Title

stata.com

me — Introduction to multilevel mixed-effects models

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 models

Mixed-effects censored regression

`metobit`	Multilevel mixed-effects tobit regression
`meintreg`	Multilevel mixed-effects interval regression

Mixed-effects binary regression

`melogit`	Multilevel mixed-effects logistic regression
`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
`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**.

Mixed-effects survival model

`mestreg`	Multilevel mixed-effects parametric survival models

Nonlinear mixed-effects regression

`menl`	Nonlinear mixed-effects regression

Postestimation tools specific to mixed-effects commands

`estat df`	Calculate and display degrees of freedom for fixed effects
`estat group`	Summarize the composition of the nested groups
`estat icc`	Estimate intraclass correlations
`estat recovariance`	Display the estimated random-effects covariance matrices
`estat sd`	Display variance components as standard deviations and correlations
`estat wcorrelation`	Display within-cluster correlations and standard deviations

Quick start

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:
```

Nonlinear mixed-effects models

Nonlinear mixed-effects regression of y on x1 and x2 with parameters {b0}, {b1}, {b2}, and {b3} and random intercepts U0 by id

```
menl y = ({b0}+{b1}*x1+{U0[id]})/(1+exp(-(x2-{b2})/{b3}))
```

Same as above, but using the more efficient specification of the linear combination

```
menl y = ({lc: x1 U0[id]})/(1+exp(-(x2-{b2})/{b3}))
```

Same as above, but using define() to specify the linear combination

```
menl y = {lc:}/(1+exp(-(x2-{b2})/{b3})), define(lc: x1 U0[id])
```

Include a random slope on continuous variable x1 in the define() option, and allow correlation between random slopes U1 and intercepts U0

```
menl y = {lc:}/(1+exp(-(x2-{b2})/{b3})),        ///
    define(lc: x1 U0[id] c.x1#U1[id]) covariance(U0 U1, unstructured)
```

Specify a heteroskedastic within-subject error structure that varies as a power of predicted mean values _yhat

```
menl y = {lc:}/(1+exp(-(x2-{b2})/{b3})),        ///
    define(lc: x1 U0[id] c.x1#U1[id])           ///
    covariance(U0 U1, unstructured) resvariance(power _yhat)
```

Three-level nonlinear regression of y on x1 with random intercepts W0 and slopes W1 on continuous x1 by lev2 and with random intercepts S0 and slopes S1 on x1 by lev3, with lev2 nested within lev3, using unstructured covariance for W0 and W1 and exchangeable covariance for S0 and S1

