

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] [, 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	
<u>noconstant</u>	suppress the constant term from the fixed-effects equation

<i>re_options</i>	description
Model	
<u>covariance</u> (<i>vartype</i>)	variance–covariance structure of the random effects
<u>noconstant</u>	suppress the constant term from the random-effects equation
<u>collinear</u>	keep collinear variables

<i>options</i>	description
Estimation	
<u>reml</u>	fit model via maximum restricted likelihood; the default
<u>mle</u>	fit model via maximum likelihood

Reporting

<u>level</u> (#)	set confidence level; default is level(95)
<u>variance</u>	show random-effects parameter estimates as variances and covariances
<u>noret</u> able	suppress random-effects table
<u>nofe</u> table	suppress fixed-effects table
<u>est</u> metric	show parameter estimates in the estimation metric
<u>no</u> header	suppress output header
<u>nog</u> roup	suppress table summarizing groups
<u>no</u> stderr	do not estimate standard errors of random-effects parameters
<u>no</u> lrtest	do not perform LR test comparing to linear regression

EM options

<u>e</u> miterate(#)	number of EM iterations; default is 20
<u>e</u> mtolerance(#)	EM convergence tolerance; default is 1e-10
<u>e</u> monly	fit model exclusively using EM
<u>e</u> mlog	show EM iteration log
<u>e</u> mdots	show EM iterations as dots

Max options

<u>maximize</u> _options	control the maximization process
--------------------------	----------------------------------

<i>vartype</i>	description
<u>i</u> ndependent	one unique variance parameter per random effect, all covariances zero; the default unless a factor variable is specified
<u>e</u> xchangeable	equal variances for random effects, and one common pairwise covariance
<u>i</u> dentity	equal variances for random effects, all covariances zero
<u>u</u> nstructured	all variances–covariances distinctly estimated

depvar and *indepvars* may contain time-series operators; see [U] **11.4.3 Time-series varlists**.

by, rolling, statsby, and xi are allowed; see [U] **11.1.10 Prefix commands**.

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

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 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. The error distribution of the linear mixed model is assumed to be Gaussian.

Options

Model

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

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

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

specifies the structure of the covariance matrix for the random effects and may be specified for each random-effects equation. An `independent` covariance structure allows for a distinct variance for each random effect within a random-effects equation and assumes that all covariances are 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.

Only `covariance(identity)` and `covariance(exchangeable)` are allowed with the factor-variable specification.

`collinear` specifies that `xtmixed` not remove the 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 option `collinear` allows the estimation to take place with the random-effects equation intact.

Estimation

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

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

`mle` specifies that the model be fitted using maximum likelihood (ML).

Reporting

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

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

`norettable` suppresses the table of random effects.

`nofetable` suppresses the table of fixed effects.

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

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

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

`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.

`nolrtest` 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.

EM options

`emiter(#)` specifies the number of EM (expectation-maximization) iterations to perform. The default is 20.

`emtolerance(#)` specifies the convergence tolerance for the EM algorithm. The default is $1e-10$. EM iterations stop once the log (restricted) likelihood changes by a relative amount less than `#`. At that point, maximization 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.

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

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

Max options

`maximize_options`: `difficult`, `technique(algorithm_spec)`, `iterate(#)`, `[no]log`, `trace`, `gradient`, `showstep`, `hessian`, `shownrtolerance`, `tolerance(#)`, `ltolerance(#)`, `gtolerance(#)`, `nrtolerance(#)`, `nonnrtolerance`; see [R] `maximize`.

For option `technique()`, the default is `technique(nr)`, and algorithm `bhhh` is not allowed.

Remarks

Remarks are presented under the following headings:

- Introduction*
- One-level models*
- Covariance structures*
- Likelihood versus restricted likelihood*
- Two-level models*
- Blocked-diagonal covariance structures*
- Factor notation and crossed-effects models*
- Diagnosing convergence problems*
- Distribution theory for likelihood-ratio tests*

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 a $n \times p$ 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 $\sigma_{\epsilon}^2 \mathbf{I}_n$.

The fixed portion of (1), $\mathbf{X}\boldsymbol{\beta}$, is analogous to the linear predictor from a standard OLS regression model with $\boldsymbol{\beta}$ 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

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

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 residual variance σ_{ϵ}^2 .

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-panel correlation. Subjects within the same panel 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.

Overviews of mixed models are provided by, among others, Searle, Casella, and McCulloch (1992); McCulloch and Searle (2001); Verbeke and Molenberghs (2000); Raudenbush and Bryk (2002); and Pinheiro and Bates (2000). In particular, chapter 2 of Searle, Casella, and McCulloch (1992) provides an excellent history.

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

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

The most popular methods, however, are 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, $\boldsymbol{\beta}$, but instead depend only on the variance components to be estimated. You then apply ML methods using the distribution of the linear contrasts to form the likelihood.

