

mixed postestimation — Postestimation tools for mixed

Postestimation commands	predict	margins	estat
test and testparm	lincom	contrast	pwcompare
Remarks and examples	Stored results	Methods and formulas	References
Also see			

Postestimation commands

The following postestimation commands are of special interest after `mixed`:

Command	Description
estat df	calculate and display degrees of freedom for fixed effects
estat group	summarize the composition of the nested groups
estat icc	estimate intraclass correlations
estat recovariance	display the estimated random-effects covariance matrix (or matrices)
estat wcorrelation	display model-implied within-cluster correlations and standard deviations

The following standard postestimation commands are also available:

Command	Description
contrast	contrasts and ANOVA-style joint tests of estimates
estat ic	Akaike's and Schwarz's Bayesian information criteria (AIC and BIC)
estat summarize	summary statistics for the estimation sample
estat vce	variance-covariance matrix of the estimators (VCE)
estimates	cataloging estimation results
hausman	Hausman's specification test
lincom	point estimates, standard errors, testing, and inference for linear combinations of coefficients
lrtest	likelihood-ratio test
margins	marginal means, predictive margins, marginal effects, and average marginal effects
marginsplot	graph the results from margins (profile plots, interaction plots, etc.)
nlcom	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
predict	predictions, residuals, influence statistics, and other diagnostic measures
predictnl	point estimates, standard errors, testing, and inference for generalized predictions
pwcompare	pairwise comparisons of estimates
test	Wald tests of simple and composite linear hypotheses
testnl	Wald tests of nonlinear hypotheses

predict

Description for predict

`predict` creates a new variable containing predictions such as linear predictions, standard errors, fitted values, residuals, and standardized residuals.

Menu for predict

Statistics > Postestimation

Syntax for predict

Syntax for obtaining BLUPs of random effects and the BLUPs' standard errors

```
predict [type] newvarsspec [if] [in], reffects [reses(newvarsspec)
relevel(levelvar) ]
```

Syntax for obtaining scores after ML estimation

```
predict [type] newvarsspec [if] [in], scores
```

Syntax for obtaining other predictions

```
predict [type] newvar [if] [in] [, statistic relevel(levelvar) ]
```

newvarsspec is *stub** or *newvarlist*.

<i>statistic</i>	Description
Main	
<code>xb</code>	linear prediction for the fixed portion of the model only; the default
<code>stdp</code>	standard error of the fixed-portion linear prediction
<code><u>f</u>itted</code>	fitted values, fixed-portion linear prediction plus contributions based on predicted random effects
<code><u>r</u>esiduals</code>	residuals, response minus fitted values
* <code><u>r</u>standard</code>	standardized residuals

Unstarred statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample. Starred statistics are calculated only for the estimation sample, even when `if e(sample)` is not specified.

Options for predict

Main

xb, the default, calculates the linear prediction $\mathbf{x}\beta$ based on the estimated fixed effects (coefficients) in the model. This is equivalent to fixing all random effects in the model to their theoretical mean value of 0.

stdp calculates the standard error of the linear predictor $\mathbf{x}\beta$.

reflects calculates best linear unbiased predictions (BLUPs) of the random effects. By default, BLUPs for all random effects in the model are calculated. However, if the **relevel**(*levelvar*) option is specified, then BLUPs for only level *levelvar* in the model are calculated. For example, if **classes** are nested within **schools**, then typing

```
. predict b*, reflects relevel(school)
```

would produce BLUPs at the school level. You must specify q new variables, where q is the number of random-effects terms in the model (or level). However, it is much easier to just specify *stub** and let Stata name the variables *stub1*, *stub2*, ..., *stubq* for you.

Rabe-Hesketh and Skrondal (2012, sec. 2.11.2) discuss the link between the empirical Bayes predictions and BLUPs and how these predictions are unbiased. They are unbiased when the groups associated with the random effects are expected to vary in repeated samples. If you expect the groups to be fixed in repeated samples, then these predictions are no longer unbiased.

reses(*newvarspec*) calculates the standard errors of the BLUPs of the random effects. By default, standard errors for all BLUPs in the model are calculated. However, if the **relevel**(*levelvar*) option is specified, then standard errors for only level *levelvar* in the model are calculated; see the **reflects** option.

You must specify q new variables, where q is the number of random-effects terms in the model (or level). However, it is much easier to just specify *stub** and let Stata name the variables *stub1*, *stub2*, ..., *stubq* for you. The new variables will have the same storage type as the corresponding random-effects variables.

The **reflects** and **reses**() options often generate multiple new variables at once. When this occurs, the random effects (or standard errors) contained in the generated variables correspond to the order in which the variance components are listed in the output of **mixed**. Still, examining the variable labels of the generated variables (with the **describe** command, for instance) can be useful in deciphering which variables correspond to which terms in the model.

fitted calculates fitted values, which are equal to the fixed-portion linear predictor *plus* contributions based on predicted random effects, or in mixed-model notation, $\mathbf{x}\beta + \mathbf{Z}\mathbf{u}$. By default, the fitted values take into account random effects from all levels in the model; however, if the **relevel**(*levelvar*) option is specified, then the fitted values are fit beginning with the topmost level down to and including level *levelvar*. For example, if **classes** are nested within **schools**, then typing

```
. predict yhat_school, fitted relevel(school)
```

would produce school-level predictions. That is, the predictions would incorporate school-specific random effects but not those for each class nested within each school.

residuals calculates residuals, equal to the responses minus fitted values. By default, the fitted values take into account random effects from all levels in the model; however, if the **relevel**(*levelvar*) option is specified, then the fitted values are fit beginning at the topmost level down to and including level *levelvar*.

rstandard calculates standardized residuals, equal to the residuals multiplied by the inverse square root of the estimated error covariance matrix.

`scores` calculates the parameter-level scores, one for each parameter in the model including regression coefficients and variance components. The score for a parameter is the first derivative of the log likelihood (or log pseudolikelihood) with respect to that parameter. One score per highest-level group is calculated, and it is placed on the last record within that group. Scores are calculated in the estimation metric as stored in `e(b)`.

`scores` is not available after restricted maximum-likelihood (REML) estimation.

`relevel(levelvar)` specifies the level in the model at which predictions involving random effects are to be obtained; see the options above for the specifics. *levelvar* is the name of the model level and is either the name of the variable describing the grouping at that level or is `_all`, a special designation for a group comprising all the estimation data.

margins

Description for margins

`margins` estimates margins of response for linear predictions.

Menu for margins

Statistics > Postestimation

Syntax for margins

```
margins [marginlist] [, options]
```

```
margins [marginlist] , predict(statistic ...) [options]
```

<i>statistic</i>	Description
<code>xb</code>	linear predictor for the fixed portion of the model only; the default
<code><u>r</u>effects</code>	not allowed with <code>margins</code>
<code><u>s</u>cores</code>	not allowed with <code>margins</code>
<code><u>s</u>tdp</code>	not allowed with <code>margins</code>
<code><u>f</u>itted</code>	not allowed with <code>margins</code>
<code><u>r</u>esiduals</code>	not allowed with <code>margins</code>
<code><u>s</u>tandard</code>	not allowed with <code>margins</code>

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.