```
menl y = {phi1:}+{b1}*cos({b2}*x1),              ///
    define(phi1: x1 W0[lev3] S0[lev3>lev2]       ///
             c.x1#(W1[lev3] S1[lev3>lev2]))      ///
    covariance(W0 W1, unstructured) covariance(S0 S1, exchangeable)
```

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 in a crossed-effects model

levelvar`: R.`*varname*

levelvar is a variable identifying the group structure for the random effects at that level or is `_all` representing one group comprising all observations.

Nonlinear mixed-effects models

`menl` *depvar* `=` <*menlexpr*> [*if*] [*in*] [`,` *options*]

<*menlexpr*> defines a nonlinear regression function as a substitutable expression that contains model parameters and random effects specified in braces `{}`, as in `exp({b}+{U[id]})`; see *Random-effects substitutable expressions* in [ME] **menl** for details.

Remarks and examples

stata.com

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 (2022) provide more examples of linear and generalized linear mixed-effects models. Nonlinear mixed-effects (NLME) models are popular in, for example, population pharmacokinetics, bioassays, and studies of biological and agricultural growth processes.

For a comprehensive treatment of mixed-effects models, see, for example, Searle, Casella, and McCulloch (1992); Verbeke and Molenberghs (2000); Raudenbush and Bryk (2002); Hedeker and Gibbons (2006); McCulloch, Searle, and Neuhaus (2008); and Rabe-Hesketh and Skrondal (2022). For NLME models, see, for example, Davidian and Giltinan (1995); Vonesh and Chinchilli (1997); Demidenko (2013); Pinheiro and Bates (2000); and Davidian and Giltinan (2003).

Shayle R. Searle (1928–2013) was born in New Zealand. He obtained his PhD in animal breeding from Cornell University in 1958, with a minor in statistics. Prior to moving to New York, he worked as a research statistician for the New Zealand Dairy Board, which provided the data that he would analyze for his thesis. After completing his doctoral degree, he worked as a research associate and published several articles. He later returned to his post as a statistician in New Zealand, a position which would have a lasting influence on his career.

Through his analysis of dairy production data, Searle made advancements in estimation methods for unbalanced data and published a book on this topic. He later returned to Cornell University, teaching courses in matrix algebra, linear regression models, and estimation of variance components. Searle was one of the first few statisticians to use matrices in statistics, and he wrote a couple of books applying matrix algebra to economics and statistics. In 2001, he published a book on mixed models, which proved to be a significant contribution considering that not many statisticians were well acquainted with random effects in the 1950s. His contributions did not go unnoticed: he was awarded the Alexander von Humboldt U.S. Senior Scientist Award and was elected a fellow of the Royal Statistical Society and of the American Statistical Association.

George Casella (1951–2012) was born in Bronx, New York. After obtaining a PhD in statistics from Purdue University, he went on to join the faculty at Rutgers University, and later Cornell University, where he taught for 19 years, and the University of Florida. He published on topics such as confidence estimation, Bayesian analysis, and empirical Bayes methods. In general, his work was motivated by applications to science, and in particular, his work on variable selection and clustering was motivated by genetics. Casella coauthored a book with Roger Berger that introduced many graduate students to mathematical statistics. He coauthored another book with Christian P. Robert on Monte Carlo methods. In addition to his own published work, Casella was an editor for three journals: *Statistical Science*, *Journal of the American Statistical Society*, and *Journal of the Royal Statistical Society*.

Casella's many contributions are reflected in his election to fellowship on behalf of four different associations and institutes and being made a foreign member of the Spanish Royal Academy of Sciences. He acquired the Spanish language during a year he spent in Spain for sabbatical and even gave talks on Monte Carlo methods in Spanish. Aside from his academic accomplishments, Casella completed 13 marathons and spent time as a volunteer firefighter.

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`, except `menl`. `menl` models the mean function nonlinearly and thus has a different syntax; see [ME] **menl**.

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] **melogit** 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` 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`—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

Linear mixed-effects (LME) models for continuous responses 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 error-covariance parameters that include the overall error variance σ^2_ϵ and the 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 within-cluster 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),

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

where $\boldsymbol{\Lambda}$ denotes the variance matrix of the level-1 errors 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.

By our convention on counting and ordering model levels, (2) is a two-level model, with extensions to three, four, or any number of levels. The observation y_{ij} is for individual i within cluster j, and the individuals compose the first level, whereas the clusters compose the second level of the model. In a 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).

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,

$$g\{E(\mathbf{y}|\mathbf{X},\mathbf{u})\} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u}, \qquad \mathbf{y} \sim F \tag{3}$$

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

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

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

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

or mixed-effects Poisson regression.

For the random portion of (3), $\mathbf{Z}\mathbf{u}$, 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

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

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

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

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, allows for modeling of the structure of the within-cluster 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.

Survival mixed-effects models

Parametric survival mixed-effects models use a trivariate response variable (t_0, t, d), where each response corresponds to a period under observation $(t_0, t]$ and results in either failure ($d = 1$) or right-censoring ($d = 0$) at time t. See [ST] **streg** for background information on parametric survival models. Two often-used models for adjusting survivor functions for the effects of covariates are the accelerated failure-time (AFT) model and the multiplicative or proportional hazards (PH) model.

In the AFT parameterization, the natural logarithm of the survival time, $\log t$, is expressed as a linear function of the covariates. When we incorporate random effects, this yields the model

$$\log(\mathbf{t}_j) = \mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j + \mathbf{v}_j$$

where $\log(\cdot)$ is an elementwise function, and $\mathbf{v}_j$ is a vector of observation-level errors. The distributional form of the error term determines the regression model.

In the PH model, the covariates have a multiplicative effect on the hazard function

$$h(\mathbf{t}_j) = h_0(\mathbf{t}_j)\exp(\mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j)$$

where all the functions are elementwise, and $h_0(\cdot)$ is a baseline hazard function. The functional form of $h_0(\cdot)$ determines the regression model.

In Stata, you can use `mestreg` to fit multilevel mixed-effects parametric survival models for the following distributions and parameterizations.

Distribution	Parameterization
`exponential`	PH, AFT
`loglogistic`	AFT
`weibull`	PH, AFT
`lognormal`	AFT
`gamma`	AFT

`mestreg` is suitable only for data that have been set using the `stset` command. By using `stset` on your data, you define the variables `_t0`, `_t`, and `_d`, which serve as the trivariate response. See [ME] **mestreg** for more details about the command. Various predictions, statistics, and diagnostic measures are available after fitting a mixed-effects survival model with `mestreg`; see [ME] **mestreg postestimation** for a detailed discussion and examples.

Nonlinear mixed-effects models

NLME models are models containing both fixed effects and random effects where some of, or all, the fixed and random effects enter the model nonlinearly. They can be viewed as a generalization of LME models, in which the conditional mean of the outcome given the random effects is a nonlinear function of the coefficients and random effects. Alternatively, they can be considered as an extension of nonlinear regression models for independent data (see [R] **nl**), in which coefficients may incorporate random effects, allowing them to vary across different levels of hierarchy and thus inducing correlation within observations at the same level.

Using the notation from *Linear mixed-effects models* for LME models for clustered data, we can write an NLME model as

$$\mathbf{y}_j = \boldsymbol{\mu}\left(\mathbf{A}_j, \boldsymbol{\beta}, \mathbf{u}_j\right) + \boldsymbol{\epsilon}_j$$

where $\boldsymbol{\mu}(\cdot)$ is a real-valued vector function and $\mathbf{A}_j$ is an $n_j \times l$ matrix of covariates for the jth cluster, which includes both within-subject and between-subject covariates. Do not be surprised to see the $\mathbf{A}_j$ matrix here instead of the more familiar fixed-effects and random-effects design matrices $\mathbf{X}_j$ and $\mathbf{Z}_j$ from previous sections. Because both covariates and parameters can enter the model nonlinearly in NLME, we cannot express the regression function as a function containing the linear term $\mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j$ as we can for LME and GLME models. The distributional assumptions on $\mathbf{u}_j$'s and $\boldsymbol{\epsilon}_j$'s are the same as for the LME models.

Parameters of NLME models often have scientifically meaningful interpretations, and research questions are formed based on them. To allow parameters to reflect phenomena of interest, NLME models are often formulated by using a multistage formulation; see *Alternative mixed-effects model specification* below for examples.

We can formulate our previous NLME model as a two-stage hierarchical model:

$$\text{Stage 1: Individual-level model } y_{ij} = m\left(\mathbf{x}_{ij}^{w},\, \boldsymbol{\phi}_j\right) + \epsilon_{ij}, \quad i = 1, \ldots, n_j$$
$$\text{Stage 2: Group-level model } \boldsymbol{\phi}_j = \boldsymbol{d}\left(\mathbf{x}_j^{b},\, \boldsymbol{\beta},\, \mathbf{u}_j\right), \quad j = 1, \ldots, M$$

In stage 1, we model the response by using a function $m(\cdot)$, which describes within-subject behavior. This function depends on subject-specific parameters $\boldsymbol{\phi}_j$'s, which have a natural physical interpretation, and a vector of within-subject covariates $\mathbf{x}_{ij}^{w}$. In stage 2, we use a known vector-valued function $\boldsymbol{d}(\cdot)$ to model between-subject behavior, that is, to model $\boldsymbol{\phi}_j$'s and to explain how they vary across subjects. The $\boldsymbol{d}(\cdot)$ function incorporates random effects and, optionally, a vector of between-subject covariates $\mathbf{x}_j^{b}$. The general idea is to specify a common functional form for each subject in stage 1 and then allow some parameters to vary randomly across subjects in stage 2.

You can use the `menl` command to fit NLME models to continuous outcomes; see [ME] **menl**. `menl` supports both the single-equation and multistage model formulations. It supports different covariance structures for random effects and can model heteroskedasticity and correlations within lowest-level groups. Various predictions, statistics, and diagnostic measures are available after fitting an NLME model; see [ME] **menl postestimation**.

For an introductory example, see *Nonlinear models*.

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. This formulation is common for NLME models; see *Nonlinear mixed-effects models*.

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

$$y_{ij} = \beta_0 + \beta_1 x_{ij} + u_j + \epsilon_{ij} \tag{4}$$

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

$$\begin{aligned} y_{ij} &= \gamma_{0j} + \beta_1 x_{ij} + \epsilon_{ij} \\ \gamma_{0j} &= \beta_{00} + u_{0j} \end{aligned} \tag{5}$$

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 by using, for example, `mixed`, we must translate the multiple-equation notation into a single-equation form. We substitute the second equation into the first one and rearrange terms.

$$\begin{aligned} y_{ij} &= \beta_{00} + u_{0j} + \beta_1 x_{ij} + \epsilon_{ij} \\ &= \beta_{00} + \beta_1 x_{ij} + u_{0j} + \epsilon_{ij} \end{aligned} \tag{6}$$

Note that model (6) is the same as model (4) with $\beta_{00} \equiv \beta_0$ and $u_{0j} \equiv u_j$. Thus the syntax for our generic random-intercept model is

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

where `id` is the variable designating the clusters.

We can extend model (5) to include a random slope. We do so by specifying an additional equation for the slope on x_{ij}.

$$\begin{aligned} y_{ij} &= \gamma_{0j} + \gamma_{1j}x_{ij} + \epsilon_{ij} \\ \gamma_{0j} &= \beta_{00} + u_{0j} \\ \gamma_{1j} &= \beta_{10} + u_{1j} \end{aligned} \tag{7}$$

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 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 (7) 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)
```

