider a longitudinal dataset, used by both Ruppert, Wand, and Carroll (2003) and Diggle et al. (2002), consisting of weight measurements of 48 pigs on 9 successive weeks. Pigs are identified by the variable id. Each pig experiences a linear trend in growth but overall weight measurements vary from pig to pig. Because we are not really interested in these particular 48 pigs per se, we instead treat them as a random sample from a larger population and model the between-pig variability as a random effect, or in the terminology of (2), as a random-intercept term at the pig level. We thus wish to fit the model

$$\texttt{weight}_{ij} = \beta_0 + \beta_1 \texttt{week}_{ij} + u_j + \epsilon_{ij}$$

for $i = 1, \ldots, 9$ weeks and $j = 1, \ldots, 48$ pigs. The fixed portion of the model, $\beta_0 + \beta_1 \texttt{week}_{ij}$, simply states that we want one overall regression line representing the population average. The random effect u_j serves to shift this regression line up or down according to each pig. Because the random effects occur at the pig level (id), we fit the model by typing

```
. use http://www.stata-press.com/data/r13/pig
(Longitudinal analysis of pig weights)
. meglm weight week || id:
Fitting fixed-effects model:
Iteration 0:   log likelihood = -1251.2506
Iteration 1:   log likelihood = -1251.2506
Refining starting values:
Grid node 0:   log likelihood = -1150.6253
Fitting full model:
Iteration 0:   log likelihood = -1150.6253  (not concave)
Iteration 1:   log likelihood = -1036.1793
Iteration 2:   log likelihood =  -1017.912
Iteration 3:   log likelihood = -1014.9537
Iteration 4:   log likelihood = -1014.9268
Iteration 5:   log likelihood = -1014.9268
Mixed-effects GLM                               Number of obs      =       432
Family:                 Gaussian
Link:                   identity
Group variable:               id                Number of groups   =        48
                                                Obs per group: min =         9
                                                               avg =       9.0
                                                               max =         9
Integration method: mvaghermite                 Integration points =         7
                                                Wald chi2(1)       =  25337.48
Log likelihood = -1014.9268                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
      weight |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        week |   6.209896   .0390124   159.18   0.000     6.133433    6.286359
       _cons |   19.35561   .5974047    32.40   0.000     18.18472    20.52651
-------------+----------------------------------------------------------------
id           |
   var(_cons)|   14.81745   3.124202                      9.801687    22.39989
-------------+----------------------------------------------------------------
var(e.weight)|   4.383264   .3163349                      3.805112    5.049261
------------------------------------------------------------------------------
LR test vs. linear regression:   chibar2(01) =   472.65 Prob>=chibar2 = 0.0000
```

At this point, a guided tour of the model specification and output is in order:

1. By typing weight week, we specified the response, weight, and the fixed portion of the model in the same way that we would if we were using regress or any other estimation command. Our fixed effects are a coefficient on week and a constant term.
2. When we added || id:, we specified random effects at the level identified by the group variable id, that is, the pig level (level two). Because we wanted only a random intercept, that is all we had to type.
3. The estimation log displays a set of iterations from optimizing the log likelihood. By default, these are Newton–Raphson iterations, but other methods are available by specifying the appropriate *maximize_options*; see [R] **maximize**.
4. The header describes the model, presents a summary of the random-effects group, reports the integration method used to fit the model, and reports a Wald test against the null hypothesis that all the coefficients on the independent variables in the mean equation are 0. Here the null hypothesis is rejected at all conventional levels. You can suppress the group information with the nogroup or the noheader option, which will suppress the rest of the header as well.
5. The estimation table reports the fixed effects, followed by the random effects, followed by the overall error term.
   a. For the fixed-effects part, we estimate $\beta_0 = 19.36$ and $\beta_1 = 6.21$.
   b. The random-effects equation is labeled id, meaning that these are random effects at the id (pig) level. We have only one random effect at this level, the random intercept. The variance of the level-two errors, σ_u^2, is estimated as 14.82 with standard error 3.12.
   c. The row labeled var(e.weight) displays the estimated variance of the overall error term: $\widehat{\sigma}_\epsilon^2 = 4.38$. This is the variance of the level-one errors, that is, the residuals.
6. Finally, a likelihood-ratio test comparing the model with ordinary linear regression is provided and is highly significant for these data. See *Distribution theory for likelihood-ratio test* in [ME] **me** for a discussion of likelihood-ratio testing of variance components.

◁

See *Remarks and examples* in [ME] **mixed** for further analysis of these data including a random-slope model and a model with an unstructured covariance structure.

Two-level models for nonlinear responses

By specifying different family–link combinations, we can fit a variety of mixed-effects models for nonlinear responses. Here we replicate the model from example 2 of meqrlogit.

▷ Example 2

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. 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.

We fit a standard logistic regression model, amended to have a random intercept for each district and a random slope on the indicator variable urban. We fit the model by typing

```
. use http://www.stata-press.com/data/r13/bangladesh
(Bangladesh Fertility Survey, 1989)
. meglm c_use urban age child* || district: urban, family(bernoulli) link(logit)
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 = -1215.8592
Fitting full model:
Iteration 0:   log likelihood = -1215.8592  (not concave)
Iteration 1:   log likelihood = -1209.6285
Iteration 2:   log likelihood = -1205.7903
Iteration 3:   log likelihood = -1205.1337
Iteration 4:   log likelihood = -1205.0034
Iteration 5:   log likelihood = -1205.0025
Iteration 6:   log likelihood = -1205.0025
Mixed-effects GLM                               Number of obs      =      1934
Family:                Bernoulli
Link:                      logit
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)       =     97.30
Log likelihood = -1205.0025                     Prob > chi2        =    0.0000
```

c_use	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
urban	.7143927	.1513595	4.72	0.000	.4177335	1.011052
age	-.0262261	.0079656	-3.29	0.001	-.0418384	-.0106138
child1	1.128973	.1599347	7.06	0.000	.815507	1.442439
child2	1.363165	.1761804	7.74	0.000	1.017857	1.708472
child3	1.352238	.1815608	7.45	0.000	.9963853	1.708091
_cons	-1.698137	.1505019	-11.28	0.000	-1.993115	-1.403159
district						
var(urban)	.2741013	.2131525			.059701	1.258463
var(_cons)	.2390807	.0857012			.1184191	.4826891

```
LR test vs. logistic regression:     chi2(2) =    47.05   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

Because we did not specify a covariance structure for the random effects $(u_{1j}, u_{0j})'$, `meglm` used the default independent structure:

$$\boldsymbol{\Sigma} = \mathrm{Var}\begin{bmatrix} u_{1j} \\ u_{0j} \end{bmatrix} = \begin{bmatrix} \sigma^2_{u1} & 0 \\ 0 & \sigma^2_{u0} \end{bmatrix}$$

with $\widehat{\sigma}^2_{u1} = 0.27$ and $\widehat{\sigma}^2_{u0} = 0.24$. You can request a different covariance structure by specifying the `covariance()` option. See *Two-level models* in [ME] **meqrlogit** for further analysis of these data, and see [ME] **me** and [ME] **mixed** for further examples of covariance structures.

◁

Three-level models for nonlinear responses

Two-level models extend naturally to models with three or more levels with nested random effects. Here we replicate the model from example 2 of [ME] **meologit**.

▷ Example 3

We use the data from the Television, School, and Family Smoking Prevention and Cessation Project (Flay et al. 1988; Rabe-Hesketh and Skrondal 2012, chap. 11), where schools were randomly assigned into one of four groups defined by two treatment variables. Students within each school are nested in classes, and classes are nested in schools. The dependent variable is the tobacco and health knowledge (THK) scale score collapsed into four ordered categories. We regress the outcome on the treatment variables, social resistance classroom curriculum and TV intervention, and their interaction and control for the pretreatment score.

```
. use http://www.stata-press.com/data/r13/tvsfpors
. meglm thk prethk cc##tv || school: || class:, family(ordinal) link(logit)
Fitting fixed-effects model:
Iteration 0:   log likelihood =  -2212.775
Iteration 1:   log likelihood =  -2125.509
Iteration 2:   log likelihood = -2125.1034
Iteration 3:   log likelihood = -2125.1032
Refining starting values:
Grid node 0:   log likelihood = -2152.1514
Fitting full model:
Iteration 0:   log likelihood = -2152.1514  (not concave)
Iteration 1:   log likelihood = -2125.9213  (not concave)
Iteration 2:   log likelihood = -2120.1861
Iteration 3:   log likelihood = -2115.6177
Iteration 4:   log likelihood = -2114.5896
Iteration 5:   log likelihood = -2114.5881
Iteration 6:   log likelihood = -2114.5881
Mixed-effects GLM                               Number of obs      =      1600
Family:                   ordinal
Link:                       logit
```

Group Variable	No. of Groups	Observations per Group Minimum	Average	Maximum
school	28	18	57.1	137
class	135	1	11.9	28

```
Integration method: mvaghermite                 Integration points =          7

                                                Wald chi2(4)       =     124.39
Log likelihood = -2114.5881                     Prob > chi2        =     0.0000

------------------------------------------------------------------------------
         thk |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      prethk |   .4085273    .039616    10.31   0.000     .3308814    .4861731
        1.cc |   .8844369   .2099124     4.21   0.000     .4730161    1.295858
        1.tv |    .236448   .2049065     1.15   0.249    -.1651614    .6380575
             |
       cc#tv |
         1 1 |  -.3717699   .2958887    -1.26   0.209     -.951701    .2081612
-------------+----------------------------------------------------------------
       /cut1 |  -.0959459   .1688988    -0.57   0.570    -.4269815    .2350896
       /cut2 |   1.177478   .1704946     6.91   0.000     .8433151    1.511642
       /cut3 |   2.383672   .1786736    13.34   0.000     2.033478    2.733865
-------------+----------------------------------------------------------------
school       |
   var(_cons)|   .0448735   .0425387                      .0069997    .2876749
-------------+----------------------------------------------------------------
school>class |
   var(_cons)|   .1482157   .0637521                       .063792    .3443674
------------------------------------------------------------------------------
LR test vs. ologit regression:       chi2(2) =    21.03   Prob > chi2 = 0.0000

Note: LR test is conservative and provided only for reference.
```

Notes:

1. Our model now has two random-effects equations, separated by `||`. The first is a random intercept (constant only) at the `school` level (level three), and the second is a random intercept at the `class` level (level two). The order in which these are specified (from left to right) is significant—`meglm` assumes that `class` is nested within `school`.

2. The information on groups is now displayed as a table, with one row for each grouping. You can suppress this table with the `nogroup` or the `noheader` option, which will suppress the rest of the header, as well.

3. The variance-component estimates are now organized and labeled according to level. The variance component for `class` is labeled `school>class` to emphasize that classes are nested within schools.

We refer you to example 2 of [ME] **meologit** and example 1 of [ME] **meologit postestimation** for a substantive interpretation of the results.

◁

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).

Crossed-effects models

Not all mixed models contain nested levels of random effects. In this section, we consider a crossed-effects model, that is, a mixed-effects model in which the levels of random effects are not nested; see [ME] **me** for more information on crossed-effects models.

▷ Example 4

We use the salamander cross-breeding data from Karim and Zeger (1992) as analyzed in Rabe-Hesketh and Skrondal (2012, chap. 16.10). The salamanders come from two populations—whiteside and roughbutt—and are labeled whiteside males (`wsm`), whiteside females (`wsf`), roughbutt males (`rbm`), and roughbutt females (`rbf`). Male identifiers are recorded in the variable `male`, and female identifiers are recorded in the variable `female`. The salamanders were divided into groups such that each group contained 60 male–female pairs, with each salamander having three potential partners from the same population and three potential partners from the other population. The outcome (`y`) is coded 1 if there was a successful mating and is coded 0 otherwise; see the references for a detailed description of the mating experiment.

We fit a crossed-effects logistic regression for successful mating, where each male has the same value of his random intercept across all females, and each female has the same value of her random intercept across all males.

To fit a crossed-effects model in Stata, we use the `_all: R.`*varname* syntax. We treat the entire dataset as one super cluster, denoted `_all`, and we nest each gender within the super cluster by using the `R.`*varname* notation. `R.male` requests a random intercept for each level of `male` and imposes an identity covariance structure on the random effects; that is, the variances of the random intercepts are restricted to be equal for all male salamanders. `R.female` accomplishes the same for the female salamanders. In Stata, we type

```
. use http://www.stata-press.com/data/r13/salamander
. meglm y wsm##wsf || _all: R.male || _all: R.female, family(bernoulli)
> link(logit) or
note: crossed random effects model specified; option intmethod(laplace)
implied
Fitting fixed-effects model:
Iteration 0:   log likelihood = -223.13998
Iteration 1:   log likelihood = -222.78752
Iteration 2:   log likelihood = -222.78735
Iteration 3:   log likelihood = -222.78735
Refining starting values:
Grid node 0:   log likelihood = -211.58149
Fitting full model:
Iteration 0:   log likelihood = -211.58149
Iteration 1:   log likelihood = -209.32221
Iteration 2:   log likelihood = -209.31084
Iteration 3:   log likelihood = -209.27663
Iteration 4:   log likelihood = -209.27659
Iteration 5:   log likelihood = -209.27659  (backed up)
Mixed-effects GLM                               Number of obs      =       360
Family:                 Bernoulli
Link:                       logit
Group variable:              _all               Number of groups   =         1
                                                Obs per group: min =       360
                                                               avg =     360.0
                                                               max =       360
Integration method:       laplace
                                                Wald chi2(3)       =     37.54
Log likelihood = -209.27659                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
           y | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       1.wsm |   .4956747   .2146564    -1.62   0.105     .2121174     1.15829
       1.wsf |   .0548105   .0300131    -5.30   0.000     .0187397    .1603114
             |
     wsm#wsf |
        1 1  |   36.17082   22.77918     5.70   0.000     10.52689    124.2844
             |
       _cons |   2.740043   .9768565     2.83   0.005     1.362368    5.510873
-------------+----------------------------------------------------------------
_all>male    |
   var(_cons)|    1.04091    .511001                      .3976885    2.724476
-------------+----------------------------------------------------------------
_all>female  |
   var(_cons)|   1.174448   .5420751                      .4752865    2.902098
------------------------------------------------------------------------------
LR test vs. logistic regression:     chi2(2) =    27.02   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

Because we specified a crossed-effects model, `meglm` defaulted to the method of Laplacian approximation to calculate the likelihood; see *Computation time and the Laplacian approximation* in [ME] **me** for a discussion of computational complexity of mixed-effects models, and see *Methods and formulas* below for the formulas used by the Laplacian approximation method.

The estimates of the random intercepts suggest that the heterogeneity among the female salamanders, 1.17, is larger than the heterogeneity among the male salamanders, 1.04.

Setting both random intercepts to 0, the odds of successful mating for a roughbutt male–female pair are given by the estimate of _cons, 2.74. Rabe-Hesketh and Skrondal (2012, chap. 16.10) show how to calculate the odds ratios for the other three salamander pairings.

◁

The R.*varname* notation is equivalent to giving a list of overparameterized (none dropped) indicator variables for use in a random-effects specification. When you specify R.*varname*, meglm handles the calculations internally rather than creating the indicators in the data. Because the set of indicators is overparameterized, R.*varname* implies noconstant. You can include factor variables in the fixed-effects specification by using standard methods; see [U] **11.4.3 Factor variables**. However, random-effects equations support only the R.*varname* factor specification. For more complex factor specifications (such as interactions) in random-effects equations, use generate to form the variables manually.

❑ Technical note

We fit the salamander model by using

```
. meglm y wsm##wsf || _all: R.male || _all: R.female ...
```

as a direct way to demonstrate the R. notation. However, we can technically treat female salamanders as nested within the _all group, yielding the equivalent way to fit the model:

```
. meglm y wsm##wsf || _all: R.male || female: ...
```

We leave it to you to verify that both produce identical results. As we note in example 8 of [ME] **me**, the latter specification, organized at the cluster (female) level with random-effects dimension one (a random intercept) is, in general, much more computationally efficient.

❑

Obtaining better starting values

Given the flexibility of mixed-effects models, you will find that some models "fail to converge" when used with your data; see *Diagnosing convergence problems* in [ME] **me** for details. What we say below applies regardless of how the convergence problem revealed itself. You might have seen the error message "initial values not feasible" or some other error message, or you might have an infinite iteration log.

meglm provides two options to help you obtain better starting values: startvalues() and startgrid().

startvalues(*svmethod*) allows you to specify one of four starting-value calculation methods: zero, constantonly, fixedonly, or iv. By default, meglm uses startvalues(fixedonly). Evidently, that did not work for you. Try the other methods, starting with startvalues(iv):

```
. meglm ..., ... startvalues(iv)
```

If that does not solve the problem, proceed through the others.

By the way, if you have starting values for some parameters but not others—perhaps you fit a simplified model to get them—you can combine the options startvalues() and from():

```
. meglm ..., ...                          // simplified model
. matrix b = e(b)
. meglm ..., ... from(b) startvalues(iv)  // full model
```

The other special option meglm provides is startgrid(), which can be used with or without startvalues(). startgrid() is a brute-force approach that tries various values for variances and covariances and chooses the ones that work best.

1. You may already be using a default form of startgrid() without knowing it. If you see meglm displaying Grid node 1, Grid node 2, ... following Grid node 0 in the iteration log, that is meglm doing a default search because the original starting values were not feasible.

   The default form tries 0.1, 1, and 10 for all variances of all random effects and, if applicable, for the residual variance.

2. startgrid(*numlist*) specifies values to try for variances of random effects.

3. startgrid(*covspec*) specifies the particular variances and covariances in which grid searches are to be performed. Variances and covariances are specified in the usual way. startgrid(_cons[id] x[id] _cons[id]*x[id]) specifies that 0.1, 1, and 10 be tried for each member of the list.

4. startgrid(*numlist covspec*) combines the two syntaxes. You can specify startgrid() multiple times so that you can search the different ranges for different variances and covariances.

Our advice to you is the following:

1. If you receive an iteration log and it does not contain Grid node 1, Grid node 2, ..., then specify startgrid(.1 1 10). Do that whether the iteration log was infinite or ended with some other error. In this case, we know that meglm did not run startgrid() on its own because it did not report Grid node 1, Grid node 2, etc. Your problem is poor starting values, not infeasible ones.

   A synonym for startgrid(.1 1 10) is just startgrid without parentheses.

   Be careful, however, if you have many random effects. Specifying startgrid() could run a long time because it runs all possible combinations. If you have 10 random effects, that means $10^3 = 1{,}000$ likelihood evaluations.

   If you have many random effects, rerun your difficult meglm command including option iterate(*#*) and look at the results. Identify the problematic variances and search across them only. Do not just look for variances going to 0. Variances getting really big can be a problem, too, and even reasonable values can be a problem. Use your knowledge and intuition about the model.

   Perhaps you will try to fit your model by specifying startgrid(.1 1 10 _cons[id] x[id] _cons[id]*x[id]).

   Values 0.1, 1, and 10 are the default. Equivalent to specifying startgrid(.1 1 10 _cons[id] x[id] _cons[id]*x[id]) is startgrid(_cons[id] x[id] _cons[id]*x[id]).

   Look at covariances as well as variances. If you expect a covariance to be negative but it is positive, then try negative starting values for the covariance by specifying startgrid(-.1 -1 -10 _cons[id]*x[id]).

   Remember that you can specify startgrid() multiple times. Thus you might specify both startgrid(_cons[id] x[id]) and startgrid(-.1 -1 -10 _cons[id]*x[id]).

2. If you receive the message "initial values not feasible", you know that meglm already tried the default startgrid().

   The default startgrid() only tried the values 0.1, 1, and 10, and only tried them on the variances of random effects. You may need to try different values or try the same values on covariances or variances of errors of observed endogenous variables.

   We suggest you first rerun the model causing difficulty and include the noestimate option. If, looking at the results, you have an idea of which variance or covariance is a problem, or if you have few variances and covariances, we would recommend running startgrid() first. On the other hand, if you have no idea as to which variance or covariance is the problem and you have many of them, you will be better off if you first simplify the model. After doing that, if your simplified model does not include all the variances and covariances, you can specify a combination of from() and startgrid().

Stored results

meglm 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_cat)	number of categories (with ordinal outcomes)
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(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(N_clust)	number of clusters
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(cmdline)	command as typed
e(depvar)	name of dependent variable
e(covariates)	list of covariates
e(ivars)	grouping variables
e(model)	name of marginal model
e(title)	title in estimation output
e(link)	link
e(family)	family
e(clustvar)	name of cluster variable
e(offset)	offset
e(binomial)	binomial number of trials (with binomial models)
e(dispersion)	mean or constant (with negative binomial models)
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(cat)	category values (with ordinal outcomes)
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

Methods and formulas are presented under the following headings:

Introduction

Without a loss of generality, consider a two-level generalized mixed-effects model

$$E(\mathbf{y}_j|\mathbf{X}_j,\mathbf{u}_j) = g^{-1}\big(\mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j\big), \qquad \mathbf{y} \sim F$$

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 $n_j \times 1$ 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, $\boldsymbol{\beta}$ is the $p \times 1$ vector of regression coefficients on the fixed predictors, and we use $\boldsymbol{\Sigma}$ to denote the unknown $q \times q$ variance matrix of the random effects. For simplicity, we consider a model with no auxiliary parameters.

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 or the exposure variable when `offset()` or `exposure()` is specified. Let y_{ij} and η_{ij} be the ith individual elements of $\mathbf{y}_j$ and $\boldsymbol{\eta}_j$, $i = 1, \ldots, n_j$. Let $f(y_{ij}|\eta_{ij})$ be the conditional density function for the response at observation i. Because the observations are assumed to be conditionally independent, we can overload the definition of $f(\cdot)$ with vector inputs to mean

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

The random effects $\mathbf{u}_j$ are assumed to be multivariate normal with mean $\mathbf{0}$ and variance $\boldsymbol{\Sigma}$. The likelihood function for cluster j is given by

$$\begin{aligned}\mathcal{L}_j(\boldsymbol{\beta},\boldsymbol{\Sigma}) &= (2\pi)^{-q/2}|\boldsymbol{\Sigma}|^{-1/2}\int_{\Re^q} f(\mathbf{y}_j|\boldsymbol{\eta}_j)\exp\left(-\frac{1}{2}\mathbf{u}_j'\boldsymbol{\Sigma}^{-1}\mathbf{u}_j\right) d\mathbf{u}_j \\ &= (2\pi)^{-q/2}|\boldsymbol{\Sigma}|^{-1/2}\int_{\Re^q} \exp\left\{\log f(\mathbf{y}_j|\boldsymbol{\eta}_j) - \frac{1}{2}\mathbf{u}_j'\boldsymbol{\Sigma}^{-1}\mathbf{u}_j\right\} d\mathbf{u}_j\end{aligned} \tag{3}$$

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

The multivariate integral in (3) is generally not tractable, so we must use numerical methods to approximate the integral. We can use a change-of-variables technique to transform this multivariate integral into a set of nested univariate integrals. Each univariate integral can then be evaluated using a form of Gaussian quadrature. `meglm` supports three types of Gauss–Hermite quadratures: mean–variance adaptive Gauss–Hermite quadrature (MVAGH), mode-curvature adaptive Gauss–Hermite quadrature (MCAGH), and nonadaptive Gauss–Hermite quadrature (GHQ). `meglm` also offers the Laplacian-approximation method, which is used as a default method for crossed mixed-effects models. Below we describe the four methods. The methods described below are based on Skrondal and Rabe-Hesketh (2004, chap. 6.3).

Gauss–Hermite quadrature

Let $\mathbf{u}_j = \mathbf{L}\mathbf{v}_j$, where $\mathbf{v}_j$ is a $q \times 1$ random vector whose elements are independently standard normal variables and $\mathbf{L}$ is the Cholesky decomposition of $\boldsymbol{\Sigma}$, $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}'$. Then $\boldsymbol{\eta}_j = \mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{L}\mathbf{v}_j$, and the likelihood in (3) becomes

$$\begin{aligned}\mathcal{L}_j(\boldsymbol{\beta},\boldsymbol{\Sigma}) &= (2\pi)^{-q/2}\int_{\Re^q} \exp\left\{\log f(\mathbf{y}_j|\boldsymbol{\eta}_j) - \frac{1}{2}\mathbf{v}_j'\mathbf{v}_j\right\} d\mathbf{v}_j \\ &= (2\pi)^{-q/2}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} \exp\left\{\log f(\mathbf{y}_j|\boldsymbol{\eta}_j) - \frac{1}{2}\sum_{k=1}^{q} v_{jk}^2\right\} d\mathbf{v}_{j1},\ldots,d\mathbf{v}_{jq}\end{aligned} \tag{4}$$

Consider a q-dimensional quadrature grid containing r quadrature points in each dimension. Let $\mathbf{a_k} = (a_{k_1}, \ldots, a_{k_q})'$ be a point on this grid, and let $\mathbf{w_k} = (w_{k_1}, \ldots, w_{k_q})'$ be the vector of corresponding weights. The GHQ approximation to the likelihood is

$$\begin{aligned}\mathcal{L}_j^{\mathrm{GHQ}}(\boldsymbol\beta, \boldsymbol\Sigma) &= \sum_{k_1=1}^{r} \ldots \sum_{k_q=1}^{r} \left[\exp\left\{\log f(\mathbf{y}_j|\boldsymbol\eta_{j\mathbf{k}})\right\} \prod_{p=1}^{q} w_{k_p} \right] \\ &= \sum_{k_1=1}^{r} \ldots \sum_{k_q=1}^{r} \left[\exp\left\{\sum_{i=1}^{n_j} \log f(y_{ij}|\eta_{ij\mathbf{k}})\right\} \prod_{p=1}^{q} w_{k_p} \right]\end{aligned}$$

where

$$\boldsymbol\eta_{j\mathbf{k}} = \mathbf{X}_j\boldsymbol\beta + \mathbf{Z}_j\mathbf{L}\mathbf{a_k}$$

and $\eta_{ij\mathbf{k}}$ is the ith element of $\boldsymbol\eta_{j\mathbf{k}}$.

Adaptive Gauss–Hermite quadrature

This section sets the stage for MVAGH quadrature and MCAGH quadrature.

Let's reconsider the likelihood in (4). We use $\phi(\mathbf{v}_j)$ to denote a multivariate standard normal with mean $\mathbf{0}$ and variance $\mathbf{I}_q$, and we use $\phi(\mathbf{v}_j|\boldsymbol\mu_j, \boldsymbol\Lambda_j)$ to denote a multivariate normal with mean $\boldsymbol\mu_j$ and variance $\boldsymbol\Lambda_j$.

For fixed model parameters, the posterior density for $\mathbf{v}_j$ is proportional to

$$\phi(\mathbf{v}_j) f(\mathbf{y}_j|\boldsymbol\eta_j)$$

where

$$\boldsymbol\eta_j = \mathbf{X}_j\boldsymbol\beta + \mathbf{Z}_j\mathbf{L}\mathbf{v}_j$$

It is reasonable to assume that this posterior density can be approximated by a multivariate normal density with mean vector $\boldsymbol\mu_j$ and variance matrix $\boldsymbol\Lambda_j$. Instead of using the prior density of $\mathbf{v}_j$ as the weighting distribution in the integral, we can use our approximation for the posterior density,

$$\mathcal{L}_j(\boldsymbol\beta, \boldsymbol\Sigma) = \int_{\Re^q} \frac{f(\mathbf{y}_j|\boldsymbol\eta_j)\phi(\mathbf{v}_j)}{\phi(\mathbf{v}_j|\boldsymbol\mu_j, \boldsymbol\Lambda_j)} \phi(\mathbf{v}_j|\boldsymbol\mu_j, \boldsymbol\Lambda_j)\, d\mathbf{v}_j$$

Then the MVAGH approximation to the likelihood is

$$\mathcal{L}_j^{\mathrm{MVAGH}}(\boldsymbol\beta, \boldsymbol\Sigma) = \sum_{k_1=1}^{r} \ldots \sum_{k_q=1}^{r} \left[\exp\left\{\log f(\mathbf{y}_j|\boldsymbol\eta_{j\mathbf{k}})\right\} \prod_{p=1}^{q} w^*_{jk_p} \right]$$

where

$$\boldsymbol\eta_{j\mathbf{k}} = \mathbf{X}_j\boldsymbol\beta + \mathbf{Z}_j\mathbf{L}\mathbf{a}^*_{j\mathbf{k}}$$

and $\mathbf{a}^*_{j\mathbf{k}}$ and $w^*_{jk_p}$ are the abscissas and weights after an orthogonalizing transformation of $\mathbf{a}_{j\mathbf{k}}$ and w_{jk_p}, respectively, which eliminates posterior covariances between the random effects.

Estimates of $\boldsymbol{\mu}_j$ and $\boldsymbol{\Lambda}_j$ are computed using one of two different methods. The mean $\boldsymbol{\mu}_j$ and variance $\boldsymbol{\Lambda}_j$ are computed iteratively by updating the posterior moments with the MVAGH approximation, starting with a $\mathbf{0}$ mean vector and identity variance matrix. For the MCAGH approximation, $\boldsymbol{\mu}_j$ and $\boldsymbol{\Lambda}_j$ are computed by optimizing the integrand with respect to $\mathbf{v}_j$, where $\boldsymbol{\mu}_j$ is the optimal value and $\boldsymbol{\Lambda}_j$ is the curvature at $\boldsymbol{\mu}_j$.

Laplacian approximation

Consider the likelihood in (3) and denote the argument in the exponential function by

$$h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j) = \log f(\mathbf{y}_j|\mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j) - \frac{1}{2}\mathbf{u}_j'\boldsymbol{\Sigma}^{-1}\mathbf{u}_j$$

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. The first and second partial derivatives with respect to $\mathbf{u}_j$ are

$$h'(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j) = \frac{\partial h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j)}{\partial \mathbf{u}_j} = \mathbf{Z}_j' \frac{\partial \log f(\mathbf{y}_j|\boldsymbol{\eta}_j)}{\partial \boldsymbol{\eta}_j} - \boldsymbol{\Sigma}^{-1}\mathbf{u}_j$$

$$h''(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j) = \frac{\partial^2 h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j)}{\partial \mathbf{u}_j \partial \mathbf{u}_j'} = \mathbf{Z}_j' \frac{\partial^2 \log f(\mathbf{y}_j|\boldsymbol{\eta}_j)}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\eta}_j'} \mathbf{Z}_j - \boldsymbol{\Sigma}^{-1}$$

The maximizer of $h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j)$ is $\widehat{\mathbf{u}}_j$ such that $h'(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j) = \mathbf{0}$. The integral in (3) is proportional to the posterior density of $\mathbf{u}_j$ given the data, so $\widehat{\mathbf{u}}_j$ is also the posterior mode.

Let

$$\widehat{\mathbf{p}}_j = \mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\widehat{\mathbf{u}}_j$$

$$\mathbf{S}_1 = \frac{\partial \log f(\mathbf{y}_j|\widehat{\mathbf{p}}_j)}{\partial \widehat{\mathbf{p}}_j}$$

$$\mathbf{S}_2 = \frac{\partial \mathbf{S}_1}{\partial \widehat{\mathbf{p}}_j'} = \frac{\partial^2 \log f(\mathbf{y}_j|\widehat{\mathbf{p}}_j)}{\partial \widehat{\mathbf{p}}_j \partial \widehat{\mathbf{p}}_j'}$$

$$\mathbf{H}_j = h''(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j) = \mathbf{Z}_j'\mathbf{S}_2\mathbf{Z}_j - \boldsymbol{\Sigma}^{-1}$$

then

$$\mathbf{0} = h'(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j) = \mathbf{Z}_j'\mathbf{S}_1 - \boldsymbol{\Sigma}^{-1}\widehat{\mathbf{u}}_j$$

Given the above, the second-order Taylor approximation takes the form

$$h(\boldsymbol{\beta},\boldsymbol{\Sigma},\mathbf{u}_j) \approx h(\boldsymbol{\beta},\boldsymbol{\Sigma},\widehat{\mathbf{u}}_j) + \frac{1}{2}(\mathbf{u}_j - \widehat{\mathbf{u}}_j)'\mathbf{H}_j(\mathbf{u}_j - \widehat{\mathbf{u}}_j)$$

because the first-order derivative term is 0. The integral is approximated by

$$\int_{\Re^q} \exp\{h(\boldsymbol{\beta},\boldsymbol{\Sigma},\mathbf{u}_j)\}\, d\mathbf{u}_j \approx (2\pi)^{q/2}\, |{-\mathbf{H}_j}|^{-1/2} \exp\{h(\boldsymbol{\beta},\boldsymbol{\Sigma},\widehat{\mathbf{u}}_j)\}$$

Thus the Laplacian approximated log likelihood is

$$\log \mathcal{L}_j^{\text{Lap}}(\boldsymbol{\beta},\boldsymbol{\Sigma}) = -\frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\log|{-\mathbf{H}_j}| + h(\boldsymbol{\beta},\boldsymbol{\Sigma},\widehat{\mathbf{u}}_j)$$

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)`. In the presence of auxiliary parameters, their estimates and standard errors are included in `e(b)` and `e(V)`, respectively.

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.

Bates, D. M., and J. C. Pinheiro. 1998. Computational methods for multilevel modelling. In *Technical Memorandum BL0112140-980226-01TM*. Murray Hill, NJ: Bell Labs, Lucent Technologies. http://stat.bell-labs.com/NLME/CompMulti.pdf.

Breslow, N. E., and D. G. Clayton. 1993. Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association* 88: 9–25.

Cameron, A. C., and P. K. Trivedi. 2010. *Microeconometrics Using Stata*. Rev. ed. College Station, TX: Stata Press.

Canette, I. 2011. Including covariates in crossed-effects models. The Stata Blog: Not Elsewhere Classified. http://blog.stata.com/2010/12/22/including-covariates-in-crossed-effects-models/.

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

Diggle, P. J., P. J. Heagerty, K.-Y. Liang, and S. L. Zeger. 2002. *Analysis of Longitudinal Data*. 2nd ed. Oxford: Oxford University Press.

Flay, B. R., B. R. Brannon, C. A. Johnson, W. B. Hansen, A. L. Ulene, D. A. Whitney-Saltiel, L. R. Gleason, S. Sussman, M. D. Gavin, K. M. Glowacz, D. F. Sobol, and D. C. Spiegel. 1988. The television, school, and family smoking cessation and prevention project: I. Theoretical basis and program development. *Preventive Medicine* 17: 585–607.

Harville, D. A. 1977. Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association* 72: 320–338.

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

Hocking, R. R. 1985. *The Analysis of Linear Models*. Monterey, CA: Brooks/Cole.

Horton, N. J. 2011. Stata tip 95: Estimation of error covariances in a linear model. *Stata Journal* 11: 145–148.

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

Karim, M. R., and S. L. Zeger. 1992. Generalized linear models with random effects; salamander mating revisited. *Biometrics* 48: 631–644.

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

LaMotte, L. R. 1973. Quadratic estimation of variance components. *Biometrics* 29: 311–330.

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.

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.

Nichols, A. 2007. Causal inference with observational data. *Stata Journal* 7: 507–541.

Pantazis, N., and G. Touloumi. 2010. Analyzing longitudinal data in the presence of informative drop-out: The jmre1 command. *Stata Journal* 10: 226–251.

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

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

Ruppert, D., M. P. Wand, and R. J. Carroll. 2003. *Semiparametric Regression*. Cambridge: Cambridge University Press.

Searle, S. R. 1989. Obituary: Charles Roy Henderson 1911–1989. *Biometrics* 45: 1333–1335.

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

Skrondal, A., and S. Rabe-Hesketh. 2004. *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models*. Boca Raton, FL: Chapman & Hall/CRC.

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

Also see

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

[ME] **mixed** — Multilevel mixed-effects linear regression

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

[R] **glm** — Generalized linear models

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

[U] 20 Estimation and postestimation commands

Title

meglm postestimation — Postestimation tools for meglm

Description

The following postestimation command is of special interest after `meglm`:

Command	Description
`estat group`	summarize the composition of the nested groups

The following standard postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat ic`	Akaike's and Schwarz's Bayesian information criteria (AIC and BIC)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

Special-interest postestimation commands

`estat group` reports the number of groups and minimum, average, and maximum group sizes for each level of the model. Model levels are identified by the corresponding group variable in the data. Because groups are treated as nested, the information in this summary may differ from what you would get if you used the `tabulate` command on each group variable individually.

Syntax for predict

Syntax for obtaining predictions of random effects and their standard errors

predict [*type*] *newvarsspec* [*if*] [*in*], {remeans | remodes} [reses(*newvarsspec*)]

Syntax for obtaining other predictions

predict [*type*] *newvarsspec* [*if*] [*in*] [, *statistic* *options*]

newvarsspec is *stub** or *newvarlist*.

statistic	Description
Main	
mu	mean response; the default
pr	synonym for mu for ordinal and binary response models
fitted	fitted linear predictor
xb	linear predictor for the fixed portion of the model only
stdp	standard error of the fixed-portion linear prediction
residuals	raw residuals; available only with the Gaussian family
pearson	Pearson residuals
deviance	deviance residuals
anscombe	Anscombe residuals

These statistics are available both in and out of sample; type predict ... if e(sample) ... if wanted only for the estimation sample.

options	Description
Main	
`means`	compute *statistic* using empirical Bayes means; the default
`modes`	compute *statistic* using empirical Bayes modes
`nooffset`	ignore the offset or exposure variable in calculating predictions; relevant only if you specified `offset()` or `exposure()` when you fit the model
`fixedonly`	prediction for the fixed portion of the model only
`outcome(`*outcome*`)`	outcome category for predicted probabilities for ordinal models
Integration	
`intpoints(`#`)`	use # quadrature points to compute empirical Bayes means
`iterate(`#`)`	set maximum number of iterations in computing statistics involving empirical Bayes estimators
`tolerance(`#`)`	set convergence tolerance for computing statistics involving empirical Bayes estimators

For ordinal outcomes, you specify one or *k* new variables in *newvarlist* with `mu` and `pr`, where *k* is the number of outcomes. If you do not specify `outcome()`, those options assume `outcome(#1)`.

Menu for predict

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

`remeans` calculates posterior means of the random effects, also known as empirical Bayes means. You must specify *q* new variables, where *q* is the number of random-effects terms in the model. However, it is much easier to just specify *stub*`*` and let Stata name the variables *stub*`1`, *stub*`2`, ..., *stubq* for you.

`remodes` calculates posterior modes of the random effects, also known as empirical Bayes modes. You must specify *q* new variables, where *q* is the number of random-effects terms in the model. However, it is much easier to just specify *stub*`*` and let Stata name the variables *stub*`1`, *stub*`2`, ..., *stubq* for you.

`reses(`*stub*`*` | *newvarlist*`)` calculates standard errors of the empirical Bayes estimators and stores the result in *newvarlist*. This option requires the `remeans` or the `remodes` option. You must specify *q* new variables, where *q* is the number of random-effects terms in the model. However, it is much easier to just specify *stub*`*` and let Stata name the variables *stub*`1`, *stub*`2`, ..., *stubq* for you.

The `remeans`, `remodes`, and `reses()` options often generate multiple new variables at once. When this occurs, the random effects (and standard errors) contained in the generated variables correspond to the order in which the variance components are listed in the output of `meglm`. Still, examining the variable labels of the generated variables (by using the `describe` command, for instance) can be useful in deciphering which variables correspond to which terms in the model.

`mu`, the default, calculates the predicted mean, that is, the inverse link function applied to the linear prediction. By default, this is based on a linear predictor that includes both the fixed effects and the random effects, and the predicted mean is conditional on the values of the random effects. Use the `fixedonly` option if you want predictions that include only the fixed portion of the model, that is, if you want random effects set to 0.

`pr` calculates predicted probabilities and is a synonym for `mu`. This option is available only for ordinal and binary response models.

`fitted` calculates the fitted linear prediction. By default, the fitted predictor includes both the fixed effects and the random effects. Use the `fixedonly` option if you want predictions that include only the fixed portion of the model, that is, if you want random effects set to 0.

`xb` calculates the linear prediction $\mathbf{x}\boldsymbol{\beta}$ based on the estimated fixed effects (coefficients) in the model. This is equivalent to fixing all random effects in the model to their theoretical (prior) mean value of 0.

`stdp` calculates the standard error of the fixed-effects linear predictor $\mathbf{x}\boldsymbol{\beta}$.

`residuals` calculates raw residuals, that is, responses minus the fitted values. This option is available only for the Gaussian family.

`pearson` calculates Pearson residuals. Pearson residuals that are large in absolute value may indicate a lack of fit. By default, residuals include both the fixed portion and the random portion of the model. The `fixedonly` option modifies the calculation to include the fixed portion only.

`deviance` calculates deviance residuals. Deviance residuals are recommended by McCullagh and Nelder (1989) as having the best properties for examining the goodness of fit of a GLM. They are approximately normally distributed if the model is correctly specified. They may be plotted against the fitted values or against a covariate to inspect the model fit. By default, residuals include both the fixed portion and the random portion of the model. The `fixedonly` option modifies the calculation to include the fixed portion only.

`anscombe` calculates Anscombe residuals, which are designed to closely follow a normal distribution. By default, residuals include both the fixed portion and the random portion of the model. The `fixedonly` option modifies the calculation to include the fixed portion only.

`means` specifies that posterior means be used as the estimates of the random effects for any *statistic* involving random effects. `means` is the default.

`modes` specifies that posterior modes be used as the estimates of the random effects for any *statistic* involving random effects.

`nooffset` is relevant only if you specified `offset(`$varname_o$`)` or `exposure(`$varname_e$`)` with `meglm`. It modifies the calculations made by `predict` so that they ignore the offset or the exposure variable; the linear prediction is treated as $\mathbf{X}\beta + \mathbf{Zu}$ rather than $\mathbf{X}\beta + \mathbf{Zu} + \text{offset}$, or $\mathbf{X}\beta + \mathbf{Zu} + \ln(\text{exposure})$, whichever is relevant.

`fixedonly` modifies predictions to include only the fixed portion of the model, equivalent to setting all random effects equal to 0.

`outcome(`*outcome*`)` specifies the outcome for which the predicted probabilities are to be calculated. `outcome()` should contain either one value of the dependent variable or one of `#1`, `#2`, ..., with `#1` meaning the first category of the dependent variable, `#2` meaning the second category, etc.

Integration

`intpoints(`*#*`)` specifies the number of quadrature points used to compute the empirical Bayes means; the default is the value from estimation.

`iterate(`*#*`)` specifies the maximum number of iterations when computing statistics involving empirical Bayes estimators; the default is the value from estimation.

`tolerance(`*#*`)` specifies convergence tolerance when computing statistics involving empirical Bayes estimators; the default is the value from estimation.

Syntax for estat group

```
estat group
```

Menu for estat

Statistics > Postestimation > Reports and statistics

Remarks and examples

Various predictions, statistics, and diagnostic measures are available after fitting a mixed-effects model using `meglm`. For the most part, calculation centers around obtaining predictions of the random effects. Random effects are not estimated when the model is fit but instead need to be predicted after estimation.

▷ Example 1

In example 2 of [ME] **meglm**, we modeled the probability of contraceptive use among Bangladeshi women by fitting a mixed-effects logistic regression model. To facilitate a more direct comparison between urban and rural women, we express rural status in terms of urban status and eliminate the constant from both the fixed-effects part and the random-effects part.

```
. use http://www.stata-press.com/data/r13/bangladesh
(Bangladesh Fertility Survey, 1989)
. generate byte rural = 1 - urban
. meglm c_use rural urban age child*, nocons || district: rural urban, nocons
> family(bernoulli) link(logit)
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 = -1208.3922
Fitting full model:
Iteration 0:   log likelihood = -1208.3922  (not concave)
Iteration 1:   log likelihood = -1203.6498  (not concave)
Iteration 2:   log likelihood = -1200.6662
Iteration 3:   log likelihood = -1199.9939
Iteration 4:   log likelihood = -1199.3784
Iteration 5:   log likelihood = -1199.3272
Iteration 6:   log likelihood = -1199.3268
Iteration 7:   log likelihood = -1199.3268
Mixed-effects GLM                               Number of obs      =      1934
Family:               Bernoulli
Link:                     logit
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(6)       =    120.59
Log likelihood = -1199.3268                     Prob > chi2        =    0.0000
 ( 1)  [c_use]_cons = 0

------------------------------------------------------------------------------
       c_use |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       rural |  -1.712549   .1603689   -10.68   0.000    -2.026866   -1.398232
       urban |  -.9004495   .1674683    -5.38   0.000    -1.228681   -.5722176
         age |  -.0264472   .0080196    -3.30   0.001    -.0421652   -.0107291
      child1 |   1.132291   .1603052     7.06   0.000     .8180983    1.446483
      child2 |   1.358692   .1769369     7.68   0.000     1.011902    1.705482
      child3 |   1.354788   .1827459     7.41   0.000     .9966122    1.712963
       _cons |          0  (omitted)
-------------+----------------------------------------------------------------
district     |
  var(rural) |   .3882825   .1284858                      .2029918    .7427064
  var(urban) |    .239777   .1403374                      .0761401    .7550947
------------------------------------------------------------------------------
LR test vs. logistic regression:     chi2(2) =    58.40   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

We used the binary variables, `rural` and `urban`, instead of the factor notation `i.urban` because, although supported in the fixed-effects specification of the model, such notation is not supported in random-effects specifications.

This particular model allows for district random effects that are specific to the rural and urban areas of that district and that can be interpreted as such. We can obtain predictions of posterior means of the random effects and their standard errors by typing

```
. predict re_rural re_urban, remeans reses(se_rural se_urban)
(calculating posterior means of random effects)
(using 7 quadrature points)
```

The order in which we specified the variables to be generated corresponds to the order in which the variance components are listed in `meglm` output. If in doubt, a simple `describe` will show how these newly generated variables are labeled just to be sure.

Having generated estimated random effects and standard errors, we can now list them for the first 10 districts:

```
. by district, sort: generate tag = (_n==1)
. list district re_rural se_rural re_urban se_urban if district <= 10 & tag,
> sep(0)
```

	district	re_rural	se_rural	re_urban	se_urban
1.	1	-.9523374	.316291	-.5619418	.2329456
118.	2	-.0425217	.3819309	2.73e-18	.4896702
138.	3	-1.25e-16	.6231232	.2229486	.4658747
140.	4	-.2703357	.3980832	.574464	.3962131
170.	5	.0691029	.3101591	.0074569	.4650451
209.	6	-.3939819	.2759802	.2622263	.4177785
274.	7	-.1904756	.4043461	4.60e-18	.4896702
292.	8	.0382993	.3177392	.2250237	.4654329
329.	9	-.3715211	.3919996	.0628076	.453568
352.	10	-.5624707	.4763545	9.03e-20	.4896702

The estimated standard errors are conditional on the values of the estimated model parameters: β and the components of Σ. Their interpretation is therefore not one of standard sample-to-sample variability but instead one that does not incorporate uncertainty in the estimated model parameters; see *Methods and formulas*. That stated, conditional standard errors can still be used as a measure of relative precision, provided that you keep this caveat in mind.

You can also obtain predictions of posterior modes and compare them with the posterior means:

```
. predict mod_rural mod_urban, remodes
(calculating posterior modes of random effects)
. list district re_rural mod_rural re_urban mod_urban if district <= 10 & tag,
> sep(0)
```

	district	re_rural	mod_rural	re_urban	mod_urban
1.	1	-.9523374	-.9295366	-.5619418	-.5584528
118.	2	-.0425217	-.0306312	2.73e-18	0
138.	3	-1.25e-16	0	.2229486	.2223551
140.	4	-.2703357	-.2529507	.574464	.5644512
170.	5	.0691029	.0789803	.0074569	.0077525
209.	6	-.3939819	-.3803784	.2622263	.2595116
274.	7	-.1904756	-.1737696	4.60e-18	0
292.	8	.0382993	.0488528	.2250237	.2244676
329.	9	-.3715211	-.3540084	.0628076	.0605462
352.	10	-.5624707	-.535444	9.03e-20	0

The two set of predictions are fairly close.

Because not all districts contain both urban and rural areas, some of the posterior modes are 0 and some of the posterior means are practically 0. A closer examination of the data reveals that district 3 has no rural areas, and districts 2, 7, and 10 have no urban areas.

Had we imposed an unstructured covariance structure in our model, the estimated posterior modes and posterior means in the cases in question would not be exactly 0 because of the correlation between urban and rural effects. For instance, if a district has no urban areas, it can still yield a nonzero (albeit small) random-effects estimate for a nonexistent urban area because of the correlation with its rural counterpart; see example 1 of [ME] **meqrlogit postestimation** for details.

◁

▷ Example 2

Continuing with the model from example 1, we can obtain predicted probabilities, and unless we specify the `fixedonly` option, these predictions will incorporate the estimated subject-specific random effects $\widetilde{\mathbf{u}}_j$.

```
. predict pr, pr
(predictions based on fixed effects and posterior means of random effects)
(using 7 quadrature points)
```

The predicted probabilities for observation i in cluster j are obtained by applying the inverse link function to the linear predictor, $\widehat{p}_{ij} = g^{-1}(\mathbf{x}_{ij}\widehat{\boldsymbol{\beta}} + \mathbf{z}_{ij}\widetilde{\mathbf{u}}_j)$; see *Methods and formulas* for details. By default, the calculation uses posterior means for $\widetilde{\mathbf{u}}_j$ unless you specify the `modes` option, in which case the calculation uses posterior modes for $\widetilde{\widetilde{\mathbf{u}}}_j$.

```
. predict prm, pr modes
(predictions based on fixed effects and posterior modes of random effects)
```

We can list the two sets of predicted probabilities together with the actual outcome for some district, let's say district 38:

```
. list c_use pr prm if district == 38
```

	c_use	pr	prm
1228.	yes	.5783408	.5780864
1229.	no	.5326623	.5324027
1230.	yes	.6411679	.6409279
1231.	yes	.5326623	.5324027
1232.	yes	.5718783	.5716228
1233.	no	.3447686	.344533
1234.	no	.4507973	.4505391
1235.	no	.1940524	.1976133
1236.	no	.2846738	.2893007
1237.	no	.1264883	.1290078
1238.	no	.206763	.2104961
1239.	no	.202459	.2061346
1240.	no	.206763	.2104961
1241.	no	.1179788	.1203522

The two sets of predicted probabilities are fairly close.

For mixed-effects models with many levels or many random effects, the calculation of the posterior means of random effects or any quantities that are based on the posterior means of random effects may take a long time. This is because we must resort to numerical integration to obtain the posterior means. In contrast, the calculation of the posterior modes of random effects is usually orders of magnitude faster because there is no numerical integration involved. For this reason, empirical modes are often used in practice as an approximation to empirical means. Note that for linear mixed-effects models, the two predictors are the same.

We can compare the observed values with the predicted values by constructing a classification table. Defining success as $\widehat{y}_{ij} = 1$ if $\widehat{p}_{ij} > 0.5$ and defining $\widehat{y}_{ij} = 0$ otherwise, we obtain the following table.

```
. gen p_use = pr > .5
. label var p_use "Predicted outcome"
. tab2 c_use p_use, row
-> tabulation of c_use by p_use
```

Key
frequency *row percentage*

Use contraception	Predicted outcome 0	1	Total
no	991 84.34	184 15.66	1,175 100.00
yes	423 55.73	336 44.27	759 100.00
Total	1,414 73.11	520 26.89	1,934 100.00

The model correctly classified 84% of women who did not use contraceptives but only 44% of women who did. In the next example, we will look at some residual diagnostics.

◁

❑ Technical note

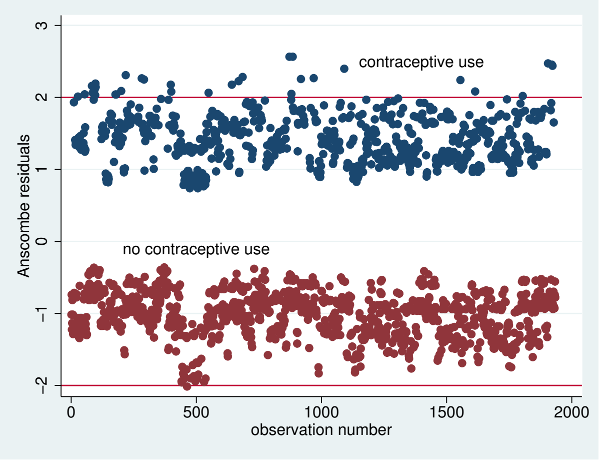
Out-of-sample predictions are permitted after `meglm`, but if these predictions involve estimated random effects, the integrity of the estimation data must be preserved. If the estimation data have changed since the model was fit, `predict` will be unable to obtain predicted random effects that are appropriate for the fitted model and will give an error. Thus to obtain out-of-sample predictions that contain random-effects terms, be sure that the data for these predictions are in observations that augment the estimation data.

❑

▷ Example 3

Continuing our discussion from example 2, here we look at residual diagnostics. `meglm` offers three kinds of predicted residuals for nonlinear responses—Pearson, Anscombe, and deviance. Of the three, Anscombe residuals are designed to be approximately normally distributed; thus we can check for outliers by plotting Anscombe residuals against observation numbers and seeing which residuals are greater than 2 in absolute value.

```
. predict anscombe, anscombe
(predictions based on fixed effects and posterior means of random effects)
. gen n = _n
. label var n "observation number"
. twoway (scatter anscombe n if c_use) (scatter anscombe n if !c_use),
> yline(-2 2) legend(off) text(2.5 1400 "contraceptive use")
> text(-.1 500 "no contraceptive use")
```



There seem to be some outliers among residuals that identify women who use contraceptives. We could examine the observations corresponding to the outliers, or we could try fitting a model with perhaps a different covariance structure, which we leave as an exercise.

◁

▷ Example 4

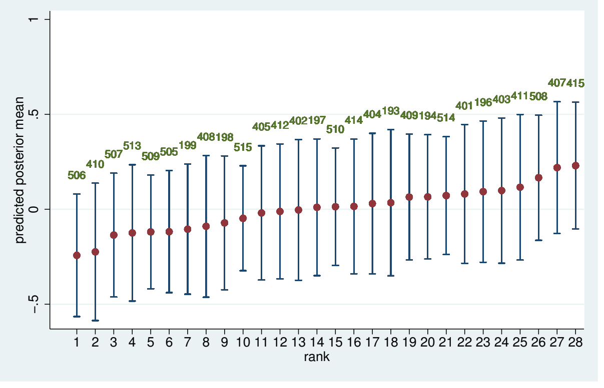
In example 3 of [ME] **meglm**, we estimated the effects of two treatments on the tobacco and health knowledge (THK) scale score of students in 28 schools. The dependent variable was collapsed into four ordered categories, and we fit a three-level ordinal logistic regression.

```
. use http://www.stata-press.com/data/r13/tvsfpors, clear
. meologit thk prethk i.cc##i.tv || school: || class:
Fitting fixed-effects model:
Iteration 0:   log likelihood =  -2212.775
Iteration 1:   log likelihood =  -2125.509
Iteration 2:   log likelihood = -2125.1034
Iteration 3:   log likelihood = -2125.1032
Refining starting values:
Grid node 0:   log likelihood = -2152.1514
Fitting full model:
  (output omitted )
Mixed-effects ologit regression                 Number of obs      =      1600

-----------------------------------------------------------
                |   No. of       Observations per Group
 Group Variable |   Groups    Minimum    Average    Maximum
----------------+------------------------------------------
         school |       28         18       57.1        137
          class |      135          1       11.9         28
-----------------------------------------------------------

Integration method: mvaghermite                 Integration points =         7
                                                Wald chi2(4)       =    124.39
Log likelihood = -2114.5881                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
          thk |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       prethk |   .4085273    .039616    10.31   0.000     .3308814    .4861731
         1.cc |   .8844369   .2099124     4.21   0.000     .4730161    1.295858
         1.tv |    .236448   .2049065     1.15   0.249    -.1651614    .6380575
              |
        cc#tv |
          1 1 |  -.3717699   .2958887    -1.26   0.209     -.951701    .2081612
-------------+----------------------------------------------------------------
        /cut1 |  -.0959459   .1688988    -0.57   0.570    -.4269815    .2350896
        /cut2 |   1.177478   .1704946     6.91   0.000     .8433151    1.511642
        /cut3 |   2.383672   .1786736    13.34   0.000     2.033478    2.733865
-------------+----------------------------------------------------------------
school       |
    var(_cons)|   .0448735   .0425387                      .0069997    .2876749
-------------+----------------------------------------------------------------
school>class |
    var(_cons)|   .1482157   .0637521                       .063792    .3443674
------------------------------------------------------------------------------
LR test vs. ologit regression:       chi2(2) =    21.03   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

Not surprisingly, the level of knowledge before the intervention is a good predictor of the level of knowledge after the intervention. The social resistance classroom curriculum is effective in raising the knowledge score, but the TV intervention and the interaction term are not.

We can rank schools by their institutional effectiveness by plotting the random effects at the school level.

```
. predict re_school re_class, remeans reses(se_school se_class)
(calculating posterior means of random effects)
(using 7 quadrature points)
. generate lower = re_school - 1.96*se_school
. generate upper = re_school + 1.96*se_school
. egen tag = tag(school)
. gsort +re_school -tag
. generate rank = sum(tag)
. generate labpos = re_school + 1.96*se_school + .1
. twoway (rcap lower upper rank) (scatter re_school rank)
> (scatter labpos rank, mlabel(school) msymbol(none) mlabpos(0)),
> xtitle(rank) ytitle(predicted posterior mean) legend(off)
> xscale(range(0 28)) xlabel(1/28) ysize(2)
```



Although there is some variability in the predicted posterior means, we cannot see significant differences among the schools in this example.

◁

Methods and formulas

Continuing the discussion in *Methods and formulas* of [ME] **meglm** and using the definitions and formulas defined there, we begin by considering the prediction of the random effects $\mathbf{u}_j$ for the jth cluster in a two-level model. Prediction of random effects in multilevel generalized linear models involves assigning values to random effects, and there are many methods for doing so; see Skrondal and Rabe-Hesketh (2009) and Skrondal and Rabe-Hesketh (2004, chap. 7) for a comprehensive review. Stata offers two methods of predicting random effects: empirical Bayes means (also known as posterior means) and empirical Bayes modes (also known as posterior modes). Below we provide more details about the two methods.

Let $\widehat{\boldsymbol{\theta}}$ denote the estimated model parameters comprising $\widehat{\boldsymbol{\beta}}$ and the unique elements of $\widehat{\boldsymbol{\Sigma}}$. Empirical Bayes (EB) predictors of the random effects are the means or modes of the empirical posterior distribution with the parameter estimates $\boldsymbol{\theta}$ replaced with their estimates $\widehat{\boldsymbol{\theta}}$. The method is called "empirical" because $\widehat{\boldsymbol{\theta}}$ is treated as known. EB combines the prior information about the random effects with the likelihood to obtain the conditional posterior distribution of random effects. Using Bayes' theorem, the empirical conditional posterior distribution of random effects for cluster j is

$$\omega(\mathbf{u}_j|\mathbf{y}_j,\mathbf{X}_j,\mathbf{Z}_j;\widehat{\boldsymbol{\theta}}) = \frac{\Pr(\mathbf{y}_j,\mathbf{u}_j|\mathbf{X}_j,\mathbf{Z}_j;\widehat{\boldsymbol{\theta}})}{\Pr(\mathbf{y}_j|\mathbf{X}_j,\mathbf{Z}_j;\widehat{\boldsymbol{\theta}})}$$
$$= \frac{f(\mathbf{y}_j|\mathbf{u}_j,\mathbf{X}_j,\mathbf{Z}_j;\widehat{\boldsymbol{\beta}})\,\phi(\mathbf{u}_j;\widehat{\boldsymbol{\Sigma}})}{\int f(\mathbf{y}_j|\mathbf{u}_j)\,\phi(\mathbf{u}_j)\,d\mathbf{u}_j}$$
$$= \frac{f(\mathbf{y}_j|\mathbf{u}_j,\mathbf{X}_j,\mathbf{Z}_j;\widehat{\boldsymbol{\beta}})\,\phi(\mathbf{u}_j;\widehat{\boldsymbol{\Sigma}})}{\mathcal{L}_j(\widehat{\boldsymbol{\theta}})}$$

The denominator is just the likelihood contribution of the jth cluster.

EB mean predictions of random effects, $\widetilde{\mathbf{u}}$, also known as posterior means, are calculated as

$$\widetilde{\mathbf{u}} = \int_{\Re^q} \mathbf{u}_j\,\omega(\mathbf{u}_j|\mathbf{y}_j,\mathbf{X}_j,\mathbf{Z}_j;\widehat{\boldsymbol{\theta}})\,d\mathbf{u}_j$$
$$= \frac{\int_{\Re^q} \mathbf{u}_j\, f(\mathbf{y}_j|\mathbf{u}_j,\mathbf{X}_j,\mathbf{Z}_j;\widehat{\boldsymbol{\beta}})\,\phi(\mathbf{u}_j;\widehat{\boldsymbol{\Sigma}})\,d\mathbf{u}_j}{\int_{\Re^q} f(\mathbf{y}_j|\mathbf{u}_j)\,\phi(\mathbf{u}_j)\,d\mathbf{u}_j}$$

where we use the notation $\widetilde{\mathbf{u}}$ rather than $\widehat{\mathbf{u}}$ to distinguish predicted values from estimates. This multivariate integral is approximated by the mean–variance adaptive Gaussian quadrature; see *Methods and formulas* of [ME] **meglm** for details about the quadrature. If you have multiple random effects within a level or random effects across levels, the calculation involves orthogonalizing transformations using the Cholesky transformation because the random effects are no longer independent under the posterior distribution.

In a linear mixed-effects model, the posterior density is multivariate normal, and EB means are also best linear unbiased predictors (BLUPs); see Skrondal and Rabe-Hesketh (2004, 227). In generalized mixed-effects models, the posterior density tends to multivariate normal as cluster size increases.

EB modal predictions can be approximated by solving for the mode $\widetilde{\widetilde{\mathbf{u}}}_j$ in

$$\frac{\partial}{\partial \mathbf{u}_j} \log \omega(\widetilde{\widetilde{\mathbf{u}}}_j|\mathbf{y}_j,\mathbf{X}_j,\mathbf{Z}_j;\widehat{\boldsymbol{\theta}}) = \mathbf{0}$$

Because the denominator in $\omega(\cdot)$ does not depend on $\mathbf{u}$, we can omit it from the calculation to obtain

$$\frac{\partial}{\partial \mathbf{u}_j} \log\left\{ f(\mathbf{y}_j|\mathbf{u}_j,\mathbf{X}_j,\mathbf{Z}_j;\widehat{\boldsymbol{\beta}})\,\phi(\mathbf{u}_j;\widehat{\boldsymbol{\Sigma}})\right\}$$
$$= \frac{\partial}{\partial \mathbf{u}_j} \log f\left(\mathbf{y}_j|\mathbf{u}_j,\mathbf{X}_j,\mathbf{Z}_j;\widehat{\boldsymbol{\beta}}\right) + \frac{\partial}{\partial \mathbf{u}_j} \log \phi\left(\mathbf{u}_j;\widehat{\boldsymbol{\Sigma}}\right) = 0$$

The calculation of EB modes does not require numerical integration, and for that reason they are often used in place of EB means. As the posterior density gets closer to being multivariate normal, EB modes get closer and closer to EB means.