For the full syntax, see [R] [margins](#).

estat

Description for estat

`estat group` reports the number of groups and minimum, average, and maximum group sizes for each level of the model. Model levels are identified by the corresponding group variable in the data. Because groups are treated as nested, the information in this summary may differ from what you would get if you used the `tabulate` command on each group variable individually.

`estat icc` displays the intraclass correlation for pairs of responses at each nested level of the model. Intraclass correlations are available for random-intercept models or for random-coefficient models conditional on random-effects covariates being equal to 0. They are not available for crossed-effects models or with residual error structures other than independent structures.

`estat recovariance` displays the estimated variance–covariance matrix of the random effects for each level in the model. Random effects can be either random intercepts, in which case the corresponding rows and columns of the matrix are labeled as `_cons`, or random coefficients, in which case the label is the name of the associated variable in the data.

`estat wcorrelation` displays the overall correlation matrix for a given cluster calculated on the basis of the design of the random effects and their assumed covariance and the correlation structure of the residuals. This allows for a comparison of different multilevel models in terms of the ultimate within-cluster correlation matrix that each model implies.

`estat df` calculates and displays the degrees of freedom (DF) for each fixed effect using the specified methods. This allows for a comparison of different DF methods. `estat df` can also be used to continue with postestimation using a different DF method without rerunning the mixed model, which is useful if one wants to change the current DF method after comparing methods.

Menu for estat

Statistics > Postestimation

Syntax for estat

Summarize the composition of the nested groups

```
estat group
```

Estimate intraclass correlations

```
estat icc [ , level(#) ]
```

Display the estimated random-effects covariance matrix (or matrices)

```
estat recovariance [ , relevel(levelvar) correlation matlist_options ]
```

Display model-implied within-cluster correlations and standard deviations

```
estat wcorrelation [ , wcor_options ]
```

Report or calculate degrees of freedom for fixed effects

```
estat df [ , method(df_methods) post[ (df_method) ] eim oim ]
```

<i>wcor_options</i>	Description
<u>at</u> (<i>at_spec</i>)	specify the cluster for which you want the correlation matrix; default is the first two-level cluster encountered in the data
<u>all</u>	display correlation matrix for all the data
<u>covariance</u>	display the covariance matrix instead of the correlation matrix
<u>list</u>	list the data corresponding to the correlation matrix
<u>nosort</u>	list the rows and columns of the correlation matrix in the order they were originally present in the data
<u>format</u> (<i>%fmt</i>)	set the display format; default is <code>format(%6.3f)</code>
<i>matlist_options</i>	style and formatting options that control how matrices are displayed

Option for estat icc

level(#) specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`; see [U] 20.7 **Specifying the width of confidence intervals**.

Options for estat recovariance

relevel(*levelvar*) specifies the level in the model for which the random-effects covariance matrix is to be displayed. By default, the covariance matrices for all levels in the model are displayed. *levelvar* is the name of the model level and is either the name of the variable describing the grouping at that level or is `_all`, a special designation for a group comprising all the estimation data.

correlation displays the covariance matrix as a correlation matrix.

matlist_options are style and formatting options that control how the matrix (or matrices) is displayed; see [P] **matlist** for a list of options that are available.

Options for estat wcorrelation

`at(at_spec)` specifies the cluster of observations for which you want the within-cluster correlation matrix. *at_spec* is

```
relevel_var = value [, relevel_var = value ...]
```

For example, if you specify

```
. estat wcorrelation, at(school = 33)
```

you get the within-cluster correlation matrix for those observations in school 33. If you specify

```
. estat wcorrelation, at(school = 33 classroom = 4)
```

you get the correlation matrix for classroom 4 in school 33.

If `at()` is not specified, then you get the correlations for the first level-two cluster encountered in the data. This is usually what you want.

`all` specifies that you want the correlation matrix for all the data. This is not recommended unless you have a relatively small dataset or you enjoy seeing large $N \times N$ matrices. However, this can prove useful in some cases.

`covariance` specifies that the within-cluster covariance matrix be displayed instead of the default correlations and standard deviations.

`list` lists the model data for those observations depicted in the displayed correlation matrix. This option is useful if you have many random-effects design variables and you wish to see the represented values of these design variables.

`nosort` lists the rows and columns of the correlation matrix in the order that they were originally present in the data. Normally, `estat wcorrelation` will first sort the data according to level variables, by-group variables, and time variables to produce correlation matrices whose rows and columns follow a natural ordering. `nosort` suppresses this.

`format(%fmt)` sets the display format for the standard-deviation vector and correlation matrix. The default is `format(%6.3f)`.

matlist_options are style and formatting options that control how the matrix (or matrices) is displayed; see [P] [matlist](#) for a list of options that are available.

Options for estat df

`method(df_methods)` specifies a list of methods to compute DF. The supported methods are `residual`, `repeated`, `anova`, `satterthwaite`, and `kroger`; more than one method may be specified. Methods `satterthwaite` and `kroger` are only available with REML estimation. If option `dfmethod()` was not specified in the most recently fit mixed model, then option `method()` is required. See [Small-sample inference for fixed effects](#) under *Remarks and examples* in [ME] [mixed](#) for more details.

`post` causes `estat df` to behave like a Stata estimation command. When `post` is specified, `estat df` will post the DF for each fixed effect as well as everything related to the DF computation to `e()` for the method specified in `method()`. Thus, after posting, you could continue to use this DF for other postestimation commands. For example, you could use `test`, `small` to perform Wald F tests on linear combination of the fixed effects.

`post` may also be specified using the syntax `post(df_method)`. You must use this syntax if you specify multiple *df_methods* in option `method()`. With this syntax, `estat df` computes the DF

using the method specified in `post()` and stores the results in `e()`. Only one computation method may be specified using the syntax `post()`.

The `df_method` specified in `post()` must be one of the DF methods specified in option `method()`. If only one method is specified in option `method()`, then one can simply use `post` to make this DF method active for postestimation and for mixed replay.

`eim` specifies that the expected information matrix be used in the DF computation. It can be used only when `method()` contains `kroger` or `satterthwaite`. `eim` is the default and may not be specified with `oim`.

`oim` specifies that the observed information matrix be used in the DF computation. It can be used only when `method()` contains `kroger` or `satterthwaite` and may not be specified with `eim`.

test and testparm

Description for test and testparm

`test` and `testparm`, by default, perform χ^2 tests of simple and composite linear hypotheses about the parameters for the most recently fit `mixed` model. They also support F tests with a small-sample adjustment for fixed effects.