Returning to (1): in panel-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 panels

$$\mathbf{y}_i = \mathbf{X}_i\boldsymbol{\beta} + \mathbf{Z}_i\mathbf{u}_i + \boldsymbol{\epsilon}_i \quad (2)$$

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

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

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 panels 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 level of random variation. For example, if classes are nested within schools, then (2) can be generalized to allow random effects at both the school and at the class-within-school levels. This we demonstrate later.

Finally, using formulation (2) and its multilevel extensions requires one important convention of terminology. Model (2) is what we call a *one-level* model, with extensions to two, three, or any number of levels. In our hypothetical two-level model with classes nested within schools, the schools are considered the first level and classes, the second level of the model. This is generally accepted terminology but differs from that of the literature on hierarchical models, e.g., Skrondal and Rabe-Hesketh (2004). In that literature, our schools and classes model would be considered a three-level model, with the students forming the first level, classes the second, and schools the third. Not only is there one more level, students, but the order is reversed.

One-level models

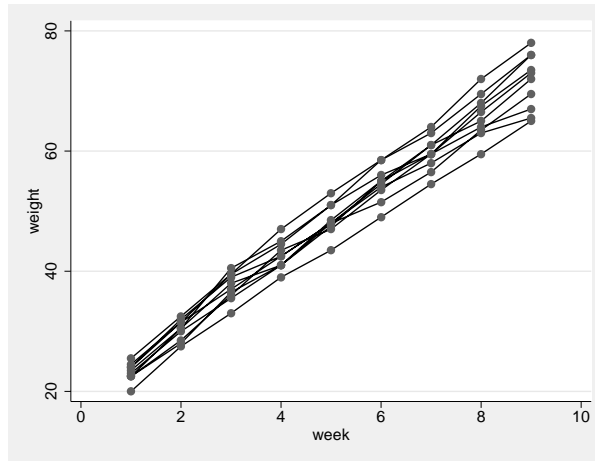
We begin with a simple application of (2).

▷ Example 1

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

```
. use http://www.stata-press.com/data/r10/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 following model

$$\text{weight}_{ij} = \beta_0 + \beta_1 \text{week}_{ij} + u_i + \epsilon_{ij} \quad (3)$$

for $i = 1, \dots, 48$ pigs and $j = 1, \dots, 9$ weeks. 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_i serves to shift this regression line up or down according to each pig. Since the random effects occur at the pig level (*id*), we fit the model by typing

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 Ph.D., 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 prediction has proved to be one of the main roots of current mixed-model methods.

```

. xtmixed weight week || id:
Performing EM optimization:
Performing gradient-based optimization:
Iteration 0:  log restricted-likelihood = -1016.8984
Iteration 1:  log restricted-likelihood = -1016.8984
Computing standard errors:
Mixed-effects REML regression      Number of obs      =      432
Group variable: id                 Number of groups   =      48
                                   Obs per group: min =      9
                                   avg      =      9.0
                                   max      =      9

                                   Wald chi2(1)        = 25271.50
Log restricted-likelihood = -1016.8984  Prob > chi2        = 0.0000

```

weight	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
week	6.209896	.0390633	158.97	0.000	6.133333	6.286458
_cons	19.35561	.603139	32.09	0.000	18.17348	20.53774

Random-effects Parameters		Estimate	Std. Err.	[95% Conf. Interval]	
id: Identity					
	sd(_cons)	3.891253	.4143198	3.158334	4.794252
	sd(Residual)	2.096356	.0757444	1.953034	2.250195

```
LR test vs. linear regression: chibar2(01) = 473.15 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 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`, i.e., the pig level. Since 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 option `emlog`.
 - (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 REML regression”, informs us that our model was fitted using REML, the default. For ML estimates, use option `mle`.

Since this model is a simple random-intercept model, specifying option `mle` would be equivalent to using `xtreg`, also with option `mle`.

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, $\Sigma = \sigma_u^2 \mathbf{I}$. Since we have only one random effect at this level, xtmixed knew that Identity is the only possible covariance structure. In any case, σ_u is estimated as 3.89 with standard error 0.414.

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

7. The row labeled “sd(Residual)” displays the estimated standard deviation of the overall error term; i.e., $\hat{\sigma}_\epsilon = 2.10$.
8. Finally, a likelihood-ratio test comparing the model to ordinary linear regression, model (3) without u_i , is provided and is highly significant for these data.

