Title

stata.com

meta me postestimation — Postestimation tools for multilevel mixed-effects meta-analysis

Postestimation commands

`meta meregress` and `meta multilevel` allow the same postestimation commands. The following postestimation commands are of special interest after `meta meregress` and `meta multilevel`:

Command	Description
`estat group`	summarize the composition of the nested groups
`estat heterogeneity`	compute multilevel heterogeneity statistics
`estat recovariance`	display the estimated random-effects covariance matrices
`estat sd`	display variance components as standard deviations and correlations

The following standard postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat ic`	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
`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 and their SEs, leverage statistics, etc.
`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. You can also obtain predictions of random effects and estimates of their standard errors.

Menu for predict

Statistics > Postestimation

Syntax for predict

Syntax for obtaining predictions other than best linear unbiased predictions (BLUPs) of random effects

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

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

predict [*type*] {*stub**|*newvarlist*} [*if*] [*in*], reffects [relevel(*levelvar*) reses(*resesspec*)]

statistic	Description
Main	
xb	linear prediction for the fixed portion of the model only; the default
stdp	standard error of the fixed-portion linear prediction
fitted	fitted values, fixed-portion linear prediction plus contributions based on predicted random effects
residuals	residuals, response minus fitted values
* rstandard	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 $\mathbf{X}_j\widehat{\boldsymbol{\beta}}$ 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 prediction $\mathbf{X}_j\widehat{\boldsymbol{\beta}}$.

fitted calculates fitted values, which are equal to the fixed-portion linear predictor *plus* contributions based on predicted random effects, $\mathbf{X}_j\widehat{\boldsymbol{\beta}}+\mathbf{Z}_j\widehat{\mathbf{u}}_j$. By default, the fitted values account for 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 trials are nested within regions, then typing

```
. predict yhat_region, fitted relevel(region)
```

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

`residuals` calculates residuals, equal to the responses minus fitted values, $\widehat{\boldsymbol{\theta}}_j - \mathbf{X}_j\widehat{\boldsymbol{\beta}} - \mathbf{Z}_j\widehat{\mathbf{u}}_j$. By default, the fitted values account for 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.

`reffects` 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 `trials` are nested within `regions`, then typing

```
. predict b*, reffects relevel(region)
```

would produce BLUPs at the region 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 *stub*1, *stub*2, ..., *stubq* for you.

`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 variable describing the grouping at that level. This option is not allowed with statistic `xb` or `stdp`.

`reses(`*resesspec*`)` calculates the standard errors of the random effects, where *resesspec* is

stub`*` | *newvarlist* [`, comparative` | `diagnostic`]

`comparative`, the default, computes comparative random-effects standard errors. For linear multilevel models, these correspond to posterior standard deviations of random effects and to standard errors of marginal prediction errors $\widehat{\mathbf{u}}_j - \mathbf{u}_j$. These standard errors are used for inference about the random effects.

`diagnostic` computes diagnostic random-effects standard errors. These correspond to marginal standard errors of BLUPs, SE$(\widehat{\mathbf{u}}_j)$. These standard errors are used for model diagnostics.

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 `reffects` 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 *stub*1, *stub*2, ..., *stubq* for you. The new variables will have the same storage type as the corresponding random-effects variables.

The `reffects` 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 `meta meregress` or `meta multilevel`. 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.

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*]

statistic	Description
xb	linear predictor for the fixed portion of the model only; the default
stdp	not allowed with margins
fitted	not allowed with margins
residuals	not allowed with margins
rstandard	not allowed with margins
reffects	not allowed with margins

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

For the full syntax, see [R] **margins**.

Remarks and examples

stata.com

Various predictions and statistics are available after fitting a multilevel meta-regression using meta multilevel or meta meregress. Below, we will discuss how to obtain BLUPs of the random effects. Random effects at different levels of hierarchy are not estimated when the model is fit but rather must be predicted after the estimation of the model parameters. The estimates of the random effects are in turn used to obtain other statistics such as the fitted values and residuals. These are useful for checking model assumptions and may be used in general as model-building tools.