Menu for test and testparm

Statistics > Postestimation

Syntax for test and testparm

```
test (spec) [(spec) ...] [, test_options small]
```

```
testparm varlist [, testparm_options small]
```

Options for test and testparm

Options

`test_options`; see [R] [test](#) options. Options `df()`, `common`, and `nosvyadjust` may not be specified together with `small`.

`testparm_options`; see options of `testparm` in [R] [test](#). Options `df()` and `nosvyadjust` may not be specified together with `small`.

`small` specifies that F tests for fixed effects be carried out with the denominator degrees of freedom (DDF) obtained by the same method used in the most recently fit `mixed` model. If option `dfmethod()` is not specified in the previous `mixed` command, option `small` is not allowed. For certain methods, the DDF for some tests may not be available. See [Small-sample inference for fixed effects](#) in [ME] [mixed](#) for more details.

lincom

Description for lincom

`lincom`, by default, computes point estimates, standard errors, z statistics, p -values, and confidence intervals for linear combinations of coefficients after `mixed`. `lincom` also provides t statistics for linear combinations of the fixed effects, with the degrees of freedom calculated by the DF method specified in option `dfmethod()` of `mixed`.

Menu for lincom

Statistics > Postestimation

Syntax for lincom

```
lincom exp [ , lincom_options small ]
```

Options for lincom

lincom_options; see [R] [lincom](#) options. Option `df()` may not be specified together with `small`.

`small` specifies that t statistics for linear combinations of fixed effects be displayed with the degrees of freedom obtained by the same method used in the most recently fit `mixed` model. If option `dfmethod()` is not specified in the previous `mixed` command, option `small` is not allowed. For certain methods, the degrees of freedom for some linear combinations may not be available. See [Small-sample inference for fixed effects](#) in [ME] [mixed](#) for more details.

contrast

Description for contrast

`contrast`, by default, performs χ^2 tests of linear hypotheses and forms contrasts involving factor variables and their interactions for the most recently fit `mixed` model. `contrast` also supports tests with small-sample adjustments after `mixed`, `dfmethod()`.

Menu for contrast

Statistics > Postestimation

Syntax for contrast

```
contrast termlist [ , contrast_options small ]
```

Options for contrast

contrast_options; see [R] **contrast** options. Options `df()` and `nosvyadjust` may not be specified together with `small`.

`small` specifies that tests for contrasts be carried out with the DDF obtained by the same method used in the most recently fit `mixed` model. If option `dfmethod()` is not specified in the previous `mixed` command, option `small` is not allowed. For certain methods, the DDF for some contrasts may not be available. See *Small-sample inference for fixed effects* in [ME] **mixed** for more details.

pwcompare

Description for pwcompare

`pwcompare` performs pairwise comparisons across the levels of factor variables from the most recently fit `mixed` model. `pwcompare`, by default, reports the comparisons as contrasts (differences) of margins along with z tests or confidence intervals for the pairwise comparisons. `pwcompare` also supports t tests with small-sample adjustments after `mixed`, `dfmethod()`.

Menu for pwcompare

Statistics > Postestimation

Syntax for pwcompare

```
pwcompare marginlist [ , pwcompare_options small ]
```

Options for pwcompare

pwcompare_options; see [R] **pwcompare** options. Option `df()` may not be specified together with `small`.

`small` specifies that t tests for pairwise comparisons be carried out with the degrees of freedom obtained by the same method used in the most recently fit `mixed` model with the `dfmethod()` option. If option `dfmethod()` is not specified in the previous `mixed` command, option `small` is not allowed. For certain methods, the degrees of freedom for some pairwise comparisons may not be available. See *Small-sample inference for fixed effects* in [ME] **mixed** for more details.

Remarks and examples

[stata.com](http://www.stata.com)

Various predictions, statistics, and diagnostic measures are available after fitting a mixed model using `mixed`. For the most part, calculation centers around obtaining BLUPs of the random effects. Random effects are not estimated when the model is fit but instead need to be predicted after estimation. Calculation of intraclass correlations, estimating the dependence between responses for different levels of nesting, may also be of interest.

▷ Example 1

In example 3 of [ME] **mixed**, we modeled the weights of 48 pigs measured on nine successive weeks as

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

for $i = 1, \dots, 9$, $j = 1, \dots, 48$, $\epsilon_{ij} \sim N(0, \sigma_\epsilon^2)$, and u_{0j} and u_{1j} normally distributed with mean 0 and variance–covariance matrix

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

```
. use http://www.stata-press.com/data/r14/pig
(Longitudinal analysis of pig weights)
. mixed weight week || id: week, covariance(unstructured)
Performing EM optimization:
Performing gradient-based optimization:
Iteration 0:  log likelihood = -868.96185
Iteration 1:  log likelihood = -868.96185
Computing standard errors:
Mixed-effects ML regression              Number of obs    =       432
Group variable: id                       Number of groups =        48
Obs per group:
      min =          9
      avg =         9.0
      max =          9
Wald chi2(1)                             =    4649.17
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 model: chi2(3) = 764.58 Prob > chi2 = 0.0000

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

Rather than see the estimated variance components listed as variance and covariances as above, we can instead see them as correlations and standard deviations in matrix form; that is, we can see $\hat{\Sigma}$ as a correlation matrix:

```
. estat recovariance, correlation
Random-effects correlation matrix for level id
```

	week	_cons
week	1	
_cons	-.0618257	1

We can use `estat wcorrelation` to display the within-cluster marginal standard deviations and correlations for one of the clusters.

```
. estat wcorrelation, format(%4.2g)
Standard deviations and correlations for id = 1:
```

Standard deviations:

obs	1	2	3	4	5	6	7	8	9
sd	2.9	3.1	3.3	3.7	4.1	4.5	5	5.5	6.1

Correlations:

obs	1	2	3	4	5	6	7	8	9
1	1								
2	.8	1							
3	.77	.83	1						
4	.72	.81	.86	1					
5	.67	.78	.85	.89	1				
6	.63	.75	.83	.88	.91	1			
7	.59	.72	.81	.87	.91	.93	1		
8	.55	.69	.79	.86	.9	.93	.94	1	
9	.52	.66	.77	.85	.89	.92	.94	.95	1

Because within-cluster correlations can vary between clusters, `estat wcorrelation` by default displays the results for the first cluster. In this example, each cluster (`pig`) has the same number of observations, and the timings of measurements (`week`) are the same between clusters. Thus the within-cluster correlations are the same for all the clusters. In [example 4](#), we fit a model where different clusters have different within-cluster correlations and show how to display these correlations.

We can also obtain BLUPs of the `pig`-level random effects (u_{0j} and u_{1j}). We need to specify the variables to be created in the order `u1 u0` because that is the order in which the corresponding variance components are listed in the output (`week _cons`). We obtain the predictions and list them for the first 10 pigs.