Just like there are many methods of assigning values to the random effects, there exist many methods of calculating standard errors of the predicted random effects; see Skrondal and Rabe-Hesketh (2009) for a comprehensive review.

Stata uses the posterior standard deviation as the standard error of the posterior means predictor of random effects. The EB posterior covariance matrix of the random effects is given by

$$\text{cov}(\widetilde{\mathbf{u}}_j|\mathbf{y}_j,\mathbf{X}_j,\mathbf{Z}_j;\widehat{\boldsymbol{\theta}}) = \int_{\Re^q} (\mathbf{u}_j - \widetilde{\mathbf{u}}_j)(\mathbf{u}_j - \widetilde{\mathbf{u}}_j)'\,\omega(\mathbf{u}_j|\mathbf{y}_j,\mathbf{X}_j,\mathbf{Z}_j;\widehat{\boldsymbol{\theta}})\,d\mathbf{u}_j$$

The posterior covariance matrix and the integrals are approximated by the mean–variance adaptive Gaussian quadrature; see *Methods and formulas* of [ME] **meglm** for details about the quadrature.

Conditional standard errors for the estimated posterior modes are derived from standard theory of maximum likelihood, which dictates that the asymptotic variance matrix of $\widetilde{\widetilde{\mathbf{u}}}_j$ is the negative inverse of the Hessian, $g''(\boldsymbol{\beta},\boldsymbol{\Sigma},\widetilde{\widetilde{\mathbf{u}}}_j)$.

In what follows, we show formulas using the posterior means estimates of random effects $\widetilde{\mathbf{u}}_j$, which are used by default or if the `means` option is specified. If the `modes` option is specified, $\widetilde{\mathbf{u}}_j$ are simply replaced with the posterior modes $\widetilde{\widetilde{\mathbf{u}}}_j$ in these formulas.

For any ith observation in the jth cluster in a two-level model, define the linear predictor as

$$\widehat{\eta}_{ij} = \mathbf{x}_{ij}\widehat{\boldsymbol{\beta}} + \mathbf{z}_{ij}\widetilde{\mathbf{u}}_j$$

The linear predictor includes the offset or exposure variable if one was specified during estimation, unless the `nooffset` option is specified. If the `fixedonly` option is specified, $\widehat{\eta}$ contains the linear predictor for only the fixed portion of the model, $\widehat{\eta}_{ij} = \mathbf{x}_{ij}\widehat{\boldsymbol{\beta}}$.

The predicted mean, conditional on the random effects $\widetilde{\mathbf{u}}_j$, is

$$\widehat{\mu}_{ij} = g^{-1}(\widehat{\eta}_{ij})$$

where $g^{-1}(\cdot)$ is the inverse link function for $\mu_{ij} = g^{-1}(\eta_{ij})$ defined as follows for the available links in `link(`*link*`)`:

link	Inverse link
`identity`	η_{ij}
`logit`	$1/\{1+\exp(-\eta_{ij})\}$
`probit`	$\Phi(\eta_{ij})$
`log`	$\exp(\eta_{ij})$
`cloglog`	$1-\exp\{-\exp(\eta_{ij})\}$

By default, random effects and any statistic based on them—`mu`, `fitted`, `pearson`, `deviance`, `anscombe`—are calculated using posterior means of random effects unless option `modes` is specified, in which case the calculations are based on posterior modes of random effects.

Raw residuals are calculated as the difference between the observed and fitted outcomes,

$$\nu_{ij} = y_{ij} - \widehat{\mu}_{ij}$$

and are only defined for the Gaussian family.

Let r_{ij} be the number of Bernoulli trials in a binomial model, α be the conditional overdispersion parameter under the mean parameterization of the negative binomial model, and δ be the conditional overdispersion parameter under the constant parameterization of the negative binomial model.

Pearson residuals are raw residuals divided by the square root of the variance function

$$\nu_{ij}^P = \frac{\nu_{ij}}{\{V(\widehat{\mu}_{ij})\}^{1/2}}$$

where $V(\widehat{\mu}_{ij})$ is the family-specific variance function defined as follows for the available families in `family(`*family*`)`:

family	Variance function $V(\widehat{\mu}_{ij})$
`bernoulli`	$\widehat{\mu}_{ij}(1-\widehat{\mu}_{ij})$
`binomial`	$\widehat{\mu}_{ij}(1-\widehat{\mu}_{ij}/r_{ij})$
`gamma`	$\widehat{\mu}_{ij}^2$
`gaussian`	1
`nbinomial mean`	$\widehat{\mu}_{ij}(1+\alpha\widehat{\mu}_{ij})$
`nbinomial constant`	$\widehat{\mu}_{ij}(1+\delta)$
`ordinal`	not defined
`poisson`	$\widehat{\mu}_{ij}$

Deviance residuals are calculated as

$$\nu_{ij}^{D} = \text{sign}(\nu_{ij})\sqrt{\widehat{d}_{ij}^{\,2}}$$

where the squared deviance residual $\widehat{d}_{ij}^{\,2}$ is defined as follows:

family	Squared deviance $\widehat{d}_{ij}^{\,2}$
`bernoulli`	$-2\log(1-\widehat{\mu}_{ij})$ if $y_{ij}=0$ $-2\log(\widehat{\mu}_{ij})$ if $y_{ij}=1$
`binomial`	$2r_{ij}\log\left(\frac{r_{ij}}{r_{ij}-\widehat{\mu}_{ij}}\right)$ if $y_{ij}=0$ $2y_{ij}\log\left(\frac{y_{ij}}{\widehat{\mu}_{ij}}\right)+2(r_{ij}-y_{ij})\log\left(\frac{r_{ij}-y_{ij}}{r_{ij}-\widehat{\mu}_{ij}}\right)$ if $0<y_{ij}<r_{ij}$ $2r_{ij}\log\left(\frac{r_{ij}}{\widehat{\mu}_{ij}}\right)$ if $y_{ij}=r_{ij}$
`gamma`	$-2\left\{\log\left(\frac{y_{ij}}{\widehat{\mu}_{ij}}\right)-\frac{\widehat{\nu}_{ij}}{\widehat{\mu}_{ij}}\right\}$
`gaussian`	$\widehat{\nu}_{ij}^{2}$
`nbinomial mean`	$2\log\left(1+\alpha\widehat{\mu}_{ij}\right)\alpha$ if $y_{ij}=0$ $2y_{ij}\log\left(\frac{y_{ij}}{\widehat{\mu}_{ij}}\right)-\frac{2}{\alpha}(1+\alpha y_{ij})\log\left(\frac{1+\alpha y_{ij}}{1+\alpha\widehat{\mu}_{ij}}\right)$ otherwise
`nbinomial constant`	not defined
`ordinal`	not defined
`poisson`	$2\widehat{\mu}_{ij}$ if $y_{ij}=0$ $2y_{ij}\log\left(\frac{y_{ij}}{\widehat{\mu}_{ij}}\right)-2\widehat{\nu}_{ij}$ otherwise

Anscombe residuals, denoted ν_{ij}^A, are calculated as follows:

family	Anscombe residual ν_{ij}^A
`bernoulli`	$\dfrac{3\left\{y_{ij}^{2/3}\mathcal{H}(y_{ij}) - \widehat{\mu}_{ij}^{2/3}\mathcal{H}(\widehat{\mu}_{ij})\right\}}{2\left(\widehat{\mu}_{ij} - \widehat{\mu}_{ij}^2\right)^{1/6}}$
`binomial`	$\dfrac{3\left\{y_{ij}^{2/3}\mathcal{H}(y_{ij}/r_{ij}) - \widehat{\mu}_{ij}^{2/3}\mathcal{H}(\widehat{\mu}_{ij}/r_{ij})\right\}}{2\left(\widehat{\mu}_{ij} - \widehat{\mu}_{ij}^2/r_{ij}\right)^{1/6}}$
`gamma`	$\dfrac{3(y_{ij}^{1/3} - \widehat{\mu}_{ij}^{1/3})}{\widehat{\mu}_{ij}^{1/3}}$
`gaussian`	ν_{ij}
`nbinomial mean`	$\dfrac{\mathcal{H}(-\alpha y_{ij}) - \mathcal{H}(-\alpha\widehat{\mu}_{ij}) + 1.5(y_{ij}^{2/3} - \widehat{\mu}_{ij}^{2/3})}{(\widehat{\mu}_{ij} + \alpha\widehat{\mu}_{ij}^2)^{1/6}}$
`nbinomial constant`	not defined
`ordinal`	not defined
`poisson`	$\dfrac{3(y_{ij}^{2/3} - \widehat{\mu}_{ij}^{2/3})}{2\widehat{\mu}_{ij}^{1/6}}$

where $\mathcal{H}(t)$ is a specific univariate case of the Hypergeometric2F1 function (Wolfram 1999, 771–772), defined here as $\mathcal{H}(t) = {}_2F_1(2/3, 1/3, 5/3, t)$.

For a discussion of the general properties of the various residuals, see Hardin and Hilbe (2012, chap. 4).

References

Hardin, J. W., and J. M. Hilbe. 2012. *Generalized Linear Models and Extensions*. 3rd ed. College Station, TX: Stata Press.

McCullagh, P., and J. A. Nelder. 1989. *Generalized Linear Models*. 2nd ed. London: Chapman & Hall/CRC.

Skrondal, A., and S. Rabe-Hesketh. 2004. *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models*. Boca Raton, FL: Chapman & Hall/CRC.

——. 2009. Prediction in multilevel generalized linear models. *Journal of the Royal Statistical Society, Series A* 172: 659–687.

Wolfram, S. 1999. *The Mathematica Book*. 4th ed. Cambridge: Cambridge University Press.

Also see

[ME] **meglm** — Multilevel mixed-effects generalized linear model

[U] 20 Estimation and postestimation commands

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	
binomial(*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, robust, or cluster *clustvar*
Reporting	
level(#)	set confidence level; default is level(95)
or	report fixed-effects coefficients as odds ratios
nocnsreport	do not display constraints
notable	suppress coefficient table
noheader	suppress output header
nogroup	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	
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
coeflegend	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
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 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, 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 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 McCulloch (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 $\mathbf{u}_j$,

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

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 $\mathit{depvar}_{ij} \neq 0$ 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)
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     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

-----------------------------------------------------------
                |   No. of       Observations per Group
 Group Variable |   Groups    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, $j = 1, \dots, M$, the conditional distribution of $\mathbf{y}_j = (y_{j1}, \dots, y_{jn_j})'$, 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 $f(\mathbf{y}_j, \mathbf{u}_j)$,

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

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

Title

melogit postestimation — Postestimation tools for melogit

Description

The following postestimation commands are of special interest after `melogit`:

Command	Description
`estat group`	summarize the composition of the nested groups
`estat icc`	estimate intraclass correlations

The following standard postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat ic`	Akaike's and Schwarz's Bayesian information criteria (AIC and BIC)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

Special-interest postestimation commands

estat group reports the number of groups and minimum, average, and maximum group sizes for each level of the model. Model levels are identified by the corresponding group variable in the data. Because groups are treated as nested, the information in this summary may differ from what you would get if you used the tabulate command on each group variable individually.

estat icc displays the intraclass correlation for pairs of latent linear responses at each nested level of the model. Intraclass correlations are available for random-intercept models or for random-coefficient models conditional on random-effects covariates being equal to 0. They are not available for crossed-effects models.

Syntax for predict

Syntax for obtaining predictions of random effects and their standard errors

predict [*type*] *newvarsspec* [*if*] [*in*] , { remeans | remodes } [reses(*newvarsspec*)]

Syntax for obtaining other predictions

predict [*type*] *newvarsspec* [*if*] [*in*] [, *statistic* *options*]

newvarsspec is *stub** or *newvarlist*.

statistic	Description
Main	
mu	predicted mean; the default
fitted	fitted linear predictor
xb	linear predictor for the fixed portion of the model only
stdp	standard error of the fixed-portion linear prediction
pearson	Pearson residuals
deviance	deviance residuals
anscombe	Anscombe residuals

These statistics are available both in and out of sample; type predict ... if e(sample) ... if wanted only for the estimation sample.

options	Description
Main	
means	compute *statistic* using empirical Bayes means; the default
modes	compute *statistic* using empirical Bayes modes
nooffset	ignore the offset variable in calculating predictions; relevant only if you specified offset() when you fit the model
fixedonly	prediction for the fixed portion of the model only
Integration	
intpoints(#)	use # quadrature points to compute empirical Bayes means
iterate(#)	set maximum number of iterations in computing statistics involving empirical Bayes estimators
tolerance(#)	set convergence tolerance for computing statistics involving empirical Bayes estimators

Menu for predict

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

remeans, remodes, reses(); see [ME] **meglm postestimation**.

mu, the default, calculates the predicted mean (the probability of a positive outcome), that is, the inverse link function applied to the linear prediction. By default, this is based on a linear predictor that includes both the fixed effects and the random effects, and the predicted mean is conditional on the values of the random effects. Use the fixedonly option if you want predictions that include only the fixed portion of the model, that is, if you want random effects set to 0.

fitted, xb, stdp, pearson, deviance, anscombe, means, modes, nooffset, fixedonly; see [ME] **meglm postestimation**.

By default or if the means option is specified, statistics mu, fitted, xb, stdp, pearson, deviance, and anscombe are based on the posterior mean estimates of random effects. If the modes option is specified, these statistics are based on the posterior mode estimates of random effects.

Integration

intpoints(), iterate(), tolerance(); see [ME] **meglm postestimation**.

Syntax for estat

Summarize the composition of the nested groups

```
estat group
```

Estimate intraclass correlations

```
estat icc [, level(#)]
```

Menu for estat

Statistics > Postestimation > Reports and statistics

Option for estat icc

`level(#)` specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`; see [U] **20.7 Specifying the width of confidence intervals**.

Remarks and examples

Various predictions, statistics, and diagnostic measures are available after fitting a logistic mixed-effects model with `melogit`. Here we show a short example of predicted probabilities and predicted random effects; refer to [ME] **meglm postestimation** for additional examples.

▷ Example 1

In example 2 of [ME] **melogit**, we analyzed the cognitive ability (`dtlm`) of patients with schizophrenia compared with their relatives and control subjects, by using a three-level logistic model with random effects at the family and subject levels. Cognitive ability was measured as the successful completion of the "Tower of London", a computerized task, measured at three levels of difficulty.

```
. use http://www.stata-press.com/data/r13/towerlondon
(Tower of London data)
. melogit dtlm difficulty i.group || family: || subject: , or
  (output omitted )
```

We obtain predicted probabilities based on the contribution of both fixed effects and random effects by typing

```
. predict pr
(predictions based on fixed effects and posterior means of random effects)
(option mu assumed)
(using 7 quadrature points)
```

As the note says, the predicted values are based on the posterior means of random effects. You can use the `modes` option to obtain predictions based on the posterior modes of random effects.

We obtain predictions of the posterior means themselves by typing

```
. predict re*, remeans
(calculating posterior means of random effects)
(using 7 quadrature points)
```

Because we have one random effect at the family level and another random effect at the subject level, Stata saved the predicted posterior means in the variables `re1` and `re2`, respectively. If you are not sure which prediction corresponds to which level, you can use the `describe` command to show the variable labels.

Here we list the data for family 16:

```
. list family subject dtlm pr re1 re2 if family==16, sepby(subject)

     +-----------------------------------------------------------------+
     | family   subject   dtlm         pr         re1          re2    |
     |-----------------------------------------------------------------|
208. |     16         5      1   .5337746    .8045555    .2204122    |
209. |     16         5      0   .1804649    .8045555    .2204122    |
210. |     16         5      0   .0406325    .8045555    .2204122    |
     |-----------------------------------------------------------------|
211. |     16        34      1   .8956181    .8045555    1.430945    |
212. |     16        34      1   .6226832    .8045555    1.430945    |
213. |     16        34      1   .2409364    .8045555    1.430945    |
     |-----------------------------------------------------------------|
214. |     16        35      0   .6627467    .8045555    -.042955    |
215. |     16        35      1   .2742936    .8045555    -.042955    |
216. |     16        35      0   .0677705    .8045555    -.042955    |
     +-----------------------------------------------------------------+
```

The predicted random effects at the family level (`re1`) are the same for all members of the family. Similarly, the predicted random effects at the individual level (`re2`) are constant within each individual. The predicted probabilities (`pr`) for this family seem to be in fair agreement with the response (`dtlm`) based on a cutoff of 0.5.

We can use `estat icc` to estimate the residual intraclass correlation (conditional on the difficulty level and the individual's category) between the latent responses of subjects within the same family or between the latent responses of the same subject and family:

```
. estat icc

Residual intraclass correlation

------------------------------------------------------------------------------
                       Level |        ICC   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
                      family |   .1139105    .0997727      .0181851    .4715289
              subject|family |   .3416307    .0889471       .192923    .5297291
------------------------------------------------------------------------------
```

`estat icc` reports two intraclass correlations for this three-level nested model. The first is the level-3 intraclass correlation at the family level, the correlation between latent measurements of the cognitive ability in the same family. The second is the level-2 intraclass correlation at the subject-within-family level, the correlation between the latent measurements of cognitive ability in the same subject and family.

There is not a strong correlation between individual realizations of the latent response, even within the same subject.

◁

Stored results

`estat icc` stores the following in `r()`:

Scalars

`r(icc#)`	level-# intraclass correlation
`r(se#)`	standard errors of level-# intraclass correlation
`r(level)`	confidence level of confidence intervals

Macros

`r(label#)`	label for level #

Matrices

`r(ci#)`	vector of confidence intervals (lower and upper) for level-# intraclass correlation

For a G-level nested model, # can be any integer between 2 and G.

Methods and formulas

Methods and formulas are presented under the following headings:

Prediction
Intraclass correlations

Prediction

Methods and formulas for predicting random effects and other statistics are given in *Methods and formulas* of [ME] **meglm postestimation**.

Intraclass correlations

Consider a simple, two-level random-intercept model, stated in terms of a latent linear response, where only $y_{ij} = I(y^*_{ij} > 0)$ is observed for the latent variable,

$$y^*_{ij} = \beta + u^{(2)}_j + \epsilon^{(1)}_{ij}$$

with $i = 1, \dots, n_j$ and level-2 groups $j = 1, \dots, M$. Here β is an unknown fixed intercept, $u^{(2)}_j$ is a level-2 random intercept, and $\epsilon^{(1)}_{ij}$ is a level-1 error term. Errors are assumed to be logistic with mean 0 and variance $\sigma^2_1 = \pi^2/3$; random intercepts are assumed to be normally distributed with mean 0 and variance σ^2_2 and to be independent of error terms.

The intraclass correlation for this model is

$$\rho = \text{Corr}(y^*_{ij}, y^*_{i'j}) = \frac{\sigma^2_2}{\pi^2/3 + \sigma^2_2}$$

It corresponds to the correlation between the latent responses i and i' from the same group j.

Now consider a three-level nested random-intercept model,

$$y^*_{ijk} = \beta + u^{(2)}_{jk} + u^{(3)}_k + \epsilon^{(1)}_{ijk}$$

for measurements $i = 1, \dots, n_{jk}$ and level-2 groups $j = 1, \dots, M_{1k}$ nested within level-3 groups $k = 1, \dots, M_2$. Here $u^{(2)}_{jk}$ is a level-2 random intercept, $u^{(3)}_k$ is a level-3 random intercept, and $\epsilon^{(1)}_{ijk}$ is a level-1 error term. The error terms have a logistic distribution with mean 0 and variance $\sigma^2_1 = \pi^2/3$. The random intercepts are assumed to be normally distributed with mean 0 and variances σ^2_2 and σ^2_3, respectively, and to be mutually independent. The error terms are also independent of the random intercepts.

We can consider two types of intraclass correlations for this model. We will refer to them as level-2 and level-3 intraclass correlations. The level-3 intraclass correlation is

$$\rho^{(3)} = \text{Corr}(y^*_{ijk}, y^*_{i'j'k}) = \frac{\sigma^2_3}{\pi^2/3 + \sigma^2_2 + \sigma^2_3}$$

This is the correlation between latent responses i and i' from the same level-3 group k and from different level-2 groups j and j'.

The level-2 intraclass correlation is

$$\rho^{(2)} = \text{Corr}(y^*_{ijk}, y^*_{i'jk}) = \frac{\sigma^2_2 + \sigma^2_3}{\pi^2/3 + \sigma^2_2 + \sigma^2_3}$$

This is the correlation between latent responses i and i' from the same level-3 group k and level-2 group j. (Note that level-1 intraclass correlation is undefined.)

More generally, for a G-level nested random-intercept model, the g-level intraclass correlation is defined as

$$\rho^{(g)} = \frac{\sum_{l=g}^{G} \sigma^2_l}{\pi^2/3 + \sum_{l=2}^{G} \sigma^2_l}$$

The above formulas also apply in the presence of fixed-effects covariates $\mathbf{X}$ in a random-effects model. In this case, intraclass correlations are conditional on fixed-effects covariates and are referred to as residual intraclass correlations. `estat icc` also uses the same formulas to compute intraclass correlations for random-coefficient models, assuming 0 baseline values for the random-effects covariates, and labels them as conditional intraclass correlations.

Intraclass correlations will always fall in [0,1] because variance components are nonnegative. To accommodate the range of an intraclass correlation, we use the logit transformation to obtain confidence intervals. We use the delta method to estimate the standard errors of the intraclass correlations.

Let $\widehat{\rho}^{(g)}$ be a point estimate of the intraclass correlation and $\widehat{\text{SE}}(\widehat{\rho}^{(g)})$ be its standard error. The $(1-\alpha) \times 100\%$ confidence interval for $\text{logit}(\rho^{(g)})$ is

$$\text{logit}(\widehat{\rho}^{(g)}) \pm z_{\alpha/2} \frac{\widehat{\text{SE}}(\widehat{\rho}^{(g)})}{\widehat{\rho}^{(g)}(1-\widehat{\rho}^{(g)})}$$

where $z_{\alpha/2}$ is the $1-\alpha/2$ quantile of the standard normal distribution and $\text{logit}(x) = \ln\{x/(1-x)\}$. Let k_u be the upper endpoint of this interval, and let k_l be the lower. The $(1-\alpha) \times 100\%$ confidence interval for $\rho^{(g)}$ is then given by

$$\left(\frac{1}{1+e^{-k_l}}, \frac{1}{1+e^{-k_u}} \right)$$

Also see

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

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

[U] 20 Estimation and postestimation commands

Title

menbreg — Multilevel mixed-effects negative binomial regression

Syntax

`menbreg` *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 the constant term from the fixed-effects equation
`exposure(`$varname_e$`)`	include $\ln(varname_e)$ in model with coefficient constrained to 1
`offset(`$varname_o$`)`	include $varname_o$ 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

options	Description
Model	
dispersion(*dispersion*)	parameterization of the conditional overdispersion; *dispersion* may be mean (default) or constant
constraints(*constraints*)	apply specified linear constraints
collinear	keep collinear variables
SE/Robust	
vce(*vcetype*)	*vcetype* may be oim, robust, or cluster *clustvar*
Reporting	
level(#)	set confidence level; default is level(95)
irr	report fixed-effects coefficients as incidence-rate ratios
nocnsreport	do not display constraints
notable	suppress coefficient table
noheader	suppress output header
nogroup	suppress table summarizing groups
nolrtest	do not perform likelihood-ratio test comparing with negative binomial regression
display_options	control 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
coeflegend	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
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 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 > Negative binomial regression

Description

menbreg fits mixed-effects negative binomial models to count data. The conditional distribution of the response given random effects is assumed to follow a Poisson-like process, except that the variation is greater than that of a true Poisson process.

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.

exposure($varname_e$) specifies a variable that reflects the amount of exposure over which the *depvar* events were observed for each observation; ln($varname_e$) is included in the fixed-effects portion of the model with the coefficient constrained to be 1.

offset($varname_o$) specifies that $varname_o$ 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*$[k, l]$.

dispersion(mean | constant) specifies the parameterization of the conditional overdispersion given random effects. dispersion(mean), the default, yields a model where the conditional overdispersion is a function of the conditional mean given random effects. For example, in a two-level model, the conditional overdispersion is equal to $1+\alpha E(y_{ij}|\mathbf{u}_j)$. dispersion(constant) yields a model where the conditional overdispersion is constant and is equal to $1+\delta$. α and δ are the respective conditional overdispersion parameters.

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**.

irr reports estimated fixed-effects coefficients transformed to incidence-rate 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 stored. irr 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 menbreg from performing a likelihood-ratio test that compares the mixed-effects negative binomial model with standard (marginal) negative binomial regression. This option may also be specified upon replay to suppress this test from the output.

display_options: 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 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 menbreg 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 menbreg 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**.

menbreg is a convenience command for meglm with a log link and an nbinomial family; see [ME] **meglm**.

Remarks are presented under the following headings:

Introduction

Mixed-effects negative binomial regression is negative binomial 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 McCulloch (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). Rabe-Hesketh and Skrondal (2012, chap. 13) is a good introductory reading on applied multilevel modeling of count data.

`menbreg` 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.

However, for simplicity, consider a two-level model, where for a series of M independent clusters, and conditional on the latent variable ζ_{ij} and a set of random effects $\mathbf{u}_j$,

$$y_{ij}|\zeta_{ij} \sim \text{Poisson}(\zeta_{ij})$$

and

$$\zeta_{ij}|\mathbf{u}_j \sim \text{Gamma}(r_{ij}, p_{ij})$$

and

$$\mathbf{u}_j \sim N(\mathbf{0}, \boldsymbol{\Sigma})$$

where y_{ij} is the count response of the ith observation, $i = 1, \dots, n_j$, from the jth cluster, $j = 1, \dots, M$, and r_{ij} and p_{ij} have two different parameterizations, (2) and (3) below. 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.

The probability that a random response y_{ij} takes the value y is then given by

$$\text{Pr}(y_{ij} = y|\mathbf{u}_j) = \frac{\Gamma(y + r_{ij})}{\Gamma(y+1)\Gamma(r_{ij})} p_{ij}^{r_{ij}} (1 - p_{ij})^y \qquad (1)$$

where for convenience we suppress the dependence of the observable data y_{ij} on r_{ij} and p_{ij}.

Model (1) is an extension of the standard negative binomial model (see [R] **nbreg**) to incorporate normally distributed random effects at different hierarchical levels. (The negative binomial model itself can be viewed as a random-effects model, a Poisson model with a gamma-distributed random effect.) The standard negative binomial model is used to model overdispersed count data for which the variance is greater than that of a Poisson model. In a Poisson model, the variance is equal to the mean, and thus overdispersion is defined as the extra variability compared with the mean. According to this definition, the negative binomial model presents two different parameterizations of the overdispersion: the mean parameterization, where the overdispersion is a function of the mean, $1 + \alpha E(Y|\mathbf{x}), \alpha > 0$; and the constant parameterization, where the overdispersion is a constant function, $1 + \delta, \delta \geq 0$. We refer to α and δ as conditional overdispersion parameters.

Let $\mu_{ij} = E(y_{ij}|\mathbf{x}, \mathbf{u_j}) = \exp(\mathbf{x}_{ij}\boldsymbol{\beta}+\mathbf{z}_{ij}\mathbf{u}_j)$, where $\mathbf{x}_{ij}$ is the $1\times p$ row vector of the fixed-effects covariates, analogous to the covariates you would find in a standard negative binomial regression model, with regression coefficients (fixed effects) $\boldsymbol{\beta}$; $\mathbf{z}_{ij}$ is the $1 \times q$ vector of the random-effects covariates 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. One special case 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 $\boldsymbol{\Sigma}$.

Similarly to the standard negative binomial model, we can consider two parameterizations of what we call the conditional overdispersion, the overdispersion conditional on random effects, in a random-effects negative binomial model. For the mean-overdispersion (or, more technically, mean-conditional-overdispersion) parameterization,

$$r_{ij} = 1/\alpha \quad \text{and} \quad p_{ij} = \frac{1}{1+\alpha\mu_{ij}} \tag{2}$$

and the conditional overdispersion is equal to $1 + \alpha\mu_{ij}$. For the constant-overdispersion (or, more technically, constant-conditional-overdispersion) parameterization,

$$r_{ij} = \mu_{ij}/\delta \quad \text{and} \quad p_{ij} = \frac{1}{1+\delta} \tag{3}$$

and the conditional overdispersion is equal to $1 + \delta$. In what follows, for brevity, we will use the term overdispersion parameter to mean conditional overdispersion parameter, unless stated otherwise.

In the context of random-effects negative binomial models, it is important to decide which model is used as a reference model for the definition of the overdispersion. For example, if we consider a corresponding random-effects Poisson model as a comparison model, the parameters α and δ can still be viewed as unconditional overdispersion parameters, as we show below, although the notion of a constant overdispersion is no longer applicable.

If we retain the definition of the overdispersion as the excess variation with respect to a Poisson process for which the variance is equal to the mean, we need to carefully distinguish between the marginal (unconditional) mean with random effects integrated out and the conditional mean given random effects.

In what follows, for simplicity, we omit the dependence of the formulas on $\mathbf{x}$. Conditionally on random effects, the (conditional) dispersion $\text{Var}(y_{ij}|\mathbf{u_j}) = (1+\alpha\mu_{ij})\mu_{ij}$ for the mean parameterization and $\text{Var}(y_{ij}|\mathbf{u_j}) = (1+\delta)\mu_{ij}$ for the constant parameterization; the usual interpretation of the parameters holds (conditionally).

If we consider the marginal mean or, specifically, the marginal dispersion for, for example, a two-level random-intercept model, then

$$\text{Var}(y_{ij}) = \left[1 + \{\exp(\sigma^2)(1+\alpha) - 1\}E(y_{ij})\right] E(y_{ij})$$

for the mean parameterization and

$$\text{Var}(y_{ij}) = \left[1 + \delta + \{\exp(\sigma^2) - 1\}E(y_{ij})\right] E(y_{ij})$$

for the constant parameterization, where σ^2 is the variance component corresponding to the random intercept.

A few things of interest compared with the standard negative binomial model. First, the random-effects negative binomial model is not strictly an overdispersed model. The combination of values of α and σ^2 can lead to an underdispersed model, a model with smaller variability than the Poisson variability. Underdispersed models are not as common in practice, so we will concentrate on the overdispersion in this entry. Second, α (or δ) no longer solely determine the overdispersion and thus cannot be viewed as unconditional overdispersion parameters. Overdispersion is now a function of both α (or δ) and σ^2. Third, the notion of a constant overdispersion is not applicable.

Two special cases are worth mentioning. When $\sigma^2 = 0$, the dispersion reduces to that of a standard negative binomial model. When $\alpha = 0$ (or $\delta = 0$), the dispersion reduces to that of a two-level random-intercept Poisson model, which itself is, in general, an overdispersed model; see Rabe-Hesketh and Skrondal (2012, chap. 13.7) for more details. As such, α and δ retain the typical interpretation as dispersion parameters relative to a random-intercept Poisson model.

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 the *Introduction* of [ME] **mixed**, 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 itself is estimated, 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.

`menbreg` supports three types of Gauss–Hermite quadrature and the Laplacian approximation method; see *Methods and formulas* of [ME] **meglm** for details.

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

Two-level models

▷ Example 1

Rabe-Hesketh and Skrondal (2012, chap. 13.7) analyze the data from Winkelmann (2004) on the impact of the 1997 health reform in Germany on the number of doctor visits. The intent of policymakers was to reduce government expenditures on health care. We use a subsample of the data restricted to 1,158 women who were employed full time the year before or after the reform.

```
. use http://www.stata-press.com/data/r13/drvisits

. describe

Contains data from http://www.stata-press.com/data/r13/drvisits.dta
  obs:         2,227
 vars:             8                          23 Jan 2013 18:39
 size:        71,264
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
id              float   %9.0g                 person id
numvisit        float   %9.0g                 number of doctor visits in the
                                                last 3 months before interview
age             float   %9.0g                 age in years
educ            float   %9.0g                 education in years
married         float   %9.0g                 =1 if married, 0 otherwise
badh            float   %9.0g                 self-reported health status, =1
                                                if bad
loginc          float   %9.0g                 log of household income
reform          float   %9.0g                 =0 if interview before reform, =1
                                                if interview after reform
-------------------------------------------------------------------------------
Sorted by:
```

The dependent variable, `numvisit`, is a count of doctor visits. The covariate of interest is a dummy variable, `reform`, which indicates whether a doctor visit took place before or after the reform. Other covariates include a self-reported health status, age, education, marital status, and a log of household income.

We first fit a two-level random-intercept Poisson model. We specify the random intercept at the `id` level, that is, an individual-person level.

```
. mepoisson numvisit reform age educ married badh loginc || id:, irr
Fitting fixed-effects model:
Iteration 0:   log likelihood = -9326.8542
Iteration 1:   log likelihood = -5989.7308
Iteration 2:   log likelihood = -5942.7581
Iteration 3:   log likelihood = -5942.7243
Iteration 4:   log likelihood = -5942.7243
Refining starting values:
Grid node 0:   log likelihood = -4761.1257
Fitting full model:
Iteration 0:   log likelihood = -4761.1257
Iteration 1:   log likelihood = -4683.2239
Iteration 2:   log likelihood = -4646.9329
Iteration 3:   log likelihood =  -4645.736
Iteration 4:   log likelihood = -4645.7371
Iteration 5:   log likelihood = -4645.7371
Mixed-effects Poisson regression                Number of obs      =      2227
Group variable:             id                  Number of groups   =      1518

                                                Obs per group: min =         1
                                                               avg =       1.5
                                                               max =         2

Integration method: mvaghermite                 Integration points =         7

                                                Wald chi2(6)       =    249.37
Log likelihood = -4645.7371                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
    numvisit |        IRR   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      reform |   .9517026   .0309352    -1.52   0.128     .8929617    1.014308
         age |   1.005821    .002817     2.07   0.038     1.000315    1.011357
        educ |   1.008788   .0127394     0.69   0.488     .9841258    1.034068
     married |   1.082078   .0596331     1.43   0.152     .9712905    1.205503
        badh |   2.471857    .151841    14.73   0.000     2.191471    2.788116
      loginc |   1.094144   .0743018     1.32   0.185     .9577909    1.249909
       _cons |   .5216748   .2668604    -1.27   0.203      .191413    1.421766
-------------+----------------------------------------------------------------
id           |
   var(_cons)|   .8177932   .0503902                       .724761    .9227673
------------------------------------------------------------------------------
LR test vs. Poisson regression:   chibar2(01) =  2593.97 Prob>=chibar2 = 0.0000
. estimates store mepoisson
```

Because we specified the `irr` option, the parameters are reported as incidence-rate ratios. The healthcare reform seems to reduce the expected number of visits by 5% but without statistical significance.

Because we have only one random effect at the `id` level, the table shows only one variance component. The estimate of σ_u^2 is 0.82 with standard error 0.05. The reported likelihood-ratio test shows that there is enough variability between women to favor a mixed-effects Poisson regression over a standard Poisson regression; see *Distribution theory for likelihood-ratio test* in [ME] **me** for a discussion of likelihood-ratio testing of variance components.

It is possible that after conditioning on the person-level random effect, the counts of doctor visits are overdispersed. For example, medical problems occurring during the time period leading to the survey can result in extra doctor visits. We thus reexamine the data with `menbreg`.

```
. menbreg numvisit reform age educ married badh loginc || id:, irr
Fitting fixed-effects model:
Iteration 0:   log likelihood = -4610.7165
Iteration 1:   log likelihood = -4563.4682
Iteration 2:   log likelihood = -4562.3241
Iteration 3:   log likelihood = -4562.3238
Refining starting values:
Grid node 0:   log likelihood = -4643.5216
Fitting full model:
Iteration 0:   log likelihood = -4643.5216  (not concave)
Iteration 1:   log likelihood =  -4555.961
Iteration 2:   log likelihood = -4518.7353
Iteration 3:   log likelihood = -4513.1951
Iteration 4:   log likelihood = -4513.1853
Iteration 5:   log likelihood = -4513.1853
Mixed-effects nbinomial regression              Number of obs      =      2227
Overdispersion:              mean
Group variable:                id               Number of groups   =      1518

                                                Obs per group: min =         1
                                                               avg =       1.5
                                                               max =         2

Integration method: mvaghermite                 Integration points =         7

                                                Wald chi2(6)       =    237.35
Log likelihood = -4513.1853                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
    numvisit |        IRR   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      reform |   .9008536    .042022    -2.24   0.025     .8221449    .9870975
         age |   1.003593   .0028206     1.28   0.202     .9980799    1.009137
        educ |   1.007026    .012827     0.55   0.583     .9821969    1.032483
     married |   1.089597    .064213     1.46   0.145      .970738    1.223008
        badh |   3.043562   .2366182    14.32   0.000     2.613404    3.544523
      loginc |   1.136342   .0867148     1.67   0.094     .9784833    1.319668
       _cons |   .5017199    .285146    -1.21   0.225     .1646994    1.528377
-------------+----------------------------------------------------------------
    /lnalpha |  -.7962692   .1190614    -6.69   0.000    -1.029625   -.5629132
-------------+----------------------------------------------------------------
id           |
   var(_cons)|   .4740088   .0582404                      .3725642    .6030754
------------------------------------------------------------------------------
LR test vs. nbinomial regression:chibar2(01) =    98.28 Prob>=chibar2 = 0.0000
```

The estimated effect of the healthcare reform now corresponds to the reduction in the number of doctor visits by 10%—twice as much compared with the Poisson model—and this effect is significant at the 5% level.

The estimate of the variance component σ_u^2 drops down to 0.47 compared with `mepoisson`, which is not surprising given that now we have an additional parameter that controls the variability of the data.

Because the conditional overdispersion α is assumed to be greater than 0, it is parameterized on the log scale, and its log estimate is reported as `/lnalpha` in the output. In our model, $\widehat{\alpha} = \exp(-0.80) = 0.45$. We can also compute the unconditional overdispersion in this model by using the corresponding formula in the *Introduction* above: $\exp(.47) \times (1 + .45) - 1 = 1.32$.

The reported likelihood-ratio test shows that there is enough variability between women to favor a mixed-effects negative binomial regression over negative binomial regression without random effects.

We can also perform a likelihood-ratio test comparing the mixed-effects negative binomial model to the mixed-effects Poisson model. Because we are comparing two different estimators, we need to use the `force` option with `lrtest`. In general, there is no guarantee as to the validity or interpretability of the resulting likelihood-ratio test, but in our case we know the test is valid because the mixed-effects Poisson model is nested within the mixed-effects negative binomial model.

```
. lrtest mepoisson ., force
Likelihood-ratio test                                 LR chi2(1)  =    265.10
(Assumption: mepoisson nested in .)                   Prob > chi2 =    0.0000
Note: The reported degrees of freedom assumes the null hypothesis is not on
      the boundary of the parameter space.  If this is not true, then the
      reported test is conservative.
```

The reported likelihood-ratio test favors the mixed-effects negative binomial model. The reported test is conservative because the test of $H_0 : \alpha = 0$ occurs on the boundary of the parameter space; see *Distribution theory for likelihood-ratio test* in [ME] **me** for details.

◁

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). To demonstrate a three-level model, we revisit example 2 from [ME] **meqrpoisson**.

▷ Example 2

Rabe-Hesketh and Skrondal (2012, exercise 13.7) describe data from the *Atlas of Cancer Mortality in the European Economic Community* (EEC) (Smans, Mair, and Boyle 1993). The data were analyzed in Langford, Bentham, and McDonald (1998) and record the number of deaths among males due to malignant melanoma during 1971–1980.

```
. use http://www.stata-press.com/data/r13/melanoma
(Skin cancer (melanoma) data)
. describe
Contains data from http://www.stata-press.com/data/r13/melanoma.dta
  obs:           354                          Skin cancer (melanoma) data
 vars:             6                          30 May 2013 17:10
 size:         4,956                          (_dta has notes)
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
nation          byte    %11.0g     n          Nation ID
region          byte    %9.0g                 Region ID: EEC level-I areas
county          int     %9.0g                 County ID: EEC level-II/level-III
                                                areas
deaths          int     %9.0g                 No. deaths during 1971-1980
expected        float   %9.0g                 No. expected deaths
uv              float   %9.0g                 UV dose, mean-centered
-------------------------------------------------------------------------------
Sorted by:
```

Nine European nations (variable `nation`) are represented, and data were collected over geographical regions defined by EEC statistical services as level I areas (variable `region`), with deaths being recorded for each of 354 counties, which are level II or level III EEC-defined areas (variable `county`, which identifies the observations). Counties are nested within regions, and regions are nested within nations.

The variable deaths records the number of deaths for each county, and expected records the expected number of deaths (the exposure) on the basis of crude rates for the combined countries. The variable uv is a measure of exposure to ultraviolet (UV) radiation.

In example 2 of [ME] meqrpoisson, we noted that because counties also identified the observations, we could model overdispersion by using a four-level Poisson model with a random intercept at the county level. Here we fit a three-level negative binomial model with the default mean-dispersion parameterization.

```
. menbreg deaths uv, exposure(expected) || nation: || region:
Fitting fixed-effects model:
Iteration 0:   log likelihood =  -1361.855
Iteration 1:   log likelihood = -1230.0211
Iteration 2:   log likelihood =  -1211.049
Iteration 3:   log likelihood = -1202.5641
Iteration 4:   log likelihood = -1202.5329
Iteration 5:   log likelihood = -1202.5329
Refining starting values:
Grid node 0:   log likelihood = -1209.6951
Fitting full model:
Iteration 0:   log likelihood = -1209.6951  (not concave)
  (output omitted )
Iteration 11:  log likelihood = -1086.3902
Mixed-effects nbinomial regression              Number of obs      =       354
Overdispersion:            mean

-----------------------------------------------------------
                |   No. of       Observations per Group
 Group Variable |   Groups    Minimum    Average    Maximum
----------------+------------------------------------------
         nation |        9          3       39.3         95
         region |       78          1        4.5         13
-----------------------------------------------------------

Integration method: mvaghermite                 Integration points =         7

                                                Wald chi2(1)       =      8.73
Log likelihood = -1086.3902                     Prob > chi2        =    0.0031

------------------------------------------------------------------------------
      deaths |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          uv |  -.0335933   .0113725    -2.95   0.003     -.055883   -.0113035
       _cons |  -.0790606   .1295931    -0.61   0.542    -.3330583    .1749372
ln(expected) |          1  (exposure)
-------------+----------------------------------------------------------------
    /lnalpha |  -4.182603   .3415036   -12.25   0.000    -4.851937   -3.513268
-------------+----------------------------------------------------------------
nation       |
   var(_cons)|   .1283614   .0678971                      .0455187    .3619758
-------------+----------------------------------------------------------------
nation>
region       |
   var(_cons)|   .0401818   .0104855                      .0240938     .067012
------------------------------------------------------------------------------
LR test vs. nbinomial regression:    chi2(2) =   232.29   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

The estimates are very close to those of meqrpoisson. The conditional overdispersion in our model is $\hat{\alpha} = \exp(-4.18) = 0.0153$. It is in agreement with the estimate of the random intercept at the county level, 0.0146, in a four-level random-effects Poisson model reported by meqrpoisson. Because the negative binomial is a three-level model, we gained some computational efficiency over the four-level Poisson model.

◁

Stored results

menbreg 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)	menbreg
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(covariates)	list of covariates
e(ivars)	grouping variables
e(model)	nbreg
e(title)	title in estimation output
e(link)	log
e(family)	nbinomial
e(clustvar)	name of cluster variable
e(dispersion)	mean or constant
e(offset)	offset
e(exposure)	exposure variable
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

Without a loss of generality, consider a two-level negative binomial model. For cluster j, $j = 1,\dots,M$, the conditional distribution of $\mathbf{y}_j = (y_{j1},\dots,y_{jn_j})'$, given a set of cluster-level random effects $\mathbf{u}_j$ and the conditional overdispersion parameter α in a mean-overdispersion parameterization, is

$$
\begin{aligned}
f(\mathbf{y}_j|\mathbf{u}_j,\alpha) &= \prod_{i=1}^{n_j} \left\{ \frac{\Gamma(y_{ij}+r)}{\Gamma(y_{ij}+1)\Gamma(r)} p_{ij}^r (1-p_{ij})^{y_{ij}} \right\} \\
&= \exp\left[\sum_{i=1}^{n_j} \{\log\Gamma(y_{ij}+r) - \log\Gamma(y_{ij}+1) - \log\Gamma(r) + c(y_{ij},\alpha)\} \right]
\end{aligned}
$$

where $c(y_{ij}, \alpha)$ is defined as

$$-\frac{1}{\alpha}\log\{1+\exp(\eta_{ij}+\log\alpha)\} - y_{ij}\log\{1+\exp(-\eta_{ij}-\log\alpha)\}$$

and $r = 1/\alpha$, $p_{ij} = 1/(1+\alpha\mu_{ij})$, and $\eta_{ij} = \mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j$.

For the constant-overdispersion parameterization with the conditional overdispersion parameter δ, the conditional distribution of $\mathbf{y}_j$ is

$$\begin{aligned} f(\mathbf{y}_j|\mathbf{u}_j,\delta) &= \prod_{i=1}^{n_j}\left\{\frac{\Gamma(y_{ij}+r_{ij})}{\Gamma(y_{ij}+1)\Gamma(r_{ij})}p^{r_{ij}}(1-p)^{y_{ij}}\right\} \\ &= \exp\left[\sum_{i=1}^{n_j}\{\log\Gamma(y_{ij}+r_{ij}) - \log\Gamma(y_{ij}+1) - \log\Gamma(r_{ij}) + c(y_{ij},\delta)\}\right] \end{aligned}$$

where $c(y_{ij}, \delta)$ is defined as

$$-\left(\frac{\mu_{ij}}{\delta}+y_{ij}\right)\log(1+\delta) + y_{ij}\log\delta$$

and $r_{ij} = \mu_{ij}/\delta$ and $p = 1/(1+\delta)$.

For conciseness, let γ denote either conditional overdispersion parameter. 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 $f(\mathbf{y}_j, \mathbf{u}_j, \gamma)$,

$$\begin{aligned} \mathcal{L}_j(\boldsymbol{\beta},\boldsymbol{\Sigma},\gamma) &= (2\pi)^{-q/2}\,|\boldsymbol{\Sigma}|^{-1/2}\int f(\mathbf{y}_j|\mathbf{u}_j,\gamma)\exp\left(-\mathbf{u}_j'\boldsymbol{\Sigma}^{-1}\mathbf{u}_j/2\right)d\mathbf{u}_j \\ &= (2\pi)^{-q/2}\,|\boldsymbol{\Sigma}|^{-1/2}\int \exp\{h(\boldsymbol{\beta},\boldsymbol{\Sigma},\mathbf{u}_j,\gamma)\}\,d\mathbf{u}_j \end{aligned} \tag{4}$$

where

$$h(\boldsymbol{\beta},\boldsymbol{\Sigma},\mathbf{u}_j,\gamma) = f(\mathbf{y}_j|\mathbf{u}_j,\gamma) - \mathbf{u}_j'\boldsymbol{\Sigma}^{-1}\mathbf{u}_j/2$$

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

The integration in (4) has no closed form and thus must be approximated. `menbreg` 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 (4) 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, $\mathcal{L}(\boldsymbol{\beta},\boldsymbol{\Sigma},\gamma) = \sum_{j=1}^{M}\mathcal{L}_j(\boldsymbol{\beta},\boldsymbol{\Sigma},\gamma)$.

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

References

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

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

Langford, I. H., G. Bentham, and A. McDonald. 1998. Multi-level modelling of geographically aggregated health data: A case study on malignant melanoma mortality and UV exposure in the European community. *Statistics in Medicine* 17: 41–57.

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

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

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.

Smans, M., C. S. Mair, and P. Boyle. 1993. *Atlas of Cancer Mortality in the European Economic Community*. Lyon, France: IARC Scientific Publications.

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

Winkelmann, R. 2004. Health care reform and the number of doctor visits—An econometric analysis. *Journal of Applied Econometrics* 19: 455–472.

Also see

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

[ME] **mepoisson** — Multilevel mixed-effects Poisson regression

[ME] **meqrpoisson** — Multilevel mixed-effects Poisson regression (QR decomposition)

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

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

[XT] **xtnbreg** — Fixed-effects, random-effects, & population-averaged negative binomial models

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

Title

menbreg postestimation — Postestimation tools for menbreg

Description

The following postestimation command is of special interest after `menbreg`:

Command	Description
`estat group`	summarize the composition of the nested groups

The following standard postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat ic`	Akaike's and Schwarz's Bayesian information criteria (AIC and BIC)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

Special-interest postestimation commands

`estat group` reports the number of groups and minimum, average, and maximum group sizes for each level of the model. Model levels are identified by the corresponding group variable in the data. Because groups are treated as nested, the information in this summary may differ from what you would get if you used the `tabulate` command on each group variable individually.

Syntax for predict

Syntax for obtaining predictions of random effects and their standard errors

`predict` [*type*] *newvarsspec* [*if*] [*in*] , { `remeans` | `remodes` } [`reses(`*newvarsspec*`)`]

Syntax for obtaining other predictions

`predict` [*type*] *newvarsspec* [*if*] [*in*] [, *statistic* *options*]

newvarsspec is *stub*`*` or *newvarlist*.

statistic	Description
Main	
`mu`	number of events; the default
`fitted`	fitted linear predictor
`xb`	linear predictor for the fixed portion of the model only
`stdp`	standard error of the fixed-portion linear prediction
`pearson`	Pearson residuals
`deviance`	deviance residuals
`anscombe`	Anscombe residuals

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

options	Description
Main	
`means`	compute *statistic* using empirical Bayes means; the default
`modes`	compute *statistic* using empirical Bayes modes
`nooffset`	ignore the offset or exposure variable in calculating predictions; relevant only if you specified `offset()` or `exposure()` when you fit the model
`fixedonly`	prediction for the fixed portion of the model only
Integration	
`intpoints(#)`	use # quadrature points to compute empirical Bayes means
`iterate(#)`	set maximum number of iterations in computing statistics involving empirical Bayes estimators
`tolerance(#)`	set convergence tolerance for computing statistics involving empirical Bayes estimators

Menu for predict

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

remeans, remodes, reses(); see [ME] **meglm postestimation**.

mu, the default, calculates the predicted mean (the predicted number of events), that is, the inverse link function applied to the linear prediction. By default, this is based on a linear predictor that includes both the fixed effects and the random effects, and the predicted mean is conditional on the values of the random effects. Use the fixedonly option if you want predictions that include only the fixed portion of the model, that is, if you want random effects set to 0.

fitted, xb, stdp, pearson, deviance, anscombe, means, modes, nooffset, fixedonly; see [ME] **meglm postestimation**.

By default or if the means option is specified, statistics mu, pr, fitted, xb, stdp, pearson, deviance, and anscombe are based on the posterior mean estimates of random effects. If the modes option is specified, these statistics are based on the posterior mode estimates of random effects.

Integration

intpoints(), iterate(), tolerance(); see [ME] **meglm postestimation**.

Syntax for estat group

```
estat group
```

Menu for estat

Statistics > Postestimation > Reports and statistics

Remarks and examples

Various predictions, statistics, and diagnostic measures are available after fitting a mixed-effects negative binomial model with menbreg. For the most part, calculation centers around obtaining estimates of the subject/group-specific random effects. Random effects are not estimated when the model is fit but instead need to be predicted after estimation.

Here we show a short example of predicted counts and predicted random effects; refer to [ME] **meglm postestimation** for additional examples applicable to mixed-effects generalized linear models.

▷ Example 1

In example 2 of [ME] **menbreg**, we modeled the number of deaths among males in nine European nations as a function of exposure to ultraviolet radiation (uv). We used a three-level negative binomial model with random effects at the nation and region levels.

```
. use http://www.stata-press.com/data/r13/melanoma
(Skin cancer (melanoma) data)
. menbreg deaths uv, exposure(expected) || nation: || region:
  (output omitted)
```

We can use `predict` to obtain the predicted counts as well as the estimates of the random effects at the nation and region levels.

```
. predict mu
(predictions based on fixed effects and posterior means of random effects)
(option mu assumed)
(using 7 quadrature points)
. predict re_nat re_reg, remeans
(calculating posterior means of random effects)
(using 7 quadrature points)
```

Stata displays a note that the predicted values of `mu` are based on the posterior means of random effects. You can use option `modes` to obtain predictions based on the posterior modes of random effects.

Here we list the data for the first nation in the dataset, which happens to be Belgium:

```
. list nation region deaths mu re_nat re_reg if nation==1, sepby(region)
```

	nation	region	deaths	mu	re_nat	re_reg
1.	Belgium	1	79	64.4892	-.0819939	.2937711
2.	Belgium	2	80	77.64736	-.0819939	.024005
3.	Belgium	2	51	44.56528	-.0819939	.024005
4.	Belgium	2	43	53.10434	-.0819939	.024005
5.	Belgium	2	89	65.35963	-.0819939	.024005
6.	Belgium	2	19	35.18457	-.0819939	.024005
7.	Belgium	3	19	8.770186	-.0819939	-.3434432
8.	Belgium	3	15	43.95521	-.0819939	-.3434432
9.	Belgium	3	33	34.17878	-.0819939	-.3434432
10.	Belgium	3	9	7.332448	-.0819939	-.3434432
11.	Belgium	3	12	12.93873	-.0819939	-.3434432

We can see that the predicted random effects at the nation level, `re_nat`, are the same for all the observations. Similarly, the predicted random effects at the region level, `re_reg`, are the same within each region. The predicted counts, `mu`, are not as close to the observed deaths as the predicted counts from the mixed-effects Poisson model in example 1 of [ME] **mepoisson postestimation**.

◁

Methods and formulas

Methods and formulas for predicting random effects and other statistics are given in *Methods and formulas* of [ME] **meglm postestimation**.

Also see

[ME] **menbreg** — Multilevel mixed-effects negative binomial regression

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

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

Title

meologit — Multilevel mixed-effects ordered logistic regression

Syntax

`meologit` *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	
`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

options	Description
Model	
constraints(*constraints*)	apply specified linear constraints
collinear	keep collinear variables
SE/Robust	
vce(*vcetype*)	*vcetype* may be oim, robust, or cluster *clustvar*
Reporting	
level(#)	set confidence level; default is level(95)
or	report fixed-effects coefficients as odds ratios
nocnsreport	do not display constraints
notable	suppress coefficient table
noheader	suppress output header
nogroup	suppress table summarizing groups
nolrtest	do not perform likelihood-ratio test comparing with ordered 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	
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
coeflegend	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
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 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 > Ordered logistic regression

Description

meologit fits mixed-effects logistic models for ordered responses. The actual values taken on by the response are irrelevant except that larger values are assumed to correspond to "higher" outcomes. The conditional distribution of the response given the random effects is assumed to be multinomial, with success probability determined by the logistic cumulative distribution function.

Options

Model

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*$[k, l]$.

noconstant suppresses the constant (intercept) term; may be specified for any or all of the random-effects equations.

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

display_options: 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 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 `meologit` 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 `meologit` 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**.

`meologit` is a convenience command for `meglm` with a `logit` link and an `ordinal` family; see [ME] **meglm**.

Remarks are presented under the following headings:

Introduction

Mixed-effects ordered logistic regression is ordered logistic regression containing both fixed effects and random effects. An ordered response is a variable that is categorical and ordered, for instance, “poor”, “good”, and “excellent”, which might indicate a person’s current health status or the repair record of a car. In the absence of random effects, mixed-effects ordered logistic regression reduces to ordered logistic regression; see [R] **ologit**.

Comprehensive treatments of mixed models are provided by, for example, Searle, Casella, and McCulloch (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). Agresti (2010, chap. 10) and Rabe-Hesketh and Skrondal (2012, chap. 11) are good introductory readings on applied multilevel modeling of ordinal data.

meologit allows for 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.

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 fixed effects $\mathbf{x}_{ij}$, a set of cutpoints $\boldsymbol{\kappa}$, and a set of random effects $\mathbf{u}_j$, the cumulative probability of the response being in a category higher than k is

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

for $j = 1, \ldots, M$ clusters, with cluster j consisting of $i = 1, \ldots, n_j$ observations. The cutpoints $\boldsymbol{\kappa}$ are labeled κ_1, κ_2, ..., κ_{K-1}, where K is the number of possible outcomes. $H(\cdot)$ is the logistic cumulative distribution function that represents cumulative probability.

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) $\boldsymbol{\beta}$. In our parameterization, $\mathbf{x}_{ij}$ does not contain a constant term because its effect is absorbed into the cutpoints. 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 $\boldsymbol{\Sigma}$.

From (1), we can derive the probability of observing outcome k as

$$\begin{aligned}
\Pr(y_{ij} = k|\boldsymbol{\kappa}, \mathbf{u}_j) &= \Pr(\kappa_{k-1} < \mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j + \epsilon_{ij} \le \kappa_k) \\
&= \Pr(\kappa_{k-1} - \mathbf{x}_{ij}\boldsymbol{\beta} - \mathbf{z}_{ij}\mathbf{u}_j < \epsilon_{ij} \le \kappa_k - \mathbf{x}_{ij}\boldsymbol{\beta} - \mathbf{z}_{ij}\mathbf{u}_j) \\
&= H(\kappa_k - \mathbf{x}_{ij}\boldsymbol{\beta} - \mathbf{z}_{ij}\mathbf{u}_j) - H(\kappa_{k-1} - \mathbf{x}_{ij}\boldsymbol{\beta} - \mathbf{z}_{ij}\mathbf{u}_j)
\end{aligned}$$

where κ_0 is taken as $-\infty$ and κ_K is taken as $+\infty$.

From the above, we may also write the model in terms of a latent linear response, where observed ordinal responses y_{ij} are generated from the latent continuous responses, such that

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

and

$$y_{ij} = \begin{cases} 1 & \text{if} \quad y_{ij}^* \le \kappa_1 \\ 2 & \text{if} \quad \kappa_1 < y_{ij}^* \le \kappa_2 \\ \vdots & \\ K & \text{if} \quad \kappa_{K-1} < y_{ij}^* \end{cases}$$

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 the *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 itself is estimated, 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.

`meologit` supports three types of Gauss–Hermite quadrature and the Laplacian approximation method; see *Methods and formulas* of [ME] **meglm** for details.

Below we present two short examples of mixed-effects ordered logistic regression; refer to [ME] **melogit** for additional examples including crossed random-effects models and to [ME] **me** and [ME] **meglm** for examples of other 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 ordered logistic regression; see [R] **ologit**.

▷ Example 1

We use the data from the Television, School, and Family Smoking Prevention and Cessation Project (Flay et al. 1988; Rabe-Hesketh and Skrondal 2012, chap. 11), where schools were randomly assigned into one of four groups defined by two treatment variables. Students within each school are nested in classes, and classes are nested in schools. In this example, we ignore the variability of classes within schools and fit a two-level model; we incorporate classes in a three-level model in example 2. The dependent variable is the tobacco and health knowledge (THK) scale score collapsed into four ordered categories. We regress the outcome on the treatment variables and their interaction and control for the pretreatment score.

```
. use http://www.stata-press.com/data/r13/tvsfpors
. meologit thk prethk cc##tv || school:
Fitting fixed-effects model:
Iteration 0:   log likelihood =  -2212.775
Iteration 1:   log likelihood =  -2125.509
Iteration 2:   log likelihood = -2125.1034
Iteration 3:   log likelihood = -2125.1032
Refining starting values:
Grid node 0:   log likelihood = -2136.2426
Fitting full model:
Iteration 0:   log likelihood = -2136.2426  (not concave)
Iteration 1:   log likelihood = -2120.2577
Iteration 2:   log likelihood = -2119.7574
Iteration 3:   log likelihood = -2119.7428
Iteration 4:   log likelihood = -2119.7428
Mixed-effects ologit regression                 Number of obs      =      1600
Group variable:           school                Number of groups   =        28
                                                Obs per group: min =        18
                                                               avg =      57.1
                                                               max =       137
Integration method: mvaghermite                 Integration points =         7
                                                Wald chi2(4)       =    128.06
Log likelihood = -2119.7428                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
         thk |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      prethk |   .4032892     .03886    10.38   0.000      .327125    .4794534
        1.cc |   .9237904    .204074     4.53   0.000     .5238127    1.323768
        1.tv |   .2749937   .1977424     1.39   0.164    -.1125744    .6625618
             |
       cc#tv |
        1 1  |  -.4659256   .2845963    -1.64   0.102    -1.023724    .0918728
-------------+----------------------------------------------------------------
       /cut1 |  -.0884493   .1641062    -0.54   0.590    -.4100916     .233193
       /cut2 |   1.153364    .165616     6.96   0.000     .8287625    1.477965
       /cut3 |    2.33195   .1734199    13.45   0.000     1.992053    2.671846
-------------+----------------------------------------------------------------
school       |
   var(_cons)|   .0735112   .0383106                      .0264695    .2041551
------------------------------------------------------------------------------
LR test vs. ologit regression:   chibar2(01) =    10.72 Prob>=chibar2 = 0.0005
```

Those of you familiar with the `mixed` command or other `me` commands will recognize the syntax and output. Below we comment on the items specific to ordered outcomes.

1. The estimation table reports the fixed effects, the estimated cutpoints $(\kappa_1, \kappa_2, \kappa_3)$, and the estimated variance components. The fixed effects can be interpreted just as you would the output from `ologit`. We find that students with higher preintervention scores tend to have higher postintervention scores. Because of their interaction, the impact of the treatment variables on the knowledge score is not straightforward; we defer this discussion to example 1 of [ME] **meologit postestimation**. You can also specify the `or` option at estimation or on replay to display the fixed effects as odds ratios instead.

2. Underneath the fixed effects and the cutpoints, the table shows the estimated variance components. The random-effects equation is labeled `school`, meaning that these are random effects at the `school` 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 0.07 with standard error 0.04. The reported likelihood-ratio test

shows that there is enough variability between schools to favor a mixed-effects ordered logistic regression over a standard ordered logistic regression; see *Distribution theory for likelihood-ratio test* in [ME] **me** for a discussion of likelihood-ratio testing of variance components.

We now store our estimates for later use.

```
. estimates store r_2
```

◁

Three-level models

Two-level models extend naturally to models with three or more levels with nested random effects. Below we continue with example 1.

▷ Example 2

In this example, we fit a three-level model incorporating classes nested within schools as an additional level. The fixed-effects part remains the same.

