Title

stata.com

mixed — Multilevel mixed-effects linear regression

Description

`mixed` fits linear mixed-effects models. These models are also known as multilevel models or hierarchical linear models. The overall error distribution of the linear mixed-effects model is assumed to be Gaussian, and heteroskedasticity and correlations within lowest-level groups also may be modeled.

Quick start

Linear mixed-effects model of `y` on `x` with random intercepts by `lev2`

```
mixed y x || lev2:
```

As above, but perform restricted maximum-likelihood (REML) estimation instead of the default maximum likelihood (ML) estimation

```
mixed y x || lev2:, reml
```

As above, but perform small-sample inference on `x` using the Kenward–Roger degrees of freedom (DF) method

```
mixed y x || lev2:, reml dfmethod(kroger)
```

Add random coefficients on `x`

```
mixed y x || lev2: x
```

As above, but allow correlation between the random slopes and intercepts

```
mixed y x || lev2: x, covariance(unstructured)
```

Three-level model with random intercepts by `lev2` and `lev3` for `lev2` nested within `lev3`

```
mixed y x || lev3: || lev2:
```

Crossed-effects model with two-way crossed effects by factors `a` and `b`

```
mixed y x || _all:R.a || b:
```

Menu

Statistics > Multilevel mixed-effects models > Linear regression

Syntax

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

where the syntax of *fe_equation* is

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

and the syntax of *re_equation* is one of the following:

for random coefficients and intercepts

```
levelvar: [varlist] [, re_options]
```

for random effects among the values of a factor variable

```
levelvar: R.varname [, re_options]
```

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

fe_options	Description
Model	
`noconstant`	suppress constant term from the fixed-effects equation

re_options	Description
Model	
`covariance(`*vartype*`)`	variance–covariance structure of the random effects
`noconstant`	suppress constant term from the random-effects equation
`collinear`	keep collinear variables
`fweight(`*exp*`)`	frequency weights at higher levels
`pweight(`*exp*`)`	sampling weights at higher levels

options	Description
Model	
mle	fit model via maximum likelihood; the default
reml	fit model via restricted maximum likelihood
dfmethod(*df_method*)	specify method for computing DF of a t distribution
pwscale(*scale_method*)	control scaling of sampling weights in two-level models
residuals(*rspec*)	structure of residual errors
SE/Robust	
vce(*vcetype*)	*vcetype* may be oim, robust, or cluster *clustvar*; types other than oim may not be combined with dfmethod()
Reporting	
level(#)	set confidence level; default is level(95)
variance	show random-effects and residual-error parameter estimates as variances and covariances; the default
stddeviations	show random-effects and residual-error parameter estimates as standard deviations and correlations
dftable(*dftable*)	specify contents of fixed-effects table; requires dfmethod() at estimation
noretable	suppress random-effects table
nofetable	suppress fixed-effects table
estmetric	show parameter estimates as stored in e(b)
noheader	suppress output header
nogroup	suppress table summarizing groups
nostderr	do not estimate standard errors of random-effects parameters
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
EM options	
emiterate(#)	number of EM iterations; default is emiterate(20)
emtolerance(#)	EM convergence tolerance; default is emtolerance(1e-10)
emonly	fit model exclusively using EM
emlog	show EM iteration log
emdots	show EM iterations as dots
Maximization	
maximize_options	control the maximization process; seldom used
matsqrt	parameterize variance components using matrix square roots; the default
matlog	parameterize variance components using matrix logarithms
small	replay small-sample inference results
coeflegend	display legend instead of statistics

vartype	Description
independent	one unique variance parameter per random effect, all covariances 0; the default unless the R. notation is used
exchangeable	equal variances for random effects, and one common pairwise covariance
identity	equal variances for random effects, all covariances 0; the default if the R. notation is used
unstructured	all variances and covariances to be distinctly estimated

df_method	Description
residual	residual degrees of freedom, $n - \text{rank}(X)$
repeated	repeated-measures ANOVA
anova	ANOVA
satterthwaite[, *dfopts*]	generalized Satterthwaite approximation; REML estimation only
kroger[, *dfopts*]	Kenward–Roger; REML estimation only

dftable	Description
default	test statistics, p-values, and confidence intervals; the default
ci	DFs and confidence intervals
pvalue	DFs, test statistics, and p-values

indepvars may contain factor variables; see [U] **11.4.3 Factor variables**.

depvar, *indepvars*, and *varlist* may contain time-series operators; see [U] **11.4.4 Time-series varlists**.

bayes, bootstrap, by, jackknife, mi estimate, rolling, and statsby are allowed; see [U] **11.1.10 Prefix commands**. For more details, see [BAYES] **bayes: mixed**.

mi estimate is not allowed if dfmethod() is specified.

Weights are not allowed with the bootstrap prefix; see [R] **bootstrap**.

pweights and fweights are allowed; see [U] **11.1.6 weight**. However, no weights are allowed if either option reml or option dfmethod() is specified.

small and coeflegend do not appear in the dialog box.

See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

noconstant suppresses the constant (intercept) term and may be specified for the fixed-effects equation and for any of or all the random-effects equations.

covariance(*vartype*) specifies the structure of the covariance matrix for the random effects and may be specified for each random-effects equation. *vartype* is one of the following: independent, exchangeable, identity, or unstructured.

independent allows for a distinct variance for each random effect within a random-effects equation and assumes that all covariances are 0.

`exchangeable` specifies one common variance for all random effects and one common pairwise covariance.

`identity` is short for "multiple of the identity"; that is, all variances are equal and all covariances are 0.

`unstructured` allows for all variances and covariances to be distinct. If an equation consists of p random-effects terms, the unstructured covariance matrix will have $p(p+1)/2$ unique parameters.

`covariance(independent)` is the default, except when the `R.` notation is used, in which case `covariance(identity)` is the default and only `covariance(identity)` and `covariance(exchangeable)` are allowed.

`collinear` specifies that `mixed` not omit collinear variables from the random-effects equation. Usually, there is no reason to leave collinear variables in place; in fact, doing so usually causes the estimation to fail because of the matrix singularity caused by the collinearity. However, with certain models (for example, a random-effects model with a full set of contrasts), the variables may be collinear, yet the model is fully identified because of restrictions on the random-effects covariance structure. In such cases, using the `collinear` option allows the estimation to take place with the random-effects equation intact.

`fweight(`*exp*`)` specifies frequency weights at higher levels in a multilevel model, whereas frequency weights at the first level (the observation level) are specified in the usual manner, for example, `[fw=`*fwtvar1*`]`. *exp* can be any valid Stata variable, and you can specify `fweight()` at levels two and higher of a multilevel model. For example, in the two-level model

```
. mixed fixed_portion [fw = wt1] || school: ..., fweight(wt2) ...
```

the variable `wt1` would hold the first-level (the observation-level) frequency weights, and `wt2` would hold the second-level (the school-level) frequency weights.

`pweight(`*exp*`)` specifies sampling weights at higher levels in a multilevel model, whereas sampling weights at the first level (the observation level) are specified in the usual manner, for example, `[pw=`*pwtvar1*`]`. *exp* can be any valid Stata variable, and you can specify `pweight()` at levels two and higher of a multilevel model. For example, in the two-level model

```
. mixed fixed_portion [pw = wt1] || school: ..., pweight(wt2) ...
```

variable `wt1` would hold the first-level (the observation-level) sampling weights, and `wt2` would hold the second-level (the school-level) sampling weights.

See *Survey data* in *Remarks and examples* below for more information regarding the use of sampling weights in multilevel models.

`mle` and `reml` specify the statistical method for fitting the model.

`mle`, the default, specifies that the model be fit using ML. Options `dfmethod(satterthwaite)` and `dfmethod(kroger)` are not supported under ML estimation.

`reml` specifies that the model be fit using REML, also known as residual maximum likelihood.

`dfmethod(`*df_method*`)` requests that reported hypothesis tests for the fixed effects (coefficients) use a small-sample adjustment. By default, inference is based on a large-sample approximation of the sampling distributions of the test statistics by normal and χ^2 distributions. Caution should be exercised when choosing a DF method; see *Small-sample inference for fixed effects* in *Remarks and examples* for details.

When `dfmethod(`*df_method*`)` is specified, the sampling distributions of the test statistics are approximated by a t distribution, according to the requested method for computing the DF. *df_method* is one of the following: `residual`, `repeated`, `anova`, `satterthwaite`, or `kroger`.

`residual` uses the residual degrees of freedom, $n - \text{rank}(X)$, as the DF for all tests of fixed effects. For a linear model without random effects with independent and identically distributed (i.i.d.) errors, the distributions of the test statistics for fixed effects are t distributions with the residual DF. For other mixed-effects models, this method typically leads to poor approximations of the actual sampling distributions of the test statistics.

`repeated` uses the repeated-measures ANOVA method for computing the DF. It is used with balanced repeated-measures designs with spherical correlation error structures. It partitions the residual degrees of freedom into the between-subject degrees of freedom and the within-subject degrees of freedom. `repeated` is supported only with two-level models. For more complex mixed-effects models or with unbalanced data, this method typically leads to poor approximations of the actual sampling distributions of the test statistics.

`anova` uses the traditional ANOVA method for computing the DF. According to this method, the DF for a test of a fixed effect of a given variable depends on whether that variable is also included in any of the random-effects equations. For traditional ANOVA models with balanced designs, this method provides exact sampling distributions of the test statistics. For more complex mixed-effects models or with unbalanced data, this method typically leads to poor approximations of the actual sampling distributions of the test statistics.

`satterthwaite`[`,` *dfopts*] implements a generalization of the Satterthwaite (1946) approximation of the unknown sampling distributions of test statistics for complex linear mixed-effect models. This method is supported only with REML estimation.

`kroger`[`,` *dfopts*] implements the Kenward and Roger (1997) method, which is designed to approximate unknown sampling distributions of test statistics for complex linear mixed-effects models. This method is supported only with REML estimation.

dfopts is either `eim` or `oim`.

`eim` specifies that the expected information matrix be used to compute Satterthwaite or Kenward–Roger degrees of freedom. This is the default.

`oim` specifies that the observed information matrix be used to compute Satterthwaite or Kenward–Roger degrees of freedom.

Residual, repeated, and ANOVA methods are suitable only when the sampling distributions of the test statistics are known to be t or F. This is usually only known for certain classes of linear mixed-effects models with simple covariance structures and when data are balanced. These methods are available with both ML and REML estimation.

For unbalanced data or balanced data with complicated covariance structures, the sampling distributions of the test statistics are unknown and can only be approximated. The Satterthwaite and Kenward–Roger methods provide approximations to the distributions in these cases. According to Schaalje, McBride, and Fellingham (2002), the Kenward–Roger method should, in general, be preferred to the Satterthwaite method. However, there are situations in which the two methods are expected to perform similarly, such as with compound symmetry covariance structures. The Kenward–Roger method is more computationally demanding than the Satterthwaite method. Both methods are available only with REML estimation. See *Small-sample inference for fixed effects* in *Remarks and examples* for examples and more detailed descriptions of the DF methods.

`dfmethod()` may not be combined with weighted estimation, the `mi estimate` prefix, or `vce()`, unless it is the default `vce(oim)`.

`pwscale(`*scale_method*`)` controls how sampling weights (if specified) are scaled in two-level models. *scale_method* is one of the following: `size`, `effective`, or `gk`.

`size` specifies that first-level (observation-level) weights be scaled so that they sum to the sample size of their corresponding second-level cluster. Second-level sampling weights are left unchanged.

`effective` specifies that first-level weights be scaled so that they sum to the effective sample size of their corresponding second-level cluster. Second-level sampling weights are left unchanged.

`gk` specifies the Graubard and Korn (1996) method. Under this method, second-level weights are set to the cluster averages of the products of the weights at both levels, and first-level weights are then set equal to 1.

`pwscale()` is supported only with two-level models. See *Survey data* in *Remarks and examples* below for more details on using `pwscale()`. `pwscale()` may not be combined with the `dfmethod()` option.

`residuals(`*rspec*`)` specifies the structure of the residual errors within the lowest-level groups (the second level of a multilevel model with the observations comprising the first level) of the linear mixed model. For example, if you are modeling random effects for classes nested within schools, then `residuals()` refers to the residual variance–covariance structure of the observations within classes, the lowest-level groups. *rspec* has the following syntax:

restype [, *residual_options*]

restype is one of the following: `independent`, `exchangeable`, `ar` #, `ma` #, `unstructured` `banded` #, `toeplitz` #, or `exponential`.

`independent`, the default, specifies that all residuals be i.i.d. Gaussian with one common variance. When combined with `by(`*varname*`)`, independence is still assumed, but you estimate a distinct variance for each level of *varname*. Unlike with the structures described below, *varname* does not need to be constant within groups.

`exchangeable` estimates two parameters, one common within-group variance and one common pairwise covariance. When combined with `by(`*varname*`)`, these two parameters are distinctly estimated for each level of *varname*. Because you are modeling a within-group covariance, *varname* must be constant within lowest-level groups.

`ar` # assumes that within-group errors have an autoregressive (AR) structure of order #; `ar 1` is the default. The `t(`*varname*`)` option is required, where *varname* is an integer-valued time variable used to order the observations within groups and to determine the lags between successive observations. Any nonconsecutive time values will be treated as gaps. For this structure, # + 1 parameters are estimated (# AR coefficients and one overall error variance). *restype* `ar` may be combined with `by(`*varname*`)`, but *varname* must be constant within groups.

`ma` # assumes that within-group errors have a moving-average (MA) structure of order #; `ma 1` is the default. The `t(`*varname*`)` option is required, where *varname* is an integer-valued time variable used to order the observations within groups and to determine the lags between successive observations. Any nonconsecutive time values will be treated as gaps. For this structure, # + 1 parameters are estimated (# MA coefficients and one overall error variance). *restype* `ma` may be combined with `by(`*varname*`)`, but *varname* must be constant within groups.

`unstructured` is the most general structure; it estimates distinct variances for each within-group error and distinct covariances for each within-group error pair. The `t(`*varname*`)` option is required, where *varname* is a nonnegative-integer–valued variable that identifies the observations within each group. The groups may be unbalanced in that not all levels of `t()` need to be observed within every group, but you may not have repeated `t()` values within any particular group. When you have p levels of `t()`, then $p(p+1)/2$ parameters are estimated. *restype*

`unstructured` may be combined with `by(`*varname*`)`, but *varname* must be constant within groups.

`banded` # is a special case of `unstructured` that restricts estimation to the covariances within the first # off-diagonals and sets the covariances outside this band to 0. The `t(`*varname*`)` option is required, where *varname* is a nonnegative-integer–valued variable that identifies the observations within each group. # is an integer between 0 and $p-1$, where p is the number of levels of `t()`. By default, # is $p-1$; that is, all elements of the covariance matrix are estimated. When # is 0, only the diagonal elements of the covariance matrix are estimated. *restype* `banded` may be combined with `by(`*varname*`)`, but *varname* must be constant within groups.

`toeplitz` # assumes that within-group errors have Toeplitz structure of order #, for which correlations are constant with respect to time lags less than or equal to # and are 0 for lags greater than #. The `t(`*varname*`)` option is required, where *varname* is an integer-valued time variable used to order the observations within groups and to determine the lags between successive observations. # is an integer between 1 and the maximum observed lag (the default). Any nonconsecutive time values will be treated as gaps. For this structure, $\#+1$ parameters are estimated (# correlations and one overall error variance). *restype* `toeplitz` may be combined with `by(`*varname*`)`, but *varname* must be constant within groups.

`exponential` is a generalization of the AR covariance model that allows for unequally spaced and noninteger time values. The `t(`*varname*`)` option is required, where *varname* is real-valued. For the exponential covariance model, the correlation between two errors is the parameter ρ, raised to a power equal to the absolute value of the difference between the `t()` values for those errors. For this structure, two parameters are estimated (the correlation parameter ρ and one overall error variance). *restype* `exponential` may be combined with `by(`*varname*`)`, but *varname* must be constant within groups.

residual_options are `by(`*varname*`)` and `t(`*varname*`)`.

`by(`*varname*`)` is for use within the `residuals()` option and specifies that a set of distinct residual-error parameters be estimated for each level of *varname*. In other words, you use `by()` to model heteroskedasticity.

`t(`*varname*`)` is for use within the `residuals()` option to specify a time variable for the `ar`, `ma`, `toeplitz`, and `exponential` structures, or to identify the observations when *restype* is `unstructured` or `banded`.

SE/Robust

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`oim`), that are robust to some kinds of misspecification (`robust`), and that allow for intragroup correlation (`cluster` *clustvar*); see [R] ***vce_option***. If `vce(robust)` is specified, robust variances are clustered at the highest level in the multilevel model.

`vce(robust)` and `vce(cluster` *clustvar*`)` are not supported with REML estimation. Only `vce(oim)` is allowed in combination with `dfmethod()`.

Reporting

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

`variance`, the default, displays the random-effects and residual-error parameter estimates as variances and covariances.

`stddeviations` displays the random-effects and residual-error parameter estimates as standard deviations and correlations.

`dftable(`*dftable*`)` specifies the contents of the fixed-effects table for small-sample inference when `dfmethod()` is used during estimation. *dftable* is one of the following: `default`, `ci`, or `pvalue`.

`default` displays the default standard fixed-effects table that contains test statistics, p-values, and confidence intervals.

`ci` displays the fixed-effects table in which the columns containing statistics and p-values are replaced with a column containing coefficient-specific DFs. Confidence intervals are also displayed.

`pvalue` displays the fixed-effects table that includes a column containing DFs with the standard columns containing test statistics and p-values. Confidence intervals are not displayed.

`noretable` suppresses the random-effects table from the output.

`nofetable` suppresses the fixed-effects table from the output.

`estmetric` displays all parameter estimates in one table using the metric in which they are stored in `e(b)`. The results are stored in the same metric regardless of the parameterization of the variance components, `matsqrt` or `matlog`, used at estimation time. Random-effects parameter estimates are stored as log-standard deviations and hyperbolic arctangents of correlations, with equation names that organize them by model level. Residual-variance parameter estimates are stored as log-standard deviations and, when applicable, as hyperbolic arctangents of correlations. Note that fixed-effects estimates are always stored and displayed in the same metric.

`noheader` suppresses the output header, either at estimation or upon replay.

`nogroup` suppresses the display of group summary information (number of groups, average group size, minimum, and maximum) from the output header.

`nostderr` prevents `mixed` from calculating standard errors for the estimated random-effects parameters, although standard errors are still provided for the fixed-effects parameters. Specifying this option will speed up computation times. `nostderr` is available only when residuals are modeled as independent with constant variance.

display_options: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(`*style*`)`, `cformat(%`*fmt*`)`, `pformat(%`*fmt*`)`, `sformat(%`*fmt*`)`, and `nolstretch`; see [R] **estimation options**.

EM options

These options control the expectation-maximization (EM) iterations that take place before estimation switches to a gradient-based method. When residuals are modeled as independent with constant variance, EM will either converge to the solution or bring parameter estimates close to the solution. For other residual structures or for weighted estimation, EM is used to obtain starting values.

`emiterate(#)` specifies the number of EM iterations to perform. The default is `emiterate(20)`.

`emtolerance(#)` specifies the convergence tolerance for the EM algorithm. The default is `emtolerance(1e-10)`. EM iterations will be halted once the log (restricted) likelihood changes by a relative amount less than #. At that point, optimization switches to a gradient-based method, unless `emonly` is specified, in which case maximization stops.

`emonly` specifies that the likelihood be maximized exclusively using EM. The advantage of specifying `emonly` is that EM iterations are typically much faster than those for gradient-based methods. The disadvantages are that EM iterations can be slow to converge (if at all) and that EM provides no facility for estimating standard errors for the random-effects parameters. `emonly` is available only with unweighted estimation and when residuals are modeled as independent with constant variance.