```
. predict u1 u0, reffects
. by id, sort: generate tolist = (_n==1)
. list id u0 u1 if id <=10 & tolist
```

	id	u0	u1
1.	1	.2369444	-.3957636
10.	2	-1.584127	.510038
19.	3	-3.526551	.3200372
28.	4	1.964378	-.7719702
37.	5	1.299236	-.9241479
46.	6	-1.147302	-.5448151
55.	7	-2.590529	.0394454
64.	8	-1.137067	-.1696566
73.	9	-3.189545	-.7365507
82.	10	1.160324	.0030772

If you forget how to order your variables in `predict`, or if you use `predict stub*`, remember that `predict` labels the generated variables for you to avoid confusion.

```
. describe u0 u1
```

variable name	storage type	display format	value label	variable label
u0	float	%9.0g		BLUP r.e. for id: _cons
u1	float	%9.0g		BLUP r.e. for id: week

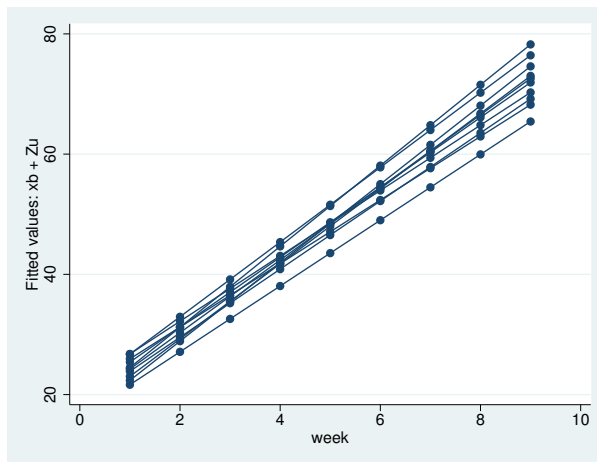
Examining (1), we see that within each pig, the successive weight measurements are modeled as simple linear regression with intercept $\beta_0 + u_{j0}$ and slope $\beta_1 + u_{j1}$. We can generate estimates of the pig-level intercepts and slopes with

```
. generate intercept = _b[_cons] + u0
. generate slope = _b[week] + u1
. list id intercept slope if id<=10 & tolist
```

	id	interc~t	slope
1.	1	19.59256	5.814132
10.	2	17.77149	6.719934
19.	3	15.82906	6.529933
28.	4	21.31999	5.437926
37.	5	20.65485	5.285748
46.	6	18.20831	5.665081
55.	7	16.76509	6.249341
64.	8	18.21855	6.040239
73.	9	16.16607	5.473345
82.	10	20.51594	6.212973

Thus we can plot estimated regression lines for each of the pigs. Equivalently, we can just plot the fitted values because they are based on both the fixed and the random effects:

```
. predict fitweight, fitted
. twoway connected fitweight week if id<=10, connect(L)
```



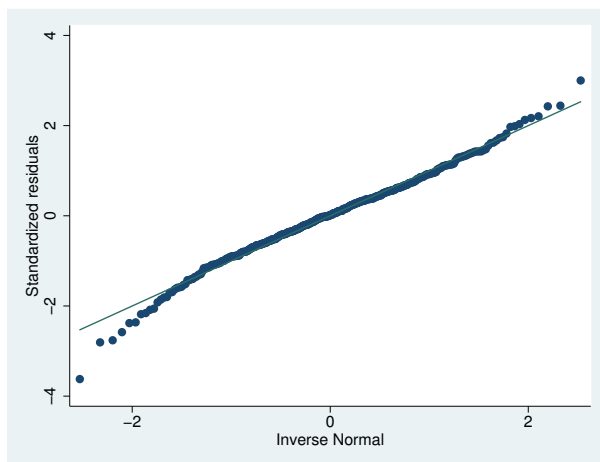
We can also generate standardized residuals and see whether they follow a standard normal distribution, as they should in any good-fitting model:

```
. predict rs, rstandard
```

```
. summarize rs
```

Variable	Obs	Mean	Std. Dev.	Min	Max
rs	432	1.01e-09	.8929356	-3.621446	3.000929

```
. qnorm rs
```



◀

► Example 2

Following [Rabe-Hesketh and Skrondal \(2012, chap. 2\)](#), we fit a two-level random-effects model for human peak-expiratory-flow rate. The subjects were each measured twice with the Mini-Wright peak-flow meter. It is of interest to determine how reliable the meter is as a measurement device. The intraclass correlation provides a measure of reliability. Formally, in a two-level random-effects model, the intraclass correlation corresponds to the correlation of measurements within the same individual and also to the proportion of variance explained by the individual random effect.

First, we fit the two-level model with mixed:

```
. use http://www.stata-press.com/data/r14/pefrate, clear
(Peak-expiratory-flow rate)
. mixed wm || id:
Performing EM optimization:
Performing gradient-based optimization:
Iteration 0:  log likelihood = -184.57839
Iteration 1:  log likelihood = -184.57839
Computing standard errors:
Mixed-effects ML regression          Number of obs    =    34
Group variable: id                  Number of groups =    17
                                     Obs per group:
                                     min =          2
                                     avg =         2.0
                                     max =          2
                                     Wald chi2(0)     =      .
                                     Prob > chi2      =      .
Log likelihood = -184.57839
```

wm	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
_cons	453.9118	26.18617	17.33	0.000	402.5878	505.2357

Random-effects Parameters		Estimate	Std. Err.	[95% Conf. Interval]	
id: Identity	var(_cons)	11458.94	3998.952	5782.176	22708.98
	var(Residual)	396.441	135.9781	202.4039	776.4942

LR test vs. linear model: $\text{chibar2}(01) = 46.27$ Prob \geq $\text{chibar2} = 0.0000$

Now we use `estat icc` to estimate the intraclass correlation:

```
. estat icc
Intraclass correlation
```

Level	ICC	Std. Err.	[95% Conf. Interval]	
id	.9665602	.0159495	.9165853	.9870185

This correlation is close to 1, indicating that the Mini-Wright peak-flow meter is reliable. But as noted by [Rabe-Hesketh and Skrondal \(2012\)](#), the reliability is not only a characteristic of the instrument but also of the between-subject variance. Here we see that the between-subject standard deviation, `sd(_cons)`, is much larger than the within-subject standard deviation, `sd(Residual)`.

In the presence of fixed-effects covariates, `estat icc` reports the residual intraclass correlation, the correlation between measurements conditional on the fixed-effects covariates. This is equivalent to the correlation of the model residuals.

In the presence of random-effects covariates, the intraclass correlation is no longer constant and depends on the values of the random-effects covariates. In this case, `estat icc` reports conditional intraclass correlations assuming 0 values for all random-effects covariates. For example, in a two-level model, this conditional correlation represents the correlation of the residuals for two measurements on the same subject, which both have random-effects covariates equal to 0. Similarly to the interpretation of intercept variances in random-coefficient models ([Rabe-Hesketh and Skrondal 2012](#), chap. 4),

interpretation of this conditional intraclass correlation relies on the usefulness of the 0 baseline values of random-effects covariates. For example, mean centering of the covariates is often used to make a 0 value a useful reference.

◀

► Example 3

In [example 4](#) of [\[ME\] mixed](#), we estimated a Cobb–Douglas production function with random intercepts at the region level and at the state-within-region level:

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

