

meoprobit postestimation — Postestimation tools for meoprobit

Postestimation commands
Remarks and examples

predict
Methods and formulas

margins
Also see

Postestimation commands

The following postestimation command is of special interest after `meoprobit`:

Command	Description
<code>estat group</code>	summarize the composition of the nested groups
<code>estat icc</code>	estimate intraclass correlations
<code>estat sd</code>	display variance components as standard deviations and correlations

The following standard postestimation commands are also available:

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

*`hausman` and `lrtest` are not appropriate with `svy` estimation results.

predict

Description for predict

`predict` creates a new variable containing predictions such as probabilities, linear predictions, density and distribution functions, and standard errors.

Menu for predict

Statistics > Postestimation

Syntax for predict

Syntax for obtaining predictions of the outcome and other statistics

```
predict [type] { stub* | newvarlist } [if] [in] [, statistic options]
```

Syntax for obtaining estimated random effects and their standard errors

```
predict [type] { stub* | newvarlist } [if] [in], reffects [re_options]
```

Syntax for obtaining ML scores

```
predict [type] { stub* | newvarlist } [if] [in], scores
```

statistic

Description

Main	
<code>pr</code>	predicted probabilities; the default
<code>eta</code>	fitted linear predictor
<code>xb</code>	linear predictor for the fixed portion of the model only
<code>stdp</code>	standard error of the fixed-portion linear prediction
<code>density</code>	predicted density function
<code>distribution</code>	predicted distribution function

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

<i>options</i>	Description
Main	
<code>conditional(<i>ctype</i>)</code>	compute <i>statistic</i> conditional on estimated random effects; default is <code>conditional(ebmeans)</code>
<code>marginal</code>	compute <i>statistic</i> marginally with respect to the random effects
<code>nooffset</code>	make calculation ignoring offset or exposure
<code>outcome(<i>outcome</i>)</code>	outcome category for predicted probabilities
Integration	
<code>int_options</code>	integration options

You specify one or k new variables in `newvarlist` with `pr`, where k is the number of outcomes. If you do not specify `outcome()`, these options assume `outcome(#1)`.

<i>ctype</i>	Description
<code>ebmeans</code>	empirical Bayes means of random effects; the default
<code>ebmodes</code>	empirical Bayes modes of random effects
<code>fixedonly</code>	prediction for the fixed portion of the model only

<i>re_options</i>	Description
Main	
<code>ebmeans</code>	use empirical Bayes means of random effects; the default
<code>ebmodes</code>	use empirical Bayes modes of random effects
<code>reses(<i>stub*</i> <i>newvarlist</i>)</code>	calculate standard errors of empirical Bayes estimates
Integration	
<code>int_options</code>	integration options

<i>int_options</i>	Description
<code>intpoints(#)</code>	use $\#$ quadrature points to compute marginal predictions and empirical Bayes means
<code>iterate(#)</code>	set maximum number of iterations in computing statistics involving empirical Bayes estimators
<code>tolerance(#)</code>	set convergence tolerance for computing statistics involving empirical Bayes estimators

Options for predict

Main

`pr`, the default, calculates the predicted probabilities.

You specify one or k new variables, where k is the number of categories of the dependent variable. If you specify the `outcome()` option, the probabilities will be predicted for the requested outcome only, in which case you specify only one new variable. If you specify one new variable and do not specify `outcome()`, `outcome(#1)` is assumed.

`eta`, `xb`, `stdp`, `density`, `distribution`, `scores`, `conditional()`, `marginal`, and `nooffset`; see [ME] [meglm postestimation](#).

`outcome(outcome)` specifies the outcome for which the predicted probabilities are to be calculated. `outcome()` should contain either one value of the dependent variable or one of #1, #2, ..., with #1 meaning the first category of the dependent variable, #2 meaning the second category, etc. `reffects`, `ebmeans`, `ebmodes`, and `reses()`, see [ME] [meglm postestimation](#).

Integration

`intpoints()`, `iterate()`, `tolerance()`; see [ME] [meglm postestimation](#).

margins

Description for margins

`margins` estimates margins of response for probabilities and linear predictions.

Menu for margins

Statistics > Postestimation

Syntax for margins

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

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

<i>statistic</i>	Description
<code>default</code>	probabilities for each outcome
<code>pr</code>	predicted probabilities for a specified outcome
<code>eta</code>	fitted linear predictor
<code>xb</code>	linear predictor for the fixed portion of the model only
<code>stdp</code>	not allowed with <code>margins</code>
<code>density</code>	not allowed with <code>margins</code>
<code>distribution</code>	not allowed with <code>margins</code>
<code>reffects</code>	not allowed with <code>margins</code>
<code>scores</code>	not allowed with <code>margins</code>

`pr` defaults to the first outcome.