`emlog` specifies that the EM iteration log be shown. The EM iteration log is, by default, not displayed unless the `emonly` option is specified.

`emdots` specifies that the EM iterations be shown as dots. This option can be convenient because the EM algorithm may require many iterations to converge.

Maximization

maximize_options: `difficult`, `technique(`*algorithm_spec*`)`, `iterate(#)`, `[no]log`, `trace`, `gradient`, `showstep`, `hessian`, `showtolerance`, `tolerance(#)`, `ltolerance(#)`, `nrtolerance(#)`, and `nonrtolerance`; see [R] **maximize**. Those that require special mention for `mixed` are listed below.

For the `technique()` option, the default is `technique(nr)`. The `bhhh` algorithm may not be specified.

`matsqrt` (the default), during optimization, parameterizes variance components by using the matrix square roots of the variance–covariance matrices formed by these components at each model level.

`matlog`, during optimization, parameterizes variance components by using the matrix logarithms of the variance–covariance matrices formed by these components at each model level.

The `matsqrt` parameterization ensures that variance–covariance matrices are positive semidefinite, while `matlog` ensures matrices that are positive definite. For most problems, the matrix square root is more stable near the boundary of the parameter space. However, if convergence is problematic, one option may be to try the alternate `matlog` parameterization. When convergence is not an issue, both parameterizations yield equivalent results.

The following options are available with `mixed` but are not shown in the dialog box:

`small` replays previously obtained small-sample results. This option is available only upon replay and requires that the `dfmethod()` option be used during estimation. `small` is equivalent to `dftable(default)` upon replay.

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

Remarks and examples

stata.com

Remarks are presented under the following headings:

Introduction

Linear mixed models are models containing both fixed effects and random effects. They are a generalization of linear regression allowing for the inclusion of random deviations (effects) other than those associated with the overall error term. In matrix notation,

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

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

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

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

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

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

Comprehensive treatments of mixed models are provided by, among others, Searle, Casella, and McCulloch (1992); McCulloch, Searle, and Neuhaus (2008); Verbeke and Molenberghs (2000); Raudenbush and Bryk (2002); and Pinheiro and Bates (2000). In particular, chapter 2 of Searle, Casella, and McCulloch (1992) provides an excellent history.

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

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

The most popular methods, however, are ML and REML, and these are the two methods that are supported by `mixed`. The ML estimates are based on the usual application of likelihood theory, given

the distributional assumptions of the model. The basic idea behind REML (Thompson 1962) is that you can form a set of linear contrasts of the response that do not depend on the fixed effects $\boldsymbol{\beta}$, but instead depend only on the variance components to be estimated. You then apply ML methods by using the distribution of the linear contrasts to form the likelihood.

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

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

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

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

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

In the sections that follow, we assume that residuals are independent with constant variance; that is, in (3) we treat $\boldsymbol{\Lambda}$ equal to the identity matrix and limit ourselves to estimating one overall residual variance, σ^2_ϵ. Beginning in *Heteroskedastic residual errors*, we relax this assumption.

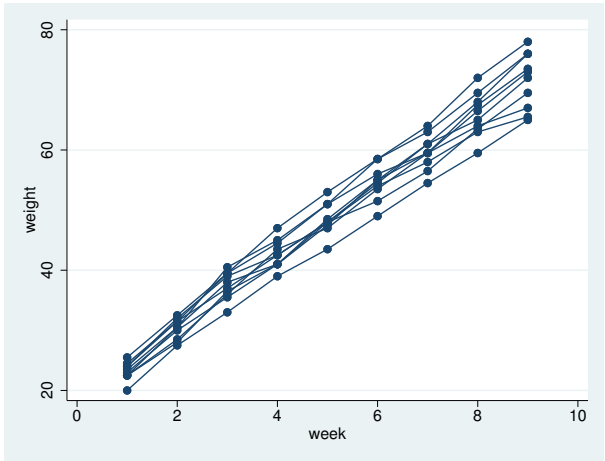
Two-level models

We begin with a simple application of (2) as a two-level model, because a one-level linear model, by our terminology, is just standard OLS regression.

▷ Example 1: Two-level random intercept 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`. Below is a plot of the growth curves for the first 10 pigs.

```
. use http://www.stata-press.com/data/r15/pig
(Longitudinal analysis of pig weights)
. twoway connected weight week if id<=10, connect(L)
```



It seems clear that each pig experiences a linear trend in growth and that overall weight measurements vary from pig to pig. Because we are not really interested in these particular 48 pigs per se, we instead treat them as a random sample from a larger population and model the between-pig variability as a random effect, or in the terminology of (2), as a random-intercept term at the pig level. We thus wish to fit the model

$$\texttt{weight}_{ij} = \beta_0 + \beta_1 \texttt{week}_{ij} + u_j + \epsilon_{ij} \tag{4}$$

for $i = 1, \ldots, 9$ 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

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

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

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
id: Identity				
var(_cons)	14.81751	3.124226	9.801716	22.40002
var(Residual)	4.383264	.3163348	3.805112	5.04926

```
LR test vs. linear model: chibar2(01) = 472.65        Prob >= chibar2 = 0.0000
```

Notes:

1. By typing `weight week`, we specified the response, `weight`, and the fixed portion of the model in the same way that we would if we were using `regress` or any other estimation command. Our fixed effects are a coefficient on `week` and a constant term.
2. When we added `|| id:`, we specified random effects at the level identified by the group variable `id`, that is, the pig level (level two). Because we wanted only a random intercept, that is all we had to type.
3. The estimation log consists of three parts:
   a. A set of EM iterations used to refine starting values. By default, the iterations themselves are not displayed, but you can display them with the `emlog` option.
   b. A set of gradient-based iterations. By default, these are Newton–Raphson iterations, but other methods are available by specifying the appropriate `maximize` options; see [R] **maximize**.
   c. The message "Computing standard errors". This is just to inform you that `mixed` has finished its iterative maximization and is now reparameterizing from a matrix-based parameterization (see *Methods and formulas*) to the natural metric of variance components and their estimated standard errors.
4. The output title, "Mixed-effects ML regression", informs us that our model was fit using ML, the default. For REML estimates, use the `reml` option.

   Because this model is a simple random-intercept model fit by ML, it would be equivalent to using `xtreg` with its `mle` option.
5. The first estimation table reports the fixed effects. We estimate $\beta_0 = 19.36$ and $\beta_1 = 6.21$.

6. The second estimation table shows the estimated variance components. The first section of the table is labeled `id: Identity`, meaning that these are random effects at the `id` (pig) level and that their variance–covariance matrix is a multiple of the identity matrix; that is, $\boldsymbol{\Sigma} = \sigma_u^2\mathbf{I}$. Because we have only one random effect at this level, `mixed` knew that `Identity` is the only possible covariance structure. In any case, the variance of the level-two errors, σ_u^2, is estimated as 14.82 with standard error 3.12.
7. The row labeled `var(Residual)` displays the estimated variance of the overall error term; that is, $\widehat{\sigma}_\epsilon^2 = 4.38$. This is the variance of the level-one errors, that is, the residuals.
8. Finally, a likelihood-ratio test comparing the model with one-level ordinary linear regression, model (4) without u_j, is provided and is highly significant for these data.

We now store our estimates for later use:

```
. estimates store randint
```

◁

▷ Example 2: Two-level random slope model

Extending (4) to allow for a random slope on `week` yields the model

$$\texttt{weight}_{ij} = \beta_0 + \beta_1\texttt{week}_{ij} + u_{0j} + u_{1j}\texttt{week}_{ij} + \epsilon_{ij} \tag{5}$$

and we fit this with `mixed`:

```
. mixed weight week || id: week
Performing EM optimization:
Performing gradient-based optimization:
Iteration 0:   log likelihood = -869.03825
Iteration 1:   log likelihood = -869.03825
Computing standard errors:
Mixed-effects ML regression                     Number of obs     =        432
Group variable: id                              Number of groups  =         48
                                                Obs per group:
                                                              min =          9
                                                              avg =        9.0
                                                              max =          9
                                                Wald chi2(1)      =    4689.51
Log likelihood = -869.03825                     Prob > chi2       =     0.0000
```

weight	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
week	6.209896	.0906819	68.48	0.000	6.032163	6.387629
_cons	19.35561	.3979159	48.64	0.000	18.57571	20.13551

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
id: Independent				
var(week)	.3680668	.0801181	.2402389	.5639103
var(_cons)	6.756364	1.543503	4.317721	10.57235
var(Residual)	1.598811	.1233988	1.374358	1.85992

```
LR test vs. linear model: chi2(2) = 764.42                Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
. estimates store randslope
```

Because we did not specify a covariance structure for the random effects $(u_{0j}, u_{1j})'$, `mixed` used the default `Independent` structure; that is,

$$\mathbf{\Sigma} = \text{Var}\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} = \begin{bmatrix} \sigma^2_{u0} & 0 \\ 0 & \sigma^2_{u1} \end{bmatrix} \tag{6}$$

with $\widehat{\sigma}^2_{u0} = 6.76$ and $\widehat{\sigma}^2_{u1} = 0.37$. Our point estimates of the fixed effects are essentially identical to those from model (4), but note that this does not hold generally. Given the 95% confidence interval for $\widehat{\sigma}^2_{u1}$, it would seem that the random slope is significant, and we can use `lrtest` and our two stored estimation results to verify this fact:

```
. lrtest randslope randint
Likelihood-ratio test                                 LR chi2(1)  =    291.78
(Assumption: randint nested in randslope)             Prob > chi2 =    0.0000

Note: The reported degrees of freedom assumes the null hypothesis is not on
      the boundary of the parameter space.  If this is not true, then the
      reported test is conservative.
```

The near-zero significance level favors the model that allows for a random pig-specific regression line over the model that allows only for a pig-specific shift.

◁

Covariance structures

In example 2, we fit a model with the default `Independent` covariance given in (6). Within any random-effects level specification, we can override this default by specifying an alternative covariance structure via the `covariance()` option.

▷ Example 3: Two-level model with correlated random effects

We generalize (6) to allow u_{0j} and u_{1j} to be correlated; that is,

$$\mathbf{\Sigma} = \text{Var}\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} = \begin{bmatrix} \sigma^2_{u0} & \sigma_{01} \\ \sigma_{01} & \sigma^2_{u1} \end{bmatrix}$$

```
. mixed weight week || id: week, covariance(unstructured)
Performing EM optimization:
Performing gradient-based optimization:
Iteration 0:   log likelihood = -868.96185
Iteration 1:   log likelihood = -868.96185
Computing standard errors:
Mixed-effects ML regression                     Number of obs     =        432
Group variable: id                              Number of groups  =         48
                                                Obs per group:
                                                              min =          9
                                                              avg =        9.0
                                                              max =          9
                                                Wald chi2(1)      =    4649.17
Log likelihood = -868.96185                     Prob > chi2       =     0.0000
```

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

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

```
LR test vs. linear model: chi2(3) = 764.58                Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

But we do not find the correlation to be at all significant.

```
. lrtest . randslope
Likelihood-ratio test                                 LR chi2(1)  =      0.15
(Assumption: randslope nested in .)                   Prob > chi2 =    0.6959
```

◁

Instead, we could have also specified `covariance(identity)`, restricting u_{0j} and u_{1j} to not only be independent but also to have common variance, or we could have specified `covariance(exchangeable)`, which imposes a common variance but allows for a nonzero correlation.

Likelihood versus restricted likelihood

Thus far, all our examples have used ML to estimate variance components. We could have just as easily asked for REML estimates. Refitting the model in example 2 by REML, we get

```
. mixed weight week || id: week, reml
Performing EM optimization:
Performing gradient-based optimization:
Iteration 0:   log restricted-likelihood = -870.51473
Iteration 1:   log restricted-likelihood = -870.51473
Computing standard errors:
Mixed-effects REML regression                   Number of obs     =        432
Group variable: id                              Number of groups  =         48
                                                Obs per group:
                                                              min =          9
                                                              avg =        9.0
                                                              max =          9
                                                Wald chi2(1)      =    4592.10
Log restricted-likelihood = -870.51473          Prob > chi2       =     0.0000
```

weight	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
week	6.209896	.0916387	67.77	0.000	6.030287	6.389504
_cons	19.35561	.4021144	48.13	0.000	18.56748	20.14374

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
id: Independent				
var(week)	.3764405	.0827027	.2447317	.5790317
var(_cons)	6.917604	1.593247	4.404624	10.86432
var(Residual)	1.598784	.1234011	1.374328	1.859898

```
LR test vs. linear model: chi2(2) = 765.92                 Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

Although ML estimators are based on the usual likelihood theory, the idea behind REML is to transform the response into a set of linear contrasts whose distribution is free of the fixed effects $\boldsymbol{\beta}$. The restricted likelihood is then formed by considering the distribution of the linear contrasts. Not only does this make the maximization problem free of $\boldsymbol{\beta}$, it also incorporates the degrees of freedom used to estimate $\boldsymbol{\beta}$ into the estimation of the variance components. This follows because, by necessity, the rank of the linear contrasts must be less than the number of observations.

As a simple example, consider a constant-only regression where $y_i \sim N(\mu, \sigma^2)$ for $i = 1, \ldots, n$. The ML estimate of σ^2 can be derived theoretically as the n-divided sample variance. The REML estimate can be derived by considering the first $n - 1$ error contrasts, $y_i - \overline{y}$, whose joint distribution is free of μ. Applying maximum likelihood to this distribution results in an estimate of σ^2, that is, the $(n - 1)$-divided sample variance, which is unbiased for σ^2.

The unbiasedness property of REML extends to all mixed models when the data are balanced, and thus REML would seem the clear choice in balanced-data problems, although in large samples the difference between ML and REML is negligible. One disadvantage of REML is that likelihood-ratio (LR) tests based on REML are inappropriate for comparing models with different fixed-effects specifications. ML is appropriate for such LR tests and has the advantage of being easy to explain and being the method of choice for other estimators.

Another factor to consider is that ML estimation under `mixed` is more feature-rich, allowing for weighted estimation and robust variance–covariance matrices, features not supported under REML. In the end, which method to use should be based both on your needs and on personal taste.

Examining the REML output, we find that the estimates of the variance components are slightly larger than the ML estimates. This is typical, because ML estimates, which do not incorporate the degrees of freedom used to estimate the fixed effects, tend to be biased downward.

Three-level models

The clustered-data representation of the mixed model given in (2) can be extended to two nested levels of clustering, creating a three-level model once the observations are considered. Formally,

$$\mathbf{y}_{jk} = \mathbf{X}_{jk}\boldsymbol{\beta} + \mathbf{Z}^{(3)}_{jk}\mathbf{u}^{(3)}_{k} + \mathbf{Z}^{(2)}_{jk}\mathbf{u}^{(2)}_{jk} + \boldsymbol{\epsilon}_{jk} \tag{7}$$

for $i = 1, \ldots, n_{jk}$ first-level observations nested within $j = 1, \ldots, M_k$ second-level groups, which are nested within $k = 1, \ldots, M$ third-level groups. Group j, k consists of n_{jk} observations, so $\mathbf{y}_{jk}$, $\mathbf{X}_{jk}$, and $\boldsymbol{\epsilon}_{jk}$ each have row dimension n_{jk}. $\mathbf{Z}^{(3)}_{jk}$ is the $n_{jk} \times q_3$ design matrix for the third-level random effects $\mathbf{u}^{(3)}_{k}$, and $\mathbf{Z}^{(2)}_{jk}$ is the $n_{jk} \times q_2$ design matrix for the second-level random effects $\mathbf{u}^{(2)}_{jk}$. Furthermore, assume that

$$\mathbf{u}^{(3)}_{k} \sim N(\mathbf{0}, \boldsymbol{\Sigma}_3); \quad \mathbf{u}^{(2)}_{jk} \sim N(\mathbf{0}, \boldsymbol{\Sigma}_2); \quad \boldsymbol{\epsilon}_{jk} \sim N(\mathbf{0}, \sigma^2_{\epsilon}\mathbf{I})$$

and that $\mathbf{u}^{(3)}_{k}$, $\mathbf{u}^{(2)}_{jk}$, and $\boldsymbol{\epsilon}_{jk}$ are independent.

Fitting a three-level model requires you to specify two random-effects equations: one for level three and then one for level two. The variable list for the first equation represents $\mathbf{Z}^{(3)}_{jk}$ and for the second equation represents $\mathbf{Z}^{(2)}_{jk}$; that is, you specify the levels top to bottom in `mixed`.

▷ Example 4: Three-level model with random intercepts

Baltagi, Song, and Jung (2001) estimate a Cobb–Douglas production function examining the productivity of public capital in each state's private output. Originally provided by Munnell (1990), the data were recorded over 1970–1986 for 48 states grouped into nine regions.

```
. use http://www.stata-press.com/data/r15/productivity
(Public Capital Productivity)
. describe
Contains data from http://www.stata-press.com/data/r15/productivity.dta
  obs:           816                          Public Capital Productivity
 vars:            11                          29 Mar 2016 10:57
 size:        29,376                          (_dta has notes)
```

variable name	storage type	display format	value label	variable label
state	byte	%9.0g		states 1-48
region	byte	%9.0g		regions 1-9
year	int	%9.0g		years 1970-1986
public	float	%9.0g		public capital stock
hwy	float	%9.0g		log(highway component of public)
water	float	%9.0g		log(water component of public)
other	float	%9.0g		log(bldg/other component of public)
private	float	%9.0g		log(private capital stock)
gsp	float	%9.0g		log(gross state product)
emp	float	%9.0g		log(non-agriculture payrolls)
unemp	float	%9.0g		state unemployment rate

```
Sorted by:
```

Because the states are nested within regions, we fit a three-level mixed model with random intercepts at both the region and the state-within-region levels. That is, we use (7) with both $\mathbf{Z}_{jk}^{(3)}$ and $\mathbf{Z}_{jk}^{(2)}$ set to the $n_{jk} \times 1$ column of ones, and $\boldsymbol{\Sigma}_3 = \sigma_3^2$ and $\boldsymbol{\Sigma}_2 = \sigma_2^2$ are both scalars.

```
. mixed gsp private emp hwy water other unemp || region: || state:
  (output omitted)
Mixed-effects ML regression                     Number of obs     =        816
```

Group Variable	No. of Groups	Observations per Group Minimum	Average	Maximum
region	9	51	90.7	136
state	48	17	17.0	17

```
                                                Wald chi2(6)      =   18829.06
Log likelihood =  1430.5017                     Prob > chi2       =     0.0000
```

gsp	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
private	.2671484	.0212591	12.57	0.000	.2254814	.3088154
emp	.754072	.0261868	28.80	0.000	.7027468	.8053973
hwy	.0709767	.023041	3.08	0.002	.0258172	.1161363
water	.0761187	.0139248	5.47	0.000	.0488266	.1034109
other	-.0999955	.0169366	-5.90	0.000	-.1331906	-.0668004
unemp	-.0058983	.0009031	-6.53	0.000	-.0076684	-.0041282
_cons	2.128823	.1543854	13.79	0.000	1.826233	2.431413

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
region: Identity				
var(_cons)	.0014506	.0012995	.0002506	.0083957
state: Identity				
var(_cons)	.0062757	.0014871	.0039442	.0099855
var(Residual)	.0013461	.0000689	.0012176	.0014882

LR test vs. linear model: chi2(2) = 1154.73 Prob > chi2 = 0.0000

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

Notes:

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

After adjusting for the nested-level error structure, we find that the highway and water components of public capital had significant positive effects on private output, whereas the other public buildings component had a negative effect.