```
. meologit thk prethk cc##tv || school: || class:
Fitting fixed-effects model:
Iteration 0:   log likelihood =  -2212.775
Iteration 1:   log likelihood =  -2125.509
Iteration 2:   log likelihood = -2125.1034
Iteration 3:   log likelihood = -2125.1032
Refining starting values:
Grid node 0:   log likelihood = -2152.1514
Fitting full model:
Iteration 0:   log likelihood = -2152.1514  (not concave)
Iteration 1:   log likelihood = -2125.9213  (not concave)
Iteration 2:   log likelihood = -2120.1861
Iteration 3:   log likelihood = -2115.6177
Iteration 4:   log likelihood = -2114.5896
Iteration 5:   log likelihood = -2114.5881
Iteration 6:   log likelihood = -2114.5881
Mixed-effects ologit regression                 Number of obs     =      1600

                  No. of       Observations per Group
 Group Variable   Groups    Minimum    Average    Maximum

         school       28         18       57.1        137
          class      135          1       11.9         28

Integration method: mvaghermite                 Integration points =        7

                                                Wald chi2(4)      =    124.39
Log likelihood = -2114.5881                     Prob > chi2       =    0.0000

         thk        Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]

      prethk     .4085273    .039616    10.31   0.000     .3308814    .4861731
        1.cc     .8844369   .2099124     4.21   0.000     .4730161    1.295858
        1.tv      .236448   .2049065     1.15   0.249    -.1651614    .6380575

       cc#tv
        1 1    -.3717699   .2958887    -1.26   0.209     -.951701    .2081612

       /cut1    -.0959459   .1688988    -0.57   0.570    -.4269815    .2350896
       /cut2     1.177478   .1704946     6.91   0.000     .8433151    1.511642
       /cut3     2.383672   .1786736    13.34   0.000     2.033478    2.733865

school
   var(_cons)    .0448735   .0425387                      .0069997    .2876749

school>class
   var(_cons)    .1482157   .0637521                       .063792    .3443674

LR test vs. ologit regression:     chi2(2) =    21.03   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

Notes:

1. Our model now has two random-effects equations, separated by `||`. The first is a random intercept (constant only) at the `school` level (level three), and the second is a random intercept at the `class` level (level two). The order in which these are specified (from left to right) is significant—`meologit` assumes that `class` is nested within `school`.

2. The information on groups is now displayed as a table, with one row for each grouping. You can suppress this table with the nogroup or the noheader option, which will suppress the rest of the header as well.
3. The variance-component estimates are now organized and labeled according to level. The variance component for class is labeled school>class to emphasize that classes are nested within schools.

Compared with the two-level model from example 1, the estimate of the variance of the random intercept at the school level dropped from 0.07 to 0.04. This is not surprising because we now use two random components versus one random component to account for unobserved heterogeneity among students. We can use lrtest and our stored estimation result from example 1 to see which model provides a better fit:

```
. lrtest r_2 .

Likelihood-ratio test                                 LR chi2(1)  =     10.31
(Assumption: r_2 nested in .)                         Prob > chi2 =    0.0013

Note: The reported degrees of freedom assumes the null hypothesis is not on
      the boundary of the parameter space.  If this is not true, then the
      reported test is conservative.
```

The likelihood-ratio test favors the three-level model. For more information about the likelihood-ratio test in the context of mixed-effects models, see *Distribution theory for likelihood-ratio test* in [ME] **me**.

◁

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

meologit 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_cat)	number of categories
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)	meologit
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(covariates)	list of covariates
e(ivars)	grouping variables
e(model)	ologit
e(title)	title in estimation output
e(link)	logit
e(family)	ordinal
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
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

Without a loss of generality, consider a two-level ordered logistic model. The probability of observing outcome k for response y_{ij} is then

$$\begin{aligned} p_{ij} = \Pr(y_{ij} = k|\boldsymbol{\kappa}, \mathbf{u}_j) &= \Pr(\kappa_{k-1} < \boldsymbol{\eta}_{ij} + \epsilon_{it} \leq \kappa_k) \\ &= \frac{1}{1+\exp(-\kappa_k + \boldsymbol{\eta}_{ij})} - \frac{1}{1+\exp(-\kappa_{k-1} + \boldsymbol{\eta}_{ij})} \end{aligned}$$

where $\boldsymbol{\eta}_{ij} = \mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j + \text{offset}_{ij}$, κ_0 is taken as $-\infty$, and κ_K is taken as $+\infty$. Here $\mathbf{x}_{ij}$ does not contain a constant term because its effect is absorbed into the cutpoints.

For cluster j, $j = 1, \dots, M$, the conditional distribution of $\mathbf{y}_j = (y_{j1}, \dots, y_{jn_j})'$ given a set of cluster-level random effects $\mathbf{u}_j$ is

$$
\begin{aligned}
f(\mathbf{y}_j|\boldsymbol{\kappa},\mathbf{u}_j) &= \prod_{i=1}^{n_j} p_{ij}^{I_k(y_{ij})} \\
&= \exp\sum_{i=1}^{n_j}\left\{I_k(y_{ij})\,\log(p_{ij})\right\}
\end{aligned}
$$

where

$$
I_k(y_{ij}) = \begin{cases} 1 & \text{if } y_{ij}=k \\ 0 & \text{otherwise} \end{cases}
$$

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 $f(\mathbf{y}_j,\mathbf{u}_j)$,

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

where

$$
h\left(\boldsymbol{\beta},\boldsymbol{\kappa},\boldsymbol{\Sigma},\mathbf{u}_j\right) = \sum_{i=1}^{n_j}\left\{I_k(y_{ij})\,\log(p_{ij})\right\} - \mathbf{u}_j'\boldsymbol{\Sigma}^{-1}\mathbf{u}_j/2
$$

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. `meologit` 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 $h\left(\boldsymbol{\beta},\boldsymbol{\kappa},\boldsymbol{\Sigma},\mathbf{u}_j\right)$ 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, $\mathcal{L}(\boldsymbol{\beta},\boldsymbol{\kappa},\boldsymbol{\Sigma}) = \sum_{j=1}^{M}\mathcal{L}_j(\boldsymbol{\beta},\boldsymbol{\kappa},\boldsymbol{\Sigma})$.

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

References

Agresti, A. 2010. *Analysis of Ordinal Categorical Data*. 2nd ed. Hoboken, NJ: Wiley.

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

Flay, B. R., B. R. Brannon, C. A. Johnson, W. B. Hansen, A. L. Ulene, D. A. Whitney-Saltiel, L. R. Gleason, S. Sussman, M. D. Gavin, K. M. Glowacz, D. F. Sobol, and D. C. Spiegel. 1988. The television, school, and family smoking cessation and prevention project: I. Theoretical basis and program development. *Preventive Medicine* 17: 585–607.

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

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

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

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.

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

Also see

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

[ME] **meoprobit** — Multilevel mixed-effects ordered probit regression

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

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

[XT] **xtologit** — Random-effects ordered logistic models

[U] 20 Estimation and postestimation commands

Title

meologit postestimation — Postestimation tools for meologit

Description

The following postestimation command is of special interest after `meologit`:

Command	Description
`estat group`	summarize the composition of the nested groups

The following standard postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat ic`	Akaike's and Schwarz's Bayesian information criteria (AIC and BIC)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

Special-interest postestimation commands

`estat group` reports the number of groups and minimum, average, and maximum group sizes for each level of the model. Model levels are identified by the corresponding group variable in the data. Because groups are treated as nested, the information in this summary may differ from what you would get if you used the `tabulate` command on each group variable individually.

Syntax for predict

Syntax for obtaining predictions of random effects and their standard errors

predict [*type*] *newvarsspec* [*if*] [*in*] , { remeans | remodes } [reses(*newvarsspec*)]

Syntax for obtaining other predictions

predict [*type*] *newvarsspec* [*if*] [*in*] [, *statistic options*]

newvarsspec is *stub** or *newvarlist*.

statistic	Description
Main	
pr	predicted probabilities; the default
fitted	fitted linear predictor
xb	linear predictor for the fixed portion of the model only
stdp	standard error of the fixed-portion linear prediction

These statistics are available both in and out of sample; type predict ... if e(sample) ... if wanted only for the estimation sample.

options	Description
Main	
means	compute *statistic* using empirical Bayes means; the default
modes	compute *statistic* using empirical Bayes modes
nooffset	ignore the offset variable in calculating predictions; relevant only if you specified offset() when you fit the model
fixedonly	prediction for the fixed portion of the model only
outcome(*outcome*)	outcome category for predicted probabilities
Integration	
intpoints(#)	use # quadrature points to compute empirical Bayes means
iterate(#)	set maximum number of iterations in computing statistics involving empirical Bayes estimators
tolerance(#)	set convergence tolerance for computing statistics involving empirical Bayes estimators

You specify one or k new variables in *newvarlist* with pr, where k is the number of outcomes. If you do not specify outcome(), those options assume outcome(#1).

Menu for predict

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

remeans, remodes, reses(); see [ME] **meglm postestimation**.

pr, the default, calculates the predicted probabilities. By default, the probabilities are based on a linear predictor that includes both the fixed effects and the random effects, and the predicted probabilities are conditional on the values of the random effects. Use the fixedonly option if you want predictions that include only the fixed portion of the model, that is, if you want random effects set to 0.

You specify one or k new variables, where k is the number of categories of the dependent variable. If you specify the outcome() option, the probabilities will be predicted for the requested outcome only, in which case you specify only one new variable. If you specify one new variable and do not specify outcome(), outcome(#1) is assumed.

fitted, xb, stdp, means, modes, nooffset, fixedonly; see [ME] **meglm postestimation**.

By default or if the means option is specified, statistics pr, fitted, xb, and stdp are based on the posterior mean estimates of random effects. If the modes option is specified, these statistics are based on the posterior mode estimates of random effects.

outcome(*outcome*) specifies the outcome for which the predicted probabilities are to be calculated. outcome() should contain either one value of the dependent variable or one of #1, #2, ..., with #1 meaning the first category of the dependent variable, #2 meaning the second category, etc.

Integration

intpoints(), iterate(), tolerance(); see [ME] **meglm postestimation**.

Syntax for estat group

```
estat group
```

Menu for estat

Statistics > Postestimation > Reports and statistics

Remarks and examples

Various predictions, statistics, and diagnostic measures are available after fitting an ordered logistic mixed-effects model with meologit. Here we show a short example of predicted probabilities and predicted random effects; refer to [ME] **meglm postestimation** for additional examples applicable to mixed-effects generalized linear models.

▷ Example 1

In example 2 of [ME] **meologit**, we modeled the tobacco and health knowledge (thk) score—coded 1, 2, 3, 4—among students as a function of two treatments (cc and tv) by using a three-level ordered logistic model with random effects at the school and class levels.

```
. use http://www.stata-press.com/data/r13/tvsfpors
. meologit thk prethk cc##tv || school: || class:
  (output omitted )
```

We obtain predicted probabilities for all four outcomes based on the contribution of both fixed effects and random effects by typing

```
. predict pr*
(predictions based on fixed effects and posterior means of random effects)
(option mu assumed)
(using 7 quadrature points)
```

As the note says, the predicted values are based on the posterior means of random effects. You can use the modes option to obtain predictions based on the posterior modes of random effects.

Because we specified a stub name, Stata saved the predicted random effects in variables pr1 through pr4. Here we list the predicted probabilities for the first two classes for school 515:

```
. list class thk pr? if school==515 & (class==515101 | class==515102),
> sepby(class)
```

	class	thk	pr1	pr2	pr3	pr4
1464.	515101	2	.1485538	.2354556	.2915916	.3243991
1465.	515101	2	.372757	.3070787	.1966117	.1235526
1466.	515101	1	.372757	.3070787	.1966117	.1235526
1467.	515101	4	.2831409	.3021398	.2397316	.1749877
1468.	515101	3	.2079277	.2760683	.2740791	.2419248
1469.	515101	3	.2831409	.3021398	.2397316	.1749877
1470.	515102	1	.3251654	.3074122	.2193101	.1481123
1471.	515102	2	.4202843	.3011963	.1749344	.103585
1472.	515102	2	.4202843	.3011963	.1749344	.103585
1473.	515102	2	.4202843	.3011963	.1749344	.103585
1474.	515102	2	.3251654	.3074122	.2193101	.1481123
1475.	515102	1	.4202843	.3011963	.1749344	.103585
1476.	515102	2	.3251654	.3074122	.2193101	.1481123

For each observation, our best guess for the predicted outcome is the one with the highest predicted probability. For example, for the very first observation in the table above, we would choose outcome 4 as the most likely to occur.

We obtain predictions of the posterior means themselves at the school and class levels by typing

```
. predict re_s re_c, remeans
(calculating posterior means of random effects)
(using 7 quadrature points)
```

Here we list the predicted random effects for the first two classes for school 515:

```
. list class re_s re_c if school==515 & (class==515101 | class==515102),
> sepby(class)
```

	class	re_s	re_c
1464.	515101	-.0473739	.0633081
1465.	515101	-.0473739	.0633081
1466.	515101	-.0473739	.0633081
1467.	515101	-.0473739	.0633081
1468.	515101	-.0473739	.0633081
1469.	515101	-.0473739	.0633081
1470.	515102	-.0473739	-.1354929
1471.	515102	-.0473739	-.1354929
1472.	515102	-.0473739	-.1354929
1473.	515102	-.0473739	-.1354929
1474.	515102	-.0473739	-.1354929
1475.	515102	-.0473739	-.1354929
1476.	515102	-.0473739	-.1354929

We can see that the predicted random effects at the school level (re_s) are the same for all classes and that the predicted random effects at the class level (re_c) are constant within each class.

◁

Methods and formulas

Methods and formulas for predicting random effects and other statistics are given in *Methods and formulas* of [ME] **meglm postestimation**.

Also see

[ME] **meologit** — Multilevel mixed-effects ordered logistic regression

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

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

Title

meoprobit — Multilevel mixed-effects ordered probit regression

Syntax

`meoprobit` *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	
`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

options	Description
Model	
constraints(*constraints*)	apply specified linear constraints
collinear	keep collinear variables
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
nolrtest	do not perform likelihood-ratio test comparing with ordered probit regression
display_options	control 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
coeflegend	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
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 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 > Ordered probit regression

Description

meoprobit fits mixed-effects probit models for ordered responses. The actual values taken on by the response are irrelevant except that larger values are assumed to correspond to “higher” outcomes. The conditional distribution of the response given the random effects is assumed to be multinomial, with success probability determined by the standard normal cumulative distribution function.

Options

Model

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*$[k, l]$.

noconstant suppresses the constant (intercept) term; may be specified for any or all of the random-effects equations.

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

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (oim), that are robust to some kinds of misspecification (robust), and that allow for intragroup correlation (cluster *clustvar*); see [R] ***vce_option***. If vce(robust) is specified, robust variances are clustered at the highest level in the multilevel model.

Reporting

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

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

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

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

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

display_options: 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 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 meoprobit 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 meoprobit 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**.

meoprobit is a convenience command for meglm with a probit link and an ordinal family; see [ME] **meglm**.

Remarks are presented under the following headings:

Introduction

Mixed-effects ordered probit regression is ordered probit regression containing both fixed effects and random effects. An ordered response is a variable that is categorical and ordered, for instance, "poor", "good", and "excellent", which might indicate a person's current health status or the repair record of a car. In the absence of random effects, mixed-effects ordered probit regression reduces to ordered probit regression; see [R] **oprobit**.

Comprehensive treatments of mixed models are provided by, for example, Searle, Casella, and McCulloch (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). Agresti (2010, chap. 10) and Rabe-Hesketh and Skrondal (2012, chap. 11) are good introductory readings on applied multilevel modeling of ordinal data.

meoprobit allows for 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.

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 fixed effects $\mathbf{x}_{ij}$, a set of cutpoints $\boldsymbol{\kappa}$, and a set of random effects $\mathbf{u}_j$, the cumulative probability of the response being in a category higher than k is

$$\Pr(y_{ij} > k|\mathbf{x}_{ij}, \boldsymbol{\kappa}, \mathbf{u}_j) = \Phi(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j - \kappa_k) \tag{1}$$

for $j = 1, \ldots, M$ clusters, with cluster j consisting of $i = 1, \ldots, n_j$ observations. The cutpoints are labeled κ_1, κ_2, ..., κ_{K-1}, where K is the number of possible outcomes. $\Phi(\cdot)$ is the standard normal cumulative distribution function that represents cumulative probability.

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 probit regression model, with regression coefficients (fixed effects) $\boldsymbol{\beta}$. In our parameterization, $\mathbf{x}_{ij}$ does not contain a constant term because its effect is absorbed into the cutpoints. 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 $\boldsymbol{\Sigma}$.

From (1), we can derive the probability of observing outcome k as

$$\begin{aligned}
\Pr(y_{ij} = k|\boldsymbol{\kappa}, \mathbf{u}_j) &= \Pr(\kappa_{k-1} < \mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j + \epsilon_{ij} \leq \kappa_k) \\
&= \Pr(\kappa_{k-1} - \mathbf{x}_{ij}\boldsymbol{\beta} - \mathbf{z}_{ij}\mathbf{u}_j < \epsilon_{ij} \leq \kappa_k - \mathbf{x}_{ij}\boldsymbol{\beta} - \mathbf{z}_{ij}\mathbf{u}_j) \\
&= \Phi(\kappa_k - \mathbf{x}_{ij}\boldsymbol{\beta} - \mathbf{z}_{ij}\mathbf{u}_j) - \Phi(\kappa_{k-1} - \mathbf{x}_{ij}\boldsymbol{\beta} - \mathbf{z}_{ij}\mathbf{u}_j)
\end{aligned}$$

where κ_0 is taken as $-\infty$ and κ_K is taken as $+\infty$.

From the above, we may also write the model in terms of a latent linear response, where observed ordinal responses y_{ij} are generated from the latent continuous responses, such that

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

and

$$y_{ij} = \begin{cases} 1 & \text{if} \quad y^*_{ij} \leq \kappa_1 \\ 2 & \text{if} \quad \kappa_1 < y^*_{ij} \leq \kappa_2 \\ \vdots & \\ K & \text{if} \quad \kappa_{K-1} < y^*_{ij} \end{cases}$$

The errors ϵ_{ij} are distributed as standard normal with mean 0 and variance 1 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 the *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 itself is estimated, 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.

`meoprobit` supports three types of Gauss–Hermite quadrature and the Laplacian approximation method; see *Methods and formulas* of [ME] **meglm** for details.

Below we present two short examples of mixed-effects ordered probit regression; refer to [ME] **melogit** for additional examples including crossed random-effects models and to [ME] **me** and [ME] **meglm** for examples of other 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 ordered probit regression; see [R] **oprobit**.

▷ Example 1

We use the data from the Television, School, and Family Smoking Prevention and Cessation Project (Flay et al. 1988; Rabe-Hesketh and Skrondal 2012, chap. 11), where schools were randomly assigned into one of four groups defined by two treatment variables. Students within each school are nested in classes, and classes are nested in schools. In this example, we ignore the variability of classes within schools and fit a two-level model; we incorporate classes in a three-level model in example 2. The dependent variable is the tobacco and health knowledge (THK) scale score collapsed into four ordered categories. We regress the outcome on the treatment variables and their interaction and control for the pretreatment score.

```
. use http://www.stata-press.com/data/r13/tvsfpors
. meoprobit thk prethk cc##tv || school:
Fitting fixed-effects model:
Iteration 0:   log likelihood =  -2212.775
Iteration 1:   log likelihood = -2127.8111
Iteration 2:   log likelihood = -2127.7612
Iteration 3:   log likelihood = -2127.7612
Refining starting values:
Grid node 0:   log likelihood = -2149.7302
Fitting full model:
Iteration 0:   log likelihood = -2149.7302  (not concave)
Iteration 1:   log likelihood = -2129.6838  (not concave)
Iteration 2:   log likelihood = -2123.5143
Iteration 3:   log likelihood = -2122.2896
Iteration 4:   log likelihood = -2121.7949
Iteration 5:   log likelihood = -2121.7716
Iteration 6:   log likelihood = -2121.7715
Mixed-effects oprobit regression                Number of obs      =      1600
Group variable:           school                Number of groups   =        28

                                                Obs per group: min =        18
                                                               avg =      57.1
                                                               max =       137

Integration method: mvaghermite                 Integration points =         7

                                                Wald chi2(4)       =    128.05
Log likelihood = -2121.7715                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
         thk |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      prethk |   .2369804   .0227739    10.41   0.000     .1923444    .2816164
        1.cc |   .5490957   .1255108     4.37   0.000      .303099    .7950923
        1.tv |   .1695405   .1215889     1.39   0.163    -.0687693    .4078504
             |
       cc#tv |
        1 1  |  -.2951837   .1751969    -1.68   0.092    -.6385634    .0481959
-------------+----------------------------------------------------------------
       /cut1 |  -.0682011   .1003374    -0.68   0.497    -.2648587    .1284565
       /cut2 |     .67681   .1008836     6.71   0.000     .4790817    .8745382
       /cut3 |   1.390649   .1037494    13.40   0.000     1.187304    1.593995
-------------+----------------------------------------------------------------
school       |
   var(_cons)|   .0288527   .0146201                      .0106874    .0778937
------------------------------------------------------------------------------
LR test vs. oprobit regression:   chibar2(01) =    11.98 Prob>=chibar2 = 0.0003
```

Those of you familiar with the `mixed` command or other `me` commands will recognize the syntax and output. Below we comment on the items specific to ordered outcomes.

1. The estimation table reports the fixed effects, the estimated cutpoints $(\kappa_1, \kappa_2, \kappa_3)$, and the estimated variance components. The fixed effects can be interpreted just as you would the output from `oprobit`. We find that students with higher preintervention scores tend to have higher postintervention scores. Because of their interaction, the impact of the treatment variables on the knowledge score is not straightforward; we defer this discussion to example 1 of [ME] **meoprobit postestimation**.
2. Underneath the fixed effects and the cutpoints, the table shows the estimated variance components. The random-effects equation is labeled `school`, meaning that these are random effects at the `school` 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 0.03 with standard error 0.01. The reported likelihood-ratio test shows that there is enough variability between schools to favor a mixed-effects ordered probit regression over a standard ordered probit regression; see *Distribution theory for likelihood-ratio test* in [ME] **me** for a discussion of likelihood-ratio testing of variance components.

We now store our estimates for later use.

```
. estimates store r_2
```

◁

Three-level models

Two-level models extend naturally to models with three or more levels with nested random effects. Below we continue with example 1.

▷ Example 2

In this example, we fit a three-level model incorporating classes nested within schools as an additional level. The fixed-effects part remains the same.

```
. meoprobit thk prethk cc##tv || school: || class:
Fitting fixed-effects model:
Iteration 0:   log likelihood =  -2212.775
Iteration 1:   log likelihood = -2127.8111
Iteration 2:   log likelihood = -2127.7612
Iteration 3:   log likelihood = -2127.7612
Refining starting values:
Grid node 0:   log likelihood = -2195.6424
Fitting full model:
Iteration 0:   log likelihood = -2195.6424  (not concave)
Iteration 1:   log likelihood = -2167.9576  (not concave)
Iteration 2:   log likelihood = -2140.2644  (not concave)
Iteration 3:   log likelihood = -2128.6948  (not concave)
Iteration 4:   log likelihood = -2119.9225
Iteration 5:   log likelihood = -2117.0947
Iteration 6:   log likelihood = -2116.7004
Iteration 7:   log likelihood = -2116.6981
Iteration 8:   log likelihood = -2116.6981
Mixed-effects oprobit regression                Number of obs      =      1600

                     No. of       Observations per Group
 Group Variable      Groups    Minimum    Average    Maximum

         school          28         18       57.1        137
          class         135          1       11.9         28

Integration method: mvaghermite                 Integration points =         7

                                                Wald chi2(4)       =    124.20
Log likelihood = -2116.6981                     Prob > chi2        =    0.0000

         thk       Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]

      prethk     .238841   .0231446    10.32   0.000     .1934784    .2842036
        1.cc    .5254813   .1285816     4.09   0.000     .2734659    .7774967
        1.tv    .1455573   .1255827     1.16   0.246    -.1005803    .3916949

       cc#tv
        1 1   -.2426203   .1811999    -1.34   0.181    -.5977656    .1125251

       /cut1    -.074617   .1029791    -0.72   0.469    -.2764523    .1272184
       /cut2    .6863046   .1034813     6.63   0.000     .4834849    .8891242
       /cut3    1.413686   .1064889    13.28   0.000     1.204972    1.622401

school
   var(_cons)   .0186456   .0160226                      .0034604    .1004695

school>class
   var(_cons)   .0519974   .0224014                      .0223496    .1209745

LR test vs. oprobit regression:     chi2(2) =    22.13   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

Notes:

1. Our model now has two random-effects equations, separated by `||`. The first is a random intercept (constant only) at the `school` level (level three), and the second is a random intercept at the `class` level (level two). The order in which these are specified (from left to right) is significant—`meoprobit` assumes that `class` is nested within `school`.

2. The information on groups is now displayed as a table, with one row for each grouping. You can suppress this table with the nogroup or the noheader option, which will suppress the rest of the header as well.

3. The variance-component estimates are now organized and labeled according to level. The variance component for class is labeled school>class to emphasize that classes are nested within schools.

Compared with the two-level model from example 1, the estimate of the random intercept at the school level dropped from 0.03 to 0.02. This is not surprising because we now use two random components versus one random component to account for unobserved heterogeneity among students. We can use lrtest and our stored estimation result from example 1 to see which model provides a better fit:

```
. lrtest r_2 .

Likelihood-ratio test                                 LR chi2(1)  =     10.15
(Assumption: r_2 nested in .)                         Prob > chi2 =    0.0014

Note: The reported degrees of freedom assumes the null hypothesis is not on
      the boundary of the parameter space.  If this is not true, then the
      reported test is conservative.
```

The likelihood-ratio test favors the three-level model. For more information about the likelihood-ratio test in the context of mixed-effects models, see *Distribution theory for likelihood-ratio test* in [ME] **me**.

◁

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

meoprobit 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_cat)	number of categories
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)	meoprobit
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(covariates)	list of covariates
e(ivars)	grouping variables
e(model)	oprobit
e(title)	title in estimation output
e(link)	probit
e(family)	ordinal
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

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

Without a loss of generality, consider a two-level ordered probit model. The probability of observing outcome k for response y_{ij} is then

$$\begin{aligned} p_{ij} = \Pr(y_{ij} = k|\boldsymbol{\kappa}, \mathbf{u}_j) &= \Pr(\kappa_{k-1} < \boldsymbol{\eta}_{ij} + \epsilon_{it} \le \kappa_k) \\ &= \Phi(\kappa_k - \boldsymbol{\eta}_{ij}) - \Phi(\kappa_{k-1} - \boldsymbol{\eta}_{ij}) \end{aligned}$$

where $\boldsymbol{\eta}_{ij} = \mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j + \text{offset}_{ij}$, κ_0 is taken as $-\infty$, and κ_K is taken as $+\infty$. Here $\mathbf{x}_{ij}$ does not contain a constant term because its effect is absorbed into the cutpoints.

For cluster j, $j = 1, \dots, M$, the conditional distribution of $\mathbf{y}_j = (y_{j1}, \dots, y_{jn_j})'$ given a set of cluster-level random effects $\mathbf{u}_j$ is

$$
\begin{aligned}
f(\mathbf{y}_j|\mathbf{u}_j) &= \prod_{i=1}^{n_j} p_{ij}^{I_k(y_{ij})} \\
&= \exp \sum_{i=1}^{n_j} \left\{ I_k(y_{ij}) \log(p_{ij}) \right\}
\end{aligned}
$$

where

$$
I_k(y_{ij}) = \begin{cases} 1 & \text{if } y_{ij} = k \\ 0 & \text{otherwise} \end{cases}
$$

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 $f(\mathbf{y}_j, \mathbf{u}_j)$,

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

where

$$
h\left(\boldsymbol{\beta}, \boldsymbol{\kappa}, \boldsymbol{\Sigma}, \mathbf{u}_j\right) = \sum_{i=1}^{n_j} \left\{ I_k(y_{ij}) \log(p_{ij}) \right\} - \mathbf{u}_j' \boldsymbol{\Sigma}^{-1} \mathbf{u}_j/2
$$

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. `meoprobit` 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 $h\left(\boldsymbol{\beta}, \boldsymbol{\kappa}, \boldsymbol{\Sigma}, \mathbf{u}_j\right)$ 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, $\mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\kappa}, \boldsymbol{\Sigma}) = \sum_{j=1}^{M} \mathcal{L}_j(\boldsymbol{\beta}, \boldsymbol{\kappa}, \boldsymbol{\Sigma})$.

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

References

Agresti, A. 2010. *Analysis of Ordinal Categorical Data*. 2nd ed. Hoboken, NJ: Wiley.

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.

Flay, B. R., B. R. Brannon, C. A. Johnson, W. B. Hansen, A. L. Ulene, D. A. Whitney-Saltiel, L. R. Gleason, S. Sussman, M. D. Gavin, K. M. Glowacz, D. F. Sobol, and D. C. Spiegel. 1988. The television, school, and family smoking cessation and prevention project: I. Theoretical basis and program development. *Preventive Medicine* 17: 585–607.

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.

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.

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

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] **meoprobit postestimation** — Postestimation tools for meoprobit

[ME] **meologit** — Multilevel mixed-effects ordered logistic regression

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

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

[XT] **xtoprobit** — Random-effects ordered probit models

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

Title

meoprobit postestimation — Postestimation tools for meoprobit

Description

The following postestimation command is of special interest after `meoprobit`:

Command	Description
`estat group`	summarize the composition of the nested groups

The following standard postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat ic`	Akaike's and Schwarz's Bayesian information criteria (AIC and BIC)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

Special-interest postestimation commands

`estat group` reports the number of groups and minimum, average, and maximum group sizes for each level of the model. Model levels are identified by the corresponding group variable in the data. Because groups are treated as nested, the information in this summary may differ from what you would get if you used the `tabulate` command on each group variable individually.

Syntax for predict

Syntax for obtaining predictions of random effects and their standard errors

predict [*type*] *newvarsspec* [*if*] [*in*], { remeans | remodes } [reses(*newvarsspec*)]

Syntax for obtaining other predictions

predict [*type*] *newvarsspec* [*if*] [*in*] [, *statistic options*]

newvarsspec is *stub** or *newvarlist*.

statistic	Description
Main	
pr	predicted probabilities; the default
fitted	fitted linear predictor
xb	linear predictor for the fixed portion of the model only
stdp	standard error of the fixed-portion linear prediction

These statistics are available both in and out of sample; type predict ... if e(sample) ... if wanted only for the estimation sample.

options	Description
Main	
means	compute *statistic* using empirical Bayes means; the default
modes	compute *statistic* using empirical Bayes modes
nooffset	ignore the offset variable in calculating predictions; relevant only if you specified offset() when you fit the model
fixedonly	prediction for the fixed portion of the model only
outcome(*outcome*)	outcome category for predicted probabilities
Integration	
intpoints(#)	use # quadrature points to compute empirical Bayes means
iterate(#)	set maximum number of iterations in computing statistics involving empirical Bayes estimators
tolerance(#)	set convergence tolerance for computing statistics involving empirical Bayes estimators

You specify one or *k* new variables in *newvarlist* with pr, where *k* is the number of outcomes. If you do not specify outcome(), those options assume outcome(#1).

Menu for predict

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

remeans, remodes, reses(); see [ME] **meglm postestimation**.

pr, the default, calculates the predicted probabilities. By default, the probabilities are based on a linear predictor that includes both the fixed effects and the random effects, and the predicted probabilities are conditional on the values of the random effects. Use the fixedonly option if you want predictions that include only the fixed portion of the model, that is, if you want random effects set to 0.

You specify one or k new variables, where k is the number of categories of the dependent variable. If you specify the outcome() option, the probabilities will be predicted for the requested outcome only, in which case you specify only one new variable. If you specify one new variable and do not specify outcome(), outcome(#1) is assumed.

fitted, xb, stdp, means, modes, nooffset, fixedonly; see [ME] **meglm postestimation**.

By default or if the means option is specified, statistics pr, fitted, xb, and stdp are based on the posterior mean estimates of random effects. If the modes option is specified, these statistics are based on the posterior mode estimates of random effects.

outcome(*outcome*) specifies the outcome for which the predicted probabilities are to be calculated. outcome() should contain either one value of the dependent variable or one of #1, #2, ..., with #1 meaning the first category of the dependent variable, #2 meaning the second category, etc.

Integration

intpoints(), iterate(), tolerance(); see [ME] **meglm postestimation**.

Syntax for estat group

```
estat group
```

Menu for estat

Statistics > Postestimation > Reports and statistics

Remarks and examples

Various predictions, statistics, and diagnostic measures are available after fitting an ordered probit mixed-effects model using meoprobit. Here we show a short example of predicted probabilities and predicted random effects; refer to [ME] **meglm postestimation** for additional examples applicable to mixed-effects generalized linear models.

▷ Example 1

In example 2 of [ME] **meoprobit**, we modeled the tobacco and health knowledge (thk) score—coded 1, 2, 3, 4—among students as a function of two treatments (cc and tv) using a three-level ordered probit model with random effects at the school and class levels.

```
. use http://www.stata-press.com/data/r13/tvsfpors
. meoprobit thk prethk cc##tv || school: || class:
  (output omitted )
```

We obtain predicted probabilities for all four outcomes based on the contribution of both fixed effects and random effects by typing

```
. predict pr*
(predictions based on fixed effects and posterior means of random effects)
(option mu assumed)
(using 7 quadrature points)
```

As the note says, the predicted values are based on the posterior means of random effects. You can use the modes option to obtain predictions based on the posterior modes of random effects.

Because we specified a stub name, Stata saved the predicted random effects in variables pr1 through pr4. Here we list the predicted probabilities for the first two classes for school 515:

```
. list class thk pr? if school==515 & (class==515101 | class==515102),
> sepby(class)
```

	class	thk	pr1	pr2	pr3	pr4
1464.	515101	2	.1503512	.2416885	.2828209	.3251394
1465.	515101	2	.3750887	.2958534	.2080368	.121021
1466.	515101	1	.3750887	.2958534	.2080368	.121021
1467.	515101	4	.2886795	.2920168	.2433916	.1759121
1468.	515101	3	.2129906	.2729831	.2696254	.2444009
1469.	515101	3	.2886795	.2920168	.2433916	.1759121
1470.	515102	1	.3318574	.2959802	.2261095	.1460529
1471.	515102	2	.4223251	.2916287	.187929	.0981172
1472.	515102	2	.4223251	.2916287	.187929	.0981172
1473.	515102	2	.4223251	.2916287	.187929	.0981172
1474.	515102	2	.3318574	.2959802	.2261095	.1460529
1475.	515102	1	.4223251	.2916287	.187929	.0981172
1476.	515102	2	.3318574	.2959802	.2261095	.1460529

For each observation, our best guess for the predicted outcome is the one with the highest predicted probability. For example, for the very first observation in the table above, we would choose outcome 4 as the most likely to occur.

We obtain predictions of the posterior means themselves at the school and class levels by typing

```
. predict re_s re_c, remeans
(calculating posterior means of random effects)
(using 7 quadrature points)
```

Here we list the predicted random effects for the first two classes for school 515:

```
. list class re_s re_c if school==515 & (class==515101 | class==515102),
> sepby(class)

       +--------------------------------+
       |  class        re_s       re_c  |
       |--------------------------------|
 1464. | 515101   -.0340769    .0390243 |
 1465. | 515101   -.0340769    .0390243 |
 1466. | 515101   -.0340769    .0390243 |
 1467. | 515101   -.0340769    .0390243 |
 1468. | 515101   -.0340769    .0390243 |
 1469. | 515101   -.0340769    .0390243 |
       |--------------------------------|
 1470. | 515102   -.0340769   -.0834322 |
 1471. | 515102   -.0340769   -.0834322 |
 1472. | 515102   -.0340769   -.0834322 |
 1473. | 515102   -.0340769   -.0834322 |
 1474. | 515102   -.0340769   -.0834322 |
 1475. | 515102   -.0340769   -.0834322 |
 1476. | 515102   -.0340769   -.0834322 |
       +--------------------------------+
```

We can see that the predicted random effects at the school level (re_s) are the same for all classes and that the predicted random effects at the class level (re_c) are constant within each class.

◁

Methods and formulas

Methods and formulas for predicting random effects and other statistics are given in *Methods and formulas* of [ME] **meglm postestimation**.

Also see

[ME] **meoprobit** — Multilevel mixed-effects ordered probit regression

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

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

Title

mepoisson — Multilevel mixed-effects Poisson regression

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

Syntax

`mepoisson` *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 the constant term from the fixed-effects equation
`exposure(`*varname*$_e$`)`	include ln(*varname*$_e$) in model with coefficient constrained to 1
`offset(`*varname*$_o$`)`	include *varname*$_o$ 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

options	Description
Model	
constraints(*constraints*)	apply specified linear constraints
collinear	keep collinear variables
SE/Robust	
vce(*vcetype*)	*vcetype* may be oim, robust, or cluster *clustvar*
Reporting	
level(#)	set confidence level; default is level(95)
irr	report fixed-effects coefficients as incidence-rate ratios
nocnsreport	do not display constraints
notable	suppress coefficient table
noheader	suppress output header
nogroup	suppress table summarizing groups
nolrtest	do not perform likelihood-ratio test comparing with Poisson regression
display_options	control 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
coeflegend	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
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 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 > Poisson regression

Description

mepoisson fits mixed-effects models for count responses. The conditional distribution of the response given the random effects is assumed to be Poisson.

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

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.

exposure(*varname*$_e$) specifies a variable that reflects the amount of exposure over which the *depvar* events were observed for each observation; ln(*varname*$_e$) is included in the fixed-effects portion of the model with the coefficient constrained to be 1.

offset(*varname*$_o$) specifies that *varname*$_o$ 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*$[k,l]$.

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**.

irr reports estimated fixed-effects coefficients transformed to incidence-rate 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 stored. irr 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 mepoisson from performing a likelihood-ratio test that compares the mixed-effects Poisson model with standard (marginal) Poisson regression. This option may also be specified upon replay to suppress this test from the output.

display_options: 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 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 mepoisson 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 mepoisson 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**.

Remarks are presented under the following headings:

Introduction

Mixed-effects Poisson regression is Poisson 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 McCulloch (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). Rabe-Hesketh and Skrondal (2012, chap. 13) is a good introductory read on applied multilevel modeling of count data.

`mepoisson` allows for not just one, but many levels of nested clusters. 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 $\mathbf{u}_j$,

$$\Pr(y_{ij} = y|\mathbf{x}_{ij}, \mathbf{u}_j) = \exp\left(-\mu_{ij}\right) \mu_{ij}^{y}/y! \tag{1}$$

for $\mu_{ij} = \exp(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j)$, $j = 1, \dots, M$ clusters, with cluster j consisting of $i = 1, \dots, n_j$ observations. The responses are counts y_{ij}. 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 Poisson 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 $\boldsymbol{\Sigma}$.

As noted in chapter 13.7 of Rabe-Hesketh and Skrondal (2012), the inclusion of a random intercept causes the marginal variance of y_{ij} to be greater than the marginal mean, provided the variance of the random intercept is not 0. Thus the random intercept in a mixed-effects Poisson model produces overdispersion, a measure of variability above and beyond that allowed by a Poisson process; see [R] **nbreg** and [ME] **menbreg**.

Model (1) is a member of the class of generalized linear mixed models (GLMMs), which generalize 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 the *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 itself is estimated, 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.

`mepoisson` supports three types of Gauss–Hermite quadrature and the Laplacian approximation method; see *Methods and formulas* of [ME] **meglm** for details.

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

A two-level model

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

▷ Example 1

Breslow and Clayton (1993) fit a mixed-effects Poisson model to data from a randomized trial of the drug progabide for the treatment of epilepsy.

```
. use http://www.stata-press.com/data/r13/epilepsy
(Epilepsy data; progabide drug treatment)

. describe

Contains data from http://www.stata-press.com/data/r13/epilepsy.dta
  obs:           236                          Epilepsy data; progabide drug
                                                treatment
 vars:             8                          31 May 2013 14:09
 size:         4,956                          (_dta has notes)
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
subject         byte    %9.0g                 Subject ID: 1-59
seizures        int     %9.0g                 No. of seizures
treat           byte    %9.0g                 1: progabide; 0: placebo
visit           float   %9.0g                 Dr. visit; coded as (-.3, -.1,
                                                .1, .3)
lage            float   %9.0g                 log(age), mean-centered
lbas            float   %9.0g                 log(0.25*baseline seizures),
                                                mean-centered
lbas_trt        float   %9.0g                 lbas/treat interaction
v4              byte    %8.0g                 Fourth visit indicator
-------------------------------------------------------------------------------
Sorted by:  subject
```

Originally from Thall and Vail (1990), data were collected on 59 subjects (31 progabide, 28 placebo). The number of epileptic seizures (`seizures`) was recorded during the two weeks prior to each of four doctor visits (`visit`). The treatment group is identified by the indicator variable `treat`. Data were also collected on the logarithm of age (`lage`) and the logarithm of one-quarter the number of seizures during the eight weeks prior to the study (`lbas`). The variable `lbas_trt` represents the interaction between `lbas` and treatment. `lage`, `lbas`, and `lbas_trt` are mean centered. Because the study originally noted a substantial decrease in seizures prior to the fourth doctor visit, an indicator `v4` for the fourth visit was also recorded.

Breslow and Clayton (1993) fit a random-effects Poisson model for the number of observed seizures,

$$\log(\mu_{ij}) = \beta_0 + \beta_1 \texttt{treat}_{ij} + \beta_2 \texttt{lbas}_{ij} + \beta_3 \texttt{lbas\_trt}_{ij} + \beta_4 \texttt{lage}_{ij} + \beta_5 \texttt{v4}_{ij} + u_j$$

for $j = 1, \ldots, 59$ subjects and $i = 1, \ldots, 4$ visits. The random effects u_j are assumed to be normally distributed with mean 0 and variance σ_u^2.

```
. mepoisson seizures treat lbas lbas_trt lage v4 || subject:
Fitting fixed-effects model:
Iteration 0:   log likelihood = -1016.4106
Iteration 1:   log likelihood = -819.20112
Iteration 2:   log likelihood = -817.66006
Iteration 3:   log likelihood = -817.65925
Iteration 4:   log likelihood = -817.65925
Refining starting values:
Grid node 0:   log likelihood = -680.40523
Fitting full model:
Iteration 0:   log likelihood = -680.40523  (not concave)
Iteration 1:   log likelihood = -672.95766  (not concave)
Iteration 2:   log likelihood = -667.14039
Iteration 3:   log likelihood = -665.51823
Iteration 4:   log likelihood = -665.29165
Iteration 5:   log likelihood = -665.29067
Iteration 6:   log likelihood = -665.29067
Mixed-effects Poisson regression                Number of obs      =       236
Group variable:         subject                 Number of groups   =        59
                                                Obs per group: min =         4
                                                               avg =       4.0
                                                               max =         4
Integration method: mvaghermite                 Integration points =         7
                                                Wald chi2(5)       =    121.70
Log likelihood = -665.29067                     Prob > chi2        =    0.0000

    seizures        Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]

       treat    -.9330306   .4007512    -2.33   0.020    -1.718489   -.1475727
        lbas     .8844225   .1312033     6.74   0.000     .6272689    1.141576
    lbas_trt     .3382561   .2033021     1.66   0.096    -.0602087     .736721
        lage     .4842226   .3471905     1.39   0.163    -.1962582    1.164703
          v4    -.1610871   .0545758    -2.95   0.003    -.2680536   -.0541206
       _cons     2.154578   .2199928     9.79   0.000       1.7234    2.585756

subject
  var(_cons)     .2528664   .0589844                      .1600801     .399434

LR test vs. Poisson regression:  chibar2(01) =   304.74 Prob>=chibar2 = 0.0000
```

The number of seizures before the fourth visit does exhibit a significant drop, and the patients on progabide demonstrate a decrease in frequency of seizures compared with the placebo group. The subject-specific random effects also appear significant: $\widehat{\sigma}_u^2 = 0.25$ with standard error 0.06.

Because this is a simple random-intercept model, you can obtain equivalent results by using `xtpoisson` with the `re` and `normal` options.

◁

A three-level model

`mepoisson` can also fit higher-level models with multiple levels of nested random effects.

▷ Example 2

Rabe-Hesketh and Skrondal (2012, exercise 13.7) describe data from the *Atlas of Cancer Mortality in the European Economic Community* (EEC) (Smans, Mair, and Boyle 1993). The data were analyzed in Langford, Bentham, and McDonald (1998) and record the number of deaths among males due to malignant melanoma during 1971–1980.

```
. use http://www.stata-press.com/data/r13/melanoma
(Skin cancer (melanoma) data)

. describe

Contains data from http://localpress.stata.com/data/r13/melanoma.dta
  obs:           354                          Skin cancer (melanoma) data
 vars:             6                          30 May 2013 17:10
 size:         4,956                          (_dta has notes)
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
nation          byte    %11.0g     n          Nation ID
region          byte    %9.0g                 Region ID: EEC level-I areas
county          int     %9.0g                 County ID: EEC level-II/level-III
                                                areas
deaths          int     %9.0g                 No. deaths during 1971-1980
expected        float   %9.0g                 No. expected deaths
uv              float   %9.0g                 UV dose, mean-centered
-------------------------------------------------------------------------------
Sorted by:
```

Nine European nations (variable `nation`) are represented, and data were collected over geographical regions defined by EEC statistical services as level I areas (variable `region`), with deaths being recorded for each of 354 counties, which are level II or level III EEC-defined areas (variable `county`, which identifies the observations). Counties are nested within regions, and regions are nested within nations.

The variable `deaths` records the number of deaths for each county, and `expected` records the expected number of deaths (the exposure) on the basis of crude rates for the combined countries. Finally, the variable `uv` is a measure of exposure to ultraviolet (UV) radiation.

In modeling the number of deaths, one possibility is to include dummy variables for the nine nations as fixed effects. Another is to treat these as random effects and fit the three-level random-intercept Poisson model,

$$\log(\mu_{ijk}) = \log(\texttt{expected}_{ijk}) + \beta_0 + \beta_1\texttt{uv}_{ijk} + u_k + v_{jk}$$

for nation k, region j, and county i. The model includes an exposure term for expected deaths.

```
. mepoisson deaths c.uv##c.uv, exposure(expected) || nation: || region:
Fitting fixed-effects model:
Iteration 0:   log likelihood = -2136.0274
Iteration 1:   log likelihood =  -1723.127
Iteration 2:   log likelihood = -1722.9762
Iteration 3:   log likelihood = -1722.9762
Refining starting values:
Grid node 0:   log likelihood = -1166.9773
Fitting full model:
Iteration 0:   log likelihood = -1166.9773  (not concave)
Iteration 1:   log likelihood = -1152.6069  (not concave)
Iteration 2:   log likelihood =  -1151.902  (not concave)
Iteration 3:   log likelihood =  -1127.412  (not concave)
Iteration 4:   log likelihood = -1101.9248
Iteration 5:   log likelihood = -1094.1984
Iteration 6:   log likelihood =   -1088.05
Iteration 7:   log likelihood = -1086.9097
Iteration 8:   log likelihood = -1086.8995
Iteration 9:   log likelihood = -1086.8994
Mixed-effects Poisson regression                Number of obs     =        354

-----------------------------------------------------------
                |   No. of       Observations per Group
 Group Variable |   Groups    Minimum    Average    Maximum
----------------+------------------------------------------
         nation |        9          3       39.3         95
         region |       78          1        4.5         13
-----------------------------------------------------------

Integration method: mvaghermite                 Integration points =         7

                                                Wald chi2(2)      =      25.70
Log likelihood = -1086.8994                     Prob > chi2       =     0.0000
------------------------------------------------------------------------------
      deaths |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          uv |   .0057002   .0137919     0.41   0.679    -.0213315    .0327318
             |
   c.uv#c.uv |  -.0058377   .0013879    -4.21   0.000     -.008558   -.0031174
             |
       _cons |   .1289989   .1581224     0.82   0.415    -.1809154    .4389132
ln(expected) |          1  (exposure)
-------------+----------------------------------------------------------------
nation       |
   var(_cons)|   .1841878   .0945722                      .0673298    .5038655
-------------+----------------------------------------------------------------
nation>      |
region       |
   var(_cons)|   .0382645   .0087757                      .0244105    .0599811
------------------------------------------------------------------------------
LR test vs. Poisson regression:      chi2(2) =   1272.15   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

By including an exposure variable that is an expected rate, we are in effect specifying a linear model for the log of the standardized mortality ratio, the ratio of observed deaths to expected deaths that is based on a reference population, the reference population being all nine nations.

Looking at the estimated variance components, we can see that there is more unobserved variability between nations than between regions within each nation. This may be due to, for example, country-specific informational campaigns on the risks of sun exposure.

◁

Stored results

mepoisson 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)	mepoisson
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(covariates)	list of covariates
e(ivars)	grouping variables
e(model)	poisson
e(title)	title in estimation output
e(link)	log
e(family)	poisson
e(clustvar)	name of cluster variable
e(offset)	offset
e(exposure)	exposure variable
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

In a two-level Poisson model, 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

$$f(\mathbf{y}_j|\mathbf{u}_j) = \prod_{i=1}^{n_j} \left[\{\exp(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j)\}^{y_{ij}} \exp\{-\exp(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j)\}/y_{ij}!\right]$$
$$= \exp\left[\sum_{i=1}^{n_j} \{y_{ij}(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j) - \exp(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j) - \log(y_{ij}!)\}\right]$$

Defining $c(\mathbf{y}_j) = \sum_{i=1}^{n_j} \log(y_{ij}!)$, where $c(\mathbf{y}_j)$ does not depend on the model parameters, we can express the above compactly in matrix notation,

$$f(\mathbf{y}_j|\mathbf{u}_j) = \exp\left\{\mathbf{y}_j'(\mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j) - \mathbf{1}'\exp(\mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j) - c(\mathbf{y}_j)\right\}$$

where $\mathbf{X}_j$ is formed by stacking the row vectors $\mathbf{x}_{ij}$ and $\mathbf{Z}_j$ is formed by stacking the row vectors $\mathbf{z}_{ij}$. We extend the definition of $\exp(\cdot)$ to be a vector function 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 $f(\mathbf{y}_j, \mathbf{u}_j)$,

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

where

$$h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j) = \mathbf{y}_j'(\mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j) - \mathbf{1}'\exp(\mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j) - \mathbf{u}_j'\boldsymbol{\Sigma}^{-1}\mathbf{u}_j/2$$

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

The integration in (2) has no closed form and thus must be approximated. `mepoisson` 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 $g(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j)$ 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.

Breslow, N. E., and D. G. Clayton. 1993. Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association* 88: 9–25.

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

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.

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

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.

Langford, I. H., G. Bentham, and A. McDonald. 1998. Multi-level modelling of geographically aggregated health data: A case study on malignant melanoma mortality and UV exposure in the European community. *Statistics in Medicine* 17: 41–57.

Leyland, A. H., and H. Goldstein, ed. 2001. *Multilevel Modelling of Health Statistics*. New York: Wiley.

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.

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

Rabe-Hesketh, S., A. Skrondal, and A. Pickles. 2005. Maximum likelihood estimation of limited and discrete dependent variable models with nested random effects. *Journal of Econometrics* 128: 301–323.

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.

Skrondal, A., and S. Rabe-Hesketh. 2004. *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models*. Boca Raton, FL: Chapman & Hall/CRC.

Smans, M., C. S. Mair, and P. Boyle. 1993. *Atlas of Cancer Mortality in the European Economic Community*. Lyon, France: IARC Scientific Publications.

Thall, P. F., and S. C. Vail. 1990. Some covariance models for longitudinal count data with overdispersion. *Biometrics* 46: 657–671.

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

Also see

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

[ME] **menbreg** — Multilevel mixed-effects negative binomial regression

[ME] **meqrpoisson** — Multilevel mixed-effects Poisson regression (QR decomposition)

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

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

[XT] **xtpoisson** — Fixed-effects, random-effects, and population-averaged Poisson models

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

Title

mepoisson postestimation — Postestimation tools for mepoisson

Description

The following postestimation command is of special interest after `mepoisson`:

Command	Description
`estat group`	summarize the composition of the nested groups

The following standard postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat ic`	Akaike's and Schwarz's Bayesian information criteria (AIC and BIC)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

Special-interest postestimation commands

`estat group` reports the number of groups and minimum, average, and maximum group sizes for each level of the model. Model levels are identified by the corresponding group variable in the data. Because groups are treated as nested, the information in this summary may differ from what you would get if you used the `tabulate` command on each group variable individually.

Syntax for predict

Syntax for obtaining predictions of random effects and their standard errors

predict [*type*] *newvarsspec* [*if*] [*in*] , { remeans | remodes } [reses(*newvarsspec*)]

Syntax for obtaining other predictions

predict [*type*] *newvarsspec* [*if*] [*in*] [, *statistic* *options*]

newvarsspec is *stub** or *newvarlist*.

statistic	Description
Main	
mu	number of events; the default
fitted	fitted linear predictor
xb	linear predictor for the fixed portion of the model only
stdp	standard error of the fixed-portion linear prediction
pearson	Pearson residuals
deviance	deviance residuals
anscombe	Anscombe residuals

These statistics are available both in and out of sample; type predict ... if e(sample) ... if wanted only for the estimation sample.

options	Description
Main	
means	compute *statistic* using empirical Bayes means; the default
modes	compute *statistic* using empirical Bayes modes
nooffset	ignore the offset or exposure variable in calculating predictions; relevant only if you specified offset() or exposure() when you fit the model
fixedonly	prediction for the fixed portion of the model only
Integration	
intpoints(#)	use # quadrature points to compute empirical Bayes means
iterate(#)	set maximum number of iterations in computing statistics involving empirical Bayes estimators
tolerance(#)	set convergence tolerance for computing statistics involving empirical Bayes estimators

Menu for predict

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

remeans, remodes, reses(); see [ME] **meglm postestimation**.

mu, the default, calculates the predicted mean (the predicted number of events), that is, the inverse link function applied to the linear prediction. By default, this is based on a linear predictor that includes both the fixed effects and the random effects, and the predicted mean is conditional on the values of the random effects. Use the fixedonly option if you want predictions that include only the fixed portion of the model, that is, if you want random effects set to 0.

fitted, xb, stdp, pearson, deviance, anscombe, means, modes, nooffset, fixedonly; see [ME] **meglm postestimation**.

By default or if the means option is specified, statistics mu, pr, fitted, xb, stdp, pearson, deviance, and anscombe are based on the posterior mean estimates of random effects. If the modes option is specified, these statistics are based on the posterior mode estimates of random effects.

Integration

intpoints(), iterate(), tolerance(); see [ME] **meglm postestimation**.

Syntax for estat group

```
estat group
```

Menu for estat

Statistics > Postestimation > Reports and statistics

Remarks and examples

Various predictions, statistics, and diagnostic measures are available after fitting a mixed-effects Poisson model with mepoisson. For the most part, calculation centers around obtaining estimates of the subject/group-specific random effects. Random effects are not estimated when the model is fit but instead need to be predicted after estimation.

Here we show a short example of predicted counts and predicted random effects; refer to [ME] **meglm postestimation** for additional examples applicable to mixed-effects generalized linear models.

▷ Example 1

In example 2 of [ME] **mepoisson**, we modeled the number of deaths among males in nine European nations as a function of exposure to ultraviolet radiation (uv). We used a three-level Poisson model with random effects at the nation and region levels.

```
. use http://www.stata-press.com/data/r13/melanoma
(Skin cancer (melanoma) data)
. mepoisson deaths c.uv##c.uv, exposure(expected) || nation: || region:
  (output omitted)
```

We can use `predict` to obtain the predicted counts as well as the estimates of the random effects at the nation and region levels.

```
. predict mu
(predictions based on fixed effects and posterior means of random effects)
(option mu assumed)
(using 7 quadrature points)
. predict re_nat re_reg, remeans
(calculating posterior means of random effects)
(using 7 quadrature points)
```

Stata displays a note that the predicted values of `mu` are based on the posterior means of random effects. You can use option `modes` to obtain predictions based on the posterior modes of random effects.

Here we list the data for the first nation in the dataset, which happens to be Belgium:

```
. list nation region deaths mu re_nat re_reg if nation==1, sepby(region)
```

	nation	region	deaths	mu	re_nat	re_reg
1.	Belgium	1	79	69.17982	-.123059	.3604518
2.	Belgium	2	80	78.14297	-.123059	.049466
3.	Belgium	2	51	46.21698	-.123059	.049466
4.	Belgium	2	43	54.25965	-.123059	.049466
5.	Belgium	2	89	66.78156	-.123059	.049466
6.	Belgium	2	19	34.83411	-.123059	.049466
7.	Belgium	3	19	8.166062	-.123059	-.4354829
8.	Belgium	3	15	40.92741	-.123059	-.4354829
9.	Belgium	3	33	30.78324	-.123059	-.4354829
10.	Belgium	3	9	6.914059	-.123059	-.4354829
11.	Belgium	3	12	12.16361	-.123059	-.4354829

We can see that the predicted random effects at the nation level, `re_nat`, are the same for all the observations. Similarly, the predicted random effects at the region level, `re_reg`, are the same within each region. The predicted counts, `mu`, are closer to the observed deaths than the predicted counts from the negative binomial mixed-effects model in example 1 of [ME] **menbreg postestimation**.

◁

Methods and formulas

Methods and formulas for predicting random effects and other statistics are given in *Methods and formulas* of [ME] **meglm postestimation**.

Also see

[ME] **mepoisson** — Multilevel mixed-effects Poisson regression

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

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

Title

meprobit — Multilevel mixed-effects probit regression

Syntax

`meprobit` *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	
binomial(*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, 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
nolrtest	do not perform likelihood-ratio test comparing with probit regression
display_options	control 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
coeflegend	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
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 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 > Probit regression

Description

meprobit 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 success probability determined by the standard normal cumulative distribution function.

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(*#*), 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 meprobit from performing a likelihood-ratio test that compares the mixed-effects probit model with standard (marginal) probit regression. This option may also be specified upon replay to suppress this test from the output.

display_options: 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 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 meprobit 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 meprobit 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**.

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

Remarks are presented under the following headings:

Introduction

Mixed-effects probit regression is probit 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 McCulloch (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.

meprobit 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.

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 probit 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 probit regression, $H(\cdot)$ is the standard normal cumulative distribution function, which maps the linear predictor to the probability of a success ($y_{ij} = 1$) with $H(v) = \Phi(v)$.

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 a standard normal with mean 0 and variance 1 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 itself is estimated, 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.

meprobit 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 meprobit is the two-level model with a random intercept,

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

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

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

▷ Example 1

In example 1 of [ME] **melogit**, we 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. 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. Here we refit that model with `meprobit`:

```
. use http://www.stata-press.com/data/r13/bangladesh
(Bangladesh Fertility Survey, 1989)
. meprobit c_use urban age child* || district:
Fitting fixed-effects model:
Iteration 0:   log likelihood = -1228.8313
Iteration 1:   log likelihood = -1228.2466
Iteration 2:   log likelihood = -1228.2466
Refining starting values:
Grid node 0:   log likelihood = -1237.3973
Fitting full model:
Iteration 0:   log likelihood = -1237.3973  (not concave)
Iteration 1:   log likelihood = -1221.2111  (not concave)
Iteration 2:   log likelihood = -1207.4451
Iteration 3:   log likelihood = -1206.7002
Iteration 4:   log likelihood = -1206.5346
Iteration 5:   log likelihood = -1206.5336
Iteration 6:   log likelihood = -1206.5336
Mixed-effects probit 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)       =    115.36
Log likelihood = -1206.5336                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
       c_use |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       urban |   .4490191   .0727176     6.17   0.000     .3064953    .5915429
         age |  -.0162203   .0048005    -3.38   0.001    -.0256291   -.0068114
      child1 |    .674377   .0947829     7.11   0.000      .488606    .8601481
      child2 |   .8281581   .1048136     7.90   0.000     .6227272    1.033589
      child3 |   .8137876   .1073951     7.58   0.000     .6032972    1.024278
       _cons |   -1.02799   .0870307   -11.81   0.000    -1.198567   -.8574132
-------------+----------------------------------------------------------------
district     |
   var(_cons)|   .0798719    .026886                      .0412921    .1544972
```

```
LR test vs. probit regression:   chibar2(01) =    43.43 Prob>=chibar2 = 0.0000
```

Comparing the estimates of `meprobit` with those of `melogit`, we observe the familiar result where the probit estimates are closer to 0 in absolute value due to the smaller variance of the error term in the probit model. Example 1 of [ME] **meprobit postestimation** shows that the marginal effect of covariates is nearly the same between the two models.

Unlike a logistic regression, coefficients from a probit regression cannot be interpreted in terms of odds ratios. Most commonly, probit regression coefficients are interpreted in terms of partial effects, as we demonstrate in example 1 of [ME] **meprobit postestimation**. For now, we only note that urban women and women with more children are more likely to use contraceptives and that contraceptive use decreases with age. The estimated variance of the random intercept at the district level, $\widehat{\sigma}^2$, is 0.08 with standard error 0.03. The reported likelihood-ratio test shows that there is enough variability between districts to favor a mixed-effects probit regression over an ordinary probit regression; see *Distribution theory for likelihood-ratio test* in [ME] **me** for a discussion of likelihood-ratio testing of variance components.

◁

Three-level models

Two-level models extend naturally to models with three or more levels with nested random effects. Below we replicate example 2 of [ME] **melogit** with `meprobit`.

