

# Title

**xtmixed** — Multilevel mixed-effects linear regression

## Syntax

```
xtmixed 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 `_all` representing one group comprising all observations.

<i>fe_options</i>	Description
-------------------	-------------

Model

<code>noconstant</code>	suppress constant term from the fixed-effects equation
-------------------------	--

<i>re_options</i>	Description
-------------------	-------------

Model

<code>covariance(vartype)</code>	variance–covariance structure of the random effects
<code>noconstant</code>	suppress constant term from the random-effects equation
<code>collinear</code>	keep collinear variables
<code>fweight(exp)</code>	frequency weights at higher levels
<code>pweight(exp)</code>	sampling weights at higher levels

<i>vartype</i>	Description
----------------	-------------

<code>independent</code>	one unique variance parameter per random effect, all covariances zero; the default unless a factor variable is specified
<code>exchangeable</code>	equal variances for random effects, and one common pairwise covariance
<code>identity</code>	equal variances for random effects, all covariances zero
<code>unstructured</code>	all variances and covariances distinctly estimated

<i>options</i>	Description
<b>Model</b>	
<u>m</u> le	fit model via maximum likelihood; the default
reml	fit model via restricted maximum likelihood
pwscale( <i>scale_method</i> )	control scaling of sampling weights in two-level models
<u>r</u> esiduals( <i>rspec</i> )	structure of residual errors
<b>SE/Robust</b>	
vce( <i>vcetype</i> )	<i>vcetype</i> may be oim, <u>r</u> obust, or <u>c</u> luster <i>clustvar</i>
<b>Reporting</b>	
<u>l</u> evel(#)	set confidence level; default is level(95)
<u>v</u> ariance	show random-effects parameter estimates as variances and covariances
<u>n</u> orettable	suppress random-effects table
<u>n</u> ofetable	suppress fixed-effects table
<u>e</u> stmetric	show parameter estimates in the estimation metric
<u>n</u> oheader	suppress output header
<u>n</u> ogroup	suppress table summarizing groups
<u>n</u> ostderr	do not estimate standard errors of random-effects parameters
<u>n</u> olrtest	do not perform LR test comparing to linear regression
<i>display_options</i>	control column formats, row spacing, line width, and display of omitted variables and base and empty cells
<b>EM options</b>	
<u>e</u> miterate(#)	number of EM iterations; default is 20
<u>e</u> mtolerance(#)	EM convergence tolerance; default is 1e-10
emonly	fit model exclusively using EM
emlog	show EM iteration log
<u>e</u> mdots	show EM iterations as dots
<b>Maximization</b>	
<i>maximize_options</i>	control the maximization process; seldom used
matsqrt	parameterize variance components using matrix square roots; the default
matlog	parameterize variance components using matrix logarithms
<u>c</u> oeflegend	display legend instead of statistics

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

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

*bootstrap*, *by*, *jackknife*, *mi estimate*, *rolling*, and *statsby* are allowed; see [U] **11.1.10 Prefix commands**.

*fweights* and *pweights* are allowed; see [U] **11.1.6 weight**.

*coeflegend* does not appear in the dialog box.

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

## Menu

Statistics > Longitudinal/panel data > Multilevel mixed-effects models > Mixed-effects linear regression

## Description

`xtmixed` fits linear mixed models. Mixed models are characterized as containing both *fixed effects* and *random effects*. The fixed effects are analogous to standard regression coefficients and are estimated directly. The random effects are not directly estimated but are summarized according to their estimated variances and covariances. Although random effects are not directly estimated, you can form best linear unbiased predictions (BLUPs) of them (and standard errors) by using `predict` after `xtmixed`; see [XT] **xtmixed postestimation**. Random effects may take the form of either random intercepts or random coefficients, and the grouping structure of the data may consist of multiple levels of nested groups. As such, mixed models are also known in the literature as *multilevel models* and *hierarchical linear models*. The overall error distribution of the linear mixed model is assumed to be Gaussian, and heteroskedasticity and correlations within lowest-level groups also may be modeled.

## Options

### Model

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

`covariance(vartype)`, where *vartype* is

`independent` | `exchangeable` | `identity` | `unstructured`

specifies the structure of the covariance matrix for the random effects and may be specified for each random-effects equation. An `independent` covariance structure allows for a distinct variance for each random effect within a random-effects equation and assumes that all covariances are zero. `exchangeable` structure specifies one common variance for all random effects and one common pairwise covariance. `identity` is short for “multiple of the identity”; that is, all variances are equal and all covariances are zero. `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 random-effects equation is a factor-variable specification `R.varname`, in which case `covariance(identity)` is the default, and only `covariance(identity)` and `covariance(exchangeable)` are allowed.

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

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

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

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

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

```
. xtmixed 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* below for more information regarding the use of sampling weights in multilevel models.

Weighted estimation, whether frequency or sampling, is not supported under restricted maximum-likelihood estimation (REML).

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

`mle`, the default, specifies that the model be fit using maximum likelihood (ML).

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

`pwscale(scale_method)`, where *scale\_method* is

```
size | effective | gk
```

controls how sampling weights (if specified) are scaled in two-level models.

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

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

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

`pwscale()` is supported only with two-level models. See *Survey data* in *Remarks* below for more details on using `pwscale()`.

`residuals(rspec)`, where *rspec* is

```
restype [ , residual_options ]
```

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

*restype* is

```
independent | exchangeable | ar # | ma # | unstructured |  
banded # | toeplitz # | exponential
```

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

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

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

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

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

*restype banded #* is a special case of `unstructured` that restricts estimation to the covariances within the first *#* off-diagonals and sets the covariances outside this band to zero. 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 zero 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 zero, only the diagonal elements of the covariance matrix are estimated. *restype banded* may be combined with `by(varname)`, but *varname* must be constant within groups.

*restype toeplitz #* assumes that within-group errors have Toeplitz structure of order *#*, for which correlations are constant with respect to time lags less than or equal to *#* and are zero 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 one and the maximum observed lag (the default). Any nonconsecutive time values will be treated as gaps. For this structure, *#* + 1 parameters are estimated (*#* correlations and one overall error variance). *restype toeplitz* may be combined with `by(varname)`, but *varname* must be constant within groups.

`restype exponential` is a generalization of the autoregressive (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  $\rho$ , 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  $\rho$  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 ID the observations when `restype` is unstructured or banded.

---

#### SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification and that allow for intragroup correlation; see [R] [vce\\_option](#). `vce(oim)` is the default. If `vce(robust)` is specified, robust variances are clustered at the highest level in the multilevel model.

`vce(robust)` and `vce(cluster clustvar)` are not supported with REML estimation.

---

#### Reporting

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

`variance` displays the random-effects and residual-error parameter estimates as variances and covariances. The default is to display them as standard deviations and correlations.

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

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

`estmetric` displays all parameter estimates in the estimation metric. Fixed-effects estimates are unchanged from those normally displayed, but random-effects parameter estimates are displayed as log-standard deviations and hyperbolic arctangents of correlations, with equation names that organize them by model level. Residual-variance parameter estimates are also displayed in their original estimation metric.