We now store our estimates for later use.

```
. estimates store randint
```

◀

► Example 2

Extending (3) to allow for a random slope on week yields the model

$$\text{weight}_{ij} = \beta_0 + \beta_1 \text{week}_{ij} + u_{0i} + u_{1i} \text{week}_{ij} + \epsilon_{ij} \quad (4)$$

fitted using xtmixed:

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

weight	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]
week	6.209896	.0916386	67.77	0.000	6.030287 6.389504
_cons	19.35561	.4021142	48.13	0.000	18.56748 20.14374

Random-effects Parameters	Estimate	Std. Err.	[95% Conf. Interval]
id: Independent			
sd(week)	.6135471	.0673971	.4947035 .7609409
sd(_cons)	2.630132	.302883	2.098719 3.296105
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.

```
. estimates store randslope
```

Since we didn't specify a covariance structure for the random effects $(u_{0i}, u_{1i})'$, `xtmixed` used the default `Independent` structure; that is,

$$\Sigma = \text{Var} \begin{bmatrix} u_{0i} \\ u_{1i} \end{bmatrix} = \begin{bmatrix} \sigma_{u0}^2 & 0 \\ 0 & \sigma_{u1}^2 \end{bmatrix} \quad (5)$$

with $\hat{\sigma}_{u0} = 2.63$ and $\hat{\sigma}_{u1} = 0.61$. Our point estimates of the fixed affects are essentially identical to those from model (3), 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 chibar2(01) =    292.77
(Assumption: randint nested in randslope) Prob > chibar2 =    0.0000
Note: LR tests based on REML are valid only when the fixed-effects
specification is identical for both models.
```

□ 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.” To compare models with different fixed-effects specifications, use a Wald test, or fit the models by ML (option `mle`).

In our example, the fixed-effects specifications for both models are identical ($\beta_0 + \beta_1\text{week}$), so our REML-based test is valid. □

The results thus favor the model that allows for a random pig-specific regression line over the model that allows only for a pig-specific shift.

Finally, we also note the message at the bottom of our `xtmixed` output informing us that our overall LR test comparing to linear regression is conservative. For an explanation, see *Distribution theory for likelihood-ratio tests* later in this entry. ◀

Covariance structures

In example 2, we fitted a model with the default `Independent` covariance given in (5). Within any random-effects level specification, we can override this default by specifying an alternative covariance structure via option `covariance()`.

▷ Example 3

We generalize (5) to allow u_{0i} and u_{1i} to be correlated; that is,

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

```
. xtmixed weight week || id: week, covariance(unstructured) variance
(output omitted)
Mixed-effects 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)      =   4552.31
Log restricted-likelihood = -870.43562 Prob > chi2       =    0.0000
```

weight	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]
week	6.209896	.0920382	67.47	0.000	6.029504 6.390287
_cons	19.35561	.4038677	47.93	0.000	18.56405 20.14718

Random-effects Parameters	Estimate	Std. Err.	[95% Conf. Interval]
id: Unstructured			
var(week)	.3799957	.0839023	.2465103 .5857635
var(_cons)	6.986465	1.616357	4.439432 10.99481
cov(week,_cons)	-.1033632	.2627309	-.6183063 .41158
var(Residual)	1.596829	.1231981	1.372736 1.857506

```
LR test vs. linear regression:      chi2(3) = 766.07 Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

But we don't find the correlation to be at all significant.

```
. lrtest . randslope
Likelihood-ratio test      LR chi2(1) = 0.16
(Assumption: randslope nested in .) Prob > chi2 = 0.6908
Note: LR tests based on REML are valid only when the fixed-effects
specification is identical for both models.
```

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

◀

We could instead have also specified `covariance(identity)`, restricting u_{0i} and u_{1i} 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 restricted maximum likelihood (REML) to estimate variance components. We could have just as easily asked for ML estimates. Refitting the model in example 2 by ML, we get