◁

❑ Technical note

In the previous example, the states are coded 1–48 and are nested within nine regions. `mixed` treated the states as nested within regions, regardless of whether the codes for each state were unique between regions. That is, even if codes for states were duplicated between regions, `mixed` would have enforced the nesting and produced the same results.

The group information at the top of the `mixed` output and that produced by the postestimation command `estat group` (see [ME] **estat group**) take the nesting into account. The statistics are thus not necessarily what you would get if you instead `tabulated` each group variable individually.

❑

Model (7) extends in a straightforward manner to more than three levels, as does the specification of such models in `mixed`.

Blocked-diagonal covariance structures

Covariance matrices of random effects within an equation can be modeled either as a multiple of the identity matrix, as diagonal (that is, `Independent`), as exchangeable, or as general symmetric (`Unstructured`). These may also be combined to produce more complex block-diagonal covariance structures, effectively placing constraints on the variance components.

▷ Example 5: Using repeated levels to induce blocked-diagonal covariance structures

Returning to our productivity data, we now add random coefficients on `hwy` and `unemp` at the `region` level. This only slightly changes the estimates of the fixed effects, so we focus our attention on the variance components:

```
. mixed gsp private emp hwy water other unemp || region: hwy unemp || state:,
> nolog nogroup nofetable
Mixed-effects ML regression                     Number of obs     =        816
                                                Wald chi2(6)      =   17137.94
Log likelihood =  1447.6787                     Prob > chi2       =     0.0000
```

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
region: Independent				
var(hwy)	.0000209	.0001103	6.71e-10	.6506957
var(unemp)	.0000238	.0000135	7.84e-06	.0000722
var(_cons)	.0030349	.0086684	.0000112	.8191291
state: Identity				
var(_cons)	.0063658	.0015611	.0039365	.0102943
var(Residual)	.0012469	.0000643	.001127	.0013795

```
LR test vs. linear model: chi2(4) = 1189.08                   Prob > chi2 = 0.0000

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

. estimates store prodrc
```

This model is the same as that fit in example 4 except that $\mathbf{Z}_{jk}^{(3)}$ is now the $n_{jk} \times 3$ matrix with columns determined by the values of `hwy`, `unemp`, and an intercept term (one), in that order, and (because we used the default `Independent` structure) $\mathbf{\Sigma}_3$ is

$$\mathbf{\Sigma}_3 = \begin{array}{c} \begin{array}{ccc} \texttt{hwy} & \texttt{unemp} & \texttt{\_cons} \end{array} \\ \begin{pmatrix} \sigma_a^2 & 0 & 0 \\ 0 & \sigma_b^2 & 0 \\ 0 & 0 & \sigma_c^2 \end{pmatrix} \end{array}$$

The random-effects specification at the state level remains unchanged; that is, $\mathbf{\Sigma}_2$ is still treated as the scalar variance of the random intercepts at the state level.

An LR test comparing this model with that from example 4 favors the inclusion of the two random coefficients, a fact we leave to the interested reader to verify.

The estimated variance components, upon examination, reveal that the variances of the random coefficients on `hwy` and `unemp` could be treated as equal. That is,

$$\mathbf{\Sigma}_3 = \begin{array}{c} \begin{array}{ccc} \texttt{hwy} & \texttt{unemp} & \texttt{\_cons} \end{array} \\ \begin{pmatrix} \sigma_a^2 & 0 & 0 \\ 0 & \sigma_a^2 & 0 \\ 0 & 0 & \sigma_c^2 \end{pmatrix} \end{array}$$

looks plausible. We can impose this equality constraint by treating $\mathbf{\Sigma}_3$ as block diagonal: the first block is a 2×2 multiple of the identity matrix, that is, $\sigma_a^2 \mathbf{I}_2$; the second is a scalar, equivalently, a 1×1 multiple of the identity.

We construct block-diagonal covariances by repeating level specifications:

```
. mixed gsp private emp hwy water other unemp || region: hwy unemp,
> cov(identity) || region: || state:, nolog nogroup nofetable
Mixed-effects ML regression                     Number of obs     =        816
                                                Wald chi2(6)      =   17136.65
Log likelihood =  1447.6784                     Prob > chi2       =     0.0000
```

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
region: Identity				
var(hwy unemp)	.0000238	.0000134	7.89e-06	.0000719
region: Identity				
var(_cons)	.0028191	.0030429	.0003399	.023383
state: Identity				
var(_cons)	.006358	.0015309	.0039661	.0101925
var(Residual)	.0012469	.0000643	.001127	.0013795

```
LR test vs. linear model: chi2(3) = 1189.08                   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

We specified two equations for the `region` level: the first for the random coefficients on `hwy` and `unemp` with covariance set to `Identity` and the second for the random intercept `_cons`, whose covariance defaults to `Identity` because it is of dimension 1. `mixed` labeled the estimate of σ_a^2 as `var(hwy unemp)` to designate that it is common to the random coefficients on both `hwy` and `unemp`.

An LR test shows that the constrained model fits equally well.

```
. lrtest . prodrc
Likelihood-ratio test                                 LR chi2(1)  =      0.00
(Assumption: . nested in prodrc)                      Prob > chi2 =    0.9784
Note: The reported degrees of freedom assumes the null hypothesis is not on
      the boundary of the parameter space.  If this is not true, then the
      reported test is conservative.
```

◁

Because the null hypothesis for this test is one of equality ($H_0 : \sigma_a^2 = \sigma_b^2$), it is not on the boundary of the parameter space. As such, we can take the reported significance as precise rather than a conservative estimate.

You can repeat level specifications as often as you like, defining successive blocks of a block-diagonal covariance matrix. However, repeated-level equations must be listed consecutively; otherwise, `mixed` will give an error.

❑ Technical note

In the previous estimation output, there was no constant term included in the first `region` equation, even though we did not use the `noconstant` option. When you specify repeated-level equations, `mixed` knows not to put constant terms in each equation because such a model would be unidentified. By default, it places the constant in the last repeated-level equation, but you can use `noconstant` creatively to override this.

❑

Linear mixed-effects models can also be fit using `meglm` with the default `gaussian` family. `meglm` provides two more covariance structures through which you can impose constraints on variance components; see [ME] **meglm** for details.

Heteroskedastic random effects

Blocked-diagonal covariance structures and repeated-level specifications of random effects can also be used to model heteroskedasticity among random effects at a given level.

▷ Example 6: Using repeated levels to model heteroskedasticity

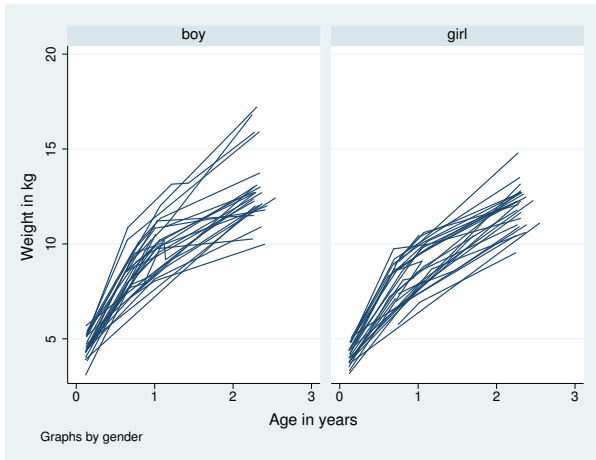
Following Rabe-Hesketh and Skrondal (2012, sec. 7.2), we analyze data from Asian children in a British community who were weighed up to four times, roughly between the ages of 6 weeks and 27 months. The dataset is a random sample of data previously analyzed by Goldstein (1986) and Prosser, Rasbash, and Goldstein (1991).

```
. use http://www.stata-press.com/data/r15/childweight
(Weight data on Asian children)

. describe

Contains data from http://www.stata-press.com/data/r15/childweight.dta
  obs:           198                          Weight data on Asian children
 vars:             5                          23 May 2016 15:12
 size:         3,168                          (_dta has notes)
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
id              int     %8.0g                 child identifier
age             float   %8.0g                 age in years
weight          float   %8.0g                 weight in Kg
brthwt          int     %8.0g                 Birth weight in g
girl            float   %9.0g      bg         gender
-------------------------------------------------------------------------------
Sorted by: id  age

. graph twoway (line weight age, connect(ascending)), by(girl)
> xtitle(Age in years) ytitle(Weight in kg)
```



Ignoring gender effects for the moment, we begin with the following model for the ith measurement on the jth child:

$$\texttt{weight}_{ij} = \beta_0 + \beta_1 \texttt{age}_{ij} + \beta_2 \texttt{age}_{ij}^2 + u_{j0} + u_{j1}\texttt{age}_{ij} + \epsilon_{ij}$$

This models overall mean growth as quadratic in age and allows for two child-specific random effects: a random intercept u_{j0}, which represents each child's vertical shift from the overall mean (β_0), and a random age slope u_{j1}, which represents each child's deviation in linear growth rate from the overall mean linear growth rate (β_1). For simplicity, we do not consider child-specific changes in the quadratic component of growth.

```
. mixed weight age c.age#c.age || id: age, nolog
Mixed-effects ML regression                     Number of obs     =        198
Group variable: id                              Number of groups  =         68
                                                Obs per group:
                                                              min =          1
                                                              avg =        2.9
                                                              max =          5
                                                Wald chi2(2)      =    1863.46
Log likelihood = -258.51915                     Prob > chi2       =     0.0000
```

weight	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
age	7.693701	.2381076	32.31	0.000	7.227019	8.160384
c.age#c.age	-1.654542	.0874987	-18.91	0.000	-1.826037	-1.483048
_cons	3.497628	.1416914	24.68	0.000	3.219918	3.775338

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
id: Independent				
var(age)	.2987207	.0827569	.1735603	.5141388
var(_cons)	.5023857	.141263	.2895294	.8717298
var(Residual)	.3092897	.0474887	.2289133	.417888

```
LR test vs. linear model: chi2(2) = 114.70                Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

◁

Because there is no reason to believe that the random effects are uncorrelated, it is always a good idea to first fit a model with the `covariance(unstructured)` option. We do not include the output for such a model because for these data the correlation between random effects is not significant; however, we did check this before reverting to `mixed`'s default `Independent` structure.

Next we introduce gender effects into the fixed portion of the model by including a main gender effect and a gender–age interaction for overall mean growth:

```
. mixed weight i.girl i.girl#c.age c.age#c.age || id: age, nolog
Mixed-effects ML regression                     Number of obs     =        198
Group variable: id                              Number of groups  =         68
                                                Obs per group:
                                                              min =          1
                                                              avg =        2.9
                                                              max =          5
                                                Wald chi2(4)      =    1942.30
Log likelihood =   -253.182                     Prob > chi2       =     0.0000
```

weight	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
girl						
girl	-.5104676	.2145529	-2.38	0.017	-.9309835	-.0899516
girl#c.age						
boy	7.806765	.2524583	30.92	0.000	7.311956	8.301574
girl	7.577296	.2531318	29.93	0.000	7.081166	8.073425
c.age#c.age	-1.654323	.0871752	-18.98	0.000	-1.825183	-1.483463
_cons	3.754275	.1726404	21.75	0.000	3.415906	4.092644

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
id: Independent				
var(age)	.2772846	.0769233	.1609861	.4775987
var(_cons)	.4076892	.12386	.2247635	.7394906
var(Residual)	.3131704	.047684	.2323672	.422072

```
LR test vs. linear model: chi2(2) = 104.39                Prob > chi2 = 0.0000

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

. estimates store homoskedastic
```

The main gender effect is significant at the 5% level, but the gender–age interaction is not:

```
. test 0.girl#c.age = 1.girl#c.age
 ( 1)  [weight]0b.girl#c.age - [weight]1.girl#c.age = 0
           chi2(  1) =    1.66
         Prob > chi2 =    0.1978
```

On average, boys are heavier than girls, but their average linear growth rates are not significantly different.

In the above model, we introduced a gender effect on average growth, but we still assumed that the variability in child-specific deviations from this average was the same for boys and girls. To check this assumption, we introduce gender into the random component of the model. Because support for factor-variable notation is limited in specifications of random effects (see *Crossed-effects models* below), we need to generate the interactions ourselves.

```
. generate boy = !girl
. generate boyXage = boy*age
. generate girlXage = girl*age
. mixed weight i.girl i.girl#c.age c.age#c.age || id: boy boyXage, noconstant
> || id: girl girlXage, noconstant nolog nofetable
Mixed-effects ML regression                     Number of obs     =        198
Group variable: id                              Number of groups  =         68
                                                Obs per group:
                                                              min =          1
                                                              avg =        2.9
                                                              max =          5
                                                Wald chi2(4)      =    2358.11
Log likelihood = -248.94752                     Prob > chi2       =     0.0000
```

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
id: Independent				
var(boy)	.3161091	.1557911	.1203181	.8305061
var(boyXage)	.4734482	.1574626	.2467028	.9085962
id: Independent				
var(girl)	.5798676	.1959725	.2989896	1.124609
var(girlXage)	.0664634	.0553274	.0130017	.3397538
var(Residual)	.3078826	.046484	.2290188	.4139037

```
LR test vs. linear model: chi2(4) = 112.86                Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
. estimates store heteroskedastic
```

In the above, we suppress displaying the fixed portion of the model (the `nofetable` option) because it does not differ much from that of the previous model.

Our previous model had the random-effects specification

```
|| id: age
```

which we have replaced with the dual repeated-level specification

```
|| id: boy boyXage, noconstant || id: girl girlXage, noconstant
```

The former models a random intercept and random slope on age, and does so treating all children as a random sample from one population. The latter also specifies a random intercept and random slope on age, but allows for the variability of the random intercepts and slopes to differ between boys and girls. In other words, it allows for heteroskedasticity in random effects due to gender. We use the `noconstant` option so that we can separate the overall random intercept (automatically provided by the former syntax) into one specific to boys and one specific to girls.

There seems to be a large gender effect in the variability of linear growth rates. We can compare both models with an LR test, recalling that we stored the previous estimation results under the name `homoskedastic`:

```
. lrtest homoskedastic heteroskedastic
Likelihood-ratio test                                 LR chi2(2)  =      8.47
(Assumption: homoskedastic nested in heteroskedas~c)  Prob > chi2 =    0.0145
Note: The reported degrees of freedom assumes the null hypothesis is not on
      the boundary of the parameter space.  If this is not true, then the
      reported test is conservative.
```

Because the null hypothesis here is one of equality of variances and not that variances are 0, the above does not test on the boundary; thus we can treat the significance level as precise and not conservative. Either way, the results favor the new model with heteroskedastic random effects.

Heteroskedastic residual errors

Up to this point, we have assumed that the level-one residual errors—the ϵ's in the stated models—have been i.i.d. Gaussian with variance σ^2_ϵ. This is demonstrated in `mixed` output in the random-effects table, where up until now we have estimated a single residual-error variance, labeled as `var(Residual)`.

To relax the assumptions of homoskedasticity or independence of residual errors, use the `residuals()` option.

▷ Example 7: Independent residual variance structure

West, Welch, and Gałecki (2015, chap. 7) analyze data studying the effect of ceramic dental veneer placement on gingival (gum) health. Data on 55 teeth located in the maxillary arches of 12 patients were considered.

```
. use http://www.stata-press.com/data/r15/veneer, clear
(Dental veneer data)
. describe
Contains data from http://www.stata-press.com/data/r15/veneer.dta
  obs:           110                          Dental veneer data
 vars:             7                          24 May 2016 12:11
 size:         1,100                          (_dta has notes)
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
patient         byte    %8.0g                 Patient ID
tooth           byte    %8.0g                 Tooth number with patient
gcf             byte    %8.0g                 Gingival crevicular fluid (GCF)
age             byte    %8.0g                 Patient age
base_gcf        byte    %8.0g                 Baseline GCF
cda             float   %9.0g                 Average contour difference after
                                                veneer placement
followup        byte    %9.0g      t          Follow-up time: 3 or 6 months
-------------------------------------------------------------------------------
Sorted by:
```

Veneers were placed to match the original contour of the tooth as closely as possible, and researchers were interested in how contour differences (variable `cda`) impacted gingival health. Gingival health was measured as the amount of gingival crevicular fluid (GCF) at each tooth, measured at baseline (variable `base_gcf`) and at two posttreatment follow-ups at 3 and 6 months. The variable `gcf` records GCF at follow-up, and the variable `followup` records the follow-up time.

Because two measurements were taken for each tooth and there exist multiple teeth per patient, we fit a three-level model with the following random effects: a random intercept and random slope on follow-up time at the patient level, and a random intercept at the tooth level. For the ith measurement of the jth tooth from the kth patient, we have

$$\begin{aligned}\texttt{gcf}_{ijk} = \beta_0 + \beta_1\texttt{followup}_{ijk} + \beta_2\texttt{base\_gcf}_{ijk} + \beta_3\texttt{cda}_{ijk} + \beta_4\texttt{age}_{ijk} + \\ u_{0k} + u_{1k}\texttt{followup}_{ijk} + v_{0jk} + \epsilon_{ijk}\end{aligned}$$

which we can fit using `mixed`:

```
. mixed gcf followup base_gcf cda age || patient: followup, cov(un) || tooth:,
> reml nolog
Mixed-effects REML regression                   Number of obs     =        110
```

Group Variable	No. of Groups	Observations per Group Minimum	Average	Maximum
patient	12	2	9.2	12
tooth	55	2	2.0	2

```
                                                Wald chi2(4)      =       7.48
Log restricted-likelihood = -420.92761          Prob > chi2       =     0.1128
```

gcf	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
followup	.3009815	1.936863	0.16	0.877	-3.4952	4.097163
base_gcf	-.0183127	.1433094	-0.13	0.898	-.299194	.2625685
cda	-.329303	.5292525	-0.62	0.534	-1.366619	.7080128
age	-.5773932	.2139656	-2.70	0.007	-.9967582	-.1580283
_cons	45.73862	12.55497	3.64	0.000	21.13133	70.34591

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
patient: Unstructured				
var(followup)	41.88772	18.79997	17.38009	100.9535
var(_cons)	524.9851	253.0205	204.1287	1350.175
cov(followup,_cons)	-140.4229	66.57623	-270.9099	-9.935907
tooth: Identity				
var(_cons)	47.45738	16.63034	23.8792	94.3165
var(Residual)	48.86704	10.50523	32.06479	74.47382

```
LR test vs. linear model: chi2(4) = 91.12                 Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

We used REML estimation for no other reason than variety.

Among the other features of the model fit, we note that the residual variance σ^2_ϵ was estimated as 48.87 and that our model assumed that the residuals were independent with constant variance (homoskedastic). Because it may be the case that the precision of `gcf` measurements could change over time, we modify the above to estimate two distinct error variances: one for the 3-month follow-up and one for the 6-month follow-up.

To fit this model, we add the `residuals(independent, by(followup))` option, which maintains independence of residual errors but allows for heteroskedasticity with respect to follow-up time.