Options `conditional(ebmeans)` and `conditional(ebmodes)` are not allowed with `margins`.

Option `marginal` is assumed where applicable if `conditional(fixedonly)` is not specified.

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](https://www.stata.com)

Various predictions, statistics, and diagnostic measures are available after fitting an ordered probit mixed-effects model using `meoprobit`. Here we show a short example of predicted probabilities and predicted random effects; refer to [\[ME\] meglm postestimation](#) for additional examples applicable to mixed-effects generalized linear models.

▷ Example 1: Obtaining predicted probabilities and random effects

In [example 2](#) of [\[ME\] meoprobit](#), we modeled the tobacco and health knowledge (`thk`) score—coded 1, 2, 3, 4—among students as a function of two treatments (`cc` and `tv`) using a three-level ordered probit model with random effects at the school and class levels.

```
. use https://www.stata-press.com/data/r17/tvsfpors
(Television, School, and Family Project)
. meoprobit thk prethk cc##tv || school: || class:
(output omitted)
```

We obtain predicted probabilities for all four outcomes based on the contribution of both fixed effects and random effects by typing

```
. predict pr*
(option pr assumed)
(predictions based on fixed effects and posterior means of random effects)
(using 7 quadrature points)
```

As the note says, the predicted values are based on the posterior means of random effects. You can use the `modes` option to obtain predictions based on the posterior modes of random effects.

Because we specified a stub name, Stata saved the predicted random effects in variables `pr1` through `pr4`. Here we list the predicted probabilities for the first two classes for school 515:

```
. list class thk pr? if school==515 & (class==515101 | class==515102),
> sepby(class)
```

	class	thk	pr1	pr2	pr3	pr4
1464.	515101	2	.1503512	.2416885	.2828209	.3251394
1465.	515101	2	.3750887	.2958534	.2080368	.121021
1466.	515101	1	.3750887	.2958534	.2080368	.121021
1467.	515101	4	.2886795	.2920168	.2433916	.1759121
1468.	515101	3	.2129906	.2729831	.2696254	.2444009
1469.	515101	3	.2886795	.2920168	.2433916	.1759121
1470.	515102	1	.3318574	.2959802	.2261095	.1460529
1471.	515102	2	.4223251	.2916287	.187929	.0981172
1472.	515102	2	.4223251	.2916287	.187929	.0981172
1473.	515102	2	.4223251	.2916287	.187929	.0981172
1474.	515102	2	.3318574	.2959802	.2261095	.1460529
1475.	515102	1	.4223251	.2916287	.187929	.0981172
1476.	515102	2	.3318574	.2959802	.2261095	.1460529

For each observation, our best guess for the predicted outcome is the one with the highest predicted probability. For example, for the very first observation in the table above, we would choose outcome 4 as the most likely to occur.

We obtain predictions of the posterior means themselves at the school and class levels by typing

```
. predict re_s re_c, reffects
(calculating posterior means of random effects)
(using 7 quadrature points)
```

Here we list the predicted random effects for the first two classes for school 515:

```
. list class re_s re_c if school==515 & (class==515101 | class==515102),
> sepby(class)
```

	class	re_s	re_c
1464.	515101	-.0340769	.0390243
1465.	515101	-.0340769	.0390243
1466.	515101	-.0340769	.0390243
1467.	515101	-.0340769	.0390243
1468.	515101	-.0340769	.0390243
1469.	515101	-.0340769	.0390243
1470.	515102	-.0340769	-.0834322
1471.	515102	-.0340769	-.0834322
1472.	515102	-.0340769	-.0834322
1473.	515102	-.0340769	-.0834322
1474.	515102	-.0340769	-.0834322
1475.	515102	-.0340769	-.0834322
1476.	515102	-.0340769	-.0834322

We can see that the predicted random effects at the school level (`re_s`) are the same for all classes and that the predicted random effects at the class level (`re_c`) are constant within each class.

Methods and formulas

Methods and formulas for predicting random effects and other statistics are given in *Methods and formulas* of [ME] [meglm postestimation](#).

Also see

[ME] [meoprobit](#) — Multilevel mixed-effects ordered probit regression

[ME] [meglm postestimation](#) — Postestimation tools for meglm

[U] [20 Estimation and postestimation commands](#)