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

(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)    =    4689.52
Log likelihood = -869.03825      Prob > chi2        =     0.0000
```

weight	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
week	6.209896	.0906818	68.48	0.000	6.032163	6.387629
_cons	19.35561	.3979158	48.64	0.000	18.57571	20.13551

Random-effects Parameters	Estimate	Std. Err.	[95% Conf. Interval]	
id: Independent				
sd(week)	.6066848	.0660293	.4901415	.7509392
sd(_cons)	2.599299	.2969071	2.077912	3.251513
sd(Residual)	1.264441	.0487958	1.172331	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.

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 β . The restricted likelihood is then formed by considering the distribution of the linear contrasts. Not only does this make the maximization problem free of β , it also incorporates the degrees of freedom used to estimate β into the estimation of the variance components. This follows since, 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 σ^2 can be derived theoretically as the n -divided sample variance. The REML estimate can be derived by considering the first $n - 1$ error contrasts, $y_i - \bar{y}$, whose joint distribution is free of μ . Applying maximum likelihood to this distribution results in an estimate of σ^2 , that is, the $(n - 1)$ divided sample variance, which is unbiased for σ^2 .

The unbiasedness property of REML extends to all mixed models when the data are balanced, and thus REML would seem the clear choice in balanced-data problems, although in large samples the difference between ML and REML is negligible. One disadvantage of REML is that 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. The question of which method to use thus remains a matter of personal taste.

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

Two-level models

The panel-data representation of the mixed model given in (2) can be extended to two nested levels. Formally

$$\mathbf{y}_{ij} = \mathbf{X}_{ij}\boldsymbol{\beta} + \mathbf{Z}_{ij}^{(1)}\mathbf{u}_i^{(1)} + \mathbf{Z}_{ij}^{(2)}\mathbf{u}_{ij}^{(2)} + \boldsymbol{\epsilon}_{ij} \quad (6)$$

for $i = 1, \dots, M$ first-level groups and $j = 1, \dots, M_i$ second-level groups that are nested within group i . Group i, j consists of n_{ij} observations, so \mathbf{y}_{ij} , \mathbf{X}_{ij} , and $\boldsymbol{\epsilon}_{ij}$ each have row dimension n_{ij} . $\mathbf{Z}_{ij}^{(1)}$ is the $n_{ij} \times q_1$ design matrix for the first-level random effects $\mathbf{u}_i^{(1)}$, and $\mathbf{Z}_{ij}^{(2)}$ is the $n_{ij} \times q_2$ design matrix for the second-level random effects $\mathbf{u}_{ij}^{(2)}$. Furthermore, assume that

$$\mathbf{u}_i^{(1)} \sim N(\mathbf{0}, \boldsymbol{\Sigma}_1); \quad \mathbf{u}_{ij}^{(2)} \sim N(\mathbf{0}, \boldsymbol{\Sigma}_2); \quad \boldsymbol{\epsilon}_{ij} \sim N(\mathbf{0}, \sigma_\epsilon^2 \mathbf{I})$$

and that $\mathbf{u}_i^{(1)}$, $\mathbf{u}_{ij}^{(2)}$, and $\boldsymbol{\epsilon}_{ij}$ are independent.

Fitting a two-level model requires you to specify two random-effects “equations”, one for each level. The variable list for the first equation represents $\mathbf{Z}_{ij}^{(1)}$, and the second equation, $\mathbf{Z}_{ij}^{(2)}$.

► Example 4

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

```
. use http://www.stata-press.com/data/r10/productivity, clear
(Public Capital Productivity)
. describe
Contains data from productivity.dta
  obs:      816                Public Capital Productivity
  vars:      11                29 Mar 2007 10:57
  size:     32,640 (99.7% of memory free)  (_dta has notes)
```

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

Sorted by:

Since the states are nested within regions, we fit a two-level mixed model with random intercepts at both the region and at the state-within-region levels. That is, we use (6) with both $\mathbf{Z}_{ij}^{(1)}$ and $\mathbf{Z}_{ij}^{(2)}$ set to the $n_{ij} \times 1$ column of ones, and $\boldsymbol{\Sigma}_1 = \sigma_1^2$ and $\boldsymbol{\Sigma}_2 = \sigma_2^2$ are both scalars.

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

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

```
Log restricted-likelihood = 1404.7101          Wald chi2(6)      = 18382.39
                                                Prob > chi2       = 0.0000
```

gsp	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
private	.2660308	.0215471	12.35	0.000	.2237993	.3082624
emp	.7555059	.0264556	28.56	0.000	.7036539	.8073579
hwy	.0718857	.0233478	3.08	0.002	.0261249	.1176464
water	.0761552	.0139952	5.44	0.000	.0487251	.1035853
other	-.1005396	.0170173	-5.91	0.000	-.1338929	-.0671862
unemp	-.0058815	.0009093	-6.47	0.000	-.0076636	-.0040994
_cons	2.126995	.1574864	13.51	0.000	1.818327	2.435663

Random-effects Parameters		Estimate	Std. Err.	[95% Conf. Interval]	
region: Identity					
	sd(_cons)	.0435471	.0186292	.0188287	.1007161
state: Identity					
	sd(_cons)	.0802737	.0095512	.0635762	.1013567
	sd(Residual)	.0368008	.0009442	.034996	.0386987

```
LR test vs. linear regression:          chi2(2) = 1162.40   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, and the second is a random intercept at the `state` level. 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 model level. You can suppress this table with option `nogroup` or with `noheader`, 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 (6) extends in a straightforward manner to more than two nested levels of random effects, 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 (i.e., **Independent**), exchangeable, or as general symmetric (**Unstructured**). These may also be combined to produce more complex blocked-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 REML regression                Number of obs    =      816
                                             Wald chi2(6)       =  16803.51
Log restricted-likelihood = 1423.3455        Prob > chi2        =    0.0000
```

Random-effects Parameters	Estimate	Std. Err.	[95% Conf. Interval]	
region: Independent				
sd(hwy)	.0052752	.0108846	.0000925	.3009897
sd(unemp)	.0052895	.001545	.002984	.0093764
sd(_cons)	.0596008	.0758296	.0049235	.721487
state: Identity				
sd(_cons)	.0807543	.009887	.0635259	.1026551
sd(Residual)	.0353932	.000914	.0336464	.0372307

```
LR test vs. linear regression:          chi2(4) = 1199.67   Prob > chi2 = 0.0000
```

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