▷ Example 1: Obtaining predictions of random effects

In example 2 of [META] **meta meregress**, we conducted a multilevel meta-analysis to assess the effect of modifying the school calendar on students' achievement test scores. For completeness, we refit that model here:

```
. use https://www.stata-press.com/data/r18/schoolcal
(Effect of modified school calendar on student achievement)
. meta meregress stdmdiff || district: || school:, essevariable(se)
  (output omitted )
```

The above model can also be fit by using the meta multilevel command as follows:

```
. meta multilevel stdmdiff, relevels(district school) essevariable(se)
```

We can use estat group to see how the data are broken down by district and school:

```
. estat group
```

Group variable	No. of groups	Observations per group Minimum	Average	Maximum
district	11	3	5.1	11
school	56	1	1.0	1

We are reminded that each district had somewhere between 3 to 11 schools and that each school reported one effect size in our dataset.

Below, we predict the random effects using `predict, reffects` and obtain their diagnostic standard errors by specifying the `reses(, diagnostic)` option. Because we have two random effects in our model (at levels 2 and 3), we need to specify two new variable names with `predict` and two new variable names within `reses()`. Although it is typically much easier to specify a stubname, say, `u*`, where `predict` constructs variables `u1` and `u2` for you, here we will explicitly specify the names `u3` and `u2` for the variables corresponding to the random intercepts at level 3 and level 2, respectively. We will also specify `se_u3` and `se_u2` within the `reses()` option instead of a stubname `u_se*`. And we will use suboption `diagnostic` of the `reses()` option to request the diagnostic standard errors instead of the default comparative standard errors. The diagnostic standard errors are used for model diagnostics (Goldstein 2011; Skrondal and Rabe-Hesketh 2009).

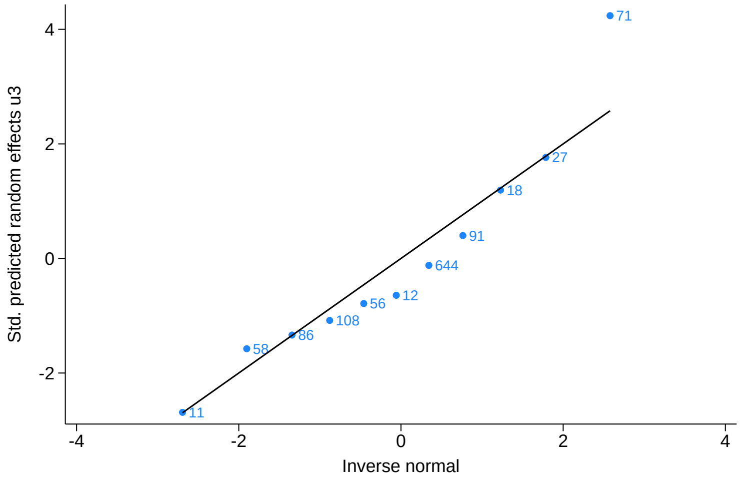
```
. predict double u3 u2, reffects reses(se_u3 se_u2, diagnostic)
. by district, sort: generate tolist = (_n==1)
. list district u3 se_u3 if tolist
```

	district	u3	se_u3
1.	11	-.18998595	.07071817
5.	12	-.08467077	.13168501
9.	18	.1407273	.11790486
12.	27	.24064814	.13641505
16.	56	-.1072942	.13633364
20.	58	-.23650899	.15003184
31.	71	.5342778	.12606073
34.	86	-.2004695	.1499012
42.	91	.05711692	.14284823
48.	108	-.14168396	.13094894
53.	644	-.01215679	.10054689

As an example, we listed the random-intercepts variable `u3` with the corresponding standard error variable `se_u3` for the $M = 11$ school districts. The purpose of variable `tolist` is to list only the unique values of `u3` and `se_u3` for each district. Had we not added the `if tolist` qualifier, row j ($j = 1, \dots, M$) would have been repeated m_j times, where m_j is the number of schools within the jth district. The random intercepts are district-specific deviations from the overall mean effect size. For example, for district 18, the predicted standardized mean difference is 0.1407 higher than the overall effect size $\widehat{b}_0 = 0.1847$, estimated in example 2 of [META] **meta meregress**, conditional on zero contribution from level-2 random intercepts.