```
. mixed gcf followup base_gcf cda age || patient: followup, cov(un) || tooth:,
> residuals(independent, by(followup)) reml nolog
Mixed-effects REML regression                   Number of obs     =        110
```

Group Variable	No. of Groups	Observations per Group Minimum	Average	Maximum
patient	12	2	9.2	12
tooth	55	2	2.0	2

```
                                                Wald chi2(4)      =       7.51
Log restricted-likelihood =  -420.4576          Prob > chi2       =     0.1113
```

gcf	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
followup	.2703944	1.933096	0.14	0.889	-3.518405	4.059193
base_gcf	.0062144	.1419121	0.04	0.965	-.2719283	.284357
cda	-.2947235	.5245126	-0.56	0.574	-1.322749	.7333023
age	-.5743755	.2142249	-2.68	0.007	-.9942487	-.1545024
_cons	45.15089	12.51452	3.61	0.000	20.62288	69.6789

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
patient: Unstructured				
var(followup)	41.75169	18.72989	17.33099	100.583
var(_cons)	515.2018	251.9661	197.5542	1343.595
cov(followup,_cons)	-139.0496	66.27806	-268.9522	-9.146946
tooth: Identity				
var(_cons)	47.35914	16.48931	23.93514	93.70693
Residual: Independent, by followup				
3 months: var(e)	61.36785	18.38913	34.10946	110.4096
6 months: var(e)	36.42861	14.97501	16.27542	81.53666

```
LR test vs. linear model: chi2(5) = 92.06                 Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

Comparison of both models via an LR test reveals the difference in residual variances to be not significant, something we leave to you to verify as an exercise.

◁

The default residual-variance structure is `independent`, and when specified without `by()` is equivalent to the default behavior of `mixed`: estimating one overall residual standard variance for the entire model.

Other residual-error structures

Besides the default `independent` residual-error structure, `mixed` supports four other structures that allow for correlation between residual errors within the lowest-level (smallest or level two) groups. For purposes of notation, in what follows we assume a two-level model, with the obvious extension to higher-level models.

The exchangeable structure assumes one overall variance and one common pairwise covariance; that is,

$$\mathrm{Var}(\boldsymbol{\epsilon}_j) = \mathrm{Var}\begin{bmatrix}\epsilon_{j1}\\ \epsilon_{j2}\\ \vdots\\ \epsilon_{jn_j}\end{bmatrix} = \begin{bmatrix}\sigma_\epsilon^2 & \sigma_1 & \cdots & \sigma_1\\ \sigma_1 & \sigma_\epsilon^2 & \cdots & \sigma_1\\ \vdots & \vdots & \ddots & \vdots\\ \sigma_1 & \sigma_1 & \sigma_1 & \sigma_\epsilon^2\end{bmatrix}$$

By default, mixed will report estimates of the two parameters as estimates of the common variance σ_ϵ^2 and of the covariance σ_1. When the by(*varname*) option is also specified, these two parameters are estimated for each level *varname*.

The ar p structure assumes that the errors have an AR structure of order p. That is,

$$\epsilon_{ij} = \phi_1\epsilon_{i-1,j} + \cdots + \phi_p\epsilon_{i-p,j} + u_{ij}$$

where u_{ij} are i.i.d. Gaussian with mean 0 and variance σ_u^2. mixed reports estimates of $\phi_1, \ldots, \phi_p$ and the overall error variance σ_ϵ^2, which can be derived from the above expression. The t(*varname*) option is required, where *varname* is a time variable used to order the observations within lowest-level groups and to determine any gaps between observations. When the by(*varname*) option is also specified, the set of $p + 1$ parameters is estimated for each level of *varname*. If $p = 1$, then the estimate of ϕ_1 is reported as rho, because in this case it represents the correlation between successive error terms.

The ma q structure assumes that the errors are an MA process of order q. That is,

$$\epsilon_{ij} = u_{ij} + \theta_1 u_{i-1,j} + \cdots + \theta_q u_{i-q,j}$$

where u_{ij} are i.i.d. Gaussian with mean 0 and variance σ_u^2. mixed reports estimates of $\theta_1, \ldots, \theta_q$ and the overall error variance σ_ϵ^2, which can be derived from the above expression. The t(*varname*) option is required, where *varname* is a time variable used to order the observations within lowest-level groups and to determine any gaps between observations. When the by(*varname*) option is also specified, the set of $q + 1$ parameters is estimated for each level of *varname*.

The unstructured structure is the most general and estimates unique variances and unique pairwise covariances for all residuals within the lowest-level grouping. Because the data may be unbalanced and the ordering of the observations is arbitrary, the t(*varname*) option is required, where *varname* is an identification variable that matches error terms in different groups. If *varname* has n distinct levels, then $n(n + 1)/2$ parameters are estimated. Not all n levels need to be observed within each group, but duplicated levels of *varname* within a given group are not allowed because they would cause a singularity in the estimated error-variance matrix for that group. When the by(*varname*) option is also specified, the set of $n(n + 1)/2$ parameters is estimated for each level of *varname*.

The banded q structure is a special case of unstructured that confines estimation to within the first q off-diagonal elements of the residual variance–covariance matrix and sets the covariances outside this band to 0. As is the case with unstructured, the t(*varname*) option is required, where *varname* is an identification variable that matches error terms in different groups. However, with banded variance structures, the ordering of the values in *varname* is significant because it determines which covariances are to be estimated and which are to be set to 0. For example, if *varname* has $n = 5$ distinct values $t = 1, 2, 3, 4, 5$, then a banded variance–covariance structure of order $q = 2$ would estimate the following:

$$\text{Var}(\boldsymbol{\epsilon}_j) = \text{Var}\begin{bmatrix}\epsilon_{1j}\\ \epsilon_{2j}\\ \epsilon_{3j}\\ \epsilon_{4j}\\ \epsilon_{5j}\end{bmatrix} = \begin{bmatrix}\sigma_1^2 & \sigma_{12} & \sigma_{13} & 0 & 0\\ \sigma_{12} & \sigma_2^2 & \sigma_{23} & \sigma_{24} & 0\\ \sigma_{13} & \sigma_{23} & \sigma_3^2 & \sigma_{34} & \sigma_{35}\\ 0 & \sigma_{24} & \sigma_{34} & \sigma_4^2 & \sigma_{45}\\ 0 & 0 & \sigma_{35} & \sigma_{45} & \sigma_5^2\end{bmatrix}$$

In other words, you would have an unstructured variance matrix that constrains $\sigma_{14} = \sigma_{15} = \sigma_{25} = 0$. If *varname* has n distinct levels, then $(q+1)(2n-q)/2$ parameters are estimated. Not all n levels need to be observed within each group, but duplicated levels of *varname* within a given group are not allowed because they would cause a singularity in the estimated error-variance matrix for that group. When the `by(`*varname*`)` option is also specified, the set of parameters is estimated for each level of *varname*. If q is left unspecified, then `banded` is equivalent to `unstructured`; that is, all variances and covariances are estimated. When $q = 0$, $\text{Var}(\boldsymbol{\epsilon}_j)$ is treated as diagonal and can thus be used to model uncorrelated yet heteroskedastic residual errors.

The `toeplitz` q structure assumes that the residual errors are homoskedastic and that the correlation between two errors is determined by the time lag between the two. That is, $\text{Var}(\epsilon_{ij}) = \sigma_\epsilon^2$ and

$$\text{Corr}(\epsilon_{ij}, \epsilon_{i+k,j}) = \rho_k$$

If the lag k is less than or equal to q, then the pairwise correlation ρ_k is estimated; if the lag is greater than q, then ρ_k is assumed to be 0. If q is left unspecified, then ρ_k is estimated for each observed lag k. The `t(`*varname*`)` option is required, where *varname* is a time variable t used to determine the lags between pairs of residual errors. As such, `t()` must be integer-valued. $q+1$ parameters are estimated: one overall variance σ_ϵ^2 and q correlations. When the `by(`*varname*`)` option is also specified, the set of $q+1$ parameters is estimated for each level of *varname*.

The `exponential` structure is a generalization of the AR structure that allows for noninteger and irregularly spaced time lags. That is, $\text{Var}(\epsilon_{ij}) = \sigma_\epsilon^2$ and

$$\text{Corr}(\epsilon_{ij}, \epsilon_{kj}) = \rho^{|i-k|}$$

for $0 \leq \rho \leq 1$, with i and k not required to be integers. The `t(`*varname*`)` option is required, where *varname* is a time variable used to determine i and k for each residual-error pair. `t()` is real-valued. `mixed` reports estimates of σ_ϵ^2 and ρ. When the `by(`*varname*`)` option is also specified, these two parameters are estimated for each level of *varname*.

▷ Example 8: Autoregressive residual variance structure

Pinheiro and Bates (2000, chap. 5) analyze data from a study of the estrus cycles of mares. Originally analyzed in Pierson and Ginther (1987), the data record the number of ovarian follicles larger than 10mm, daily over a period ranging from three days before ovulation to three days after the subsequent ovulation.

```
. use http://www.stata-press.com/data/r15/ovary
(Ovarian follicles in mares)
. describe
Contains data from http://www.stata-press.com/data/r15/ovary.dta
  obs:           308                          Ovarian follicles in mares
 vars:             6                          20 May 2016 13:49
 size:         5,544                          (_dta has notes)
```

variable name	storage type	display format	value label	variable label
mare	byte	%9.0g		mare ID
stime	float	%9.0g		Scaled time
follicles	byte	%9.0g		Number of ovarian follicles > 10 mm in diameter
sin1	float	%9.0g		sine(2*pi*stime)
cos1	float	%9.0g		cosine(2*pi*stime)
time	float	%9.0g		time order within mare

```
Sorted by: mare  stime
```

The `stime` variable is time that has been scaled so that ovulation occurs at scaled times 0 and 1, and the `time` variable records the time ordering within mares. Because graphical evidence suggests a periodic behavior, the analysis includes the `sin1` and `cos1` variables, which are sine and cosine transformations of scaled time, respectively.

We consider the following model for the ith measurement on the jth mare:

$$\texttt{follicles}_{ij} = \beta_0 + \beta_1 \texttt{sin1}_{ij} + \beta_2 \texttt{cos1}_{ij} + u_j + \epsilon_{ij}$$

The above model incorporates the cyclical nature of the data as affecting the overall average number of follicles and includes mare-specific random effects u_j. Because we believe successive measurements within each mare are probably correlated (even after controlling for the periodicity in the average), we also model the within-mare errors as being AR of order 2.

```
. mixed follicles sin1 cos1 || mare:, residuals(ar 2, t(time)) reml nolog
Mixed-effects REML regression                   Number of obs     =        308
Group variable: mare                            Number of groups  =         11
                                                Obs per group:
                                                              min =         25
                                                              avg =       28.0
                                                              max =         31
                                                Wald chi2(2)      =      34.72
Log restricted-likelihood = -772.59855          Prob > chi2       =     0.0000
```

follicles	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
sin1	-2.899227	.5110786	-5.67	0.000	-3.900923	-1.897532
cos1	-.8652936	.5432925	-1.59	0.111	-1.930127	.1995402
_cons	12.14455	.9473712	12.82	0.000	10.28774	14.00136

```
------------------------------------------------------------------------------
  Random-effects Parameters |   Estimate   Std. Err.      [95% Conf. Interval]
----------------------------+-------------------------------------------------
mare: Identity              |
                  var(_cons) |   7.092607   4.402031      2.101392    23.93892
----------------------------+-------------------------------------------------
Residual: AR(2)             |
                       phi1 |    .5386104   .0624897      .4161329     .661088
                       phi2 |    .1446712   .0632039      .0207939    .2685486
                    var(e)  |   14.25104   2.435233      10.19512    19.92052
------------------------------------------------------------------------------
LR test vs. linear model: chi2(3) = 251.67                Prob > chi2 = 0.0000

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

We picked an order of 2 as a guess, but we could have used LR tests of competing AR models to determine the optimal order, because models of smaller order are nested within those of larger order. ◁

▷ Example 9: Unstructured residual variance structure

Fitzmaurice, Laird, and Ware (2011, chap. 7) analyzed data on 37 subjects who participated in an exercise therapy trial.

```
. use http://www.stata-press.com/data/r15/exercise
(Exercise Therapy Trial)

. describe

Contains data from http://www.stata-press.com/data/r15/exercise.dta
  obs:           259                          Exercise Therapy Trial
 vars:             4                          24 Jun 2016 18:35
 size:         1,036                          (_dta has notes)
------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
------------------------------------------------------------------------------
id              byte    %9.0g                 Person ID
day             byte    %9.0g                 Day of measurement
program         byte    %9.0g                 1 = reps increase; 2 = weights
                                                increase
strength        byte    %9.0g                 Strength measurement
------------------------------------------------------------------------------
Sorted by: id  day
```

Subjects (variable `id`) were placed on either an increased-repetition regimen (`program==1`) or a program that kept the repetitions constant but increased weight (`program==2`). Muscle-strength measurements (variable `strength`) were taken at baseline (`day==0`) and then every two days over the next twelve days.

Following Fitzmaurice, Laird, and Ware (2011, chap. 7), and to demonstrate fitting residual-error structures to data collected at uneven time points, we confine our analysis to those data collected at baseline and at days 4, 6, 8, and 12. We fit a full two-way factorial model of `strength` on `program` and `day`, with an unstructured residual-error covariance matrix over those repeated measurements taken on the same subject:

```
. keep if inlist(day, 0, 4, 6, 8, 12)
(74 observations deleted)
. mixed strength i.program##i.day || id:,
> noconstant residuals(unstructured, t(day)) nolog
Mixed-effects ML regression                     Number of obs     =        173
Group variable: id                              Number of groups  =         37
                                                Obs per group:
                                                              min =          3
                                                              avg =        4.7
                                                              max =          5
                                                Wald chi2(9)      =      45.85
Log likelihood = -296.58215                     Prob > chi2       =     0.0000
```

strength	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
2.program	1.360119	1.003549	1.36	0.175	-.6068016	3.32704
day						
4	1.125	.3322583	3.39	0.001	.4737858	1.776214
6	1.360127	.3766894	3.61	0.000	.6218298	2.098425
8	1.583563	.4905876	3.23	0.001	.6220287	2.545097
12	1.623576	.5372947	3.02	0.003	.5704977	2.676654
program#day						
2 4	-.169034	.4423472	-0.38	0.702	-1.036019	.6979506
2 6	.2113012	.4982385	0.42	0.671	-.7652283	1.187831
2 8	-.1299763	.6524813	-0.20	0.842	-1.408816	1.148864
2 12	.3212829	.7306782	0.44	0.660	-1.11082	1.753386
_cons	79.6875	.7560448	105.40	0.000	78.20568	81.16932

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
id: (empty)				
Residual: Unstructured				
var(e0)	9.14566	2.126233	5.798599	14.42471
var(e4)	11.87114	2.761187	7.524987	18.72747
var(e6)	10.06571	2.348835	6.371125	15.90275
var(e8)	13.22464	3.113885	8.336026	20.98014
var(e12)	13.16909	3.167316	8.219245	21.09985
cov(e0,e4)	9.625235	2.331946	5.054705	14.19577
cov(e0,e6)	8.489042	2.106352	4.360668	12.61742
cov(e0,e8)	9.280413	2.369524	4.636232	13.92459
cov(e0,e12)	8.898006	2.348212	4.295594	13.50042
cov(e4,e6)	10.49184	2.492498	5.606639	15.37705
cov(e4,e8)	11.89787	2.848714	6.314492	17.48125
cov(e4,e12)	11.28344	2.804991	5.78576	16.78112
cov(e6,e8)	11.0507	2.646955	5.862762	16.23863
cov(e6,e12)	10.5006	2.590246	5.423812	15.57739
cov(e8,e12)	12.4091	3.010761	6.508121	18.31009

```
LR test vs. linear model: chi2(14) = 314.67                   Prob > chi2 = 0.0000
Note: The reported degrees of freedom assumes the null hypothesis is not on
      the boundary of the parameter space.  If this is not true, then the
      reported test is conservative.
```

Because we are using the variable `id` only to group the repeated measurements and not to introduce random effects at the subject level, we use the `noconstant` option to omit any subject-level effects.

The unstructured covariance matrix is the most general and contains many parameters. In this example, we estimate a distinct residual variance for each day and a distinct covariance for each pair of days.

That there is positive covariance between all pairs of measurements is evident, but what is not as evident is whether the covariances may be more parsimoniously represented. One option would be to explore whether the correlation diminishes as the time gap between strength measurements increases and whether it diminishes systematically. Given the irregularity of the time intervals, an exponential structure would be more appropriate than, say, an AR or MA structure.

```
. estimates store unstructured

. mixed strength i.program##i.day || id:, noconstant
> residuals(exponential, t(day)) nolog nofetable
Mixed-effects ML regression                     Number of obs     =        173
Group variable: id                              Number of groups  =         37
                                                Obs per group:
                                                              min =          3
                                                              avg =        4.7
                                                              max =          5
                                                Wald chi2(9)      =      36.77
Log likelihood = -307.83324                     Prob > chi2       =     0.0000
```

Random-effects Parameters		Estimate	Std. Err.	[95% Conf.	Interval]
id:	(empty)				
Residual: Exponential					
	rho	.9786462	.0051238	.9659207	.9866854
	var(e)	11.22349	2.338371	7.460762	16.88389

```
LR test vs. linear model: chi2(1) = 292.17                Prob > chi2 = 0.0000

Note: The reported degrees of freedom assumes the null hypothesis is not on
      the boundary of the parameter space.  If this is not true, then the
      reported test is conservative.
```

In the above example, we suppressed displaying the main regression parameters because they did not differ much from those of the previous model. While the unstructured model estimated 15 variance–covariance parameters, the exponential model claims to get the job done with just 2, a fact that is not disputed by an LR test comparing the two nested models (at least not at the 0.01 level).

```
. lrtest unstructured .