Although the `menl` command is not as suitable for fitting LME models as `mixed`, it can accommodate a multistage formulation. For example, (5) can be fit in `menl` as

```
. menl y = {gamma0:}+{b1}*x, define(gamma0: {b00}+{U0[id]})
```

and (7) as

```
. menl y = {gamma0:}+{gamma1:}*x, define(gamma0: {b00}+{U0[id]}) ///
                                  define(gamma1: {b10}+{U1[id]})
```

In the above `menl`'s specifications, `gamma0` and `gamma1` can be specified more efficiently by using linear combinations; see [ME] **menl** for details.

We refer you to Raudenbush and Bryk (2002) and Rabe-Hesketh and Skrondal (2022) 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 mixed-effects model require integrating out the random effects. For LME models, this integral has a closed-form solution; for GLME and NLME 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 quasilikelihood (PQL) and marginal quasilikelihood (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 "quasilikelihood" 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. Stata commands for fitting GLME models such as `meglm` 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.

In the context of NLME models, the use of an adaptive quadrature to fit these models can be often computationally infeasible. A popular alternative method used to fit NLME models is the linearization method of Lindstrom and Bates (1990), also known as the conditional first-order linearization method. It is based on a first-order Taylor-series approximation of the mean function and essentially linearizes the mean function with respect to fixed and random effects. The linearization method is computationally efficient because it avoids the intractable integration, but the approximation cannot be made arbitrarily accurate. Despite its potential limiting accuracy, the linearization method has proven the most popular in practice (Fitzmaurice et al. 2009, sec 5.4.8). The linearization method of Lindstrom and Bates (1990), with extensions from Pinheiro and Bates (1995), is the method of estimation in `menl`.

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.

Furthermore, the Pinheiro and Chao (2006) algorithm for the random-effects mode and curvature estimates, available with option `intmethod(pclaplace)`, can speed up computations dramatically for hierarchical models with four or more levels, especially when random slopes are included.

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, except `menl`, 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.

Achieving convergence and diagnosing convergence problems can be even more challenging with NLME models. As with other mixed-effects models, complicated variance–covariance structures for random effects and errors can often lead to overparameterized models that fail to converge. In addition, highly nonlinear mean specifications can lead to multiple solutions and thus to potential convergence to a local maximum. `menl` uses the linearization estimation method that alternates between the penalized least-squares estimation of the fixed-effects parameters and the Newton–Raphson estimation of the random-effects parameters of the approximating LME model, which was the result of the linearization of the original NLME model. This alternating method does not provide a joint Hessian matrix for all parameters, so there is no check for the tolerance of the scaled gradient, and thus the convergence cannot be established in its strict sense. The convergence is declared based on the stopping rules described in *Methods and formulas* of [ME] **menl**. Exploring different initial values to investigate convergence is particularly important with NLME models; see *Obtaining initial values* in [ME] **menl**.

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

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

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^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 https://www.stata-press.com/data/r18/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	Coefficient	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	3.124225	9.801716	22.40002
var(Residual)	4.383264	.3163348	3.805112	5.04926

```
LR test vs. linear model: 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 or the variance of 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 https://www.stata-press.com/data/r18/clippings, clear
(Turfgrass experiment)
. describe
Contains data from https://www.stata-press.com/data/r18/clippings.dta
 Observations:            24                  Turfgrass experiment
    Variables:             4                  21 Feb 2022 14:57
-------------------------------------------------------------------------------
Variable      Storage   Display    Value
    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 = -13.203147
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	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
nitrogen						
Urea	5.245833	.3986014	13.16	0.000	4.464589	6.027078
Ammonium s..	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 s.. # 2	.8999996	.6551081	1.37	0.169	-.3839887	2.183988
Ammonium s.. # 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	2.117345
var(Residual)	.2145833	.1072917	.080537	.5717376