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

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

`nostderr` prevents `xtmixed` 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.

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

*display\_options*: noomitted, vsquish, noemptycells, baselevels, allbaselevels, cformat(%fmt), pformat(%fmt), sformat(%fmt), and nolstretch; see [R] **estimation options**.

#### EM options

These options control the EM (expectation-maximization) 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 `xtmixed` are listed below.

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

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

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

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

The following option is available with `xtmixed` but is not shown in the dialog box:

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

## Remarks

Remarks are presented under the following headings:

- Introduction*
- Two-level models*
- Covariance structures*
- Likelihood versus restricted likelihood*
- Three-level models*
- Blocked-diagonal covariance structures*
- Heteroskedastic random effects*
- Heteroskedastic residual errors*
- Other residual-error structures*
- Random-effects factor notation and crossed-effects models*
- Diagnosing convergence problems*
- Distribution theory for likelihood-ratio tests*
- Survey data*

## 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,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + \boldsymbol{\epsilon} \quad (1)$$

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 zero and variance matrix  $\sigma_\epsilon^2 \mathbf{R}$ .

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

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

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

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

Comprehensive treatments of mixed models are provided by, among others, Searle, Casella, and McCulloch (1992); McCulloch, Searle, and Neuhaus (2008); Verbeke and Molenberghs (2000); Raudenbush and Bryk (2002); Demidenko (2004); and Pinheiro and Bates (2000). In particular, chapter 2 of Searle, Casella, and McCulloch (1992) provides an excellent history.

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

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

The most popular methods, however, are maximum likelihood (ML) and restricted maximum-likelihood (REML), and these are the two methods that are supported by `xtmixed`. 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,  $\beta$ , 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)

$$\mathbf{y}_j = \mathbf{X}_j\beta + \mathbf{Z}_j\mathbf{u}_j + \epsilon_j \quad (2)$$

for  $j = 1, \dots, M$ , with cluster  $j$  consisting of  $n_j$  observations. The response,  $\mathbf{y}_j$ , comprises the rows of  $\mathbf{y}$  corresponding with the  $j$ th cluster, with  $\mathbf{X}_j$  and  $\epsilon_j$  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  $\Sigma$ . The matrix  $\mathbf{Z}_i$  is the  $n_j \times q$  design matrix for the  $j$ th cluster random effects. Relating this to (1), note that

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

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.

Finally, we state our convention on counting and ordering model levels. Model (2) is what we call a *two-level* model, with extensions to three, four, or any number of levels. The observation  $y_{i,j}$  is for individual  $i$  within cluster  $j$ , and the individuals comprise the first level and the clusters comprise the second level of the model. In our hypothetical three-level model with classes nested within schools, the observations within schools (the students, presumably) would constitute the first level, the classes would constitute the second level, and the schools would constitute the third level.

This differs from certain citations in the classical ANOVA literature and texts such as Pinheiro and Bates (2000) but is the standard in the vast literature on hierarchical models, for example, Skrondal and Rabe-Hesketh (2004).

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

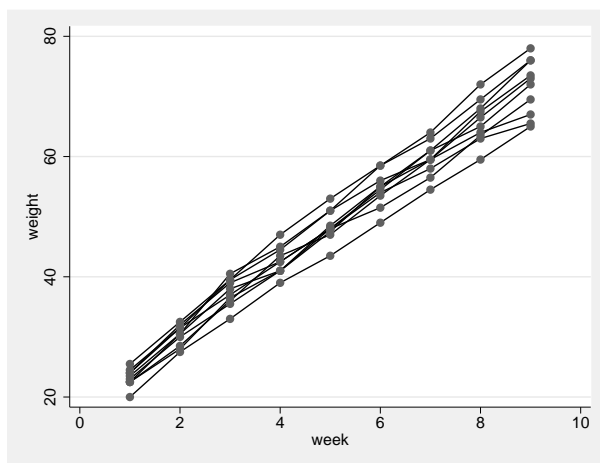
## Two-level models

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

### ► Example 1

Consider a longitudinal dataset, used by both Ruppert, Wand, and Carroll (2003) and Diggle et al. (2002), consisting of `weight` measurements of 48 pigs on 9 successive weeks. Pigs are identified by variable `id`. Below is a plot of the growth curves for the first 10 pigs.

```
. use http://www.stata-press.com/data/r12/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

$$\text{weight}_{ij} = \beta_0 + \beta_1 \text{week}_{ij} + u_j + \epsilon_{ij} \quad (4)$$

for  $i = 1, \dots, 9$  weeks and  $j = 1, \dots, 48$  pigs. The fixed portion of the model,  $\beta_0 + \beta_1 \text{week}_{ij}$ , 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

```

. xtmixed 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	sd(_cons)	3.849352	.4058119	3.130769	4.732866
	sd(Residual)	2.093625	.0755472	1.95067	2.247056

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

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

1. By typing “weight week”, we specified the response, `weight`, and the fixed portion of the model in the same way that we would if we were using `regress` or any other estimation command. Our fixed effects are a coefficient on `week` and a constant term.
2. When we added “|| id:”, we specified random effects at the level identified by 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 expectation-maximization (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 `xtmixed` 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,  $\Sigma = \sigma_u^2 \mathbf{I}$ . Because we have only one random effect at this level, `xtmixed` knew that `Identity` is the only possible covariance structure. In any case, the standard deviation of the level-two errors,  $\sigma_u$ , is estimated as 3.85 with standard error 0.406.

If you prefer variance estimates,  $\widehat{\sigma}_u^2$ , to standard deviation estimates,  $\widehat{\sigma}_u$ , then specify the `variance` option either at estimation or on `replay`.

7. The row labeled “`sd(Residual)`” displays the estimated standard deviation of the overall error term; that is,  $\widehat{\sigma}_\epsilon = 2.09$ . This is the standard deviation of the level-one errors, that is, the residuals.

8. Finally, a likelihood-ratio test comparing the model with one-level ordinary linear regression, model (4) without  $u_j$ , is provided and is highly significant for these data.

We now store our estimates for later use:

```
. estimates store randint
```

◀

## ► Example 2

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

$$\text{weight}_{ij} = \beta_0 + \beta_1 \text{week}_{ij} + u_{0j} + u_{1j} \text{week}_{ij} + \epsilon_{ij} \quad (5)$$

fit using `xtmixed`:

```
. xtmixed 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
```

```
Group variable: id
```

```
Number of obs      =      432
```

```
Number of groups   =       48
```

```
Obs per group: min =        9
```

```
                  avg =       9.0
```

```
                  max =        9
```

```
Wald chi2(1)       =    4689.51
```

```
Prob > chi2        =     0.0000
```

```
Log likelihood = -869.03825
```

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				
sd(week)	.6066851	.0660294	.4901417	.7509396
sd(_cons)	2.599301	.2969073	2.077913	3.251515
sd(Residual)	1.264441	.0487958	1.17233	1.363789

```
LR test vs. linear regression:      chi2(2) =    764.42  Prob > chi2 = 0.0000
```

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

```
. estimates store randslope
```

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

$$\Sigma = \text{Var} \begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} = \begin{bmatrix} \sigma_{u0}^2 & 0 \\ 0 & \sigma_{u1}^2 \end{bmatrix} \quad (6)$$

with  $\hat{\sigma}_{u0} = 2.60$  and  $\hat{\sigma}_{u1} = 0.61$ . 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  $\hat{\sigma}_{u1}$ , it would seem that the random slope is significant, and we can use `lrtest` and our two saved 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.