```
. use http://www.stata-press.com/data/r14/productivity
(Public Capital Productivity)
. mixed gsp private emp hwy water other unemp || region: || state:
(output omitted)
```

We can use `estat group` to see how the data are broken down by state and region:

```
. estat group
```

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

We are reminded that we have balanced productivity data for 17 years for each state.

We can use `predict, fitted` to get the fitted values

$$\hat{y}_{jk} = \mathbf{X}_{jk}\hat{\beta} + \hat{u}_k^{(3)} + \hat{u}_{jk}^{(2)}$$

but if we instead want fitted values at the region level, that is,

$$\hat{y}_{jk} = \mathbf{X}_{jk}\hat{\beta} + \hat{u}_k^{(3)}$$

we need to use the `relevel()` option:

```
. predict gsp_region, fitted relevel(region)
. list gsp gsp_region in 1/10
```

	gsp	gsp_re-n
1.	10.25478	10.40529
2.	10.2879	10.42336
3.	10.35147	10.47343
4.	10.41721	10.52648
5.	10.42671	10.54947
6.	10.4224	10.53537
7.	10.4847	10.60781
8.	10.53111	10.64727
9.	10.59573	10.70503
10.	10.62082	10.72794

□ Technical note

Out-of-sample predictions are permitted after `mixed`, but if these predictions involve BLUPs of random effects, the integrity of the estimation data must be preserved. If the estimation data have changed since the mixed model was fit, `predict` will be unable to obtain predicted random effects that are appropriate for the fitted model and will give an error. Thus to obtain out-of-sample predictions that contain random-effects terms, be sure that the data for these predictions are in observations that augment the estimation data. □

We can use `estat icc` to estimate residual intraclass correlations between productivity years in the same region and in the same state and region.

```
. estat icc
```

```
Residual intraclass correlation
```

Level	ICC	Std. Err.	[95% Conf. Interval]	
region	.159893	.127627	.0287143	.5506202
state region	.8516265	.0301733	.7823466	.9016272

`estat icc` reports two intraclass correlations for this three-level nested model. The first is the level-3 intraclass correlation at the region level, the correlation between productivity years in the same region. The second is the level-2 intraclass correlation at the state-within-region level, the correlation between productivity years in the same state and region.

Conditional on the fixed-effects covariates, we find that annual productivity is only slightly correlated within the same region, but it is highly correlated within the same state and region. We estimate that state and region random effects compose approximately 85% of the total residual variance. ◀

▷ Example 4

In [example 1](#), we fit a model where each cluster had the same model-implied within-cluster correlations. Here we fit a model where different clusters have different within-cluster correlations, and we show how to display them for different clusters. We use the Asian children weight data from [example 6](#) of [ME] `mixed`.

```
. use http://www.stata-press.com/data/r14/childweight, clear
(Weight data on Asian children)
. mixed weight age || id: age, covariance(unstructured)
```

Performing EM optimization:

Performing gradient-based optimization:

```
Iteration 0: log likelihood = -344.37065
Iteration 1: log likelihood = -342.83887
Iteration 2: log likelihood = -342.71863
Iteration 3: log likelihood = -342.71777
Iteration 4: log likelihood = -342.71777
```

Computing standard errors:

```
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(1)   =      755.27
                                Prob > chi2    =      0.0000
```

Log likelihood = -342.71777

weight	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
age	3.459671	.1258877	27.48	0.000	3.212936	3.706406
_cons	5.110496	.1494781	34.19	0.000	4.817524	5.403468

Random-effects Parameters	Estimate	Std. Err.	[95% Conf. Interval]	
id: Unstructured				
var(age)	.202392	.1242868	.0607406	.6743838
var(_cons)	.0970272	.1107998	.0103483	.9097447
cov(age, _cons)	.140134	.0566901	.0290234	.2512445
var(Residual)	1.357922	.1650502	1.070076	1.723198

LR test vs. linear model: chi2(3) = 27.38 Prob > chi2 = 0.0000

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

We use `estat wcorrelation` to display the within-cluster correlations for the first cluster.

```
. estat wcorrelation, list
Standard deviations and correlations for id = 45:
Standard deviations:
      obs |      1      2      3      4      5
-----+-----
      sd |  1.224  1.314  1.448  1.506  1.771
Correlations:
      obs |      1      2      3      4      5
-----+-----
      1 |  1.000
      2 |  0.141  1.000
      3 |  0.181  0.274  1.000
      4 |  0.193  0.293  0.376  1.000
      5 |  0.230  0.348  0.447  0.477  1.000
Data:
```

	id	weight	age
1.	45	5.171	.136893
2.	45	10.86	.657084
3.	45	13.15	1.21834
4.	45	13.2	1.42916
5.	45	15.88	2.27242

We specified the `list` option to display the data associated with the cluster. The next cluster in the dataset has ID 258. To display the within-cluster correlations for this cluster, we specify the `at()` option.

```
. estat wcorrelation, at(id=258) list
Standard deviations and correlations for id = 258:
Standard deviations:
      obs |      1      2      3      4
-----+-----
      sd |  1.231  1.320  1.424  1.782
Correlations:
      obs |      1      2      3      4
-----+-----
      1 |  1.000
      2 |  0.152  1.000
      3 |  0.186  0.270  1.000
      4 |  0.244  0.356  0.435  1.000
Data:
```

	id	weight	age
1.	258	5.3	.19165
2.	258	9.74	.687201
3.	258	9.98	1.12799
4.	258	11.34	2.30527

The within-cluster correlations for this model depend on age. The values for `age` in the two clusters are different, as are the corresponding within-cluster correlations.

▷ Example 5

To illustrate the use of `estat df`, we refit the dental veneer data from [example 14](#) of [ME] `mixed` using the Kenward–Roger method (option `dfmethod(kroger)`) to compute the DF for fixed effects.