```
. estimates store prodc
```

This model is the same as that fitted in example 4, except that $\mathbf{Z}_{ij}^{(1)}$ is now the $n_{ij} \times 3$ matrix with columns determined by the values of `hwy`, `unemp`, and an intercept term (`one`), in that order, and (since we used the default Independent structure) Σ_1 is

$$\Sigma_1 = \begin{pmatrix} & \text{hwy} & \text{unemp} & \text{_cons} \\ \begin{pmatrix} \sigma_a^2 & 0 & 0 \\ 0 & \sigma_b^2 & 0 \\ 0 & 0 & \sigma_c^2 \end{pmatrix} \end{pmatrix}$$

The random-effects specification at the state level remains unchanged; i.e., Σ_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

$$\Sigma_1 = \begin{pmatrix} & \text{hwy} & \text{unemp} & \text{_cons} \\ \begin{pmatrix} \sigma_a^2 & 0 & 0 \\ 0 & \sigma_a^2 & 0 \\ 0 & 0 & \sigma_c^2 \end{pmatrix} \end{pmatrix}$$

looks plausible. We can impose this equality constraint by treating Σ_1 as blocked diagonal: the first block is a 2×2 multiple of the identity matrix, i.e., $\sigma_a^2 \mathbf{I}_2$; the second is a scalar, equivalently a 1×1 multiple of the identity.

We construct blocked-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 REML regression                Number of obs      =      816
                                           Wald chi2(6)          = 16803.41
Log restricted-likelihood = 1423.3455        Prob > chi2           =    0.0000
```

Random-effects Parameters	Estimate	Std. Err.	[95% Conf. Interval]	
region: Identity sd(hwy unemp)	.0052896	.0015446	.0029844	.0093752
region: Identity sd(_cons)	.0595029	.0318238	.0208589	.1697401
state: Identity sd(_cons)	.080752	.0097453	.0637425	.1023006
sd(Residual)	.0353932	.0009139	.0336465	.0372306

```
LR test vs. linear regression:      chi2(3) = 1199.67  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 σ_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 . prodc
Likelihood-ratio test          LR chibar2(01)  =      0.00
(Assumption: . nested in prodc) Prob > chibar2   =      1.0000
Note: LR tests based on REML are valid only when the fixed-effects
specification is identical for both models.
```

◀

You can repeat level specifications as often as you like, defining successive blocks of a blocked-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 option `noconstant`. When you specify repeated level equations, `xtmixed` knows not to put constant terms in each equation since 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.

□

Factor notation and crossed-effects models

Not all mixed models contain nested levels of random effects.

▷ Example 6

Returning to our longitudinal analysis of pig weights, suppose that instead of (4) we wish to fit

$$\text{weight}_{ij} = \beta_0 + \beta_1 \text{week}_{ij} + u_i + v_j + \epsilon_{ij} \quad (7)$$

for the $i = 1, \dots, 48$ pigs and $j = 1, \dots, 9$ weeks 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 (4) and (7) 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 (4), we assume that the effect due to `week` is linear and pig specific (a random slope); in (7), we assume that the effect due to `week`, v_j , is systematic to that week and common to all pigs. The rationale behind (7) 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 (7) is an example of a two-way *crossed-effects* model, with the pig effects u_i being crossed with the week effects v_j . One way to fit such models is to consider all the data as one big panel and treat the u_i and v_j as a series of $48 + 9 = 57$ random coefficients on indicator variables for `pig` and `week`. In the notation of (1),

$$\mathbf{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_{48} \\ v_1 \\ \vdots \\ v_9 \end{bmatrix} \sim N(\mathbf{0}, \mathbf{G}); \quad \mathbf{G} = \begin{bmatrix} \sigma_u^2 \mathbf{I}_{48} & \mathbf{0} \\ \mathbf{0} & \sigma_v^2 \mathbf{I}_9 \end{bmatrix}$$

Since \mathbf{G} is blocked 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 `pig` and `week` as factor variables (or equivalently, as two sets of overparameterized indicator variables identifying pigs and weeks, 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/r10/pig, clear
(Longitudinal analysis of pig weights)
. xtmixed weight week || _all: R.id || _all: R.week
Performing EM optimization:
Performing gradient-based optimization:
Iteration 0:  log restricted-likelihood = -1015.4214
Iteration 1:  log restricted-likelihood = -1015.4214
Computing standard errors:
Mixed-effects REML regression                Number of obs      =      432
Group variable: _all                         Number of groups   =        1
                                             Obs per group: min =      432
                                             avg =              432.0
                                             max =              432
Log restricted-likelihood = -1015.4214        Wald chi2(1)       = 11516.16
                                             Prob > chi2        =   0.0000
```

weight	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
week	6.209896	.0578669	107.31	0.000	6.096479	6.323313
_cons	19.35561	.6493996	29.81	0.000	18.08281	20.62841

Random-effects Parameters		Estimate	Std. Err.	[95% Conf. Interval]	
_all: Identity	sd(R.id)	3.892648	.4141707	3.15994	4.795252
_all: Identity	sd(R.week)	.3337581	.1611824	.1295268	.8600111
	sd(Residual)	2.072917	.0755915	1.929931	2.226496