Likelihood-ratio test                                 LR chi2(13) =     22.50
(Assumption: . nested in unstructured)                Prob > chi2 =    0.0481

Note: The reported degrees of freedom assumes the null hypothesis is not on
      the boundary of the parameter space.  If this is not true, then the
      reported test is conservative.
```

◁

Crossed-effects models

Not all mixed models contain nested levels of random effects.

▷ Example 10: Crossed-effects model

Returning to our longitudinal analysis of pig weights, suppose that instead of (5) we wish to fit

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

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

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

all independently. Both (5) and (8) assume an overall population-average growth curve $\beta_0 + \beta_1 \texttt{week}$ and a random pig-specific shift.

The models differ in how `week` enters into the random part of the model. In (5), we assume that the effect due to `week` is linear and pig specific (a random slope); in (8), we assume that the effect due to `week`, u_i, is systematic to that week and common to all pigs. The rationale behind (8) could be that, assuming that the pigs were measured contemporaneously, we might be concerned that week-specific random factors such as weather and feeding patterns had significant systematic effects on all pigs.

Model (8) is an example of a two-way crossed-effects model, with the pig effects v_j being crossed with the week effects u_i. One way to fit such models is to consider all the data as one big cluster, and treat the u_i and v_j as a series of $9 + 48 = 57$ random coefficients on indicator variables for `week` and `pig`. In the notation of (2),

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

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

```
. use http://www.stata-press.com/data/r15/pig, clear
(Longitudinal analysis of pig weights)
. mixed weight week || _all: R.week || _all: R.id
Performing EM optimization:
Performing gradient-based optimization:
Iteration 0:   log likelihood =  -1013.824
Iteration 1:   log likelihood =  -1013.824
Computing standard errors:
Mixed-effects ML regression                     Number of obs     =        432
Group variable: _all                            Number of groups  =          1
                                                Obs per group:
                                                              min =        432
                                                              avg =      432.0
                                                              max =        432
                                                Wald chi2(1)      =   13258.28
Log likelihood =  -1013.824                     Prob > chi2       =     0.0000
```

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

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
_all: Identity var(R.week)	.0849874	.0868856	.0114588	.6303302
_all: Identity var(R.id)	14.83623	3.126142	9.816733	22.42231
var(Residual)	4.297328	.3134404	3.724888	4.957741

```
LR test vs. linear model: chi2(2) = 474.85                Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
. estimates store crossed
```

Thus we estimate $\widehat{\sigma}_u^2 = 0.08$ and $\widehat{\sigma}_v^2 = 14.84$. Both (5) and (8) estimate a total of five parameters: two fixed effects and three variance components. The models, however, are not nested within each other, which precludes the use of an LR test to compare both models. Refitting model (5) and looking at the Akaike information criteria values by using `estimates stats`,

```
. quietly mixed weight week || id:week
. estimates stats crossed .
Akaike's information criterion and Bayesian information criterion
```

Model	Obs	ll(null)	ll(model)	df	AIC	BIC
crossed	432	.	-1013.824	5	2037.648	2057.99
.	432	.	-869.0383	5	1748.077	1768.419

Note: N=Obs used in calculating BIC; see **[R] BIC note**.

definitely favors model (5). This finding is not surprising given that our rationale behind (8) was somewhat fictitious. In our `estimates stats` output, the values of `ll(null)` are missing. `mixed` does not fit a constant-only model as part of its usual estimation of the full model, but you can use `mixed` to fit a constant-only model directly, if you wish.

◁

The R.*varname* notation is equivalent to giving a list of overparameterized (none dropped) indicator variables for use in a random-effects specification. When you specify R.*varname*, mixed handles the calculations internally rather than creating the indicators in the data. Because the set of indicators is overparameterized, R.*varname* implies noconstant. You can include factor variables in the fixed-effects specification by using standard methods; see [U] **11.4.3 Factor variables**. However, random-effects equations support only the R.*varname* factor specification. For more complex factor specifications (such as interactions) in random-effects equations, use generate to form the variables manually, as we demonstrated in example 6.

❑ Technical note

Although we were able to fit the crossed-effects model (8), it came at the expense of increasing the column dimension of our random-effects design from 2 in model (5) to 57 in model (8). Computation time and memory requirements grow (roughly) quadratically with the dimension of the random effects. As a result, fitting such crossed-effects models is feasible only when the total column dimension is small to moderate.

Reexamining model (8), we note that if we drop u_i, we end up with a model equivalent to (4), meaning that we could have fit (4) by typing

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

instead of

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

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

Furthermore, we fit model (8) by using

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

as a direct way to demonstrate the R. notation. However, we can technically treat pigs as nested within the _all group, yielding the equivalent and more efficient (total column dimension 10) way to fit (8):

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

We leave it to you to verify that both produce identical results. See Rabe-Hesketh and Skrondal (2012) for additional techniques to make calculations more efficient in more complex models.

❑

▷ Example 11: Three-level model expressed in terms of a two-level model

As another example of how the same model may be fit in different ways by using mixed (and as a way to demonstrate covariance(exchangeable)), consider the three-level model used in example 4:

$$\mathbf{y}_{jk} = \mathbf{X}_{jk}\boldsymbol{\beta} + u_k^{(3)} + u_{jk}^{(2)} + \boldsymbol{\epsilon}_{jk}$$

where $\mathbf{y}_{jk}$ represents the logarithms of gross state products for the $n_{jk} = 17$ observations from state j in region k, $\mathbf{X}_{jk}$ is a set of regressors, $u_k^{(3)}$ is a random intercept at the region level, and $u_{jk}^{(2)}$ is a random intercept at the state (nested within region) level. We assume that $u_k^{(3)} \sim N(0, \sigma_3^2)$ and $u_{jk}^{(2)} \sim N(0, \sigma_2^2)$ independently. Define

$$\mathbf{v}_k = \begin{bmatrix} u_k^{(3)} + u_{1k}^{(2)} \\ u_k^{(3)} + u_{2k}^{(2)} \\ \vdots \\ u_k^{(3)} + u_{M_k,k}^{(2)} \end{bmatrix}$$

where M_k is the number of states in region k. Making this substitution, we can stack the observations for all the states within region k to get

$$\mathbf{y}_k = \mathbf{X}_k\boldsymbol{\beta} + \mathbf{Z}_k\mathbf{v}_k + \boldsymbol{\epsilon}_k$$

where $\mathbf{Z}_k$ is a set of indicators identifying the states within each region; that is,

$$\mathbf{Z}_k = \mathbf{I}_{M_k} \otimes \mathbf{J}_{17}$$

for a k-column vector of 1s $\mathbf{J}_k$, and

$$\boldsymbol{\Sigma} = \mathrm{Var}(\mathbf{v}_k) = \begin{bmatrix} \sigma_3^2 + \sigma_2^2 & \sigma_3^2 & \cdots & \sigma_3^2 \\ \sigma_3^2 & \sigma_3^2 + \sigma_2^2 & \cdots & \sigma_3^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_3^2 & \sigma_3^2 & \sigma_3^2 & \sigma_3^2 + \sigma_2^2 \end{bmatrix}_{M_k \times M_k}$$

Because $\boldsymbol{\Sigma}$ is an exchangeable matrix, we can fit this alternative form of the model by specifying the `exchangeable` covariance structure.

```
. use http://www.stata-press.com/data/r15/productivity
(Public Capital Productivity)
. mixed gsp private emp hwy water other unemp || region: R.state,
> cov(exchangeable)
```

(*output omitted*)

```
Mixed-effects ML regression                     Number of obs     =        816
Group variable: region                          Number of groups  =          9
                                                Obs per group:
                                                              min =         51
                                                              avg =       90.7
                                                              max =        136
                                                Wald chi2(6)      =   18829.06
Log likelihood =  1430.5017                     Prob > chi2       =     0.0000
```

gsp	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
private	.2671484	.0212591	12.57	0.000	.2254813	.3088154
emp	.7540721	.0261868	28.80	0.000	.7027468	.8053973
hwy	.0709767	.023041	3.08	0.002	.0258172	.1161363
water	.0761187	.0139248	5.47	0.000	.0488266	.1034109
other	-.0999955	.0169366	-5.90	0.000	-.1331907	-.0668004
unemp	-.0058983	.0009031	-6.53	0.000	-.0076684	-.0041282
_cons	2.128823	.1543855	13.79	0.000	1.826233	2.431413

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
region: Exchangeable				
var(R.state)	.0077263	.0017926	.0049032	.0121749
cov(R.state)	.0014506	.0012995	-.0010963	.0039975
var(Residual)	.0013461	.0000689	.0012176	.0014882

```
LR test vs. linear model: chi2(2) = 1154.73                   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

The estimates of the fixed effects and their standard errors are equivalent to those from example 4, and remapping the variance components from $(\sigma_3^2 + \sigma_2^2, \sigma_3^2, \sigma_\epsilon^2)$, as displayed here, to $(\sigma_3^2, \sigma_2^2, \sigma_\epsilon^2)$, as displayed in example 4, will show that they are equivalent as well.

Of course, given the discussion in the previous technical note, it is more efficient to fit this model as we did originally, as a three-level model.

◁

Diagnosing convergence problems

Given the flexibility of mixed-effects models, you will find that some models fail to converge when used with your data; see *Diagnosing convergence problems* in [ME] **me** for advice applicable to mixed-effects models in general.

In unweighted LME models with independent and homoskedastic residuals, one useful way to diagnose problems of nonconvergence is to rely on the EM algorithm (Dempster, Laird, and Rubin 1977), normally used by `mixed` only as a means of refining starting values. The advantages of EM are that it does not require a Hessian calculation, each successive EM iteration will result in a larger likelihood, iterations can be calculated quickly, and iterations will quickly bring parameter estimates into a

neighborhood of the solution. The disadvantages of EM are that, once in a neighborhood of the solution, it can be slow to converge, if at all, and EM provides no facility for estimating standard errors of the estimated variance components. One useful property of EM is that it is always willing to provide a solution if you allow it to iterate enough times, if you are satisfied with being in a neighborhood of the optimum rather than right on the optimum, and if standard errors of variance components are not crucial to your analysis.

If you encounter a nonconvergent model, try using the `emonly` option to bypass gradient-based optimization. Use `emiterate(#)` to specify the maximum number of EM iterations, which you will usually want to set much higher than the default of 20. If your EM solution shows an estimated variance component that is near 0, a ridge is formed by an interval of values near 0, which produces the same likelihood and looks equally good to the optimizer. In this case, the solution is to drop the offending variance component from the model.

Survey data

Multilevel modeling of survey data is a little different from standard modeling in that weighted sampling can take place at multiple levels in the model, resulting in multiple sampling weights. Most survey datasets, regardless of the design, contain one overall inclusion weight for each observation in the data. This weight reflects the inverse of the probability of ultimate selection, and by "ultimate" we mean that it factors in all levels of clustered sampling, corrections for noninclusion and oversampling, poststratification, etc.

For simplicity, in what follows assume a simple two-stage sampling design where groups are randomly sampled and then individuals within groups are sampled. Also assume that no additional weight corrections are performed; that is, sampling weights are simply the inverse of the probability of selection. The sampling weight for observation i in cluster j in our two-level sample is then $w_{ij} = 1/\pi_{ij}$, where π_{ij} is the probability that observation i, j is selected. If you were performing a standard analysis such as OLS regression with `regress`, you would simply use a variable holding w_{ij} as your `pweight` variable, and the fact that it came from two levels of sampling would not concern you. Perhaps you would type `vce(cluster` *groupvar*`)` where *groupvar* identifies the top-level groups to get standard errors that control for correlation within these groups, but you would still use only a single weight variable.

Now take these same data and fit a two-level model with `mixed`. As seen in (14) in *Methods and formulas* later in this entry, it is not sufficient to use the single sampling weight w_{ij}, because weights enter into the log likelihood at both the group level and the individual level. Instead, what is required for a two-level model under this sampling design is w_j, the inverse of the probability that group j is selected in the first stage, and $w_{i|j}$, the inverse of the probability that individual i from group j is selected at the second stage *conditional on group j already being selected*. It simply will not do to just use w_{ij} without making any assumptions about w_j.

Given the rules of conditional probability, $w_{ij} = w_j w_{i|j}$. If your dataset has only w_{ij}, then you will need to either assume equal probability sampling at the first stage ($w_j = 1$ for all j) or find some way to recover w_j from other variables in your data; see Rabe-Hesketh and Skrondal (2006) and the references therein for some suggestions on how to do this, but realize that there is little yet known about how well these approximations perform in practice.

What you really need to fit your two-level model are data that contain w_j in addition to either w_{ij} or $w_{i|j}$. If you have w_{ij}—that is, the unconditional inclusion weight for observation i, j—then you need to either divide w_{ij} by w_j to obtain $w_{i|j}$ or rescale w_{ij} so that its dependence on w_j disappears. If you already have $w_{i|j}$, then rescaling becomes optional (but still an important decision to make).

Weight rescaling is not an exact science, because the scale of the level-one weights is at issue regardless of whether they represent w_{ij} or $w_{i|j}$: because w_{ij} is unique to group j, the group-to-group magnitudes of these weights need to be normalized so that they are "consistent" from group to group. This is in stark contrast to a standard analysis, where the scale of sampling weights does not factor into estimation, instead only affecting the estimate of the total population size.

`mixed` offers three methods for standardizing weights in a two-level model, and you can specify which method you want via the `pwscale()` option. If you specify `pwscale(size)`, then the $w_{i|j}$ (or w_{ij}, it does not matter) are scaled to sum to the cluster size n_j. Method `pwscale(effective)` adds in a dependence on the sum of the squared weights so that level-one weights sum to the "effective" sample size. Just like `pwscale(size)`, `pwscale(effective)` also behaves the same whether you have $w_{i|j}$ or w_{ij}, and so it can be used with either.

Although both `pwscale(size)` and `pwscale(effective)` leave w_j untouched, the `pwscale(gk)` method is a little different in that 1) it changes the weights at both levels and 2) it does assume you have $w_{i|j}$ for level-one weights and not w_{ij} (if you have the latter, then first divide by w_j). Using the method of Graubard and Korn (1996), it sets the weights at the group level (level two) to the cluster averages of the products of both level weights (this product being w_{ij}). It then sets the individual weights to 1 everywhere; see *Methods and formulas* for the computational details of all three methods.

Determining which method is "best" is a tough call and depends on cluster size (the smaller the clusters, the greater the sensitivity to scale), whether the sampling is informative (that is, the sampling weights are correlated with the residuals), whether you are interested primarily in regression coefficients or in variance components, whether you have a simple random-intercept model or a more complex random-coefficients model, and other factors; see Rabe-Hesketh and Skrondal (2006), Carle (2009), and Pfeffermann et al. (1998) for some detailed advice. At the very least, you want to compare estimates across all three scaling methods (four, if you add no scaling) and perform a sensitivity analysis.

If you choose to rescale level-one weights, it does not matter whether you have $w_{i|j}$ or w_{ij}. For the `pwscale(size)` and `pwscale(effective)` methods, you get identical results, and even though `pwscale(gk)` assumes $w_{i|j}$, you can obtain this as $w_{i|j} = w_{ij}/w_j$ before proceeding.

If you do not specify `pwscale()`, then no scaling takes place, and thus at a minimum, you need to make sure you have $w_{i|j}$ in your data and not w_{ij}.

▷ Example 12: Mixed-effect models with survey data

Rabe-Hesketh and Skrondal (2006) analyzed data from the 2000 Programme for International Student Assessment (PISA) study on reading proficiency among 15-year-old American students, as performed by the Organisation for Economic Co-operation and Development (OECD). The original study was a three-stage cluster sample, where geographic areas were sampled at the first stage, schools at the second, and students at the third. Our version of the data does not contain the geographic-areas variable, so we treat this as a two-stage sample where schools are sampled at the first stage and students at the second.

```
. use http://www.stata-press.com/data/r15/pisa2000
(Programme for International Student Assessment (PISA) 2000 data)
. describe
Contains data from http://www.stata-press.com/data/r15/pisa2000.dta
  obs:         2,069                          Programme for International
                                                Student Assessment (PISA) 2000
                                                data
 vars:            11                          12 Jun 2016 10:08
 size:        37,242                          (_dta has notes)
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
female          byte    %8.0g                 1 if female
isei            byte    %8.0g                 International socio-economic index
w_fstuwt        float   %9.0g                 Student-level weight
wnrschbw        float   %9.0g                 School-level weight
high_school     byte    %8.0g                 1 if highest level by either
                                                parent is high school
college         byte    %8.0g                 1 if highest level by either
                                                parent is college
one_for         byte    %8.0g                 1 if one parent foreign born
both_for        byte    %8.0g                 1 if both parents are foreign born
test_lang       byte    %8.0g                 1 if English (the test language)
                                                is spoken at home
pass_read       byte    %8.0g                 1 if passed reading proficiency
                                                threshold
id_school       int     %8.0g                 School ID
-------------------------------------------------------------------------------
Sorted by:
```

For student i in school j, where the variable `id_school` identifies the schools, the variable `w_fstuwt` is a student-level overall inclusion weight (w_{ij}, not $w_{i|j}$) adjusted for noninclusion and nonparticipation of students, and the variable `wnrschbw` is the school-level weight w_j adjusted for oversampling of schools with more minority students. The weight adjustments do not interfere with the methods prescribed above, and thus we can treat the weight variables simply as w_{ij} and w_j, respectively.

Rabe-Hesketh and Skrondal (2006) fit a two-level logistic model for passing a reading proficiency threshold. We fit a two-level linear random-intercept model for socioeconomic index. Because we have w_{ij} and not $w_{i|j}$, we rescale using `pwscale(size)` and thus obtain results as if we had $w_{i|j}$.

```
. mixed isei female high_school college one_for both_for test_lang
> [pw=w_fstuwt] || id_school:, pweight(wnrschbw) pwscale(size)
```

(*output omitted*)

```
Mixed-effects regression                        Number of obs     =      2,069
Group variable: id_school                       Number of groups  =        148

                                                Obs per group:
                                                              min =          1
                                                              avg =       14.0
                                                              max =         28

                                                Wald chi2(6)      =     187.23
Log pseudolikelihood = -1443093.9               Prob > chi2       =     0.0000

                            (Std. Err. adjusted for 148 clusters in id_school)
```

isei	Coef.	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
female	.59379	.8732886	0.68	0.497	-1.117824	2.305404
high_school	6.410618	1.500337	4.27	0.000	3.470011	9.351224
college	19.39494	2.121145	9.14	0.000	15.23757	23.55231
one_for	-.9584613	1.789947	-0.54	0.592	-4.466692	2.54977
both_for	-.2021101	2.32633	-0.09	0.931	-4.761633	4.357413
test_lang	2.519539	2.393165	1.05	0.292	-2.170978	7.210056
_cons	28.10788	2.435712	11.54	0.000	23.33397	32.88179

Random-effects Parameters	Estimate	Robust Std. Err.	[95% Conf.	Interval]
id_school: Identity				
var(_cons)	34.69374	8.574865	21.37318	56.31617
var(Residual)	218.7382	11.22111	197.8147	241.8748

Notes:

1. We specified the level-one weights using standard Stata weight syntax, that is, `[pw=w_fstuwt]`.
2. We specified the level-two weights via the `pweight(wnrschbw)` option as part of the random-effects specification for the `id_school` level. As such, it is treated as a school-level weight. Accordingly, `wnrschbw` needs to be constant within schools, and `mixed` did check for that before estimating.
3. Because our level-one weights are unconditional, we specified `pwscale(size)` to rescale them.
4. As is the case with other estimation commands in Stata, standard errors in the presence of sampling weights are robust.
5. Robust standard errors are clustered at the top level of the model, and this will always be true unless you specify `vce(cluster` *clustvar*`)`, where *clustvar* identifies an even higher level of grouping.

As a form of sensitivity analysis, we compare the above with scaling via `pwscale(gk)`. Because `pwscale(gk)` assumes $w_{i|j}$, you want to first divide w_{ij} by w_j. But you can handle that within the weight specification itself.

```
. mixed isei female high_school college one_for both_for test_lang
> [pw=w_fstuwt/wnrschbw] || id_school:, pweight(wnrschbw) pwscale(gk)
  (output omitted)
Mixed-effects regression                        Number of obs     =      2,069
Group variable: id_school                       Number of groups  =        148
                                                Obs per group:
                                                              min =          1
                                                              avg =       14.0
                                                              max =         28
                                                Wald chi2(6)      =     291.37
Log pseudolikelihood = -7270505.6               Prob > chi2       =     0.0000
                          (Std. Err. adjusted for 148 clusters in id_school)
```

isei	Coef.	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
female	-.3519458	.7436334	-0.47	0.636	-1.80944	1.105549
high_school	7.074911	1.139777	6.21	0.000	4.84099	9.308833
college	19.27285	1.286029	14.99	0.000	16.75228	21.79342
one_for	-.9142879	1.783091	-0.51	0.608	-4.409082	2.580506
both_for	1.214151	1.611885	0.75	0.451	-1.945085	4.373388
test_lang	2.661866	1.556491	1.71	0.087	-.3887996	5.712532
_cons	31.20145	1.907413	16.36	0.000	27.46299	34.93991

Random-effects Parameters	Estimate	Robust Std. Err.	[95% Conf.	Interval]
id_school: Identity				
var(_cons)	31.67522	6.792239	20.80622	48.22209
var(Residual)	226.2429	8.150714	210.8188	242.7955

The results are somewhat similar to before, which is good news from a sensitivity standpoint. Note that we specified `[pw=w_fstwtw/wnrschbw]` and thus did the conversion from w_{ij} to $w_{i|j}$ within our call to `mixed`.

◁

We close this section with a bit of bad news. Although weight rescaling and the issues that arise have been well studied for two-level models, as pointed out by Carle (2009), "... a best practice for scaling weights across multiple levels has yet to be advanced." As such, `pwscale()` is currently supported only for two-level models. If you are fitting a higher-level model with survey data, you need to make sure your sampling weights are conditional on selection at the previous stage and not overall inclusion weights, because there is currently no rescaling option to fall back on if you do not.