◀

## □ Technical note

At the bottom of the previous `xtmixed` output, there is a note stating that the likelihood ratio (LR) test comparing our model with standard linear regression is conservative. Also, our `lrtest` output warns us that our test comparing the random-slope model with the random-intercept model may be conservative if the null hypothesis is on the boundary. For the former, the null hypothesis is  $H_0: \sigma_{u0}^2 = \sigma_{u1}^2 = 0$ . For the latter, the null hypothesis is  $H_0: \sigma_{u1}^2 = 0$ . Because variances are constrained to be positive, both null hypotheses are on the boundaries of their respective parameter spaces. `xtmixed` is capable of detecting this automatically because it compares with linear regression. `lrtest`, on the other hand, can be used to compare a wide variety of nested mixed models, making automatic detection of boundary conditions impractical. With `lrtest`, the onus is on the user to verify testing on the boundary.

By “conservative”, we mean that when boundary conditions exist, the reported significance level is an upper bound on the actual significance; see *Distribution theory for likelihood-ratio tests* later in this entry for further details.

□

## □ Technical note

LR tests with REML require identical fixed-effects specifications for both models. As stated in Ruppert, Wand, and Carroll (2003), “The reason for this is that restricted likelihood is the likelihood of the residuals after fitting the fixed effects and so is not appropriate when there is more than one fixed effects model under consideration.” This is not an issue above because we used the default ML estimation, but had we fit the models using the `reml` option, we would have to confine our tests to models comparing different variance structures and not different  $\beta$ s.

In our example, the fixed-effects specifications for both models are identical ( $\beta_0 + \beta_1 \text{week}$ ), so either ML or REML would have produced valid LR tests.

Finally, `lrtest` is capable of detecting when you change fixed-effects structures under REML and will issue an error directing you to refit your models with ML. As such, there is no danger of making an inappropriate inference. □

## Covariance structures

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

### ► Example 3

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

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

```
. xtmixed weight week || id: week, covariance(unstructured) variance
(output omitted)
```

```
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
                                Prob > chi2      =     0.0000

Log likelihood = -868.96185
```

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

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

```
LR test vs. linear regression:      chi2(3) =   764.58   Prob > chi2 = 0.0000
```

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

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

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

In addition to specifying an alternate covariance structure, we specified the `variance` option to display variance components in the variance–covariance metric, rather than the default, which displays them as standard deviations and correlations.

◀

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 maximum likelihood (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

```
. xtmixed weight week || id: week, reml
(output omitted)
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				
sd(week)	.6135475	.0673971	.4947037	.7609413
sd(_cons)	2.630134	.3028832	2.09872	3.296107
sd(Residual)	1.26443	.0487971	1.172317	1.363781

```
LR test vs. linear regression:         chi2(2) =    765.92   Prob > chi2 = 0.0000
```

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

Although ML estimators are based on the usual likelihood theory, the idea behind REML is to transform the response into a set of linear contrasts whose distribution is free of the fixed effects  $\beta$ . The restricted likelihood is then formed by considering the distribution of the linear contrasts. Not only does this make the maximization problem free of  $\beta$ , it also incorporates the degrees of freedom used to estimate  $\beta$  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  $i = 1, \dots, n$ . The ML estimate of  $\sigma^2$  can be derived theoretically as the  $n$ -divided sample variance. The REML estimate can be derived by considering the first  $n - 1$  error contrasts,  $y_i - \bar{y}$ , whose joint distribution is free of  $\mu$ . Applying maximum likelihood to this distribution results in an estimate of  $\sigma^2$ , that is, the  $(n - 1)$  divided sample variance, which is unbiased for  $\sigma^2$ .

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 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 `xtmixed` 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,

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

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

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

and that  $\mathbf{u}_k^{(3)}$ ,  $\mathbf{u}_{jk}^{(2)}$ , 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  $\mathbf{Z}_{jk}^{(3)}$ , and for the second equation represents  $\mathbf{Z}_{jk}^{(2)}$ ; that is, you specify the levels top to bottom in `xtmixed`.

#### ► Example 4

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



Random-effects Parameters	Estimate	Std. Err.	[95% Conf. Interval]	
region: Identity				
sd(_cons)	.038087	.0170591	.0158316	.091628
state: Identity				
sd(_cons)	.0792193	.0093861	.0628027	.0999273
sd(Residual)	.0366893	.000939	.0348944	.0385766

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

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

Some items of note:

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—`xtmixed` 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. `xtmixed` treated the states as nested within regions, regardless of whether the codes for each state are unique between regions. That is, even if codes for states were duplicated between regions, `xtmixed` would have enforced the nesting and produced the same results.

The group information at the top of `xtmixed` output and that produced by the postestimation command `estat group` (see [XT] **xtmixed postestimation**) 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 `xtmixed`.

## Blocked-diagonal covariance structures

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

## ► Example 5

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

```
. xtmixed 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]	
<b>region: Independent</b>				
sd(hwy)	.0045717	.0120663	.0000259	.8066567
sd(unemp)	.0048777	.0013807	.0028007	.0084948
sd(_cons)	.0550901	.0786743	.0033533	.9050571
<b>state: Identity</b>				
sd(_cons)	.0797859	.0097832	.0627412	.101461
sd(Residual)	.0353108	.0009104	.0335708	.037141

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

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

```
. estimates store prodc
```

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)  $\Sigma_3$  is

$$\Sigma_3 = \begin{pmatrix} \text{hwy} & \text{unemp} & \text{\_cons} \\ \sigma_a^2 & 0 & 0 \\ 0 & \sigma_b^2 & 0 \\ 0 & 0 & \sigma_c^2 \end{pmatrix}$$

The random-effects specification at the state level remains unchanged; that is,  $\Sigma_2$  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.

Examining the estimated variance components reveals that the variances of the random coefficients on `hwy` and `unemp` could be treated as equal. That is,

$$\Sigma_3 = \begin{pmatrix} \text{hwy} & \text{unemp} & \text{\_cons} \\ \sigma_a^2 & 0 & 0 \\ 0 & \sigma_a^2 & 0 \\ 0 & 0 & \sigma_c^2 \end{pmatrix}$$

looks plausible. We can impose this equality constraint by treating  $\Sigma_3$  as block diagonal: the first block is a  $2 \times 2$  multiple of the identity matrix, that is,  $\sigma_a^2 \mathbf{I}_2$ ; the second is a scalar, equivalently, a  $1 \times 1$  multiple of the identity.