```
LR test vs. linear regression:      chi2(2) =   476.10  Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
. estimates store crossed
```

and thus we estimate $\hat{\sigma}_u = 3.89$ and $\hat{\sigma}_v = 0.33$. Both (4) and (7) 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 (4) and looking at the AIC values using `estimates stats`

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

Model	Obs	ll(null)	ll(model)	df	AIC	BIC
crossed	432	.	-1015.421	5	2040.843	2061.185
.	432	.	-870.5147	5	1751.029	1771.372

definitely favors model (4). This finding is not surprising, given that our rationale behind (7) 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 we can use `xtmixed` to fit a constant-only model directly.

◀

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. Since the set of indicators is overparameterized, *R.varname* implies `noconstant`. To include indicator variables in the fixed-effects specification, use `xi`; see [R] `xi`.

□ Technical Note

Although we were able to fit the crossed-effects model (7), it came at the expense of increasing the column dimension of our random-effects design from two in model (4) to 57 in model (7). 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 (7), we note that if we drop v_j , we end up with a model equivalent to (3), meaning that we could have fitted (3) by typing

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

instead of

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

as we did when we originally fitted the model. The results of both estimations are identical, but the latter specification, organized at the panel (`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 (7) by using

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

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 (7)

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

We leave it to you to verify that both produce identical results.

□

▷ **Example 7**

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

$$\mathbf{y}_{ij} = \mathbf{X}_{ij}\boldsymbol{\beta} + u_i^{(1)} + u_{ij}^{(2)} + \boldsymbol{\epsilon}_{ij}$$

where \mathbf{y}_{ij} represents the logarithms of gross state products for the $n_{ij} = 17$ observations from state j in region i , \mathbf{X}_{ij} is a set of regressors, $u_i^{(1)}$ is a random intercept at the region level, and $u_{ij}^{(2)}$ is a random intercept at the state (nested within region) level. We assume that $u_i^{(1)} \sim N(0, \sigma_1^2)$ and $u_{ij}^{(2)} \sim N(0, \sigma_2^2)$ independently. Define

$$\mathbf{v}_i = \begin{bmatrix} u_i^{(1)} + u_{i1}^{(2)} \\ u_i^{(1)} + u_{i2}^{(2)} \\ \vdots \\ u_i^{(1)} + u_{iM_i}^{(2)} \end{bmatrix}$$

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

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

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

$$\mathbf{Z}_i = \mathbf{I}_{M_i} \otimes \mathbf{J}_{17}$$

for a k -column vector of ones \mathbf{J}_k , and

$$\boldsymbol{\Sigma} = \text{Var}(\mathbf{v}_i) = \begin{bmatrix} \sigma_1^2 + \sigma_2^2 & \sigma_1^2 & \cdots & \sigma_1^2 \\ \sigma_1^2 & \sigma_1^2 + \sigma_2^2 & \cdots & \sigma_1^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_1^2 & \sigma_1^2 & \sigma_1^2 & \sigma_1^2 + \sigma_2^2 \end{bmatrix}_{M_i \times M_i}$$

Since $\boldsymbol{\Sigma}$ 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/r10/productivity, clear
(Public Capital Productivity)
```

```
. xtmixed gsp private emp hwy water other unemp || region: R.state,
> cov(exchangeable) variance
(output omitted)
Mixed-effects REML 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)      =    18382.39
Log restricted-likelihood = 1404.7101   Prob > chi2       =      0.0000
```

gsp	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]
private	.2660308	.0215471	12.35	0.000	.2237993 .3082623
emp	.7555059	.0264556	28.56	0.000	.7036539 .8073579
hwy	.0718857	.0233478	3.08	0.002	.0261249 .1176464
water	.0761552	.0139952	5.44	0.000	.0487251 .1035853
other	-.1005396	.0170173	-5.91	0.000	-.1338929 -.0671862
unemp	-.0058815	.0009093	-6.47	0.000	-.0076636 -.0040994
_cons	2.126995	.1574864	13.51	0.000	1.818327 2.435663

Random-effects Parameters	Estimate	Std. Err.	[95% Conf. Interval]
region: Exchangeable			
var(R.state)	.0083402	.0020718	.0051254 .0135715
cov(R.state)	.0018963	.0016225	-.0012836 .0050763
var(Residual)	.0013543	.0000695	.0012247 .0014976

LR test vs. linear regression: chi2(2) = 1162.40 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 $(\sigma_1^2 + \sigma_2^2, \sigma_1^2, \sigma_\epsilon^2)$, as displayed here, to $(\sigma_1, \sigma_2, \sigma_\epsilon)$, 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 two-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;
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 two-level nested random intercept model

$$y_{ij} = \mathbf{x}_{ij}\boldsymbol{\beta} + u_i^{(1)} + u_{ij}^{(2)} + \epsilon_{ij}$$

without any replicated measurements at the (i, j) level. This model is unidentified for such data since the random intercepts $u_{ij}^{(2)}$ are confounded with the overall errors ϵ_{ij} ; 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.