Small-sample inference for fixed effects

Researchers are often interested in making inferences about fixed effects in a linear mixed-effects model. In the special case where the data are balanced and the mixed-effects model has a simple covariance structure, the sampling distributions of the statistics for testing hypotheses about fixed effects are known to follow an F distribution with specific denominator degrees of freedom (DDF) under the null hypothesis. For example, the test statistics for testing hypotheses about fixed effects in balanced split-plot designs and balanced repeated-measures designs have exact t or F distributions. In general, however, the null sampling distributions of test statistics for fixed effects are not known and can only be approximated in more complicated mixed-effects models.

For a large sample, the null sampling distributions of the test statistics can be approximated by a normal distribution for a one-hypothesis test and a χ^2 distribution for a multiple-hypotheses test. This is the default behavior of `mixed`. However, these large-sample approximations may not be appropriate in small samples, and t and F distributions may provide better approximations.

You can specify the `dfmethod()` option to request small-sample inference for fixed effects. `mixed` with the `dfmethod()` option uses a t distribution for one-hypothesis tests and an F distribution for multiple-hypotheses tests for inference about fixed effects. We use DF to refer to degrees of freedom of a t distribution, and we use DDF to refer to denominator degrees of freedom of an F distribution.

Researchers have proposed various approximations that use t and F distributions but differ in how respective DF and DDF are computed (for example, Khuri, Mathew, and Sinha [1998]; Brown and Prescott [2015]; Schluchter and Elashoff [1990]; Elston [1998]; Kackar and Harville [1984]; Giesbrecht and Burns [1985]; Fai and Cornelius [1996]; and Kenward and Roger [1997, 2009]). `mixed` provides five methods with the `dfmethod()` option for calculating the DF of a t distribution: `residual`, `repeated`, `anova`, `satterthwaite`, and `kroger`.

Residual DDF (DF). This method uses the residual degrees of freedom, $n - p$, as the DDF for all tests of fixed effects. For a linear model without random effects and with i.i.d errors, the distributions of the test statistics for testing the fixed effects are exact t or F distributions with the residual DF.

Repeated DDF (DF). This method partitions the residual degrees of freedom into the between-subject degrees of freedom and the within-subject degrees of freedom. This partitioning of the degrees of freedom arises from balanced repeated-measures ANOVA analysis. If levels of a fixed effect change within a subject, then the within-subject degrees of freedom is assigned to the fixed effect of interest; otherwise, the between-subject degrees of freedom is assigned to that fixed effect. Winer, Brown, and Michels (1991) showed that this method is appropriate only when the data are balanced and the correlation structure is assumed to be spherical. The repeated DDF method is supported only with two-level models. For DDF methods accounting for unbalanced repeated measures, see, for example, Schluchter and Elashoff (1990).

ANOVA DDF (DF). This method mimics the traditional ANOVA method. It determines the DDF for a fixed effect depending on whether the corresponding covariate is contained in any of the random-effects equations. If the covariate is contained in a random-effects equation, the DDF for the fixed effect is computed as the number of levels of the level variable from that equation minus one. If the covariate is specified in more than one random-effects equation, the DDF for the fixed effect is computed as the smallest number of levels of the level variables from those equations minus one and is a conservative estimate of the true DDF. If the covariate is specified only in the fixed-effects equation, the DDF is computed as $\nu_{\text{ddf}} = n - \text{rank}(\mathbf{X}, \mathbf{Z})$. This method leads to an exact sampling distribution of the test statistics only when random effects are balanced and the residuals are i.i.d; see, for example, chapter 1.6 in Brown and Prescott (2015) for details.

Satterthwaite DDF (DF). This method performs a generalization of the Satterthwaite approximation based on Kackar and Harville (1984), Giesbrecht and Burns (1985), and Fai and Cornelius (1996). Giesbrecht and Burns (1985) developed a method of computing the DDF for a single-hypothesis test that is analogous to Satterthwaite's approximation of the degrees of freedom of a linear combination of ANOVA mean squares. For a multiple-hypotheses test, Fai and Cornelius (1996) proposed an extension of the Giesbrecht–Burns single-degree-of-freedom method. This method involves the spectral decomposition of the contrast matrix of the hypothesis test and repeated application of the single-degree-of-freedom t test. See *Denominator degrees of freedom* in *Methods and formulas* for more computational details.

Kenward–Roger DDF (DF). This method, developed by Kenward and Roger (1997), was designed to provide an approximation that improves the performance of hypothesis tests about fixed effects in small samples for complicated mixed-effects models and reproduces the exact inference available

for simpler mixed-effects models. It provides adjusted test statistics, more appropriate DDFs for the approximate F distributions when exact inference is not available, and yields the exact t and F distributions when exact inference is available. This method first accounts for the small-sample bias and the variability of the estimated random effects to obtain an adjusted estimator of the fixed-effects covariance matrix. Then, it proposes an approximate F test based on a scaled Wald test statistic that uses the adjusted variance–covariance estimator. See *Denominator degrees of freedom* in *Methods and formulas* for more computational details.

Residual, repeated, and ANOVA are known as "exact" methods in the literature. These methods are suitable only when the sampling distributions of the test statistics are known to be t or F. This is usually only known for certain classes of linear mixed-effects models with simple covariance structures and when data are balanced. These methods are available with both ML and REML estimation.

Satterthwaite and Kenward–Roger are known as "approximation" methods in the literature. These methods are for unbalanced data and complicated covariance structures where the sampling distributions of test statistics are unknown and can only be approximated. Both methods are available only with REML estimation. For single-hypothesis tests, DDFs calculated with the Kenward–Roger method are the same as those calculated with the Satterthwaite method, but they differ for multiple-hypotheses tests. Although DDFs of the two methods are the same for single-hypothesis tests, the inference is not the same because the Kenward–Roger method uses bias-adjusted standard errors.

Except for the special cases for which the sampling distributions are known, there is no definitive recommendation for which approximation performs best. Schaalje, McBride, and Fellingham (2002) compared the Satterthwaite method with the Kenward–Roger method via simulation using different covariance structures and various sample sizes. They concluded that the Kenward–Roger method outperforms the Satterthwaite method in most situations. They recommend using the Satterthwaite method only when the covariance structure of the data is compound symmetry and the sample size is moderately large. The Kenward–Roger method, however, is not guaranteed to work well in all situations. For example, for more complicated covariance structures and very small-sample sizes, the Kenward–Roger method may produce inflated type I error rates. In conclusion, you should choose your DDF method carefully. See, for example, Schaalje, McBride, and Fellingham (2002), Chen and Wei (2003), Vallejo et al. (2004), and West, Welch, and Gałecki (2015) for a comparison of different approximations.

Both types of methods, exact and approximation, are available for single-hypothesis tests. For multiple-hypotheses tests, exact methods are available only if DDFs associated with fixed effects are the same for all tested covariates. See *Denominator degrees of freedom* in *Methods and formulas* for details.

▷ Example 13: Small-sample inference with a balanced repeated-measures design

Consider an example from Winer, Brown, and Michels (1991, table 4.3), also analyzed in example 15 of [R] **anova**, which reports the reaction time for five subjects who were tested with four drugs. The reaction time was recorded in the variable `score`. Assume that `person` is random (that is, we wish to infer to the larger population of possible subjects) and `drug` is fixed (that is, only four drugs are of interest). This is an example of a mixed-effects model with a simple covariance structure—a balanced repeated-measures design. The dataset contains only 20 observations, so we would like to account for the small sample in our analysis. Because this is a balanced repeated-measures design, we can use the repeated method to obtain small-sample inference for fixed effects. We specify the `dfmethod(repeated)` option with `mixed`. We also request REML estimates by specifying the `reml` option to account for the small number of groups.

```
. use http://www.stata-press.com/data/r15/t43
(T4.3 -- Winer, Brown, Michels)
. mixed score i.drug || person:, reml dfmethod(repeated)
Performing EM optimization:
Performing gradient-based optimization:
Iteration 0:   log restricted-likelihood = -49.640099
Iteration 1:   log restricted-likelihood = -49.640099
Computing standard errors:
Computing degrees of freedom:
Mixed-effects REML regression                   Number of obs     =         20
Group variable: person                          Number of groups  =          5
                                                Obs per group:
                                                              min =          4
                                                              avg =        4.0
                                                              max =          4
DF method: Repeated                             DF:           min =       4.00
                                                              avg =      10.00
                                                              max =      12.00
                                                F(3,    12.00)    =      24.76
Log restricted-likelihood = -49.640099          Prob > F          =     0.0000
```

score	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
drug						
2	-.8	1.939072	-0.41	0.687	-5.024874	3.424874
3	-10.8	1.939072	-5.57	0.000	-15.02487	-6.575126
4	5.6	1.939072	2.89	0.014	1.375126	9.824874
_cons	26.4	3.149604	8.38	0.001	17.6553	35.1447

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
person: Identity				
var(_cons)	40.20004	30.10272	9.264606	174.4319
var(Residual)	9.399997	3.837532	4.22305	20.92325

```
LR test vs. linear model: chibar2(01) = 15.03         Prob >= chibar2 = 0.0001
```

In the table for fixed effects, t statistics are reported instead of the default z statistics. We can compare our small-sample inference with the corresponding large-sample inference for fixed effects. We do not need to rerun the estimation command, because we can obtain large-sample results upon replay by default.

```
. mixed
Mixed-effects REML regression                   Number of obs     =         20
Group variable: person                          Number of groups  =          5
                                                Obs per group:
                                                              min =          4
                                                              avg =        4.0
                                                              max =          4
                                                Wald chi2(3)      =      74.28
Log restricted-likelihood = -49.640099          Prob > chi2       =     0.0000
```

score	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
drug						
2	-.8	1.939072	-0.41	0.680	-4.600511	3.000511
3	-10.8	1.939072	-5.57	0.000	-14.60051	-6.999489
4	5.6	1.939072	2.89	0.004	1.799489	9.400511
_cons	26.4	3.149604	8.38	0.000	20.22689	32.57311

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
person: Identity				
var(_cons)	40.20004	30.10272	9.264606	174.4319
var(Residual)	9.399997	3.837532	4.22305	20.92325

```
LR test vs. linear model: chibar2(01) = 15.03          Prob >= chibar2 = 0.0001
```

Comparing the above large-sample inference for fixed effects of `drug` with the small-sample inference, we see that the p-value for the level 4 of `drug` changes from 0.014 to 0.004.

If we wanted to replay our small-sample estimation results, we would type

```
. mixed, small
```

(*output omitted*)

The specified DF method and summaries of the coefficient-specific DFs are reported in the output header. We can use the `dftable()` option to display a fixed-effects table containing coefficient-specific DFs. `dftable(pvalue)` reports the fixed-effects table containing DFs, t statistics, and p-values, and `dftable(ci)` reports the fixed-effects table containing DFs and confidence intervals.

```
. mixed, dftable(pvalue) noretable
Mixed-effects REML regression                   Number of obs     =         20
Group variable: person                          Number of groups  =          5
                                                Obs per group:
                                                              min =          4
                                                              avg =        4.0
                                                              max =          4
DF method: Repeated                             DF:           min =       4.00
                                                              avg =      10.00
                                                              max =      12.00
                                                F(3,     12.00)   =      24.76
Log restricted-likelihood = -49.640099          Prob > F         =     0.0000
```

score	Coef.	Std. Err.	DF	t	P>|t|
drug					
2	-.8	1.939072	12.0	-0.41	0.687
3	-10.8	1.939072	12.0	-5.57	0.000
4	5.6	1.939072	12.0	2.89	0.014
_cons	26.4	3.149604	4.0	8.38	0.001

Because levels of `drug` vary within `person`, the within-subject degrees of freedom, 12, are assigned to the coefficients for the levels of `drug`. The DF for the constant term is always the between-subject degrees of freedom, 4 in this example, because it is constant within random-effects levels.

The model F test is reported in the output header instead of the default χ^2 test. The F statistic for testing `drug` $= 0$ is 24.76 with DDF $= 12$, which agrees with the results of `anova, repeated()`:

```
. anova score person drug, repeated(drug)
                    Number of obs =         20    R-squared     =  0.9244
                    Root MSE      =    3.06594    Adj R-squared =  0.8803
```

Source	Partial SS	df	MS	F	Prob>F
Model	1379	7	197	20.96	0.0000
person	680.8	4	170.2	18.11	0.0001
drug	698.2	3	232.73333	24.76	0.0000
Residual	112.8	12	9.4		
Total	1491.8	19	78.515789		

```
Between-subjects error term:  person
                     Levels:  5          (4 df)
     Lowest b.s.e. variable:  person
Repeated variable: drug
                                  Huynh-Feldt epsilon        =  1.0789
                                  *Huynh-Feldt epsilon reset to 1.0000
                                  Greenhouse-Geisser epsilon =  0.6049
                                  Box's conservative epsilon =  0.3333
```

			Prob > F			
Source	df	F	Regular	H-F	G-G	Box
drug	3	24.76	0.0000	0.0000	0.0006	0.0076
Residual	12					

◁

▷ Example 14: Small-sample inference with an unbalanced repeated-measures design

Consider West, Welch, and Gałecki's (2015) dental veneer dataset from example 7, containing two measurements on each tooth from multiple teeth per patient. Because of small-sample size, we would like to obtain small-sample inference for fixed effects.

Some patients are missing observations for some teeth:

```
. use http://www.stata-press.com/data/r15/veneer, clear
(Dental veneer data)
. table patient tooth
```

Patient ID	Tooth number with patient 6	7	8	9	10	11
1	2	2	2	2	2	2
3	2	2	2	2	2	2
4	2	2	2	2	2	2
5		2	2	2	2	
6	2	2	2	2	2	2
7	2	2	2	2	2	2
8	2	2	2	2	2	2
9	2	2				
10	2	2	2	2	2	2
12		2	2	2	2	
13					2	
14			2		2	

The dataset is unbalanced; therefore, exact F tests for fixed effects are unavailable. As such, we will use the Satterthwaite and the Kenward–Roger approximation methods for calculating DF. Let's fit the model using the Kenward–Roger method first by specifying `dfmethod(kroger)`.

```
. mixed gcf followup base_gcf cda age || patient: followup, cov(un)
> || tooth:, reml nolog dfmethod(kroger)
Mixed-effects REML regression                   Number of obs     =        110
```

Group Variable	No. of Groups	Observations per Group Minimum	Average	Maximum
patient	12	2	9.2	12
tooth	55	2	2.0	2

```
DF method: Kenward-Roger                        DF:           min =      10.41
                                                              avg =      28.96
                                                              max =      50.71
                                                F(4,     27.96)    =       1.47
Log restricted-likelihood = -420.92761          Prob > F          =     0.2370
```

gcf	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
followup	.3009815	1.938641	0.16	0.879	-3.96767	4.569633
base_gcf	-.0183127	.1466261	-0.12	0.901	-.3132419	.2766164
cda	-.329303	.5533506	-0.60	0.554	-1.440355	.7817493
age	-.5773932	.2350491	-2.46	0.033	-1.098324	-.056462
_cons	45.73862	13.21824	3.46	0.002	18.53866	72.93858

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
patient: Unstructured				
var(followup)	41.88772	18.79997	17.38009	100.9535
var(_cons)	524.9851	253.0205	204.1287	1350.175
cov(followup,_cons)	-140.4229	66.57623	-270.9099	-9.935907
tooth: Identity				
var(_cons)	47.45738	16.63034	23.8792	94.3165
var(Residual)	48.86704	10.50523	32.06479	74.47382

LR test vs. linear model: chi2(4) = 91.12 Prob > chi2 = 0.0000

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

Compared with the p-values of the large-sample results from example 7, the p-values for `age` and `_cons` change substantially from 0.007 and 0.000 to 0.033 and 0.002, respectively. Note that for the Kenward–Roger method, not only the p-values and confidence intervals differ from those of the large-sample results but also the standard errors for the fixed effects differ. The standard errors differ because this method uses a bias-adjusted estimator of the variance–covariance matrix of fixed effects.

Now, let's fit the model using the Satterthwaite approximation:

```
. mixed gcf followup base_gcf cda age || patient: followup, cov(un)
> || tooth:, reml nolog dfmethod(satterthwaite)
```

Mixed-effects REML regression Number of obs = 110

Group Variable	No. of Groups	Observations per Group Minimum	Average	Maximum
patient	12	2	9.2	12
tooth	55	2	2.0	2

DF method: Satterthwaite DF: min = 10.41
avg = 28.96
max = 50.71

F(4, 16.49) = 1.87

Log restricted-likelihood = -420.92761 Prob > F = 0.1638

gcf	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
followup	.3009815	1.936863	0.16	0.879	-3.963754	4.565717
base_gcf	-.0183127	.1433094	-0.13	0.899	-.3065704	.269945
cda	-.329303	.5292525	-0.62	0.537	-1.39197	.7333636
age	-.5773932	.2139656	-2.70	0.022	-1.051598	-.1031885
_cons	45.73862	12.55497	3.64	0.001	19.90352	71.57372

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
patient: Unstructured				
var(followup)	41.88772	18.79997	17.38009	100.9535
var(_cons)	524.9851	253.0205	204.1287	1350.175
cov(followup,_cons)	-140.4229	66.57623	-270.9099	-9.935907
tooth: Identity				
var(_cons)	47.45738	16.63034	23.8792	94.3165
var(Residual)	48.86704	10.50523	32.06479	74.47382

LR test vs. linear model: chi2(4) = 91.12 Prob > chi2 = 0.0000

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

Using the Satterthwaite method, we see that the p-value for `age` is 0.022 and for `_cons` is 0.001 and that these are again substantially different from their large-sample counterparts. On the other hand, unlike the standard errors for the Kenward–Roger method, those for the Satterthwaite method are the same as the standard errors from the large-sample results.

Looking at the DF summaries in the output header of the two methods, we notice that they are exactly the same. This is because DFs for fixed effects obtained using the Kenward–Roger and Satterthwaite methods are the same for single-hypothesis tests. (You can verify this by specifying, for example, `dftable(pvalue)` with the above commands or by using `estat df`; see [ME] **estat df**.) The DDFs differ, however, for multiple-hypotheses tests. For example, DDF computed for the overall model test using `dfmethod(satterthwaite)` (16.49) is smaller than that computed using `dfmethod(kroger)` (27.96).

There are no general guidelines to which method should be preferred, but according to Schaalje, McBride, and Fellingham (2002), the Kenward–Roger method outperforms the Satterthwaite method when the variance–covariance structure of the random effects is unstructured, which is the case in our example.

◁

Determining which DDF method is best is a difficult task and may often need simulation. The choice of the method depends on the specified covariance structure, sample size, and imbalance of the data. No method applies to all situations; thus you should use caution when choosing among methods.