```
LR test vs. linear model: chibar2(01) = 0.53          Prob >= chibar2 = 0.2324
```

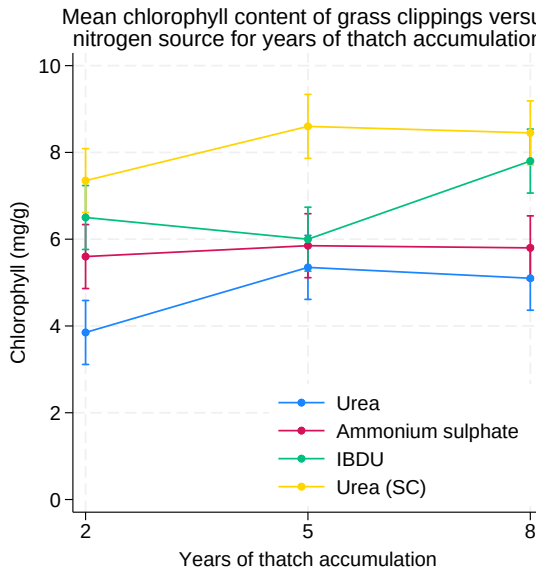
We can calculate the cell means for source of nitrogen and years of thatch accumulation by using `margins`.

```
. 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 s..	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 s..	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 s..	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()
```

	Margin	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
nitrogen						
Urea	4.766667	.2643563	18.03	0.000	4.248538	5.284796
Ammonium s..	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 https://www.stata-press.com/data/r18/admissions, clear
(Graduate school admissions data)
. describe
Contains data from https://www.stata-press.com/data/r18/admissions.dta
 Observations:            46                  Graduate school admissions data
    Variables:             4                  25 Feb 2022 09:28
                                              (_dta has notes)
```

Variable name	Storage type	Display format	Value label	Variable label
department	byte	%8.0g	dept	Department ID
nadmitted	byte	%8.0g		Number of admissions
napplied	int	%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 pts.  =          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

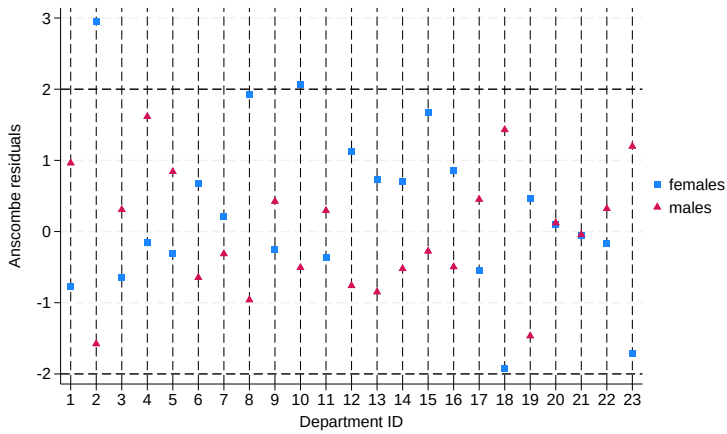
```
Note: Estimates are transformed only in the first equation to odds ratios.
Note: _cons estimates baseline odds (conditional on zero random effects).
LR test vs. logistic model: chibar2(01) = 318.77      Prob >= chibar2 = 0.0000
```

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 https://www.stata-press.com/data/r18/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	Coefficient	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

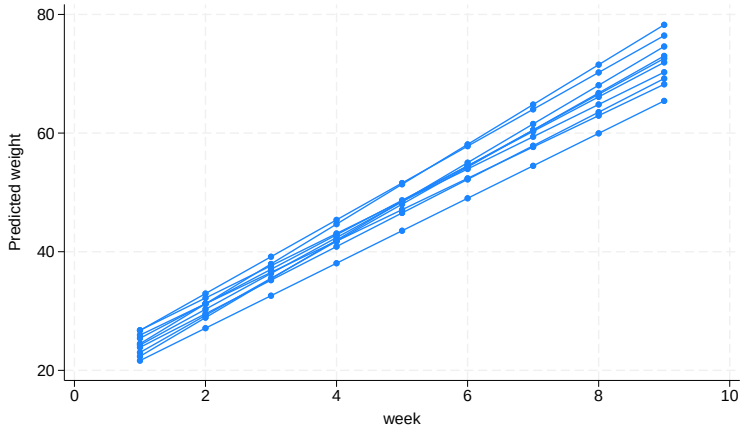
Random-effects parameters	Estimate	Std. err.	[95% conf.	interval]
id: Unstructured				
var(week)	.3715251	.0812958	.2419532	.570486
var(_cons)	6.823363	1.566194	4.351297	10.69986
cov(week,_cons)	-.0984378	.2545767	-.5973991	.4005234
var(Residual)	1.596829	.123198	1.372735	1.857505

```
LR test vs. linear model: 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 a factor variable for the number of `children`. Below we fit a standard logistic regression model amended to have random coefficients for the `children` factor variable and an overall district random intercept.

```
. use https://www.stata-press.com/data/r18/bangladesh, clear
(Bangladesh Fertility Survey, 1989)
```

```
. melogit c_use i.urban age i.children
> || district: i.children, cov(exchangeable)
> || district:, or nolog baselevel nofvlabel
```

```
Mixed-effects logistic regression               Number of obs     =      1,934
Group variable: district                        Number of groups  =         60

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

Integration method: mvaghermite                 Integration pts.  =          7

                                                Wald chi2(5)      =     100.01
Log likelihood = -1206.2397                     Prob > chi2       =     0.0000
```