We construct block-diagonal covariances by repeating level specifications:

```
. xtmixed 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 sd(hwy unemp)	.0048802	.001376	.0028082	.0084809
region: Identity sd(_cons)	.0530951	.0286555	.0184356	.1529149
state: Identity sd(_cons)	.0797369	.0095999	.0629766	.1009577
sd(Residual)	.0353111	.0009104	.0335712	.0371413

```
LR test vs. linear regression:      chi2(3) = 1189.08   Prob > chi2 = 0.0000
```

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

We specified two equations for the `region` level: the first for the random coefficients on `hwy` and `unemp` with covariance set to `Identity` and the second for the random intercept `_cons`, whose covariance defaults to `Identity` because it is of dimension one. `xtmixed` labeled the estimate of  $\sigma_a$  as “`sd(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 . prodr
Likelihood-ratio test                LR chi2(1) =      0.00
(Assumption: . nested in prodr)      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, `xtmixed` 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, `xtmixed` 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.

□

## Heteroskedastic random effects

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

### ▷ Example 6

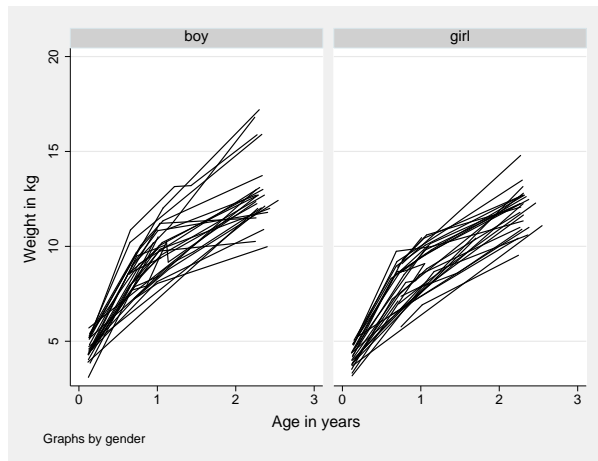
Following Rabe-Hesketh and Skrondal (2008, sec. 5.10), we analyze data from Asian children in a British community who were weighed up to four times, roughly between the ages of 6 weeks and 27 months. The dataset is a random sample of data previously analyzed by Goldstein (1986) and Prosser, Rasbash, and Goldstein (1991).

```
. use http://www.stata-press.com/data/r12/childweight
(Weight data on Asian children)
. describe
Contains data from http://www.stata-press.com/data/r12/childweight.dta
  obs:      198                Weight data on Asian children
  vars:      5                23 May 2011 15:12
  size:     3,168            (_dta has notes)
```

variable name	storage type	display format	value 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  $i$ th measurement on the  $j$ th child:

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

The above models overall mean growth as quadratic in age and allows for two child-specific random effects: a random intercept,  $u_{j0}$ , that represents each child's vertical shift from the overall mean ( $\beta_0$ ), and a random age slope,  $u_{j1}$ , that represents each child's deviation in linear growth rate from the overall mean linear growth rate ( $\beta_1$ ). For reasons of simplicity, we do not consider child-specific changes in the quadratic component of growth.

```
. xtmixed 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				
sd(age)	.5465535	.075708	.4166057	.7170347
sd(_cons)	.7087917	.0996506	.5380794	.9336647
sd(Residual)	.5561382	.0426951	.4784488	.6464426

```
LR test vs. linear regression:      chi2(2) = 114.70  Prob > chi2 = 0.0000
```

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

◀

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

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

```
. xtmixed 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]	
1.girl	-.5104676	.2145529	-2.38	0.017	-.9309835	-.0899516
girl#c.age						
0	7.806765	.2524583	30.92	0.000	7.311956	8.301574
1	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				
sd(age)	.5265782	.0730408	.4012307	.6910851
sd(_cons)	.6385054	.0969921	.4740922	.8599364
sd(Residual)	.5596163	.0426042	.4820449	.6496707

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

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

. estimates store homoskedastic

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

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

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

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

```

. gen boy = !girl
. gen boyXage = boy*age
. gen girlXage = girl*age
. xtmixed 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				
sd(boy)	.5622358	.138546	.3468691	.9113211
sd(boyXage)	.6880757	.1144225	.4966919	.9532031
id: Independent				
sd(girl)	.7614904	.1286769	.5467994	1.060476
sd(girlXage)	.257805	.1073047	.1140251	.582884
sd(Residual)	.5548717	.0418872	.4785591	.6433534

```
LR test vs. linear regression:      chi2(4) = 112.86  Prob > chi2 = 0.0000
```

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

```
. estimates store heteroskedastic
```

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

Our previous model had the random effects specification

```
|| id: age
```

which we have replaced with the dual repeated-level specification

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

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

There seems to be a large gender effect in the variability of linear growth rates. We can compare both models with a likelihood-ratio test, recalling that we saved 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 zero, the above does not test on the boundary, and 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  $\epsilon$ 's in the stated models—have been i.i.d. Gaussian with variance  $\sigma_\epsilon^2$ . This is demonstrated in `xtmixed` output in the random-effects table, where up until now we have estimated a single residual-error standard deviation or variance, labeled as `sd(Residual)` or `var(Residual)`, respectively.

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

### ► Example 7

West, Welch, and Galecki (2007, chap. 7) analyze data studying the effect of ceramic dental veneer placement on gingival (gum) health. Data on 55 teeth located in the maxillary arches of 12 patients were considered.

```
. use http://www.stata-press.com/data/r12/veneer, clear
(Dental veneer data)
. describe
Contains data from http://www.stata-press.com/data/r12/veneer.dta
obs:          110          Dental veneer data
vars:         7           24 May 2011 12:11
size:        1,100       (_dta has notes)
```

variable name	storage type	display format	value label	variable label
patient	byte	%8.0g		Patient ID
tooth	byte	%8.0g		Tooth number with patient
gcf	byte	%8.0g		Gingival crevicular fluid (GCF)
age	byte	%8.0g		Patient age
base_gcf	byte	%8.0g		Baseline GCF
cda	float	%9.0g		Average contour difference after veneer placement
followup	byte	%9.0g	t	Follow-up time: 3 or 6 months

Sorted by:

Veneers were placed to match the original contour of the tooth as closely as possible, and researchers were interested in how contour differences (variable `cda`) impacted gingival health. Gingival health was measured as the amount of gingival crevical fluid (GCF) at each tooth, measured at baseline (variable `base_gcf`) and at two posttreatment follow-ups at 3 and 6 months. Variable `gcf` records GCF at follow-up, and 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  $i$ th measurement of the  $j$ th tooth from the  $k$ th patient, we have

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

which we can fit using `xtmixed` as

```
. xtmixed 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

```
Log restricted-likelihood = -420.92761      Wald chi2(4)      =       7.48
                                           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					
sd(followup)	6.472072	1.452392	4.168943	10.04756	
sd(_cons)	22.91255	5.521438	14.28736	36.74472	
corr(followup,_cons)	-.9469371	.0394744	-.9878843	-.7827271	
tooth: Identity					
sd(_cons)	6.888932	1.207033	4.886635	9.711668	
sd(Residual)	6.990496	.7513934	5.662578	8.629822	

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

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

We used REML estimation above for no other reason than variety.