Stored results

mixed stores the following in e():

Scalars

e(N)	number of observations
e(k)	number of parameters
e(k_f)	number of fixed-effects parameters
e(k_r)	number of random-effects parameters
e(k_rs)	number of variances
e(k_rc)	number of covariances
e(k_res)	number of residual-error parameters
e(N_clust)	number of clusters
e(nrgroups)	number of residual-error by() groups
e(ar_p)	AR order of residual errors, if specified
e(ma_q)	MA order of residual errors, if specified
e(res_order)	order of residual-error structure, if appropriate
e(df_m)	model degrees of freedom
e(small)	1 if dfmethod() option specified, 0 otherwise
e(F)	overall F test statistic when dfmethod() is specified
e(ddf_m)	model DDF
e(df_max)	maximum DF
e(df_avg)	average DF
e(df_min)	minimum DF
e(ll)	log (restricted) likelihood
e(chi2)	χ^2
e(p)	p-value for model test
e(ll_c)	log likelihood, comparison model
e(chi2_c)	χ^2, comparison test
e(df_c)	degrees of freedom, comparison test
e(p_c)	p-value for comparison test
e(rank)	rank of e(V)
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros

e(cmd)	mixed
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(wtype)	weight type (first-level weights)
e(wexp)	weight expression (first-level weights)
e(fweight$k$)	fweight variable for kth highest level, if specified
e(pweight$k$)	pweight variable for kth highest level, if specified
e(ivars)	grouping variables
e(title)	title in estimation output
e(redim)	random-effects dimensions
e(vartypes)	variance-structure types
e(revars)	random-effects covariates
e(resopt)	residuals() specification, as typed
e(rstructure)	residual-error structure
e(rstructlab)	residual-error structure output label
e(rbyvar)	residual-error by() variable, if specified
e(rglabels)	residual-error by() groups labels
e(pwscale)	sampling-weight scaling method
e(timevar)	residual-error t() variable, if specified
e(dfmethod)	DF method specified in dfmethod()
e(dftitle)	title for DF method
e(chi2type)	Wald; type of model χ^2 test
e(clustvar)	name of cluster variable
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. Err.
e(method)	ML or REML
e(opt)	type of optimization

e(optmetric)	matsqrt or matlog; random-effects matrix parameterization
e(emonly)	emonly, if specified
e(ml_method)	type of ml method
e(technique)	maximization technique
e(datasignature)	the checksum
e(datasignaturevars)	variables used in calculation of checksum
e(properties)	b V
e(estat_cmd)	program used to implement estat
e(predict)	program used to implement predict
e(marginswtype)	weight type for margins
e(marginswexp)	weight expression for margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved
Matrices	
e(b)	coefficient vector
e(N_g)	group counts
e(g_min)	group-size minimums
e(g_avg)	group-size averages
e(g_max)	group-size maximums
e(tmap)	ID mapping for unstructured residual errors
e(V)	variance–covariance matrix of the estimators
e(V_modelbased)	model-based variance
e(df)	parameter-specific DF for fixed effects
e(V_df)	variance–covariance matrix of the estimators when dfmethod(kroger) is specified
Functions	
e(sample)	marks estimation sample

Methods and formulas

Methods and formulas are presented under the following headings:

Estimation using ML and REML

As given by (1), in the absence of weights we have the linear mixed model

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

where $\mathbf{y}$ is the $n \times 1$ vector of responses, $\mathbf{X}$ is an $n \times p$ design/covariate matrix for the fixed effects $\boldsymbol{\beta}$, and $\mathbf{Z}$ is the $n \times q$ design/covariate matrix for the random effects $\mathbf{u}$. The $n \times 1$ vector of errors $\boldsymbol{\epsilon}$ is for now assumed to be multivariate normal with mean 0 and variance matrix $\sigma^2_\epsilon \mathbf{I}_n$. We also assume that $\mathbf{u}$ has variance–covariance matrix $\mathbf{G}$ and that $\mathbf{u}$ is orthogonal to $\boldsymbol{\epsilon}$ so that

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

Considering the combined error term $\mathbf{Z}\mathbf{u} + \boldsymbol{\epsilon}$, we see that $\mathbf{y}$ is multivariate normal with mean $\mathbf{X}\boldsymbol{\beta}$ and $n \times n$ variance–covariance matrix

$$\mathbf{V} = \mathbf{Z}\mathbf{G}\mathbf{Z}' + \sigma^2_\epsilon \mathbf{I}_n$$

Defining $\boldsymbol{\theta}$ as the vector of unique elements of $\mathbf{G}$ results in the log likelihood

$$L(\boldsymbol{\beta},\boldsymbol{\theta},\sigma^2_\epsilon) = -\frac{1}{2}\left\{n\log(2\pi) + \log|\mathbf{V}| + (\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'\mathbf{V}^{-1}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\right\} \tag{9}$$

which is maximized as a function of $\boldsymbol{\beta}$, $\boldsymbol{\theta}$, and σ^2_ϵ. As explained in chapter 6 of Searle, Casella, and McCulloch (1992), considering instead the likelihood of a set of linear contrasts $\mathbf{Ky}$ that do not depend on $\boldsymbol{\beta}$ results in the restricted log likelihood

$$L_R(\boldsymbol{\beta},\boldsymbol{\theta},\sigma^2_\epsilon) = L(\boldsymbol{\beta},\boldsymbol{\theta},\sigma^2_\epsilon) - \frac{1}{2}\log\left|\mathbf{X}'\mathbf{V}^{-1}\mathbf{X}\right| \tag{10}$$

Given the high dimension of $\mathbf{V}$, however, the log-likelihood and restricted log-likelihood criteria are not usually computed by brute-force application of the above expressions. Instead, you can simplify the problem by subdividing the data into independent clusters (and subclusters if possible) and using matrix decomposition methods on the smaller matrices that result from treating each cluster one at a time.

Consider the two-level model described previously in (2),

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

for $j = 1,\ldots,M$ clusters with cluster j containing n_j observations, with $\mathrm{Var}(\mathbf{u}_j) = \boldsymbol{\Sigma}$, a $q \times q$ matrix.

Efficient methods for computing (9) and (10) are given in chapter 2 of Pinheiro and Bates (2000). Namely, for the two-level model, define $\boldsymbol{\Delta}$ to be the Cholesky factor of $\sigma^2_\epsilon\boldsymbol{\Sigma}^{-1}$, such that $\sigma^2_\epsilon\boldsymbol{\Sigma}^{-1} = \boldsymbol{\Delta}'\boldsymbol{\Delta}$. For $j = 1,\ldots,M$, decompose

$$\begin{bmatrix}\mathbf{Z}_j\\ \boldsymbol{\Delta}\end{bmatrix} = \mathbf{Q}_j\begin{bmatrix}\mathbf{R}_{11j}\\ \mathbf{0}\end{bmatrix}$$

by using an orthogonal-triangular (QR) decomposition, with $\mathbf{Q}_j$ a (n_j+q)-square matrix and $\mathbf{R}_{11j}$ a q-square matrix. We then apply $\mathbf{Q}_j$ as follows:

$$\begin{bmatrix}\mathbf{R}_{10j}\\ \mathbf{R}_{00j}\end{bmatrix} = \mathbf{Q}_j'\begin{bmatrix}\mathbf{X}_j\\ \mathbf{0}\end{bmatrix}; \qquad \begin{bmatrix}\mathbf{c}_{1j}\\ \mathbf{c}_{0j}\end{bmatrix} = \mathbf{Q}_j'\begin{bmatrix}\mathbf{y}_j\\ \mathbf{0}\end{bmatrix}$$

Stack the $\mathbf{R}_{00j}$ and $\mathbf{c}_{0j}$ matrices, and perform the additional QR decomposition

$$\begin{bmatrix}\mathbf{R}_{001} & \mathbf{c}_{01}\\ \vdots & \vdots\\ \mathbf{R}_{00M} & \mathbf{c}_{0M}\end{bmatrix} = \mathbf{Q}_0\begin{bmatrix}\mathbf{R}_{00} & \mathbf{c}_0\\ \mathbf{0} & \mathbf{c}_1\end{bmatrix}$$

Pinheiro and Bates (2000) show that ML estimates of $\boldsymbol{\beta}$, σ^2_ϵ, and $\boldsymbol{\Delta}$ (the unique elements of $\boldsymbol{\Delta}$, that is) are obtained by maximizing the profile log likelihood (profiled in $\boldsymbol{\Delta}$)

$$L(\boldsymbol{\Delta}) = \frac{n}{2}\{\log n - \log(2\pi) - 1\} - n\log||\mathbf{c}_1|| + \sum_{j=1}^{M}\log\left|\frac{\det(\boldsymbol{\Delta})}{\det(\mathbf{R}_{11j})}\right| \tag{11}$$

where $||\cdot||$ denotes the 2-norm. Following this maximization with

$$\widehat{\boldsymbol{\beta}} = \mathbf{R}_{00}^{-1}\mathbf{c}_0; \quad \widehat{\sigma}^2_\epsilon = n^{-1}||\mathbf{c}_1||^2 \tag{12}$$

REML estimates are obtained by maximizing

$$L_R(\boldsymbol{\Delta}) = \frac{n-p}{2}\left\{\log(n-p) - \log(2\pi) - 1\right\} - (n-p)\log||\mathbf{c}_1|| - \log|\det(\mathbf{R}_{00})| + \sum_{j=1}^{M}\log\left|\frac{\det(\boldsymbol{\Delta})}{\det(\mathbf{R}_{11j})}\right| \tag{13}$$

followed by

$$\widehat{\boldsymbol{\beta}} = \mathbf{R}_{00}^{-1}\mathbf{c}_0; \quad \widehat{\sigma}_\epsilon^2 = (n-p)^{-1}||\mathbf{c}_1||^2$$

For numerical stability, maximization of (11) and (13) is not performed with respect to the unique elements of $\boldsymbol{\Delta}$ but instead with respect to the unique elements of the matrix square root (or matrix logarithm if the `matlog` option is specified) of $\boldsymbol{\Sigma}/\sigma_\epsilon^2$; define $\boldsymbol{\gamma}$ to be the vector containing these elements.

Once maximization with respect to $\boldsymbol{\gamma}$ is completed, $(\boldsymbol{\gamma}, \sigma_\epsilon^2)$ is reparameterized to $\{\boldsymbol{\alpha}, \log(\sigma_\epsilon)\}$, where $\boldsymbol{\alpha}$ is a vector containing the unique elements of $\boldsymbol{\Sigma}$, expressed as logarithms of standard deviations for the diagonal elements and hyperbolic arctangents of the correlations for off-diagonal elements. This last step is necessary 1) to obtain a joint variance–covariance estimate of the elements of $\boldsymbol{\Sigma}$ and σ_ϵ^2; 2) to obtain a parameterization under which parameter estimates can be interpreted individually, rather than as elements of a matrix square root (or logarithm); and 3) to parameterize these elements such that their ranges each encompass the entire real line.

Obtaining a joint variance–covariance matrix for the estimated $\{\boldsymbol{\alpha}, \log(\sigma_\epsilon)\}$ requires the evaluation of the log likelihood (or log-restricted likelihood) with only $\boldsymbol{\beta}$ profiled out. For ML, we have

$$\begin{aligned} L^*\{\boldsymbol{\alpha}, \log(\sigma_\epsilon)\} &= L\{\boldsymbol{\Delta}(\boldsymbol{\alpha}, \sigma_\epsilon^2), \sigma_\epsilon^2\} \\ &= -\frac{n}{2}\log(2\pi\sigma_\epsilon^2) - \frac{||\mathbf{c}_1||^2}{2\sigma_\epsilon^2} + \sum_{j=1}^{M}\log\left|\frac{\det(\boldsymbol{\Delta})}{\det(\mathbf{R}_{11j})}\right| \end{aligned}$$

with the analogous expression for REML.

The variance–covariance matrix of $\widehat{\boldsymbol{\beta}}$ is estimated as

$$\widehat{\mathrm{Var}}(\widehat{\boldsymbol{\beta}}) = \widehat{\sigma}_\epsilon^2\mathbf{R}_{00}^{-1}\left(\mathbf{R}_{00}^{-1}\right)'$$

but this does not mean that $\widehat{\mathrm{Var}}(\widehat{\boldsymbol{\beta}})$ is identical under both ML and REML because $\mathbf{R}_{00}$ depends on $\boldsymbol{\Delta}$. Because $\widehat{\boldsymbol{\beta}}$ is asymptotically uncorrelated with $\{\widehat{\boldsymbol{\alpha}}, \log(\widehat{\sigma}_\epsilon)\}$, the covariance of $\widehat{\boldsymbol{\beta}}$ with the other estimated parameters is treated as 0.

Parameter estimates are stored in `e(b)` as $\{\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\alpha}}, \log(\widehat{\sigma}_\epsilon)\}$, with the corresponding (block-diagonal) variance–covariance matrix stored in `e(V)`. Parameter estimates can be displayed in this metric by specifying the `estmetric` option. However, in `mixed` output, variance components are most often displayed either as variances and covariances or as standard deviations and correlations.

EM iterations are derived by considering the $\mathbf{u}_j$ in (2) as missing data. Here we describe the procedure for maximizing the log likelihood via EM; the procedure for maximizing the restricted log likelihood is similar. The log likelihood for the full data $(\mathbf{y}, \mathbf{u})$ is

$$L_F(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \sigma_\epsilon^2) = \sum_{j=1}^{M}\left\{\log f_1(\mathbf{y}_j|\mathbf{u}_j, \boldsymbol{\beta}, \sigma_\epsilon^2) + \log f_2(\mathbf{u}_j|\boldsymbol{\Sigma})\right\}$$

where $f_1(\cdot)$ is the density function for multivariate normal with mean $\mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j$ and variance $\sigma^2_\epsilon\mathbf{I}_{n_j}$, and $f_2(\cdot)$ is the density for multivariate normal with mean $\mathbf{0}$ and $q \times q$ covariance matrix $\boldsymbol{\Sigma}$. As before, we can profile $\boldsymbol{\beta}$ and σ^2_ϵ out of the optimization, yielding the following EM iterative procedure:

1. For the current iterated value of $\boldsymbol{\Sigma}^{(t)}$, fix $\widehat{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}(\boldsymbol{\Sigma}^{(t)})$ and $\widehat{\sigma}^2_\epsilon = \widehat{\sigma}^2_\epsilon(\boldsymbol{\Sigma}^{(t)})$ according to (12).
2. Expectation step: Calculate

$$\begin{aligned} D(\boldsymbol{\Sigma}) &\equiv E\left\{L_F(\widehat{\boldsymbol{\beta}}, \boldsymbol{\Sigma}, \widehat{\sigma}^2_\epsilon)|\mathbf{y}\right\} \\ &= C - \frac{M}{2}\log\det(\boldsymbol{\Sigma}) - \frac{1}{2}\sum_{j=1}^{M} E\left(\mathbf{u}_j'\boldsymbol{\Sigma}^{-1}\mathbf{u}_j|\mathbf{y}\right) \end{aligned}$$

   where C is a constant that does not depend on $\boldsymbol{\Sigma}$, and the expected value of the quadratic form $\mathbf{u}_j'\boldsymbol{\Sigma}^{-1}\mathbf{u}_j$ is taken with respect to the conditional density $f(\mathbf{u}_j|\mathbf{y}, \widehat{\boldsymbol{\beta}}, \boldsymbol{\Sigma}^{(t)}, \widehat{\sigma}^2_\epsilon)$.
3. Maximization step: Maximize $D(\boldsymbol{\Sigma})$ to produce $\boldsymbol{\Sigma}^{(t+1)}$.

For general, symmetric $\boldsymbol{\Sigma}$, the maximizer of $D(\boldsymbol{\Sigma})$ can be derived explicitly, making EM iterations quite fast.

For general, residual-error structures,

$$\mathrm{Var}(\boldsymbol{\epsilon}_j) = \sigma^2_\epsilon\boldsymbol{\Lambda}_j$$

where the subscript j merely represents that $\boldsymbol{\epsilon}_j$ and $\boldsymbol{\Lambda}_j$ vary in dimension in unbalanced data, the data are first transformed according to

$$\mathbf{y}^*_j = \widehat{\boldsymbol{\Lambda}}_j^{-1/2}\mathbf{y}_j; \qquad \mathbf{X}^*_j = \widehat{\boldsymbol{\Lambda}}_j^{-1/2}\mathbf{X}_j; \qquad \mathbf{Z}^*_j = \widehat{\boldsymbol{\Lambda}}_j^{-1/2}\mathbf{Z}_j;$$

and the likelihood-evaluation techniques described above are applied to $\mathbf{y}^*_j$, $\mathbf{X}^*_j$, and $\mathbf{Z}^*_j$ instead. The unique elements of $\boldsymbol{\Lambda}$, $\boldsymbol{\rho}$, are estimated along with the fixed effects and variance components. Because σ^2_ϵ is always estimated and multiplies the entire $\boldsymbol{\Lambda}_j$ matrix, $\widehat{\boldsymbol{\rho}}$ is parameterized to take this into account.

In the presence of sampling weights, following Rabe-Hesketh and Skrondal (2006), the weighted log pseudolikelihood for a two-level model is given as

$$L(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \sigma^2_\epsilon) = \sum_{j=1}^{M} w_j \log\left[\int \exp\left\{\sum_{i=1}^{n_j} w_{i|j}\log f_1(y_{ij}|\mathbf{u}_j, \boldsymbol{\beta}, \sigma^2_\epsilon)\right\} f_2(\mathbf{u}_j|\boldsymbol{\Sigma})d\mathbf{u}_j\right] \tag{14}$$

where w_j is the inverse of the probability of selection for the jth cluster, $w_{i|j}$ is the inverse of the conditional probability of selection of individual i given the selection of cluster j, and $f_1(\cdot)$ and $f_2(\cdot)$ are the multivariate normal densities previously defined.

Weighted estimation is achieved through incorporating w_j and $w_{i|j}$ into the matrix decomposition methods detailed above to reflect replicated clusters for w_j and replicated observations within clusters for $w_{i|j}$. Because this estimation is based on replicated clusters and observations, frequency weights are handled similarly.

Rescaling of sampling weights can take one of three available forms:

Under `pwscale(size)`,

$$w^*_{i|j} = n_j w_{i|j} \left\{ \sum_{i=1}^{n_j} w_{i|j} \right\}^{-1}$$

Under `pwscale(effective)`,

$$w^*_{i|j} = w_{i|j} \left\{ \sum_{i=1}^{n_j} w_{i|j} \right\} \left\{ \sum_{i=1}^{n_j} w^2_{i|j} \right\}^{-1}$$

Under both the above, w_j remains unchanged. For method `pwscale(gk)`, however, both weights are modified:

$$w^*_j = n_j^{-1} \sum_{i=1}^{n_j} w_{i|j} w_j \qquad w^*_{i|j} = 1$$

Under ML estimation, robust standard errors are obtained in the usual way (see [P] **_robust**) with the one distinction being that in multilevel models, robust variances are, at a minimum, clustered at the highest level. This is because given the form of the log likelihood, scores aggregate at the top-level clusters. For a two-level model, scores are obtained as the partial derivatives of $L_j(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \sigma^2_\epsilon)$ with respect to $\{\boldsymbol{\beta}, \boldsymbol{\alpha}, \log(\sigma_\epsilon)\}$, where L_j is the log likelihood for cluster j and $L = \sum_{j=1}^{M} L_j$. Robust variances are not supported under REML estimation because the form of the log restricted likelihood does not lend itself to separation by highest-level clusters.

EM iterations always assume equal weighting and an independent, homoskedastic error structure. As such, with weighted data or when error structures are more complex, EM is used only to obtain starting values.

For extensions to models with three or more levels, see Bates and Pinheiro (1998) and Rabe-Hesketh and Skrondal (2006).

Denominator degrees of freedom

When the `dfmethod()` option is specified, `mixed` uses a t distribution with ν_{ddf} degrees of freedom to perform single-hypothesis tests for fixed effects H_0: $\beta_i = 0$ for $i = 1, 2, \ldots, p$ or an F distribution with model numerator degrees of freedom and ν_{ddf_m} DDF for a model (joint) test of all coefficients (except the constant) being equal to zero. Denominator degrees of freedom ν_{ddf} and ν_{ddf_m} are computed according to the specified DDF method.

Residual DDF

This method uses the residual degrees of freedom as the DDF, $\nu_{\text{ddf}} = n - p$, where n is the total number of observations, and p is the rank of the design matrix $\mathbf{X}$.