```
. use http://www.stata-press.com/data/r14/veneer, clear
(Dental veneer data)
. mixed gcf followup base_gcf cda age || patient: followup, cov(un)
> || tooth:, reml nolog dfmethod(kroger)
Mixed-effects REML regression                Number of obs   =          110
```

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

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

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

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

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

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

Rather than specifying option `dftable(pvalue)` or `dftable(ci)` at estimation, we can display the covariate-specific DFs during postestimation by typing

```
. estat df
Degrees of freedom
```

	Kenward-Roger
gcf	
followup	10.96355
base_gcf	47.2708
cda	50.70932
age	10.41127
_cons	25.43377

`estat df` can also compare different DF methods using the `method()` option. For example, we can compare the Kenward–Roger method with the Satterthwaite method by typing

```
. estat df, method(kroger satterthwaite)
Degrees of freedom
```

	Kenward-Roger	Satterthwaite
gcf		
followup	10.96355	10.96355
base_gcf	47.2708	47.2708
cda	50.70932	50.70932
age	10.41127	10.41127
_cons	25.43377	25.43377

The two methods produce the same estimates of DFs for single-hypothesis tests, but the results differ for multiple-hypotheses tests; see [example 6](#) for details.

Suppose that we decide to proceed with the Satterthwaite method in subsequent analysis. Rather than retyping our `mixed` command with the `dfmethod(satterthwaite)` option, we can post the Satterthwaite DFs using the `post` option of `estat df`.

```
. estat df, method(satterthwaite) post
Degrees of freedom
```

	Satterthwaite
gcf	
followup	10.96355
base_gcf	47.2708
cda	50.70932
age	10.41127
_cons	25.43377

The returned values associated with `dfmethod(kroger)` from the `mixed` command will be replaced with those of `dfmethod(satterthwaite)`.

◀

► Example 6

Continuing with [example 5](#), after posting coefficient-specific DFs computed using the `dfmethod(satterthwaite)` method, we can use `test` or `testparm` with the `small` option for small-sample adjusted tests for fixed effects. For example, we can test the hypotheses that all fixed effects are zero by typing

```

. testparm *, small
( 1) [gcf]followup = 0
( 2) [gcf]base_gcf = 0
( 3) [gcf]cda = 0
( 4) [gcf]age = 0
      F( 4, 16.49) =    1.87
      Prob > F =    0.1638

```

The F statistic for the overall test is 1.87, and the DDF is estimated to be 16.49. These results are different from the model test using the Kenward–Roger DDF method reported in the header of the estimation output in [example 5](#) (the F statistic is 1.47, and the model DDF is 27.96).

The results differ because the Kenward–Roger method uses an adjusted F -test statistic and adjusts the fixed-effects variance–covariance estimator for a small sample. Both methods, however, lead to the same conclusion of no joint significance of the fixed effects.

Without option `small`, the commands `test` and `testparm` report large-sample χ^2 Wald tests. We can compare the small-sample and large-sample tests of the joint hypotheses that the coefficient on `followup` and the coefficient on `age` equal zero.

```

. test followup = age = 0, small
( 1) [gcf]followup - [gcf]age = 0
( 2) [gcf]followup = 0
      F( 2, 10.75) =    3.65
      Prob > F =    0.0617

. test followup = age = 0
( 1) [gcf]followup - [gcf]age = 0
( 2) [gcf]followup = 0
      chi2( 2) =    7.30
      Prob > chi2 =    0.0260

```

The DDF of the F test, which is computed using the Satterthwaite method from our posted results, is 10.75. The p -values are very different (0.0617 versus 0.0260), and they lead to different conclusions of whether we should reject the null hypotheses at the $\alpha = 0.05$ level.

Similarly, you can use the `small` option with `lincom` to perform small-sample inference for linear combinations of fixed effects.

◀

► Example 7

As we did with `test`, after fitting a mixed model with the `dfmethod()` option for small-sample adjustment, we can use the `small` option with `contrast` to adjust for a small sample when estimating contrasts. Suppose we have collected data on a vigilance performance test. This experiment has been designed to test the response latency scores of two modes of signal during a four-hour monitoring period. This is a split-plot factorial design where `signal` is the whole-plot factor, `hour` is the subplot factor, and `subject` is the block factor. The whole-plot factor and the subplot factor are fixed; the block factor is random. Also, suppose that two measurements are missing in this dataset.

```
. use http://www.stata-press.com/data/r14/vptscores, clear
(Vigilance Performance Test Scores With Missing Data)
. tabdisp subject hour, cellvar(score) by(signal) concise missing
```

Signal and Subject ID	Monitoring period			
	1	2	3	4
Auditory				
1	3	4	7	7
2	6	5	.	8
3	3	4	7	9
4	3	3	6	8
Visual				
5	1	2	5	10
6	2	3	6	.
7	2	4	5	9
8	2	3	6	11

We start by fitting a mixed model. Because the dataset is small and unbalanced, we apply the Kenward–Roger method for small-sample adjustment:

```
. mixed score signal##hour || subject:, reml dfmethod(kroger) nolog nogroup
Mixed-effects REML regression          Number of obs   =       30
DF method: Kenward-Roger              DF:              min =      16.02
                                       avg =      16.76
                                       max =      18.29
                                       F(7, 16.08)     =      43.84
                                       Prob > F        =      0.0000
Log restricted-likelihood = -32.9724
```

score	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
signal						
Visual	-2	.6288677	-3.18	0.005	-3.319693	-.6803071
hour						
2	.25	.5359916	0.47	0.647	-.8861371	1.386137
3	3.108222	.5911044	5.26	0.000	1.859163	4.357281
4	4.25	.5359916	7.93	0.000	3.113863	5.386137
signal#hour						
Visual#2	1	.7580066	1.32	0.206	-.6067405	2.606741
Visual#3	.6417778	.7979294	0.80	0.433	-1.046666	2.330221
Visual#4	4.044205	.7979294	5.07	0.000	2.355762	5.732649
_cons	3.75	.4446766	8.43	0.000	2.816836	4.683164

Random-effects Parameters	Estimate	Std. Err.	[95% Conf. Interval]	
subject: Identity				
var(_cons)	.2163751	.2345718	.0258477	1.811312
var(Residual)	.574574	.2062107	.2843515	1.161011

LR test vs. linear model: chibar2(01) = 1.55 Prob >= chibar2 = 0.1069

We can test the main effects and the interaction effects by typing the `contrast` command. With the `small` option, `contrast` reports small-sample adjusted F tests. Without the `small` option, `contrast` performs large-sample χ^2 Wald tests. Below is the comparison of the small-sample and the large-sample contrasts:

```
. contrast signal##hour, small
Contrasts of marginal linear predictions
Margins      : asbalanced
```

	df	ddf	F	P>F
score				
signal	1	5.95	1.78	0.2307
hour	3	16.35	100.62	0.0000
signal#hour	3	16.35	9.66	0.0007

```
. contrast signal##hour
Contrasts of marginal linear predictions
Margins      : asbalanced
```

	df	chi2	P>chi2
score			
signal	1	1.79	0.1810
hour	3	304.95	0.0000
signal#hour	3	29.35	0.0000

From these results, we can see that the p -values for the main effect of `signal` and the interaction effect vary between small-sample and large-sample tests. However, both tests indicate that the `hour` effect and the interaction effects are significant. We can decompose the interaction effect into separate interaction contrasts for further investigation.