```
 ( 1)  [/]var(1.children[district]) - [/]var(3.children[district]) = 0
 ( 2)  [/]cov(1.children[district],2.children[district]) -
       [/]cov(2.children[district],3.children[district]) = 0
 ( 3)  [/]cov(1.children[district],3.children[district]) -
       [/]cov(2.children[district],3.children[district]) = 0
 ( 4)  [/]var(2.children[district]) - [/]var(3.children[district]) = 0
```

c_use	Odds ratio	Std. err.	z	P>|z|	[95% conf.	interval]
urban						
0	1	(constrained)				
1	2.105163	.2546604	6.15	0.000	1.660796	2.668426
age	.9735765	.0077461	-3.37	0.001	.9585122	.9888775
children						
0	1	(constrained)				
1	2.992596	.502149	6.53	0.000	2.153867	4.157931
2	3.879345	.7094125	7.41	0.000	2.710815	5.551584
3	3.774627	.7055812	7.11	0.000	2.616744	5.444863
_cons	.1859471	.0274813	-11.38	0.000	.1391841	.2484214
district						
var(1.children)	.0841518	.0880698			.0108201	.654479
var(2.children)	.0841518	.0880698			.0108201	.654479
var(3.children)	.0841518	.0880698			.0108201	.654479
var(_cons)	.1870273	.0787274			.0819596	.426786
district						
cov(1.children, 2.children)	.0616875	.0844681	0.73	0.465	-.1038669	.2272419
cov(1.children, 3.children)	.0616875	.0844681	0.73	0.465	-.1038669	.2272419
cov(2.children, 3.children)	.0616875	.0844681	0.73	0.465	-.1038669	.2272419

```
Note: Estimates are transformed only in the first equation to odds ratios.
Note: _cons estimates baseline odds (conditional on zero random effects).
LR test vs. logistic model: 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 for the `children` factor levels are constrained to be the same, and the estimated covariances for the `children` factor levels 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 `lnor`) estimated from each study, and the corresponding estimated variance (variable `var`). The square root of the variance is stored in the variable `std` and the trial identifier is stored in the variable `trial`.

```
. use https://www.stata-press.com/data/r18/diuretics
(Meta analysis of clinical trials studying diuretics and pre-eclampsia)
. list
```

	trial	lnor	var	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 `var`, on the main diagonal. We will use this matrix to constrain the variances.

```
. mkmat var, mat(f)
. matrix f = diag(f)
```

In the random-effects equation part, we need to specify nine random slopes, one for each trial. Because random-effects equations support factor variables (see **[U] 11.4.3 Factor variables**), we can use the `ibn.trial` notation. 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.

```
. meglm lnor || _all: ibn.trial, 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       =          .
```

lnor	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
_cons	-.5166151	.2059448	-2.51	0.012	-.9202594	-.1129707
_all						
var(1.trial)	.16	(constrained)				
var(2.trial)	.12	(constrained)				
var(3.trial)	.18	(constrained)				
var(4.trial)	.3	(constrained)				
var(5.trial)	.11	(constrained)				
var(6.trial)	.01	(constrained)				
var(7.trial)	.12	(constrained)				
var(8.trial)	.69	(constrained)				
var(9.trial)	.07	(constrained)				
var(e.lnor)	.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])] = 1
. meglm lnor || 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 pts.  =          7
                                                Wald chi2(0)      =          .
Log likelihood = -9.4552759                     Prob > chi2       =          .
 ( 1)  [/]var(std[trial]) = 1
```

lnor	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
_cons	-.5166151	.2059448	-2.51	0.012	-.9202594	-.1129708
trial						
var(std)	1	(constrained)				
var(e.lnor)	.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 2022, 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 https://www.stata-press.com/data/r18/tvsfpors, clear
(Television, School, and Family Project)
. 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     =      1,600
        Grouping information
```

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 pts.  =          7
                                                Wald chi2(4)      =     124.20
Log likelihood = -2116.6981                     Prob > chi2       =     0.0000
```

thk	Coefficient	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			-.2764523	.1272184
/cut2	.6863046	.1034813			.4834849	.8891242
/cut3	1.413686	.1064889			1.204972	1.622401
school						
var(_cons)	.0186456	.0160226			.0034604	.1004695
school>class						
var(_cons)	.0519974	.0224014			.0223496	.1209745

```
LR test vs. oprobit model: 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

$$\texttt{weight}_{ij} = \beta_0 + \beta_1 \texttt{week}_{ij} + u_i + v_j + \epsilon_{ij} \tag{8}$$

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. 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 (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 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 crossed-effects 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 https://www.stata-press.com/data/r18/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	Coefficient	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 model: 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 (8):

```
. 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`—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 (2022) and Marchenko (2006) for additional techniques to make calculations more efficient in more complex mixed-effects models.

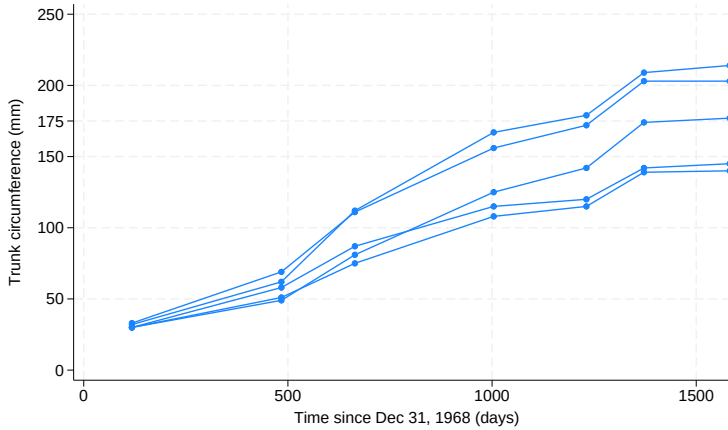
◁

Nonlinear models

NLME models are popular in population pharmacokinetics, bioassays, studies of biological and agricultural growth processes, and other applications, where the mean function is a nonlinear function of fixed and random effects. *Remarks and examples* of [ME] **menl** provide many examples of fitting different NLME models by using `menl`, including a pharmacokinetics model in example 15. Here we consider simple data from Draper and Smith (1998) that contain trunk circumference (in mm) of five different orange trees measured over seven different time points.

Let's plot our data first.