Repeated DDF

This method partitions the residual degrees of freedom into the between-subject degrees of freedom and the within-subject degrees of freedom. This partitioning of the degrees of freedom arises from balanced repeated-measures ANOVA analysis. If levels of a fixed effect change within a subject, then the within-subject degrees of freedom is assigned to the fixed effect of interest; otherwise, the between-subject degrees of freedom is assigned to that fixed effect. See Schluchter and Elashoff (1990) for more computational details and, specifically, for the expressions of between-subject and within-subject degrees of freedom.

ANOVA DDF

This method determines the DDF for a fixed effect depending on whether the corresponding covariate is contained in any of the random-effects equations. If the covariate is contained in a random-effects equation, the DDF ν_{ddf} for the fixed effect is computed as the number of levels of the level variable from that equation minus one. If the covariate is specified in more than one random-effects equation, the DDF ν_{ddf} for the fixed effect is computed as the smallest number of levels of the level variables from those equations minus one and is a conservative estimate of the true DDF. If the covariate is specified only in the fixed-effects equation, the DDF is computed as $\nu_{\text{ddf}} = n - \text{rank}(\mathbf{X}, \mathbf{Z})$.

For example, suppose we have the following mixed model,

```
mixed y A B C || D: A || E: A B
```

where `A`, `B`, and `C` are fixed effects, and `D` and `E` are nested random effects. For the fixed effect `A`, ν_{ddf} is the smaller number of levels of variables `D` and `E` minus one because `A` is included in random-effects equations at both levels `D` and `E`. For the fixed effect `B`, ν_{ddf} is the number of levels of level variable `E` minus one because `B` is included in the random-effects equation at the level `E`. For the fixed effect `C`, $\nu_{\text{ddf}} = n - \text{rank}(\mathbf{X}, \mathbf{Z})$ because `C` is not included in any of the random-effects equations.

For the three methods above, the DDF for a model test of H_0: $\boldsymbol{\beta} = \mathbf{0}$ is computed as follows. If all corresponding single-hypothesis tests H_0: $\beta_i = 0$ have the same DDF ν_{ddf}, then model DDF $\nu_{\text{ddf}_m} = \nu_{\text{ddf}}$. If the single-hypothesis DDF differs, then ν_{ddf_m} is not defined, and the large-sample χ^2 test is reported instead of the F test.

To provide formulas for the Satterthwaite and Kenward–Roger methods, consider a general linear-hypotheses test of fixed effects H_0: $\mathbf{C}'\boldsymbol{\beta} = \mathbf{b}$ with a $p \times l$ matrix of linear hypotheses $\mathbf{C}$ of rank l.

The variance–covariance matrix of $\mathbf{y}$ is $\text{Var}(\mathbf{y}) = \mathbf{V} = \mathbf{ZGZ}' + \mathbf{R} = \mathbf{V}(\boldsymbol{\sigma})$ and can be viewed as a function of variance components $\boldsymbol{\sigma}$ $(r \times 1)$. Suppose that the first two partial derivatives of $\mathbf{V}(\boldsymbol{\sigma})$ with respect to $\boldsymbol{\sigma}$ exist.

Let $\widehat{\boldsymbol{\sigma}}$ be the REML estimator of $\boldsymbol{\sigma}$. Then, the REML estimator of the fixed effects $\boldsymbol{\beta}$ is the generalized least-squares estimator

$$\widehat{\boldsymbol{\beta}} = \left\{\mathbf{X}'\mathbf{V}^{-1}(\widehat{\boldsymbol{\sigma}})\mathbf{X}\right\}^{-1}\mathbf{X}'\mathbf{V}^{-1}(\widehat{\boldsymbol{\sigma}})\mathbf{Y}$$

where $\widehat{\text{Var}}(\widehat{\boldsymbol{\beta}}) = \widehat{\boldsymbol{\Phi}} = \boldsymbol{\Phi}(\widehat{\boldsymbol{\sigma}}) = \left\{\mathbf{X}'\mathbf{V}^{-1}(\widehat{\boldsymbol{\sigma}})\mathbf{X}\right\}^{-1}$ is the conventional estimator of the variance–covariance matrix of the fixed effects $\widehat{\boldsymbol{\beta}}$, and $\mathbf{V}(\widehat{\boldsymbol{\sigma}})$ is the estimator of the covariance matrix of $\mathbf{y}$.

Under the null H_0: $\mathbf{C}'\boldsymbol{\beta} = \mathbf{b}$, the F test statistic is

$$F = \frac{1}{l}(\mathbf{C}'\widehat{\boldsymbol{\beta}} - \mathbf{b})'(\mathbf{C}'\widehat{\boldsymbol{\Phi}}\mathbf{C})^{-1}(\mathbf{C}'\widehat{\boldsymbol{\beta}} - \mathbf{b})$$

and it has an F distribution with l numerator and ν_{ddf_C} DDF.

Satterthwaite DDF

This method is derived from the DDF formula of the original approximation attributable to Satterthwaite (1946):

$$\text{ddf} = \frac{2(\mathbf{C}'\widehat{\boldsymbol{\Phi}}\mathbf{C})^2}{\text{Var}(\mathbf{C}'\widehat{\boldsymbol{\Phi}}\mathbf{C})}$$

For a single-hypothesis test of H_0: $\mathbf{c}'\boldsymbol{\beta} = \mathbf{b}$, where $\mathbf{c}$ and $\mathbf{b}$ are vectors of known constants, Giesbrecht and Burns (1985) proposed using

$$\nu_{\text{ddf}} = \frac{2(\mathbf{c}'\widehat{\boldsymbol{\Phi}}\mathbf{c})^2}{\text{Var}(\mathbf{c}'\widehat{\boldsymbol{\Phi}}\mathbf{c})} = \frac{2(\mathbf{c}'\widehat{\boldsymbol{\Phi}}\mathbf{c})^2}{\mathbf{d}'\mathbf{W}\mathbf{d}} \tag{15}$$

where $\mathbf{d}$ is a vector of partial derivatives of $\mathbf{c}'\boldsymbol{\Phi}(\boldsymbol{\sigma})\mathbf{c}$ with respect to $\boldsymbol{\sigma}$ evaluated at $\widehat{\boldsymbol{\sigma}}$, and $\widehat{\text{Var}}(\widehat{\boldsymbol{\sigma}}) = \mathbf{W}$ is the estimator of the variance–covariance matrix of $\widehat{\boldsymbol{\sigma}}$ computed based on the expected information matrix $\mathbf{I_E}$ in (17) or on the observed information matrix if suboption `oim` of `dfmethod()` is specified.

For a multiple-hypotheses test (when the rank of $\mathbf{C}$ is greater than 1), Fai and Cornelius (1996) proposed an extension of the Giesbrecht–Burns single-degree-of-freedom method. Their method involves the spectral decomposition $\mathbf{C}'\widehat{\boldsymbol{\Phi}}\mathbf{C} = \mathbf{P}'\mathbf{D}\mathbf{P}$, where $\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_l)$ is an orthogonal matrix of eigenvectors, and $\mathbf{D} = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_l)$ is a diagonal matrix of the corresponding eigenvalues. Using this decomposition, we can write the F-test statistic as a sum of l independent approximate t random variates, $F = Q/l$ with

$$Q = \sum_{k=1}^{l} \frac{\{\mathbf{p}_k'(\mathbf{C}'\widehat{\boldsymbol{\beta}} - \mathbf{b})\}^2}{\lambda_k} = \sum_{k=1}^{l} t_{v_k}^2$$

where v_k is computed using (15). Because t_{v_k}s are independent and have approximate t distributions with v_k degrees of freedom,

$$E(Q) = \sum_{k=1}^{l} \frac{v_k}{v_k - 2} I(v_k > 2)$$

Then, the DDF for a multiple-hypotheses test can be approximately written as

$$\nu_{\text{ddf}_C} = \frac{2E(Q)}{E(Q) - l}$$

For more computational details of the Satterthwaite method, see Fai and Cornelius (1996).

Kenward–Roger DDF

This method was developed by Kenward and Roger (1997). It is based on adjusting the conventional variance–covariance estimator of fixed effects $\widehat{\boldsymbol{\Phi}}$ for small-sample bias and introducing a scaled F test that improves the small-sample performance of the conventional F test of fixed effects.

Kenward and Roger (1997) propose the adjusted estimator,

$$\widehat{\boldsymbol{\Phi}}_A = \widehat{\boldsymbol{\Phi}} + 2\widehat{\boldsymbol{\Phi}}\left\{\sum_{i=1}^{r}\sum_{j=1}^{r} W_{ij}(\mathbf{Q}_{ij} - \mathbf{P}_i\widehat{\boldsymbol{\Phi}}\mathbf{P}_j - \frac{1}{4}\mathbf{R}_{ij})\right\}\widehat{\boldsymbol{\Phi}} \tag{16}$$

where $\mathbf{P}_i = \mathbf{X}'\{\partial\mathbf{V}^{-1}(\boldsymbol{\sigma})/\partial\sigma_i\}\mathbf{X}$, $\mathbf{Q}_{ij} = \mathbf{X}'\{\partial\mathbf{V}^{-1}(\boldsymbol{\sigma})/\partial\sigma_i\}\mathbf{V}(\boldsymbol{\sigma})\{\partial\mathbf{V}^{-1}(\boldsymbol{\sigma})/\partial\sigma_j\}\mathbf{X}$, and $\mathbf{R}_{ij} = \mathbf{X}'\mathbf{V}^{-1}(\boldsymbol{\sigma})\{\partial^2\mathbf{V}(\boldsymbol{\sigma})/\partial\sigma_i\partial\sigma_j\}\mathbf{V}^{-1}(\boldsymbol{\sigma})\mathbf{X}$ evaluated at $\widehat{\boldsymbol{\sigma}}$ and W_{ij} is the (i,j)th element of $\mathbf{W}$, the estimator of the variance–covariance matrix of $\widehat{\boldsymbol{\sigma}}$ computed from the inverse of the expected information matrix $\mathbf{I}_E$, where the element I_E^{ij} of $\mathbf{I}_E$ is defined as

$$2I_E^{ij} = \mathrm{tr}\left(\frac{\partial\mathbf{V}^{-1}}{\partial\sigma_i}\mathbf{V}\frac{\partial\mathbf{V}^{-1}}{\partial\sigma_j}\mathbf{V}\right) - \mathrm{tr}(2\boldsymbol{\Phi}\mathbf{Q}_{ij} - \boldsymbol{\Phi}\mathbf{P}_i\boldsymbol{\Phi}\mathbf{P}_j) \tag{17}$$

Alternatively, you can use the observed information matrix as $\mathbf{W}$ by specifying suboption `oim` in `dfmethod()`.

All terms in (16), except those involving $\mathbf{R}_{ij}$, are invariant under reparameterization of the covariance structures. Also, the second derivative requires more computational resources and may not be numerically stable. For these reasons, the $\mathbf{R}_{ij}$ terms are ignored in the computation of $\widehat{\boldsymbol{\Phi}}_A$ in (16).

For multiple-hypotheses testing, Kenward and Roger (1997) propose the scaled F-test statistic, which under the null hypothesis can be written as

$$F_{\mathrm{KR}} = \frac{\lambda}{l}(\mathbf{C}'\widehat{\boldsymbol{\beta}} - \mathbf{b})'(\mathbf{C}'\widehat{\boldsymbol{\Phi}}_A\mathbf{C})^{-1}(\mathbf{C}'\widehat{\boldsymbol{\beta}} - \mathbf{b})$$

and has an F distribution with l numerator and ν_{ddf_C} DDF. The scale factor $\lambda = \nu_{\mathrm{ddf}_C}/(l-1+\nu_{\mathrm{ddf}_C})$.

The DDF ν_{ddf_C} and λ are approximated as

$$\nu_{\mathrm{ddf}_C} = 4 + \frac{l+2}{l\times\rho - 1} \qquad \text{and} \qquad \lambda = \frac{\nu_{\mathrm{ddf}_C}}{E^*(\nu_{\mathrm{ddf}_C} - 2)}$$

where $\rho = V^*/2(E^*)^2$ and E^* and V^* are the respective approximate mean and variance of the F_{KR} statistic; see Kenward and Roger (1997, 987) for expressions for E^* and V^*.

Acknowledgments

We thank Badi Baltagi of the Department of Economics at Syracuse University and Ray Carroll of the Department of Statistics at Texas A&M University for each providing us with a dataset used in this entry.

We also thank Mike Kenward of the Medical Statistics Unit at the London School of Hygiene and Tropical Medicine and James Roger (retired) of the Research Statistics Unit at GlaxoSmithKline for answering our questions about their methods.

Charles Roy Henderson (1911–1989) was born in Iowa and grew up on the family farm. His education in animal husbandry, animal nutrition, and statistics at Iowa State was interspersed with jobs in the Iowa Extension Service, Ohio University, and the U.S. Army. After completing his PhD, Henderson joined the Animal Science faculty at Cornell. He developed and applied statistical methods in the improvement of farm livestock productivity through genetic selection, with particular focus on dairy cattle. His methods are general and have been used worldwide in livestock breeding and beyond agriculture. Henderson's work on variance components and best linear unbiased predictions has proved to be one of the main roots of current mixed-model methods.

References

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

Baltagi, B. H., S. H. Song, and B. C. Jung. 2001. The unbalanced nested error component regression model. *Journal of Econometrics* 101: 357–381.

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

Brown, H., and R. Prescott. 2015. *Applied Mixed Models in Medicine*. 3rd ed. Chichester, UK: Wiley.

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

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

——. 2014. Using gsem to combine estimation results. *The Stata Blog: Not Elsewhere Classified*. http://blog.stata.com/2014/08/18/using-gsem-to-combine-estimation-results/.

Carle, A. C. 2009. Fitting multilevel models in complex survey data with design weights: Recommendations. *BMC Medical Research Methodology* 9: 49.

Chen, X., and L. Wei. 2003. A comparison of recent methods for the analysis of small-sample cross-over studies. *Statistics in Medicine* 22: 2821–2833.

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

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

Elston, D. A. 1998. Estimation of denominator degrees of freedom of F-distributions for assessing Wald statistics for fixed-effect factors in unbalanced mixed models. *Biometrics* 54: 1085–1096.

Fai, A. H.-T., and P. L. Cornelius. 1996. Approximate F-tests of multiple degree of freedom hypotheses in generalized least squares analyses of unbalanced split-plot experiments. *Journal of Statistical Computation and Simulation* 54: 363–378.

Fitzmaurice, G. M., N. M. Laird, and J. H. Ware. 2011. *Applied Longitudinal Analysis*. 2nd ed. Hoboken, NJ: Wiley.

Giesbrecht, F. G., and J. C. Burns. 1985. Two-stage analysis based on a mixed model: Large-sample asymptotic theory and small-sample simulation results. *Biometrics* 41: 477–486.

Goldstein, H. 1986. Efficient statistical modelling of longitudinal data. *Annals of Human Biology* 13: 129–141.

Graubard, B. I., and E. L. Korn. 1996. Modelling the sampling design in the analysis of health surveys. *Statistical Methods in Medical Research* 5: 263–281.

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

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

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

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

Huber, C. 2013a. Multilevel linear models in Stata, part 1: Components of variance. *The Stata Blog: Not Elsewhere Classified*. http://blog.stata.com/2013/02/04/multilevel-linear-models-in-stata-part-1-components-of-variance/.

——. 2013b. Multilevel linear models in Stata, part 2: Longitudinal data. *The Stata Blog: Not Elsewhere Classified*. http://blog.stata.com/2013/02/18/multilevel-linear-models-in-stata-part-2-longitudinal-data/.

——. 2014. How to simulate multilevel/longitudinal data. *The Stata Blog: Not Elsewhere Classified*. http://blog.stata.com/2014/07/18/how-to-simulate-multilevellongitudinal-data/.

Kackar, R. N., and D. A. Harville. 1984. Approximations for standard errors of estimators of fixed and random effects in mixed linear models. *Journal of the American Statistical Association* 79: 853–862.

Kenward, M. G., and J. H. Roger. 1997. Small sample inference for fixed effects from restricted maximum likelihood. *Biometrics* 53: 983–997.

——. 2009. An improved approximation to the precision of fixed effects from restricted maximum likelihood. *Computational Statistics & Data Analysis* 53: 2583–2595.

Khuri, A. I., T. Mathew, and B. K. Sinha. 1998. *Statistical Tests for Mixed Linear Models*. New York: Wiley.

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

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

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

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

Munnell, A. H. 1990. Why has productivity growth declined? Productivity and public investment. *New England Economic Review* Jan./Feb.: 3–22.

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

Palmer, T. M., C. M. Macdonald-Wallis, D. A. Lawlor, and K. Tilling. 2014. Estimating adjusted associations between random effects from multilevel models: The reffadjust package. *Stata Journal* 14: 119–140.

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

Pfeffermann, D., C. J. Skinner, D. J. Holmes, H. Goldstein, and J. Rasbash. 1998. Weighting for unequal selection probabilities in multilevel models. *Journal of the Royal Statistical Society, Series B* 60: 23–40.

Pierson, R. A., and O. J. Ginther. 1987. Follicular population dynamics during the estrous cycle of the mare. *Animal Reproduction Science* 14: 219–231.

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

Prosser, R., J. Rasbash, and H. Goldstein. 1991. *ML3 Software for 3-Level Analysis: User's Guide for V. 2*. London: Institute of Education, University of London.

Rabe-Hesketh, S., and A. Skrondal. 2006. Multilevel modelling of complex survey data. *Journal of the Royal Statistical Society, Series A* 169: 805–827.

——. 2012. *Multilevel and Longitudinal Modeling Using Stata*. 3rd ed. College Station, TX: Stata Press.

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

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

Robson, K., and D. Pevalin. 2016. *Multilevel Modeling in Plain Language*. London: Sage.

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

Satterthwaite, F. E. 1946. An approximate distribution of estimates of variance components. *Biometrics Bulletin* 2: 110–114.

Schaalje, G. B., J. B. McBride, and G. W. Fellingham. 2002. Adequacy of approximations to distributions of test statistics in complex mixed linear models. *Journal of Agricultural, Biological, and Environmental Statistics* 7: 512–524.

Schluchter, M. D., and J. D. Elashoff. 1990. Small-sample adjustments to tests with unbalanced repeated measures assuming several covariance structures. *Journal of Statistical Computation and Simulation* 37: 69–87.

Schunck, R. 2013. Within and between estimates in random-effects models: Advantages and drawbacks of correlated random effects and hybrid models. *Stata Journal* 13: 65–76.

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

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

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

Vallejo, G., P. Fernández, F. J. Herrero, and N. M. Conejo. 2004. Alternative procedures for testing fixed effects in repeated measures designs when assumptions are violated. *Psicothema* 16: 498–508.

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

West, B. T., K. B. Welch, and A. T. Gałecki. 2015. *Linear Mixed Models: A Practical Guide Using Statistical Software*. 2nd ed. Boca Raton, FL: Chapman & Hall/CRC.

Wiggins, V. L. 2011. Multilevel random effects in xtmixed and sem—the long and wide of it. *The Stata Blog: Not Elsewhere Classified*. http://blog.stata.com/2011/09/28/multilevel-random-effects-in-xtmixed-and-sem-the-long-and-wide-of-it/.

Winer, B. J., D. R. Brown, and K. M. Michels. 1991. *Statistical Principles in Experimental Design*. 3rd ed. New York: McGraw–Hill.

Also see

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

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

[ME] **menl** — Nonlinear mixed-effects regression

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

[MI] **estimation** — Estimation commands for use with mi estimate

[BAYES] **bayes: mixed** — Bayesian multilevel linear regression

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

[XT] **xtrc** — Random-coefficients model

[XT] **xtreg** — Fixed-, between-, and random-effects and population-averaged linear models

[U] 20 Estimation and postestimation commands