▷ Example 2

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 probit 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.

```
. use http://www.stata-press.com/data/r13/towerlondon
(Tower of London data)
. meprobit dtlm difficulty i.group || family: || subject:
Fitting fixed-effects model:
Iteration 0:   log likelihood = -317.11238
Iteration 1:   log likelihood = -314.50535
Iteration 2:   log likelihood = -314.50121
Iteration 3:   log likelihood = -314.50121
Refining starting values:
Grid node 0:   log likelihood = -326.18533
Fitting full model:
Iteration 0:   log likelihood = -326.18533  (not concave)
Iteration 1:   log likelihood = -313.16256  (not concave)
Iteration 2:   log likelihood = -308.47528
Iteration 3:   log likelihood = -305.02228
Iteration 4:   log likelihood = -304.88972
Iteration 5:   log likelihood = -304.88845
Iteration 6:   log likelihood = -304.88845
Mixed-effects probit regression                 Number of obs      =       677

-------------------------------------------------------------
                |   No. of       Observations per Group
 Group Variable |   Groups    Minimum    Average    Maximum
----------------+--------------------------------------------
         family |      118          2        5.7         27
        subject |      226          2        3.0          3
-------------------------------------------------------------

Integration method: mvaghermite                 Integration points =         7

                                                Wald chi2(3)       =     83.28
Log likelihood = -304.88845                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
        dtlm |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
  difficulty |  -.9329891   .1037376    -8.99   0.000    -1.136311   -.7296672
             |
       group |
          2  |  -.1632243    .204265    -0.80   0.424    -.5635763    .2371276
          3  |  -.6220196    .228063    -2.73   0.006    -1.069015   -.1750244
             |
       _cons |  -.8405154   .1597223    -5.26   0.000    -1.153565   -.5274654
-------------+----------------------------------------------------------------
family       |
   var(_cons)|   .2120948   .1736281                      .0426292    1.055244
-------------+----------------------------------------------------------------
family>      |
subject      |
   var(_cons)|   .3559141    .219331                       .106364    1.190956
------------------------------------------------------------------------------
LR test vs. probit regression:       chi2(2) =    19.23   Prob > chi2 = 0.0001
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—`meprobit` 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.

◁

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

meprobit 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)`	`meprobit`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(covariates)`	list of covariates
`e(ivars)`	grouping variables
`e(model)`	`probit`
`e(title)`	title in estimation output
`e(link)`	`probit`
`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 `meprobit` (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, $j = 1, \dots, M$, the conditional distribution of $\mathbf{y}_j = (y_{j1}, \dots, y_{jn_j})'$, given a set of cluster-level random effects $\mathbf{u}_j$, is

$$
\begin{aligned}
f(\mathbf{y}_j|\mathbf{u}_j) &= \prod_{i=1}^{n_j} \left[\binom{r_{ij}}{y_{ij}} \left\{\Phi(\boldsymbol{\eta}_{ij})\right\}^{y_{ij}} \left\{1-\Phi(\boldsymbol{\eta}_{ij})\right\}^{r_{ij}-y_{ij}} \right] \\
&= \exp\left(\sum_{i=1}^{n_j} \left[y_{ij} \log\left\{\Phi(\boldsymbol{\eta}_{ij})\right\} - (r_{ij}-y_{ij}) \log\left\{\Phi(-\boldsymbol{\eta}_{ij})\right\} + \log\binom{r_{ij}}{y_{ij}} \right] \right)
\end{aligned}
$$

for $\boldsymbol{\eta}_{ij} = \mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j + \text{offset}_{ij}$.

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

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

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

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

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

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

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. `meprobit` 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 $h\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right)$ 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

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.

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.

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

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.

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

Also see

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

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

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

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

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

[XT] **xtprobit** — Random-effects and population-averaged probit models

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

Title

meprobit postestimation — Postestimation tools for meprobit

Description

The following postestimation commands are of special interest after `meprobit`:

Command	Description
`estat group`	summarize the composition of the nested groups
`estat icc`	estimate intraclass correlations

The following standard postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat ic`	Akaike's and Schwarz's Bayesian information criteria (AIC and BIC)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

Special-interest postestimation commands

`estat group` reports the number of groups and minimum, average, and maximum group sizes for each level of the model. Model levels are identified by the corresponding group variable in the data. Because groups are treated as nested, the information in this summary may differ from what you would get if you used the `tabulate` command on each group variable individually.

`estat icc` displays the intraclass correlation for pairs of latent linear responses at each nested level of the model. Intraclass correlations are available for random-intercept models or for random-coefficient models conditional on random-effects covariates being equal to 0. They are not available for crossed-effects models.

Syntax for predict

Syntax for obtaining predictions of random effects and their standard errors

`predict` [*type*] *newvarsspec* [*if*] [*in*], { `remeans` | `remodes` } [`reses(`*newvarsspec*`)`]

Syntax for obtaining other predictions

`predict` [*type*] *newvarsspec* [*if*] [*in*] [, *statistic* *options*]

newvarsspec is *stub** or *newvarlist*.

statistic	Description
Main	
`mu`	predicted mean; the default
`fitted`	fitted linear predictor
`xb`	linear predictor for the fixed portion of the model only
`stdp`	standard error of the fixed-portion linear prediction
`pearson`	Pearson residuals
`deviance`	deviance residuals
`anscombe`	Anscombe residuals

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

options	Description
Main	
means	compute *statistic* using empirical Bayes means; the default
modes	compute *statistic* using empirical Bayes modes
nooffset	ignore the offset variable in calculating predictions; relevant only if you specified offset() when you fit the model
fixedonly	prediction for the fixed portion of the model only
Integration	
intpoints(#)	use # quadrature points to compute empirical Bayes means
iterate(#)	set maximum number of iterations in computing statistics involving empirical Bayes estimators
tolerance(#)	set convergence tolerance for computing statistics involving empirical Bayes estimators

Menu for predict

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

remeans, remodes, reses(); see [ME] **meglm postestimation**.

mu, the default, calculates the predicted mean (the probability of a positive outcome), that is, the inverse link function applied to the linear prediction. By default, this is based on a linear predictor that includes both the fixed effects and the random effects, and the predicted mean is conditional on the values of the random effects. Use the fixedonly option if you want predictions that include only the fixed portion of the model, that is, if you want random effects set to 0.

fitted, xb, stdp, pearson, deviance, anscombe, means, modes, nooffset, fixedonly; see [ME] **meglm postestimation**.

By default or if the means option is specified, statistics mu, fitted, xb, stdp, pearson, deviance, and anscombe are based on the posterior mean estimates of random effects. If the modes option is specified, these statistics are based on the posterior mode estimates of random effects.

Integration

intpoints(), iterate(), tolerance(); see [ME] **meglm postestimation**.

Syntax for estat

Summarize the composition of the nested groups

```
estat group
```

Estimate intraclass correlations

```
estat icc [, level(#)]
```

Menu for estat

Statistics > Postestimation > Reports and statistics

Option for estat icc

`level(#)` specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`; see **[U] 20.7 Specifying the width of confidence intervals**.

Remarks and examples

Various predictions, statistics, and diagnostic measures are available after fitting a mixed-effects probit model using `meprobit`. Here we show a short example of predicted probabilities and predicted random effects; refer to **[ME] meglm postestimation** for additional examples.

▷ Example 1

In example 2 of **[ME] meprobit**, we analyzed the cognitive ability (`dtlm`) of patients with schizophrenia compared with their relatives and control subjects, by using a three-level probit model with random effects at the family and subject levels. Cognitive ability was measured as the successful completion of the "Tower of London", a computerized task, measured at three levels of difficulty.

```
. use http://www.stata-press.com/data/r13/towerlondon
(Tower of London data)
. meprobit dtlm difficulty i.group || family: || subject:
  (output omitted)
```

We obtain predicted probabilities based on the contribution of both fixed effects and random effects by typing

```
. predict pr
(predictions based on fixed effects and posterior means of random effects)
(option mu assumed)
(using 7 quadrature points)
```

As the note says, the predicted values are based on the posterior means of random effects. You can use the `modes` option to obtain predictions based on the posterior modes of random effects.

We obtain predictions of the posterior means themselves by typing

```
. predict re*, remeans
(calculating posterior means of random effects)
(using 7 quadrature points)
```

Because we have one random effect at the family level and another random effect at the subject level, Stata saved the predicted posterior means in the variables `re1` and `re2`, respectively. If you are not sure which prediction corresponds to which level, you can use the `describe` command to show the variable labels.

Here we list the data for family 16:

```
. list family subject dtlm pr re1 re2 if family==16, sepby(subject)

     +------------------------------------------------------------+
     | family   subject   dtlm         pr        re1         re2 |
     |------------------------------------------------------------|
208. |     16         5      1   .5301687   .5051272    .1001124 |
209. |     16         5      0   .1956408   .5051272    .1001124 |
210. |     16         5      0   .0367041   .5051272    .1001124 |
     |------------------------------------------------------------|
211. |     16        34      1   .8876646   .5051272    .7798247 |
212. |     16        34      1   .6107262   .5051272    .7798247 |
213. |     16        34      1   .2572725   .5051272    .7798247 |
     |------------------------------------------------------------|
214. |     16        35      0   .6561904   .5051272   -.0322885 |
215. |     16        35      1   .2977437   .5051272   -.0322885 |
216. |     16        35      0    .071612   .5051272   -.0322885 |
     +------------------------------------------------------------+
```

The predicted random effects at the family level (`re1`) are the same for all members of the family. Similarly, the predicted random effects at the individual level (`re2`) are constant within each individual. The predicted probabilities (`pr`) for this family seem to be in fair agreement with the response (`dtlm`) based on a cutoff of 0.5.

We can use `estat icc` to estimate the residual intraclass correlation (conditional on the difficulty level and the individual's category) between the latent responses of subjects within the same family or between the latent responses of the same subject and family:

```
. estat icc

Residual intraclass correlation

------------------------------------------------------------------------------
                       Level |        ICC   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
                      family |   .1352637   .1050492      .0261998    .4762821
              subject|family |   .3622485   .0877459      .2124808    .5445812
------------------------------------------------------------------------------
```

`estat icc` reports two intraclass correlations for this three-level nested model. The first is the level-3 intraclass correlation at the family level, the correlation between latent measurements of the cognitive ability in the same family. The second is the level-2 intraclass correlation at the subject-within-family level, the correlation between the latent measurements of cognitive ability in the same subject and family.

There is not a strong correlation between individual realizations of the latent response, even within the same subject.

◁

Stored results

estat icc stores the following in r():

Scalars

r(icc#)	level-# intraclass correlation
r(se#)	standard errors of level-# intraclass correlation
r(level)	confidence level of confidence intervals

Macros

r(label#)	label for level #

Matrices

r(ci#)	vector of confidence intervals (lower and upper) for level-# intraclass correlation

For a G-level nested model, # can be any integer between 2 and G.

Methods and formulas

Methods and formulas are presented under the following headings:

Prediction
Intraclass correlations

Prediction

Methods and formulas for predicting random effects and other statistics are given in *Methods and formulas* of [ME] **meglm postestimation**.

Intraclass correlations

Consider a simple, two-level random-intercept model, stated in terms of a latent linear response, where only $y_{ij} = I(y^*_{ij} > 0)$ is observed for the latent variable,

$$y^*_{ij} = \beta + u^{(2)}_j + \epsilon^{(1)}_{ij}$$

with $i = 1, \ldots, n_j$ and level-2 groups $j = 1, \ldots, M$. Here β is an unknown fixed intercept, $u^{(2)}_j$ is a level-2 random intercept, and $\epsilon^{(1)}_{ij}$ is a level-1 error term. Errors are assumed to be distributed as standard normal with mean 0 and variance 1; random intercepts are assumed to be normally distributed with mean 0 and variance σ^2_2 and to be independent of error terms.

The intraclass correlation for this model is

$$\rho = \text{Corr}(y^*_{ij}, y^*_{i'j}) = \frac{\sigma^2_2}{1 + \sigma^2_2}$$

It corresponds to the correlation between the latent responses i and i' from the same group j.

Now consider a three-level nested random-intercept model,

$$y^*_{ijk} = \beta + u^{(2)}_{jk} + u^{(3)}_k + \epsilon^{(1)}_{ijk}$$

for measurements $i = 1, \ldots, n_{jk}$ and level-2 groups $j = 1, \ldots, M_{1k}$ nested within level-3 groups $k = 1, \ldots, M_2$. Here $u^{(2)}_{jk}$ is a level-2 random intercept, $u^{(3)}_k$ is a level-3 random intercept, and $\epsilon^{(1)}_{ijk}$ is a level-1 error term. The error terms have a standard normal distribution with mean 0 and variance 1. The random intercepts are assumed to be normally distributed with mean 0 and variances σ^2_2 and σ^2_3, respectively, and to be mutually independent. The error terms are also independent of the random intercepts.

We can consider two types of intraclass correlations for this model. We will refer to them as level-2 and level-3 intraclass correlations. The level-3 intraclass correlation is

$$\rho^{(3)} = \text{Corr}(y^*_{ijk}, y^*_{i'j'k}) = \frac{\sigma^2_3}{1 + \sigma^2_2 + \sigma^2_3}$$

This is the correlation between latent responses i and i' from the same level-3 group k and from different level-2 groups j and j'.

The level-2 intraclass correlation is

$$\rho^{(2)} = \text{Corr}(y^*_{ijk}, y^*_{i'jk}) = \frac{\sigma^2_2 + \sigma^2_3}{1 + \sigma^2_2 + \sigma^2_3}$$

This is the correlation between latent responses i and i' from the same level-3 group k and level-2 group j. (Note that level-1 intraclass correlation is undefined.)

More generally, for a G-level nested random-intercept model, the g-level intraclass correlation is defined as

$$\rho^{(g)} = \frac{\sum_{l=g}^{G} \sigma^2_l}{1 + \sum_{l=2}^{G} \sigma^2_l}$$

The above formulas also apply in the presence of fixed-effects covariates $\mathbf{X}$ in a random-effects model. In this case, intraclass correlations are conditional on fixed-effects covariates and are referred to as residual intraclass correlations. `estat icc` also uses the same formulas to compute intraclass correlations for random-coefficient models, assuming 0 baseline values for the random-effects covariates, and labels them as conditional intraclass correlations.

Intraclass correlations will always fall in [0,1] because variance components are nonnegative. To accommodate the range of an intraclass correlation, we use the logit transformation to obtain confidence intervals. We use the delta method to estimate the standard errors of the intraclass correlations.

Let $\widehat{\rho}^{(g)}$ be a point estimate of the intraclass correlation and $\widehat{\text{SE}}(\widehat{\rho}^{(g)})$ be its standard error. The $(1-\alpha) \times 100\%$ confidence interval for $\text{logit}(\rho^{(g)})$ is

$$\text{logit}(\widehat{\rho}^{(g)}) \pm z_{\alpha/2} \frac{\widehat{\text{SE}}(\widehat{\rho}^{(g)})}{\widehat{\rho}^{(g)}(1 - \widehat{\rho}^{(g)})}$$

where $z_{\alpha/2}$ is the $1-\alpha/2$ quantile of the standard normal distribution and $\text{logit}(x) = \ln\{x/(1-x)\}$. Let k_u be the upper endpoint of this interval, and let k_l be the lower. The $(1-\alpha) \times 100\%$ confidence interval for $\rho^{(g)}$ is then given by

$$\left(\frac{1}{1 + e^{-k_l}}, \frac{1}{1 + e^{-k_u}} \right)$$

Also see

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

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

[U] 20 Estimation and postestimation commands

Title

meqrlogit — Multilevel mixed-effects logistic regression (QR decomposition)

Syntax

`meqrlogit` *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* [, *re_options*]

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
`collinear`	keep collinear variables

options	Description
Model	
binomial(*varname* | *#*)	set binomial trials if data are in binomial form
Reporting	
level(*#*)	set confidence level; default is level(95)
or	report fixed-effects coefficients as odds ratios
variance	show random-effects parameter estimates as variances and covariances; the default
stddeviations	show random-effects parameter estimates as standard deviations and correlations
noretable	suppress random-effects table
nofetable	suppress fixed-effects table
estmetric	show parameter estimates in the estimation metric
noheader	suppress output header
nogroup	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	
intpoints(*#* [*#* ...])	set the number of integration (quadrature) points; default is intpoints(7)
laplace	use Laplacian approximation; equivalent to intpoints(1)
Maximization	
maximize_options	control the maximization process; seldom used
retolerance(*#*)	tolerance for random-effects estimates; default is retolerance(1e-8); seldom used
reiterate(*#*)	maximum number of iterations for random-effects estimation; default is reiterate(50); seldom used
matsqrt	parameterize variance components using matrix square roots; the default
matlog	parameterize variance components using matrix logarithms
refineopts(*maximize_options*)	control the maximization process during refinement of starting values
coeflegend	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

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

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

`bootstrap`, `by`, `jackknife`, `mi estimate`, `rolling`, and `statsby` are allowed; see [U] **11.1.10 Prefix commands**.

`coeflegend` does 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 > Estimation by QR decomposition > Logistic regression

Description

`meqrlogit`, like `melogit`, 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 success probability determined by the logistic cumulative distribution function.

`meqrlogit` provides an alternative estimation method, which uses the QR decomposition of the variance-components matrix. This method may aid convergence when variance components are near the boundary of the parameter space.

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.

`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`, and `unstructured`.

`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)`, except when the R. notation is used, in which case the default is `covariance(identity)` and only `covariance(identity)` and `covariance(exchangeable)` are allowed.

`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.

collinear specifies that meqrlogit not omit collinear variables from the random-effects equation. Usually, there is no reason to leave collinear variables in place; in fact, doing so usually causes the estimation to fail because of the matrix singularity caused by the collinearity. However, with certain models (for example, a random-effects model with a full set of contrasts), the variables may be collinear, yet the model is fully identified because of restrictions on the random-effects covariance structure. In such cases, using the collinear option allows the estimation to take place with the random-effects equation intact.

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.

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.

variance, the default, displays the random-effects parameter estimates as variances and covariances.

stddeviations displays the random-effects parameter estimates as standard deviations and correlations.

noretable suppresses the random-effects table.

nofetable suppresses the fixed-effects table.

estmetric displays all parameter estimates in the estimation metric. Fixed-effects estimates are unchanged from those normally displayed, but random-effects parameter estimates are displayed as log-standard deviations and hyperbolic arctangents of correlations, with equation names that organize them by model level.

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 meqrlogit 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, nofvlabel, fvwrap(*#*), fvwrapon(*style*), cformat(*%fmt*), pformat(*%fmt*), sformat(*%fmt*), and nolstretch; see [R] **estimation options**.

Integration

intpoints(*#* [*#* ...]) sets the number of integration points for adaptive Gaussian quadrature. The more integration points, the more accurate the approximation to the log likelihood. However, computation time increases with the number of quadrature points, and in models with many levels or many random coefficients, this increase can be substantial.

You may specify one number of integration points applying to all levels of random effects in the model, or you may specify distinct numbers of points for each level. `intpoints(7)` is the default; that is, by default seven quadrature points are used for each level.

`laplace` specifies that log likelihoods be calculated using the Laplacian approximation, equivalent to adaptive Gaussian quadrature with one integration point for each level in the model; `laplace` is equivalent to `intpoints(1)`. Computation time increases as a function of the number of quadrature points raised to a power equaling the dimension of the random-effects specification. The computational time saved by using `laplace` can thus be substantial, especially when you have many levels or random coefficients.

The Laplacian approximation has been known to produce biased parameter estimates, but the bias tends to be more prominent in the estimates of the variance components rather than in the estimates of the fixed effects. If your interest lies primarily with the fixed-effects estimates, the Laplace approximation may be a viable faster alternative to adaptive quadrature with multiple integration points.

When the `R.`*varname* notation is used, the dimension of the random effects increases by the number of distinct values of *varname*. Even when this number is small to moderate, it increases the total random-effects dimension to the point where estimation with more than one quadrature point is prohibitively intensive.

For this reason, when you use the `R.` notation in your random-effects equations, the `laplace` option is assumed. You can override this behavior by using the `intpoints()` option, but doing so is not recommended.

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 `meqrlogit` are listed below.

For the `technique()` option, the default is `technique(nr)`. The `bhhh` algorithm may not be specified.

`from(`*init_specs*`)` is particularly useful when combined with `refineopts(iterate(0))` (see the description below), which bypasses the initial optimization stage.

`retolerance(`*#*`)` specifies the convergence tolerance for the estimated random effects used by adaptive Gaussian quadrature. Although not estimated as model parameters, random-effects estimators are used to adapt the quadrature points. Estimating these random effects is an iterative procedure, with convergence declared when the maximum relative change in the random effects is less than `retolerance()`. The default is `retolerance(1e-8)`. You should seldom have to use this option.

`reiterate(`*#*`)` specifies the maximum number of iterations used when estimating the random effects to be used in adapting the Gaussian quadrature points; see the `retolerance()` option. The default is `reiterate(50)`. You should seldom have to use this option.

`matsqrt` (the default), during optimization, parameterizes variance components by using the matrix square roots of the variance–covariance matrices formed by these components at each model level.

`matlog`, during optimization, parameterizes variance components by using the matrix logarithms of the variance–covariance matrices formed by these components at each model level.

The `matsqrt` parameterization ensures that variance–covariance matrices are positive semidefinite, while `matlog` ensures matrices that are positive definite. For most problems, the matrix square root is more stable near the boundary of the parameter space. However, if convergence is problematic, one option may be to try the alternate `matlog` parameterization. When convergence is not an issue, both parameterizations yield equivalent results.

`refineopts(`*maximize_options*`)` controls the maximization process during the refinement of starting values. Estimation in `meqrlogit` takes place in two stages. In the first stage, starting values are refined by holding the quadrature points fixed between iterations. During the second stage, quadrature points are adapted with each evaluation of the log likelihood. Maximization options specified within `refineopts()` control the first stage of optimization; that is, they control the refining of starting values.

maximize_options specified outside `refineopts()` control the second stage.

The one exception to the above rule is the `nolog` option, which when specified outside `refine-opts()` applies globally.

`from(`*init_specs*`)` is not allowed within `refineopts()` and instead must be specified globally.

Refining starting values helps make the iterations of the second stage (those that lead toward the solution) more numerically stable. In this regard, of particular interest is `refineopts(iterate(`*#*`))`, with two iterations being the default. Should the maximization fail because of instability in the Hessian calculations, one possible solution may be to increase the number of iterations here.

The following option is available with `meqrlogit` but is not shown in the dialog box:

`coeflegend`; see [R] **estimation options**.

Remarks and examples

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.

`meqrlogit` 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.

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 $\mathbf{u}_j$,

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

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) $\boldsymbol{\beta}$.

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.

Multilevel models with binary responses have been used extensively in the health and social sciences. As just one example, Leyland and Goldstein (2001, sec. 3.6) describe a study of equity of healthcare in Great Britain. Multilevel models with binary and other limited dependent responses also have a long history in econometrics; Rabe-Hesketh, Skrondal, and Pickles (2005) provide an excellent survey.

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. The estimation method used by `meqrlogit` is a multicoefficient and multilevel extension of one of these quadrature types, namely, adaptive Gaussian quadrature (AGQ) based on conditional modes, with the multicoefficient extension from Pinheiro and Bates (1995) and the multilevel extension from Pinheiro and Chao (2006); see *Methods and formulas*.

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) analyze 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)
--------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     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.

Consider a standard logistic regression model, amended to have random effects for each district. Defining $\pi_{ij} = \Pr(\texttt{c\_use}_{ij} = 1)$, we have

$$\text{logit}(\pi_{ij}) = \beta_0 + \beta_1 \texttt{urban}_{ij} + \beta_2 \texttt{age}_{ij} + \beta_3 \texttt{child1}_{ij} + \beta_4 \texttt{child2}_{ij} + \beta_5 \texttt{child3}_{ij} + u_j \quad (2)$$

for $j = 1, \ldots, 60$ districts, with $i = 1, \ldots, n_j$ women in district j.

```
. meqrlogit c_use urban age child* || district:
Refining starting values:
Iteration 0:   log likelihood = -1219.2682
Iteration 1:   log likelihood = -1209.3544
Iteration 2:   log likelihood = -1207.1912
Performing gradient-based optimization:
Iteration 0:   log likelihood = -1207.1912
Iteration 1:   log likelihood = -1206.8323
Iteration 2:   log likelihood = -1206.8322
Iteration 3:   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 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    .7322764   .1194857     6.13   0.000     .4980887    .9664641
         age   -.0264982   .0078916    -3.36   0.001    -.0419654   -.0110309
      child1    1.116002   .1580921     7.06   0.000     .8061466    1.425856
      child2    1.365895    .174669     7.82   0.000      1.02355     1.70824
      child3    1.344031   .1796549     7.48   0.000     .9919141    1.696148
       _cons    -1.68929   .1477592   -11.43   0.000    -1.978892   -1.399687

  Random-effects Parameters     Estimate   Std. Err.     [95% Conf. Interval]

district: Identity
                   var(_cons)   .2156188   .0733234      .1107202    .4199007

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

Notes:

1. The estimation log consists of two parts:

   (a) A set of iterations aimed at refining starting values. These are designed to be relatively quick iterations aimed at getting the parameter estimates within a neighborhood of the eventual solution, making the iterations in (b) more numerically stable.

   (b) A set of gradient-based iterations. By default, these are Newton–Raphson iterations, but other methods are available by specifying the appropriate *maximize_options*; see [R] **maximize**.

2. The first estimation table reports the fixed effects, and these 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.

3. The second estimation table shows the estimated variance components. The first section of the table is labeled `district: Identity`, meaning that these are random effects at the `district` level and that their variance–covariance matrix is a multiple of the identity matrix; that is, $\mathbf{\Sigma} = \sigma_u^2\mathbf{I}$.

Because we have only one random effect at this level, `meqrlogit` knew that `Identity` is the only possible covariance structure. In any case, σ_u^2 was estimated as 0.22 with standard error 0.07.

If you prefer standard deviation estimates $\widehat{\sigma}_u$ to variance estimates $\widehat{\sigma}_u^2$, specify the `stddeviations` option either at estimation or on replay.

4. A likelihood-ratio test comparing the model to ordinary logistic regression, (2) without u_j, is provided and is highly significant for these data.
5. Finally, because (2) is a simple random-intercept model, you can also fit it with `xtlogit`, specifying the `re` option.

   We now store our estimates for later use.

```
. estimates store r_int
```

◁

In what follows, we will be extending (2), focusing on the variable `urban`. Before we begin, to keep things short we restate (2) as

$$\text{logit}(\pi_{ij}) = \beta_0 + \beta_1 \texttt{urban}_{ij} + \mathcal{F}_{ij} + u_j$$

where $\mathcal{F}_{ij}$ is merely shorthand for the portion of the fixed-effects specification having to do with age and children.

▷ Example 2

Extending (2) to allow for a random slope on the indicator variable `urban` yields the model

$$\text{logit}(\pi_{ij}) = \beta_0 + \beta_1 \texttt{urban}_{ij} + \mathcal{F}_{ij} + u_j + v_j \texttt{urban}_{ij} \tag{3}$$

which we can fit by typing

```
. meqrlogit c_use urban age child* || district: urban
  (output omitted)
. estimates store r_urban
```

Extending the model was as simple as adding `urban` to the random-effects specification so that the model now includes a random intercept *and* a random coefficient on `urban`. We dispense with the output because, although this is an improvement over the random-intercept model (2),

```
. lrtest r_int r_urban
Likelihood-ratio test                                 LR chi2(1)  =      3.66
(Assumption: r_int nested in r_urban)                 Prob > chi2 =    0.0558
Note: The reported degrees of freedom assumes the null hypothesis is not on
      the boundary of the parameter space.  If this is not true, then the
      reported test is conservative.
```

we find the default covariance structure for (u_j, v_j), `covariance(Independent)`,

$$\boldsymbol{\Sigma} = \text{Var}\begin{bmatrix} u_j \\ v_j \end{bmatrix} = \begin{bmatrix} \sigma_u^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix}$$

to be inadequate. We state that the random-coefficient model is an "improvement" over the random-intercept model because the null hypothesis of the likelihood-ratio comparison test ($H_0\colon \sigma_v^2 = 0$) is on the boundary of the parameter test. This makes the reported p-value, 5.6%, an upper bound on the actual p-value, which is actually half of that; see *Distribution theory for likelihood-ratio test* in [ME] **me**.

We see below that we can reject this model in favor of one that allows correlation between u_j and v_j.

```
. meqrlogit c_use urban age child* || district: urban, covariance(unstructured)
Refining starting values:
Iteration 0:   log likelihood = -1215.8594  (not concave)
Iteration 1:   log likelihood = -1204.0802
Iteration 2:   log likelihood =  -1199.798
Performing gradient-based optimization:
Iteration 0:   log likelihood =  -1199.798
Iteration 1:   log likelihood = -1199.4744
Iteration 2:   log likelihood = -1199.3158
Iteration 3:   log likelihood =  -1199.315
Iteration 4:   log likelihood =  -1199.315
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 points =   7                        Wald chi2(5)       =     97.50
Log likelihood =  -1199.315                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
       c_use |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       urban |   .8157872    .171552     4.76   0.000     .4795516    1.152023
         age |   -.026415    .008023    -3.29   0.001    -.0421398   -.0106902
      child1 |    1.13252   .1603285     7.06   0.000     .8182819    1.446758
      child2 |   1.357739   .1770522     7.67   0.000     1.010723    1.704755
      child3 |   1.353827   .1828801     7.40   0.000     .9953881    1.712265
       _cons |   -1.71165   .1605617   -10.66   0.000    -2.026345   -1.396954
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
district: Unstructured       |
                  var(urban) |   .6663222   .3224715      .2580709      1.7204
                  var(_cons) |   .3897434   .1292459      .2034723    .7465388
            cov(urban,_cons) |  -.4058846   .1755418     -.7499403   -.0618289
------------------------------------------------------------------------------
LR test vs. logistic regression:     chi2(3) =    58.42   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
. estimates store r_urban_corr
. lrtest r_urban r_urban_corr
Likelihood-ratio test                                 LR chi2(1)  =     11.38
(Assumption: r_urban nested in r_urban_corr)          Prob > chi2 =    0.0007
```

By specifying `covariance(unstructured)` above, we told `meqrlogit` to allow correlation between random effects at the `district` level; that is,

$$\boldsymbol{\Sigma} = \text{Var}\begin{bmatrix} u_j \\ v_j \end{bmatrix} = \begin{bmatrix} \sigma_u^2 & \sigma_{uv} \\ \sigma_{uv} & \sigma_v^2 \end{bmatrix}$$

◁

▷ Example 3

The purpose of introducing a random coefficient on the binary variable `urban` in (3) was to allow for separate random effects, within each district, for the urban and rural areas of that district. Hence, if we have the binary variable `rural` in our data such that $\texttt{rural}_{ij} = 1 - \texttt{urban}_{ij}$, then we can reformulate (3) as

$$\text{logit}(\pi_{ij}) = \beta_0\texttt{rural}_{ij} + (\beta_0+\beta_1)\texttt{urban}_{ij} + \mathcal{F}_{ij} + u_j\texttt{rural}_{ij} + (u_j+v_j)\texttt{urban}_{ij} \qquad \text{(3a)}$$

where we have translated both the fixed portion and the random portion to be in terms of `rural` rather than a random intercept. Translating the fixed portion is not necessary to make the point we make below, but we do so anyway for uniformity.

Translating the estimated random-effects parameters from the previous output to ones appropriate for (3a), we get $\text{Var}(u_j) = \widehat{\sigma}^2_u = 0.39$,

$$\begin{aligned}\text{Var}(u_j+v_j) &= \widehat{\sigma}^2_u + \widehat{\sigma}^2_v + 2\widehat{\sigma}_{uv} \\ &= 0.39 + 0.67 - 2(0.41) = 0.24\end{aligned}$$

and $\text{Cov}(u_j, u_j+v_j) = \widehat{\sigma}^2_u + \widehat{\sigma}_{uv} = 0.39 - 0.41 = -0.02$.

An alternative that does not require remembering how to calculate variances and covariances involving sums—and one that also gives you standard errors—is to let Stata do the work for you:

```
. generate byte rural = 1 - urban
. meqrlogit c_use rural urban age child*, noconstant || district: rural urban,
> noconstant cov(unstr)
  (output omitted)
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 points =   7                        Wald chi2(6)       =    120.24
Log likelihood =  -1199.315                     Prob > chi2        =    0.0000
```

c_use	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
rural	-1.71165	.1605618	-10.66	0.000	-2.026345	-1.396954
urban	-.8958623	.1704961	-5.25	0.000	-1.230028	-.5616961
age	-.026415	.008023	-3.29	0.001	-.0421398	-.0106902
child1	1.13252	.1603285	7.06	0.000	.818282	1.446758
child2	1.357739	.1770522	7.67	0.000	1.010724	1.704755
child3	1.353827	.1828801	7.40	0.000	.9953882	1.712265

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
district: Unstructured				
var(rural)	.3897439	.1292459	.2034726	.7465394
var(urban)	.2442965	.1450673	.0762886	.7823029
cov(rural,urban)	-.0161411	.1057469	-.2234011	.1911189

```
LR test vs. logistic regression:     chi2(3) =     58.42   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

The above output demonstrates an equivalent fit to that we displayed for model (3), with the added benefit of a more direct comparison of the parameters for rural and urban areas. ◁

❑ Technical note

We used the binary variables `rural` and `urban` instead of the factor notation `i.urban` because, although supported in the fixed-effects specification of the model, such notation is not supported in random-effects specifications. ❑

❑ Technical note

Our model fits for (3) and (3a) are equivalent only because we allowed for correlation in the random effects for both. Had we used the default `Independent` covariance structure, we would be fitting different models; in (3) we would be making the restriction that $\text{Cov}(u_j, v_j) = 0$, whereas in (3a) we would be assuming that $\text{Cov}(u_j, u_j + v_j) = 0$.

The moral here is that although `meqrlogit` will do this by default, one should be cautious when imposing an independent covariance structure, because the correlation between random effects is not invariant to model translations that would otherwise yield equivalent results in standard regression models. In our example, we remapped an intercept and binary coefficient to two complementary binary coefficients, something we could do in standard logistic regression without consequence but that here required more consideration.

Rabe-Hesketh and Skrondal (2012, sec. 11.4) provide a nice discussion of this phenomenon in the related case of recentering a continuous covariate. ❑

Other covariance structures

In the above examples, we demonstrated the `Independent` and `Unstructured` covariance structures. Also available are `Identity` (seen previously in output but not directly specified), which restricts random effects to be uncorrelated and share a common variance, and `Exchangeable`, which assumes a common variance and a common pairwise covariance.

You can also specify multiple random-effects equations at the same level, in which case the above four covariance types can be combined to form more complex blocked-diagonal covariance structures. This could be used, for example, to impose an equality constraint on a subset of variance components or to otherwise group together a set of related random effects.

Continuing the previous example, typing

```
. meqrlogit c_use urban age child* || district: child*, cov(exchangeable) ||
> district:
```

would fit a model with the same fixed effects as (3) but with random-effects structure

$$\text{logit}(\pi_{ij}) = \beta_0 + \cdots + u_{1j}\texttt{child1}_{ij} + u_{2j}\texttt{child2}_{ij} + u_{3j}\texttt{child3}_{ij} + v_j$$

That is, we have random coefficients on each indicator variable for children (the first `district:` specification) and an overall district random intercept (the second `district:` specification). The above syntax fits a model with overall covariance structure

$$\mathbf{\Sigma} = \text{Var}\begin{bmatrix} u_{1j} \\ u_{2j} \\ u_{3j} \\ v_j \end{bmatrix} = \begin{bmatrix} \sigma_u^2 & \sigma_c & \sigma_c & 0 \\ \sigma_c & \sigma_u^2 & \sigma_c & 0 \\ \sigma_c & \sigma_c & \sigma_u^2 & 0 \\ 0 & 0 & 0 & \sigma_v^2 \end{bmatrix}$$

reflecting the relationship among the random coefficients for children. We did not have to specify noconstant on the first district: specification. meqrlogit automatically avoids collinearity by including an intercept on only the final specification among repeated-level equations.

Of course, if we fit the above model, we would heed our own advice from the previous technical note and make sure that not only our data but also our specification characterization of the random effects permitted the above structure. That is, we would check the above against a model that had an Unstructured covariance for all four random effects and then perhaps against a model that assumed an Unstructured covariance among the three random coefficients on children, coupled with independence with the random intercept. All comparisons can be made by storing estimates (command estimates store) and then using lrtest, as demonstrated previously.

Three-level models

The methods we have discussed so far extend from two-level models to models with three or more levels with nested random effects.

▷ Example 4

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, clear
(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, and we request to see odds ratios.

```
. meqrlogit dtlm difficulty i.group || family: || subject: , or
Refining starting values:
Iteration 0:   log likelihood = -310.28433
Iteration 1:   log likelihood = -306.42785  (not concave)
Iteration 2:   log likelihood = -305.26009
Performing gradient-based optimization:
Iteration 0:   log likelihood = -305.26009
Iteration 1:   log likelihood = -305.12089
Iteration 2:   log likelihood = -305.12043
Iteration 3:   log likelihood = -305.12043
Mixed-effects logistic regression               Number of obs      =       677
```

Group Variable	No. of Groups	Observations per Group Minimum	Average	Maximum	Integration Points
family	118	2	5.7	27	7
subject	226	2	3.0	3	7

```
                                                Wald chi2(3)       =     74.89
Log likelihood = -305.12043                     Prob > chi2        =    0.0000
```

dtlm	Odds Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
difficulty	.192337	.0371622	-8.53	0.000	.131704	.2808839
group						
2	.7798295	.2763766	-0.70	0.483	.3893394	1.561964
3	.3491338	.1396499	-2.63	0.009	.1594117	.7646517
_cons	.2263075	.064463	-5.22	0.000	.1294902	.3955132

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
family: Identity				
var(_cons)	.569182	.5216584	.0944323	3.430694
subject: Identity				
var(_cons)	1.137931	.6857496	.3492673	3.70744

```
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—`meqrlogit` 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. Also the number of integration points for adaptive Gaussian quadrature is displayed within this table, because you can choose to have it vary by model level. As with two-level models, the default is seven points. You can suppress this table with the `nogroup` or the `noheader` option, which will suppress the rest of the header as well.
3. The variance-component estimates are now organized and labeled according to level.

After adjusting for the random-effects structure, the odds of successful completion of the Tower of London decrease 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.

◁

❑ Technical note

In the previous example, the subjects are coded with unique values between 1 and 251 (with some gaps), but such coding is not necessary to produce nesting within families. Once we specified the nesting structure to `meqrlogit`, all that was important was the relative coding of `subject` within each unique value of `family`. We could have coded `subjects` as the numbers 1, 2, 3, and so on, restarting at 1 with each new family, and `meqrlogit` would have produced the same results.

Group identifiers may also be coded using string variables.

❑

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).

Crossed-effects models

Not all mixed-effects models contain nested random effects.

▷ Example 5

Rabe-Hesketh and Skrondal (2012, 443–460) perform an analysis on school data from Fife, Scotland. The data, originally from Paterson (1991), are from a study measuring students' attainment as an integer score from 1 to 10, based on the Scottish school exit examination taken at age 16. The study comprises 3,435 students who first attended any one of 148 primary schools and then any one of 19 secondary schools.

```
. use http://www.stata-press.com/data/r13/fifeschool
(School data from Fife, Scotland)

. describe

Contains data from http://www.stata-press.com/data/r13/fifeschool.dta
  obs:         3,435                          School data from Fife, Scotland
 vars:             5                          28 May 2013 10:08
 size:        24,045                          (_dta has notes)
───────────────────────────────────────────────────────────────────────────────
              storage   display    value
variable name   type    format     label      variable label
───────────────────────────────────────────────────────────────────────────────
pid             int     %9.0g                 Primary school ID
sid             byte    %9.0g                 Secondary school ID
attain          byte    %9.0g                 Attainment score at age 16
vrq             int     %9.0g                 Verbal-reasoning score from final
                                                year of primary school
sex             byte    %9.0g                 1: female; 0: male
───────────────────────────────────────────────────────────────────────────────
Sorted by:

. generate byte attain_gt_6 = attain > 6
```

To make the analysis relevant to our present discussion, we focus not on the attainment score itself but instead on whether the score is greater than 6. We wish to model this indicator as a function of the fixed effect `sex` and of random effects due to primary and secondary schools.

For this analysis, it would make sense to assume that the random effects are not nested, but instead crossed, meaning that the effect due to primary school is the same regardless of the secondary school attended. Our model is thus

$$\text{logit}\{\Pr(\texttt{attain}_{ijk} > 6)\} = \beta_0 + \beta_1 \texttt{sex}_{ijk} + u_j + v_k \tag{4}$$

for student i, $i = 1, \ldots, n_{jk}$, who attended primary school j, $j = 1, \ldots, 148$, and then secondary school k, $k = 1, \ldots, 19$.

Because there is no evident nesting, one solution would be to consider the data as a whole and fit a two-level, one-cluster model with random-effects structure

$$\mathbf{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_{148} \\ v_1 \\ \vdots \\ v_{19} \end{bmatrix} \sim N(\mathbf{0}, \boldsymbol{\Sigma}); \quad \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_u^2 \mathbf{I}_{148} & \mathbf{0} \\ \mathbf{0} & \sigma_v^2 \mathbf{I}_{19} \end{bmatrix}$$

We can fit such a model by using the group designation `_all:`, which tells `meqrlogit` to treat the whole dataset as one cluster, and the `R.`*varname* notation, which mimics the creation of indicator variables identifying schools:

```
. meqrlogit attain_gt_6 sex || _all:R.pid || _all:R.sid, or
```

But we do not recommend fitting the model this way because of high total dimension ($148+19 = 167$) of the random effects. This would require working with matrices of column dimension 167, which is probably not a problem for most current hardware, but would be a problem if this number got much larger.

An equivalent way to fit (4) that has a smaller dimension is to treat the clusters identified by primary schools as nested within all the data, that is, as nested within the `_all` group.

```
. meqrlogit attain_gt_6 sex || _all:R.sid || pid:, or
Note: factor variables specified; option laplace assumed
  (output omitted )
Mixed-effects logistic regression               Number of obs      =      3435

-----------------------------------------------------------------------------
                |     No. of       Observations per Group       Integration
 Group Variable |     Groups    Minimum    Average    Maximum      Points
----------------+------------------------------------------------------------
           _all |          1       3435     3435.0       3435           1
            pid |        148          1       23.2         72           1
-----------------------------------------------------------------------------

                                                Wald chi2(1)       =     14.28
Log likelihood = -2220.0035                     Prob > chi2        =    0.0002

------------------------------------------------------------------------------
 attain_gt_6 | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         sex |    1.32512   .0986967     3.78   0.000     1.145135    1.533395
       _cons |   .5311497   .0622641    -5.40   0.000     .4221188    .6683427
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
_all: Identity               |
                  var(R.sid) |   .1239738   .0694742      .0413353     .371825
-----------------------------+------------------------------------------------
pid: Identity                |
                   var(_cons) |   .4520502   .0953867       .298934    .6835936
------------------------------------------------------------------------------
LR test vs. logistic regression:     chi2(2) =   195.80   Prob > chi2 = 0.0000

Note: LR test is conservative and provided only for reference.
Note: log-likelihood calculations are based on the Laplacian approximation.
```

Choosing the primary schools as those to nest was no accident; because there are far fewer secondary schools than primary schools, the above required only 19 random coefficients for the secondary schools and one random intercept at the primary school level, for a total dimension of 20. Our data also include a measurement of verbal reasoning, the variable `vrq`. Adding a fixed effect due to `vrq` in (4) would negate the effect due to secondary school, a fact we leave to you to verify as an exercise.

◁

See [ME] **mixed** for a similar discussion of crossed effects in the context of linear mixed models. Also see Rabe-Hesketh and Skrondal (2012) for more examples of crossed-effects models, including models with random interactions, and for more techniques on how to avoid high-dimensional estimation.

❑ Technical note

The estimation in the previous example was performed using a Laplacian approximation, even though we did not specify this. Whenever the `R.` notation is used in random-effects specifications, estimation reverts to the Laplacian method because of the high dimension induced by having the `R.` variables.

In the above example, through some creative nesting, we reduced the dimension of the random effects to 20, but this is still too large to permit estimation via adaptive Gaussian quadrature; see *Computation time and the Laplacian approximation* in [ME] **me**. Even with two quadrature points, our rough formula for computation time would contain within it a factor of $2^{20} = 1{,}048{,}576$.

The `laplace` option is therefore assumed when you use `R.` notation. If the number of distinct levels of your `R.` variables is small enough (say, five or fewer) to permit estimation via AGQ, you can override the imposition of `laplace` by specifying the `intpoints()` option.

❑

Stored results

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

Scalars

`e(N)`	number of observations
`e(k)`	number of parameters
`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(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(reparm_rc)`	return code, final reparameterization
`e(rc)`	return code
`e(converged)`	`1` if converged, `0` otherwise

Macros

`e(cmd)`	`meqrlogit`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(ivars)`	grouping variables
`e(model)`	`logistic`
`e(title)`	title in estimation output
`e(offset)`	offset
`e(binomial)`	binomial number of trials
`e(redim)`	random-effects dimensions
`e(vartypes)`	variance-structure types
`e(revars)`	random-effects covariates
`e(n_quad)`	number of integration points
`e(laplace)`	`laplace`, if Laplace approximation
`e(chi2type)`	`Wald`; type of model χ^2
`e(vce)`	`bootstrap` or `jackknife` if defined
`e(vcetype)`	title used to label Std. Err.
`e(method)`	`ML`
`e(opt)`	type of optimization
`e(ml_method)`	type of `ml` method
`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(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`

Matrices	
`e(b)`	coefficient vector
`e(Cns)`	constraints matrix
`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
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 `meqrlogit` (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, $j = 1, \dots, M$, the conditional distribution of $\mathbf{y}_j = (y_{j1}, \dots, y_{jn_j})'$, given a set of cluster-level random effects $\mathbf{u}_j$, is

$$\begin{aligned} f(\mathbf{y}_j|\mathbf{u}_j) &= \prod_{i=1}^{n_j} \left[\binom{r_{ij}}{y_{ij}} \{H(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j)\}^{y_{ij}} \{1 - H(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j)\}^{r_{ij}-y_{ij}} \right] \\ &= \exp\left(\sum_{i=1}^{n_j} \left[y_{ij}(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j) - r_{ij}\log\{1 + \exp(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j)\} + \log\binom{r_{ij}}{y_{ij}} \right] \right) \end{aligned}$$

for $H(v) = \exp(v)/\{1 + \exp(v)\}$.

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

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

where $\mathbf{X}_j$ is formed by stacking the row vectors $\mathbf{x}_{ij}$ and $\mathbf{Z}_j$ is formed by stacking the row vectors $\mathbf{z}_{ij}$. 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 $f(\mathbf{y}_j, \mathbf{u}_j)$,

$$\begin{aligned} \mathcal{L}_j(\boldsymbol{\beta}, \boldsymbol{\Sigma}) &= (2\pi)^{-q/2} |\boldsymbol{\Sigma}|^{-1/2} \int f(\mathbf{y}_j|\mathbf{u}_j) \exp\left(-\mathbf{u}_j'\boldsymbol{\Sigma}^{-1}\mathbf{u}_j/2\right) d\mathbf{u}_j \\ &= \exp\{c(\mathbf{y}_j, \mathbf{r}_j)\} (2\pi)^{-q/2} |\boldsymbol{\Sigma}|^{-1/2} \int \exp\{h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j)\} d\mathbf{u}_j \end{aligned} \tag{5}$$

where

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

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 (5) has no closed form and thus must be approximated. The Laplacian approximation (Tierney and Kadane 1986; Pinheiro and Bates 1995) 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. Taking first and second derivatives, we obtain

$$\begin{aligned} h'(\boldsymbol{\beta},\boldsymbol{\Sigma},\mathbf{u}_j) &= \frac{\partial h(\boldsymbol{\beta},\boldsymbol{\Sigma},\mathbf{u}_j)}{\partial \mathbf{u}_j} &&= \mathbf{Z}_j'\{\mathbf{y}_j - \mathbf{m}(\boldsymbol{\beta},\mathbf{u}_j)\} - \boldsymbol{\Sigma}^{-1}\mathbf{u}_j \\ h''(\boldsymbol{\beta},\boldsymbol{\Sigma},\mathbf{u}_j) &= \frac{\partial^2 h(\boldsymbol{\beta},\boldsymbol{\Sigma},\mathbf{u}_j)}{\partial \mathbf{u}_j\partial \mathbf{u}_j'} &&= -\left\{\mathbf{Z}_j'\mathbf{V}(\boldsymbol{\beta},\mathbf{u}_j)\mathbf{Z}_j + \boldsymbol{\Sigma}^{-1}\right\} \end{aligned}$$

where $\mathbf{m}(\boldsymbol{\beta},\mathbf{u}_j)$ is the vector function with the ith element equal to the conditional mean of y_{ij} given $\mathbf{u}_j$, that is, $r_{ij}H(\mathbf{x}_{ij}\boldsymbol{\beta}+\mathbf{z}_{ij}\mathbf{u}_j)$. $\mathbf{V}(\boldsymbol{\beta},\mathbf{u}_j)$ is the diagonal matrix whose diagonal entries v_{ij} are the conditional variances of y_{ij} given $\mathbf{u}_j$, namely,

$$v_{ij} = r_{ij}H(\mathbf{x}_{ij}\boldsymbol{\beta}+\mathbf{z}_{ij}\mathbf{u}_j)\{1 - H(\mathbf{x}_{ij}\boldsymbol{\beta}+\mathbf{z}_{ij}\mathbf{u}_j)\}$$

The maximizer of $h(\boldsymbol{\beta},\boldsymbol{\Sigma},\mathbf{u}_j)$ is $\widehat{\mathbf{u}}_j$ such that $h'(\boldsymbol{\beta},\boldsymbol{\Sigma},\widehat{\mathbf{u}}_j) = \mathbf{0}$. The integrand in (5) is proportional to the posterior density $f(\mathbf{u}_j|\mathbf{y}_j)$, so $\widehat{\mathbf{u}}_j$ also represents the posterior mode, a plausible estimator of $\mathbf{u}_j$ in its own right.

Given the above derivatives, the second-order Taylor approximation then takes the form

$$h(\boldsymbol{\beta},\boldsymbol{\Sigma},\mathbf{u}_j) \approx h(\boldsymbol{\beta},\boldsymbol{\Sigma},\widehat{\mathbf{u}}_j) + \frac{1}{2}(\mathbf{u}_j - \widehat{\mathbf{u}}_j)'h''(\boldsymbol{\beta},\boldsymbol{\Sigma},\widehat{\mathbf{u}}_j)(\mathbf{u}_j - \widehat{\mathbf{u}}_j) \tag{6}$$

The first-derivative term vanishes because $h'(\boldsymbol{\beta},\boldsymbol{\Sigma},\widehat{\mathbf{u}}_j) = \mathbf{0}$. Therefore,

$$\begin{aligned} \int \exp\{h(\boldsymbol{\beta},\boldsymbol{\Sigma},\mathbf{u}_j)\}\,d\mathbf{u}_j &\approx \exp\{h(\boldsymbol{\beta},\boldsymbol{\Sigma},\widehat{\mathbf{u}}_j)\} \\ &\quad\times \int \exp\left[-\frac{1}{2}(\mathbf{u}_j - \widehat{\mathbf{u}}_j)'\{-h''(\boldsymbol{\beta},\boldsymbol{\Sigma},\widehat{\mathbf{u}}_j)\}(\mathbf{u}_j - \widehat{\mathbf{u}}_j)\right]d\mathbf{u}_j \\ &= \exp\{h(\boldsymbol{\beta},\boldsymbol{\Sigma},\widehat{\mathbf{u}}_j)\}(2\pi)^{q/2}\left|-h''(\boldsymbol{\beta},\boldsymbol{\Sigma},\widehat{\mathbf{u}}_j)\right|^{-1/2} \end{aligned} \tag{7}$$

because the latter integrand can be recognized as the "kernel" of a multivariate normal density.

Combining the above with (5) (and taking logs) gives the Laplacian log-likelihood contribution of the jth cluster,

$$\mathcal{L}_j^{\text{Lap}}(\boldsymbol{\beta},\boldsymbol{\Sigma}) = -\frac{1}{2}\log|\boldsymbol{\Sigma}| - \log|\mathbf{R}_j| + h(\boldsymbol{\beta},\boldsymbol{\Sigma},\widehat{\mathbf{u}}_j) + c(\mathbf{y}_j,\mathbf{r}_j)$$

where $\mathbf{R}_j$ is an upper-triangular matrix such that $-h''(\boldsymbol{\beta},\boldsymbol{\Sigma},\widehat{\mathbf{u}}_j) = \mathbf{R}_j\mathbf{R}_j'$. Pinheiro and Chao (2006) show that $\widehat{\mathbf{u}}_j$ and $\mathbf{R}_j$ can be efficiently computed as the iterative solution to a least-squares problem by using matrix decomposition methods similar to those used in fitting LME models (Bates and Pinheiro 1998; Pinheiro and Bates 2000; [ME] **mixed**).

The fidelity of the Laplacian approximation is determined wholly by the accuracy of the approximation in (6). An alternative that does not depend so heavily on this approximation is integration via AGQ (Naylor and Smith 1982; Liu and Pierce 1994).

The application of AGQ to this particular problem is from Pinheiro and Bates (1995). When we reexamine the integral in question, a transformation of integration variables yields

$$\begin{aligned}\int \exp\left\{h\left(\boldsymbol{\beta},\boldsymbol{\Sigma},\mathbf{u}_j\right)\right\} d\mathbf{u}_j &= |\mathbf{R}_j|^{-1}\int \exp\left\{h\left(\boldsymbol{\beta},\boldsymbol{\Sigma},\widehat{\mathbf{u}}_j+\mathbf{R}_j^{-1}\mathbf{t}\right)\right\} d\mathbf{t} \\ &= (2\pi)^{q/2}\,|\mathbf{R}_j|^{-1}\int \exp\left\{h\left(\boldsymbol{\beta},\boldsymbol{\Sigma},\widehat{\mathbf{u}}_j+\mathbf{R}_j^{-1}\mathbf{t}\right)+\mathbf{t}'\mathbf{t}/2\right\}\phi(\mathbf{t})d\mathbf{t}\end{aligned} \tag{8}$$

where $\phi(\cdot)$ is the standard multivariate normal density. Because the integrand is now expressed as some function multiplied by a normal density, it can be estimated by applying the rules of standard Gauss–Hermite quadrature. For a predetermined number of quadrature points N_Q, define $a_k = \sqrt{2}a_k^*$ and $w_k = w_k^*/\sqrt{\pi}$, for $k = 1,\ldots,N_Q$, where (a_k^*, w_k^*) are a set of abscissas and weights for Gauss–Hermite quadrature approximations of $\int \exp(-x^2)f(x)dx$, as obtained from Abramowitz and Stegun (1972, 924).

Define $\mathbf{a_k} = (a_{k_1}, a_{k_2}, \ldots, a_{k_q})'$; that is, $\mathbf{a_k}$ is a vector that spans the N_Q abscissas over the dimension q of the random effects. Applying quadrature rules to (8) yields the AGQ approximation,

$$\begin{aligned}&\int \exp\left\{h\left(\boldsymbol{\beta},\boldsymbol{\Sigma},\mathbf{u}_j\right)\right\} d\mathbf{u}_j \\ &\approx (2\pi)^{q/2}\,|\mathbf{R}_j|^{-1}\sum_{k_1=1}^{N_Q}\cdots\sum_{k_q=1}^{N_Q}\left[\exp\left\{h\left(\boldsymbol{\beta},\boldsymbol{\Sigma},\widehat{\mathbf{u}}_j+\mathbf{R}_j^{-1}\mathbf{a_k}\right)+\mathbf{a}_\mathbf{k}'\mathbf{a_k}/2\right\}\prod_{p=1}^{q} w_{k_p}\right] \\ &\equiv (2\pi)^{q/2}\widehat{G}_j(\boldsymbol{\beta},\boldsymbol{\Sigma})\end{aligned}$$

resulting in the AGQ log-likelihood contribution of the ith cluster,

$$\mathcal{L}_j^{\rm AGQ}(\boldsymbol{\beta},\boldsymbol{\Sigma}) = -\frac{1}{2}\log|\boldsymbol{\Sigma}| + \log\left\{\widehat{G}_j(\boldsymbol{\beta},\boldsymbol{\Sigma})\right\} + c(\mathbf{y}_j,\mathbf{r}_j)$$

The "adaptive" part of adaptive Gaussian quadrature lies in the translation and rescaling of the integration variables in (8) by using $\widehat{\mathbf{u}}_j$ and $\mathbf{R}_j^{-1}$, respectively. This transformation of quadrature abscissas (centered at 0 in standard form) is chosen to better capture the features of the integrand, which through (7) can be seen to resemble a multivariate normal distribution with mean $\widehat{\mathbf{u}}_j$ and variance $\mathbf{R}_j^{-1}\mathbf{R}_j^{-T}$. AGQ is therefore not as dependent as the Laplace method upon the approximation in (6). In AGQ, (6) serves merely to redirect the quadrature abscissas, with the AGQ approximation improving as the number of quadrature points N_Q increases. In fact, Pinheiro and Bates (1995) point out that AGQ with only one quadrature point ($a = 0$ and $w = 1$) reduces to the Laplacian approximation.

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

Maximization of $\mathcal{L}(\boldsymbol{\beta},\boldsymbol{\Sigma})$ is performed with respect to $(\boldsymbol{\beta},\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a vector comprising the unique elements of the matrix square root of $\boldsymbol{\Sigma}$. This is done to ensure that $\boldsymbol{\Sigma}$ is always positive semidefinite. If the `matlog` option is specified, then $\boldsymbol{\theta}$ instead consists of the unique elements of the matrix logarithm of $\boldsymbol{\Sigma}$. For well-conditioned problems, both methods produce equivalent results, yet our experience deems the former as more numerically stable near the boundary of the parameter space.

Once maximization is achieved, parameter estimates are mapped from $(\widehat{\boldsymbol\beta}, \widehat{\boldsymbol\theta})$ to $(\widehat{\boldsymbol\beta}, \widehat{\gamma})$, where $\widehat{\gamma}$ is a vector containing the unique (estimated) elements of $\boldsymbol\Sigma$, expressed as logarithms of standard deviations for the diagonal elements and hyperbolic arctangents of the correlations for off-diagonal elements. This last step is necessary to (a) obtain a parameterization under which parameter estimates can be displayed and interpreted individually, rather than as elements of a matrix square root (or logarithm), and (b) parameterize these elements such that their ranges each encompass the entire real line.

Parameter estimates are stored in `e(b)` as $(\widehat{\boldsymbol\beta}, \widehat{\gamma})$, with the corresponding variance–covariance matrix stored in `e(V)`. Parameter estimates can be displayed in this metric by specifying the `estmetric` option. However, in `meqrlogit` output, variance components are most often displayed either as variances and covariances (the default) or as standard deviations and correlations (option `stddeviations`).

The approach outlined above can be extended from two-level models to higher-level models; see Pinheiro and Chao (2006) for details.

References

Abramowitz, M., and I. A. Stegun, ed. 1972. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. 10th ed. Washington, DC: National Bureau of Standards.

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.

Bates, D. M., and J. C. Pinheiro. 1998. Computational methods for multilevel modelling. In *Technical Memorandum BL0112140-980226-01TM*. Murray Hill, NJ: Bell Labs, Lucent Technologies. http://stat.bell-labs.com/NLME/CompMulti.pdf.

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.

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.

Leyland, A. H., and H. Goldstein, ed. 2001. *Multilevel Modelling of Health Statistics*. New York: Wiley.

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.

Liu, Q., and D. A. Pierce. 1994. A note on Gauss–Hermite quadrature. *Biometrika* 81: 624–629.

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.

Naylor, J. C., and A. F. M. Smith. 1982. Applications of a method for the efficient computation of posterior distributions. *Journal of the Royal Statistical Society, Series C* 31: 214–225.

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.

Palmer, T. M., C. M. Macdonald-Wallis, D. A. Lawlor, and K. Tilling. 2014. Estimating adjusted associations between random effects from multilevel models: The reffadjust package. *Stata Journal* 14: 119–140.

Paterson, L. 1991. Socio-economic status and educational attainment: A multidimensional and multilevel study. *Evaluation and Research in Education* 5: 97–121.

Pinheiro, J. C., and D. M. Bates. 1995. Approximations to the log-likelihood function in the nonlinear mixed-effects model. *Journal of Computational and Graphical Statistics* 4: 12–35.

——. 2000. *Mixed-Effects Models in S and S-PLUS*. New York: Springer.

Pinheiro, J. C., and E. C. Chao. 2006. Efficient Laplacian and adaptive Gaussian quadrature algorithms for multilevel generalized linear mixed models. *Journal of Computational and Graphical Statistics* 15: 58–81.

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

Rabe-Hesketh, S., A. Skrondal, and A. Pickles. 2005. Maximum likelihood estimation of limited and discrete dependent variable models with nested random effects. *Journal of Econometrics* 128: 301–323.

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.

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.

Tierney, L., and J. B. Kadane. 1986. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association* 81: 82–86.

Also see

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

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

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

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

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

[MI] **estimation** — Estimation commands for use with mi estimate

[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**

Title

meqrlogit postestimation — Postestimation tools for meqrlogit

Description

The following postestimation commands are of special interest after `meqrlogit`:

Command	Description
`estat group`	summarize the composition of the nested groups
`estat recovariance`	display the estimated random-effects covariance matrix (or matrices)
`estat icc`	estimate intraclass correlations

The following standard postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat ic`	Akaike's and Schwarz's Bayesian information criteria (AIC and BIC)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

Special-interest postestimation commands

`estat group` reports the number of groups and minimum, average, and maximum group sizes for each level of the model. Model levels are identified by the corresponding group variable in the data. Because groups are treated as nested, the information in this summary may differ from what you would get if you used the `tabulate` command on each group variable individually.

`estat recovariance` displays the estimated variance–covariance matrix of the random effects for each level in the model. Random effects can be either random intercepts, in which case the corresponding rows and columns of the matrix are labeled as `_cons`, or random coefficients, in which case the label is the name of the associated variable in the data.

`estat icc` displays the intraclass correlation for pairs of latent linear responses at each nested level of the model. Intraclass correlations are available for random-intercept models or for random-coefficient models conditional on random-effects covariates being equal to 0. They are not available for crossed-effects models.

Syntax for predict

Syntax for obtaining estimated random effects and their standard errors

`predict` [*type*] { *stub** | *newvarlist* } [*if*] [*in*], { `reffects` | `reses` }
  [`relevel(`*levelvar*`)`]

Syntax for obtaining other predictions

`predict` [*type*] *newvar* [*if*] [*in*] [, *statistic* `nooffset` `fixedonly`]

statistic	Description
Main	
`mu`	predicted mean; the default
`xb`	linear predictor for the fixed portion of the model only
`stdp`	standard error of the fixed-portion linear prediction
`pearson`	Pearson residuals
`deviance`	deviance residuals
`anscombe`	Anscombe residuals

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

Menu for predict

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

`reffects` calculates posterior modal estimates of the random effects. By default, estimates for all random effects in the model are calculated. However, if the `relevel(`*levelvar*`)` option is specified, then estimates for only level *levelvar* in the model are calculated. For example, if `classes` are nested within `schools`, then typing

```
. predict b*, reffects relevel(school)
```

would yield random-effects estimates at the school level. You must specify q new variables, where q is the number of random-effects terms in the model (or level). However, it is much easier to just specify *stub*`*` and let Stata name the variables *stub*`1`, *stub*`2`, ..., *stubq* for you.

`reses` calculates standard errors for the random-effects estimates obtained by using the `reffects` option. By default, standard errors for all random effects in the model are calculated. However, if the `relevel(`*levelvar*`)` option is specified, then standard errors for only level *levelvar* in the model are calculated. For example, if `classes` are nested within `schools`, then typing

```
. predict se*, reses relevel(school)
```

would yield standard errors at the school level. You must specify q new variables, where q is the number of random-effects terms in the model (or level). However, it is much easier to just specify *stub*`*` and let Stata name the variables *stub*`1`, *stub*`2`, ..., *stubq* for you.

The `reffects` and `reses` options often generate multiple new variables at once. When this occurs, the random effects (or standard errors) contained in the generated variables correspond to the order in which the variance components are listed in the output of `meqrlogit`. Still, examining the variable labels of the generated variables (with the `describe` command, for instance) can be useful in deciphering which variables correspond to which terms in the model.

`relevel(`*levelvar*`)` specifies the level in the model at which predictions for random effects and their standard errors are to be obtained. *levelvar* is the name of the model level and is either the name of the variable describing the grouping at that level or is `_all`, a special designation for a group comprising all the estimation data.

`mu`, the default, calculates the predicted mean. By default, this is based on a linear predictor that includes both the fixed effects and the random effects, and the predicted mean is conditional on the values of the random effects. Use the `fixedonly` option (see below) if you want predictions that include only the fixed portion of the model, that is, if you want random effects set to 0.

`xb` calculates the linear prediction $\mathbf{x}\boldsymbol{\beta}$ based on the estimated fixed effects (coefficients) in the model. This is equivalent to fixing all random effects in the model to their theoretical (prior) mean value of 0.

`stdp` calculates the standard error of the fixed-effects linear predictor $\mathbf{x}\boldsymbol{\beta}$.

`pearson` calculates Pearson residuals. Pearson residuals large in absolute value may indicate a lack of fit. By default, residuals include both the fixed portion and the random portion of the model. The `fixedonly` option modifies the calculation to include the fixed portion only.

`deviance` calculates deviance residuals. Deviance residuals are recommended by McCullagh and Nelder (1989) as having the best properties for examining the goodness of fit of a GLM. They are approximately normally distributed if the model is correctly specified. They may be plotted against the fitted values or against a covariate to inspect the model's fit. By default, residuals include both the fixed portion and the random portion of the model. The `fixedonly` option modifies the calculation to include the fixed portion only.