```
. contrast r.signal#ar.hour, small
Contrasts of marginal linear predictions
Margins      : asbalanced
```

	df	ddf	F	P>F
score				
signal#hour				
(Visual vs Auditory) (2 vs 1)	1	16.02	1.74	0.2056
(Visual vs Auditory) (3 vs 2)	1	16.37	0.20	0.6594
(Visual vs Auditory) (4 vs 3)	1	16.66	16.57	0.0008
Joint	3	16.35	9.66	0.0007

	Contrast	Std. Err.	DF	[95% Conf. Interval]	
score					
signal#hour					
(Visual vs Auditory) (2 vs 1)	1	.7580066	16.0	-.6067405	2.606741
(Visual vs Auditory) (3 vs 2)	-.3582222	.7979294	16.4	-2.046666	1.330221
(Visual vs Auditory) (4 vs 3)	3.402427	.8359478	16.7	1.635991	5.168863

From previous analysis, we already knew the overall interaction was significant. From the decomposition, we can easily see that the overall significance is driven by differences in the third and fourth hours; the change in response latency from hour three to hour four is greater for visual signals than for auditory signals.

We can also calculate the pairwise differences of the hourly marginal means by typing the `pwcompare` command. With the `small` option, `pwcompare` reports small-sample adjusted pairwise comparisons along with the degrees of freedom for each pairwise comparison.

```
. pwcompare hour, small
Pairwise comparisons of marginal linear predictions
Margins      : asbalanced
```

	Contrast	Std. Err.	DF	Unadjusted [95% Conf. Interval]	
score					
hour					
2 vs 1	.75	.3790033	16.0	-.0533703	1.55337
3 vs 1	3.429111	.3989647	16.4	2.584889	4.273333
4 vs 1	6.272103	.3989647	16.4	5.427881	7.116324
3 vs 2	2.679111	.3989647	16.4	1.834889	3.523333
4 vs 2	5.522103	.3989647	16.4	4.677881	6.366324
4 vs 3	2.842991	.4179739	16.7	1.959774	3.726209

When we compare these results with the large-sample results below, we can see that the confidence interval of hour 2 versus hour 1 changes to include 0. Therefore, after adjusting for small-sample

size, we would not reject the hypothesis that the means for hour 1 and hour 2 are equivalent at the 5% significance level.

```
. pwcompare hour
Pairwise comparisons of marginal linear predictions
Margins      : asbalanced
```

		Contrast	Std. Err.	Unadjusted [95% Conf. Interval]	
score	hour				
	2 vs 1	.75	.3790033	.0071672	1.492833
	3 vs 1	3.429111	.3971529	2.650706	4.207516
	4 vs 1	6.272103	.3971529	5.493697	7.050508
	3 vs 2	2.679111	.3971529	1.900706	3.457516
	4 vs 2	5.522103	.3971529	4.743697	6.300508
	4 vs 3	2.842991	.4145085	2.03057	3.655413

◀

Stored results

`estat icc` stores the following in `r()`:

Scalars

```
r(icc#)      level-# intraclass correlation
r(se#)      standard errors of level-# intraclass correlation
r(level)    confidence level of confidence intervals
```

Macros

```
r(label#)   label for level #
```

Matrices

```
r(ci#)      vector of confidence intervals (lower and upper) for level-# intraclass correlation
```

For a G -level nested model, $\#$ can be any integer between 2 and G .

`estat recovariance` stores the following in `r()`:

Scalars

```
r(relevels) number of levels
```

Matrices

```
r(Cov#)     level-# random-effects covariance matrix
r(Corr#)    level-# random-effects correlation matrix (if option correlation was specified)
```

For a G -level nested model, $\#$ can be any integer between 2 and G .

`estat wcorrelation` stores the following in `r()`:

Matrices