Let's use the predicted random effects and their standard errors to compute a standardized random-effects variable, `ustan3`, to check for outliers. This new variable corresponds to the standardized random effects at the `district` level (level 3). We will use the `qnorm` command (see [R] **Diagnostic plots**) to obtain the normal quantile plot.

```
. generate double ustan3 = u3/se_u3
. label variable ustan3 "Std. predicted random effects u3"
. qnorm ustan3 if tolist, mlabel(district) xtitle("Inverse normal")
```



From the plot, district 71 appears to be an outlier. Let's list the values for districts 71 and, for comparison, 27.

```
. list district school stdmdiff if inlist(district, 71, 27), separator(4)
```

	district	school	stdmdiff
12.	27	1	.16
13.	27	2	.65
14.	27	3	.36
15.	27	4	.6
31.	71	1	.3
32.	71	2	.98
33.	71	3	1.19

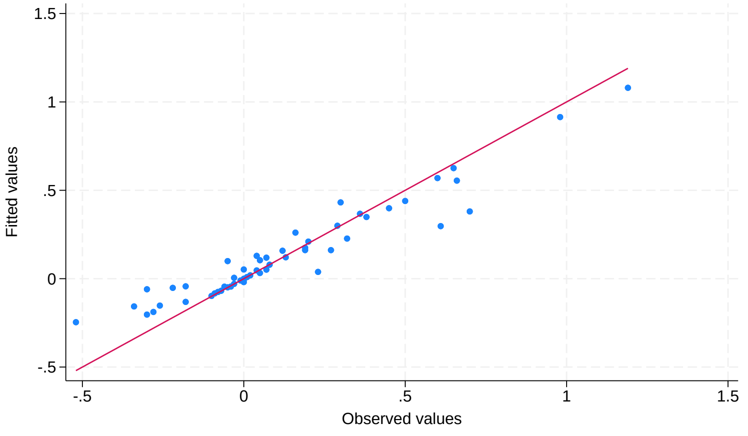
District 71 has 3 schools with students following the modified calendar scoring substantially higher ($\widehat{\theta}_{71,1} = 0.3$, $\widehat{\theta}_{71,2} = 0.98$, and $\widehat{\theta}_{71,3} = 1.19$) compared with students from schools with modified calendars in other districts such as district 27.

◁

▷ Example 2: Checking model fit

Continuing with example 1, we specify the `fitted` option to obtain the fitted values and plot them against the observed values of `stdmdiff`. By default, for the kth school within the jth district, the fitted values, $\widehat{\beta}_0 + \widehat{u}_j^{(3)} + \widehat{u}_{jk}^{(2)}$, are computed based on random-effects contributions from all levels of hierarchy. Alternatively, we could specify the `relevel(district)` option to compute the fitted values, $\widehat{\beta}_0 + \widehat{u}_j^{(3)}$, based on contributions from level-3 random-effects only.

```
. predict double fit, fitted
. twoway (scatter fit stdmdiff)
>         (function y = x, range(stdmdiff)),
> legend(off) xtitle(Observed values)  ytitle(Fitted values)
```



In the above code, we computed the fitted values using `predict, fitted`. We then produced a scatterplot of the fitted values versus the observed values of `stdmdiff`. We added a reference line $y = x$ to assess goodness of fit. Studies that are close to the reference line have their fitted values close to the observed values. Overall, it seems that goodness of fit is satisfactory.

You could also use the `rstandard` option with `predict` to compute standardized residuals. In theory, the standardized residuals are useful for checking the normality assumption of the level-1 error terms.