`anscombe` calculates Anscombe residuals, which are designed to closely follow a normal distribution. By default, residuals include both the fixed portion and the random portion of the model. The `fixedonly` option modifies the calculation to include the fixed portion only.

`nooffset` is relevant only if you specified `offset(`*varname*`)` with `meqrlogit`. It modifies the calculations made by `predict` so that they ignore the offset variable; the linear prediction is treated as $\mathbf{X}\beta + \mathbf{Z}\mathbf{u}$ rather than $\mathbf{X}\beta + \mathbf{Z}\mathbf{u} + \text{offset}$.

`fixedonly` modifies predictions to include only the fixed portion of the model, equivalent to setting all random effects equal to 0; see the `mu` option.

Syntax for estat

Summarize the composition of the nested groups

```
estat group
```

Display the estimated random-effects covariance matrix (or matrices)

`estat recovariance` [`, relevel(`*levelvar*`) correlation` *matlist_options*]

Estimate intraclass correlations

`estat icc` [`, level(`*#*`)`]

Menu for estat

Statistics > Postestimation > Reports and statistics

Options for estat recovariance

`relevel(`*levelvar*`)` specifies the level in the model for which the random-effects covariance matrix is to be displayed and returned in `r(cov)`. By default, the covariance matrices for all levels in the model are displayed. *levelvar* is the name of the model level and is either the name of the variable describing the grouping at that level or is `_all`, a special designation for a group comprising all the estimation data.

`correlation` displays the covariance matrix as a correlation matrix and returns the correlation matrix in `r(corr)`.

matlist_options are style and formatting options that control how the matrix (or matrices) is displayed; see [P] **matlist** for a list of options that are available.

Option for estat icc

`level(`*#*`)` specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`; see [U] **20.7 Specifying the width of confidence intervals**.

Remarks and examples

Various predictions, statistics, and diagnostic measures are available after fitting a logistic mixed-effects model with meqrlogit. For the most part, calculation centers around obtaining estimates of the subject/group-specific random effects. Random effects are not provided as estimates when the model is fit but instead need to be predicted after estimation. Calculation of intraclass correlations, estimating the dependence between latent linear responses for different levels of nesting, may also be of interest.

▷ Example 1

Following Rabe-Hesketh and Skrondal (2012, chap. 10), we consider a two-level mixed-effects model for onycholysis (separation of toenail plate from nail bed) among those who contract toenail fungus. The data are obtained from De Backer et al. (1998) and were also studied by Lesaffre and Spiessens (2001). The onycholysis outcome is dichotomously coded as 1 (moderate or severe onycholysis) or 0 (none or mild onycholysis). Fixed-effects covariates include treatment (0: itraconazole; 1: terbinafine), the month of measurement, and their interaction.

We fit the two-level model with meqrlogit:

```
. use http://www.stata-press.com/data/r13/toenail
(Onycholysis severity)
. meqrlogit outcome treatment month trt_month || patient:, intpoints(30)
Refining starting values:
Iteration 0:   log likelihood = -749.95893
Iteration 1:   log likelihood =  -630.0759
Iteration 2:   log likelihood =  -625.6965
Performing gradient-based optimization:
Iteration 0:   log likelihood =  -625.6965
Iteration 1:   log likelihood = -625.39741
Iteration 2:   log likelihood = -625.39709
Iteration 3:   log likelihood = -625.39709
Mixed-effects logistic regression               Number of obs      =      1908
Group variable: patient                         Number of groups   =       294
                                                Obs per group: min =         1
                                                               avg =       6.5
                                                               max =         7
Integration points =  30                        Wald chi2(3)       =    150.52
Log likelihood = -625.39709                     Prob > chi2        =    0.0000
```

outcome	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
treatment	-.1609377	.5842081	-0.28	0.783	-1.305965	.9840892
month	-.3910603	.0443958	-8.81	0.000	-.4780744	-.3040463
trt_month	-.1368073	.0680236	-2.01	0.044	-.270131	-.0034836
_cons	-1.618961	.4347772	-3.72	0.000	-2.471109	-.7668132

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
patient: Identity				
var(_cons)	16.06538	3.057362	11.06372	23.32819

```
LR test vs. logistic regression: chibar2(01) =   565.22 Prob>=chibar2 = 0.0000
```

It is of interest to determine the dependence among responses for the same subject (between-subject heterogeneity). Under the latent-linear-response formulation, this dependence can be obtained with the intraclass correlation. Formally, in a two-level random-effects model, the intraclass correlation corresponds to the correlation of latent responses within the same individual and also to the proportion of variance explained by the individual random effect.

In the presence of fixed-effects covariates, `estat icc` reports the residual intraclass correlation, which is the correlation between latent linear responses conditional on the fixed-effects covariates.

We use `estat icc` to estimate the residual intraclass correlation:

```
. estat icc
Residual intraclass correlation
```

Level	ICC	Std. Err.	[95% Conf.	Interval]
patient	.8300271	.026849	.7707982	.8764047

Conditional on treatment and month of treatment, we estimate that latent responses within the same patient have a large correlation of approximately 0.83. Further, 83% of the variance of a latent response is explained by the between-patient variability.

◁

▷ Example 2

In example 3 of [ME] **meqrlogit**, we represented the probability of contraceptive use among Bangladeshi women by using the model (stated with slightly different notation here)

$$\text{logit}(\pi_{ij}) = \beta_0\texttt{rural}_{ij} + \beta_1\texttt{urban}_{ij} + \beta_2\texttt{age}_{ij} + \\ \beta_3\texttt{child1}_{ij} + \beta_4\texttt{child2}_{ij} + \beta_5\texttt{child3}_{ij} + a_j\texttt{rural}_{ij} + b_j\texttt{urban}_{ij}$$

where π_{ij} is the probability of contraceptive use, $j = 1, \dots, 60$ districts, $i = 1, \dots, n_j$ women within each district, and a_j and b_j are normally distributed with mean 0 and variance–covariance matrix

$$\boldsymbol{\Sigma} = \text{Var}\begin{bmatrix} a_j \\ b_j \end{bmatrix} = \begin{bmatrix} \sigma_a^2 & \sigma_{ab} \\ \sigma_{ab} & \sigma_b^2 \end{bmatrix}$$

```
. use http://www.stata-press.com/data/r13/bangladesh
(Bangladesh Fertility Survey, 1989)
. generate byte rural = 1 - urban
. meqrlogit c_use rural urban age child*, noconstant || district: rural urban,
> noconstant cov(unstructured)
Refining starting values:
Iteration 0:   log likelihood = -1208.3924
Iteration 1:   log likelihood = -1204.1317
Iteration 2:   log likelihood = -1200.6022
Performing gradient-based optimization:
Iteration 0:   log likelihood = -1200.6022
Iteration 1:   log likelihood = -1199.3331
Iteration 2:   log likelihood =  -1199.315
Iteration 3:   log likelihood =  -1199.315
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 points =   7                        Wald chi2(6)       =    120.24
Log likelihood =  -1199.315                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
       c_use |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       rural |   -1.71165   .1605618   -10.66   0.000    -2.026345   -1.396954
       urban |  -.8958623   .1704961    -5.25   0.000    -1.230028   -.5616961
         age |   -.026415    .008023    -3.29   0.001    -.0421398   -.0106902
      child1 |    1.13252   .1603285     7.06   0.000      .818282    1.446758
      child2 |   1.357739   .1770522     7.67   0.000     1.010724    1.704755
      child3 |   1.353827   .1828801     7.40   0.000     .9953882    1.712265
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
district: Unstructured       |
                  var(rural) |   .3897439   .1292459      .2034726    .7465394
                  var(urban) |   .2442965   .1450673      .0762886    .7823029
            cov(rural,urban) |  -.0161411   .1057469     -.2234011    .1911189
------------------------------------------------------------------------------
LR test vs. logistic regression:     chi2(3) =    58.42   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

Rather than see the estimated variance components listed as variance and covariances as above, we can instead see them as correlations and standard deviations in matrix form; that is, we can see $\widehat{\boldsymbol{\Sigma}}$ as a correlation matrix:

```
. estat recovariance, correlation
Random-effects correlation matrix for level district
             |     rural      urban
-------------+----------------------
       rural |         1
       urban |    -.05231          1
```

The purpose of using this particular model was to allow for district random effects that were specific to the rural and urban areas of that district and that could be interpreted as such. We can obtain predictions of these random effects,

```
. predict re_rural re_urban, reffects
```

and their corresponding standard errors,

```
. predict se_rural se_urban, reses
```

The order in which we specified the variables to be generated corresponds to the order in which the variance components are listed in `meqrlogit` output. If in doubt, a simple `describe` will show how these newly generated variables are labeled just to be sure.

Having generated estimated random effects and standard errors, we can now list them for the first 10 districts:

```
. by district, sort: generate tolist = (_n==1)
. list district re_rural se_rural re_urban se_urban if district <= 10 & tolist,
> sep(0)
```

	district	re_rural	se_rural	re_urban	se_urban
1.	1	-.9206641	.3129662	-.5551252	.2321872
118.	2	-.0307772	.3784629	.0012746	.4938357
138.	3	-.0149147	.6242095	.2257356	.4689535
140.	4	-.2684802	.3951617	.5760574	.3970433
170.	5	.0787537	.3078451	.004534	.4675103
209.	6	-.3842217	.2741989	.2727722	.4184851
274.	7	-.1742786	.4008164	.0072177	.4938659
292.	8	.0447142	.315396	.2256405	.46799
329.	9	-.3561363	.3885605	.0733451	.4555067
352.	10	-.5368572	.4743089	.0222337	.4939776

◁

❑ Technical note

When these data were first introduced in [ME] **meqrlogit**, we noted that not all districts contained both urban and rural areas. This fact is somewhat demonstrated by the random effects that are nearly 0 in the above. A closer examination of the data would reveal that district 3 has no rural areas, and districts 2, 7, and 10 have no urban areas.

The estimated random effects are not exactly 0 in these cases because of the correlation between urban and rural effects. For instance, if a district has no urban areas, it can still yield a nonzero (albeit small) random-effects estimate for a nonexistent urban area because of the correlation with its rural counterpart.

Had we imposed an independent covariance structure in our model, the estimated random effects in the cases in question would be exactly 0.

❑

❑ Technical note

The estimated standard errors produced above with the `reses` option are conditional on the values of the estimated model parameters: $\boldsymbol{\beta}$ and the components of $\boldsymbol{\Sigma}$. Their interpretation is therefore not one of standard sample-to-sample variability but instead one that does not incorporate uncertainty in the estimated model parameters; see *Methods and formulas*.

That stated, conditional standard errors can still be used as a measure of relative precision, provided that you keep this caveat in mind.

❑

▷ Example 3

Continuing with example 2, we can obtain predicted probabilities, the default prediction:

```
. predict p
(option mu assumed; predicted means)
```

These predictions are based on a linear predictor that includes both the fixed effects and the random effects due to district. Specifying the `fixedonly` option gives predictions that set the random effects to their prior mean of 0. Below we compare both over the first 20 observations:

```
. predict p_fixed, fixedonly
(option mu assumed; predicted means)
. list c_use p p_fixed age child* in 1/20
```

	c_use	p	p_fixed	age	child1	child2	child3
1.	no	.3579543	.4927183	18.44	0	0	1
2.	no	.2134724	.3210403	-5.56	0	0	0
3.	no	.4672256	.6044016	1.44	0	1	0
4.	no	.4206505	.5584864	8.44	0	0	1
5.	no	.2510909	.3687281	-13.56	0	0	0
6.	no	.2412878	.3565185	-11.56	0	0	0
7.	no	.3579543	.4927183	18.44	0	0	1
8.	no	.4992191	.6345999	-3.56	0	0	1
9.	no	.4572049	.594723	-5.56	1	0	0
10.	no	.4662518	.6034657	1.44	0	0	1
11.	yes	.2412878	.3565185	-11.56	0	0	0
12.	no	.2004691	.3040173	-2.56	0	0	0
13.	no	.4506573	.5883407	-4.56	1	0	0
14.	no	.4400747	.5779263	5.44	0	0	1
15.	no	.4794195	.6160359	-0.56	0	0	1
16.	yes	.4465936	.5843561	4.44	0	0	1
17.	no	.2134724	.3210403	-5.56	0	0	0
18.	yes	.4794195	.6160359	-0.56	0	0	1
19.	yes	.4637674	.6010735	-6.56	1	0	0
20.	no	.5001973	.6355067	-3.56	0	1	0

◁

❑ Technical note

Out-of-sample predictions are permitted after `meqrlogit`, but if these predictions involve estimated random effects, the integrity of the estimation data must be preserved. If the estimation data have changed since the model was fit, `predict` will be unable to obtain predicted random effects that are appropriate for the fitted model and will give an error. Thus to obtain out-of-sample predictions that contain random-effects terms, be sure that the data for these predictions are in observations that augment the estimation data.

❑

▷ Example 4

Continuing with example 2, we can also compute intraclass correlations for the model.

In the presence of random-effects covariates, the intraclass correlation is no longer constant and depends on the values of the random-effects covariates. In this case, `estat icc` reports conditional intraclass correlations assuming 0 values for all random-effects covariates. For example, in a two-level model, this conditional correlation represents the correlation of the latent responses for two measurements on the same subject, both of which have random-effects covariates equal to 0. Similarly to the interpretation of intercept variances in random-coefficient models (Rabe-Hesketh and Skrondal 2012, chap. 16), interpretation of this conditional intraclass correlation relies on the usefulness of the 0 baseline values of random-effects covariates. For example, mean centering of the covariates is often used to make a 0 value a useful reference.

Estimation of the conditional intraclass correlation in the Bangladeshi contraceptive study setting of example 2 is of interest. In example 2, the random-effects covariates `rural` and `urban` for the random level `district` are mutually exclusive indicator variables and can never be simultaneously 0. Thus we could not use `estat icc` to estimate the conditional intraclass correlation for this model, because `estat icc` requires that the random intercept is included in all random-effects specifications.

Instead, we consider an alternative model for the Bangladeshi contraceptive study. In example 2 of [ME] **meqrlogit**, we represented the probability of contraceptive use among Bangladeshi women with fixed-effects for urban residence (`urban`), age (`age`), and the number of children (`child1–child3`). The random effects for urban and rural residence are represented as a random slope for urban residence and a random intercept at the `district` level.

We fit the model with `meqrlogit`:

```
. use http://www.stata-press.com/data/r13/bangladesh, clear
(Bangladesh Fertility Survey, 1989)
. meqrlogit c_use urban age child* || district: urban, covariance(unstructured)
Refining starting values:
Iteration 0:   log likelihood = -1215.8594  (not concave)
Iteration 1:   log likelihood = -1204.0802
Iteration 2:   log likelihood =  -1199.798
Performing gradient-based optimization:
Iteration 0:   log likelihood =  -1199.798
Iteration 1:   log likelihood = -1199.4744
Iteration 2:   log likelihood = -1199.3158
Iteration 3:   log likelihood =  -1199.315
Iteration 4:   log likelihood =  -1199.315
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 points =   7                        Wald chi2(5)       =     97.50
Log likelihood =  -1199.315                     Prob > chi2        =    0.0000
```

c_use	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
urban	.8157872	.171552	4.76	0.000	.4795516	1.152023
age	-.026415	.008023	-3.29	0.001	-.0421398	-.0106902
child1	1.13252	.1603285	7.06	0.000	.8182819	1.446758
child2	1.357739	.1770522	7.67	0.000	1.010723	1.704755
child3	1.353827	.1828801	7.40	0.000	.9953881	1.712265
_cons	-1.71165	.1605617	-10.66	0.000	-2.026345	-1.396954

```
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
district: Unstructured       |
                  var(urban) |   .6663222   .3224715      .2580709      1.7204
                  var(_cons) |   .3897434   .1292459      .2034723    .7465388
            cov(urban,_cons) |  -.4058846   .1755418     -.7499403   -.0618289
------------------------------------------------------------------------------
LR test vs. logistic regression:     chi2(3) =     58.42   Prob > chi2 = 0.0000

Note: LR test is conservative and provided only for reference.
```

We use `estat icc` to estimate the intraclass correlation conditional on `urban` being equal to 0:

```
. estat icc

Conditional intraclass correlation

------------------------------------------------------------------------------
                       Level |        ICC   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
                    district |   .1059197   .0314044      .0582458    .1849513
------------------------------------------------------------------------------
Note: ICC is conditional on zero values of random-effects covariates.
```

This estimate suggests that the latent responses are not strongly correlated for rural residents (`urban == 0`) within the same district, conditional on the fixed-effects covariates.

◁

▷ Example 5

In example 4 of [ME] **meqrlogit**, we fit a three-level model for the cognitive ability of schizophrenia patients as compared with their relatives and a control. Fixed-effects covariates include the difficulty of the test, `difficulty`, and an individual's category, `group` (control, family member of patient, or patient). Family units (`family`) represent the third nesting level, and individual subjects (`subject`) represent the second nesting level. Three measurements were taken on all but one subject, one for each difficulty measure.

We fit the model with `meqrlogit`:

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

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

Refining starting values:

Iteration 0:   log likelihood = -310.28433
Iteration 1:   log likelihood = -306.42785  (not concave)
Iteration 2:   log likelihood = -305.26009

Performing gradient-based optimization:

Iteration 0:   log likelihood = -305.26009
Iteration 1:   log likelihood = -305.12089
Iteration 2:   log likelihood = -305.12043
Iteration 3:   log likelihood = -305.12043

Mixed-effects logistic regression               Number of obs      =       677

--------------------------------------------------------------------------
                |   No. of       Observations per Group       Integration
 Group Variable |   Groups    Minimum    Average    Maximum      Points
----------------+---------------------------------------------------------
         family |      118          2        5.7         27           7
        subject |      226          2        3.0          3           7
--------------------------------------------------------------------------
```

```
                                                Wald chi2(3)      =     74.89
Log likelihood = -305.12043                     Prob > chi2       =    0.0000

------------------------------------------------------------------------------
        dtlm |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
  difficulty |  -1.648506   .1932139    -8.53   0.000    -2.027198   -1.269814
             |
       group |
          2  |   -.24868    .3544065    -0.70   0.483    -.9433039     .445944
          3  |   -1.0523    .3999896    -2.63   0.009    -1.836265   -.2683348
             |
       _cons |  -1.485861   .2848469    -5.22   0.000     -2.04415    -.927571
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
family: Identity             |
                   var(_cons) |    .569182   .5216584      .0944323    3.430694
-----------------------------+------------------------------------------------
subject: Identity            |
                   var(_cons) |   1.137931   .6857496      .3492673     3.70744
------------------------------------------------------------------------------
LR test vs. logistic regression:     chi2(2) =    17.54   Prob > chi2 = 0.0002
Note: LR test is conservative and provided only for reference.
```

We can use `estat icc` to estimate the residual intraclass correlation (conditional on the difficulty level and the individual's category) between the latent responses of subjects within the same family or between the latent responses of the same subject and family:

```
. estat icc
Residual intraclass correlation

------------------------------------------------------------------------------
                       Level |        ICC   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
                      family |   .1139052   .0997976      .0181741    .4716556
              subject|family |   .3416289   .0889531      .1929134    .5297405
------------------------------------------------------------------------------
```

`estat icc` reports two intraclass correlations for this three-level nested model. The first is the level-3 intraclass correlation at the family level, the correlation between latent measurements of the cognitive ability in the same family. The second is the level-2 intraclass correlation at the subject-within-family level, the correlation between the latent measurements of cognitive ability in the same subject and family.

There is not a strong correlation between individual realizations of the latent response, even within the same subject.

◁

Stored results

estat recovariance stores the following in r():

Scalars	
r(relevels)	number of levels
Matrices	
r(Cov#)	level-# random-effects covariance matrix
r(Corr#)	level-# random-effects correlation matrix (if option correlation was specified)

For a G-level nested model, # can be any integer between 2 and G.

estat icc stores the following in r():

Scalars	
r(icc#)	level-# intraclass correlation
r(se#)	standard errors of level-# intraclass correlation
r(level)	confidence level of confidence intervals
Macros	
r(label#)	label for level #
Matrices	
r(ci#)	vector of confidence intervals (lower and upper) for level-# intraclass correlation

For a G-level nested model, # can be any integer between 2 and G.

Methods and formulas

Methods and formulas are presented under the following headings:

Prediction
Intraclass correlations

Prediction

Continuing the discussion in *Methods and formulas* of [ME] **meqrlogit**, and using the definitions and formulas defined there, we begin by considering the prediction of the random effects $\mathbf{u}_j$ for the jth cluster in a two-level model.

Given a set of estimated meqrlogit parameters $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\Sigma}})$, a profile likelihood in $\mathbf{u}_j$ is derived from the joint distribution $f(\mathbf{y}_j, \mathbf{u}_j)$ as

$$\mathcal{L}_j(\mathbf{u}_j) = \exp\left\{c\left(\mathbf{y}_j, \mathbf{r}_j\right)\right\}(2\pi)^{-q/2}|\widehat{\boldsymbol{\Sigma}}|^{-1/2}\exp\left\{g\left(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\Sigma}}, \mathbf{u}_j\right)\right\} \tag{1}$$

The conditional maximum likelihood estimator of $\mathbf{u}_j$—conditional on fixed $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\Sigma}})$—is the maximizer of $\mathcal{L}_j(\mathbf{u}_j)$ or, equivalently, the value of $\widehat{\mathbf{u}}_j$ that solves

$$\mathbf{0} = g'\left(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\Sigma}}, \widehat{\mathbf{u}}_j\right) = \mathbf{Z}_j'\left\{\mathbf{y}_j - \mathbf{m}(\widehat{\boldsymbol{\beta}}, \widehat{\mathbf{u}}_j)\right\} - \widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\mathbf{u}}_j$$

Because (1) is proportional to the conditional density $f(\mathbf{u}_j|\mathbf{y}_j)$, you can also refer to $\widehat{\mathbf{u}}_j$ as the conditional mode (or posterior mode if you lean toward Bayesian terminology). Regardless, you are referring to the same estimator.

Conditional standard errors for the estimated random effects are derived from standard theory of maximum likelihood, which dictates that the asymptotic variance matrix of $\widehat{\mathbf{u}}_j$ is the negative inverse of the Hessian, which is estimated as

$$g''\left(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\Sigma}}, \widehat{\mathbf{u}}_j\right) = -\left\{\mathbf{Z}_j'\mathbf{V}(\widehat{\boldsymbol{\beta}}, \widehat{\mathbf{u}}_j)\mathbf{Z}_j + \widehat{\boldsymbol{\Sigma}}^{-1}\right\}$$

Similar calculations extend to models with more than one level of random effects; see Pinheiro and Chao (2006).

For any observation i in the jth cluster in a two-level model, define the linear predictor as

$$\widehat{\eta}_{ij} = \mathbf{x}_{ij}\widehat{\boldsymbol{\beta}} + \mathbf{z}_{ij}\widehat{\mathbf{u}}_j$$

In a three-level model, for the ith observation within the jth level-two cluster within the kth level-three cluster,

$$\widehat{\eta}_{ijk} = \mathbf{x}_{ijk}\widehat{\boldsymbol{\beta}} + \mathbf{z}^{(3)}_{ijk}\widehat{\mathbf{u}}^{(3)}_k + \mathbf{z}^{(2)}_{ijk}\widehat{\mathbf{u}}^{(2)}_{jk}$$

where $\mathbf{z}^{(p)}$ and $\mathbf{u}^{(p)}$ refer to the level p design variables and random effects, respectively. For models with more than three levels, the definition of $\widehat{\eta}$ extends in the natural way, with only the notation becoming more complicated.

If the `fixedonly` option is specified, $\widehat{\eta}$ contains the linear predictor for only the fixed portion of the model, for example, in a two-level model $\widehat{\eta}_{ij} = \mathbf{x}_{ij}\widehat{\boldsymbol{\beta}}$. In what follows, we assume a two-level model, with the only necessary modification for multilevel models being the indexing.

The predicted mean conditional on the random effects $\widehat{\mathbf{u}}_j$ is

$$\widehat{\mu}_{ij} = r_{ij}H(\widehat{\eta}_{ij})$$

Pearson residuals are calculated as

$$\nu^P_{ij} = \frac{y_{ij} - \widehat{\mu}_{ij}}{\{V(\widehat{\mu}_{ij})\}^{1/2}}$$

for $V(\widehat{\mu}_{ij}) = \widehat{\mu}_{ij}(1 - \widehat{\mu}_{ij}/r_{ij})$.

Deviance residuals are calculated as

$$\nu^D_{ij} = \text{sign}(y_{ij} - \widehat{\mu}_{ij})\sqrt{\widehat{d}^2_{ij}}$$

where

$$\widehat{d}^2_{ij} = \begin{cases} 2r_{ij}\log\left(\dfrac{r_{ij}}{r_{ij} - \widehat{\mu}_{ij}}\right) & \text{if } y_{ij} = 0 \\ 2y_{ij}\log\left(\dfrac{y_{ij}}{\widehat{\mu}_{ij}}\right) + 2(r_{ij} - y_{ij})\log\left(\dfrac{r_{ij} - y_{ij}}{r_{ij} - \widehat{\mu}_{ij}}\right) & \text{if } 0 < y_{ij} < r_{ij} \\ 2r_{ij}\log\left(\dfrac{r_{ij}}{\widehat{\mu}_{ij}}\right) & \text{if } y_{ij} = r_{ij} \end{cases}$$

Anscombe residuals are calculated as

$$\nu^A_{ij} = \frac{3\left\{y_{ij}^{2/3}\mathcal{H}(y_{ij}/r_{ij}) - \widehat{\mu}^{2/3}\mathcal{H}(\widehat{\mu}_{ij}/r_{ij})\right\}}{2\left(\widehat{\mu}_{ij} - \widehat{\mu}^2_{ij}/r_{ij}\right)^{1/6}}$$

where $\mathcal{H}(t)$ is a specific univariate case of the Hypergeometric2F1 function (Wolfram 1999, 771–772). For Anscombe residuals for binomial regression, the specific form of the Hypergeometric2F1 function that we require is $\mathcal{H}(t) = {}_2F_1(2/3, 1/3, 5/3, t)$.

For a discussion of the general properties of the above residuals, see Hardin and Hilbe (2012, chap. 4).

Intraclass correlations

Consider a simple, two-level random-intercept model, stated in terms of a latent linear response, where only $y_{ij} = I(y^*_{ij} > 0)$ is observed for the latent variable,

$$y^*_{ij} = \beta + u^{(2)}_j + \epsilon^{(1)}_{ij}$$

with $i = 1, \dots, n_j$ and level-2 groups $j = 1, \dots, M$. Here β is an unknown fixed intercept, $u^{(2)}_j$ is a level-2 random intercept, and $\epsilon^{(1)}_{ij}$ is a level-1 error term. Errors are assumed to be logistic with mean 0 and variance $\sigma^2_1 = \pi^2/3$; random intercepts are assumed to be normally distributed with mean 0 and variance σ^2_2 and to be independent of error terms.

The intraclass correlation for this model is

$$\rho = \mathrm{Corr}(y^*_{ij}, y^*_{i'j}) = \frac{\sigma^2_2}{\pi^2/3 + \sigma^2_2}$$

It corresponds to the correlation between the latent responses i and i' from the same group j.

Now consider a three-level nested random-intercept model,

$$y^*_{ijk} = \beta + u^{(2)}_{jk} + u^{(3)}_k + \epsilon^{(1)}_{ijk}$$

for measurements $i = 1, \dots, n_{jk}$ and level-2 groups $j = 1, \dots, M_{1k}$ nested within level-3 groups $k = 1, \dots, M_2$. Here $u^{(2)}_{jk}$ is a level-2 random intercept, $u^{(3)}_k$ is a level-3 random intercept, and $\epsilon^{(1)}_{ijk}$ is a level-1 error term. The error terms have a logistic distribution with mean 0 and variance $\sigma^2_1 = \pi^2/3$. The random intercepts are assumed to be normally distributed with mean 0 and variances σ^2_2 and σ^2_3, respectively, and to be mutually independent. The error terms are also independent of the random intercepts.

We can consider two types of intraclass correlations for this model. We will refer to them as level-2 and level-3 intraclass correlations. The level-3 intraclass correlation is

$$\rho^{(3)} = \mathrm{Corr}(y^*_{ijk}, y^*_{i'j'k}) = \frac{\sigma^2_3}{\pi^2/3 + \sigma^2_2 + \sigma^2_3}$$

This is the correlation between latent responses i and i' from the same level-3 group k and from different level-2 groups j and j'.

The level-2 intraclass correlation is

$$\rho^{(2)} = \mathrm{Corr}(y^*_{ijk}, y^*_{i'jk}) = \frac{\sigma^2_2 + \sigma^2_3}{\pi^2/3 + \sigma^2_2 + \sigma^2_3}$$

This is the correlation between latent responses i and i' from the same level-3 group k and level-2 group j. (Note that level-1 intraclass correlation is undefined.)

More generally, for a G-level nested random-intercept model, the g-level intraclass correlation is defined as

$$\rho^{(g)} = \frac{\sum_{l=g}^{G} \sigma^2_l}{\pi^2/3 + \sum_{l=2}^{G} \sigma^2_l}$$

The above formulas also apply in the presence of fixed-effects covariates $\mathbf{X}$ in a random-effects model. In this case, intraclass correlations are conditional on fixed-effects covariates and are referred to as residual intraclass correlations. `estat icc` also uses the same formulas to compute intraclass correlations for random-coefficient models, assuming 0 baseline values for the random-effects covariates, and labels them as conditional intraclass correlations.

Intraclass correlations will always fall in [0,1] because variance components are nonnegative. To accommodate the range of an intraclass correlation, we use the logit transformation to obtain confidence intervals. We use the delta method to estimate the standard errors of the intraclass correlations.

Let $\widehat{\rho}^{(g)}$ be a point estimate of the intraclass correlation and $\widehat{\text{SE}}(\widehat{\rho}^{(g)})$ be its standard error. The $(1-\alpha) \times 100\%$ confidence interval for $\text{logit}(\rho^{(g)})$ is

$$\text{logit}(\widehat{\rho}^{(g)}) \pm z_{\alpha/2} \frac{\widehat{\text{SE}}(\widehat{\rho}^{(g)})}{\widehat{\rho}^{(g)}(1-\widehat{\rho}^{(g)})}$$

where $z_{\alpha/2}$ is the $1-\alpha/2$ quantile of the standard normal distribution and $\text{logit}(x) = \ln\{x/(1-x)\}$. Let k_u be the upper endpoint of this interval, and let k_l be the lower. The $(1-\alpha) \times 100\%$ confidence interval for $\rho^{(g)}$ is then given by

$$\left(\frac{1}{1+e^{-k_l}}, \frac{1}{1+e^{-k_u}} \right)$$

References

De Backer, M., C. De Vroey, E. Lesaffre, I. Scheys, and P. De Keyser. 1998. Twelve weeks of continuous oral therapy for toenail onychomycosis caused by dermatophytes: A double-blind comparative trial of terbinafine 250 mg/day versus itraconazole 200 mg/day. *Journal of the American Academy of Dermatology* 38: S57–S63.

Hardin, J. W., and J. M. Hilbe. 2012. *Generalized Linear Models and Extensions*. 3rd ed. College Station, TX: Stata Press.

Lesaffre, E., and B. Spiessens. 2001. On the effect of the number of quadrature points in a logistic random-effects model: An example. *Journal of the Royal Statistical Society, Series C* 50: 325–335.

McCullagh, P., and J. A. Nelder. 1989. *Generalized Linear Models*. 2nd ed. London: Chapman & Hall/CRC.

Pinheiro, J. C., and E. C. Chao. 2006. Efficient Laplacian and adaptive Gaussian quadrature algorithms for multilevel generalized linear mixed models. *Journal of Computational and Graphical Statistics* 15: 58–81.

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

Wolfram, S. 1999. *The Mathematica Book*. 4th ed. Cambridge: Cambridge University Press.

Also see

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

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

Title

meqrpoisson — Multilevel mixed-effects Poisson regression (QR decomposition)

Syntax

`meqrpoisson` *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* [, *re_options*]

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
`exposure(`*varname*$_e$`)`	include ln(*varname*$_e$) in model with coefficient constrained to 1
`offset(`*varname*$_o$`)`	include *varname*$_o$ 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
`collinear`	keep collinear variables

options	Description
Reporting	
level(#)	set confidence level; default is level(95)
irr	report fixed-effects coefficients as incidence-rate ratios
variance	show random-effects parameter estimates as variances and covariances; the default
stddeviations	show random-effects parameter estimates as standard deviations and correlations
noretable	suppress random-effects table
nofetable	suppress fixed-effects table
estmetric	show parameter estimates in the estimation metric
noheader	suppress output header
nogroup	suppress table summarizing groups
nolrtest	do not perform likelihood-ratio test comparing with Poisson regression
display_options	control column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Integration	
intpoints(# [# ...])	set the number of integration (quadrature) points; default is intpoints(7)
laplace	use Laplacian approximation; equivalent to intpoints(1)
Maximization	
maximize_options	control the maximization process; seldom used
retolerance(#)	tolerance for random-effects estimates; default is retolerance(1e-8); seldom used
reiterate(#)	maximum number of iterations for random-effects estimation; default is reiterate(50); seldom used
matsqrt	parameterize variance components using matrix square roots; the default
matlog	parameterize variance components using matrix logarithms
refineopts(*maximize_options*)	control the maximization process during refinement of starting values
coeflegend	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

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

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

`bootstrap`, `by`, `jackknife`, `mi estimate`, `rolling`, and `statsby` are allowed; see [U] 11.1.10 Prefix commands.

`coeflegend` does 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 > Estimation by QR decomposition > Poisson regression

Description

`meqrpoisson`, like `mepoisson`, fits mixed-effects models for count responses, for which the conditional distribution of the response given the random effects is assumed to be Poisson.

`meqrpoisson` provides an alternative estimation method that uses the QR decomposition of the variance-components matrix. This method may aid convergence when variance components are near the boundary of the parameter space.

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.

`exposure(`*varname*$_e$`)` specifies a variable that reflects the amount of exposure over which the *depvar* events were observed for each observation; ln(*varname*$_e$) is included in the fixed-effects portion of the model with the coefficient constrained to be 1.

`offset(`*varname*$_o$`)` specifies that *varname*$_o$ 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`, and `unstructured`.

`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)`, except when the `R.` notation is used, in which case the default is `covariance(identity)` and only `covariance(identity)` and `covariance(exchangeable)` are allowed.

`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.

`collinear` specifies that `meqrpoisson` not omit collinear variables from the random-effects equation. Usually, there is no reason to leave collinear variables in place; in fact, doing so usually causes the estimation to fail because of the matrix singularity caused by the collinearity. However, with certain models (for example, a random-effects model with a full set of contrasts), the variables may be collinear, yet the model is fully identified because of restrictions on the random-effects covariance structure. In such cases, using the `collinear` option allows the estimation to take place with the random-effects equation intact.

Reporting

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

`irr` reports estimated fixed-effects coefficients transformed to incidence-rate 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. `irr` may be specified at estimation or upon replay.

`variance`, the default, displays the random-effects parameter estimates as variances and covariances.

`stddeviations` displays the random-effects parameter estimates as standard deviations and correlations.

`noretable` suppresses the random-effects table.

`nofetable` suppresses the fixed-effects table.

`estmetric` displays all parameter estimates in the estimation metric. Fixed-effects estimates are unchanged from those normally displayed, but random-effects parameter estimates are displayed as log-standard deviations and hyperbolic arctangents of correlations, with equation names that organize them by model level.

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

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

Integration

`intpoints(`*#* [*# ...*]`)` sets the number of integration points for adaptive Gaussian quadrature. The more integration points, the more accurate the approximation to the log likelihood. However, computation time increases with the number of quadrature points, and in models with many levels or many random coefficients, this increase can be substantial.

You may specify one number of integration points applying to all levels of random effects in the model, or you may specify distinct numbers of points for each level. `intpoints(7)` is the default; that is, by default seven quadrature points are used for each level.

`laplace` specifies that log likelihoods be calculated using the Laplacian approximation, equivalent to adaptive Gaussian quadrature with one integration point for each level in the model; `laplace` is equivalent to `intpoints(1)`. Computation time increases as a function of the number of quadrature points raised to a power equaling the dimension of the random-effects specification.

The computational time saved by using laplace can thus be substantial, especially when you have many levels or random coefficients.

The Laplacian approximation has been known to produce biased parameter estimates, but the bias tends to be more prominent in the estimates of the variance components rather than in the estimates of the fixed effects. If your interest lies primarily with the fixed-effects estimates, the Laplace approximation may be a viable faster alternative to adaptive quadrature with multiple integration points.

When the R.*varname* notation is used, the dimension of the random effects increases by the number of distinct values of *varname*. Even when this number is small to moderate, it increases the total random-effects dimension to the point where estimation with more than one quadrature point is prohibitively intensive.

For this reason, when you use the R. notation in your random-effects equations, the laplace option is assumed. You can override this behavior by using the intpoints() option, but doing so is not recommended.

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 meqrpoisson are listed below.

For the technique() option, the default is technique(nr). The bhhh algorithm may not be specified.

from(*init_specs*) is particularly useful when combined with refineopts(iterate(0)) (see the description below), which bypasses the initial optimization stage.

retolerance(#) specifies the convergence tolerance for the estimated random effects used by adaptive Gaussian quadrature. Although not estimated as model parameters, random-effects estimators are used to adapt the quadrature points. Estimating these random effects is an iterative procedure, with convergence declared when the maximum relative change in the random effects is less than retolerance(). The default is retolerance(1e-8). You should seldom have to use this option.

reiterate(#) specifies the maximum number of iterations used when estimating the random effects to be used in adapting the Gaussian quadrature points; see the retolerance() option. The default is reiterate(50). You should seldom have to use this option.

matsqrt (the default), during optimization, parameterizes variance components by using the matrix square roots of the variance–covariance matrices formed by these components at each model level.

matlog, during optimization, parameterizes variance components by using the matrix logarithms of the variance–covariance matrices formed by these components at each model level.

The matsqrt parameterization ensures that variance–covariance matrices are positive semidefinite, while matlog ensures matrices that are positive definite. For most problems, the matrix square root is more stable near the boundary of the parameter space. However, if convergence is problematic, one option may be to try the alternate matlog parameterization. When convergence is not an issue, both parameterizations yield equivalent results.

refineopts(*maximize_options*) controls the maximization process during the refinement of starting values. Estimation in meqrpoisson takes place in two stages. In the first stage, starting values are refined by holding the quadrature points fixed between iterations. During the second stage, quadrature points are adapted with each evaluation of the log likelihood. Maximization options specified within refineopts() control the first stage of optimization; that is, they control the refining of starting values.

maximize_options specified outside refineopts() control the second stage.

The one exception to the above rule is the nolog option, which when specified outside refineopts() applies globally.

from(*init_specs*) is not allowed within refineopts() and instead must be specified globally.

Refining starting values helps make the iterations of the second stage (those that lead toward the solution) more numerically stable. In this regard, of particular interest is refineopts(iterate(*#*)), with two iterations being the default. Should the maximization fail because of instability in the Hessian calculations, one possible solution may be to increase the number of iterations here.

The following option is available with meqrpoisson but is not shown in the dialog box:

coeflegend; see [R] **estimation options**.

Remarks and examples

Remarks are presented under the following headings:

Introduction

Mixed-effects Poisson regression is Poisson 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.

meqrpoisson allows for not just one, but many levels of nested clusters. For example, in a three-level model you can specify random effects for schools and then random effects for classes nested within schools. The observations (students, presumably) would comprise level one of the model, the classes would comprise level two, and the schools would comprise level three.

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 $\mathbf{u}_j$,

$$\Pr(y_{ij} = y|\mathbf{u}_j) = \exp\left(-\mu_{ij}\right) \mu_{ij}^y / y! \tag{1}$$

for $\mu_{ij} = \exp(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j)$, $j = 1, \dots, M$ clusters, and with cluster j consisting of $i = 1, \dots, n_j$ observations. The responses are counts y_{ij}. 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 Poisson regression model, with regression coefficients (fixed effects) $\boldsymbol{\beta}$.

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 $\boldsymbol{\Sigma}$.

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 the *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. The estimation method used by `meqrpoisson` is a multicoefficient and multilevel extension of one of these quadrature types, namely, adaptive Gaussian quadrature (AGQ) based on conditional modes, with the multicoefficient extension from Pinheiro and Bates (1995) and the multilevel extension from Pinheiro and Chao (2006); see *Methods and formulas*.

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

A two-level model

In this section, we begin with a two-level mixed-effects Poisson regression, because a one-level model, in multilevel-model terminology, is just standard Poisson regression; see [R] **poisson**.

▷ Example 1

Breslow and Clayton (1993) fit a mixed-effects Poisson model to data from a randomized trial of the drug progabide for the treatment of epilepsy.

```
. use http://www.stata-press.com/data/r13/epilepsy
(Epilepsy data; progabide drug treatment)
. describe
Contains data from http://www.stata-press.com/data/r13/epilepsy.dta
  obs:           236                          Epilepsy data; progabide drug
                                                treatment
 vars:             8                          31 May 2013 14:09
 size:         4,956                          (_dta has notes)
--------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
--------------------------------------------------------------------------------
subject         byte    %9.0g                 Subject ID: 1-59
seizures        int     %9.0g                 No. of seizures
treat           byte    %9.0g                 1: progabide; 0: placebo
visit           float   %9.0g                 Dr. visit; coded as (-.3, -.1,
                                                .1, .3)
lage            float   %9.0g                 log(age), mean-centered
lbas            float   %9.0g                 log(0.25*baseline seizures),
                                                mean-centered
lbas_trt        float   %9.0g                 lbas/treat interaction
v4              byte    %8.0g                 Fourth visit indicator
--------------------------------------------------------------------------------
Sorted by:  subject
```

Originally from Thall and Vail (1990), data were collected on 59 subjects (31 on progabide, 28 on placebo). The number of epileptic seizures (`seizures`) was recorded during the two weeks prior to each of four doctor visits (`visit`). The treatment group is identified by the indicator variable `treat`.

Data were also collected on the logarithm of age (`lage`) and the logarithm of one-quarter the number of seizures during the eight weeks prior to the study (`lbas`). The variable `lbas_trt` represents the interaction between `lbas` and treatment. `lage`, `lbas`, and `lbas_trt` are mean centered. Because the study originally noted a substantial decrease in seizures prior to the fourth doctor visit, an indicator, `v4`, for the fourth visit was also recorded.

Breslow and Clayton (1993) fit a random-effects Poisson model for the number of observed seizures

$$\log(\mu_{ij}) = \beta_0 + \beta_1 \texttt{treat}_{ij} + \beta_2 \texttt{lbas}_{ij} + \beta_3 \texttt{lbas\_trt}_{ij} + \beta_4 \texttt{lage}_{ij} + \beta_5 \texttt{v4}_{ij} + u_j$$

for $j = 1, \ldots, 59$ subjects and $i = 1, \ldots, 4$ visits. The random effects u_j are assumed to be normally distributed with mean 0 and variance σ_u^2.

```
. meqrpoisson seizures treat lbas lbas_trt lage v4 || subject:
Refining starting values:
Iteration 0:   log likelihood = -680.40577  (not concave)
Iteration 1:   log likelihood = -668.60112
Iteration 2:   log likelihood =  -666.3822
Performing gradient-based optimization:
Iteration 0:   log likelihood =  -666.3822
Iteration 1:   log likelihood =  -665.4603
Iteration 2:   log likelihood = -665.29075
Iteration 3:   log likelihood = -665.29068
Mixed-effects Poisson regression                Number of obs      =       236
Group variable: subject                         Number of groups   =        59
                                                Obs per group: min =         4
                                                               avg =       4.0
                                                               max =         4
Integration points =   7                        Wald chi2(5)       =    121.67
Log likelihood = -665.29068                     Prob > chi2        =    0.0000
```

seizures	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
treat	-.9330388	.4008345	-2.33	0.020	-1.71866	-.1474177
lbas	.8844331	.1312313	6.74	0.000	.6272246	1.141642
lbas_trt	.3382609	.2033384	1.66	0.096	-.0602751	.7367969
lage	.4842391	.3472774	1.39	0.163	-.1964121	1.16489
v4	-.1610871	.0545758	-2.95	0.003	-.2680537	-.0541206
_cons	2.154575	.2200425	9.79	0.000	1.723299	2.58585

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
subject: Identity				
var(_cons)	.2528263	.0589559	.1600784	.3993115

```
LR test vs. Poisson regression:  chibar2(01) =   304.74 Prob>=chibar2 = 0.0000
```

The number of seizures before the fourth visit does exhibit a significant drop, and the patients on progabide demonstrate a decrease in frequency of seizures compared with the placebo group. The subject-specific random effects also appear significant: $\widehat{\sigma}_u^2 = 0.25$ with standard error 0.06. The above results are also in good agreement with those of Breslow and Clayton (1993, table 4), who fit this model by the method of penalized quasi-likelihood (PQL).

Because this is a simple random-intercept model, you can obtain equivalent results by using `xtpoisson` with the `re` and `normal` options.

◁

▷ Example 2

In their study of PQL, Breslow and Clayton (1993) also fit a model where they dropped the fixed effect on v4 and replaced it with a random subject-specific linear trend over the four doctor visits. The model they fit is

$$\log(\mu_{ij}) = \beta_0 + \beta_1\texttt{treat}_{ij} + \beta_2\texttt{lbas}_{ij} + \beta_3\texttt{lbas\_trt}_{ij} + \beta_4\texttt{lage}_{ij} + \beta_5\texttt{visit}_{ij} + u_j + v_j\texttt{visit}_{ij}$$

where (u_j, v_j) are bivariate normal with 0 mean and variance–covariance matrix

$$\boldsymbol{\Sigma} = \text{Var}\begin{bmatrix} u_j \\ v_j \end{bmatrix} = \begin{bmatrix} \sigma_u^2 & \sigma_{uv} \\ \sigma_{uv} & \sigma_v^2 \end{bmatrix}$$

```
. meqrpoisson seizures treat lbas lbas_trt lage visit || subject: visit,
> cov(unstructured) intpoints(9)
Refining starting values:
Iteration 0:   log likelihood = -672.17188  (not concave)
Iteration 1:   log likelihood = -660.46056
Iteration 2:   log likelihood = -655.86727
Performing gradient-based optimization:
Iteration 0:   log likelihood = -655.86727
Iteration 1:   log likelihood =  -655.6822
Iteration 2:   log likelihood = -655.68103
Iteration 3:   log likelihood = -655.68103
Mixed-effects Poisson regression                Number of obs      =       236
Group variable: subject                         Number of groups   =        59
                                                Obs per group: min =         4
                                                               avg =       4.0
                                                               max =         4
Integration points =   9                        Wald chi2(5)       =    115.56
Log likelihood = -655.68103                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
    seizures |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       treat |  -.9286588   .4021643    -2.31   0.021    -1.716886   -.1404313
        lbas |   .8849767    .131252     6.74   0.000     .6277275    1.142226
    lbas_trt |   .3379757   .2044445     1.65   0.098    -.0627281    .7386795
        lage |   .4767192    .353622     1.35   0.178    -.2163673    1.169806
       visit |  -.2664098   .1647096    -1.62   0.106    -.5892347    .0564151
       _cons |   2.099555   .2203712     9.53   0.000     1.667635    2.531474
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
subject: Unstructured        |
                  var(visit) |   .5314808   .2293851      .2280931    1.238406
                  var(_cons) |   .2514928   .0587892      .1590552    .3976522
            cov(visit,_cons) |   .0028715   .0887018     -.1709808    .1767238
------------------------------------------------------------------------------
LR test vs. Poisson regression:      chi2(3) =   324.54   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

In the above, we specified the `cov(unstructured)` option to allow correlation between u_j and v_j, although on the basis of the above output it probably was not necessary—the default `Independent` structure would have sufficed. In the interest of getting more accurate estimates, we also increased the number of quadrature points to nine, although the estimates do not change much when compared with estimates based on the default seven quadrature points.

The essence of the above-fitted model is that after adjusting for other covariates, the log trend in seizures is modeled as a random subject-specific line, with intercept distributed as $N(\beta_0, \sigma_u^2)$ and slope distributed as $N(\beta_5, \sigma_v^2)$. From the above output, $\widehat{\beta}_0 = 2.10$, $\widehat{\sigma}_u^2 = 0.25$, $\widehat{\beta}_5 = -0.27$, and $\widehat{\sigma}_v^2 = 0.53$.

You can predict the random effects u_j and v_j by using `predict` after `meqrpoisson`; see [ME] **meqrpoisson postestimation**. Better still, you can obtain a predicted number of seizures that takes these random effects into account.

◁

A three-level model

`meqrpoisson` can also fit higher-level models with multiple levels of nested random effects.

▷ Example 3

Rabe-Hesketh and Skrondal (2012, exercise 13.7) describe data from the *Atlas of Cancer Mortality in the European Economic Community* (EEC) (Smans, Mair, and Boyle 1993). The data were analyzed in Langford, Bentham, and McDonald (1998) and record the number of deaths among males due to malignant melanoma during 1971–1980.

```
. use http://www.stata-press.com/data/r13/melanoma
(Skin cancer (melanoma) data)

. describe

Contains data from http://www.stata-press.com/data/r13/melanoma.dta
  obs:           354                          Skin cancer (melanoma) data
 vars:             6                          30 May 2013 17:10
 size:         4,956                          (_dta has notes)
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
nation          byte    %11.0g     n          Nation ID
region          byte    %9.0g                 Region ID: EEC level-I areas
county          int     %9.0g                 County ID: EEC level-II/level-III
                                                areas
deaths          int     %9.0g                 No. deaths during 1971-1980
expected        float   %9.0g                 No. expected deaths
uv              float   %9.0g                 UV dose, mean-centered
-------------------------------------------------------------------------------
Sorted by:
```

Nine European nations (variable `nation`) are represented, and data were collected over geographical regions defined by EEC statistical services as level I areas (variable `region`), with deaths being recorded for each of 354 counties, which are level II or level III EEC-defined areas (variable `county`, which identifies the observations). Counties are nested within regions, and regions are nested within nations.

The variable `deaths` records the number of deaths for each county, and `expected` records the expected number of deaths (the exposure) on the basis of crude rates for the combined countries. Finally, the variable `uv` is a measure of exposure to ultraviolet (UV) radiation.

In modeling the number of deaths, one possibility is to include dummy variables for the nine nations as fixed effects. Another is to treat these as random effects and fit the three-level random-intercept Poisson model,

$$\log(\mu_{ijk}) = \log(\texttt{expected}_{ijk}) + \beta_0 + \beta_1 \texttt{uv}_{ijk} + u_k + v_{jk}$$

for nation k, region j, and county i. The model includes an exposure term for expected deaths.

```
. meqrpoisson deaths uv, exposure(expected) || nation: || region:
Refining starting values:
Iteration 0:   log likelihood = -1169.0851  (not concave)
Iteration 1:   log likelihood =  -1156.523  (not concave)
Iteration 2:   log likelihood = -1101.8313
Performing gradient-based optimization:
Iteration 0:   log likelihood = -1101.8313
Iteration 1:   log likelihood = -1100.7407
Iteration 2:   log likelihood = -1098.0445
Iteration 3:   log likelihood = -1097.7212
Iteration 4:   log likelihood =  -1097.714
Iteration 5:   log likelihood =  -1097.714
Mixed-effects Poisson regression                Number of obs      =       354
```

Group Variable	No. of Groups	Observations per Group Minimum	Average	Maximum	Integration Points
nation	9	3	39.3	95	7
region	78	1	4.5	13	7

```
                                                Wald chi2(1)       =      6.12
Log likelihood =  -1097.714                     Prob > chi2        =    0.0134
```

deaths	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
uv	-.0281991	.0114027	-2.47	0.013	-.050548	-.0058503
_cons	-.0639473	.1335245	-0.48	0.632	-.3256505	.1977559
ln(expected)	1	(exposure)				

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
nation: Identity				
var(_cons)	.1370339	.0722797	.0487365	.3853022
region: Identity				
var(_cons)	.0483853	.010927	.0310802	.0753257

```
LR test vs. Poisson regression:      chi2(2) =  1252.12   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

By including an exposure variable that is an expected rate, we are in effect specifying a linear model for the log of the standardized mortality ratio, the ratio of observed deaths to expected deaths that is based on a reference population. Here the reference population is all nine nations.

We now add a random intercept for counties nested within regions, making this a four-level model. Because counties also identify the observations, the corresponding variance component can be interpreted as a measure of overdispersion, variability above and beyond that allowed by a Poisson process; see [R] **nbreg** and [ME] **menbreg**.

```
. meqrpoisson deaths uv, exposure(expected) || nation: || region: || county:,
> laplace
Refining starting values:
Iteration 0:   log likelihood = -1381.1202  (not concave)
Iteration 1:   log likelihood = -1144.7025  (not concave)
Iteration 2:   log likelihood = -1138.6807
Performing gradient-based optimization:
Iteration 0:   log likelihood = -1138.6807
Iteration 1:   log likelihood =   -1123.31
Iteration 2:   log likelihood = -1095.0497
Iteration 3:   log likelihood = -1086.9521
Iteration 4:   log likelihood = -1086.7321
Iteration 5:   log likelihood = -1086.7309
Iteration 6:   log likelihood = -1086.7309
Mixed-effects Poisson regression                Number of obs      =       354

                No. of       Observations per Group       Integration
 Group Variable  Groups    Minimum    Average    Maximum      Points

         nation       9          3       39.3         95           1
         region      78          1        4.5         13           1
         county     354          1        1.0          1           1

                                                Wald chi2(1)       =      8.63
Log likelihood = -1086.7309                     Prob > chi2        =    0.0033

      deaths        Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]

          uv    -.0334681   .0113919    -2.94   0.003    -.0557959   -.0111404
       _cons    -.0864109   .1298713    -0.67   0.506    -.3409539    .1681321
ln(expected)            1  (exposure)

  Random-effects Parameters     Estimate   Std. Err.     [95% Conf. Interval]

nation: Identity
                  var(_cons)    .1287416   .0680887        .04566    .3629957

region: Identity
                  var(_cons)    .0405965   .0105002      .0244527    .0673986

county: Identity
                  var(_cons)    .0146027   .0050766      .0073878    .0288637

LR test vs. Poisson regression:      chi2(3) =  1274.08   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
Note: log-likelihood calculations are based on the Laplacian approximation.
```

In the above, we used a Laplacian approximation, which is not only faster but also produces estimates that closely agree with those obtained with the default seven quadrature points.

See *Computation time and the Laplacian approximation* in [ME] **me** for a discussion comparing Laplacian approximation with adaptive quadrature.

◁

Stored results

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

Scalars

e(N)	number of observations
e(k)	number of parameters
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(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(reparm_rc)	return code, final reparameterization
e(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros

e(cmd)	meqrpoisson
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(ivars)	grouping variables
e(exposurevar)	exposure variable
e(model)	Poisson
e(title)	title in estimation output
e(offset)	offset
e(redim)	random-effects dimensions
e(vartypes)	variance-structure types
e(revars)	random-effects covariates
e(n_quad)	number of integration points
e(laplace)	laplace, if Laplace approximation
e(chi2type)	Wald; type of model χ^2
e(vce)	bootstrap or jackknife if defined
e(vcetype)	title used to label Std. Err.
e(method)	ML
e(opt)	type of optimization
e(ml_method)	type of ml method
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(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved

Matrices

e(b)	coefficient vector
e(Cns)	constraints matrix
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

Functions

e(sample)	marks estimation sample

Methods and formulas

In a two-level Poisson model, 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

$$
\begin{aligned}
f(\mathbf{y}_j|\mathbf{u}_j) &= \prod_{i=1}^{n_j} \left[\{\exp\left(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j\right)\}^{y_{ij}} \exp\left\{-\exp\left(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j\right)\right\} / y_{ij}! \right] \\
&= \exp\left[\sum_{i=1}^{n_j} \left\{ y_{ij}\left(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j\right) - \exp\left(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j\right) - \log(y_{ij}!) \right\} \right]
\end{aligned}
$$

Defining $c\left(\mathbf{y}_j\right) = \sum_{i=1}^{n_j} \log(y_{ij}!)$, where $c(\mathbf{y}_j)$ does not depend on the model parameters, we can express the above compactly in matrix notation,

$$
f(\mathbf{y}_j|\mathbf{u}_j) = \exp\left\{ \mathbf{y}_j'\left(\mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j\right) - \mathbf{1}'\exp\left(\mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j\right) - c\left(\mathbf{y}_j\right) \right\}
$$

where $\mathbf{X}_j$ is formed by stacking the row vectors $\mathbf{x}_{ij}$ and $\mathbf{Z}_j$ is formed by stacking the row vectors $\mathbf{z}_{ij}$. We extend the definition of $\exp(\cdot)$ to be a vector function 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 $f(\mathbf{y}_j, \mathbf{u}_j)$,

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

where

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

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

The integration in (2) has no closed form and thus must be approximated. The Laplacian approximation (Tierney and Kadane 1986; Pinheiro and Bates 1995) is based on a second-order Taylor expansion of $h\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right)$ about the value of $\mathbf{u}_j$ that maximizes it. Taking first and second derivatives, we obtain

$$
\begin{aligned}
h'\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right) &= \frac{\partial h\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right)}{\partial \mathbf{u}_j} &= \mathbf{Z}_j'\left\{\mathbf{y}_j - \mathbf{m}(\boldsymbol{\beta}, \mathbf{u}_j)\right\} - \boldsymbol{\Sigma}^{-1}\mathbf{u}_j \\
h''\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right) &= \frac{\partial^2 h\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right)}{\partial \mathbf{u}_j \partial \mathbf{u}_j'} &= -\left\{\mathbf{Z}_j'\mathbf{V}(\boldsymbol{\beta}, \mathbf{u}_j)\mathbf{Z}_j + \boldsymbol{\Sigma}^{-1}\right\}
\end{aligned}
$$

where $\mathbf{m}(\boldsymbol{\beta}, \mathbf{u}_j)$ is the vector function with the ith element equal to the conditional mean of y_{ij} given $\mathbf{u}_j$, that is, $\exp(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j)$. $\mathbf{V}(\boldsymbol{\beta}, \mathbf{u}_j)$ is the diagonal matrix whose diagonal entries v_{ij} are the conditional variances of y_{ij} given $\mathbf{u}_j$, namely,

$$
v_{ij} = \exp\left(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j\right)
$$

because equality of mean and variance is a characteristic of the Poisson distribution.

The maximizer of $h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j)$ is $\widehat{\mathbf{u}}_j$ such that $h'(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j) = \mathbf{0}$. The integrand in (2) is proportional to the posterior density $f(\mathbf{u}_j|\mathbf{y}_j)$, so $\widehat{\mathbf{u}}_j$ also represents the posterior mode, a plausible estimator of $\mathbf{u}_j$ in its own right.

Given the above derivatives, the second-order Taylor approximation then takes the form

$$h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j) \approx h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j) + \frac{1}{2}(\mathbf{u}_j - \widehat{\mathbf{u}}_j)' h''(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j)(\mathbf{u}_j - \widehat{\mathbf{u}}_j) \tag{3}$$

The first-derivative term vanishes because $h'(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j) = \mathbf{0}$. Therefore,

$$\begin{aligned}\int \exp\{h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j)\}\, d\mathbf{u}_j &\approx \exp\{h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j)\} \\ &\quad\times \int \exp\left[-\frac{1}{2}(\mathbf{u}_j - \widehat{\mathbf{u}}_j)'\{-h''(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j)\}(\mathbf{u}_j - \widehat{\mathbf{u}}_j)\right] d\mathbf{u}_j \\ &= \exp\{h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j)\}(2\pi)^{q/2}\left|-h''(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j)\right|^{-1/2}\end{aligned} \tag{4}$$

because the latter integrand can be recognized as the "kernel" of a multivariate normal density.

Combining the above with (2) (and taking logs) gives the Laplacian log-likelihood contribution of the jth cluster,

$$\mathcal{L}_j^{\text{Lap}}(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = -\frac{1}{2}\log|\boldsymbol{\Sigma}| - \log|\mathbf{R}_j| + h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j) - c(\mathbf{y}_j)$$

where $\mathbf{R}_j$ is an upper-triangular matrix such that $-h''(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j) = \mathbf{R}_j\mathbf{R}_j'$. Pinheiro and Chao (2006) show that $\widehat{\mathbf{u}}_j$ and $\mathbf{R}_j$ can be efficiently computed as the iterative solution to a least-squares problem by using matrix decomposition methods similar to those used in fitting LME models (Bates and Pinheiro 1998; Pinheiro and Bates 2000; [ME] **mixed**).

The fidelity of the Laplacian approximation is determined wholly by the accuracy of the approximation in (3). An alternative that does not depend so heavily on this approximation is integration via AGQ (Naylor and Smith 1982; Liu and Pierce 1994).

The application of AGQ to this particular problem is from Pinheiro and Bates (1995). When we reexamine the integral in question, a transformation of integration variables yields

$$\begin{aligned}\int \exp\{h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j)\}\, d\mathbf{u}_j &= |\mathbf{R}_j|^{-1}\int \exp\left\{h\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j + \mathbf{R}_j^{-1}\mathbf{t}\right)\right\} d\mathbf{t} \\ &= (2\pi)^{q/2}|\mathbf{R}_j|^{-1}\int \exp\left\{h\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j + \mathbf{R}_j^{-1}\mathbf{t}\right) + \mathbf{t}'\mathbf{t}/2\right\}\phi(\mathbf{t})d\mathbf{t}\end{aligned} \tag{5}$$

where $\phi(\cdot)$ is the standard multivariate normal density. Because the integrand is now expressed as some function multiplied by a normal density, it can be estimated by applying the rules of standard Gauss–Hermite quadrature. For a predetermined number of quadrature points N_Q, define $a_k = \sqrt{2}a_k^*$ and $w_k = w_k^*/\sqrt{\pi}$, for $k = 1, \ldots, N_Q$, where (a_k^*, w_k^*) are a set of abscissas and weights for Gauss–Hermite quadrature approximations of $\int \exp(-x^2)f(x)dx$, as obtained from Abramowitz and Stegun (1972, 924).

Define $\mathbf{a_k} = (a_{k_1}, a_{k_2}, \ldots, a_{k_q})'$; that is, $\mathbf{a_k}$ is a vector that spans the N_Q abscissas over the dimension q of the random effects. Applying quadrature rules to (5) yields the AGQ approximation,

$$\begin{aligned}\int \exp\left\{h\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right)\right\} d\mathbf{u}_j \\ &\approx (2\pi)^{q/2} \left|\mathbf{R}_j\right|^{-1} \sum_{k_1=1}^{N_Q} \cdots \sum_{k_q=1}^{N_Q} \left[\exp\left\{h\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j + \mathbf{R}_j^{-1}\mathbf{a_k}\right) + \mathbf{a}_\mathbf{k}'\mathbf{a_k}/2\right\} \prod_{p=1}^{q} w_{k_p}\right] \\ &\equiv (2\pi)^{q/2} \widehat{G}_j(\boldsymbol{\beta}, \boldsymbol{\Sigma})\end{aligned}$$

resulting in the AGQ log-likelihood contribution of the jth cluster,

$$\mathcal{L}_j^{\mathrm{AGQ}}(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = -\frac{1}{2}\log|\boldsymbol{\Sigma}| + \log\left\{\widehat{G}_j(\boldsymbol{\beta}, \boldsymbol{\Sigma})\right\} - c(\mathbf{y}_j)$$

The "adaptive" part of adaptive Gaussian quadrature lies in the translation and rescaling of the integration variables in (5) by using $\widehat{\mathbf{u}}_j$ and $\mathbf{R}_j^{-1}$, respectively. This transformation of quadrature abscissas (centered at 0 in standard form) is chosen to better capture the features of the integrand, through which (4) can be seen to resemble a multivariate normal distribution with mean $\widehat{\mathbf{u}}_j$ and variance $\mathbf{R}_j^{-1}\mathbf{R}_j^{-T}$. AGQ is therefore not as dependent as the Laplace method upon the approximation in (3). In AGQ, (3) serves merely to redirect the quadrature abscissas, with the AGQ approximation improving as the number of quadrature points, N_Q, increases. In fact, Pinheiro and Bates (1995) point out that AGQ with only one quadrature point ($a = 0$ and $w = 1$) reduces to the Laplacian approximation.