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 option `emonly` 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 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 option `difficult` can be particularly helpful if you are seeing many “nonconcave” messages; you may also consider changing the `technique()` or using option `nonrtolerance`; see [R] **maximize**.

Distribution theory for likelihood-ratio tests

When determining the asymptotic distribution of a likelihood-ratio (LR) test comparing two nested models fitted by `xtmixed`, issues concerning boundary problems imposed by estimating strictly positive quantities (i.e., 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 `lrtest`), you may thus sometimes see a test labeled as “`chibar`” rather than the usual “`chi2`” or see a `chi2` test with a note attached stating that the test is conservative.

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 χ^2 with degrees of freedom equal to the difference in the number of estimated parameters between both models.

When there is only one variance being set to zero in the reduced model, the asymptotic distribution of the likelihood-ratio test statistic is a 50:50 mixture of a χ_k^2 and a χ_{k+1}^2 distribution, where k 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

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

Since the random component of the mixed model comprises three parameters $(\sigma_0^2, \sigma_{01}, \sigma_1^2)$, on the surface it would seem that the LR comparison test would be distributed as χ_3^2 . However, two complications need to be considered. First, the variances σ_0^2 and σ_1^2 are restricted to be positive, and second, constraints such as $\sigma_1^2 = 0$ implicitly restrict the covariance σ_{01} to be 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 (e.g., Stram and Lee 1994) and empirical studies (e.g., McLachlan and Basford 1988) have demonstrated that, whatever the distribution of the LR test statistic, its tail probabilities are bounded above by those of the χ^2 distribution with degrees of freedom equal to the full number of restricted parameters (three in the above example).

`xtmixed` uses this reference distribution, the χ^2 with full degrees of freedom, to produce a conservative test and places a note in the output labeling the test as such. Since 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 multi-equation 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.

□

Saved Results

`xtmixed` saves the following in `e()`:

Scalars

<code>e(N)</code>	number of observations	<code>e(chi2)</code>	χ^2
<code>e(k)</code>	number of parameters	<code>e(p)</code>	p-value for χ^2
<code>e(k_f)</code>	number of FE parameters	<code>e(ll_c)</code>	log-likelihood, comparison model
<code>e(k_r)</code>	number of RE parameters	<code>e(chi2_c)</code>	χ^2 , comparison model
<code>e(k_rs)</code>	number of std. deviations	<code>e(df_c)</code>	degrees of freedom, comparison model
<code>e(k_rc)</code>	number of correlations	<code>e(p_c)</code>	p-value, comparison model
<code>e(df_m)</code>	model degrees of freedom	<code>e(converged)</code>	1 if converged, 0 otherwise
<code>e(ll)</code>	log (restricted)-likelihood	<code>e(rc)</code>	return code

Macros

<code>e(cmd)</code>	<code>xtmixed</code>	<code>e(chi2type)</code>	Wald, type of model χ^2
<code>e(cmdline)</code>	command as typed	<code>e(opt)</code>	type of optimization
<code>e(title)</code>	title in estimation output	<code>e(ml_method)</code>	type of ml method
<code>e(depvar)</code>	name of dependent variable	<code>e(technique)</code>	maximization technique
<code>e(method)</code>	ML or REML	<code>e(crittype)</code>	optimization criterion
<code>e(ivars)</code>	grouping variables	<code>e(properties)</code>	b V
<code>e(redim)</code>	random-effects dimensions	<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(vartypes)</code>	variance-structure types	<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(revars)</code>	random-effects covariates		

Matrices

<code>e(b)</code>	coefficient vector	<code>e(V)</code>	variance–covariance matrix of the estimator
<code>e(N_g)</code>	group counts	<code>e(g_avg)</code>	group size averages
<code>e(g_min)</code>	group size minimums		
<code>e(g_max)</code>	group size maximums		

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

Methods and Formulas

`xtmixed` is implemented as an ado-file.

As given by (1), 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 a $n \times p$ 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 $\sigma_\epsilon^2 \mathbf{I}_n$. 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

$$L(\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma_\epsilon^2) = -\frac{1}{2} \{n \log(2\pi) + \log |\mathbf{V}| + (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})' \mathbf{V}^{-1} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\} \quad (8)$$

which is maximized as a function of $\boldsymbol{\beta}$, $\boldsymbol{\theta}$, and σ_ϵ^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

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

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 panels (and subpanels if possible) and using matrix decomposition methods on the smaller matrices that result from treating each panel one at time.

Consider the one-level model described previously in (2)

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

for $i = 1, \dots, M$ panels with panel i containing n_i observations, with $\text{Var}(\mathbf{u}_i) = \boldsymbol{\Sigma}$, a $q \times q$ matrix.