Among the other features of the model fit, we note that the residual standard deviation,  $\sigma_\epsilon$ , was estimated as 6.99 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 standard deviations: 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.

```
. xtmixed 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

```
Log restricted-likelihood = -420.4576          Wald chi2(4)      =       7.51
                                          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					
sd(followup)	6.461555	1.449333	4.163051	10.02911	
sd(_cons)	22.69806	5.55039	14.0554	36.65509	
corr(followup,_cons)	-.9480776	.0395764	-.9885662	-.7800707	
tooth: Identity					
sd(_cons)	6.881798	1.198038	4.892355	9.680234	
Residual: Independent, by followup					
3 months: sd(e)	7.833764	1.17371	5.840331	10.5076	
6 months: sd(e)	6.035612	1.240554	4.034281	9.029765	

```
LR test vs. linear regression:          chi2(5) =    92.06   Prob > chi2 = 0.0000
```

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

Comparison of both models via a likelihood-ratio test reveals the difference in residual standard deviations as 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 `xtmixed`: estimating one overall residual standard deviation/variance for the entire model.

## Other residual-error structures

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

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

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

By default, `xtmixed` will report estimates of the two parameters as estimates of the common standard deviation,  $\sigma_\epsilon$ , and of pairwise correlation. If the `variance` option is specified, you obtain estimates of  $\sigma_\epsilon^2$  and the covariance  $\sigma_1$ . 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 autoregressive 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 zero and variance  $\sigma_u^2$ . `xtmixed` reports estimates of  $\phi_1, \dots, \phi_p$  and the overall error standard deviation  $\sigma_\epsilon$  (or variance if the `variance` option is specified), 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  $\phi_1$  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 a moving average 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 zero and variance  $\sigma_u^2$ . `xtmixed` reports estimates of  $\theta_1, \dots, \theta_q$  and the overall error standard deviation  $\sigma_\epsilon$  (or variance if the `variance` option is specified), 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 ID 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 zero. As is the case with `unstructured`, the `t(varname)` is required, where `varname` is an ID 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 zero. 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:

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

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  $q = 0$ ,  $\text{Var}(\epsilon_j)$  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  $\rho_k$  is estimated; if the lag is greater than  $q$ , then  $\rho_k$  is assumed to be zero. If  $q$  is left unspecified, then  $\rho_k$  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  $\sigma_\epsilon^2$  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. `xtmixed` reports estimates of  $\sigma_\epsilon^2$  and  $\rho$ . When the `by(varname)` option is also specified, these two parameters are estimated for each level of *varname*.

## ► Example 8

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

```
. use http://www.stata-press.com/data/r12/ovary
(Ovarian follicles in mares)
. describe
Contains data from http://www.stata-press.com/data/r12/ovary.dta
  obs:          308          Ovarian follicles in mares
  vars:          6           20 May 2011 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  $i$ th measurement on the  $j$ th mare:

$$\text{follicles}_{ij} = \beta_0 + \beta_1 \sin 1_{ij} + \beta_2 \cos 1_{ij} + u_j + \epsilon_{ij}$$

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 autoregressive of order 2.

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

follicles	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
sin1	-2.899228	.5110786	-5.67	0.000	-3.900923	-1.897532
cos1	-.8652936	.5432926	-1.59	0.111	-1.930127	.1995402
_cons	12.14455	.9473617	12.82	0.000	10.28775	14.00134

Random-effects Parameters		Estimate	Std. Err.	[95% Conf. Interval]	
mare: Identity					
	sd(_cons)	2.663158	.8264476	1.449596	4.892683
Residual: AR(2)					
	phi1	.5386104	.0624899	.4161325	.6610883
	phi2	.144671	.063204	.0207934	.2685486
	sd(e)	3.775055	.3225437	3.192979	4.463244

LR test vs. linear regression:  $\chi^2(3) = 251.67$  Prob >  $\chi^2 = 0.0000$

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

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

◀

## ▶ Example 9

Fitzmaurice, Laird, and Ware (2004, chap. 7) analyzed data on 37 subjects who participated in an exercise therapy trial.

```
. use http://www.stata-press.com/data/r12/exercise
(Exercise Therapy Trial)
. describe
Contains data from http://www.stata-press.com/data/r12/exercise.dta
  obs:          259          Exercise Therapy Trial
  vars:          4           24 Jun 2010 18:35
  size:         1,036        (_dta has notes)
```

variable name	storage type	display format	value 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 at every two days over the next twelve days.

Following Fitzmaurice, Laird, and Ware (2004, 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 (day 0) 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)
. xtmixed 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
                                      Prob > chi2       =       0.0000

Log likelihood = -296.58215

```

strength	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
2.program	1.360119	1.003549	1.36	0.175	-.6068016	3.32704
day						
4	1.125	.3322583	3.39	0.001	.4737858	1.776214
6	1.360127	.3766894	3.61	0.000	.6218298	2.098425
8	1.583563	.4905876	3.23	0.001	.6220287	2.545097
12	1.623576	.5372947	3.02	0.003	.5704977	2.676654
program#day						
2 4	-.169034	.4423472	-0.38	0.702	-1.036019	.6979505
2 6	.2113012	.4982385	0.42	0.671	-.7652283	1.187831
2 8	-.1299763	.6524813	-0.20	0.842	-1.408816	1.148864
2 12	.3212829	.7306782	0.44	0.660	-1.11082	1.753386
_cons	79.6875	.7560448	105.40	0.000	78.20568	81.16932

Random-effects Parameters	Estimate	Std. Err.	[95% Conf. Interval]	
id:	(empty)			
Residual: Unstructured				
sd(e0)	3.024179	.3515413	2.408024	3.797993
sd(e4)	3.445452	.4007049	2.743164	4.327536
sd(e6)	3.17265	.3701737	2.524102	3.987837
sd(e8)	3.636569	.42814	2.88721	4.580421
sd(e12)	3.628924	.4364031	2.866916	4.593468
corr(e0,e4)	.9237568	.0241659	.8594221	.9592936
corr(e0,e6)	.8847673	.0360256	.7902873	.9381525
corr(e0,e8)	.8438552	.0481853	.7193946	.915815
corr(e0,e12)	.8107881	.0591609	.6589166	.899148
corr(e4,e6)	.9598061	.0131155	.9242041	.9788692
corr(e4,e8)	.949579	.016828	.9036835	.9739048
corr(e4,e12)	.9024383	.0333957	.81189	.9505891
corr(e6,e8)	.957802	.0157897	.9127914	.9798265
corr(e6,e12)	.9120406	.0293324	.8329488	.9546129
corr(e8,e12)	.9403092	.0213539	.8808047	.9705727

LR test vs. linear regression:      chi2(14) =   314.67   Prob > chi2 = 0.0000

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

Because we are using 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 standard error for each day and a distinct correlation for each pair of days.