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

Maximization of $\mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\Sigma})$ is performed with respect to $(\boldsymbol{\beta}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a vector comprising the unique elements of the matrix square root of $\boldsymbol{\Sigma}$. This is done to ensure that $\boldsymbol{\Sigma}$ is always positive semidefinite. If the `matlog` option is specified, then $\boldsymbol{\theta}$ instead consists of the unique elements of the matrix logarithm of $\boldsymbol{\Sigma}$. For well-conditioned problems, both methods produce equivalent results, yet our experience deems the former as more numerically stable near the boundary of the parameter space.

Once maximization is achieved, parameter estimates are mapped from $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\theta}})$ to $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\gamma}})$, where $\widehat{\boldsymbol{\gamma}}$ is a vector containing the unique (estimated) elements of $\boldsymbol{\Sigma}$, expressed as logarithms of standard deviations for the diagonal elements and hyperbolic arctangents of the correlations for off-diagonal elements. This last step is necessary to (a) obtain a parameterization under which parameter estimates can be displayed and interpreted individually, rather than as elements of a matrix square root (or logarithm), and (b) parameterize these elements such that their ranges each encompass the entire real line.

Parameter estimates are stored in `e(b)` as $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\gamma}})$, with the corresponding variance–covariance matrix stored in `e(V)`. Parameter estimates can be displayed in this metric by specifying the `estmetric` option. However, in `meqrpoisson` output, variance components are most often displayed either as variances and covariances (the default) or as standard deviations and correlations (option `stddeviations`).

The approach outlined above can be extended from two-level models to models with three or more levels; see Pinheiro and Chao (2006) for details.

References

Abramowitz, M., and I. A. Stegun, ed. 1972. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. 10th ed. Washington, DC: National Bureau of Standards.

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.

Bates, D. M., and J. C. Pinheiro. 1998. Computational methods for multilevel modelling. In *Technical Memorandum BL0112140-980226-01TM*. Murray Hill, NJ: Bell Labs, Lucent Technologies.
http://stat.bell-labs.com/NLME/CompMulti.pdf.

Breslow, N. E., and D. G. Clayton. 1993. Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association* 88: 9–25.

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.

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.

Langford, I. H., G. Bentham, and A. McDonald. 1998. Multi-level modelling of geographically aggregated health data: A case study on malignant melanoma mortality and UV exposure in the European community. *Statistics in Medicine* 17: 41–57.

Leyland, A. H., and H. Goldstein, ed. 2001. *Multilevel Modelling of Health Statistics*. New York: Wiley.

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.

Liu, Q., and D. A. Pierce. 1994. A note on Gauss–Hermite quadrature. *Biometrika* 81: 624–629.

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.

Naylor, J. C., and A. F. M. Smith. 1982. Applications of a method for the efficient computation of posterior distributions. *Journal of the Royal Statistical Society, Series C* 31: 214–225.

Palmer, T. M., C. M. Macdonald-Wallis, D. A. Lawlor, and K. Tilling. 2014. Estimating adjusted associations between random effects from multilevel models: The reffadjust package. *Stata Journal* 14: 119–140.

Pinheiro, J. C., and D. M. Bates. 1995. Approximations to the log-likelihood function in the nonlinear mixed-effects model. *Journal of Computational and Graphical Statistics* 4: 12–35.

———. 2000. *Mixed-Effects Models in S and S-PLUS*. New York: Springer.

Pinheiro, J. C., and E. C. Chao. 2006. Efficient Laplacian and adaptive Gaussian quadrature algorithms for multilevel generalized linear mixed models. *Journal of Computational and Graphical Statistics* 15: 58–81.

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

Rabe-Hesketh, S., A. Skrondal, and A. Pickles. 2005. Maximum likelihood estimation of limited and discrete dependent variable models with nested random effects. *Journal of Econometrics* 128: 301–323.

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

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.

Skrondal, A., and S. Rabe-Hesketh. 2004. *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models*. Boca Raton, FL: Chapman & Hall/CRC.

Smans, M., C. S. Mair, and P. Boyle. 1993. *Atlas of Cancer Mortality in the European Economic Community*. Lyon, France: IARC Scientific Publications.

Thall, P. F., and S. C. Vail. 1990. Some covariance models for longitudinal count data with overdispersion. *Biometrics* 46: 657–671.

Tierney, L., and J. B. Kadane. 1986. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association* 81: 82–86.

Also see

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

[ME] **menbreg** — Multilevel mixed-effects negative binomial regression

[ME] **mepoisson** — Multilevel mixed-effects Poisson regression

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

[MI] **estimation** — Estimation commands for use with mi estimate

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

[XT] **xtpoisson** — Fixed-effects, random-effects, and population-averaged Poisson models

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

Title

meqrpoisson postestimation — Postestimation tools for meqrpoisson

Description

The following postestimation commands are of special interest after `meqrpoisson`:

Command	Description
`estat group`	summarize the composition of the nested groups
`estat recovariance`	display the estimated random-effects covariance matrix (or matrices)

The following standard postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat ic`	Akaike's and Schwarz's Bayesian information criteria (AIC and BIC)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

Special-interest postestimation commands

estat group reports the number of groups and minimum, average, and maximum group sizes for each level of the model. Model levels are identified by the corresponding group variable in the data. Because groups are treated as nested, the information in this summary may differ from what you would get if you used the tabulate command on each group variable individually.

estat recovariance displays the estimated variance–covariance matrix of the random effects for each level in the model. Random effects can be either random intercepts, in which case the corresponding rows and columns of the matrix are labeled as _cons, or random coefficients, in which case the label is the name of the associated variable in the data.

Syntax for predict

Syntax for obtaining estimated random effects and their standard errors

predict [*type*] {*stub** | *newvarlist*} [*if*] [*in*], {reffects | reses}
 [relevel(*levelvar*)]

Syntax for obtaining other predictions

predict [*type*] *newvar* [*if*] [*in*] [, *statistic* nooffset fixedonly]

statistic	Description
Main	
mu	predicted mean; the default
xb	linear predictor for the fixed portion of the model only
stdp	standard error of the fixed-portion linear prediction
pearson	Pearson residuals
deviance	deviance residuals
anscombe	Anscombe residuals

These statistics are available both in and out of sample; type predict ... if e(sample) ... if wanted only for the estimation sample.

Menu for predict

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

reffects calculates posterior modal estimates of the random effects. By default, estimates for all random effects in the model are calculated. However, if the relevel(*levelvar*) option is specified, then estimates for only level *levelvar* in the model are calculated. For example, if classes are nested within schools, then typing

```
. predict b*, reffects relevel(school)
```

would yield random-effects estimates at the school level. You must specify q new variables, where q is the number of random-effects terms in the model (or level). However, it is much easier to just specify *stub** and let Stata name the variables *stub*1, *stub*2, ..., *stubq* for you.

`reses` calculates standard errors for the random-effects estimates obtained by using the `reffects` option. By default, standard errors for all random effects in the model are calculated. However, if the `relevel(`*levelvar*`)` option is specified, then standard errors for only level *levelvar* in the model are calculated. For example, if `classes` are nested within `schools`, then typing

```
. predict se*, reses relevel(school)
```

would yield standard errors at the school level. You must specify q new variables, where q is the number of random-effects terms in the model (or level). However, it is much easier to just specify *stub** and let Stata name the variables *stub*1, *stub*2, ..., *stubq* for you.

The `reffects` and `reses` options often generate multiple new variables at once. When this occurs, the random effects (or standard errors) contained in the generated variables correspond to the order in which the variance components are listed in the output of `meqrpoisson`. Still, examining the variable labels of the generated variables (with the `describe` command, for instance) can be useful in deciphering which variables correspond to which terms in the model.

`relevel(`*levelvar*`)` specifies the level in the model at which predictions for random effects and their standard errors are to be obtained. *levelvar* is the name of the model level and is either the name of the variable describing the grouping at that level or is `_all`, a special designation for a group comprising all the estimation data.

`mu`, the default, calculates the predicted mean, that is, the predicted count. By default, this is based on a linear predictor that includes both the fixed effects and the random effects, and the predicted mean is conditional on the values of the random effects. Use the `fixedonly` option (see below) if you want predictions that include only the fixed portion of the model, that is, if you want random effects set to 0.

`xb` calculates the linear prediction $\mathbf{x}\boldsymbol{\beta}$ based on the estimated fixed effects (coefficients) in the model. This is equivalent to fixing all random effects in the model to their theoretical (prior) mean value of 0.

`stdp` calculates the standard error of the fixed-effects linear predictor $\mathbf{x}\boldsymbol{\beta}$.

`pearson` calculates Pearson residuals. Pearson residuals large in absolute value may indicate a lack of fit. By default, residuals include both the fixed portion and the random portion of the model. The `fixedonly` option modifies the calculation to include the fixed portion only.

`deviance` calculates deviance residuals. Deviance residuals are recommended by McCullagh and Nelder (1989) as having the best properties for examining the goodness of fit of a GLM. They are approximately normally distributed if the model is correctly specified. They may be plotted against the fitted values or against a covariate to inspect the model's fit. By default, residuals include both the fixed portion and the random portion of the model. The `fixedonly` option modifies the calculation to include the fixed portion only.

`anscombe` calculates Anscombe residuals, which are designed to closely follow a normal distribution. By default, residuals include both the fixed portion and the random portion of the model. The `fixedonly` option modifies the calculation to include the fixed portion only.

`nooffset` is relevant only if you specified `offset(`*$varname_o$*`)` or `exposure(`*$varname_e$*`)` for `meqrpoisson`. It modifies the calculations made by `predict` so that they ignore the offset/exposure variable; the linear prediction is treated as $\mathbf{X}\beta + \mathbf{Zu}$ rather than $\mathbf{X}\beta + \mathbf{Zu} + \text{offset}$, or $\mathbf{X}\beta + \mathbf{Zu} + \ln(\text{exposure})$, whichever is relevant.

`fixedonly` modifies predictions to include only the fixed portion of the model, equivalent to setting all random effects equal to 0; see the `mu` option.

Syntax for estat

Summarize the composition of the nested groups

```
estat group
```

Display the estimated random-effects covariance matrix (or matrices)

estat recovariance [, relevel(*levelvar*) correlation *matlist_options*]

Menu for estat

Statistics > Postestimation > Reports and statistics

Options for estat recovariance

`relevel(`*levelvar*`)` specifies the level in the model for which the random-effects covariance matrix is to be displayed and returned in `r(cov)`. By default, the covariance matrices for all levels in the model are displayed. *levelvar* is the name of the model level and is either the name of the variable describing the grouping at that level or is `_all`, a special designation for a group comprising all the estimation data.

`correlation` displays the covariance matrix as a correlation matrix and returns the correlation matrix in `r(corr)`.

matlist_options are style and formatting options that control how the matrix (or matrices) is displayed; see [P] **matlist** for a list of that are available.

Remarks and examples

Various predictions, statistics, and diagnostic measures are available after fitting a Poisson mixed-effects model with `meqrpoisson`. For the most part, calculation centers around obtaining estimates of the subject/group-specific random effects. Random effects are not estimated when the model is fit but instead need to be predicted after estimation.

▷ Example 1

In example 2 of [ME] **meqrpoisson**, we modeled the number of observed epileptic seizures as a function of treatment with the drug progabide and other covariates,

$$\log(\mu_{ij}) = \beta_0 + \beta_1 \texttt{treat}_{ij} + \beta_2 \texttt{lbas}_{ij} + \beta_3 \texttt{lbas\_trt}_{ij} + \beta_4 \texttt{lage}_{ij} + \beta_5 \texttt{visit}_{ij} + u_j + v_j \texttt{visit}_{ij}$$

where (u_j, v_j) are bivariate normal with 0 mean and variance–covariance matrix

$$\mathbf{\Sigma} = \mathrm{Var}\begin{bmatrix} u_j \\ v_j \end{bmatrix} = \begin{bmatrix} \sigma_u^2 & \sigma_{uv} \\ \sigma_{uv} & \sigma_v^2 \end{bmatrix}$$

```
. use http://www.stata-press.com/data/r13/epilepsy
(Epilepsy data; progabide drug treatment)

. meqrpoisson seizures treat lbas lbas_trt lage visit || subject: visit,
> cov(unstructured) intpoints(9)

Refining starting values:

Iteration 0:   log likelihood = -672.17188  (not concave)
Iteration 1:   log likelihood = -660.46056
Iteration 2:   log likelihood = -655.86727

Performing gradient-based optimization:

Iteration 0:   log likelihood = -655.86727
Iteration 1:   log likelihood =  -655.6822
Iteration 2:   log likelihood = -655.68103
Iteration 3:   log likelihood = -655.68103

Mixed-effects Poisson regression                Number of obs      =       236
Group variable: subject                         Number of groups   =        59

                                                Obs per group: min =         4
                                                               avg =       4.0
                                                               max =         4

Integration points =   9                        Wald chi2(5)       =    115.56
Log likelihood = -655.68103                     Prob > chi2        =    0.0000
```

seizures	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
treat	-.9286588	.4021643	-2.31	0.021	-1.716886	-.1404313
lbas	.8849767	.131252	6.74	0.000	.6277275	1.142226
lbas_trt	.3379757	.2044445	1.65	0.098	-.0627281	.7386795
lage	.4767192	.353622	1.35	0.178	-.2163673	1.169806
visit	-.2664098	.1647096	-1.62	0.106	-.5892347	.0564151
_cons	2.099555	.2203712	9.53	0.000	1.667635	2.531474

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
subject: Unstructured				
var(visit)	.5314808	.2293851	.2280931	1.238406
var(_cons)	.2514928	.0587892	.1590552	.3976522
cov(visit,_cons)	.0028715	.0887018	-.1709808	.1767238

```
LR test vs. Poisson regression:      chi2(3) =   324.54   Prob > chi2 = 0.0000

Note: LR test is conservative and provided only for reference.
```

The purpose of this model was to allow subject-specific linear log trends over each subject's four doctor visits, after adjusting for the other covariates. The intercepts of these lines are distributed $N(\beta_0, \sigma_u^2)$, and the slopes are distributed $N(\beta_5, \sigma_v^2)$, based on the fixed effects and assumed distribution of the random effects.

We can use `predict` to obtain estimates of the random effects u_j and v_j and combine these with our estimates of β_0 and β_5 to obtain the intercepts and slopes of the linear log trends.

```
. predict re_visit re_cons, reffects

. generate b1 = _b[visit] + re_visit

. generate b0 = _b[_cons] + re_cons

. by subject, sort: generate tolist = _n==1
```

```
. list subject treat b1 b0 if tolist & (subject <=5 | subject >=55)

     +----------------------------------------+
     | subject   treat          b1         b0 |
     |----------------------------------------|
  1. |       1       0   -.4284563   2.164691 |
  5. |       2       0   -.2727145   2.179111 |
  9. |       3       0    .0026486   2.450811 |
 13. |       4       0   -.3194157   2.268827 |
 17. |       5       0    .6063656   2.123723 |
     |----------------------------------------|
217. |      55       1   -.2304782   2.311493 |
221. |      56       1    .2904741   3.211369 |
225. |      57       1   -.4831492   1.457485 |
229. |      58       1    -.252236   1.168154 |
233. |      59       1   -.1266651   2.204869 |
     +----------------------------------------+
```

We list these slopes (b1) and intercepts (b0) for five control subjects and five subjects on the treatment.

```
. count if tolist & treat
  31
. count if tolist & treat & b1 < 0
  25
. count if tolist & !treat
  28
. count if tolist & !treat & b1 < 0
  20
```

We also find that 25 of the 31 subjects taking progabide were estimated to have a downward trend in seizures over their four doctor visits, compared with 20 of the 28 control subjects.

We also obtain predictions for number of seizures, and unless we specify the fixedonly option, these predictions will incorporate the estimated subject-specific random effects.

```
. predict n
(option mu assumed; predicted means)
. list subject treat visit seizures n if subject <= 2 | subject >= 58, sep(0)

     +-----------------------------------------------+
     | subject   treat   visit   seizures          n |
     |-----------------------------------------------|
  1. |       1       0     -.3          5   3.887582 |
  2. |       1       0     -.1          3   3.568324 |
  3. |       1       0      .1          3   3.275285 |
  4. |       1       0      .3          3    3.00631 |
  5. |       2       0     -.3          3   3.705628 |
  6. |       2       0     -.1          5   3.508926 |
  7. |       2       0      .1          3   3.322664 |
  8. |       2       0      .3          3    3.14629 |
229. |      58       1     -.3          0   .9972093 |
230. |      58       1     -.1          0   .9481507 |
231. |      58       1      .1          0   .9015056 |
232. |      58       1      .3          0   .8571553 |
233. |      59       1     -.3          1   2.487858 |
234. |      59       1     -.1          4   2.425625 |
235. |      59       1      .1          3   2.364948 |
236. |      59       1      .3          2   2.305789 |
     +-----------------------------------------------+
```

◁

❑ Technical note

Out-of-sample predictions are permitted after meqrpoisson, but if these predictions involve estimated random effects, the integrity of the estimation data must be preserved. If the estimation data have changed since the model was fit, predict will be unable to obtain predicted random effects that are appropriate for the fitted model and will give an error. Thus to obtain out-of-sample predictions that contain random-effects terms, be sure that the data for these predictions are in observations that augment the estimation data.

❑

Stored results

estat recovariance stores the following in r():

Scalars	
r(relevels)	number of levels
Matrices	
r(Cov#)	level-# random-effects covariance matrix
r(Corr#)	level-# random-effects correlation matrix (if option correlation was specified)

For a G-level nested model, # can be any integer between 2 and G.

Methods and formulas

Continuing the discussion in *Methods and formulas* of [ME] **meqrpoisson** and using the definitions and formulas defined there, we begin by considering the prediction of the random effects $\mathbf{u}_j$ for the jth cluster in a two-level model.

Given a set of estimated meqrpoisson parameters, $(\widehat{\boldsymbol\beta},\widehat{\boldsymbol\Sigma})$, a profile likelihood in $\mathbf{u}_j$ is derived from the joint distribution $f(\mathbf{y}_j,\mathbf{u}_j)$ as

$$\mathcal{L}_j(\mathbf{u}_j) = \exp\{-c(\mathbf{y}_j)\}(2\pi)^{-q/2}|\widehat{\boldsymbol\Sigma}|^{-1/2}\exp\left\{g\left(\widehat{\boldsymbol\beta},\widehat{\boldsymbol\Sigma},\mathbf{u}_j\right)\right\} \tag{1}$$

The conditional maximum likelihood estimator of $\mathbf{u}_j$—conditional on fixed $(\widehat{\boldsymbol\beta},\widehat{\boldsymbol\Sigma})$—is the maximizer of $\mathcal{L}_j(\mathbf{u}_j)$ or, equivalently, the value of $\widehat{\mathbf{u}}_j$ that solves

$$\mathbf{0} = g'\left(\widehat{\boldsymbol\beta},\widehat{\boldsymbol\Sigma},\widehat{\mathbf{u}}_j\right) = \mathbf{Z}_j'\left\{\mathbf{y}_j - \mathbf{m}(\widehat{\boldsymbol\beta},\widehat{\mathbf{u}}_j)\right\} - \widehat{\boldsymbol\Sigma}^{-1}\widehat{\mathbf{u}}_j$$

Because (1) is proportional to the conditional density $f(\mathbf{u}_j|\mathbf{y}_j)$, you can also refer to $\widehat{\mathbf{u}}_j$ as the conditional mode (or posterior mode if you lean toward Bayesian terminology). Regardless, you are referring to the same estimator.

Conditional standard errors for the estimated random effects are derived from standard theory of maximum likelihood, which dictates that the asymptotic variance matrix of $\widehat{\mathbf{u}}_j$ is the negative inverse of the Hessian, which is estimated as

$$g''\left(\widehat{\boldsymbol\beta},\widehat{\boldsymbol\Sigma},\widehat{\mathbf{u}}_j\right) = -\left\{\mathbf{Z}_j'\mathbf{V}(\widehat{\boldsymbol\beta},\widehat{\mathbf{u}}_j)\mathbf{Z}_j + \widehat{\boldsymbol\Sigma}^{-1}\right\}$$

Similar calculations extend to models with more than one level of random effects; see Pinheiro and Chao (2006).

For any observation i in the jth cluster in a two-level model, define the linear predictor as

$$\widehat{\eta}_{ij} = \mathbf{x}_{ij}\widehat{\boldsymbol{\beta}} + \mathbf{z}_{ij}\widehat{\mathbf{u}}_j$$

In a three-level model, for the ith observation within the jth level-two cluster within the kth level-three cluster,

$$\widehat{\eta}_{ijk} = \mathbf{x}_{ijk}\widehat{\boldsymbol{\beta}} + \mathbf{z}^{(3)}_{ijk}\widehat{\mathbf{u}}^{(3)}_k + \mathbf{z}^{(2)}_{ijk}\widehat{\mathbf{u}}^{(2)}_{jk}$$

where $\mathbf{z}^{(p)}$ and $\mathbf{u}^{(p)}$ refer to the level p design variables and random effects, respectively. For models with more than three levels, the definition of $\widehat{\eta}$ extends in the natural way, with only the notation becoming more complicated.

If the `fixedonly` option is specified, $\widehat{\eta}$ contains the linear predictor for only the fixed portion of the model, for example, in a two-level model $\widehat{\eta}_{ij} = \mathbf{x}_{ij}\widehat{\boldsymbol{\beta}}$. In what follows, we assume a two-level model, with the only necessary modification for multilevel models being the indexing.

The predicted mean conditional on the random effects $\widehat{\mathbf{u}}_j$ is

$$\widehat{\mu}_{ij} = \exp(\widehat{\eta}_{ij})$$

Pearson residuals are calculated as

$$\nu^P_{ij} = \frac{y_{ij} - \widehat{\mu}_{ij}}{\{V(\widehat{\mu}_{ij})\}^{1/2}}$$

for $V(\widehat{\mu}_{ij}) = \widehat{\mu}_{ij}$.

Deviance residuals are calculated as

$$\nu^D_{ij} = \text{sign}(y_{ij} - \widehat{\mu}_{ij})\sqrt{\widehat{d}^2_{ij}}$$

where

$$\widehat{d}^2_{ij} = \begin{cases} 2\widehat{\mu}_{ij} & \text{if } y_{ij} = 0 \\ 2\left\{ y_{ij}\log\left(\dfrac{y_{ij}}{\widehat{\mu}_{ij}}\right) - (y_{ij} - \widehat{\mu}_{ij})\right\} & \text{otherwise} \end{cases}$$

Anscombe residuals are calculated as

$$\nu^A_{ij} = \frac{3\left(y^{2/3}_{ij} - \widehat{\mu}^{2/3}_{ij}\right)}{2\widehat{\mu}^{1/6}_{ij}}$$

For a discussion of the general properties of the above residuals, see Hardin and Hilbe (2012, chap. 4).

References

Hardin, J. W., and J. M. Hilbe. 2012. *Generalized Linear Models and Extensions*. 3rd ed. College Station, TX: Stata Press.

McCullagh, P., and J. A. Nelder. 1989. *Generalized Linear Models*. 2nd ed. London: Chapman & Hall/CRC.

Pinheiro, J. C., and E. C. Chao. 2006. Efficient Laplacian and adaptive Gaussian quadrature algorithms for multilevel generalized linear mixed models. *Journal of Computational and Graphical Statistics* 15: 58–81.

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

Also see

[ME] **meqrpoisson** — Multilevel mixed-effects Poisson regression (QR decomposition)

[U] 20 Estimation and postestimation commands

Title

mixed — Multilevel mixed-effects linear regression

Syntax

`mixed` *depvar fe_equation* [`||` *re_equation*] [`||` *re_equation* ...] [, *options*]

where the syntax of *fe_equation* is

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

and the syntax of *re_equation* is one of the following:

for random coefficients and intercepts

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

for random effects among the values of a factor variable

levelvar`:` `R.`*varname* [, *re_options*]

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

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

options	Description
Model	
mle	fit model via maximum likelihood; the default
reml	fit model via restricted maximum likelihood
pwscale(*scale_method*)	control scaling of sampling weights in two-level models
residuals(*rspec*)	structure of residual errors
SE/Robust	
vce(*vcetype*)	*vcetype* may be oim, robust, or cluster *clustvar*
Reporting	
level(#)	set confidence level; default is level(95)
variance	show random-effects and residual-error parameter estimates as variances and covariances; the default
stddeviations	show random-effects and residual-error parameter estimates as standard deviations
noretable	suppress random-effects table
nofetable	suppress fixed-effects table
estmetric	show parameter estimates in the estimation metric
noheader	suppress output header
nogroup	suppress table summarizing groups
nostderr	do not estimate standard errors of random-effects parameters
nolrtest	do not perform likelihood-ratio test comparing with linear regression
display_options	control column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
EM options	
emiterate(#)	number of EM iterations; default is emiterate(20)
emtolerance(#)	EM convergence tolerance; default is emtolerance(1e-10)
emonly	fit model exclusively using EM
emlog	show EM iteration log
emdots	show EM iterations as dots
Maximization	
maximize_options	control the maximization process; seldom used
matsqrt	parameterize variance components using matrix square roots; the default
matlog	parameterize variance components using matrix logarithms
coeflegend	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

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**.

bootstrap, by, jackknife, mi estimate, rolling, and statsby are allowed; see **[U] 11.1.10 Prefix commands**.

Weights are not allowed with the bootstrap prefix; see **[R] bootstrap**.

pweights and fweights are allowed; see **[U] 11.1.6 weight**.

coeflegend does 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 > Linear regression

Description

mixed fits linear mixed-effects models. The overall error distribution of the linear mixed-effects model is assumed to be Gaussian, and heteroskedasticity and correlations within lowest-level groups also may be modeled.

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.

covariance(*vartype*), where *vartype* is

independent | exchangeable | identity | unstructured

specifies the structure of the covariance matrix for the random effects and may be specified for each random-effects equation. An independent covariance structure allows for a distinct variance for each random effect within a random-effects equation and assumes that all covariances are 0. exchangeable structure specifies one common variance for all random effects and one common pairwise covariance. identity is short for "multiple of the identity"; that is, all variances are equal and all covariances are 0. 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(independent) is the default, except when the R. notation is used, in which case covariance(identity) is the default and only covariance(identity) and covariance(exchangeable) are allowed.

`collinear` specifies that `mixed` not omit collinear variables from the random-effects equation. Usually, there is no reason to leave collinear variables in place; in fact, doing so usually causes the estimation to fail because of the matrix singularity caused by the collinearity. However, with certain models (for example, a random-effects model with a full set of contrasts), the variables may be collinear, yet the model is fully identified because of restrictions on the random-effects covariance structure. In such cases, using the `collinear` option allows the estimation to take place with the random-effects equation intact.

`fweight(`*exp*`)` 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*`]`. *exp* can be any valid Stata expression, and you can specify `fweight()` at levels two and higher of a multilevel model. For example, in the two-level model

```
. mixed 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.

`pweight(`*exp*`)` 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*`]`. *exp* can be any valid Stata expression, and you can specify `pweight()` at levels two and higher of a multilevel model. For example, in the two-level model

```
. mixed 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.

See *Survey data* in *Remarks and examples* below for more information regarding the use of sampling weights in multilevel models.

Weighted estimation, whether frequency or sampling, is not supported under restricted maximum-likelihood estimation (REML).

`mle` and `reml` specify the statistical method for fitting the model.

`mle`, the default, specifies that the model be fit using maximum likelihood (ML).

`reml` specifies that the model be fit using restricted maximum likelihood (REML), also known as residual maximum likelihood.

`pwscale(`*scale_method*`)`, where *scale_method* is

`size` | `effective` | `gk`

controls how sampling weights (if specified) are scaled in two-level models.

scale_method `size` specifies that first-level (observation-level) weights be scaled so that they sum to the sample size of their corresponding second-level cluster. Second-level sampling weights are left unchanged.

scale_method `effective` specifies that first-level weights be scaled so that they sum to the effective sample size of their corresponding second-level cluster. Second-level sampling weights are left unchanged.

scale_method `gk` specifies the Graubard and Korn (1996) method. Under this method, second-level weights are set to the cluster averages of the products of the weights at both levels, and first-level weights are then set equal to 1.

`pwscale()` is supported only with two-level models. See *Survey data* in *Remarks and examples* below for more details on using `pwscale()`.

`residuals(`*rspec*`)`, where *rspec* is

restype [, *residual_options*]

specifies the structure of the residual errors within the lowest-level groups (the second level of a multilevel model with the observations comprising the first level) of the linear mixed model. For example, if you are modeling random effects for classes nested within schools, then `residuals()` refers to the residual variance–covariance structure of the observations within classes, the lowest-level groups.

restype is

`independent` | `exchangeable` | `ar` # | `ma` # | `unstructured` |
`banded` # | `toeplitz` # | `exponential`

By default, *restype* is `independent`, which means that all residuals are independent and identically distributed (i.i.d.) Gaussian with one common variance. When combined with `by(`*varname*`)`, independence is still assumed, but you estimate a distinct variance for each level of *varname*. Unlike with the structures described below, *varname* does not need to be constant within groups.

restype `exchangeable` estimates two parameters, one common within-group variance and one common pairwise covariance. When combined with `by(`*varname*`)`, these two parameters are distinctly estimated for each level of *varname*. Because you are modeling a within-group covariance, *varname* must be constant within lowest-level groups.

restype `ar` # assumes that within-group errors have an autoregressive (AR) structure of order #; `ar 1` is the default. The `t(`*varname*`)` option is required, where *varname* is an integer-valued time variable used to order the observations within groups and to determine the lags between successive observations. Any nonconsecutive time values will be treated as gaps. For this structure, # + 1 parameters are estimated (# AR coefficients and one overall error variance). *restype* `ar` may be combined with `by(`*varname*`)`, but *varname* must be constant within groups.

restype `ma` # assumes that within-group errors have a moving average (MA) structure of order #; `ma 1` is the default. The `t(`*varname*`)` option is required, where *varname* is an integer-valued time variable used to order the observations within groups and to determine the lags between successive observations. Any nonconsecutive time values will be treated as gaps. For this structure, # + 1 parameters are estimated (# MA coefficients and one overall error variance). *restype* `ma` may be combined with `by(`*varname*`)`, but *varname* must be constant within groups.

restype `unstructured` is the most general structure; it estimates distinct variances for each within-group error and distinct covariances for each within-group error pair. The `t(`*varname*`)` option is required, where *varname* is a nonnegative-integer–valued variable that identifies the observations within each group. The groups may be unbalanced in that not all levels of `t()` need to be observed within every group, but you may not have repeated `t()` values within any particular group. When you have p levels of `t()`, then $p(p+1)/2$ parameters are estimated. *restype* `unstructured` may be combined with `by(`*varname*`)`, but *varname* must be constant within groups.

restype `banded` # is a special case of `unstructured` that restricts estimation to the covariances within the first # off-diagonals and sets the covariances outside this band to 0. The `t(`*varname*`)` option is required, where *varname* is a nonnegative-integer–valued variable that identifies the observations within each group. # is an integer between 0 and $p-1$, where p is the number of levels of `t()`. By default, # is $p-1$; that is, all elements of the covariance matrix are estimated. When # is 0, only the diagonal elements of the covariance matrix are estimated. *restype* `banded` may be combined with `by(`*varname*`)`, but *varname* must be constant within groups.

restype `toeplitz` # assumes that within-group errors have Toeplitz structure of order #, for which correlations are constant with respect to time lags less than or equal to # and are 0 for lags greater than #. The `t(`*varname*`)` option is required, where *varname* is an integer-valued time variable used to order the observations within groups and to determine the lags between successive observations. # is an integer between 1 and the maximum observed lag (the default). Any nonconsecutive time values will be treated as gaps. For this structure, # + 1 parameters are estimated (# correlations and one overall error variance). *restype* `toeplitz` may be combined with `by(`*varname*`)`, but *varname* must be constant within groups.

restype `exponential` is a generalization of the AR covariance model that allows for unequally spaced and noninteger time values. The `t(`*varname*`)` option is required, where *varname* is real-valued. For the exponential covariance model, the correlation between two errors is the parameter ρ, raised to a power equal to the absolute value of the difference between the `t()` values for those errors. For this structure, two parameters are estimated (the correlation parameter ρ and one overall error variance). *restype* `exponential` may be combined with `by(`*varname*`)`, but *varname* must be constant within groups.

residual_options are `by(`*varname*`)` and `t(`*varname*`)`.

`by(`*varname*`)` is for use within the `residuals()` option and specifies that a set of distinct residual-error parameters be estimated for each level of *varname*. In other words, you use `by()` to model heteroskedasticity.

`t(`*varname*`)` is for use within the `residuals()` option to specify a time variable for the `ar`, `ma`, `toeplitz`, and `exponential` structures, or to identify the observations when *restype* is `unstructured` or `banded`.

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.

`vce(robust)` and `vce(cluster` *clustvar*`)` are not supported with REML estimation.

Reporting

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

`variance`, the default, displays the random-effects and residual-error parameter estimates as variances and covariances.

`stddeviations` displays the random-effects and residual-error parameter estimates as standard deviations and correlations.

`noretable` suppresses the random-effects table from the output.

`nofetable` suppresses the fixed-effects table from the output.

`estmetric` displays all parameter estimates in the estimation metric. Fixed-effects estimates are unchanged from those normally displayed, but random-effects parameter estimates are displayed as log-standard deviations and hyperbolic arctangents of correlations, with equation names that organize them by model level. Residual-variance parameter estimates are also displayed in their original estimation metric.

`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.

`nostderr` prevents `mixed` from calculating standard errors for the estimated random-effects parameters, although standard errors are still provided for the fixed-effects parameters. Specifying this option will speed up computation times. `nostderr` is available only when residuals are modeled as independent with constant variance.

`nolrtest` prevents `mixed` from fitting a reference linear regression model and using this model to calculate a likelihood-ratio test comparing the mixed model to ordinary regression. This option may also be specified on replay to suppress this test from the output.

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

EM options

These options control the expectation-maximization (EM) iterations that take place before estimation switches to a gradient-based method. When residuals are modeled as independent with constant variance, EM will either converge to the solution or bring parameter estimates close to the solution. For other residual structures or for weighted estimation, EM is used to obtain starting values.

`emiterate(#)` specifies the number of EM iterations to perform. The default is `emiterate(20)`.

`emtolerance(#)` specifies the convergence tolerance for the EM algorithm. The default is `emtolerance(1e-10)`. EM iterations will be halted once the log (restricted) likelihood changes by a relative amount less than #. At that point, optimization switches to a gradient-based method, unless `emonly` is specified, in which case maximization stops.

`emonly` specifies that the likelihood be maximized exclusively using EM. The advantage of specifying `emonly` is that EM iterations are typically much faster than those for gradient-based methods. The disadvantages are that EM iterations can be slow to converge (if at all) and that EM provides no facility for estimating standard errors for the random-effects parameters. `emonly` is available only with unweighted estimation and when residuals are modeled as independent with constant variance.

`emlog` specifies that the EM iteration log be shown. The EM iteration log is, by default, not displayed unless the `emonly` option is specified.

`emdots` specifies that the EM iterations be shown as dots. This option can be convenient because the EM algorithm may require many iterations to converge.

Maximization

maximize_options: `difficult`, `technique(`*algorithm_spec*`)`, `iterate(#)`, [`no`]`log`, `trace`, `gradient`, `showstep`, `hessian`, `showtolerance`, `tolerance(#)`, `ltolerance(#)`, `nrtolerance(#)`, and `nonrtolerance`; see [R] **maximize**. Those that require special mention for `mixed` are listed below.

For the `technique()` option, the default is `technique(nr)`. The `bhhh` algorithm may not be specified.

`matsqrt` (the default), during optimization, parameterizes variance components by using the matrix square roots of the variance–covariance matrices formed by these components at each model level.

matlog, during optimization, parameterizes variance components by using the matrix logarithms of the variance–covariance matrices formed by these components at each model level.

The matsqrt parameterization ensures that variance–covariance matrices are positive semidefinite, while matlog ensures matrices that are positive definite. For most problems, the matrix square root is more stable near the boundary of the parameter space. However, if convergence is problematic, one option may be to try the alternate matlog parameterization. When convergence is not an issue, both parameterizations yield equivalent results.

The following option is available with mixed but is not shown in the dialog box:

coeflegend; see [R] **estimation options**.

Remarks and examples

Remarks are presented under the following headings:

Introduction

Linear mixed models are models containing both fixed effects and random effects. They are a generalization of linear regression allowing for the inclusion of random deviations (effects) other than those associated with the overall error term. In matrix notation,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + \boldsymbol{\epsilon} \tag{1}$$

where $\mathbf{y}$ is the $n \times 1$ vector of responses, $\mathbf{X}$ is an $n \times p$ design/covariate matrix for the fixed effects $\boldsymbol{\beta}$, and $\mathbf{Z}$ is the $n \times q$ design/covariate matrix for the random effects $\mathbf{u}$. The $n \times 1$ vector of errors $\boldsymbol{\epsilon}$ is assumed to be multivariate normal with mean 0 and variance matrix $\sigma^2_\epsilon \mathbf{R}$.

The fixed portion of (1), $\mathbf{X}\boldsymbol{\beta}$, is analogous to the linear predictor from a standard OLS regression model with $\boldsymbol{\beta}$ being the regression coefficients to be estimated. For the random portion of (1), $\mathbf{Z}\mathbf{u} + \boldsymbol{\epsilon}$, we assume that $\mathbf{u}$ has variance–covariance matrix $\mathbf{G}$ and that $\mathbf{u}$ is orthogonal to $\boldsymbol{\epsilon}$ so that

$$\text{Var}\begin{bmatrix} \mathbf{u} \\ \boldsymbol{\epsilon} \end{bmatrix} = \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \sigma^2_\epsilon \mathbf{R} \end{bmatrix}$$

The random effects $\mathbf{u}$ are not directly estimated (although they may be predicted), but instead are characterized by the elements of $\mathbf{G}$, known as variance components, that are estimated along with the overall residual variance σ^2_ϵ and the residual-variance parameters that are contained within $\mathbf{R}$.

The general forms of the design matrices $\mathbf{X}$ and $\mathbf{Z}$ allow estimation for a broad class of linear models: blocked designs, split-plot designs, growth curves, multilevel or hierarchical designs, etc. They also allow a flexible method of modeling within-cluster correlation. Subjects within the same cluster can be correlated as a result of a shared random intercept, or through a shared random slope on (say) age, or both. The general specification of $\mathbf{G}$ also provides additional flexibility—the random intercept and random slope could themselves be modeled as independent, or correlated, or independent with equal variances, and so forth. The general structure of $\mathbf{R}$ also allows for residual errors to be heteroskedastic and correlated, and allows flexibility in exactly how these characteristics can be modeled.

Comprehensive treatments of mixed models are provided by, among others, Searle, Casella, and McCulloch (1992); McCulloch, Searle, and Neuhaus (2008); Verbeke and Molenberghs (2000); Raudenbush and Bryk (2002); Demidenko (2004); and Pinheiro and Bates (2000). In particular, chapter 2 of Searle, Casella, and McCulloch (1992) provides an excellent history.

The key to fitting mixed models lies in estimating the variance components, and for that there exist many methods. Most of the early literature in mixed models dealt with estimating variance components in ANOVA models. For simple models with balanced data, estimating variance components amounts to solving a system of equations obtained by setting expected mean-squares expressions equal to their observed counterparts. Much of the work in extending the ANOVA method to unbalanced data for general ANOVA designs is due to Henderson (1953).

The ANOVA method, however, has its shortcomings. Among these is a lack of uniqueness in that alternative, unbiased estimates of variance components could be derived using other quadratic forms of the data in place of observed mean squares (Searle, Casella, and McCulloch 1992, 38–39). As a result, ANOVA methods gave way to more modern methods, such as minimum norm quadratic unbiased estimation (MINQUE) and minimum variance quadratic unbiased estimation (MIVQUE); see Rao (1973) for MINQUE and LaMotte (1973) for MIVQUE. Both methods involve finding optimal quadratic forms of the data that are unbiased for the variance components.

The most popular methods, however, are ML and REML, and these are the two methods that are supported by `mixed`. The ML estimates are based on the usual application of likelihood theory, given the distributional assumptions of the model. The basic idea behind REML (Thompson 1962) is that you can form a set of linear contrasts of the response that do not depend on the fixed effects $\boldsymbol{\beta}$, but instead depend only on the variance components to be estimated. You then apply ML methods by using the distribution of the linear contrasts to form the likelihood.

Returning to (1): in clustered-data situations, it is convenient not to consider all n observations at once but instead to organize the mixed model as a series of M independent groups (or clusters)

$$\mathbf{y}_j = \mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j + \boldsymbol{\epsilon}_j \tag{2}$$

for $j = 1, \ldots, M$, with cluster j consisting of n_j observations. The response $\mathbf{y}_j$ comprises the rows of $\mathbf{y}$ corresponding with the jth cluster, with $\mathbf{X}_j$ and $\boldsymbol{\epsilon}_j$ defined analogously. The random effects $\mathbf{u}_j$ can now be thought of as M realizations of a $q \times 1$ vector that is normally distributed with mean $\mathbf{0}$ and $q \times q$ variance matrix $\boldsymbol{\Sigma}$. The matrix $\mathbf{Z}_i$ is the $n_j \times q$ design matrix for the jth cluster random effects. Relating this to (1), note that

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Z}_M \end{bmatrix}; \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_M \end{bmatrix}; \quad \mathbf{G} = \mathbf{I}_M \otimes \boldsymbol{\Sigma}; \quad \mathbf{R} = \mathbf{I}_M \otimes \boldsymbol{\Lambda} \tag{3}$$

The mixed-model formulation (2) is from Laird and Ware (1982) and offers two key advantages. First, it makes specifications of random-effects terms easier. If the clusters are schools, you can simply specify a random effect at the school level, as opposed to thinking of what a school-level random effect would mean when all the data are considered as a whole (if it helps, think Kronecker products). Second, representing a mixed-model with (2) generalizes easily to more than one set of random effects. For example, if classes are nested within schools, then (2) can be generalized to allow random effects at both the school and the class-within-school levels. This we demonstrate later.

In the sections that follow, we assume that residuals are independent with constant variance; that is, in (3) we treat $\mathbf{\Lambda}$ equal to the identity matrix and limit ourselves to estimating one overall residual variance, σ^2_ϵ. Beginning in *Heteroskedastic residual errors*, we relax this assumption.

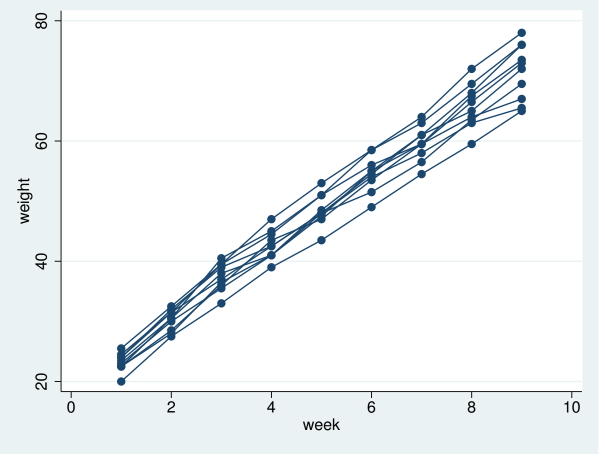
Two-level models

We begin with a simple application of (2) as a two-level model, because a one-level linear model, by our terminology, is just standard OLS regression.

▷ Example 1

Consider a longitudinal dataset, used by both Ruppert, Wand, and Carroll (2003) and Diggle et al. (2002), consisting of `weight` measurements of 48 pigs on 9 successive `weeks`. Pigs are identified by the variable `id`. Below is a plot of the growth curves for the first 10 pigs.

```
. use http://www.stata-press.com/data/r13/pig
(Longitudinal analysis of pig weights)
. twoway connected weight week if id<=10, connect(L)
```



It seems clear that each pig experiences a linear trend in growth and that overall weight measurements vary from pig to pig. Because we are not really interested in these particular 48 pigs per se, we instead treat them as a random sample from a larger population and model the between-pig variability as a random effect, or in the terminology of (2), as a random-intercept term at the pig level. We thus wish to fit the model

$$\texttt{weight}_{ij} = \beta_0 + \beta_1 \texttt{week}_{ij} + u_j + \epsilon_{ij} \tag{4}$$

for $i = 1, \dots, 9$ weeks and $j = 1, \dots, 48$ pigs. The fixed portion of the model, $\beta_0 + \beta_1\texttt{week}_{ij}$, simply states that we want one overall regression line representing the population average. The random effect u_j serves to shift this regression line up or down according to each pig. Because the random effects occur at the pig level (`id`), we fit the model by typing

```
. mixed weight week || id:
Performing EM optimization:
Performing gradient-based optimization:
Iteration 0:   log likelihood = -1014.9268
Iteration 1:   log likelihood = -1014.9268
Computing standard errors:
Mixed-effects ML regression                     Number of obs      =       432
Group variable: id                              Number of groups   =        48
                                                Obs per group: min =         9
                                                               avg =       9.0
                                                               max =         9
                                                Wald chi2(1)       =  25337.49
Log likelihood = -1014.9268                     Prob > chi2        =    0.0000

      weight        Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]

        week     6.209896   .0390124   159.18   0.000     6.133433    6.286359
       _cons     19.35561   .5974059    32.40   0.000     18.18472    20.52651


  Random-effects Parameters     Estimate   Std. Err.     [95% Conf. Interval]

id: Identity
                  var(_cons)    14.81751   3.124226      9.801716    22.40002

               var(Residual)    4.383264   .3163348      3.805112     5.04926

LR test vs. linear regression: chibar2(01) =   472.65 Prob >= chibar2 = 0.0000
```

Notes:

1. By typing `weight week`, we specified the response, `weight`, and the fixed portion of the model in the same way that we would if we were using `regress` or any other estimation command. Our fixed effects are a coefficient on `week` and a constant term.
2. When we added `|| id:`, we specified random effects at the level identified by the group variable `id`, that is, the pig level (level two). Because we wanted only a random intercept, that is all we had to type.
3. The estimation log consists of three parts:
   a. A set of EM iterations used to refine starting values. By default, the iterations themselves are not displayed, but you can display them with the `emlog` option.
   b. A set of gradient-based iterations. By default, these are Newton–Raphson iterations, but other methods are available by specifying the appropriate `maximize` options; see [R] **maximize**.
   c. The message "Computing standard errors". This is just to inform you that `mixed` has finished its iterative maximization and is now reparameterizing from a matrix-based parameterization (see *Methods and formulas*) to the natural metric of variance components and their estimated standard errors.
4. The output title, "Mixed-effects ML regression", informs us that our model was fit using ML, the default. For REML estimates, use the `reml` option.

   Because this model is a simple random-intercept model fit by ML, it would be equivalent to using `xtreg` with its `mle` option.
5. The first estimation table reports the fixed effects. We estimate $\beta_0 = 19.36$ and $\beta_1 = 6.21$.

6. The second estimation table shows the estimated variance components. The first section of the table is labeled `id: Identity`, meaning that these are random effects at the `id` (pig) level and that their variance–covariance matrix is a multiple of the identity matrix; that is, $\boldsymbol{\Sigma} = \sigma_u^2 \mathbf{I}$. Because we have only one random effect at this level, `mixed` knew that `Identity` is the only possible covariance structure. In any case, the variance of the level-two errors, σ_u^2, is estimated as 14.82 with standard error 3.12.
7. The row labeled `var(Residual)` displays the estimated variance of the overall error term; that is, $\widehat{\sigma}_\epsilon^2 = 4.38$. This is the variance of the level-one errors, that is, the residuals.
8. Finally, a likelihood-ratio test comparing the model with one-level ordinary linear regression, model (4) without u_j, is provided and is highly significant for these data.

We now store our estimates for later use:

```
. estimates store randint
```

◁

▷ Example 2

Extending (4) to allow for a random slope on `week` yields the model

$$\texttt{weight}_{ij} = \beta_0 + \beta_1 \texttt{week}_{ij} + u_{0j} + u_{1j}\texttt{week}_{ij} + \epsilon_{ij} \tag{5}$$

and we fit this with `mixed`:

```
. mixed weight week || id: week
Performing EM optimization:
Performing gradient-based optimization:
Iteration 0:   log likelihood = -869.03825
Iteration 1:   log likelihood = -869.03825
Computing standard errors:
Mixed-effects ML regression                     Number of obs     =        432
Group variable: id                              Number of groups  =         48
                                                Obs per group: min =         9
                                                               avg =       9.0
                                                               max =         9
                                                Wald chi2(1)      =    4689.51
Log likelihood = -869.03825                     Prob > chi2       =     0.0000

------------------------------------------------------------------------------
      weight |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        week |   6.209896   .0906819    68.48   0.000     6.032163    6.387629
       _cons |   19.35561   .3979159    48.64   0.000     18.57571    20.13551
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
id: Independent              |
                   var(week) |   .3680668   .0801181      .2402389    .5639103
                  var(_cons) |   6.756364   1.543503      4.317721    10.57235
-----------------------------+------------------------------------------------
               var(Residual) |   1.598811   .1233988      1.374358     1.85992
------------------------------------------------------------------------------
LR test vs. linear regression:       chi2(2) =   764.42   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
. estimates store randslope
```

Because we did not specify a covariance structure for the random effects $(u_{0j}, u_{1j})'$, `mixed` used the default `Independent` structure; that is,

$$\boldsymbol{\Sigma} = \text{Var}\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} = \begin{bmatrix} \sigma^2_{u0} & 0 \\ 0 & \sigma^2_{u1} \end{bmatrix} \tag{6}$$

with $\widehat{\sigma}^2_{u0} = 6.76$ and $\widehat{\sigma}^2_{u1} = 0.37$. Our point estimates of the fixed effects are essentially identical to those from model (4), but note that this does not hold generally. Given the 95% confidence interval for $\widehat{\sigma}^2_{u1}$, it would seem that the random slope is significant, and we can use `lrtest` and our two stored estimation results to verify this fact:

```
. lrtest randslope randint

Likelihood-ratio test                               LR chi2(1)  =    291.78
(Assumption: randint nested in randslope)           Prob > chi2 =    0.0000

Note: The reported degrees of freedom assumes the null hypothesis is not on
      the boundary of the parameter space.  If this is not true, then the
      reported test is conservative.
```

The near-zero significance level favors the model that allows for a random pig-specific regression line over the model that allows only for a pig-specific shift.

◁

Covariance structures

In example 2, we fit a model with the default `Independent` covariance given in (6). Within any random-effects level specification, we can override this default by specifying an alternative covariance structure via the `covariance()` option.

▷ Example 3

We generalize (6) to allow u_{0j} and u_{1j} to be correlated; that is,

$$\boldsymbol{\Sigma} = \text{Var}\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} = \begin{bmatrix} \sigma^2_{u0} & \sigma_{01} \\ \sigma_{01} & \sigma^2_{u1} \end{bmatrix}$$

```
. mixed weight week || id: week, covariance(unstructured)
Performing EM optimization:
Performing gradient-based optimization:
Iteration 0:   log likelihood = -868.96185
Iteration 1:   log likelihood = -868.96185
Computing standard errors:
Mixed-effects ML regression                     Number of obs      =       432
Group variable: id                              Number of groups   =        48
                                                Obs per group: min =         9
                                                               avg =       9.0
                                                               max =         9
                                                Wald chi2(1)       =   4649.17
Log likelihood = -868.96185                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
      weight |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        week |   6.209896   .0910745    68.18   0.000     6.031393    6.388399
       _cons |   19.35561   .3996387    48.43   0.000     18.57234    20.13889
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
id: Unstructured             |
                  var(week)  |   .3715251   .0812958      .2419532     .570486
                 var(_cons)  |   6.823363   1.566194      4.351297    10.69986
           cov(week,_cons)   |  -.0984378   .2545767     -.5973991    .4005234
-----------------------------+------------------------------------------------
              var(Residual)  |   1.596829    .123198      1.372735    1.857505
------------------------------------------------------------------------------
LR test vs. linear regression:       chi2(3) =   764.58   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

But we do not find the correlation to be at all significant.

```
. lrtest . randslope
Likelihood-ratio test                                 LR chi2(1)  =      0.15
(Assumption: randslope nested in .)                   Prob > chi2 =    0.6959
```

◁

Instead, we could have also specified `covariance(identity)`, restricting u_{0j} and u_{1j} to not only be independent but also to have common variance, or we could have specified `covariance(exchangeable)`, which imposes a common variance but allows for a nonzero correlation.

Likelihood versus restricted likelihood

Thus far, all our examples have used ML to estimate variance components. We could have just as easily asked for REML estimates. Refitting the model in example 2 by REML, we get

```
. mixed weight week || id: week, reml
Performing EM optimization:
Performing gradient-based optimization:
Iteration 0:   log restricted-likelihood = -870.51473
Iteration 1:   log restricted-likelihood = -870.51473
Computing standard errors:
Mixed-effects REML regression                   Number of obs     =        432
Group variable: id                              Number of groups  =         48

                                                Obs per group: min =         9
                                                               avg =       9.0
                                                               max =         9

                                                Wald chi2(1)      =    4592.10
Log restricted-likelihood = -870.51473          Prob > chi2       =     0.0000

------------------------------------------------------------------------------
      weight |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        week |   6.209896   .0916387    67.77   0.000     6.030287    6.389504
       _cons |   19.35561   .4021144    48.13   0.000     18.56748    20.14374
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
id: Independent              |
                   var(week) |   .3764405   .0827027      .2447317    .5790317
                  var(_cons) |   6.917604   1.593247      4.404624    10.86432
-----------------------------+------------------------------------------------
               var(Residual) |   1.598784   .1234011      1.374328    1.859898
------------------------------------------------------------------------------
LR test vs. linear regression:       chi2(2) =   765.92   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

Although ML estimators are based on the usual likelihood theory, the idea behind REML is to transform the response into a set of linear contrasts whose distribution is free of the fixed effects $\boldsymbol{\beta}$. The restricted likelihood is then formed by considering the distribution of the linear contrasts. Not only does this make the maximization problem free of $\boldsymbol{\beta}$, it also incorporates the degrees of freedom used to estimate $\boldsymbol{\beta}$ into the estimation of the variance components. This follows because, by necessity, the rank of the linear contrasts must be less than the number of observations.

As a simple example, consider a constant-only regression where $y_i \sim N(\mu, \sigma^2)$ for $i = 1, \ldots, n$. The ML estimate of σ^2 can be derived theoretically as the n-divided sample variance. The REML estimate can be derived by considering the first $n - 1$ error contrasts, $y_i - \overline{y}$, whose joint distribution is free of μ. Applying maximum likelihood to this distribution results in an estimate of σ^2, that is, the $(n - 1)$-divided sample variance, which is unbiased for σ^2.

The unbiasedness property of REML extends to all mixed models when the data are balanced, and thus REML would seem the clear choice in balanced-data problems, although in large samples the difference between ML and REML is negligible. One disadvantage of REML is that likelihood-ratio (LR) tests based on REML are inappropriate for comparing models with different fixed-effects specifications. ML is appropriate for such LR tests and has the advantage of being easy to explain and being the method of choice for other estimators.

Another factor to consider is that ML estimation under `mixed` is more feature-rich, allowing for weighted estimation and robust variance–covariance matrices, features not supported under REML. In the end, which method to use should be based both on your needs and on personal taste.

Examining the REML output, we find that the estimates of the variance components are slightly larger than the ML estimates. This is typical, because ML estimates, which do not incorporate the degrees of freedom used to estimate the fixed effects, tend to be biased downward.

Three-level models

The clustered-data representation of the mixed model given in (2) can be extended to two nested levels of clustering, creating a three-level model once the observations are considered. Formally,

$$\mathbf{y}_{jk} = \mathbf{X}_{jk}\boldsymbol{\beta} + \mathbf{Z}^{(3)}_{jk}\mathbf{u}^{(3)}_{k} + \mathbf{Z}^{(2)}_{jk}\mathbf{u}^{(2)}_{jk} + \boldsymbol{\epsilon}_{jk} \tag{7}$$

for $i = 1, \dots, n_{jk}$ first-level observations nested within $j = 1, \dots, M_k$ second-level groups, which are nested within $k = 1, \dots, M$ third-level groups. Group j, k consists of n_{jk} observations, so $\mathbf{y}_{jk}$, $\mathbf{X}_{jk}$, and $\boldsymbol{\epsilon}_{jk}$ each have row dimension n_{jk}. $\mathbf{Z}^{(3)}_{jk}$ is the $n_{jk} \times q_3$ design matrix for the third-level random effects $\mathbf{u}^{(3)}_{k}$, and $\mathbf{Z}^{(2)}_{jk}$ is the $n_{jk} \times q_2$ design matrix for the second-level random effects $\mathbf{u}^{(2)}_{jk}$. Furthermore, assume that

$$\mathbf{u}^{(3)}_{k} \sim N(\mathbf{0}, \boldsymbol{\Sigma}_3); \quad \mathbf{u}^{(2)}_{jk} \sim N(\mathbf{0}, \boldsymbol{\Sigma}_2); \quad \boldsymbol{\epsilon}_{jk} \sim N(\mathbf{0}, \sigma^2_\epsilon \mathbf{I})$$

and that $\mathbf{u}^{(3)}_{k}$, $\mathbf{u}^{(2)}_{jk}$, and $\boldsymbol{\epsilon}_{jk}$ are independent.

Fitting a three-level model requires you to specify two random-effects equations: one for level three and then one for level two. The variable list for the first equation represents $\mathbf{Z}^{(3)}_{jk}$ and for the second equation represents $\mathbf{Z}^{(2)}_{jk}$; that is, you specify the levels top to bottom in `mixed`.

▷ Example 4

Baltagi, Song, and Jung (2001) estimate a Cobb–Douglas production function examining the productivity of public capital in each state's private output. Originally provided by Munnell (1990), the data were recorded over 1970–1986 for 48 states grouped into nine regions.

```
. use http://www.stata-press.com/data/r13/productivity
(Public Capital Productivity)

. describe

Contains data from http://www.stata-press.com/data/r13/productivity.dta
  obs:           816                          Public Capital Productivity
 vars:            11                          29 Mar 2013 10:57
 size:        29,376                          (_dta has notes)
--------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
--------------------------------------------------------------------------------
state           byte    %9.0g                 states 1-48
region          byte    %9.0g                 regions 1-9
year            int     %9.0g                 years 1970-1986
public          float   %9.0g                 public capital stock
hwy             float   %9.0g                 log(highway component of public)
water           float   %9.0g                 log(water component of public)
other           float   %9.0g                 log(bldg/other component of
                                                public)
private         float   %9.0g                 log(private capital stock)
gsp             float   %9.0g                 log(gross state product)
emp             float   %9.0g                 log(non-agriculture payrolls)
unemp           float   %9.0g                 state unemployment rate
--------------------------------------------------------------------------------
Sorted by:
```

Because the states are nested within regions, we fit a three-level mixed model with random intercepts at both the region and the state-within-region levels. That is, we use (7) with both $\mathbf{Z}^{(3)}_{jk}$ and $\mathbf{Z}^{(2)}_{jk}$ set to the $n_{jk} \times 1$ column of ones, and $\boldsymbol{\Sigma}_3 = \sigma^2_3$ and $\boldsymbol{\Sigma}_2 = \sigma^2_2$ are both scalars.

```
. mixed gsp private emp hwy water other unemp || region: || state:
  (output omitted)
Mixed-effects ML regression                     Number of obs      =       816

-----------------------------------------------------------
                |   No. of       Observations per Group
 Group Variable |   Groups    Minimum    Average    Maximum
----------------+------------------------------------------
         region |        9         51       90.7        136
          state |       48         17       17.0         17
-----------------------------------------------------------

                                                Wald chi2(6)       =  18829.06
Log likelihood =  1430.5017                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
         gsp |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     private |   .2671484   .0212591    12.57   0.000     .2254814    .3088154
         emp |    .754072   .0261868    28.80   0.000     .7027468    .8053973
         hwy |   .0709767    .023041     3.08   0.002     .0258172    .1161363
       water |   .0761187   .0139248     5.47   0.000     .0488266    .1034109
       other |  -.0999955   .0169366    -5.90   0.000    -.1331906   -.0668004
       unemp |  -.0058983   .0009031    -6.53   0.000    -.0076684   -.0041282
       _cons |   2.128823   .1543854    13.79   0.000     1.826233    2.431413
------------------------------------------------------------------------------
```

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
region: Identity				
var(_cons)	.0014506	.0012995	.0002506	.0083957
state: Identity				
var(_cons)	.0062757	.0014871	.0039442	.0099855
var(Residual)	.0013461	.0000689	.0012176	.0014882

```
LR test vs. linear regression:        chi2(2) =   1154.73   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

Notes:

1. Our model now has two random-effects equations, separated by `||`. The first is a random intercept (constant only) at the `region` level (level three), and the second is a random intercept at the `state` level (level two). The order in which these are specified (from left to right) is significant—`mixed` assumes that `state` is nested within `region`.
2. The information on groups is now displayed as a table, with one row for each grouping. You can suppress this table with the `nogroup` or the `noheader` option, which will suppress the rest of the header, as well.
3. The variance-component estimates are now organized and labeled according to level.

After adjusting for the nested-level error structure, we find that the highway and water components of public capital had significant positive effects on private output, whereas the other public buildings component had a negative effect.

◁

❑ Technical note

In the previous example, the states are coded 1–48 and are nested within nine regions. `mixed` treated the states as nested within regions, regardless of whether the codes for each state were unique between regions. That is, even if codes for states were duplicated between regions, `mixed` would have enforced the nesting and produced the same results.

The group information at the top of the `mixed` output and that produced by the postestimation command `estat group` (see [ME] **mixed postestimation**) take the nesting into account. The statistics are thus not necessarily what you would get if you instead `tabulate`d each group variable individually.

❑

Model (7) extends in a straightforward manner to more than three levels, as does the specification of such models in `mixed`.

Blocked-diagonal covariance structures

Covariance matrices of random effects within an equation can be modeled either as a multiple of the identity matrix, as diagonal (that is, `Independent`), as exchangeable, or as general symmetric (`Unstructured`). These may also be combined to produce more complex block-diagonal covariance structures, effectively placing constraints on the variance components.

▷ Example 5

Returning to our productivity data, we now add random coefficients on hwy and unemp at the region level. This only slightly changes the estimates of the fixed effects, so we focus our attention on the variance components:

```
. mixed gsp private emp hwy water other unemp || region: hwy unemp || state:,
> nolog nogroup nofetable
Mixed-effects ML regression                     Number of obs     =        816
                                                Wald chi2(6)      =   17137.94
Log likelihood =  1447.6787                     Prob > chi2       =     0.0000
```

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
region: Independent				
var(hwy)	.0000209	.0001103	6.71e-10	.650695
var(unemp)	.0000238	.0000135	7.84e-06	.0000722
var(_cons)	.0030349	.0086684	.0000112	.8191296
state: Identity				
var(_cons)	.0063658	.0015611	.0039365	.0102943
var(Residual)	.0012469	.0000643	.001127	.0013795

```
LR test vs. linear regression:       chi2(4) =  1189.08   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
. estimates store prodrc
```

This model is the same as that fit in example 4 except that $\mathbf{Z}^{(3)}_{jk}$ is now the $n_{jk} \times 3$ matrix with columns determined by the values of hwy, unemp, and an intercept term (one), in that order, and (because we used the default Independent structure) $\mathbf{\Sigma}_3$ is

$$\mathbf{\Sigma}_3 = \begin{array}{c} \begin{matrix} \texttt{hwy} & \texttt{unemp} & \texttt{\_cons} \end{matrix} \\ \begin{pmatrix} \sigma^2_a & 0 & 0 \\ 0 & \sigma^2_b & 0 \\ 0 & 0 & \sigma^2_c \end{pmatrix} \end{array}$$

The random-effects specification at the state level remains unchanged; that is, $\mathbf{\Sigma}_2$ is still treated as the scalar variance of the random intercepts at the state level.

An LR test comparing this model with that from example 4 favors the inclusion of the two random coefficients, a fact we leave to the interested reader to verify.

The estimated variance components, upon examination, reveal that the variances of the random coefficients on hwy and unemp could be treated as equal. That is,

$$\mathbf{\Sigma}_3 = \begin{array}{c} \begin{matrix} \texttt{hwy} & \texttt{unemp} & \texttt{\_cons} \end{matrix} \\ \begin{pmatrix} \sigma^2_a & 0 & 0 \\ 0 & \sigma^2_a & 0 \\ 0 & 0 & \sigma^2_c \end{pmatrix} \end{array}$$

looks plausible. We can impose this equality constraint by treating $\mathbf{\Sigma}_3$ as block diagonal: the first block is a 2×2 multiple of the identity matrix, that is, $\sigma^2_a\mathbf{I}_2$; the second is a scalar, equivalently, a 1×1 multiple of the identity.

We construct block-diagonal covariances by repeating level specifications:

```
. mixed gsp private emp hwy water other unemp || region: hwy unemp,
> cov(identity) || region: || state:, nolog nogroup nofetable
Mixed-effects ML regression                     Number of obs      =       816
                                                Wald chi2(6)       =  17136.65
Log likelihood =  1447.6784                     Prob > chi2        =    0.0000

  Random-effects Parameters    Estimate   Std. Err.     [95% Conf. Interval]

region: Identity
              var(hwy unemp)   .0000238   .0000134      7.89e-06    .0000719

region: Identity
                  var(_cons)   .0028191   .0030429      .0003399     .023383

state: Identity
                  var(_cons)    .006358   .0015309      .0039661    .0101925

               var(Residual)   .0012469   .0000643       .001127    .0013795

LR test vs. linear regression:       chi2(3) =  1189.08   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

We specified two equations for the `region` level: the first for the random coefficients on `hwy` and `unemp` with covariance set to `Identity` and the second for the random intercept `_cons`, whose covariance defaults to `Identity` because it is of dimension 1. `mixed` labeled the estimate of σ_a^2 as `var(hwy unemp)` to designate that it is common to the random coefficients on both `hwy` and `unemp`.