```
. use https://www.stata-press.com/data/r18/orange
(Growth of orange trees (Draper and Smith, 1998))
. twoway scatter circumf age, connect(L) ylabel(#6 175)
```



Consider the following nonlinear growth model for these data,

$$\texttt{circumf}_{ij} = \frac{\beta_1}{1 + \exp\left\{-\left(\texttt{age}_{ij} - \beta_2\right)/\beta_3\right\}} + \epsilon_{ij}$$

where ϵ_{ij}'s are i.i.d. N(0, σ^2_ϵ). In this model, β_1 can be interpreted as the average asymptotic trunk circumference of trees as $\texttt{age}_{ij} \to \infty$. We can crudely estimate it as the average of the trunk circumference values at the last observed time point, which for these data is roughly 175 mm. β_2 is the age at which a tree attains half of the average asymptotic trunk circumference β_1; that is, if we set $\texttt{age}_{ij} = \beta_2$, then $E(\texttt{circumf}_{ij}) = 0.5\beta_1$. β_3 is a scale parameter that represents the number of days it takes for a tree to grow from 50% to about 73% of the average asymptotic trunk circumference. That is, if we set $\texttt{age} = t_{0.73} = \beta_2 + \beta_3$, then $E(\texttt{circumf}_{ij}) = \beta_1/\{1 + \exp(-1)\} = 0.73\beta_1$ and then $\beta_3 = t_{0.73} - \beta_2$.

The above model can be easily fit by using, for example, `nl`; see [R] **nl**. However, if we study the graph more carefully, we will notice that there is an increasing variability in the trunk circumferences of trees as they approach their limiting age. So it may be more reasonable to allow β_1 to vary between trees,

$$\texttt{circumf}_{ij} = \frac{\beta_1 + u_{1j}}{1 + \exp\left\{-\left(\texttt{age}_{ij} - \beta_2\right)/\beta_3\right\}} + \epsilon_{ij} \tag{9}$$

where u_{1j}'s are i.i.d. $N(0, \sigma^2_{u_1})$. We use `menl` to fit this model.

The specification of NLME models in `menl` is fairly straightforward. Following the dependent variable and the equality sign (`=`), we specify the expression for the mean function as a usual Stata expression but with parameters and random effects enclosed in curly braces (`{}`).

```
. menl circumf = ({b1}+{U1[tree]})/(1+exp(-(age-{b2})/{b3}))
Obtaining starting values by EM:
Alternating PNLS/LME algorithm:
Iteration 1:  Linearization log likelihood = -131.58458
Computing standard errors:
Mixed-effects ML nonlinear regression           Number of obs     =         35
Group variable: tree                            Number of groups  =          5
                                                Obs per group:
                                                              min =          7
                                                              avg =        7.0
                                                              max =          7
Linearization log likelihood = -131.58458
```

circumf	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
/b1	191.049	16.15403	11.83	0.000	159.3877	222.7103
/b2	722.556	35.15082	20.56	0.000	653.6616	791.4503
/b3	344.1624	27.14739	12.68	0.000	290.9545	397.3703

Random-effects parameters	Estimate	Std. err.	[95% conf.	interval]
tree: Identity				
var(U1)	991.1514	639.4637	279.8776	3510.038
var(Residual)	61.56371	15.89568	37.11466	102.1184

In the above specification, we used `{U1[tree]}` to include random intercepts at the `tree` level in our model. `U1` is the name or label associated with these random intercepts.

The output of `menl` is similar to that of `mixed`—the header information is displayed first, fixed-effects parameter estimates are displayed in the first or the fixed-effects parameter table, and the estimates of variance components are displayed in the second or the random-effects parameter table.

The header information is similar to that of `mixed`, but unlike `mixed`, `menl` in general does not report a model χ^2 statistic in the header because a test of the joint significance of all fixed-effects parameters (except the constant term) may not be relevant in a nonlinear model. `menl` also reports the so-called linearization log likelihood. `menl` uses the linearization method of Lindstrom and Bates (1990), with extensions from Pinheiro and Bates (1995), for estimation. This method is based on the approximation of the NLME model by an LME model, in which a first-order Taylor-series approximation is used to linearize the nonlinear mean function with respect to fixed and random effects; see *Introduction* and *Methods and formulas* in [ME] **menl** for details. The linearization log likelihood is the log likelihood of this approximating LME model. We can use this log likelihood for model comparison of different NLME models and to form likelihood-ratio tests, but note that this is not the log likelihood of the corresponding NLME model. Depending on the accuracy of the approximation, the linearization log likelihood may be close to the true NLME log likelihood.

As part of Stata's standard estimation output, `menl` reports z tests against zeros for the estimated fixed-effects parameters. Testing a parameter against zero may or may not be of interest, or may not even be appropriate, in a nonlinear model. In our example, `{b3}` is the denominator of a fraction, so the test of `{b3}` against zero may not be feasible in this model. Instead, we may be interested in testing `{b3}` against, for example, 300, which would correspond to testing whether the average trunk circumference of orange trees increases from 50% to 73% of its asymptotic value in 300 days. We can perform this test by using, for instance, the `test` command; see [R] **test**. As a side note, setting $\beta_3 = 0$ in (9) results in a simple random-intercept model, in a limiting sense.

From the random-effects table, the variability in limiting growth β_1 between trees, labeled as `var(U1)`, is statistically significant in this model with an estimate of 991 (mm^2) and a 95% CI of [280, 3510].

We can rewrite (9) as a two-stage model,

$$\texttt{circumf}_{ij} = \frac{\phi_{1j}}{1 + \exp\left\{-\left(\texttt{age}_{ij} - \phi_{2j}\right)/\phi_{3j}\right\}} + \epsilon_{ij} \tag{10}$$

where the stage 2 specification is

$$\boldsymbol{\phi}_j = \begin{bmatrix} \phi_{1j} \\ \phi_{2j} \\ \phi_{3j} \end{bmatrix} = \begin{bmatrix} \beta_1 + u_{1j} \\ \beta_2 \\ \beta_3 \end{bmatrix} \tag{11}$$

The model defined by (10) and (11) is the same as that defined by (9) but with a different parameterization.

In `menl`, we can accommodate this two-stage formulation with the `define()` option. For example, we can fit the two-stage model defined by (10) and (11) as follows:

```
. menl circumf = {phi1:}/(1+exp(-(age-{b2})/{b3})), define(phi1: {b1}+{U1[tree]})
Obtaining starting values by EM:
Alternating PNLS/LME algorithm:
Iteration 1:  Linearization log likelihood = -131.58458
Computing standard errors:
Mixed-effects ML nonlinear regression           Number of obs     =         35
Group variable: tree                            Number of groups  =          5
                                                Obs per group:
                                                              min =          7
                                                              avg =        7.0
                                                              max =          7
Linearization log likelihood = -131.58458
        phi1: {b1}+{U1[tree]}
```

circumf	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
/b1	191.049	16.15403	11.83	0.000	159.3877	222.7103
/b2	722.556	35.15082	20.56	0.000	653.6616	791.4503
/b3	344.1624	27.14739	12.68	0.000	290.9545	397.3703

Random-effects parameters	Estimate	Std. err.	[95% conf.	interval]
tree: Identity				
var(U1)	991.1514	639.4637	279.8776	3510.038
var(Residual)	61.56371	15.89568	37.11466	102.1184

The results are identical to the previous model. Here we defined a substitutable expression `phi1` in the `define()` option as a function of `{b1}` and `{U1[tree]}` and included it in our main expression as `{phi1:}`. Including a colon (`:`) in `{phi1:}` is important to notify `menl` that it is a substitutable expression rather than a simple scalar parameter `{phi1}`.

In general, we can accommodate multistage formulations by using the `define()` option repeatedly.

More conveniently, we can use a linear-combination specification (see *Linear combinations* in [ME] **menl**) within the `define()` option to define the linear combination `{b1}+{U1[tree]}`.

```
. menl circumf = {phi1:}/(1+exp(-(age-{b2})/{b3})), define(phi1: U1[tree], xb)
Obtaining starting values by EM:
Alternating PNLS/LME algorithm:
Iteration 1:  Linearization log likelihood = -131.58458
Computing standard errors:
Mixed-effects ML nonlinear regression           Number of obs     =         35
Group variable: tree                            Number of groups  =          5
                                                Obs per group:
                                                              min =          7
                                                              avg =        7.0
                                                              max =          7
Linearization log likelihood = -131.58458
        phi1: U1[tree], xb
```

circumf	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
phi1						
_cons	191.049	16.15403	11.83	0.000	159.3877	222.7103
/b2	722.556	35.15082	20.56	0.000	653.6616	791.4503
/b3	344.1624	27.14739	12.68	0.000	290.9545	397.3703

Random-effects parameters	Estimate	Std. err.	[95% conf.	interval]
tree: Identity				
var(U1)	991.1514	639.4637	279.8776	3510.038
var(Residual)	61.56371	15.89568	37.11466	102.1184

The `{phi1: U1[tree], xb}` specification used in the `define()` option, but without curly braces, creates a linear combination named `phi1` that contains a constant `{phi1:_cons}` and random intercepts `{U1}` at the `tree` level. In the linear-combination specification, the constant is included automatically unless you specify the `noconstant` option such as `{phi1: U1[tree], xb noconstant}`. Also, you do not specify curly braces around random effects within the linear-combination specification. If we had covariates, say, `x1` and `x2`, that we also wanted to include in the linear combination, we would have used `{phi1: x1 x2 U1[tree]}`. Notice that we did not specify the `xb` option in the previous linear combination. When a linear combination contains more than one term, this option is implied. When a linear combination contains only one term, such as in `{phi1: U1[tree], xb}`, the `xb` option must be specified to request that `menl` treat the specification as a linear combination instead of a scalar parameter; see *Random-effects substitutable expressions* in [ME] **menl** for details.

Instead of using `define()`, we could have similarly specified the linear combination directly in the main expression:

```
. menl circumf = {phi1: U1[tree], xb}/(1+exp(-(age-{b2})/{b3}))
  (output omitted)
```

However, by using the `define()` option, we simplified the look of the main equation.

We can extend the stage 2 specification (11) to allow, for example, β_2 to vary across trees by including random intercepts at the tree level for ϕ_{2j},

$$\boldsymbol{\phi}_j = \begin{bmatrix} \phi_{1j} \\ \phi_{2j} \\ \phi_{3j} \end{bmatrix} = \begin{bmatrix} \beta_1 + u_{1j} \\ \beta_2 + u_{2j} \\ \beta_3 \end{bmatrix}$$

We can then fit the corresponding model by using `menl` as follows:

```
. menl circumf = {phi1:}/(1+exp(-(age-{phi2:})/{b3})),
> define(phi1: U1[tree], xb) define(phi2: U2[tree], xb)
Obtaining starting values by EM:
Alternating PNLS/LME algorithm:
Iteration 1:  Linearization log likelihood = -131.60539
Iteration 2:  Linearization log likelihood =  -131.5827
Iteration 3:  Linearization log likelihood =  -131.5805
Iteration 4:  Linearization log likelihood = -131.58027
Iteration 5:  Linearization log likelihood = -131.58026
Computing standard errors:
Mixed-effects ML nonlinear regression           Number of obs     =         35
Group variable: tree                            Number of groups  =          5
                                                Obs per group:
                                                              min =          7
                                                              avg =        7.0
                                                              max =          7
Linearization log likelihood = -131.58026
        phi1: U1[tree], xb
        phi2: U2[tree], xb
```

circumf	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
phi1						
_cons	190.5939	16.211	11.76	0.000	158.8209	222.3669
phi2						
_cons	719.6027	35.77597	20.11	0.000	649.4831	789.7223
/b3	342.0794	26.42036	12.95	0.000	290.2965	393.8624

Random-effects parameters	Estimate	Std. err.	[95% conf.	interval]
tree: Independent				
var(U1)	1012.15	666.2808	278.557	3677.698
var(U2)	503.2308	2401.324	.0436507	5801534
var(Residual)	59.27073	18.21298	32.45482	108.2434

The large standard error for the estimate of the variance component `var(U2)` suggests that our model is overparameterized—a common problem when fitting NLME models. We could verify this, for instance, by computing information criteria ([R] **estimates stats**) or by performing a likelihood-ratio test ([R] **lrtest**).

By default, `menl` assumes an independent covariance structure for the random effects such as `U1` and `U2` in our example. We can specify, for example, an unstructured model by using the `covariance()` option. We demonstrate this only for illustration, given that our simpler model that assumed independence between `U1` and `U2` was already overparameterized.