```
r(sd)       standard deviations
r(Corr)     within-cluster correlation matrix
r(Cov)     within-cluster variance-covariance matrix
r(G)       variance-covariance matrix of random effects
r(Z)       model-based design matrix
r(R)       variance-covariance matrix of level-one errors
```

`estat df` stores the following in `r()`:

Macros	
<code>r(dfmethods)</code>	DF methods
Matrices	
<code>r(df)</code>	parameter-specific DFs for each method specified in <code>method()</code>
<code>r(V_df)</code>	variance-covariance matrix of the estimators when <code>kröger</code> method is specified

If option `post()` is specified, `estat df` also stores the following in `e()`:

Scalars	
<code>e(F)</code>	overall F test statistic for the method specified in <code>post()</code>
<code>e(ddf_m)</code>	model DDF for the method specified in <code>post()</code>
<code>e(df_max)</code>	maximum DF for the method specified in <code>post()</code>
<code>e(df_avg)</code>	average DF for the method specified in <code>post()</code>
<code>e(df_min)</code>	minimum DF for the method specified in <code>post()</code>
Macros	
<code>e(dfmethod)</code>	DF method specified in <code>post()</code>
<code>e(dftitle)</code>	title for DF method
Matrices	
<code>e(df)</code>	parameter-specific DFs for the method specified in <code>post()</code>
<code>e(V_df)</code>	variance-covariance matrix of the estimators when <code>kröger</code> method is posted

`pwcompare` with option `small` stores the following in `r()`:

Matrices	
<code>r(L_df)</code>	degrees of freedom for each margin difference
<code>r(M_df)</code>	degrees of freedom for each margin estimate

`pwcompare` with options `post` and `small` stores the following in `e()`:

Matrices	
<code>e(L_df)</code>	degrees of freedom for each margin difference
<code>e(M_df)</code>	degrees of freedom for each margin estimate

Methods and formulas

Methods and formulas are presented under the following headings:

Prediction
Intraclass correlations
Within-cluster covariance matrix
Small-sample inference

Prediction

Following the notation defined throughout [ME] **mixed**, BLUPs of random effects \mathbf{u} are obtained as

$$\tilde{\mathbf{u}} = \tilde{\mathbf{G}}\mathbf{Z}'\tilde{\mathbf{V}}^{-1}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})$$

where $\tilde{\mathbf{G}}$ and $\tilde{\mathbf{V}}$ are \mathbf{G} and $\mathbf{V} = \mathbf{Z}\mathbf{G}\mathbf{Z}' + \sigma_e^2\mathbf{R}$ with maximum likelihood (ML) or REML estimates of the variance components plugged in. Standard errors for BLUPs are calculated based on the iterative technique of Bates and Pinheiro (1998, sec. 3.3) for estimating the BLUPs themselves. If estimation is done by REML, these standard errors account for uncertainty in the estimate of $\boldsymbol{\beta}$, while for ML the standard errors treat $\boldsymbol{\beta}$ as known. As such, standard errors of REML-based BLUPs will usually be larger.

Fitted values are given by $\mathbf{X}\widehat{\beta} + \mathbf{Z}\widetilde{\mathbf{u}}$, residuals as $\widehat{\epsilon} = \mathbf{y} - \mathbf{X}\widehat{\beta} - \mathbf{Z}\widetilde{\mathbf{u}}$, and standardized residuals as

$$\widehat{\epsilon}_* = \widehat{\sigma}_\epsilon^{-1} \widehat{\mathbf{R}}^{-1/2} \widehat{\epsilon}$$

If the `relevel(levelvar)` option is specified, fitted values, residuals, and standardized residuals consider only those random-effects terms up to and including level *levelvar* in the model.

For details concerning the calculation of scores, see [Methods and formulas](#) in [ME] `mixed`.

Intraclass correlations

Consider a simple, two-level random-intercept model,

$$y_{ij} = \beta + u_j^{(2)} + \epsilon_{ij}^{(1)}$$

for measurements $i = 1, \dots, n_j$ and level-2 groups $j = 1, \dots, M$, where y_{ij} is a response, β is an unknown fixed intercept, u_j is a level-2 random intercept, and $\epsilon_{ij}^{(1)}$ is a level-1 error term. Errors are assumed to be normally distributed with mean 0 and variance σ_1^2 ; random intercepts are assumed to be normally distributed with mean 0 and variance σ_2^2 and to be independent of error terms.

The intraclass correlation for this model is

$$\rho = \text{Corr}(y_{ij}, y_{i'j}) = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

It corresponds to the correlation between measurements i and i' from the same group j .

Now consider a three-level nested random-intercept model,

$$y_{ijk} = \beta + u_{jk}^{(2)} + u_k^{(3)} + \epsilon_{ijk}^{(1)}$$

for measurements $i = 1, \dots, n_{jk}$ and level-2 groups $j = 1, \dots, M_{1k}$ nested within level-3 groups $k = 1, \dots, M_2$. Here $u_{jk}^{(2)}$ is a level-2 random intercept, $u_k^{(3)}$ is a level-3 random intercept, and $\epsilon_{ijk}^{(1)}$ is a level-1 error term. The error terms and random intercepts are assumed to be normally distributed with mean 0 and variances σ_1^2 , σ_2^2 , and σ_3^2 , respectively, and to be mutually independent.

We can consider two types of intraclass correlations for this model. We will refer to them as level-2 and level-3 intraclass correlations. The level-3 intraclass correlation is

$$\rho^{(3)} = \text{Corr}(y_{ijk}, y_{i'j'k}) = \frac{\sigma_3^2}{\sigma_1^2 + \sigma_2^2 + \sigma_3^2}$$

This is the correlation between measurements i and i' from the same level-3 group k and from different level-2 groups j and j' .

The level-2 intraclass correlation is

$$\rho^{(2)} = \text{Corr}(y_{ijk}, y_{i'jk}) = \frac{\sigma_2^2 + \sigma_3^2}{\sigma_1^2 + \sigma_2^2 + \sigma_3^2}$$

This is the correlation between measurements i and i' from the same level-3 group k and level-2 group j . (Note that level-1 intraclass correlation is undefined.)

More generally, for a G -level nested random-intercept model, the g -level intraclass correlation is defined as

$$\rho^{(g)} = \frac{\sum_{l=g}^G \sigma_l^2}{\sum_{l=1}^G \sigma_l^2}$$

The above formulas also apply in the presence of fixed-effects covariates \mathbf{X} in a random-effects model. In this case, intraclass correlations are conditional on fixed-effects covariates and are referred to as residual intraclass correlations. `estat icc` also uses the same formulas to compute intraclass correlations for random-coefficient models, assuming 0 baseline values for the random-effects covariates, and labels them as conditional intraclass correlations. The above formulas assume independent residual structures.

Intraclass correlations are estimated using the delta method and will always fall in (0,1) because variance components are nonnegative. To accommodate the range of an intraclass correlation, we use the logit transformation to obtain confidence intervals.

Let $\hat{\rho}^{(g)}$ be a point estimate of the intraclass correlation and $\widehat{\text{SE}}(\hat{\rho}^{(g)})$ be its standard error. The $(1 - \alpha) \times 100\%$ confidence interval for $\text{logit}(\rho^{(g)})$ is

$$\text{logit}(\hat{\rho}^{(g)}) \pm z_{\alpha/2} \frac{\widehat{\text{SE}}(\hat{\rho}^{(g)})}{\hat{\rho}^{(g)}(1 - \hat{\rho}^{(g)})}$$

where $z_{\alpha/2}$ is the $1 - \alpha/2$ quantile of the standard normal distribution and $\text{logit}(x) = \ln\{x/(1-x)\}$. Let k_u be the upper endpoint of this interval, and let k_l be the lower. The $(1 - \alpha) \times 100\%$ confidence interval for $\rho^{(g)}$ is then given by

$$\left(\frac{1}{1 + e^{-k_l}}, \frac{1}{1 + e^{-k_u}} \right)$$

Within-cluster covariance matrix

A two-level linear mixed model of the form

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

implies the marginal model

$$\mathbf{y}_j = \mathbf{X}_j \boldsymbol{\beta} + \boldsymbol{\epsilon}_j^*$$

where $\boldsymbol{\epsilon}_j^* \sim N(\mathbf{0}, \mathbf{V}_j)$, $\mathbf{V}_j = \mathbf{Z}_j \mathbf{G} \mathbf{Z}_j' + \mathbf{R}$. In a marginal model, the random part is described in terms of the marginal or total residuals $\boldsymbol{\epsilon}_j^*$, and \mathbf{V}_j is the covariance structure of these residuals.

`estat wcorrelation` calculates the marginal covariance matrix $\tilde{\mathbf{V}}_j$ for cluster j and by default displays the results in terms of standard deviations and correlations. This allows for a comparison of different multilevel models in terms of the ultimate within-cluster correlation matrix that each model implies.

Calculation of the marginal covariance matrix extends naturally to higher-level models; see, for example, chapter 4.8 in [West, Welch, and Galecki \(2015\)](#).

Small-sample inference

For small-sample computations performed when the `small` option is used with `test`, `testparm`, `lincom`, `contrast`, or `pwcompare`, see *Denominator degrees of freedom* in *Methods and formulas* of [ME] **mixed**.

References

- 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>.
- Mitchell, M. N. 2015. *Stata for the Behavioral Sciences*. College Station, TX: Stata Press.
- Rabe-Hesketh, S., and A. Skrondal. 2012. *Multilevel and Longitudinal Modeling Using Stata*. 3rd ed. College Station, TX: Stata Press.
- West, B. T., K. B. Welch, and A. T. Galecki. 2015. *Linear Mixed Models: A Practical Guide Using Statistical Software*. 2nd ed. Boca Raton, FL: Chapman & Hall/CRC.

Also see

[ME] **mixed** — Multilevel mixed-effects linear regression

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