That there is very high positive correlation between all pairs of measurements is evident, but what is not as evident is whether the pairwise correlation 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 autoregressive or moving-average structure.

```
. estimates store unstructured
. xtmixed 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
                                Prob > chi2        =      0.0000
Log likelihood = -307.83324
```

Random-effects Parameters	Estimate	Std. Err.	[95% Conf. Interval]	
id: (empty)				
Residual: Exponential				
rho	.9786462	.0051238	.9659207	.9866854
sd(e)	3.350148	.3489952	2.73144	4.109001

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

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

In the above example, we suppressed displaying the main regression parameters because they did not differ much from those of the previous model. While the unstructured model estimated fifteen variance–covariance parameters, the exponential model claims to get the job done with just two, 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.
```

## Random-effects factor notation and crossed-effects models

Not all mixed models contain nested levels of random effects.

### ▷ Example 10

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

$$\text{weight}_{ij} = \beta_0 + \beta_1 \text{week}_{ij} + u_i + v_j + \epsilon_{ij} \quad (8)$$

for the  $i = 1, \dots, 9$  weeks and  $j = 1, \dots, 48$  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  $\beta_0 + \beta_1 \text{week}$  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 `xtmixed` as repeated-level equations. All we need is an ID variable to identify all the observations as one big group and a way to tell `xtmixed` to treat `week` and `pig` as factor variables (or equivalently, as two sets of overparameterized indicator variables identifying weeks and pigs, respectively). `xtmixed` supports the special group designation `_all` for the former and the factor notation `R.varname` for the latter.

```

. use http://www.stata-press.com/data/r12/pig, clear
(Longitudinal analysis of pig weights)
. xtmixed 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
                                         Prob > chi2       =      0.0000
Log likelihood = -1013.824

```

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 sd(R.week)	.2915259	.1490187	.107046	.7939333
_all: Identity sd(R.id)	3.851783	.4058045	3.133167	4.73522
sd(Residual)	2.073	.0756007	1.929997	2.226598

```
LR test vs. linear regression:          chi2(2) = 474.85   Prob > chi2 = 0.0000
```

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

```
. estimates store crossed
```

Thus we estimate  $\hat{\sigma}_u = 0.29$  and  $\hat{\sigma}_v = 3.85$ . 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 AIC values by using `estimates stats`,

```

. quietly xtmixed weight week || id:week
. estimates stats crossed .

```

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. `xtmixed` does not fit a constant-only model as part of its usual estimation of the full model, but you can use `xtmixed` 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 use R.*varname*, `xtmixed` 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 two 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

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

instead of

```
. xtmixed 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 one (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

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

as a direct way to demonstrate factor 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):

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

We leave it to you to verify that both produce identical results. See Rabe-Hesketh and Skrondal (2008, chap. 11) for more techniques for making calculations more efficient in more complex models. □

## ▷ Example 11

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

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

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 ones  $\mathbf{J}_k$ , and

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

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

```
. use http://www.stata-press.com/data/r12/productivity
(Public Capital Productivity)
. xtmixed gsp private emp hwy water other unemp || region: R.state,
> cov(exchangeable) variance
(output omitted)
Mixed-effects ML regression          Number of obs   =      816
Group variable: region              Number of groups =       9
                                     Obs per group: min =       51
                                     avg =          90.7
                                     max =          136
                                     Wald chi2(6)     =  18829.06
Log likelihood = 1430.5017          Prob > chi2     =    0.0000
```

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

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

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

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

The estimates of the fixed effects and their standard errors are equivalent to those from example 4, and remapping the variance components from  $(\sigma_3^2 + \sigma_2^2, \sigma_3^2, \sigma_\epsilon^2)$ , as displayed here, to  $(\sigma_3, \sigma_2, \sigma_\epsilon)$ , 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 the class of linear mixed models, you will find that some models “fail to converge” when used with your data. The default gradient-based method used by `xtmixed` is the Newton–Raphson algorithm, requiring the calculation of a gradient vector and Hessian (second-derivative) matrix; see [R] `ml`.

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

1. repeated “nonconcave” or “backed-up” iterations without convergence;
2. a Hessian (second-derivative) calculation that has become asymmetric, unstable, or has missing values; or

3. the message “standard-error calculation has failed” when computing standard errors.

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

Such behavior is usually the result of either

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

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

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

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

In unweighted models with independent and homoskedastic residuals, one useful way to diagnose problems of nonconvergence is to rely on the expectation-maximization (EM) algorithm (Dempster, Laird, and Rubin 1977), normally used by `xtmixed` 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 zero, this provides evidence that reason B is the cause of the nonconvergence of the gradient-based method, in which case the solution would be to drop the offending variance component from the model. If no estimated variance components are near zero, reason A could be the culprit.

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

## Distribution theory for likelihood-ratio tests

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

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

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

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

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

Because the random component of the mixed model comprises three parameters ( $\sigma_0^2, \sigma_{01}, \sigma_1^2$ ), on the surface it would seem that the LR comparison test would be distributed as  $\chi_3^2$ . However, two complications need to be considered. First, the variances  $\sigma_0^2$  and  $\sigma_1^2$  are restricted to be positive, and second, constraints such as  $\sigma_1^2 = 0$  implicitly restrict the covariance  $\sigma_{01}$  to be zero as well. From a technical standpoint, it is unclear how many parameters must be restricted to reduce the model to linear regression.

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

`xtmixed` uses this reference distribution, the  $\chi^2$  with full degrees of freedom, to produce a conservative test and places a note in the output labeling the test as such. Because the displayed significance level is an upper bound, rejection of the null hypothesis based on the reported level would imply rejection on the basis of the actual level.

## □ Technical note

It may seem that `xtmixed` does not follow Stata's standard syntax for multiple-equation models, but it does. In example 2, we typed

```
. xtmixed weight week || id:
```

but we could have used the standard `multiequation` syntax:

```
. xtmixed (weight week) (id:)
```

`xtmixed` will understand either and produce the same results. We prefer the syntax using `||` because it better emphasizes the nested structure of the levels.

□

## Survey data

Multilevel modeling of survey data is a little different than 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  $\pi_{ij}$  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 `xtmixed`. 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  $w_{i|j}$ . The reason it is an issue is that 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.

`xtmixed` 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 one 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 Pfeiffermann 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 if you have  $w_{i|j}$  or  $w_{ij}$ . For the `pwscale(size)` and `pwscale(effective)` methods, you get identical results, and even though `pwscale(gk)` assumes  $w_{i|j}$ , you can obtain this as  $w_{i|j} = w_{ij}/w_j$  before proceeding.

