Title

me — Introduction to multilevel mixed-effects models

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:

1

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 $\sigma_\epsilon^2 \mathbf{R}$.

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

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

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_ϵ^2 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,

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

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.

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

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.

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

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

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

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 β 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_p^2 and a χ_{p+1}^2 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

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

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_3^2. 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 `week`s. 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
```

variable name	storage type	display format	value 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.
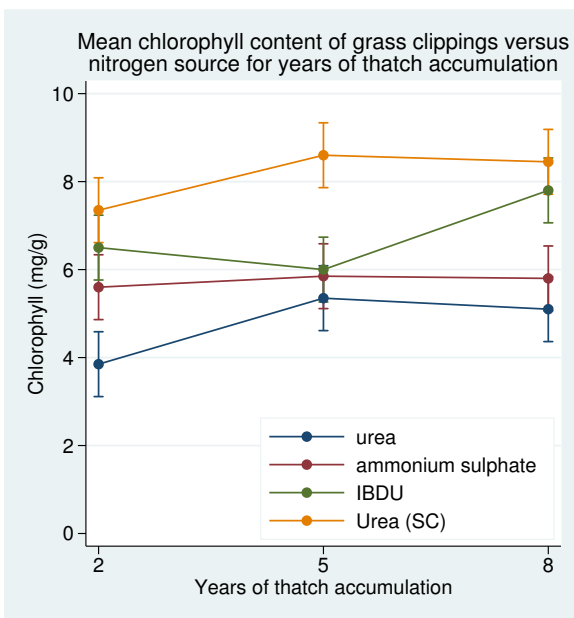
```
. margins thatch#nitrogen
Predictive margins                              Number of obs   =          24
Expression   : Linear prediction, fixed portion, predict()
```

	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)

variable name	storage type	display format	value 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

Binomial variable: napplied

Group variable: department

Number of obs	=	46
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"))
```
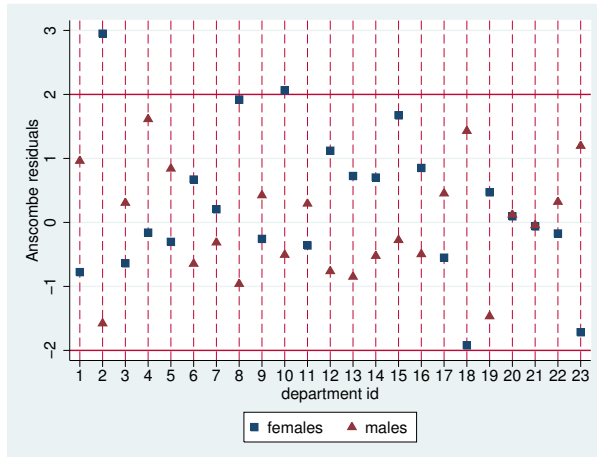

Anscombe residuals are constructed to be approximately normally distributed, thus residuals that are above two in absolute value are usually considered outliers. In the graph above, the residual for female admissions in department 2 is a clear outlier, suggesting a poor fit for that particular observation; 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")
```


◁
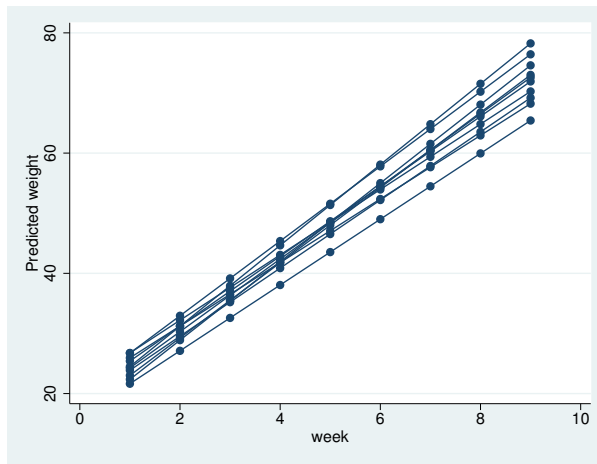
▷ Example 5: Blocked-diagonal covariance structures

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

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)   =   124.20
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

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**