◁

Methods and formulas

Continuing with the notation in *Methods and formulas* of [META] **meta meregress**, the three-level meta-regression model can be expressed compactly as

$$\widehat{\boldsymbol{\theta}}_j = \mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j + \boldsymbol{\epsilon}_j, \quad j = 1, \ldots, M$$

where $m_{j.} \times (q_3 + m_j q_2)$ matrix $\mathbf{Z}_j = (\mathbf{Z}_j^{(3)}, \mathbf{Z}_j^{(2)})$ and $(q_3 + m_j q_2) \times 1$ vector $\mathbf{u}_j = (\mathbf{u}_j^{(3)\prime}, \mathbf{u}_j^{(2)\prime})'$ with a $(q_3 + m_j q_2) \times (q_3 + m_j q_2)$ covariance matrix $\boldsymbol{\Sigma}_j$

$$\boldsymbol{\Sigma}_j = \text{Var}\left(\mathbf{u}_j\right) = \begin{bmatrix} \boldsymbol{\Sigma}^{(3)} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{m_j} \otimes \boldsymbol{\Sigma}^{(2)} \end{bmatrix}$$

If we let $\mathbf{X} = (\mathbf{X}_1', \mathbf{X}_2', \ldots, \mathbf{X}_M')'$, $\boldsymbol{\Lambda} = \oplus_{j=1}^{M} \boldsymbol{\Lambda}_j$, $\mathbf{Z} = \oplus_{j=1}^{M} \mathbf{Z}_j$, and $\mathbf{u} = (\mathbf{u}_1', \mathbf{u}_2', \ldots, \mathbf{u}_M')'$, then the formulas used by `predict` for predicting random effects, residuals, etc. are described in *Methods and formulas* of [ME] **mixed postestimation** with $\mathbf{G} = \oplus_{j=1}^{M} \boldsymbol{\Sigma}_j$, $\mathbf{R} = \boldsymbol{\Lambda}$, and $\sigma_\epsilon^2 = 1$.

When the `reses()` option is specified with `reffects`, the estimated covariance matrix of $\widehat{\mathbf{u}}_j - \mathbf{u}_j$ is computed:

$$\widehat{\mathrm{Var}}\left(\widehat{\mathbf{u}}_j - \mathbf{u}_j\right) = \widehat{\boldsymbol{\Sigma}} - \widehat{\boldsymbol{\Sigma}}\mathbf{Z}_j'\mathbf{W}_j\left\{\left(\mathbf{W}_j\right)^{-1} - \mathbf{X}_j\mathrm{Var}(\widehat{\boldsymbol{\beta}})\mathbf{X}_j'\right\}\mathbf{W}_j\mathbf{Z}_j\widehat{\boldsymbol{\Sigma}}$$

The comparative standard errors of the random effects can be obtained by taking the square root of the diagonal elements of $\widehat{\mathrm{Var}}(\widehat{\mathbf{u}}_j - \mathbf{u}_j)$.

If the `diagnostic` suboption is specified within `reses()`, then the estimated covariance matrix of $\widehat{\mathbf{u}}_j$ is computed:

$$\widehat{\mathrm{Var}}\left(\widehat{\mathbf{u}}_j\right) = \widehat{\boldsymbol{\Sigma}}\mathbf{Z}_j'\mathbf{W}_j\left\{\left(\mathbf{W}_j\right)^{-1} - \mathbf{X}_j\mathrm{Var}(\widehat{\boldsymbol{\beta}})\mathbf{X}_j'\right\}\mathbf{W}_j\mathbf{Z}_j\widehat{\boldsymbol{\Sigma}}$$

The diagnostic standard errors of the random effects can be obtained by taking the square root of the diagonal elements of $\widehat{\mathrm{Var}}(\widehat{\mathbf{u}}_j)$.

See Goldstein (2011), Skrondal and Rabe-Hesketh (2009), and Rabe-Hesketh and Skrondal (2022) for more details.

References

Goldstein, H. 2011. *Multilevel Statistical Models*. 4th ed. Chichester, UK: Wiley.

Rabe-Hesketh, S., and A. Skrondal. 2022. *Multilevel and Longitudinal Modeling Using Stata*. 4th ed. College Station, TX: Stata Press.

Skrondal, A., and S. Rabe-Hesketh. 2009. Prediction in multilevel generalized linear models. *Journal of the Royal Statistical Society*, Series A 172: 659–687. https://doi.org/10.1111/j.1467-985X.2009.00587.x.

Also see

[META] **meta meregress** — Multilevel mixed-effects meta-regression

[META] **meta multilevel** — Multilevel random-intercepts meta-regression

[META] **meta** — Introduction to meta

[META] **Glossary**

[META] **Intro** — Introduction to meta-analysis

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