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

## ► Example 12

Rabe-Hesketh and Skrondal (2006) analyzed their 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/r12/pisa2000
(Programme for International Student Assessment (PISA) 2000 data)
. describe
Contains data from http://www.stata-press.com/data/r12/pisa2000.dta
  obs:          2,069          Programme for International Student
                               Assessment (PISA) 2000 data
  vars:          11           12 Jun 2011 10:08
  size:         37,242       (_dta has notes)
```

variable name	storage type	display format	value label	variable label
female	byte	%8.0g		1 if female
isei	byte	%8.0g		International socio-economic index
w_fstuw	float	%9.0g		Student-level weight
w_nrschbw	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 variable `id_school` identifies the schools, variable `w_fstuw` is a student-level overall inclusion weight ( $w_{ij}$ , not  $w_{i|j}$ ) adjusted for noninclusion and nonparticipation of students, and variable `w_nrschbw` 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  $w_{i|j}$ .

```

. xtmixed isei female high_school college one_for both_for test_lang
> [pw=wfstuwt] || id_school: , pweight(wnrshbw) pwscale(size)
(output omitted)
Mixed-effects regression                Number of obs    =    2069
Group variable: id_school              Number of groups =    148
                                      Obs per group: min =     1
                                      avg      =    14.0
                                      max      =    28

                                      Wald chi2(6)     =   73452.90
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				
sd(_cons)	5.890139	.7279	4.623113	7.50441
sd(Residual)	14.7898	.3793531	14.06466	15.55232

Some notes:

1. We specified the level-one weights using standard Stata weight syntax, that is, [pw=wfstuwt].
2. We specified the level-two weights via the `pweight(wnrshbw)` option as part of the random-effects specification for the `id_school` level. As such, it is treated as a school-level weight. Accordingly, `wnrshbw` needs to be constant within schools, and `xtmixed` 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  $w_{ij}$ , you want to first divide  $w_{ij}$  by  $w_j$ . But you can handle that within the weight specification itself.

```

. xtmixed isei female high_school college one_for both_for test_lang
> [pw=w_fstwt/wnrschbw] || id_school:, pweight(wnrschbw) pwscale(gk)
(output omitted)
Mixed-effects regression                Number of obs    =    2069
Group variable: id_school                Number of groups =    148
                                         Obs per group: min =     1
                                         avg =    14.0
                                         max =    28

                                         Wald chi2(6)     = 321599.34
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				
sd(_cons)	5.628074	.6034248	4.561384	6.944213
sd(Residual)	15.04137	.2709432	14.5196	15.5819

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

◀

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.

## Saved results

xtmixed saves the following in `e()`:

### Scalars

<code>e(N)</code>	number of observations
<code>e(k)</code>	number of parameters
<code>e(k_f)</code>	number of FE parameters
<code>e(k_r)</code>	number of RE parameters
<code>e(k_rs)</code>	number of standard deviations
<code>e(k_rc)</code>	number of correlations
<code>e(k_res)</code>	number of residual-error parameters
<code>e(N_clust)</code>	number of clusters
<code>e(nrgroups)</code>	number of residual-error <code>by()</code> groups
<code>e(ar_p)</code>	AR order of residual errors, if specified
<code>e(ma_q)</code>	MA order of residual errors, if specified
<code>e(res_order)</code>	order of residual-error structure, if appropriate
<code>e(df_m)</code>	model degrees of freedom
<code>e(ll)</code>	log (restricted) likelihood
<code>e(chi2)</code>	$\chi^2$
<code>e(p)</code>	$p$ -value for $\chi^2$
<code>e(ll_c)</code>	log likelihood, comparison model
<code>e(chi2_c)</code>	$\chi^2$ , comparison model
<code>e(df_c)</code>	degrees of freedom, comparison model
<code>e(p_c)</code>	$p$ -value, comparison model
<code>e(rank)</code>	rank of <code>e(V)</code>
<code>e(rc)</code>	return code
<code>e(converged)</code>	1 if converged, 0 otherwise

### Macros

<code>e(cmd)</code>	<code>xtmixed</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of dependent variable
<code>e(wtype)</code>	weight type (first-level weights)
<code>e(wexp)</code>	weight expression (first-level weights)
<code>e(fweight<math>k</math>)</code>	$fweight$ expression for $k$ th highest level, if specified
<code>e(pweight<math>k</math>)</code>	$pweight$ expression for $k$ th highest level, if specified
<code>e(ivals)</code>	grouping variables
<code>e(title)</code>	title in estimation output
<code>e(redim)</code>	random-effects dimensions
<code>e(vartypes)</code>	variance-structure types
<code>e(revars)</code>	random-effects covariates
<code>e(resopt)</code>	<code>residuals()</code> specification, as typed
<code>e(rstructure)</code>	residual-error structure
<code>e(rstructlab)</code>	residual-error structure output label
<code>e(rbyvar)</code>	residual-error <code>by()</code> variable, if specified
<code>e(rglabels)</code>	residual-error <code>by()</code> groups labels
<code>e(pwscale)</code>	sampling-weight scaling method
<code>e(timevar)</code>	residual-error <code>t()</code> variable, if specified
<code>e(chi2type)</code>	Wald; type of model $\chi^2$ test
<code>e(clustvar)</code>	name of cluster variable
<code>e(vce)</code>	<code>vcetype</code> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. Err.
<code>e(method)</code>	ML or REML
<code>e(opt)</code>	type of optimization
<code>e(optmetric)</code>	<code>matsqrt</code> or <code>matlog</code> ; random-effects matrix parameterization
<code>e(emonly)</code>	<code>emonly</code> , if specified
<code>e(ml_method)</code>	type of ml method
<code>e(technique)</code>	maximization technique
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

Matrices	
e(b)	coefficient vector
e(N_g)	group counts
e(g_min)	group-size minimums
e(g_avg)	group-size averages
e(g_max)	group-size maximums
e(tmap)	ID mapping for unstructured residual errors
e(V)	variance–covariance matrix of the estimator
e(V_modelbased)	model-based variance
Functions	
e(sample)	marks estimation sample

## Methods and formulas

xtmixed is implemented as an ado-file that uses Mata.

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 zero and variance matrix  $\sigma_\epsilon^2 \mathbf{I}_n$ . We also assume that  $\mathbf{u}$  has variance–covariance matrix  $\mathbf{G}$  and that  $\mathbf{u}$  is orthogonal to  $\boldsymbol{\epsilon}$  so that

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

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

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

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

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

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

$$L_R(\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma_\epsilon^2) = L(\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma_\epsilon^2) - \frac{1}{2} \log |\mathbf{X}' \mathbf{V}^{-1} \mathbf{X}| \quad (10)$$

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

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

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

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:

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

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  $\beta$ ,  $\sigma_\epsilon^2$ , and  $\mathbf{\Delta}$  (the unique elements of  $\mathbf{\Delta}$ , that is) are obtained by maximizing the profile log likelihood (profiled in  $\mathbf{\Delta}$ )

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

where  $\|\cdot\|$  denotes the 2-norm, and following this maximization with

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

REML estimates are obtained by maximizing

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

followed by

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

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

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

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

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

with the analogous expression for REML.

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

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

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

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 `xtmixed` output, variance components are most often displayed either as variances and covariances or as standard deviations and correlations.

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

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

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

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

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

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

3. Maximization step: Maximize  $D(\boldsymbol{\Sigma})$  to produce  $\boldsymbol{\Sigma}^{(t+1)}$ .

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

For general residual-error structures,

$$\text{Var}(\epsilon_j) = \sigma_\epsilon^2 \Lambda_j$$

where the subscript  $j$  merely represents that  $\epsilon_j$  and  $\Lambda_j$  vary in dimension in unbalanced data, the data are first transformed according to

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

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  $\Lambda$ ,  $\rho$ , are estimated along with the fixed effects and variance components. Because  $\sigma_\epsilon^2$  is always estimated and multiplies the entire  $\Lambda_j$  matrix,  $\widehat{\rho}$  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

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

where  $w_j$  is the inverse of the probability of selection for the  $j$ th cluster,  $w_{i|j}$  is the inverse of the conditional probability of selection of individual  $i$  given the selection of cluster  $j$ , and  $f_1()$  and  $f_2()$  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 so as to reflect replicated clusters for  $w_j$  and replicated observations within clusters for  $w_{i|j}$ . Because this estimation is based on replicated clusters and observations, frequency weights are handled similarly.

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

Under `pwscale(size)`,

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

Under `pwscale(effective)`,

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

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

$$w_j^* = n_j^{-1} \sum_{i=1}^{n_j} w_{i|j} w_j; \quad w_{i|j}^* = 1$$

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  $L_j(\beta, \Sigma, \sigma_\epsilon^2)$  with respect to  $\{\beta, \alpha, \log(\sigma_\epsilon)\}$ , 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 three-level models and higher, see Bates and Pinheiro (1998) and Rabe-Hesketh and Skrondal (2006).

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

## Acknowledgments

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

## References

- Andrews, M., T. Schank, and R. Upward. 2006. Practical fixed-effects estimation methods for the three-way error-components model. *Stata Journal* 6: 461–481.
- Baltagi, B. H., S. H. Song, and B. C. Jung. 2001. The unbalanced nested error component regression model. *Journal of Econometrics* 101: 357–381.
- Bates, D. M., and J. C. Pinheiro. 1998. Computational methods for multilevel modelling. In *Technical Memorandum BL0112140-980226-01TM*. Murray Hill, NJ: Bell Labs, Lucent Technologies.  
<http://stat.bell-labs.com/NLME/CompMulti.pdf>.
- Cameron, A. C., and P. K. Trivedi. 2010. *Microeconometrics Using Stata*. Rev. ed. College Station, TX: Stata Press.
- Canette, I. 2011. Including covariates in crossed-effects models. The Stata Blog: Not Elsewhere Classified.  
<http://blog.stata.com/2011/12/22/ncluding-covariates-in-crossed-effects-models/>
- Carle, A. C. 2009. Fitting multilevel models in complex survey data with design weights: Recommendations. *BMC Medical Research Methodology* 9: 49.
- Demidenko, E. 2004. *Mixed Models: Theory and Applications*. Hoboken, NJ: Wiley.
- Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B* 39: 1–38.
- Diggle, P. J., P. J. Heagerty, K.-Y. Liang, and S. L. Zeger. 2002. *Analysis of Longitudinal Data*. 2nd ed. Oxford: Oxford University Press.
- Fitzmaurice, G. M., N. M. Laird, and J. H. Ware. 2004. *Applied Longitudinal Analysis*. Hoboken, NJ: Wiley.
- Goldstein, H. 1986. Efficient statistical modelling of longitudinal data. *Annals of Human Biology* 13: 129–141.
- Graubard, B. I., and E. L. Korn. 1996. Modelling the sampling design in the analysis of health surveys .
- Gutierrez, R. G., S. Carter, and D. M. Drukker. 2001. sg160: On boundary-value likelihood-ratio tests. *Stata Technical Bulletin* 60: 15–18. Reprinted in *Stata Technical Bulletin Reprints*, vol. 10, pp. 269–273. College Station, TX: Stata Press.

- Harville, D. A. 1977. Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association* 72: 320–338.
- Henderson, C. R. 1953. Estimation of variance and covariance components. *Biometrics* 9: 226–252.
- Hocking, R. R. 1985. *The Analysis of Linear Models*. Monterey, CA: Brooks/Cole.
- Horton, N. J. 2011. Stata tip 95: Estimation of error covariances in a linear model. *Stata Journal* 11: 145–148.
- Laird, N. M., and J. H. Ware. 1982. Random-effects models for longitudinal data. *Biometrics* 38: 963–974.
- LaMotte, L. R. 1973. Quadratic estimation of variance components. *Biometrics* 29: 311–330.
- Marchenko, Y. V. 2006. Estimating variance components in Stata. *Stata Journal* 6: 1–21.
- McCulloch, C. E., S. R. Searle, and J. M. Neuhaus. 2008. *Generalized, Linear, and Mixed Models*. 2nd ed. Hoboken, NJ: Wiley.
- McLachlan, G. J., and K. E. Basford. 1988. *Mixture Models*. New York: Dekker.
- Munnell, A. 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.
- 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.
- . 2008. *Multilevel and Longitudinal Modeling Using Stata*. 2nd ed. College Station, TX: Stata Press.
- Rao, C. R. 1973. *Linear Statistical Inference and Its Applications*. 2nd ed. New York: Wiley.
- Raudenbush, S. W., and A. S. Bryk. 2002. *Hierarchical Linear Models: Applications and Data Analysis Methods*. 2nd ed. Thousand Oaks, CA: Sage.
- Ruppert, D., M. P. Wand, and R. J. Carroll. 2003. *Semiparametric Regression*. Cambridge: Cambridge University Press.
- 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.
- Self, S. G., and K.-Y. Liang. 1987. Asymptotic properties of maximum likelihood estimators and likelihood ratio tests under nonstandard conditions. *Journal of the American Statistical Association* 82: 605–610.
- Skrondal, A., and S. Rabe-Hesketh. 2004. *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models*. Boca Raton, FL: Chapman & Hall/CRC.
- Stram, D. O., and J. W. Lee. 1994. Variance components testing in the longitudinal mixed effects model. *Biometrics* 50: 1171–1177.
- Thompson, W. A., Jr. 1962. The problem of negative estimates of variance components. *Annals of Mathematical Statistics* 33: 273–289.
- Verbeke, G., and G. Molenberghs. 2000. *Linear Mixed Models for Longitudinal Data*. New York: Springer.
- West, B. T., K. B. Welch, and A. T. Galecki. 2007. *Linear Mixed Models: A Practical Guide Using Statistical Software*. Boca Raton, FL: Chapman & Hall/CRC.

## Also see

[XT] **xtmixed postestimation** — Postestimation tools for xtmixed

[XT] **xtmelogit** — Multilevel mixed-effects logistic regression

[XT] **xtmepoisson** — Multilevel mixed-effects Poisson regression

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

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

[XT] **xtgee** — Fit population-averaged panel-data models by using GEE

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

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