An LR test shows that the constrained model fits equally well.

```
. lrtest . prodrc
Likelihood-ratio test                                 LR chi2(1)  =      0.00
(Assumption: . nested in prodrc)                      Prob > chi2 =    0.9784
Note: The reported degrees of freedom assumes the null hypothesis is not on
      the boundary of the parameter space.  If this is not true, then the
      reported test is conservative.
```

◁

Because the null hypothesis for this test is one of equality ($H_0 : \sigma_a^2 = \sigma_b^2$), it is not on the boundary of the parameter space. As such, we can take the reported significance as precise rather than a conservative estimate.

You can repeat level specifications as often as you like, defining successive blocks of a block-diagonal covariance matrix. However, repeated-level equations must be listed consecutively; otherwise, `mixed` will give an error.

❑ Technical note

In the previous estimation output, there was no constant term included in the first `region` equation, even though we did not use the `noconstant` option. When you specify repeated-level equations, `mixed` knows not to put constant terms in each equation because such a model would be unidentified. By default, it places the constant in the last repeated-level equation, but you can use `noconstant` creatively to override this.

❑

Linear mixed-effects models can also be fit using `meglm` with the default `gaussian` family. `meglm` provides two more covariance structures through which you can impose constraints on variance components; see [ME] **meglm** for details.

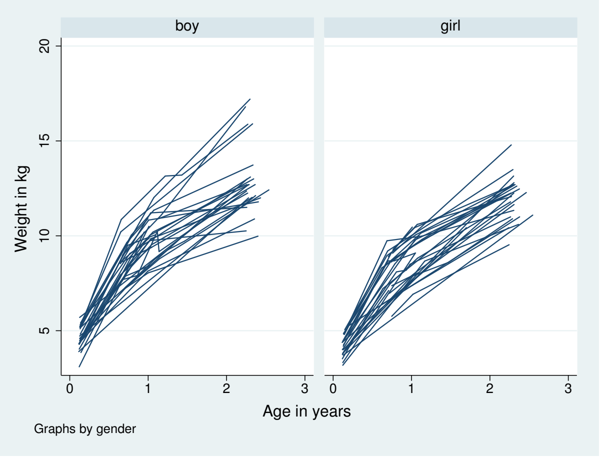
Heteroskedastic random effects

Blocked-diagonal covariance structures and repeated-level specifications of random effects can also be used to model heteroskedasticity among random effects at a given level.

▷ Example 6

Following Rabe-Hesketh and Skrondal (2012, sec. 7.2), we analyze data from Asian children in a British community who were weighed up to four times, roughly between the ages of 6 weeks and 27 months. The dataset is a random sample of data previously analyzed by Goldstein (1986) and Prosser, Rasbash, and Goldstein (1991).

```
. use http://www.stata-press.com/data/r13/childweight
(Weight data on Asian children)

. describe

Contains data from http://www.stata-press.com/data/r13/childweight.dta
  obs:           198                          Weight data on Asian children
 vars:             5                          23 May 2013 15:12
 size:         3,168                          (_dta has notes)
--------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
--------------------------------------------------------------------------------
id              int     %8.0g                 child identifier
age             float   %8.0g                 age in years
weight          float   %8.0g                 weight in Kg
brthwt          int     %8.0g                 Birth weight in g
girl            float   %9.0g      bg         gender
--------------------------------------------------------------------------------
Sorted by:  id  age

. graph twoway (line weight age, connect(ascending)), by(girl)
> xtitle(Age in years) ytitle(Weight in kg)
```



Ignoring gender effects for the moment, we begin with the following model for the ith measurement on the jth child:

$$\texttt{weight}_{ij} = \beta_0 + \beta_1 \texttt{age}_{ij} + \beta_2 \texttt{age}^2_{ij} + u_{j0} + u_{j1} \texttt{age}_{ij} + \epsilon_{ij}$$

This models overall mean growth as quadratic in age and allows for two child-specific random effects: a random intercept u_{j0}, which represents each child's vertical shift from the overall mean (β_0), and a random age slope u_{j1}, which represents each child's deviation in linear growth rate from the overall mean linear growth rate (β_1). For simplicity, we do not consider child-specific changes in the quadratic component of growth.

```
. mixed weight age c.age#c.age || id: age, nolog
Mixed-effects ML regression                     Number of obs      =       198
Group variable: id                              Number of groups   =        68

                                                Obs per group: min =         1
                                                               avg =       2.9
                                                               max =         5

                                                Wald chi2(2)       =   1863.46
Log likelihood = -258.51915                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
      weight |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   7.693701   .2381076    32.31   0.000     7.227019    8.160384
 c.age#c.age |  -1.654542   .0874987   -18.91   0.000    -1.826037   -1.483048
       _cons |   3.497628   .1416914    24.68   0.000     3.219918    3.775338
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
id: Independent              |
                    var(age) |   .2987207   .0827569      .1735603    .5141388
                  var(_cons) |   .5023857    .141263      .2895294    .8717297
-----------------------------+------------------------------------------------
               var(Residual) |   .3092897   .0474887      .2289133     .417888
------------------------------------------------------------------------------
LR test vs. linear regression:       chi2(2) =   114.70   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

◁

Because there is no reason to believe that the random effects are uncorrelated, it is always a good idea to first fit a model with the `covariance(unstructured)` option. We do not include the output for such a model because for these data the correlation between random effects is not significant; however, we did check this before reverting to `mixed`'s default `Independent` structure.

Next we introduce gender effects into the fixed portion of the model by including a main gender effect and a gender–age interaction for overall mean growth:

```
. mixed weight i.girl i.girl#c.age c.age#c.age || id: age, nolog
Mixed-effects ML regression                     Number of obs     =        198
Group variable: id                              Number of groups  =         68

                                                Obs per group: min =          1
                                                               avg =        2.9
                                                               max =          5

                                                Wald chi2(4)      =    1942.30
Log likelihood =   -253.182                     Prob > chi2       =     0.0000

------------------------------------------------------------------------------
      weight |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        girl |
       girl  |  -.5104676   .2145529    -2.38   0.017    -.9309835   -.0899516
             |
  girl#c.age |
        boy  |   7.806765   .2524583    30.92   0.000     7.311956    8.301574
       girl  |   7.577296   .2531318    29.93   0.000     7.081166    8.073425
             |
 c.age#c.age |  -1.654323   .0871752   -18.98   0.000    -1.825183   -1.483463
             |
       _cons |   3.754275   .1726404    21.75   0.000     3.415906    4.092644
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
id: Independent              |
                    var(age) |   .2772846   .0769233      .1609861    .4775987
                  var(_cons) |   .4076892     .12386      .2247635    .7394906
-----------------------------+------------------------------------------------
               var(Residual) |   .3131704    .047684      .2323672     .422072
------------------------------------------------------------------------------
LR test vs. linear regression:       chi2(2) =   104.39   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
. estimates store homoskedastic
```

The main gender effect is significant at the 5% level, but the gender–age interaction is not:

```
. test 0.girl#c.age = 1.girl#c.age
 ( 1)  [weight]0b.girl#c.age - [weight]1.girl#c.age = 0
           chi2(  1) =    1.66
         Prob > chi2 =    0.1978
```

On average, boys are heavier than girls, but their average linear growth rates are not significantly different.

In the above model, we introduced a gender effect on average growth, but we still assumed that the variability in child-specific deviations from this average was the same for boys and girls. To check this assumption, we introduce gender into the random component of the model. Because support for factor-variable notation is limited in specifications of random effects (see *Crossed-effects models* below), we need to generate the interactions ourselves.

```
. gen boy = !girl
. gen boyXage = boy*age
. gen girlXage = girl*age
. mixed weight i.girl i.girl#c.age c.age#c.age || id: boy boyXage, noconstant
> || id: girl girlXage, noconstant nolog nofetable
Mixed-effects ML regression                     Number of obs     =        198
Group variable: id                              Number of groups  =         68
                                                Obs per group: min =         1
                                                               avg =       2.9
                                                               max =         5
                                                Wald chi2(4)      =    2358.11
Log likelihood = -248.94752                     Prob > chi2       =     0.0000

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
id: Independent              |
                    var(boy) |   .3161091   .1557911      .1203181    .8305061
                var(boyXage) |   .4734482   .1574626      .2467028    .9085962
-----------------------------+------------------------------------------------
id: Independent              |
                   var(girl) |   .5798676   .1959725      .2989896    1.124609
               var(girlXage) |   .0664634   .0553274      .0130017    .3397538
-----------------------------+------------------------------------------------
               var(Residual) |   .3078826    .046484      .2290188    .4139037
------------------------------------------------------------------------------
LR test vs. linear regression:       chi2(4) =    112.86   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
. estimates store heteroskedastic
```

In the above, we suppress displaying the fixed portion of the model (the `nofetable` option) because it does not differ much from that of the previous model.

Our previous model had the random-effects specification

```
|| id: age
```

which we have replaced with the dual repeated-level specification

```
|| id: boy boyXage, noconstant || id: girl girlXage, noconstant
```

The former models a random intercept and random slope on age, and does so treating all children as a random sample from one population. The latter also specifies a random intercept and random slope on age, but allows for the variability of the random intercepts and slopes to differ between boys and girls. In other words, it allows for heteroskedasticity in random effects due to gender. We use the `noconstant` option so that we can separate the overall random intercept (automatically provided by the former syntax) into one specific to boys and one specific to girls.

There seems to be a large gender effect in the variability of linear growth rates. We can compare both models with an LR test, recalling that we stored the previous estimation results under the name `homoskedastic`:

```
. lrtest homoskedastic heteroskedastic
Likelihood-ratio test                                 LR chi2(2)  =      8.47
(Assumption: homoskedastic nested in heteroskedas~c)  Prob > chi2 =    0.0145
Note: The reported degrees of freedom assumes the null hypothesis is not on
      the boundary of the parameter space.  If this is not true, then the
      reported test is conservative.
```

Because the null hypothesis here is one of equality of variances and not that variances are 0, the above does not test on the boundary; thus we can treat the significance level as precise and not conservative. Either way, the results favor the new model with heteroskedastic random effects.

Heteroskedastic residual errors

Up to this point, we have assumed that the level-one residual errors—the ϵ's in the stated models—have been i.i.d. Gaussian with variance σ_ϵ^2. This is demonstrated in `mixed` output in the random-effects table, where up until now we have estimated a single residual-error variance, labeled as `var(Residual)`.

To relax the assumptions of homoskedasticity or independence of residual errors, use the `residuals()` option.

▷ Example 7

West, Welch, and Galecki (2007, chap. 7) analyze data studying the effect of ceramic dental veneer placement on gingival (gum) health. Data on 55 teeth located in the maxillary arches of 12 patients were considered.

```
. use http://www.stata-press.com/data/r13/veneer, clear
(Dental veneer data)

. describe

Contains data from http://www.stata-press.com/data/r13/veneer.dta
  obs:           110                          Dental veneer data
 vars:             7                          24 May 2013 12:11
 size:         1,100                          (_dta has notes)
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
patient         byte    %8.0g                 Patient ID
tooth           byte    %8.0g                 Tooth number with patient
gcf             byte    %8.0g                 Gingival crevicular fluid (GCF)
age             byte    %8.0g                 Patient age
base_gcf        byte    %8.0g                 Baseline GCF
cda             float   %9.0g                 Average contour difference after
                                                veneer placement
followup        byte    %9.0g      t          Follow-up time: 3 or 6 months
-------------------------------------------------------------------------------
Sorted by:
```

Veneers were placed to match the original contour of the tooth as closely as possible, and researchers were interested in how contour differences (variable `cda`) impacted gingival health. Gingival health was measured as the amount of gingival crevical fluid (GCF) at each tooth, measured at baseline (variable `base_gcf`) and at two posttreatment follow-ups at 3 and 6 months. The variable `gcf` records GCF at follow-up, and the variable `followup` records the follow-up time.

Because two measurements were taken for each tooth and there exist multiple teeth per patient, we fit a three-level model with the following random effects: a random intercept and random slope on follow-up time at the patient level, and a random intercept at the tooth level. For the ith measurement of the jth tooth from the kth patient, we have

$$\texttt{gcf}_{ijk} = \beta_0 + \beta_1\texttt{followup}_{ijk} + \beta_2\texttt{base\_gcf}_{ijk} + \beta_3\texttt{cda}_{ijk} + \beta_4\texttt{age}_{ijk} + \\ u_{0k} + u_{1k}\texttt{followup}_{ijk} + v_{0jk} + \epsilon_{ijk}$$

which we can fit using mixed:

```
. mixed gcf followup base_gcf cda age || patient: followup, cov(un) || tooth:,
> reml nolog
Mixed-effects REML regression                   Number of obs      =       110

---------------------------------------------------------------
                |   No. of       Observations per Group
 Group Variable |   Groups    Minimum    Average    Maximum
----------------+----------------------------------------------
        patient |       12          2        9.2         12
          tooth |       55          2        2.0          2
---------------------------------------------------------------

                                                Wald chi2(4)       =      7.48
Log restricted-likelihood = -420.92761          Prob > chi2        =    0.1128

------------------------------------------------------------------------------
         gcf |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    followup |   .3009815   1.936863     0.16   0.877      -3.4952    4.097163
    base_gcf |  -.0183127   .1433094    -0.13   0.898     -.299194    .2625685
         cda |   -.329303   .5292525    -0.62   0.534    -1.366619    .7080128
         age |  -.5773932   .2139656    -2.70   0.007    -.9967582   -.1580283
       _cons |   45.73862   12.55497     3.64   0.000     21.13133    70.34591
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
patient: Unstructured        |
               var(followup) |   41.88772   18.79997      17.38009    100.9535
                  var(_cons) |   524.9851   253.0205      204.1287    1350.175
         cov(followup,_cons) |  -140.4229   66.57623     -270.9099   -9.935908
-----------------------------+------------------------------------------------
tooth: Identity              |
                  var(_cons) |   47.45738   16.63034       23.8792     94.3165
-----------------------------+------------------------------------------------
               var(Residual) |   48.86704   10.50523      32.06479    74.47382
------------------------------------------------------------------------------
LR test vs. linear regression:       chi2(4) =    91.12   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

We used REML estimation for no other reason than variety.

Among the other features of the model fit, we note that the residual variance σ_ϵ^2 was estimated as 48.87 and that our model assumed that the residuals were independent with constant variance (homoskedastic). Because it may be the case that the precision of gcf measurements could change over time, we modify the above to estimate two distinct error variances: one for the 3-month follow-up and one for the 6-month follow-up.

To fit this model, we add the residuals(independent, by(followup)) option, which maintains independence of residual errors but allows for heteroskedasticity with respect to follow-up time.

```
. mixed gcf followup base_gcf cda age || patient: followup, cov(un) || tooth:,
> residuals(independent, by(followup)) reml nolog
Mixed-effects REML regression                   Number of obs      =       110

-----------------------------------------------------------
                |   No. of       Observations per Group
 Group Variable |   Groups    Minimum    Average    Maximum
----------------+------------------------------------------
        patient |       12          2        9.2         12
          tooth |       55          2        2.0          2
-----------------------------------------------------------

                                                Wald chi2(4)       =      7.51
Log restricted-likelihood =  -420.4576          Prob > chi2        =    0.1113

------------------------------------------------------------------------------
         gcf |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    followup |   .2703944   1.933096     0.14   0.889    -3.518405    4.059193
    base_gcf |   .0062144   .1419121     0.04   0.965    -.2719283     .284357
         cda |  -.2947235   .5245126    -0.56   0.574    -1.322749    .7333023
         age |  -.5743755   .2142249    -2.68   0.007    -.9942487   -.1545024
       _cons |   45.15089   12.51452     3.61   0.000     20.62288     69.6789
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
patient: Unstructured        |
              var(followup) |   41.75169   18.72989      17.33099     100.583
                  var(_cons) |   515.2018   251.9661      197.5542    1343.596
         cov(followup,_cons) |  -139.0496   66.27806     -268.9522    -9.14694
-----------------------------+------------------------------------------------
tooth: Identity              |
                  var(_cons) |   47.35914   16.48931      23.93514    93.70693
-----------------------------+------------------------------------------------
Residual: Independent,       |
    by followup              |
            3 months: var(e) |   61.36785   18.38913      34.10946    110.4096
            6 months: var(e) |   36.42861   14.97501      16.27542    81.53666
------------------------------------------------------------------------------
LR test vs. linear regression:       chi2(5) =    92.06   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

Comparison of both models via an LR test reveals the difference in residual variances to be not significant, something we leave to you to verify as an exercise.

◁

The default residual-variance structure is `independent`, and when specified without `by()` is equivalent to the default behavior of `mixed`: estimating one overall residual standard variance for the entire model.

Other residual-error structures

Besides the default `independent` residual-error structure, `mixed` supports four other structures that allow for correlation between residual errors within the lowest-level (smallest or level two) groups. For purposes of notation, in what follows we assume a two-level model, with the obvious extension to higher-level models.

The `exchangeable` structure assumes one overall variance and one common pairwise covariance; that is,

$$\text{Var}(\boldsymbol{\epsilon}_j) = \text{Var}\begin{bmatrix} \epsilon_{j1} \\ \epsilon_{j2} \\ \vdots \\ \epsilon_{jn_j} \end{bmatrix} = \begin{bmatrix} \sigma_\epsilon^2 & \sigma_1 & \cdots & \sigma_1 \\ \sigma_1 & \sigma_\epsilon^2 & \cdots & \sigma_1 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_1 & \sigma_1 & \sigma_1 & \sigma_\epsilon^2 \end{bmatrix}$$

By default, `mixed` will report estimates of the two parameters as estimates of the common variance σ_ϵ^2 and of the covariance σ_1. If the `stddeviations` option is specified, you obtain estimates of σ_ϵ and the pairwise correlation. When the `by(`*varname*`)` option is also specified, these two parameters are estimated for each level *varname*.

The `ar` p structure assumes that the errors have an AR structure of order p. That is,

$$\epsilon_{ij} = \phi_1 \epsilon_{i-1,j} + \cdots + \phi_p \epsilon_{i-p,j} + u_{ij}$$

where u_{ij} are i.i.d. Gaussian with mean 0 and variance σ_u^2. `mixed` reports estimates of $\phi_1, \ldots, \phi_p$ and the overall error variance σ_ϵ^2, which can be derived from the above expression. The `t(`*varname*`)` option is required, where *varname* is a time variable used to order the observations within lowest-level groups and to determine any gaps between observations. When the `by(`*varname*`)` option is also specified, the set of $p+1$ parameters is estimated for each level of *varname*. If $p = 1$, then the estimate of ϕ_1 is reported as `rho`, because in this case it represents the correlation between successive error terms.

The `ma` q structure assumes that the errors are an MA process of order q. That is,

$$\epsilon_{ij} = u_{ij} + \theta_1 u_{i-1,j} + \cdots + \theta_q u_{i-q,j}$$

where u_{ij} are i.i.d. Gaussian with mean 0 and variance σ_u^2. `mixed` reports estimates of $\theta_1, \ldots, \theta_q$ and the overall error variance σ_ϵ^2, which can be derived from the above expression. The `t(`*varname*`)` option is required, where *varname* is a time variable used to order the observations within lowest-level groups and to determine any gaps between observations. When the `by(`*varname*`)` option is also specified, the set of $q+1$ parameters is estimated for each level of *varname*.

The `unstructured` structure is the most general and estimates unique variances and unique pairwise covariances for all residuals within the lowest-level grouping. Because the data may be unbalanced and the ordering of the observations is arbitrary, the `t(`*varname*`)` option is required, where *varname* is an identification variable that matches error terms in different groups. If *varname* has n distinct levels, then $n(n+1)/2$ parameters are estimated. Not all n levels need to be observed within each group, but duplicated levels of *varname* within a given group are not allowed because they would cause a singularity in the estimated error-variance matrix for that group. When the `by(`*varname*`)` option is also specified, the set of $n(n+1)/2$ parameters is estimated for each level of *varname*.

The `banded` q structure is a special case of `unstructured` that confines estimation to within the first q off-diagonal elements of the residual variance–covariance matrix and sets the covariances outside this band to 0. As is the case with `unstructured`, the `t(`*varname*`)` option is required, where *varname* is an identification variable that matches error terms in different groups. However, with `banded` variance structures, the ordering of the values in *varname* is significant because it determines which covariances are to be estimated and which are to be set to 0. For example, if *varname* has $n = 5$ distinct values $t = 1, 2, 3, 4, 5$, then a banded variance–covariance structure of order $q = 2$ would estimate the following:

$$\text{Var}(\boldsymbol{\epsilon}_j) = \text{Var}\begin{bmatrix} \epsilon_{1j} \\ \epsilon_{2j} \\ \epsilon_{3j} \\ \epsilon_{4j} \\ \epsilon_{5j} \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} & 0 & 0 \\ \sigma_{12} & \sigma_2^2 & \sigma_{23} & \sigma_{24} & 0 \\ \sigma_{13} & \sigma_{23} & \sigma_3^2 & \sigma_{34} & \sigma_{35} \\ 0 & \sigma_{24} & \sigma_{34} & \sigma_4^2 & \sigma_{45} \\ 0 & 0 & \sigma_{35} & \sigma_{45} & \sigma_5^2 \end{bmatrix}$$

In other words, you would have an unstructured variance matrix that constrains $\sigma_{14} = \sigma_{15} = \sigma_{25} = 0$. If *varname* has n distinct levels, then $(q+1)(2n-q)/2$ parameters are estimated. Not all n levels need to be observed within each group, but duplicated levels of *varname* within a given group are not allowed because they would cause a singularity in the estimated error-variance matrix for that group. When the `by(`*varname*`)` option is also specified, the set of parameters is estimated for each level of *varname*. If q is left unspecified, then `banded` is equivalent to `unstructured`; that is, all variances and covariances are estimated. When $q = 0$, $\text{Var}(\boldsymbol{\epsilon}_j)$ is treated as diagonal and can thus be used to model uncorrelated yet heteroskedastic residual errors.

The `toeplitz` q structure assumes that the residual errors are homoskedastic and that the correlation between two errors is determined by the time lag between the two. That is, $\text{Var}(\epsilon_{ij}) = \sigma_\epsilon^2$ and

$$\text{Corr}(\epsilon_{ij}, \epsilon_{i+k,j}) = \rho_k$$

If the lag k is less than or equal to q, then the pairwise correlation ρ_k is estimated; if the lag is greater than q, then ρ_k is assumed to be 0. If q is left unspecified, then ρ_k is estimated for each observed lag k. The `t(`*varname*`)` option is required, where *varname* is a time variable t used to determine the lags between pairs of residual errors. As such, `t()` must be integer-valued. $q+1$ parameters are estimated: one overall variance σ_ϵ^2 and q correlations. When the `by(`*varname*`)` option is also specified, the set of $q+1$ parameters is estimated for each level of *varname*.

The `exponential` structure is a generalization of the AR structure that allows for noninteger and irregularly spaced time lags. That is, $\text{Var}(\epsilon_{ij}) = \sigma_\epsilon^2$ and

$$\text{Corr}(\epsilon_{ij}, \epsilon_{kj}) = \rho^{|i-k|}$$

for $0 \le \rho \le 1$, with i and k not required to be integers. The `t(`*varname*`)` option is required, where *varname* is a time variable used to determine i and k for each residual-error pair. `t()` is real-valued. `mixed` reports estimates of σ_ϵ^2 and ρ. When the `by(`*varname*`)` option is also specified, these two parameters are estimated for each level of *varname*.

▷ Example 8

Pinheiro and Bates (2000, chap. 5) analyze data from a study of the estrus cycles of mares. Originally analyzed in Pierson and Ginther (1987), the data record the number of ovarian follicles larger than 10mm, daily over a period ranging from three days before ovulation to three days after the subsequent ovulation.

```
. use http://www.stata-press.com/data/r13/ovary
(Ovarian follicles in mares)

. describe

Contains data from http://www.stata-press.com/data/r13/ovary.dta
  obs:           308                          Ovarian follicles in mares
 vars:             6                          20 May 2013 13:49
 size:         5,544                          (_dta has notes)
--------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
--------------------------------------------------------------------------------
mare            byte    %9.0g                 mare ID
stime           float   %9.0g                 Scaled time
follicles       byte    %9.0g                 Number of ovarian follicles > 10
                                                mm in diameter
sin1            float   %9.0g                 sine(2*pi*stime)
cos1            float   %9.0g                 cosine(2*pi*stime)
time            float   %9.0g                 time order within mare
--------------------------------------------------------------------------------
Sorted by:  mare  stime
```

The `stime` variable is time that has been scaled so that ovulation occurs at scaled times 0 and 1, and the `time` variable records the time ordering within mares. Because graphical evidence suggests a periodic behavior, the analysis includes the `sin1` and `cos1` variables, which are sine and cosine transformations of scaled time, respectively.

We consider the following model for the ith measurement on the jth mare:

$$\texttt{follicles}_{ij} = \beta_0 + \beta_1 \texttt{sin1}_{ij} + \beta_2 \texttt{cos1}_{ij} + u_j + \epsilon_{ij}$$

The above model incorporates the cyclical nature of the data as affecting the overall average number of follicles and includes mare-specific random effects u_j. Because we believe successive measurements within each mare are probably correlated (even after controlling for the periodicity in the average), we also model the within-mare errors as being AR of order 2.

```
. mixed follicles sin1 cos1 || mare:, residuals(ar 2, t(time)) reml nolog

Mixed-effects REML regression                   Number of obs      =       308
Group variable: mare                            Number of groups   =        11

                                                Obs per group: min =        25
                                                               avg =      28.0
                                                               max =        31

                                                Wald chi2(2)       =     34.72
Log restricted-likelihood = -772.59855          Prob > chi2        =    0.0000

------------------------------------------------------------------------------
   follicles |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        sin1 |  -2.899228   .5110786    -5.67   0.000    -3.900923   -1.897532
        cos1 |  -.8652936   .5432926    -1.59   0.111    -1.930127    .1995402
       _cons |   12.14455   .9473631    12.82   0.000     10.28775    14.00135
------------------------------------------------------------------------------
```

```
------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
mare: Identity               |
                  var(_cons) |   7.092439   4.401937      2.101337    23.93843
-----------------------------+------------------------------------------------
Residual: AR(2)              |
                        phi1 |   .5386104   .0624899      .4161325    .6610883
                        phi2 |    .144671   .0632041      .0207933    .2685488
                      var(e) |   14.25104   2.435238      10.19512    19.92054
------------------------------------------------------------------------------
LR test vs. linear regression:       chi2(3) =   251.67   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

We picked an order of 2 as a guess, but we could have used LR tests of competing AR models to determine the optimal order, because models of smaller order are nested within those of larger order.

◁

▷ Example 9

Fitzmaurice, Laird, and Ware (2011, chap. 7) analyzed data on 37 subjects who participated in an exercise therapy trial.

```
. use http://www.stata-press.com/data/r13/exercise
(Exercise Therapy Trial)
. describe
Contains data from http://www.stata-press.com/data/r13/exercise.dta
  obs:           259                          Exercise Therapy Trial
 vars:             4                          24 Jun 2012 18:35
 size:         1,036                          (_dta has notes)
--------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
--------------------------------------------------------------------------------
id              byte    %9.0g                 Person ID
day             byte    %9.0g                 Day of measurement
program         byte    %9.0g                 1 = reps increase; 2 = weights
                                                increase
strength        byte    %9.0g                 Strength measurement
--------------------------------------------------------------------------------
Sorted by:  id  day
```

Subjects (variable `id`) were placed on either an increased-repetition regimen (`program==1`) or a program that kept the repetitions constant but increased weight (`program==2`). Muscle-strength measurements (variable `strength`) were taken at baseline (`day==0`) and then every two days over the next twelve days.

Following Fitzmaurice, Laird, and Ware (2011, chap. 7), and to demonstrate fitting residual-error structures to data collected at uneven time points, we confine our analysis to those data collected at baseline and at days 4, 6, 8, and 12. We fit a full two-way factorial model of `strength` on `program` and `day`, with an unstructured residual-error covariance matrix over those repeated measurements taken on the same subject:

```
. keep if inlist(day, 0, 4, 6, 8, 12)
(74 observations deleted)

. mixed strength i.program##i.day || id:,
> noconstant residuals(unstructured, t(day)) nolog

Mixed-effects ML regression                     Number of obs      =       173
Group variable: id                              Number of groups   =        37

                                                Obs per group: min =         3
                                                               avg =       4.7
                                                               max =         5

                                                Wald chi2(9)       =     45.85
Log likelihood = -296.58215                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
    strength |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
   2.program |   1.360119   1.003549     1.36   0.175    -.6068016     3.32704
             |
         day |
          4  |      1.125   .3322583     3.39   0.001     .4737858    1.776214
          6  |   1.360127   .3766894     3.61   0.000     .6218298    2.098425
          8  |   1.583563   .4905876     3.23   0.001     .6220287    2.545097
         12  |   1.623576   .5372947     3.02   0.003     .5704977    2.676654
             |
 program#day |
       2  4  |   -.169034   .4423472    -0.38   0.702    -1.036019    .6979505
       2  6  |   .2113012   .4982385     0.42   0.671    -.7652283    1.187831
       2  8  |  -.1299763   .6524813    -0.20   0.842    -1.408816    1.148864
       2 12  |   .3212829   .7306782     0.44   0.660     -1.11082    1.753386
             |
       _cons |    79.6875   .7560448   105.40   0.000     78.20568    81.16932
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
id:                  (empty) |
-----------------------------+------------------------------------------------
Residual: Unstructured       |
                     var(e0) |    9.14566   2.126248       5.79858    14.42475
                     var(e4) |   11.87114   2.761219      7.524948    18.72757
                     var(e6) |   10.06571   2.348863      6.371091    15.90284
                     var(e8) |   13.22464   3.113921      8.335981    20.98026
                    var(e12) |   13.16909   3.167347      8.219208    21.09995
                  cov(e0,e4) |   9.625236    2.33197      5.054659    14.19581
                  cov(e0,e6) |   8.489043   2.106377       4.36062    12.61747
                  cov(e0,e8) |   9.280414   2.369554      4.636173    13.92465
                 cov(e0,e12) |   8.898006   2.348243      4.295535    13.50048
                  cov(e4,e6) |   10.49185   2.492529      5.606578    15.37711
                  cov(e4,e8) |   11.89787   2.848751      6.314421    17.48132
                 cov(e4,e12) |   11.28344   2.805027      5.785689    16.78119
                  cov(e6,e8) |    11.0507   2.646988      5.862697     16.2387
                 cov(e6,e12) |    10.5006   2.590278      5.423748    15.57745
                 cov(e8,e12) |    12.4091   3.010796      6.508051    18.31016
------------------------------------------------------------------------------
LR test vs. linear regression:       chi2(14) =   314.67   Prob > chi2 = 0.0000

Note: The reported degrees of freedom assumes the null hypothesis is not on
      the boundary of the parameter space.  If this is not true, then the
      reported test is conservative.
```

Because we are using the variable `id` only to group the repeated measurements and not to introduce random effects at the subject level, we use the `noconstant` option to omit any subject-level effects.

The unstructured covariance matrix is the most general and contains many parameters. In this example, we estimate a distinct residual variance for each day and a distinct covariance for each pair of days.

That there is positive covariance between all pairs of measurements is evident, but what is not as evident is whether the covariances may be more parsimoniously represented. One option would be to explore whether the correlation diminishes as the time gap between strength measurements increases and whether it diminishes systematically. Given the irregularity of the time intervals, an exponential structure would be more appropriate than, say, an AR or MA structure.

```
. estimates store unstructured

. mixed strength i.program##i.day || id:, noconstant
> residuals(exponential, t(day)) nolog nofetable

Mixed-effects ML regression                     Number of obs      =       173
Group variable: id                              Number of groups   =        37

                                                Obs per group: min =         3
                                                               avg =       4.7
                                                               max =         5

                                                Wald chi2(9)       =     36.77
Log likelihood = -307.83324                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
id:                  (empty) |
-----------------------------+------------------------------------------------
Residual: Exponential        |
                         rho |   .9786462   .0051238      .9659207    .9866854
                      var(e) |   11.22349   2.338371      7.460765    16.88389
------------------------------------------------------------------------------
LR test vs. linear regression:       chi2(1) =   292.17   Prob > chi2 = 0.0000

Note: The reported degrees of freedom assumes the null hypothesis is not on
      the boundary of the parameter space.  If this is not true, then the
      reported test is conservative.
```

In the above example, we suppressed displaying the main regression parameters because they did not differ much from those of the previous model. While the unstructured model estimated 15 variance–covariance parameters, the exponential model claims to get the job done with just 2, a fact that is not disputed by an LR test comparing the two nested models (at least not at the 0.01 level).

```
. lrtest unstructured .

Likelihood-ratio test                                 LR chi2(13) =     22.50
(Assumption: . nested in unstructured)                Prob > chi2 =    0.0481

Note: The reported degrees of freedom assumes the null hypothesis is not on
      the boundary of the parameter space.  If this is not true, then the
      reported test is conservative.
```

◁

Crossed-effects models

Not all mixed models contain nested levels of random effects.

▷ Example 10

Returning to our longitudinal analysis of pig weights, suppose that instead of (5) we wish to fit

$$\texttt{weight}_{ij} = \beta_0 + \beta_1 \texttt{week}_{ij} + u_i + v_j + \epsilon_{ij} \tag{8}$$

for the $i = 1, \ldots, 9$ weeks and $j = 1, \ldots, 48$ pigs and

$$u_i \sim N(0, \sigma_u^2); \quad v_j \sim N(0, \sigma_v^2); \quad \epsilon_{ij} \sim N(0, \sigma_\epsilon^2)$$

all independently. Both (5) and (8) assume an overall population-average growth curve $\beta_0 + \beta_1 \texttt{week}$ and a random pig-specific shift.

The models differ in how `week` enters into the random part of the model. In (5), we assume that the effect due to `week` is linear and pig specific (a random slope); in (8), we assume that the effect due to `week`, u_i, is systematic to that week and common to all pigs. The rationale behind (8) could be that, assuming that the pigs were measured contemporaneously, we might be concerned that week-specific random factors such as weather and feeding patterns had significant systematic effects on all pigs.

Model (8) is an example of a two-way crossed-effects model, with the pig effects v_j being crossed with the week effects u_i. One way to fit such models is to consider all the data as one big cluster, and treat the u_i and v_j as a series of $9 + 48 = 57$ random coefficients on indicator variables for `week` and `pig`. In the notation of (2),

$$\mathbf{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_9 \\ v_1 \\ \vdots \\ v_{48} \end{bmatrix} \sim N(\mathbf{0}, \mathbf{G}); \quad \mathbf{G} = \begin{bmatrix} \sigma_u^2 \mathbf{I}_9 & \mathbf{0} \\ \mathbf{0} & \sigma_v^2 \mathbf{I}_{48} \end{bmatrix}$$

Because $\mathbf{G}$ is block diagonal, it can be represented in `mixed` as repeated-level equations. All we need is an identification variable to identify all the observations as one big group and a way to tell `mixed` to treat `week` and `pig` as factor variables (or equivalently, as two sets of overparameterized indicator variables identifying weeks and pigs, respectively). `mixed` supports the special group designation `_all` for the former and the `R.`*varname* notation for the latter.

```
. use http://www.stata-press.com/data/r13/pig, clear
(Longitudinal analysis of pig weights)
. mixed weight week || _all: R.week || _all: R.id
Performing EM optimization:
Performing gradient-based optimization:
Iteration 0:   log likelihood =  -1013.824
Iteration 1:   log likelihood =  -1013.824
Computing standard errors:
Mixed-effects ML regression                     Number of obs      =       432
Group variable: _all                            Number of groups   =         1
                                                Obs per group: min =       432
                                                               avg =     432.0
                                                               max =       432
                                                Wald chi2(1)       =  13258.28
Log likelihood =  -1013.824                     Prob > chi2        =    0.0000
```

weight	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
week	6.209896	.0539313	115.14	0.000	6.104192	6.315599
_cons	19.35561	.6333982	30.56	0.000	18.11418	20.59705

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
_all: Identity				
var(R.week)	.0849874	.0868856	.0114588	.6303302
_all: Identity				
var(R.id)	14.83623	3.126142	9.816733	22.42231
var(Residual)	4.297328	.3134404	3.724888	4.957741

```
LR test vs. linear regression:       chi2(2) =   474.85   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
. estimates store crossed
```

Thus we estimate $\widehat{\sigma}_u^2 = 0.08$ and $\widehat{\sigma}_v^2 = 14.84$. Both (5) and (8) estimate a total of five parameters: two fixed effects and three variance components. The models, however, are not nested within each other, which precludes the use of an LR test to compare both models. Refitting model (5) and looking at the Akaike information criteria values by using `estimates stats`,

```
. quietly mixed weight week || id:week
. estimates stats crossed .
Akaike's information criterion and Bayesian information criterion
```

Model	Obs	ll(null)	ll(model)	df	AIC	BIC
crossed	432	.	-1013.824	5	2037.648	2057.99
.	432	.	-869.0383	5	1748.077	1768.419

Note: N=Obs used in calculating BIC; see **[R] BIC note**

definitely favors model (5). This finding is not surprising given that our rationale behind (8) was somewhat fictitious. In our `estimates stats` output, the values of `ll(null)` are missing. `mixed` does not fit a constant-only model as part of its usual estimation of the full model, but you can use `mixed` to fit a constant-only model directly, if you wish.

◁

The `R.`*varname* notation is equivalent to giving a list of overparameterized (none dropped) indicator variables for use in a random-effects specification. When you specify `R.`*varname*, `mixed` handles the calculations internally rather than creating the indicators in the data. Because the set of indicators is overparameterized, `R.`*varname* implies `noconstant`. You can include factor variables in the fixed-effects specification by using standard methods; see **[U] 11.4.3 Factor variables**. However, random-effects equations support only the `R.`*varname* factor specification. For more complex factor specifications (such as interactions) in random-effects equations, use `generate` to form the variables manually, as we demonstrated in example 6.

❑ Technical note

Although we were able to fit the crossed-effects model (8), it came at the expense of increasing the column dimension of our random-effects design from 2 in model (5) to 57 in model (8). Computation time and memory requirements grow (roughly) quadratically with the dimension of the random effects. As a result, fitting such crossed-effects models is feasible only when the total column dimension is small to moderate.

Reexamining model (8), we note that if we drop u_i, we end up with a model equivalent to (4), meaning that we could have fit (4) by typing

```
. mixed weight week || _all: R.id
```

instead of

```
. mixed weight week || id:
```

as we did when we originally fit the model. The results of both estimations are identical, but the latter specification, organized at the cluster (pig) level with random-effects dimension 1 (a random intercept) is much more computationally efficient. Whereas with the first form we are limited in how many pigs we can analyze, there is no such limitation with the second form.

Furthermore, we fit model (8) by using

```
. mixed weight week || _all: R.week || _all: R.id
```

as a direct way to demonstrate the `R.` notation. However, we can technically treat pigs as nested within the `_all` group, yielding the equivalent and more efficient (total column dimension 10) way to fit (8):

```
. mixed weight week || _all: R.week || id:
```

We leave it to you to verify that both produce identical results. See Rabe-Hesketh and Skrondal (2012) for additional techniques to make calculations more efficient in more complex models.

❑

▷ Example 11

As another example of how the same model may be fit in different ways by using `mixed` (and as a way to demonstrate `covariance(exchangeable)`), consider the three-level model used in example 4:

$$\mathbf{y}_{jk} = \mathbf{X}_{jk}\boldsymbol{\beta} + u_k^{(3)} + u_{jk}^{(2)} + \boldsymbol{\epsilon}_{jk}$$

where $\mathbf{y}_{jk}$ represents the logarithms of gross state products for the $n_{jk} = 17$ observations from state j in region k, $\mathbf{X}_{jk}$ is a set of regressors, $u_k^{(3)}$ is a random intercept at the region level, and $u_{jk}^{(2)}$ is a random intercept at the state (nested within region) level. We assume that $u_k^{(3)} \sim N(0, \sigma_3^2)$ and $u_{jk}^{(2)} \sim N(0, \sigma_2^2)$ independently. Define

$$\mathbf{v}_k = \begin{bmatrix} u_k^{(3)} + u_{1k}^{(2)} \\ u_k^{(3)} + u_{2k}^{(2)} \\ \vdots \\ u_k^{(3)} + u_{M_k,k}^{(2)} \end{bmatrix}$$

where M_k is the number of states in region k. Making this substitution, we can stack the observations for all the states within region k to get

$$\mathbf{y}_k = \mathbf{X}_k \boldsymbol{\beta} + \mathbf{Z}_k \mathbf{v}_k + \boldsymbol{\epsilon}_k$$

where $\mathbf{Z}_k$ is a set of indicators identifying the states within each region; that is,

$$\mathbf{Z}_k = \mathbf{I}_{M_k} \otimes \mathbf{J}_{17}$$

for a k-column vector of 1s $\mathbf{J}_k$, and

$$\boldsymbol{\Sigma} = \text{Var}(\mathbf{v}_k) = \begin{bmatrix} \sigma_3^2 + \sigma_2^2 & \sigma_3^2 & \cdots & \sigma_3^2 \\ \sigma_3^2 & \sigma_3^2 + \sigma_2^2 & \cdots & \sigma_3^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_3^2 & \sigma_3^2 & \sigma_3^2 & \sigma_3^2 + \sigma_2^2 \end{bmatrix}_{M_k \times M_k}$$

Because $\boldsymbol{\Sigma}$ is an exchangeable matrix, we can fit this alternative form of the model by specifying the `exchangeable` covariance structure.

```
. use http://www.stata-press.com/data/r13/productivity
(Public Capital Productivity)
. mixed gsp private emp hwy water other unemp || region: R.state,
> cov(exchangeable)
  (output omitted)
Mixed-effects ML regression                     Number of obs      =       816
Group variable: region                          Number of groups   =         9
                                                Obs per group: min =        51
                                                               avg =      90.7
                                                               max =       136
                                                Wald chi2(6)       =  18829.06
Log likelihood =  1430.5017                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
         gsp |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     private |   .2671484   .0212591    12.57   0.000     .2254813    .3088154
         emp |   .7540721   .0261868    28.80   0.000     .7027468    .8053973
         hwy |   .0709767    .023041     3.08   0.002     .0258172    .1161363
       water |   .0761187   .0139248     5.47   0.000     .0488266    .1034109
       other |  -.0999955   .0169366    -5.90   0.000    -.1331907   -.0668004
       unemp |  -.0058983   .0009031    -6.53   0.000    -.0076684   -.0041282
       _cons |   2.128823   .1543855    13.79   0.000     1.826233    2.431413
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
region: Exchangeable         |
                var(R.state) |   .0077263   .0017926      .0049032    .0121749
                cov(R.state) |   .0014506   .0012995     -.0010963    .0039975
-----------------------------+------------------------------------------------
               var(Residual) |   .0013461   .0000689      .0012176    .0014882
------------------------------------------------------------------------------
LR test vs. linear regression:       chi2(2) =  1154.73   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

The estimates of the fixed effects and their standard errors are equivalent to those from example 4, and remapping the variance components from $(\sigma_3^2 + \sigma_2^2, \sigma_3^2, \sigma_\epsilon^2)$, as displayed here, to $(\sigma_3^2, \sigma_2^2, \sigma_\epsilon^2)$, as displayed in example 4, will show that they are equivalent as well.

Of course, given the discussion in the previous technical note, it is more efficient to fit this model as we did originally, as a three-level model.

◁

Diagnosing convergence problems

Given the flexibility of mixed-effects models, you will find that some models fail to converge when used with your data; see *Diagnosing convergence problems* in [ME] **me** for advice applicable to mixed-effects models in general.

In unweighted LME models with independent and homoskedastic residuals, one useful way to diagnose problems of nonconvergence is to rely on the EM algorithm (Dempster, Laird, and Rubin 1977), normally used by `mixed` only as a means of refining starting values. The advantages of EM are that it does not require a Hessian calculation, each successive EM iteration will result in a larger likelihood, iterations can be calculated quickly, and iterations will quickly bring parameter estimates into a neighborhood of the solution. The disadvantages of EM are that, once in a neighborhood of the solution, it can be slow to converge, if at all, and EM provides no facility for estimating standard errors of the estimated variance components. One useful property of EM is that it is always willing to provide a solution if you allow it to iterate enough times, if you are satisfied with being in a neighborhood of the optimum rather than right on the optimum, and if standard errors of variance components are not crucial to your analysis.

If you encounter a nonconvergent model, try using the `emonly` option to bypass gradient-based optimization. Use `emiterate(#)` to specify the maximum number of EM iterations, which you will usually want to set much higher than the default of 20. If your EM solution shows an estimated variance component that is near 0, a ridge is formed by an interval of values near 0, which produces the same likelihood and looks equally good to the optimizer. In this case, the solution is to drop the offending variance component from the model.

Survey data

Multilevel modeling of survey data is a little different from standard modeling in that weighted sampling can take place at multiple levels in the model, resulting in multiple sampling weights. Most survey datasets, regardless of the design, contain one overall inclusion weight for each observation in the data. This weight reflects the inverse of the probability of ultimate selection, and by “ultimate” we mean that it factors in all levels of clustered sampling, corrections for noninclusion and oversampling, poststratification, etc.

For simplicity, in what follows assume a simple two-stage sampling design where groups are randomly sampled and then individuals within groups are sampled. Also assume that no additional weight corrections are performed; that is, sampling weights are simply the inverse of the probability of selection. The sampling weight for observation i in cluster j in our two-level sample is then $w_{ij} = 1/\pi_{ij}$, where π_{ij} is the probability that observation i, j is selected. If you were performing a standard analysis such as OLS regression with `regress`, you would simply use a variable holding w_{ij} as your `pweight` variable, and the fact that it came from two levels of sampling would not concern you. Perhaps you would type `vce(cluster` *groupvar*`)` where *groupvar* identifies the top-level groups to get standard errors that control for correlation within these groups, but you would still use only a single weight variable.

Now take these same data and fit a two-level model with `mixed`. As seen in (14) in *Methods and formulas* later in this entry, it is not sufficient to use the single sampling weight w_{ij}, because weights enter into the log likelihood at both the group level and the individual level. Instead, what is required for a two-level model under this sampling design is w_j, the inverse of the probability that group j is selected in the first stage, and $w_{i|j}$, the inverse of the probability that individual i from group j is selected at the second stage *conditional on group j already being selected.* It simply will not do to just use w_{ij} without making any assumptions about w_j.

Given the rules of conditional probability, $w_{ij} = w_j w_{i|j}$. If your dataset has only w_{ij}, then you will need to either assume equal probability sampling at the first stage ($w_j = 1$ for all j) or find some way to recover w_j from other variables in your data; see Rabe-Hesketh and Skrondal (2006) and the references therein for some suggestions on how to do this, but realize that there is little yet known about how well these approximations perform in practice.

What you really need to fit your two-level model are data that contain w_j in addition to either w_{ij} or $w_{i|j}$. If you have w_{ij}—that is, the unconditional inclusion weight for observation i, j—then you need to either divide w_{ij} by w_j to obtain $w_{i|j}$ or rescale w_{ij} so that its dependence on w_j disappears. If you already have $w_{i|j}$, then rescaling becomes optional (but still an important decision to make).

Weight rescaling is not an exact science, because the scale of the level-one weights is at issue regardless of whether they represent w_{ij} or $w_{i|j}$: because w_{ij} is unique to group j, the group-to-group magnitudes of these weights need to be normalized so that they are "consistent" from group to group. This is in stark contrast to a standard analysis, where the scale of sampling weights does not factor into estimation, instead only affecting the estimate of the total population size.

`mixed` offers three methods for standardizing weights in a two-level model, and you can specify which method you want via the `pwscale()` option. If you specify `pwscale(size)`, then the $w_{i|j}$ (or w_{ij}, it does not matter) are scaled to sum to the cluster size n_j. Method `pwscale(effective)` adds in a dependence on the sum of the squared weights so that level-one weights sum to the "effective" sample size. Just like `pwscale(size)`, `pwscale(effective)` also behaves the same whether you have $w_{i|j}$ or w_{ij}, and so it can be used with either.

Although both `pwscale(size)` and `pwscale(effective)` leave w_j untouched, the `pwscale(gk)` method is a little different in that 1) it changes the weights at both levels and 2) it does assume you have $w_{i|j}$ for level-one weights and not w_{ij} (if you have the latter, then first divide by w_j). Using the method of Graubard and Korn (1996), it sets the weights at the group level (level two) to the cluster averages of the products of both level weights (this product being w_{ij}). It then sets the individual weights to 1 everywhere; see *Methods and formulas* for the computational details of all three methods.

Determining which method is "best" is a tough call and depends on cluster size (the smaller the clusters, the greater the sensitivity to scale), whether the sampling is informative (that is, the sampling weights are correlated with the residuals), whether you are interested primarily in regression coefficients or in variance components, whether you have a simple random-intercept model or a more complex random-coefficients model, and other factors; see Rabe-Hesketh and Skrondal (2006), Carle (2009), and Pfeffermann et al. (1998) for some detailed advice. At the very least, you want to compare estimates across all three scaling methods (four, if you add no scaling) and perform a sensitivity analysis.

If you choose to rescale level-one weights, it does not matter whether you have $w_{i|j}$ or w_{ij}. For the `pwscale(size)` and `pwscale(effective)` methods, you get identical results, and even though `pwscale(gk)` assumes $w_{i|j}$, you can obtain this as $w_{i|j} = w_{ij}/w_j$ before proceeding.

If you do not specify `pwscale()`, then no scaling takes place, and thus at a minimum, you need to make sure you have $w_{i|j}$ in your data and not w_{ij}.

▷ Example 12

Rabe-Hesketh and Skrondal (2006) analyzed data from the 2000 Programme for International Student Assessment (PISA) study on reading proficiency among 15-year-old American students, as performed by the Organisation for Economic Co-operation and Development (OECD). The original study was a three-stage cluster sample, where geographic areas were sampled at the first stage, schools at the second, and students at the third. Our version of the data does not contain the geographic-areas variable, so we treat this as a two-stage sample where schools are sampled at the first stage and students at the second.

```
. use http://www.stata-press.com/data/r13/pisa2000
(Programme for International Student Assessment (PISA) 2000 data)
. describe
Contains data from http://www.stata-press.com/data/r13/pisa2000.dta
  obs:         2,069                          Programme for International
                                                Student Assessment (PISA) 2000
                                                data
 vars:            11                          12 Jun 2012 10:08
 size:        37,242                          (_dta has notes)
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
female          byte    %8.0g                 1 if female
isei            byte    %8.0g                 International socio-economic
                                                index
w_fstuwt        float   %9.0g                 Student-level weight
wnrschbw        float   %9.0g                 School-level weight
high_school     byte    %8.0g                 1 if highest level by either
                                                parent is high school
college         byte    %8.0g                 1 if highest level by either
                                                parent is college
one_for         byte    %8.0g                 1 if one parent foreign born
both_for        byte    %8.0g                 1 if both parents are foreign
                                                born
test_lang       byte    %8.0g                 1 if English (the test language)
                                                is spoken at home
pass_read       byte    %8.0g                 1 if passed reading proficiency
                                                threshold
id_school       int     %8.0g                 School ID
-------------------------------------------------------------------------------
Sorted by:
```

For student i in school j, where the variable `id_school` identifies the schools, the variable `w_fstuwt` is a student-level overall inclusion weight (w_{ij}, not $w_{i|j}$) adjusted for noninclusion and nonparticipation of students, and the variable `wnrschbw` is the school-level weight w_j adjusted for oversampling of schools with more minority students. The weight adjustments do not interfere with the methods prescribed above, and thus we can treat the weight variables simply as w_{ij} and w_j, respectively.

Rabe-Hesketh and Skrondal (2006) fit a two-level logistic model for passing a reading proficiency threshold. We fit a two-level linear random-intercept model for socioeconomic index. Because we have w_{ij} and not $w_{i|j}$, we rescale using `pwscale(size)` and thus obtain results as if we had $w_{i|j}$.

```
. mixed isei female high_school college one_for both_for test_lang
> [pw=w_fstuwt] || id_school:, pweight(wnrschbw) pwscale(size)
  (output omitted )
Mixed-effects regression                        Number of obs      =      2069
Group variable: id_school                       Number of groups   =       148

                                                Obs per group: min =         1
                                                               avg =      14.0
                                                               max =        28

                                                Wald chi2(6)       =    187.23
Log pseudolikelihood = -1443093.9               Prob > chi2        =    0.0000

                            (Std. Err. adjusted for 148 clusters in id_school)
```

isei	Coef.	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
female	.59379	.8732886	0.68	0.497	-1.117824	2.305404
high_school	6.410618	1.500337	4.27	0.000	3.470011	9.351224
college	19.39494	2.121145	9.14	0.000	15.23757	23.55231
one_for	-.9584613	1.789947	-0.54	0.592	-4.466692	2.54977
both_for	-.2021101	2.32633	-0.09	0.931	-4.761633	4.357413
test_lang	2.519539	2.393165	1.05	0.292	-2.170978	7.210056
_cons	28.10788	2.435712	11.54	0.000	23.33397	32.88179

Random-effects Parameters	Estimate	Robust Std. Err.	[95% Conf.	Interval]
id_school: Identity				
var(_cons)	34.69374	8.574865	21.37318	56.31617
var(Residual)	218.7382	11.22111	197.8147	241.8748

Notes:

1. We specified the level-one weights using standard Stata weight syntax, that is, [pw=w_fstuwt].
2. We specified the level-two weights via the pweight(wnrschbw) option as part of the random-effects specification for the id_school level. As such, it is treated as a school-level weight. Accordingly, wnrschbw needs to be constant within schools, and mixed did check for that before estimating.
3. Because our level-one weights are unconditional, we specified pwscale(size) to rescale them.
4. As is the case with other estimation commands in Stata, standard errors in the presence of sampling weights are robust.
5. Robust standard errors are clustered at the top level of the model, and this will always be true unless you specify vce(cluster *clustvar*), where *clustvar* identifies an even higher level of grouping.

As a form of sensitivity analysis, we compare the above with scaling via pwscale(gk). Because pwscale(gk) assumes $w_{i|j}$, you want to first divide w_{ij} by w_j. But you can handle that within the weight specification itself.