```
. menl circumf = {phi1:}/(1+exp(-(age-{phi2:})/{b3})),
> define(phi1: U1[tree], xb) define(phi2: U2[tree], xb)
> covariance(U1 U2, unstructured)

Obtaining starting values by EM:

Alternating PNLS/LME algorithm:

Iteration 1:  Linearization log likelihood = -130.90452
Iteration 2:  Linearization log likelihood = -130.90205
Iteration 3:  Linearization log likelihood = -130.90177
Iteration 4:  Linearization log likelihood = -130.90177

Computing standard errors:

Mixed-effects ML nonlinear regression           Number of obs     =         35
Group variable: tree                            Number of groups  =          5
                                                Obs per group:
                                                              min =          7
                                                              avg =        7.0
                                                              max =          7

Linearization log likelihood = -130.90177

        phi1: U1[tree], xb
        phi2: U2[tree], xb
```

circumf	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
phi1						
_cons	189.8349	17.20035	11.04	0.000	156.1228	223.5469
phi2						
_cons	709.5333	37.24229	19.05	0.000	636.5397	782.5268
/b3	340.4731	25.52176	13.34	0.000	290.4514	390.4948

Random-effects parameters	Estimate	Std. err.	[95% conf.	interval]
tree: Unstructured				
var(U1)	1180.097	775.0821	325.7263	4275.46
var(U2)	1469.879	2777.134	36.22873	59636.18
cov(U1,U2)	1015.504	1124.568	-1188.609	3219.617
var(Residual)	56.07332	16.20294	31.82681	98.79146

In `menl`, we need to list the names of the random effects in the `covariance()` option for which we want to specify a covariance structure other than the independent one used by default.

In our example, parameters ϕ_{1j} and ϕ_{2j} were modeled as linear functions of random effects and parameters β_1 and β_2. The relationship does not have to be linear; see example 15 in [ME] **menl**.

This example has a small number of trees or clusters, so REML estimation would have been more appropriate. We could have obtained REML estimates in our examples by specifying the `reml` option with `menl`.

See [ME] **menl** for more examples of and details about the `menl` command.

Acknowledgments

We are indebted to Sophia Rabe-Hesketh of the University of California, Berkeley, and coauthor of the Stata Press book *Multilevel and Longitudinal Modeling Using Stata*; Anders Skrondal of the University of Oslo and the Norwegian Institute of Public Health, and coauthor of the Stata Press book *Multilevel and Longitudinal Modeling Using Stata*; and Andrew Pickles of King's College London 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.

Breslow, N. E., and D. G. Clayton. 1993. Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association* 88: 9–25. https://doi.org/10.2307/2290687.

Davidian, M., and D. M. Giltinan. 1995. *Nonlinear Models for Repeated Measurement Data*. Boca Raton, FL: Chapman and Hall/CRC.

——. 2003. Nonlinear models for repeated measurement data: An overview and update. *Journal of Agricultural, Biological, and Environmental Statistics* 8: 387–419. https://doi.org/10.1198/1085711032697.

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

Demidenko, E. 2013. *Mixed Models: Theory and Applications with R*. 2nd ed. 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. https://doi.org/10.1111/j.2517-6161.1977.tb01600.x.

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

Draper, N., and H. Smith. 1998. *Applied Regression Analysis*. 3rd ed. New York: Wiley.

Fitzmaurice, G. M., M. Davidian, G. Verbeke, and G. Molenberghs, ed. 2009. *Longitudinal Data Analysis*. Boca Raton, FL: Chapman and Hall/CRC.

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. https://doi.org/10.1016/0091-7435(88)90053-9.

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. https://doi.org/10.2307/2526504.

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. https://doi.org/10.2307/3001853.

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. https://doi.org/10.2307/2529876.

LaMotte, L. R. 1973. Quadratic estimation of variance components. *Biometrics* 29: 311–330. https://doi.org/10.2307/2529395.

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. https://doi.org/10.1111/1467-9876.00237.

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. https://doi.org/10.2307/2291720.

Lindstrom, M. J., and D. M. Bates. 1990. Nonlinear mixed effects models for repeated measures data. *Biometrics* 46: 673–687. https://doi.org/10.2307/2532087.

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. https://doi.org/10.2307/2337136.

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: Inference and Applications to Clustering*. 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. https://doi.org/10.1191/1471082X06st106oa.

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. https://doi.org/10.2307/1390625.

——. 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. https://doi.org/10.1198/106186006X96962.

Rabe-Hesketh, S., and A. Skrondal. 2022. *Multilevel and Longitudinal Modeling Using Stata*. 4th 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. https://doi.org/10.2307/2983404.

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. https://doi.org/10.2307/2289471.

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

Stram, D. O., and J. W. Lee. 1994. Variance components testing in the longitudinal mixed effects model. *Biometrics* 50: 1171–1177. https://doi.org/10.2307/2533455.

Thompson, W. A., Jr. 1962. The problem of negative estimates of variance components. *Annals of Mathematical Statistics* 33: 273–289. https://doi.org/10.1214/aoms/1177704731.

Tierney, L., and J. B. Kadane. 1986. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association* 81: 82–86. https://doi.org/10.1080/01621459.1986.10478240.

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. https://doi.org/10.1002/1097-0258(20001230)19:24⟨3417::aid-sim614⟩3.0.co;2-l.

Tutz, G., and W. Hennevogl. 1996. Random effects in ordinal regression models. *Computational Statistics and Data Analysis* 22: 537–557. https://doi.org/10.1016/0167-9473(96)00004-7.

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. https://doi.org/10.1002/(SICI)1099-1255(199803/04)13:2⟨163::AID-JAE460⟩3.0.CO;2-Y.

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

Vonesh, E. F., and V. M. Chinchilli. 1997. *Linear and Nonlinear Models for the Analysis of Repeated Measurements*. New York: Dekker.

Also see

[ME] **Glossary**