Efficient methods for computing (8) and (9) are given in chapter 2 of Pinheiro and Bates (2000). Namely, for the one-level model, define $\boldsymbol{\Delta}$ to be the Cholesky factor of $\sigma_\epsilon^2 \boldsymbol{\Sigma}^{-1}$, such that $\sigma_\epsilon^2 \boldsymbol{\Sigma}^{-1} = \boldsymbol{\Delta}' \boldsymbol{\Delta}$. For $i = 1, \dots, M$, decompose

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

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

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

stack the \mathbf{R}_{00i} and \mathbf{c}_{0i} matrices, and perform the additional QR decomposition

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

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

$$L(\boldsymbol{\Delta}) = \frac{n}{2} \{\log n - \log(2\pi) - 1\} - n \log \|\mathbf{c}_1\| + \sum_{i=1}^M \log \left| \frac{\det(\boldsymbol{\Delta})}{\det(\mathbf{R}_{11i})} \right| \quad (10)$$

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

$$\widehat{\boldsymbol{\beta}} = \mathbf{R}_{00}^{-1} \mathbf{c}_0; \quad \widehat{\sigma}_\epsilon^2 = n^{-1} \|\mathbf{c}_1\|^2 \quad (11)$$

REML estimates are obtained by maximizing

$$L_R(\mathbf{\Delta}) = \frac{n-p}{2} \{ \log(n-p) - \log(2\pi) - 1 \} - (n-p) \log \|\mathbf{c}_1\| \\ - \log |\det(\mathbf{R}_{00})| + \sum_{i=1}^M \log \left| \frac{\det(\mathbf{\Delta})}{\det(\mathbf{R}_{11i})} \right| \quad (12)$$

followed by

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

For numerical stability, maximization of (10) and (12) is not performed with respect to the unique elements of $\mathbf{\Delta}$ but instead with respect to the unique elements of the matrix logarithm of $\mathbf{\Sigma}/\sigma_\epsilon^2$; define $\boldsymbol{\gamma}$ to be the vector containing these elements.

Once maximization with respect to $\boldsymbol{\gamma}$ is completed, $(\boldsymbol{\gamma}, \sigma_\epsilon^2)$ is reparameterized to $\{\boldsymbol{\alpha}, \log(\sigma_\epsilon)\}$, where $\boldsymbol{\alpha}$ is a vector containing the unique elements of $\mathbf{\Sigma}$, 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 σ_ϵ^2 ; (b) obtain a parameterization under which parameter estimates can be interpreted individually, rather than as elements of a matrix 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

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

with the analogous expression for REML.

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

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

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

Parameter estimates are stored in `e(b)` as $\{\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\alpha}}, \log(\hat{\sigma}_\epsilon)\}$, with the corresponding (blocked diagonal) variance–covariance matrix stored in `e(V)`. Parameter estimates can be displayed in this metric by specifying option `estmetric`. 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}_i 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

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

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

1. For the current iterated value of $\boldsymbol{\Sigma}^{(t)}$, fix $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}(\boldsymbol{\Sigma}^{(t)})$ and $\hat{\sigma}_\epsilon^2 = \hat{\sigma}_\epsilon^2(\boldsymbol{\Sigma}^{(t)})$ according to (11).
2. E-step: calculate

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

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

3. M-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 extensions to two or more nested levels of random effects, see Bates and Pinheiro (1998).

Acknowledgments

We thank Badi Baltagi, Department of Economics, Syracuse University, and Ray Carroll, Department of Statistics, Texas A&M University, for providing us with the datasets 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 models. *Technical Memorandum BL0112140-980226-01TM*. Murray Hill, NJ: Bell Labs, Lucent Technologies.
- 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: 1–22.
- Diggle, P. J., P. Heagerty, K.-Y. Liang, and S. L. Zeger. 2002. *Analysis of Longitudinal Data*. 2nd ed. Oxford: Oxford University Press.
- Gutierrez, R. G., S. L. Carter, and D. M. Drukker. 2001. sg160: On boundary-value likelihood-ratio tests. *Stata Technical Bulletin* 60: 15–18. Reprinted in *Stata Technical Bulletin Reprints*, vol. 10, pp. 269–273.
- Harville, D. A. 1977. Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association* 72: 320–340.
- 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.
- 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. 2006. Estimating variance components in Stata. *Stata Journal* 6: 1–21.

- McCulloch, C. E., and S. R. Searle. 2001. *Generalized, Linear, and Mixed Models*. New York: 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.
- Pinheiro, J. C., and D. M. Bates. 2000. *Mixed-Effects Models in S and S-PLUS*. New York: Springer.
- Rabe-Hesketh, S., and A. Skrondal. 2006. *Multilevel and Longitudinal Modeling Using Stata*. 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. 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 Press.
- Stram, D. O., and J. W. Lee. 1994. Variance components testing in the longitudinal mixed effects model. *Biometrics* 50: 1171–1177.
- Thompson, W. A. 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.

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