```
. mixed isei female high_school college one_for both_for test_lang
> [pw=w_fstuwt/wnrschbw] || id_school:, pweight(wnrschbw) pwscale(gk)
  (output omitted)
Mixed-effects regression                        Number of obs     =       2069
Group variable: id_school                       Number of groups  =        148

                                                Obs per group: min =         1
                                                               avg =      14.0
                                                               max =        28

                                                Wald chi2(6)      =     291.37
Log pseudolikelihood = -7270505.6               Prob > chi2       =     0.0000

                           (Std. Err. adjusted for 148 clusters in id_school)
```

isei	Coef.	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
female	-.3519458	.7436334	-0.47	0.636	-1.80944	1.105549
high_school	7.074911	1.139777	6.21	0.000	4.84099	9.308833
college	19.27285	1.286029	14.99	0.000	16.75228	21.79342
one_for	-.9142879	1.783091	-0.51	0.608	-4.409082	2.580506
both_for	1.214151	1.611885	0.75	0.451	-1.945085	4.373388
test_lang	2.661866	1.556491	1.71	0.087	-.3887996	5.712532
_cons	31.20145	1.907413	16.36	0.000	27.46299	34.93991

Random-effects Parameters	Estimate	Robust Std. Err.	[95% Conf.	Interval]
id_school: Identity				
var(_cons)	31.67522	6.792239	20.80622	48.22209
var(Residual)	226.2429	8.150714	210.8188	242.7955

The results are somewhat similar to before, which is good news from a sensitivity standpoint. Note that we specified `[pw=w_fstwtw/wnrschbw]` and thus did the conversion from w_{ij} to $w_{i|j}$ within our call to `mixed`.

◁

We close this section with a bit of bad news. Although weight rescaling and the issues that arise have been well studied for two-level models, as pointed out by Carle (2009), "... a best practice for scaling weights across multiple levels has yet to be advanced." As such, `pwscale()` is currently supported only for two-level models. If you are fitting a higher-level model with survey data, you need to make sure your sampling weights are conditional on selection at the previous stage and not overall inclusion weights, because there is currently no rescaling option to fall back on if you do not.

Stored results

mixed stores the following in e():

Scalars

e(N)	number of observations
e(k)	number of parameters
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(k_res)	number of residual-error parameters
e(N_clust)	number of clusters
e(nrgroups)	number of residual-error by() groups
e(ar_p)	AR order of residual errors, if specified
e(ma_q)	MA order of residual errors, if specified
e(res_order)	order of residual-error structure, if appropriate
e(df_m)	model degrees of freedom
e(ll)	log (restricted) likelihood
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(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros

e(cmd)	mixed
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(wtype)	weight type (first-level weights)
e(wexp)	weight expression (first-level weights)
e(fweight*k*)	fweight expression for *k*th highest level, if specified
e(pweight*k*)	pweight expression for *k*th highest level, if specified
e(ivars)	grouping variables
e(title)	title in estimation output
e(redim)	random-effects dimensions
e(vartypes)	variance-structure types
e(revars)	random-effects covariates
e(resopt)	residuals() specification, as typed
e(rstructure)	residual-error structure
e(rstructlab)	residual-error structure output label
e(rbyvar)	residual-error by() variable, if specified
e(rglabels)	residual-error by() groups labels
e(pwscale)	sampling-weight scaling method
e(timevar)	residual-error t() variable, if specified
e(chi2type)	Wald; type of model χ^2 test
e(clustvar)	name of cluster variable
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. Err.
e(method)	ML or REML
e(opt)	type of optimization
e(optmetric)	matsqrt or matlog; random-effects matrix parameterization
e(emonly)	emonly, if specified
e(ml_method)	type of ml method
e(technique)	maximization technique
e(properties)	b V
e(estat_cmd)	program used to implement estat
e(predict)	program used to implement predict
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved

Matrices

`e(b)`	coefficient 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(tmap)`	ID mapping for unstructured residual errors
`e(V)`	variance–covariance matrix of the estimator
`e(V_modelbased)`	model-based variance

Functions

`e(sample)`	marks estimation sample

Methods and formulas

As given by (1), in the absence of weights we have the linear mixed model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + \boldsymbol{\epsilon}$$

where $\mathbf{y}$ is the $n \times 1$ vector of responses, $\mathbf{X}$ is an $n \times p$ design/covariate matrix for the fixed effects $\boldsymbol{\beta}$, and $\mathbf{Z}$ is the $n \times q$ design/covariate matrix for the random effects $\mathbf{u}$. The $n \times 1$ vector of errors $\boldsymbol{\epsilon}$ is for now assumed to be multivariate normal with mean 0 and variance matrix $\sigma^2_\epsilon \mathbf{I}_n$. We also assume that $\mathbf{u}$ has variance–covariance matrix $\mathbf{G}$ and that $\mathbf{u}$ is orthogonal to $\boldsymbol{\epsilon}$ so that

$$\text{Var}\begin{bmatrix}\mathbf{u}\\ \boldsymbol{\epsilon}\end{bmatrix} = \begin{bmatrix}\mathbf{G} & \mathbf{0}\\ \mathbf{0} & \sigma^2_\epsilon \mathbf{I}_n\end{bmatrix}$$

Considering the combined error term $\mathbf{Z}\mathbf{u} + \boldsymbol{\epsilon}$, we see that $\mathbf{y}$ is multivariate normal with mean $\mathbf{X}\boldsymbol{\beta}$ and $n \times n$ variance–covariance matrix

$$\mathbf{V} = \mathbf{Z}\mathbf{G}\mathbf{Z}' + \sigma^2_\epsilon \mathbf{I}_n$$

Defining $\boldsymbol{\theta}$ as the vector of unique elements of $\mathbf{G}$ results in the log likelihood

$$L(\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma^2_\epsilon) = -\frac{1}{2}\left\{n \log(2\pi) + \log|\mathbf{V}| + (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'\mathbf{V}^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\right\} \tag{9}$$

which is maximized as a function of $\boldsymbol{\beta}$, $\boldsymbol{\theta}$, and σ^2_ϵ. As explained in chapter 6 of Searle, Casella, and McCulloch (1992), considering instead the likelihood of a set of linear contrasts $\mathbf{K}\mathbf{y}$ that do not depend on $\boldsymbol{\beta}$ results in the restricted log likelihood

$$L_R(\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma^2_\epsilon) = L(\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma^2_\epsilon) - \frac{1}{2}\log\left|\mathbf{X}'\mathbf{V}^{-1}\mathbf{X}\right| \tag{10}$$

Given the high dimension of $\mathbf{V}$, however, the log-likelihood and restricted log-likelihood criteria are not usually computed by brute-force application of the above expressions. Instead, you can simplify the problem by subdividing the data into independent clusters (and subclusters if possible) and using matrix decomposition methods on the smaller matrices that result from treating each cluster one at a time.

Consider the two-level model described previously in (2),

$$\mathbf{y}_j = \mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j + \boldsymbol{\epsilon}_j$$

for $j = 1, \ldots, M$ clusters with cluster j containing n_j observations, with $\text{Var}(\mathbf{u}_j) = \boldsymbol{\Sigma}$, a $q \times q$ matrix.

Efficient methods for computing (9) and (10) are given in chapter 2 of Pinheiro and Bates (2000). Namely, for the two-level model, define $\boldsymbol{\Delta}$ to be the Cholesky factor of $\sigma_\epsilon^2\boldsymbol{\Sigma}^{-1}$, such that $\sigma_\epsilon^2\boldsymbol{\Sigma}^{-1} = \boldsymbol{\Delta}'\boldsymbol{\Delta}$. For $j = 1,\dots,M$, decompose

$$\begin{bmatrix} \mathbf{Z}_j \\ \boldsymbol{\Delta} \end{bmatrix} = \mathbf{Q}_j \begin{bmatrix} \mathbf{R}_{11j} \\ \mathbf{0} \end{bmatrix}$$

by using an orthogonal-triangular (QR) decomposition, with $\mathbf{Q}_j$ a $(n_j + q)$-square matrix and $\mathbf{R}_{11j}$ a q-square matrix. We then apply $\mathbf{Q}_j$ as follows:

$$\begin{bmatrix} \mathbf{R}_{10j} \\ \mathbf{R}_{00j} \end{bmatrix} = \mathbf{Q}_j' \begin{bmatrix} \mathbf{X}_j \\ \mathbf{0} \end{bmatrix}; \qquad \begin{bmatrix} \mathbf{c}_{1j} \\ \mathbf{c}_{0j} \end{bmatrix} = \mathbf{Q}_j' \begin{bmatrix} \mathbf{y}_j \\ \mathbf{0} \end{bmatrix}$$

Stack the $\mathbf{R}_{00j}$ and $\mathbf{c}_{0j}$ matrices, and perform the additional QR decomposition

$$\begin{bmatrix} \mathbf{R}_{001} & \mathbf{c}_{01} \\ \vdots & \vdots \\ \mathbf{R}_{00M} & \mathbf{c}_{0M} \end{bmatrix} = \mathbf{Q}_0 \begin{bmatrix} \mathbf{R}_{00} & \mathbf{c}_0 \\ \mathbf{0} & \mathbf{c}_1 \end{bmatrix}$$

Pinheiro and Bates (2000) show that ML estimates of $\boldsymbol{\beta}$, σ_ϵ^2, and $\boldsymbol{\Delta}$ (the unique elements of $\boldsymbol{\Delta}$, that is) are obtained by maximizing the profile log likelihood (profiled in $\boldsymbol{\Delta}$)

$$L(\boldsymbol{\Delta}) = \frac{n}{2}\left\{\log n - \log(2\pi) - 1\right\} - n\log||\mathbf{c}_1|| + \sum_{j=1}^{M} \log\left|\frac{\det(\boldsymbol{\Delta})}{\det(\mathbf{R}_{11j})}\right| \tag{11}$$

where $||\cdot||$ denotes the 2-norm. Following this maximization with

$$\widehat{\boldsymbol{\beta}} = \mathbf{R}_{00}^{-1}\mathbf{c}_0; \quad \widehat{\sigma}_\epsilon^2 = n^{-1}||\mathbf{c}_1||^2 \tag{12}$$

REML estimates are obtained by maximizing

$$\begin{aligned} L_R(\boldsymbol{\Delta}) = \frac{n-p}{2}\left\{\log(n-p) - \log(2\pi) - 1\right\} - (n-p)\log||\mathbf{c}_1|| \\ - \log|\det(\mathbf{R}_{00})| + \sum_{j=1}^{M} \log\left|\frac{\det(\boldsymbol{\Delta})}{\det(\mathbf{R}_{11j})}\right| \end{aligned} \tag{13}$$

followed by

$$\widehat{\boldsymbol{\beta}} = \mathbf{R}_{00}^{-1}\mathbf{c}_0; \quad \widehat{\sigma}_\epsilon^2 = (n-p)^{-1}||\mathbf{c}_1||^2$$

For numerical stability, maximization of (11) and (13) is not performed with respect to the unique elements of $\boldsymbol{\Delta}$ but instead with respect to the unique elements of the matrix square root (or matrix logarithm if the `matlog` option is specified) of $\boldsymbol{\Sigma}/\sigma_\epsilon^2$; define $\boldsymbol{\gamma}$ to be the vector containing these elements.

Once maximization with respect to $\boldsymbol{\gamma}$ is completed, $(\boldsymbol{\gamma}, \sigma_\epsilon^2)$ is reparameterized to $\{\boldsymbol{\alpha}, \log(\sigma_\epsilon)\}$, where $\boldsymbol{\alpha}$ is a vector containing the unique elements of $\boldsymbol{\Sigma}$, expressed as logarithms of standard deviations for the diagonal elements and hyperbolic arctangents of the correlations for off-diagonal elements. This last step is necessary 1) to obtain a joint variance–covariance estimate of the elements of $\boldsymbol{\Sigma}$ and σ_ϵ^2; 2) to obtain a parameterization under which parameter estimates can be interpreted individually, rather than as elements of a matrix square root (or logarithm); and 3) to parameterize these elements such that their ranges each encompass the entire real line.

Obtaining a joint variance–covariance matrix for the estimated $\{\boldsymbol{\alpha}, \log(\sigma_\epsilon)\}$ requires the evaluation of the log likelihood (or log-restricted likelihood) with only $\boldsymbol{\beta}$ profiled out. For ML, we have

$$
\begin{aligned}
L^*\{\boldsymbol{\alpha}, \log(\sigma_\epsilon)\} &= L\{\boldsymbol{\Delta}(\boldsymbol{\alpha}, \sigma_\epsilon^2), \sigma_\epsilon^2\} \\
&= -\frac{n}{2}\log(2\pi\sigma_\epsilon^2) - \frac{||\mathbf{c}_1||^2}{2\sigma_\epsilon^2} + \sum_{j=1}^{M} \log\left|\frac{\det(\boldsymbol{\Delta})}{\det(\mathbf{R}_{11j})}\right|
\end{aligned}
$$

with the analogous expression for REML.

The variance–covariance matrix of $\widehat{\boldsymbol{\beta}}$ is estimated as

$$
\widehat{\text{Var}}(\widehat{\boldsymbol{\beta}}) = \widehat{\sigma}_\epsilon^2 \mathbf{R}_{00}^{-1} \left(\mathbf{R}_{00}^{-1}\right)'
$$

but this does not mean that $\widehat{\text{Var}}(\widehat{\boldsymbol{\beta}})$ is identical under both ML and REML because $\mathbf{R}_{00}$ depends on $\boldsymbol{\Delta}$. Because $\widehat{\boldsymbol{\beta}}$ is asymptotically uncorrelated with $\{\widehat{\boldsymbol{\alpha}}, \log(\widehat{\sigma}_\epsilon)\}$, the covariance of $\widehat{\boldsymbol{\beta}}$ with the other estimated parameters is treated as 0.

Parameter estimates are stored in `e(b)` as $\{\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\alpha}}, \log(\widehat{\sigma}_\epsilon)\}$, with the corresponding (block-diagonal) variance–covariance matrix stored in `e(V)`. Parameter estimates can be displayed in this metric by specifying the `estmetric` option. However, in `mixed` output, variance components are most often displayed either as variances and covariances or as standard deviations and correlations.

EM iterations are derived by considering the $\mathbf{u}_j$ in (2) as missing data. Here we describe the procedure for maximizing the log likelihood via EM; the procedure for maximizing the restricted log likelihood is similar. The log likelihood for the full data $(\mathbf{y}, \mathbf{u})$ is

$$
L_F(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \sigma_\epsilon^2) = \sum_{j=1}^{M} \left\{\log f_1(\mathbf{y}_j|\mathbf{u}_j, \boldsymbol{\beta}, \sigma_\epsilon^2) + \log f_2(\mathbf{u}_j|\boldsymbol{\Sigma})\right\}
$$

where $f_1(\cdot)$ is the density function for multivariate normal with mean $\mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j$ and variance $\sigma_\epsilon^2\mathbf{I}_{n_j}$, and $f_2(\cdot)$ is the density for multivariate normal with mean $\mathbf{0}$ and $q \times q$ covariance matrix $\boldsymbol{\Sigma}$. As before, we can profile $\boldsymbol{\beta}$ and σ_ϵ^2 out of the optimization, yielding the following EM iterative procedure:

1. For the current iterated value of $\boldsymbol{\Sigma}^{(t)}$, fix $\widehat{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}(\boldsymbol{\Sigma}^{(t)})$ and $\widehat{\sigma}_\epsilon^2 = \widehat{\sigma}_\epsilon^2(\boldsymbol{\Sigma}^{(t)})$ according to (12).
2. Expectation step: Calculate

$$
\begin{aligned}
D(\boldsymbol{\Sigma}) &\equiv E\left\{L_F(\widehat{\boldsymbol{\beta}}, \boldsymbol{\Sigma}, \widehat{\sigma}_\epsilon^2)|\mathbf{y}\right\} \\
&= C - \frac{M}{2}\log\det\left(\boldsymbol{\Sigma}\right) - \frac{1}{2}\sum_{j=1}^{M} E\left(\mathbf{u}_j'\boldsymbol{\Sigma}^{-1}\mathbf{u}_j|\mathbf{y}\right)
\end{aligned}
$$

   where C is a constant that does not depend on $\boldsymbol{\Sigma}$, and the expected value of the quadratic form $\mathbf{u}_j'\boldsymbol{\Sigma}^{-1}\mathbf{u}_j$ is taken with respect to the conditional density $f(\mathbf{u}_j|\mathbf{y}, \widehat{\boldsymbol{\beta}}, \boldsymbol{\Sigma}^{(t)}, \widehat{\sigma}_\epsilon^2)$.

3. Maximization step: Maximize $D(\boldsymbol{\Sigma})$ to produce $\boldsymbol{\Sigma}^{(t+1)}$.

For general, symmetric $\boldsymbol{\Sigma}$, the maximizer of $D(\boldsymbol{\Sigma})$ can be derived explicitly, making EM iterations quite fast.

For general, residual-error structures,

$$\text{Var}(\boldsymbol{\epsilon}_j) = \sigma_\epsilon^2 \boldsymbol{\Lambda}_j$$

where the subscript j merely represents that $\boldsymbol{\epsilon}_j$ and $\boldsymbol{\Lambda}_j$ vary in dimension in unbalanced data, the data are first transformed according to

$$\mathbf{y}_j^* = \widehat{\boldsymbol{\Lambda}}_j^{-1/2}\mathbf{y}_j; \qquad \mathbf{X}_j^* = \widehat{\boldsymbol{\Lambda}}_j^{-1/2}\mathbf{X}_j; \qquad \mathbf{Z}_j^* = \widehat{\boldsymbol{\Lambda}}_j^{-1/2}\mathbf{Z}_j;$$

and the likelihood-evaluation techniques described above are applied to $\mathbf{y}_j^*$, $\mathbf{X}_j^*$, and $\mathbf{Z}_j^*$ instead. The unique elements of $\boldsymbol{\Lambda}$, $\boldsymbol{\rho}$, are estimated along with the fixed effects and variance components. Because σ_ϵ^2 is always estimated and multiplies the entire $\boldsymbol{\Lambda}_j$ matrix, $\widehat{\boldsymbol{\rho}}$ is parameterized to take this into account.

In the presence of sampling weights, following Rabe-Hesketh and Skrondal (2006), the weighted log pseudolikelihood for a two-level model is given as

$$L(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \sigma_\epsilon^2) = \sum_{j=1}^{M} w_j \log \left[\int \exp\left\{ \sum_{i=1}^{n_j} w_{i|j} \log f_1(y_{ij}|\mathbf{u}_j, \boldsymbol{\beta}, \sigma_\epsilon^2) \right\} f_2(\mathbf{u}_j|\boldsymbol{\Sigma}) d\mathbf{u}_j \right] \tag{14}$$

where w_j is the inverse of the probability of selection for the jth cluster, $w_{i|j}$ is the inverse of the conditional probability of selection of individual i given the selection of cluster j, and $f_1(\cdot)$ and $f_2(\cdot)$ are the multivariate normal densities previously defined.

Weighted estimation is achieved through incorporating w_j and $w_{i|j}$ into the matrix decomposition methods detailed above to reflect replicated clusters for w_j and replicated observations within clusters for $w_{i|j}$. Because this estimation is based on replicated clusters and observations, frequency weights are handled similarly.

Rescaling of sampling weights can take one of three available forms:

Under `pwscale(size)`,

$$w_{i|j}^* = n_j w_{i|j}^* \left\{ \sum_{i=1}^{n_j} w_{i|j} \right\}^{-1}$$

Under `pwscale(effective)`,

$$w_{i|j}^* = w_{i|j}^* \left\{ \sum_{i=1}^{n_j} w_{i|j} \right\} \left\{ \sum_{i=1}^{n_j} w_{i|j}^2 \right\}^{-1}$$

Under both the above, w_j remains unchanged. For method `pwscale(gk)`, however, both weights are modified:

$$w_j^* = n_j^{-1} \sum_{i=1}^{n_j} w_{i|j} w_j; \qquad w_{i|j}^* = 1$$

Under ML estimation, robust standard errors are obtained in the usual way (see [P] **_robust**) with the one distinction being that in multilevel models, robust variances are, at a minimum, clustered at the highest level. This is because given the form of the log likelihood, scores aggregate at the top-level clusters. For a two-level model, scores are obtained as the partial derivatives of $L_j(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \sigma_\epsilon^2)$ with respect to $\{\boldsymbol{\beta}, \boldsymbol{\alpha}, \log(\sigma_\epsilon)\}$, where L_j is the log likelihood for cluster j and $L = \sum_{j=1}^M L_j$. Robust variances are not supported under REML estimation because the form of the log restricted likelihood does not lend itself to separation by highest-level clusters.

EM iterations always assume equal weighting and an independent, homoskedastic error structure. As such, with weighted data or when error structures are more complex, EM is used only to obtain starting values.

For extensions to models with three or more levels, see Bates and Pinheiro (1998) and Rabe-Hesketh and Skrondal (2006).

Charles Roy Henderson (1911–1989) was born in Iowa and grew up on the family farm. His education in animal husbandry, animal nutrition, and statistics at Iowa State was interspersed with jobs in the Iowa Extension Service, Ohio University, and the U.S. Army. After completing his PhD, Henderson joined the Animal Science faculty at Cornell. He developed and applied statistical methods in the improvement of farm livestock productivity through genetic selection, with particular focus on dairy cattle. His methods are general and have been used worldwide in livestock breeding and beyond agriculture. Henderson's work on variance components and best linear unbiased predictions has proved to be one of the main roots of current mixed-model methods.

Acknowledgments

We thank Badi Baltagi of the Department of Economics at Syracuse University and Ray Carroll of the Department of Statistics at Texas A&M University for each providing us with a dataset used in this entry.

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.

Baltagi, B. H., S. H. Song, and B. C. Jung. 2001. The unbalanced nested error component regression model. *Journal of Econometrics* 101: 357–381.

Bates, D. M., and J. C. Pinheiro. 1998. Computational methods for multilevel modelling. In *Technical Memorandum BL0112140-980226-01TM*. Murray Hill, NJ: Bell Labs, Lucent Technologies. http://stat.bell-labs.com/NLME/CompMulti.pdf.

Cameron, A. C., and P. K. Trivedi. 2010. *Microeconometrics Using Stata*. Rev. ed. College Station, TX: Stata Press.

Canette, I. 2011. Including covariates in crossed-effects models. The Stata Blog: Not Elsewhere Classified. http://blog.stata.com/2010/12/22/including-covariates-in-crossed-effects-models/.

Carle, A. C. 2009. Fitting multilevel models in complex survey data with design weights: Recommendations. *BMC Medical Research Methodology* 9: 49.

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

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B* 39: 1–38.

Diggle, P. J., P. J. Heagerty, K.-Y. Liang, and S. L. Zeger. 2002. *Analysis of Longitudinal Data*. 2nd ed. Oxford: Oxford University Press.

Fitzmaurice, G. M., N. M. Laird, and J. H. Ware. 2011. *Applied Longitudinal Analysis*. 2nd ed. Hoboken, NJ: Wiley.

Goldstein, H. 1986. Efficient statistical modelling of longitudinal data. *Annals of Human Biology* 13: 129–141.

Graubard, B. I., and E. L. Korn. 1996. Modelling the sampling design in the analysis of health surveys. *Statistical Methods in Medical Research* 5: 263–281.

Harville, D. A. 1977. Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association* 72: 320–338.

Henderson, C. R. 1953. Estimation of variance and covariance components. *Biometrics* 9: 226–252.

Hocking, R. R. 1985. *The Analysis of Linear Models*. Monterey, CA: Brooks/Cole.

Horton, N. J. 2011. Stata tip 95: Estimation of error covariances in a linear model. *Stata Journal* 11: 145–148.

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

LaMotte, L. R. 1973. Quadratic estimation of variance components. *Biometrics* 29: 311–330.

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.

Munnell, A. H. 1990. Why has productivity growth declined? Productivity and public investment. *New England Economic Review* Jan./Feb.: 3–22.

Nichols, A. 2007. Causal inference with observational data. *Stata Journal* 7: 507–541.

Palmer, T. M., C. M. Macdonald-Wallis, D. A. Lawlor, and K. Tilling. 2014. Estimating adjusted associations between random effects from multilevel models: The reffadjust package. *Stata Journal* 14: 119–140.

Pantazis, N., and G. Touloumi. 2010. Analyzing longitudinal data in the presence of informative drop-out: The jmre1 command. *Stata Journal* 10: 226–251.

Pfeffermann, D., C. J. Skinner, D. J. Holmes, H. Goldstein, and J. Rasbash. 1998. Weighting for unequal selection probabilities in multilevel models. *Journal of the Royal Statistical Society, Series B* 60: 23–40.

Pierson, R. A., and O. J. Ginther. 1987. Follicular population dynamics during the estrous cycle of the mare. *Animal Reproduction Science* 14: 219–231.

Pinheiro, J. C., and D. M. Bates. 2000. *Mixed-Effects Models in S and S-PLUS*. New York: Springer.

Prosser, R., J. Rasbash, and H. Goldstein. 1991. *ML3 Software for 3-Level Analysis: User's Guide for V. 2*. London: Institute of Education, University of London.

Rabe-Hesketh, S., and A. Skrondal. 2006. Multilevel modelling of complex survey data. *Journal of the Royal Statistical Society, Series A* 169: 805–827.

——. 2012. *Multilevel and Longitudinal Modeling Using Stata*. 3rd ed. College Station, TX: Stata Press.

Rao, C. R. 1973. *Linear Statistical Inference and Its Applications*. 2nd ed. New York: Wiley.

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

Ruppert, D., M. P. Wand, and R. J. Carroll. 2003. *Semiparametric Regression*. Cambridge: Cambridge University Press.

Schunck, R. 2013. Within and between estimates in random-effects models: Advantages and drawbacks of correlated random effects and hybrid models. *Stata Journal* 13: 65–76.

Searle, S. R. 1989. Obituary: Charles Roy Henderson 1911–1989. *Biometrics* 45: 1333–1335.

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

Thompson, W. A., Jr. 1962. The problem of negative estimates of variance components. *Annals of Mathematical Statistics* 33: 273–289.

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

West, B. T., K. B. Welch, and A. T. Galecki. 2007. *Linear Mixed Models: A Practical Guide Using Statistical Software*. Boca Raton, FL: Chapman & Hall/CRC.

Wiggins, V. L. 2011. Multilevel random effects in xtmixed and sem—the long and wide of it. The Stata Blog: Not Elsewhere Classified.
http://blog.stata.com/2011/09/28/multilevel-random-effects-in-xtmixed-and-sem-the-long-and-wide-of-it/.

Also see

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

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

[MI] **estimation** — Estimation commands for use with mi estimate

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

[XT] **xtrc** — Random-coefficients model

[XT] **xtreg** — Fixed-, between-, and random-effects and population-averaged linear models

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

Title

mixed postestimation — Postestimation tools for mixed

Description

The following postestimation commands are of special interest after `mixed`:

Command	Description
`estat group`	summarize the composition of the nested groups
`estat icc`	estimate intraclass correlations
`estat recovariance`	display the estimated random-effects covariance matrix (or matrices)
`estat wcorrelation`	display model-implied within-cluster correlations and standard deviations

The following standard postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat ic`	Akaike's and Schwarz's Bayesian information criteria (AIC and BIC)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

Special-interest postestimation commands

estat group reports the number of groups and minimum, average, and maximum group sizes for each level of the model. Model levels are identified by the corresponding group variable in the data. Because groups are treated as nested, the information in this summary may differ from what you would get if you used the tabulate command on each group variable individually.

estat icc displays the intraclass correlation for pairs of responses at each nested level of the model. Intraclass correlations are available for random-intercept models or for random-coefficient models conditional on random-effects covariates being equal to 0. They are not available for crossed-effects models or with residual error structures other than independent structures.

estat recovariance displays the estimated variance–covariance matrix of the random effects for each level in the model. Random effects can be either random intercepts, in which case the corresponding rows and columns of the matrix are labeled as _cons, or random coefficients, in which case the label is the name of the associated variable in the data.

estat wcorrelation displays the overall correlation matrix for a given cluster calculated on the basis of the design of the random effects and their assumed covariance and the correlation structure of the residuals. This allows for a comparison of different multilevel models in terms of the ultimate within-cluster correlation matrix that each model implies.

Syntax for predict

Syntax for obtaining BLUPs of random effects, or the BLUPs' standard errors

predict [*type*] { *stub** | *newvarlist* } [*if*] [*in*] , { reffects | reses }

[relevel(*levelvar*)]

Syntax for obtaining scores after ML estimation

predict [*type*] { *stub** | *newvarlist* } [*if*] [*in*] , scores

Syntax for obtaining other predictions

predict [*type*] *newvar* [*if*] [*in*] [, *statistic* relevel(*levelvar*)]

statistic	Description
Main	
xb	linear prediction for the fixed portion of the model only; the default
stdp	standard error of the fixed-portion linear prediction
fitted	fitted values, fixed-portion linear prediction plus contributions based on predicted random effects
residuals	residuals, response minus fitted values
* rstandard	standardized residuals

Unstarred statistics are available both in and out of sample; type predict ... if e(sample) ... if wanted only for the estimation sample. Starred statistics are calculated only for the estimation sample, even when if e(sample) is not specified.

Menu for predict

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

`xb`, the default, calculates the linear prediction $\mathbf{x}\boldsymbol{\beta}$ based on the estimated fixed effects (coefficients) in the model. This is equivalent to fixing all random effects in the model to their theoretical mean value of 0.

`stdp` calculates the standard error of the linear predictor $\mathbf{x}\boldsymbol{\beta}$.

`reffects` calculates best linear unbiased predictions (BLUPs) of the random effects. By default, BLUPs for all random effects in the model are calculated. However, if the `relevel(`*levelvar*`)` option is specified, then BLUPs for only level *levelvar* in the model are calculated. For example, if `classes` are nested within `schools`, then typing

```
. predict b*, reffects relevel(school)
```

would produce BLUPs at the school level. You must specify q new variables, where q is the number of random-effects terms in the model (or level). However, it is much easier to just specify *stub*`*` and let Stata name the variables *stub*`1`, *stub*`2`, ..., *stubq* for you.

`reses` calculates the standard errors of the best linear unbiased predictions (BLUPs) of the random effects. By default, standard errors for all BLUPs in the model are calculated. However, if the `relevel(`*levelvar*`)` option is specified, then standard errors for only level *levelvar* in the model are calculated; see the `reffects` option. You must specify q new variables, where q is the number of random-effects terms in the model (or level). However, it is much easier to just specify *stub*`*` and let Stata name the variables *stub*`1`, *stub*`2`, ..., *stubq* for you.

The `reffects` and `reses` options often generate multiple new variables at once. When this occurs, the random effects (or standard errors) contained in the generated variables correspond to the order in which the variance components are listed in the output of `mixed`. Still, examining the variable labels of the generated variables (with the `describe` command, for instance) can be useful in deciphering which variables correspond to which terms in the model.

`fitted` calculates fitted values, which are equal to the fixed-portion linear predictor *plus* contributions based on predicted random effects, or in mixed-model notation, $\mathbf{x}\boldsymbol{\beta}+\mathbf{Zu}$. By default, the fitted values take into account random effects from all levels in the model; however, if the `relevel(`*levelvar*`)` option is specified, then the fitted values are fit beginning with the topmost level down to and including level *levelvar*. For example, if `classes` are nested within `schools`, then typing

```
. predict yhat_school, fitted relevel(school)
```

would produce school-level predictions. That is, the predictions would incorporate school-specific random effects but not those for each class nested within each school.

`residuals` calculates residuals, equal to the responses minus fitted values. By default, the fitted values take into account random effects from all levels in the model; however, if the `relevel(`*levelvar*`)` option is specified, then the fitted values are fit beginning at the topmost level down to and including level *levelvar*.

`rstandard` calculates standardized residuals, equal to the residuals multiplied by the inverse square root of the estimated error covariance matrix.

`scores` calculates the parameter-level scores, one for each parameter in the model including regression coefficients and variance components. The score for a parameter is the first derivative of the log likelihood (or log pseudolikelihood) with respect to that parameter. One score per highest-level group is calculated, and it is placed on the last record within that group. Scores are calculated in the estimation metric as stored in `e(b)`.

`scores` is not available after restricted maximum-likelihood (REML) estimation.

`relevel(`*levelvar*`)` specifies the level in the model at which predictions involving random effects are to be obtained; see the options above for the specifics. *levelvar* is the name of the model level and is either the name of the variable describing the grouping at that level or is `_all`, a special designation for a group comprising all the estimation data.

Syntax for estat

Summarize the composition of the nested groups

```
estat group
```

Estimate intraclass correlations

```
estat icc [, level(#)]
```

Display the estimated random-effects covariance matrix (or matrices)

```
estat recovariance [, relevel(levelvar) correlation matlist_options]
```

Display model-implied within-cluster correlations and standard deviations

```
estat wcorrelation [, wcor_options]
```

wcor_options	Description
`at(`*at_spec*`)`	specify the cluster for which you want the correlation matrix; default is the first two-level cluster encountered in the data
`all`	display correlation matrix for all the data
`covariance`	display the covariance matrix instead of the correlation matrix
`list`	list the data corresponding to the correlation matrix
`nosort`	list the rows and columns of the correlation matrix in the order they were originally present in the data
`format(`*%fmt*`)`	set the display format; default is `format(%6.3f)`
matlist_options	style and formatting options that control how matrices are displayed

Menu for estat

Statistics > Postestimation > Reports and statistics

Option for estat icc

level(#) specifies the confidence level, as a percentage, for confidence intervals. The default is level(95) or as set by set level; see [U] **20.7 Specifying the width of confidence intervals**.

Options for estat recovariance

relevel(*levelvar*) specifies the level in the model for which the random-effects covariance matrix is to be displayed. By default, the covariance matrices for all levels in the model are displayed. *levelvar* is the name of the model level and is either the name of the variable describing the grouping at that level or is _all, a special designation for a group comprising all the estimation data.

correlation displays the covariance matrix as a correlation matrix.

matlist_options are style and formatting options that control how the matrix (or matrices) is displayed; see [P] **matlist** for a list of what is available.

Options for estat wcorrelation

at(*at_spec*) specifies the cluster of observations for which you want the within-cluster correlation matrix. *at_spec* is

relevel_var = *value* [, *relevel_var* = *value* ...]

For example, if you specify

```
. estat wcorrelation, at(school = 33)
```

you get the within-cluster correlation matrix for those observations in school 33. If you specify

```
. estat wcorrelation, at(school = 33 classroom = 4)
```

you get the correlation matrix for classroom 4 in school 33.

If at() is not specified, then you get the correlations for the first level-two cluster encountered in the data. This is usually what you want.

all specifies that you want the correlation matrix for all the data. This is not recommended unless you have a relatively small dataset or you enjoy seeing large $N \times N$ matrices. However, this can prove useful in some cases.

covariance specifies that the within-cluster covariance matrix be displayed instead of the default correlations and standard deviations.

list lists the model data for those observations depicted in the displayed correlation matrix. This option is useful if you have many random-effects design variables and you wish to see the represented values of these design variables.

nosort lists the rows and columns of the correlation matrix in the order that they were originally present in the data. Normally, estat wcorrelation will first sort the data according to level variables, by-group variables, and time variables to produce correlation matrices whose rows and columns follow a natural ordering. nosort suppresses this.

format(%*fmt*) sets the display format for the standard-deviation vector and correlation matrix. The default is format(%6.3f).

matlist_options are style and formatting options that control how the matrix (or matrices) is displayed; see [P] **matlist** for a list of what is available.

Remarks and examples

Various predictions, statistics, and diagnostic measures are available after fitting a mixed model using mixed. For the most part, calculation centers around obtaining BLUPs of the random effects. Random effects are not estimated when the model is fit but instead need to be predicted after estimation. Calculation of intraclass correlations, estimating the dependence between responses for different levels of nesting, may also be of interest.

▷ Example 1

In example 3 of [ME] **mixed**, we modeled the weights of 48 pigs measured on nine successive weeks as

$$\texttt{weight}_{ij} = \beta_0 + \beta_1\texttt{week}_{ij} + u_{0j} + u_{1j}\texttt{week}_{ij} + \epsilon_{ij} \qquad (1)$$

for $i = 1, \dots, 9$, $j = 1, \dots, 48$, $\epsilon_{ij} \sim N(0, \sigma^2_\epsilon)$, and u_{0j} and u_{1j} normally distributed with mean 0 and variance–covariance matrix

$$\boldsymbol{\Sigma} = \text{Var}\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} = \begin{bmatrix} \sigma^2_{u0} & \sigma_{01} \\ \sigma_{01} & \sigma^2_{u1} \end{bmatrix}$$

```
. use http://www.stata-press.com/data/r13/pig
(Longitudinal analysis of pig weights)
. mixed weight week || id: week, covariance(unstructured)
Performing EM optimization:
Performing gradient-based optimization:
Iteration 0:   log likelihood = -868.96185
Iteration 1:   log likelihood = -868.96185
Computing standard errors:
Mixed-effects ML regression                     Number of obs      =       432
Group variable: id                              Number of groups   =        48
                                                Obs per group: min =         9
                                                               avg =       9.0
                                                               max =         9
                                                Wald chi2(1)       =   4649.17
Log likelihood = -868.96185                     Prob > chi2        =    0.0000
```

weight	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
week	6.209896	.0910745	68.18	0.000	6.031393	6.388399
_cons	19.35561	.3996387	48.43	0.000	18.57234	20.13889

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
id: Unstructured				
var(week)	.3715251	.0812958	.2419532	.570486
var(_cons)	6.823363	1.566194	4.351297	10.69986
cov(week,_cons)	-.0984378	.2545767	-.5973991	.4005234
var(Residual)	1.596829	.123198	1.372735	1.857505

```
LR test vs. linear regression:       chi2(3) =   764.58   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

Rather than see the estimated variance components listed as variance and covariances as above, we can instead see them as correlations and standard deviations in matrix form; that is, we can see $\widehat{\boldsymbol{\Sigma}}$ as a correlation matrix:

```
. estat recovariance, correlation
Random-effects correlation matrix for level id

             |      week      _cons
-------------+---------------------
        week |         1
       _cons | -.0618257          1
```

We can use `estat wcorrelation` to display the within-cluster marginal standard deviations and correlations for one of the clusters.

```
. estat wcorrelation, format(%4.2g)
Standard deviations and correlations for id = 1:
Standard deviations:
         obs |     1     2     3     4     5     6     7     8     9
-------------+------------------------------------------------------
          sd |   2.9   3.1   3.3   3.7   4.1   4.5     5   5.5   6.1
Correlations:
         obs |     1     2     3     4     5     6     7     8     9
-------------+------------------------------------------------------
           1 |     1
           2 |    .8     1
           3 |   .77   .83     1
           4 |   .72   .81   .86     1
           5 |   .67   .78   .85   .89     1
           6 |   .63   .75   .83   .88   .91     1
           7 |   .59   .72   .81   .87   .91   .93     1
           8 |   .55   .69   .79   .86    .9   .93   .94     1
           9 |   .52   .66   .77   .85   .89   .92   .94   .95     1
```

Because within-cluster correlations can vary between clusters, `estat wcorrelation` by default displays the results for the first cluster. In this example, each cluster (pig) has the same number of observations, and the timings of measurements (`week`) are the same between clusters. Thus the within-cluster correlations are the same for all the clusters. In example 4, we fit a model where different clusters have different within-cluster correlations and show how to display these correlations.

We can also obtain BLUPs of the pig-level random effects (u_{0j} and u_{1j}). We need to specify the variables to be created in the order `u1 u0` because that is the order in which the corresponding variance components are listed in the output (`week _cons`). We obtain the predictions and list them for the first 10 pigs.

```
. predict u1 u0, reffects
. by id, sort: generate tolist = (_n==1)
. list id u0 u1 if id <=10 & tolist

     +----------------------------+
     | id          u0          u1 |
     |----------------------------|
  1. |  1    .2369444   -.3957636 |
 10. |  2   -1.584127     .510038 |
 19. |  3   -3.526551    .3200372 |
 28. |  4    1.964378   -.7719702 |
 37. |  5    1.299236   -.9241479 |
     |----------------------------|
 46. |  6   -1.147302   -.5448151 |
 55. |  7   -2.590529    .0394454 |
 64. |  8   -1.137067   -.1696566 |
 73. |  9   -3.189545   -.7365507 |
 82. | 10    1.160324    .0030772 |
     +----------------------------+
```

If you forget how to order your variables in `predict`, or if you use `predict` *stub**, remember that `predict` labels the generated variables for you to avoid confusion.

```
. describe u0 u1

              storage   display    value
variable name   type    format     label      variable label
---------------------------------------------------------------------------
u0              float   %9.0g                 BLUP r.e. for id: _cons
u1              float   %9.0g                 BLUP r.e. for id: week
```

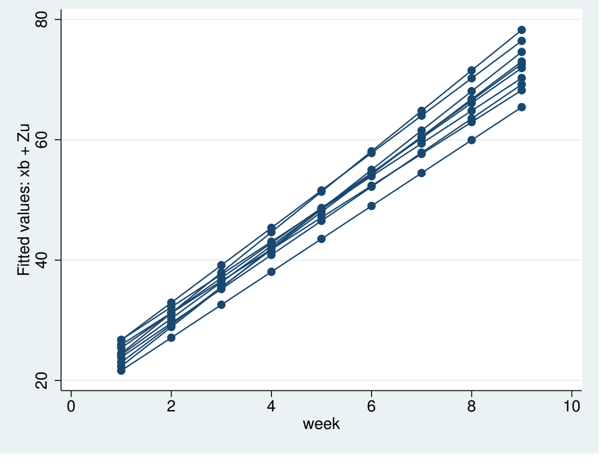
Examining (1), we see that within each pig, the successive weight measurements are modeled as simple linear regression with intercept $\beta_0 + u_{j0}$ and slope $\beta_1 + u_{j1}$. We can generate estimates of the pig-level intercepts and slopes with

```
. generate intercept = _b[_cons] + u0
. generate slope = _b[week] + u1
. list id intercept slope if id<=10 & tolist
```

	id	interc~t	slope
1.	1	19.59256	5.814132
10.	2	17.77149	6.719934
19.	3	15.82906	6.529933
28.	4	21.31999	5.437926
37.	5	20.65485	5.285748
46.	6	18.20831	5.665081
55.	7	16.76509	6.249341
64.	8	18.21855	6.040239
73.	9	16.16607	5.473345
82.	10	20.51594	6.212973

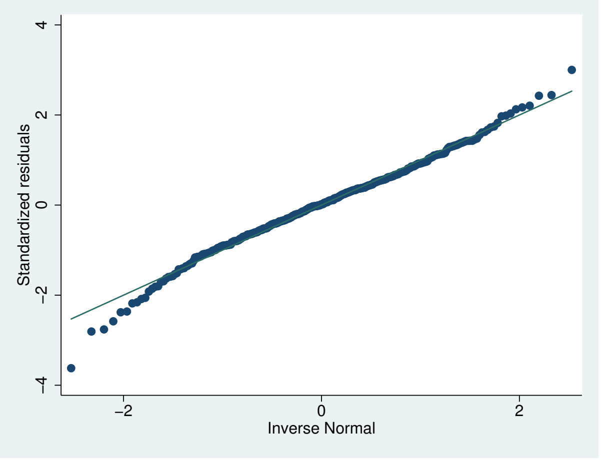
Thus we can plot estimated regression lines for each of the pigs. Equivalently, we can just plot the fitted values because they are based on both the fixed and the random effects:

```
. predict fitweight, fitted
. twoway connected fitweight week if id<=10, connect(L)
```



We can also generate standardized residuals and see whether they follow a standard normal distribution, as they should in any good-fitting model:

```
. predict rs, rstandard
. summarize rs
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
          rs |        432    1.01e-09    .8929356  -3.621446   3.000929
. qnorm rs
```



◁

▷ Example 2

Following Rabe-Hesketh and Skrondal (2012, chap. 2), we fit a two-level random-effects model for human peak-expiratory-flow rate. The subjects were each measured twice with the Mini-Wright peak-flow meter. It is of interest to determine how reliable the meter is as a measurement device. The intraclass correlation provides a measure of reliability. Formally, in a two-level random-effects model, the intraclass correlation corresponds to the correlation of measurements within the same individual and also to the proportion of variance explained by the individual random effect.

First, we fit the two-level model with mixed:

```
. use http://www.stata-press.com/data/r13/pefrate, clear
(Peak-expiratory-flow rate)
. mixed wm || id:
Performing EM optimization:
Performing gradient-based optimization:
Iteration 0:   log likelihood = -184.57839
Iteration 1:   log likelihood = -184.57839
Computing standard errors:
Mixed-effects ML regression                     Number of obs      =        34
Group variable: id                              Number of groups   =        17
                                                Obs per group: min =         2
                                                               avg =       2.0
                                                               max =         2
                                                Wald chi2(0)       =         .
Log likelihood = -184.57839                     Prob > chi2        =         .
```

wm	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
_cons	453.9118	26.18617	17.33	0.000	402.5878	505.2357

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
id: Identity				
var(_cons)	11458.94	3998.952	5782.176	22708.98
var(Residual)	396.441	135.9781	202.4039	776.4942

```
LR test vs. linear regression: chibar2(01) =    46.27 Prob >= chibar2 = 0.0000
```

Now we use `estat icc` to estimate the intraclass correlation:

```
. estat icc
Intraclass correlation
```

Level	ICC	Std. Err.	[95% Conf.	Interval]
id	.9665602	.0159495	.9165853	.9870185

This correlation is close to 1, indicating that the Mini-Wright peak-flow meter is reliable. But as noted by Rabe-Hesketh and Skrondal (2012), the reliability is not only a characteristic of the instrument but also of the between-subject variance. Here we see that the between-subject standard deviation, `sd(_cons)`, is much larger than the within-subject standard deviation, `sd(Residual)`.

In the presence of fixed-effects covariates, `estat icc` reports the residual intraclass correlation, the correlation between measurements conditional on the fixed-effects covariates. This is equivalent to the correlation of the model residuals.

In the presence of random-effects covariates, the intraclass correlation is no longer constant and depends on the values of the random-effects covariates. In this case, `estat icc` reports conditional intraclass correlations assuming 0 values for all random-effects covariates. For example, in a two-level model, this conditional correlation represents the correlation of the residuals for two measurements on the same subject, which both have random-effects covariates equal to 0. Similarly to the interpretation of intercept variances in random-coefficient models (Rabe-Hesketh and Skrondal 2012, chap. 4), interpretation of this conditional intraclass correlation relies on the usefulness of the 0 baseline values of random-effects covariates. For example, mean centering of the covariates is often used to make a 0 value a useful reference.

◁

▷ Example 3

In example 4 of [ME] mixed, we estimated a Cobb–Douglas production function with random intercepts at the region level and at the state-within-region level:

$$\mathbf{y}_{jk} = \mathbf{X}_{jk}\boldsymbol{\beta} + u_k^{(3)} + u_{jk}^{(2)} + \boldsymbol{\epsilon}_{jk}$$

```
. use http://www.stata-press.com/data/r13/productivity
(Public Capital Productivity)
. mixed gsp private emp hwy water other unemp || region: || state:
  (output omitted)
```

We can use `estat group` to see how the data are broken down by state and region:

```
. estat group
```

Group Variable	No. of Groups	Observations per Group Minimum	Average	Maximum
region	9	51	90.7	136
state	48	17	17.0	17

We are reminded that we have balanced productivity data for 17 years for each state.

We can use `predict, fitted` to get the fitted values

$$\widehat{\mathbf{y}}_{jk} = \mathbf{X}_{jk}\widehat{\boldsymbol{\beta}} + \widehat{u}_k^{(3)} + \widehat{u}_{jk}^{(2)}$$

but if we instead want fitted values at the region level, that is,

$$\widehat{\mathbf{y}}_{jk} = \mathbf{X}_{jk}\widehat{\boldsymbol{\beta}} + \widehat{u}_k^{(3)}$$

we need to use the `relevel()` option:

```
. predict gsp_region, fitted relevel(region)
. list gsp gsp_region in 1/10
```

	gsp	gsp_re~n
1.	10.25478	10.40529
2.	10.2879	10.42336
3.	10.35147	10.47343
4.	10.41721	10.52648
5.	10.42671	10.54947
6.	10.4224	10.53537
7.	10.4847	10.60781
8.	10.53111	10.64727
9.	10.59573	10.70503
10.	10.62082	10.72794

❑ Technical note

Out-of-sample predictions are permitted after `mixed`, but if these predictions involve BLUPs of random effects, the integrity of the estimation data must be preserved. If the estimation data have changed since the mixed model was fit, `predict` will be unable to obtain predicted random effects that are appropriate for the fitted model and will give an error. Thus to obtain out-of-sample predictions that contain random-effects terms, be sure that the data for these predictions are in observations that augment the estimation data.

❑

We can use `estat icc` to estimate residual intraclass correlations between productivity years in the same region and in the same state and region.

```
. estat icc
Residual intraclass correlation
```

Level	ICC	Std. Err.	[95% Conf.	Interval]
region	.159893	.127627	.0287143	.5506202
state|region	.8516265	.0301733	.7823466	.9016272

`estat icc` reports two intraclass correlations for this three-level nested model. The first is the level-3 intraclass correlation at the region level, the correlation between productivity years in the same region. The second is the level-2 intraclass correlation at the state-within-region level, the correlation between productivity years in the same state and region.

Conditional on the fixed-effects covariates, we find that annual productivity is only slightly correlated within the same region, but it is highly correlated within the same state and region. We estimate that state and region random effects compose approximately 85% of the total residual variance.

◁

▷ Example 4

In example 1, we fit a model where each cluster had the same model-implied within-cluster correlations. Here we fit a model where different clusters have different within-cluster correlations, and we show how to display them for different clusters. We use the Asian children weight data from example 6 of [ME] **mixed**.

```
. use http://www.stata-press.com/data/r13/childweight, clear
(Weight data on Asian children)

. mixed weight age || id: age, covariance(unstructured)

Performing EM optimization:

Performing gradient-based optimization:

Iteration 0:   log likelihood = -344.37065
Iteration 1:   log likelihood = -342.83973
Iteration 2:   log likelihood = -342.71863
Iteration 3:   log likelihood = -342.71777
Iteration 4:   log likelihood = -342.71777

Computing standard errors:

Mixed-effects ML regression                     Number of obs      =       198
Group variable: id                              Number of groups   =        68

                                                Obs per group: min =         1
                                                               avg =       2.9
                                                               max =         5

                                                Wald chi2(1)       =    755.27
Log likelihood = -342.71777                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
      weight |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   3.459671   .1258877    27.48   0.000     3.212936    3.706406
       _cons |   5.110496   .1494781    34.19   0.000     4.817524    5.403468
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
id: Unstructured             |
                    var(age) |   .2023917   .1242867      .0607405    .6743834
                  var(_cons) |   .0970274   .1107994      .0103484    .9097357
              cov(age,_cons) |    .140134   .0566901      .0290235    .2512445
-----------------------------+------------------------------------------------
               var(Residual) |   1.357922   .1650514      1.070074    1.723201
------------------------------------------------------------------------------
LR test vs. linear regression:       chi2(3) =    27.38   Prob > chi2 = 0.0000

Note: LR test is conservative and provided only for reference.
```

We use `estat wcorrelation` to display the within-cluster correlations for the first cluster.

```
. estat wcorrelation, list
Standard deviations and correlations for id = 45:
Standard deviations:
```

obs	1	2	3	4	5
sd	1.224	1.314	1.448	1.506	1.771

```
Correlations:
```

obs	1	2	3	4	5
1	1.000				
2	0.141	1.000			
3	0.181	0.274	1.000		
4	0.193	0.293	0.376	1.000	
5	0.230	0.348	0.447	0.477	1.000

```
Data:
```

	id	weight	age
1.	45	5.171	.136893
2.	45	10.86	.657084
3.	45	13.15	1.21834
4.	45	13.2	1.42916
5.	45	15.88	2.27242

We specified the `list` option to display the data associated with the cluster. The next cluster in the dataset has ID 258. To display the within-cluster correlations for this cluster, we specify the `at()` option.

```
. estat wcorrelation, at(id=258) list
Standard deviations and correlations for id = 258:
Standard deviations:
```

obs	1	2	3	4
sd	1.231	1.320	1.424	1.782

```
Correlations:
```

obs	1	2	3	4
1	1.000			
2	0.152	1.000		
3	0.186	0.270	1.000	
4	0.244	0.356	0.435	1.000

```
Data:
```

	id	weight	age
1.	258	5.3	.19165
2.	258	9.74	.687201
3.	258	9.98	1.12799
4.	258	11.34	2.30527

The within-cluster correlations for this model depend on age. The values for `age` in the two clusters are different, as are the corresponding within-cluster correlations.

◁

Stored results

estat icc stores the following in r():

Scalars	
r(icc#)	level-# intraclass correlation
r(se#)	standard errors of level-# intraclass correlation
r(level)	confidence level of confidence intervals
Macros	
r(label#)	label for level #
Matrices	
r(ci#)	vector of confidence intervals (lower and upper) for level-# intraclass correlation

For a G-level nested model, # can be any integer between 2 and G.

estat recovariance stores the following in r():

Scalars	
r(relevels)	number of levels
Matrices	
r(Cov#)	level-# random-effects covariance matrix
r(Corr#)	level-# random-effects correlation matrix (if option correlation was specified)

For a G-level nested model, # can be any integer between 2 and G.

estat wcorrelation stores the following in r():

Matrices	
r(sd)	standard deviations
r(Corr)	within-cluster correlation matrix
r(Cov)	within-cluster variance–covariance matrix
r(G)	variance–covariance matrix of random effects
r(Z)	model-based design matrix
r(R)	variance–covariance matrix of level-one errors

Methods and formulas

Methods and formulas are presented under the following headings:

Prediction
Intraclass correlations
Within-cluster covariance matrix

Prediction

Following the notation defined throughout [ME] **mixed**, BLUPs of random effects $\mathbf{u}$ are obtained as

$$\widetilde{\mathbf{u}} = \widetilde{\mathbf{G}}\mathbf{Z}'\widetilde{\mathbf{V}}^{-1}\left(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}\right)$$

where $\widetilde{\mathbf{G}}$ and $\widetilde{\mathbf{V}}$ are $\mathbf{G}$ and $\mathbf{V} = \mathbf{ZGZ}' + \sigma^2_\epsilon \mathbf{R}$ with maximum likelihood (ML) or REML estimates of the variance components plugged in. Standard errors for BLUPs are calculated based on the iterative technique of Bates and Pinheiro (1998, sec. 3.3) for estimating the BLUPs themselves. If estimation is done by REML, these standard errors account for uncertainty in the estimate of $\boldsymbol{\beta}$, while for ML the standard errors treat $\boldsymbol{\beta}$ as known. As such, standard errors of REML-based BLUPs will usually be larger.

Fitted values are given by $\mathbf{X}\widehat{\boldsymbol{\beta}} + \mathbf{Z}\widetilde{\mathbf{u}}$, residuals as $\widehat{\boldsymbol{\epsilon}} = \mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}} - \mathbf{Z}\widetilde{\mathbf{u}}$, and standardized residuals as

$$\widehat{\boldsymbol{\epsilon}}_* = \widehat{\sigma}_\epsilon^{-1}\widehat{\mathbf{R}}^{-1/2}\widehat{\boldsymbol{\epsilon}}$$

If the `relevel(`*levelvar*`)` option is specified, fitted values, residuals, and standardized residuals consider only those random-effects terms up to and including level *levelvar* in the model.

For details concerning the calculation of scores, see *Methods and formulas* in [ME] **mixed**.

Intraclass correlations

Consider a simple, two-level random-intercept model,

$$y_{ij} = \beta + u_j^{(2)} + \epsilon_{ij}^{(1)}$$

for measurements $i = 1, \dots, n_j$ and level-2 groups $j = 1, \dots, M$, where y_{ij} is a response, β is an unknown fixed intercept, u_j is a level-2 random intercept, and $\epsilon_{ij}^{(1)}$ is a level-1 error term. Errors are assumed to be normally distributed with mean 0 and variance σ_1^2; random intercepts are assumed to be normally distributed with mean 0 and variance σ_2^2 and to be independent of error terms.

The intraclass correlation for this model is

$$\rho = \mathrm{Corr}(y_{ij}, y_{i'j}) = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

It corresponds to the correlation between measurements i and i' from the same group j.

Now consider a three-level nested random-intercept model,

$$y_{ijk} = \beta + u_{jk}^{(2)} + u_k^{(3)} + \epsilon_{ijk}^{(1)}$$

for measurements $i = 1, \dots, n_{jk}$ and level-2 groups $j = 1, \dots, M_{1k}$ nested within level-3 groups $k = 1, \dots, M_2$. Here $u_{jk}^{(2)}$ is a level-2 random intercept, $u_k^{(3)}$ is a level-3 random intercept, and $\epsilon_{ijk}^{(1)}$ is a level-1 error term. The error terms and random intercepts are assumed to be normally distributed with mean 0 and variances σ_1^2, σ_2^2, and σ_3^2, respectively, and to be mutually independent.

We can consider two types of intraclass correlations for this model. We will refer to them as level-2 and level-3 intraclass correlations. The level-3 intraclass correlation is

$$\rho^{(3)} = \mathrm{Corr}(y_{ijk}, y_{i'j'k}) = \frac{\sigma_3^2}{\sigma_1^2 + \sigma_2^2 + \sigma_3^2}$$

This is the correlation between measurements i and i' from the same level-3 group k and from different level-2 groups j and j'.

The level-2 intraclass correlation is

$$\rho^{(2)} = \mathrm{Corr}(y_{ijk}, y_{i'jk}) = \frac{\sigma_2^2 + \sigma_3^2}{\sigma_1^2 + \sigma_2^2 + \sigma_3^2}$$

This is the correlation between measurements i and i' from the same level-3 group k and level-2 group j. (Note that level-1 intraclass correlation is undefined.)

More generally, for a G-level nested random-intercept model, the g-level intraclass correlation is defined as

$$\rho^{(g)} = \frac{\sum_{l=g}^{G} \sigma_l^2}{\sum_{l=1}^{G} \sigma_l^2}$$

The above formulas also apply in the presence of fixed-effects covariates $\mathbf{X}$ in a random-effects model. In this case, intraclass correlations are conditional on fixed-effects covariates and are referred to as residual intraclass correlations. `estat icc` also uses the same formulas to compute intraclass correlations for random-coefficient models, assuming 0 baseline values for the random-effects covariates, and labels them as conditional intraclass correlations. The above formulas assume independent residual structures.

Intraclass correlations are estimated using the delta method and will always fall in (0,1) because variance components are nonnegative. To accommodate the range of an intraclass correlation, we use the logit transformation to obtain confidence intervals.

Let $\widehat{\rho}^{(g)}$ be a point estimate of the intraclass correlation and $\widehat{\mathrm{SE}}(\widehat{\rho}^{(g)})$ be its standard error. The $(1-\alpha)\times 100\%$ confidence interval for $\mathrm{logit}(\rho^{(g)})$ is

$$\mathrm{logit}(\widehat{\rho}^{(g)}) \pm z_{\alpha/2} \frac{\widehat{\mathrm{SE}}(\widehat{\rho}^{(g)})}{\widehat{\rho}^{(g)}(1-\widehat{\rho}^{(g)})}$$

where $z_{\alpha/2}$ is the $1-\alpha/2$ quantile of the standard normal distribution and $\mathrm{logit}(x) = \ln\{x/(1-x)\}$. Let k_u be the upper endpoint of this interval, and let k_l be the lower. The $(1-\alpha)\times 100\%$ confidence interval for $\rho^{(g)}$ is then given by

$$\left(\frac{1}{1+e^{-k_l}}, \frac{1}{1+e^{-k_u}} \right)$$

Within-cluster covariance matrix

A two-level linear mixed model of the form

$$\mathbf{y}_j = \mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j + \boldsymbol{\epsilon}_j$$

implies the marginal model

$$\mathbf{y}_j = \mathbf{X}_j\boldsymbol{\beta} + \boldsymbol{\epsilon}_j^*$$

where $\boldsymbol{\epsilon}_j^* \sim N(\mathbf{0}, \mathbf{V}_j)$, $\mathbf{V}_j = \mathbf{Z}_j\mathbf{G}\mathbf{Z}_j' + \mathbf{R}$. In a marginal model, the random part is described in terms of the marginal or total residuals $\boldsymbol{\epsilon}_j^*$, and $\mathbf{V}_j$ is the covariance structure of these residuals.

`estat wcorrelation` calculates the marginal covariance matrix $\widetilde{\mathbf{V}}_j$ for cluster j and by default displays the results in terms of standard deviations and correlations. This allows for a comparison of different multilevel models in terms of the ultimate within-cluster correlation matrix that each model implies.

Calculation of the marginal covariance matrix extends naturally to higher-level models; see, for example, chapter 4.8 in West, Welch, and Galecki (2007).

References

Bates, D. M., and J. C. Pinheiro. 1998. Computational methods for multilevel modelling. In *Technical Memorandum BL0112140-980226-01TM*. Murray Hill, NJ: Bell Labs, Lucent Technologies. http://stat.bell-labs.com/NLME/CompMulti.pdf.

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

West, B. T., K. B. Welch, and A. T. Galecki. 2007. *Linear Mixed Models: A Practical Guide Using Statistical Software*. Boca Raton, FL: Chapman & Hall/CRC.

Also see

[ME] **mixed** — Multilevel mixed-effects linear regression

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

Glossary

BLUPs. BLUPs are best linear unbiased predictions of either random effects or linear combinations of random effects. In linear models containing random effects, these effects are not estimated directly but instead are integrated out of the estimation. Once the fixed effects and variance components have been estimated, you can use these estimates to predict group-specific random effects. These predictions are called BLUPs because they are unbiased and have minimal mean squared errors among all linear functions of the response.

canonical link. Corresponding to each family of distributions in a generalized linear model (GLM) is a canonical link function for which there is a sufficient statistic with the same dimension as the number of parameters in the linear predictor. The use of canonical link functions provides the GLM with desirable statistical properties, especially when the sample size is small.

conditional overdispersion. In a negative binomial mixed-effects model, conditional overdispersion is overdispersion conditional on random effects. Also see *overdispersion*.

covariance structure. In a mixed-effects model, covariance structure refers to the variance–covariance structure of the random effects.

crossed-effects model. A crossed-effects model is a mixed-effects model in which the levels of random effects are not nested. A simple crossed-effects model for cross-sectional time-series data would contain a random effect to control for panel-specific variation and a second random effect to control for time-specific random variation. Rather than being nested within panel, in this model a random effect due to a given time is the same for all panels.

crossed-random effects. See *crossed-effects model*.

EB. See *empirical Bayes*.

empirical Bayes. In generalized linear mixed-effects models, empirical Bayes refers to the method of prediction of the random effects after the model parameters have been estimated. The empirical Bayes method uses Bayesian principles to obtain the posterior distribution of the random effects, but instead of assuming a prior distribution for the model parameters, the parameters are treated as given.

empirical Bayes mean. See *posterior mean*.

empirical Bayes mode. See *posterior mode*.

fixed effects. In the context of multilevel mixed-effects models, fixed effects represent effects that are constant for all groups at any level of nesting. In the ANOVA literature, fixed effects represent the levels of a factor for which the inference is restricted to only the specific levels observed in the study. See also *fixed-effects model* in [XT] **Glossary**.

Gauss–Hermite quadrature. In the context of generalized linear mixed models, Gauss–Hermite quadrature is a method of approximating the integral used in the calculation of the log likelihood. The quadrature locations and weights for individual clusters are fixed during the optimization process.

generalized linear mixed-effects model. A generalized linear mixed-effect model is an extension of a generalized linear model allowing for the inclusion of random deviations (effects).

generalized linear model. The generalized linear model is an estimation framework in which the user specifies a distributional family for the dependent variable and a link function that relates the dependent variable to a linear combination of the regressors. The distribution must be a member of the exponential family of distributions. The generalized linear model encompasses many common models, including linear, probit, and Poisson regression.

GHQ. See *Gauss–Hermite quadrature*.

GLM. See *generalized linear model*.

GLME model. See *generalized linear mixed-effects model*.

GLMM. Generalized linear mixed model. See *generalized linear mixed-effects model*.

hierarchical model. A hierarchical model is one in which successively more narrowly defined groups are nested within larger groups. For example, in a hierarchical model, patients may be nested within doctors who are in turn nested within the hospital at which they practice.

intraclass correlation. In the context of mixed-effects models, intraclass correlation refers to the correlation for pairs of responses at each nested level of the model.

Laplacian approximation. Laplacian approximation is a technique used to approximate definite integrals without resorting to quadrature methods. In the context of mixed-effects models, Laplacian approximation is as a rule faster than quadrature methods at the cost of producing biased parameter estimates of variance components.

linear mixed model. See *linear mixed-effects model*.

linear mixed-effects model. A linear mixed-effects model is an extension of a linear model allowing for the inclusion of random deviations (effects).

link function. In a generalized linear model or a generalized linear mixed-effects model, the link function relates a linear combination of predictors to the expected value of the dependent variable. In a linear regression model, the link function is simply the identity function.

LME model. See *linear mixed-effects model*.

MCAGH. See *mode-curvature adaptive Gauss–Hermite quadrature*.

mean–variance adaptive Gauss–Hermite quadrature. In the context of generalized linear mixed models, mean–variance adaptive Gauss–Hermite quadrature is a method of approximating the integral used in the calculation of the log likelihood. The quadrature locations and weights for individual clusters are updated during the optimization process by using the posterior mean and the posterior standard deviation.

mixed model. See *mixed-effects model*.

mixed-effects model. A mixed-effects model contains both fixed and random effects. The fixed effects are estimated directly, whereas the random effects are summarized according to their (co)variances. Mixed-effects models are used primarily to perform estimation and inference on the regression coefficients in the presence of complicated within-subject correlation structures induced by multiple levels of grouping.

mode-curvature adaptive Gauss–Hermite quadrature. In the context of generalized linear mixed models, mode-curvature adaptive Gauss–Hermite quadrature is a method of approximating the integral used in the calculation of the log likelihood. The quadrature locations and weights for individual clusters are updated during the optimization process by using the posterior mode and the standard deviation of the normal density that approximates the log posterior at the mode.

MVAGH. See *mean–variance adaptive Gauss–Hermite quadrature*.

nested random effects. In the context of mixed-effects models, nested random effects refer to the nested grouping factors for the random effects. For example, we may have data on students who are nested in classes that are nested in schools.

one-level model. A one-level model has no multilevel structure and no random effects. Linear regression is a one-level model.

overdispersion. In count-data models, overdispersion occurs when there is more variation in the data than would be expected if the process were Poisson.

posterior mean. In generalized linear mixed-effects models, posterior mean refer to the predictions of random effects based on the mean of the posterior distribution.

posterior mode. In generalized linear mixed-effects models, posterior mode refer to the predictions of random effects based on the mode of the posterior distribution.

QR decomposition. QR decomposition is an orthogonal-triangular decomposition of an augmented data matrix that speeds up the calculation of the log likelihood; see *Methods and formulas* in [ME] **mixed** for more details.

quadrature. Quadrature is a set of numerical methods to evaluate a definite integral.

random coefficient. In the context of mixed-effects models, a random coefficient is a counterpart to a slope in the fixed-effects equation. You can think of a random coefficient as a randomly varying slope at a specific level of nesting.

random effects. In the context of mixed-effects models, random effects represent effects that may vary from group to group at any level of nesting. In the ANOVA literature, random effects represent the levels of a factor for which the inference can be generalized to the underlying population represented by the levels observed in the study. See also *random-effects model* in [XT] **Glossary**.

random intercept. In the context of mixed-effects models, a random intercept is a counterpart to the intercept in the fixed-effects equation. You can think of a random intercept as a randomly varying intercept at a specific level of nesting.

REML. See *restricted maximum likelihood*.

restricted maximum likelihood. Restricted maximum likelihood is a method of fitting linear mixed-effects models that involves transforming out the fixed effects to focus solely on variance–component estimation.

three-level model. A three-level mixed-effects model has one level of observations and two levels of grouping. Suppose that you have a dataset consisting of patients overseen by doctors at hospitals, and each doctor practices at one hospital. Then a three-level model would contain a set of random effects to control for hospital-specific variation, a second set of random effects to control for doctor-specific random variation within a hospital, and a random-error term to control for patients' random variation.

two-level model. A two-level mixed-effects model has one level of observations and one level of grouping. Suppose that you have a panel dataset consisting of patients at hospitals; a two-level model would contain a set of random effects at the hospital level (the second level) to control for hospital-specific random variation and a random-error term at the observation level (the first level) to control for within-hospital variation.

variance components. In a mixed-effects model, the variance components refer to the variances and covariances of the various random effects.

Subject and author index

This is the subject and author index for the *Multilevel Mixed-Effects Reference Manual*. Readers interested in topics other than multilevel mixed-effects should see the combined subject index (and the combined author index) in the *Glossary and Index*.

N

O

P