

hetoprobit postestimation — Postestimation tools for hetoprobit

Postestimation commands
Methods and formulas

predict
Also see

margins

Remarks and examples

Postestimation commands

The following postestimation commands are available after `hetoprobit`:

Command	Description
<code>contrast</code>	contrasts and ANOVA-style joint tests of estimates
<code>estat ic</code>	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, 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>forecast</code>	dynamic forecasts and simulations
* <code>hausman</code>	Hausman's specification test
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of coefficients
<code>linktest</code>	link test for model specification
* <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>	probabilities, linear predictions, etc.
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>pwcompare</code>	pairwise comparisons of estimates
<code>suest</code>	seemingly unrelated estimation
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

*`forecast`, `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, and standard deviations.

Menu for predict

Statistics > Postestimation

Syntax for predict

```
predict [type] { stub* | newvar | newvarlist } [if] [in] [, statistic
    outcome(outcome) nooffset ]
```

```
predict [type] stub* [if] [in], scores
```

<i>statistic</i>	Description
------------------	-------------

Main

<code>pr</code>	predicted probabilities; the default
<code>xb</code>	linear prediction
<code>stdp</code>	standard error of the linear prediction
<code>sigma</code>	standard deviation of the error term

You specify one or k new variables with `pr`, where k is the number of outcomes. If you specify one new variable and you do not specify `outcome()`, then `outcome(#1)` is assumed.

You specify one new variable with `xb`, `stdp`, or `sigma`.

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

Options for predict

Main

`pr`, the default, computes the predicted probabilities for all outcomes or for a specific outcome. To compute probabilities for all outcomes, you specify k new variables, where k is the number of categories of the dependent variable. Alternatively, you can specify `stub*`; in which case, `pr` will store predicted probabilities in variables `stub1`, `stub2`, ..., `stubk`. To compute the probability for a specific outcome, you specify one new variable and, optionally, the outcome value in option `outcome()`; if you omit `outcome()`, the first outcome value, `outcome(#1)`, is assumed.

Say that you fit a model by typing `estimation_cmd y x1 x2`, and `y` takes on four values. Then, you could type `predict p1 p2 p3 p4` to obtain all four predicted probabilities; alternatively, you could type `predict p*` to generate the four predicted probabilities. To compute specific probabilities one at a time, you can type `predict p1, outcome(#1)` (or simply `predict p1`), `predict p2, outcome(#2)`, and so on. See option `outcome()` for other ways to refer to outcome values.

`xb` calculates the linear prediction. The linear prediction is defined by ignoring the contribution of the estimated cutpoints.

`stdp` calculates the standard error of the linear prediction.

`sigma` calculates the standard deviation of the error term.

`outcome(outcome)` specifies for which outcome 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. `outcome()` is available only with the default `pr` option.

`nooffset` is relevant only if you specified `offset(varname)` for `hetoprobit` or within the `het()` option. `nooffset` modifies the calculations made by `predict` so that they ignore the offset variable: the linear prediction is treated as $\mathbf{x}_j\mathbf{b}$ rather than as $\mathbf{x}_j\mathbf{b} + \text{offset}_j^b$, and the prediction of $\ln(\sigma)$ is treated as $\mathbf{z}_j\mathbf{g}$ rather than as $\mathbf{z}_j\mathbf{g} + \text{offset}_j^g$. `nooffset` is not allowed with `scores`.

`scores` calculates equation-level score variables.

The first new variable will contain $\partial \ln L / \partial (\mathbf{x}_j\boldsymbol{\beta})$.

The next new variable will contain $\partial \ln L / \partial (\mathbf{z}_j\boldsymbol{\gamma})$.

The next new variable will contain $\partial \ln L / \partial \kappa_1$.

The next new variable (if any) will contain $\partial \ln L / \partial \kappa_2$.

...

The last new variable will contain $\partial \ln L / \partial \kappa_H$, where κ_h for $h = 1, 2, \dots, H$ refers to the h th cutpoint. If the linear predictor had no *indepvars*, the first new variable will contain $\partial \ln L / \partial (\mathbf{z}_j\boldsymbol{\gamma})$.

margins

Description for margins

`margins` estimates margins of response for probabilities, linear predictions, and standard deviations.

Menu for margins

Statistics > Postestimation

Syntax for margins

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

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

<i>statistic</i>	Description
Main	
default	probabilities for each outcome
pr	probability for a specified outcome
xb	linear prediction
stdp	not allowed with <code>margins</code>
sigma	standard deviation of the error term

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

See [U] [20 Estimation and postestimation commands](#) for an overview of postestimation commands, including information on obtaining the variance–covariance matrix of the estimators, predicted values, and hypothesis tests.

Once you have fit a model with `hetoprobit`, you may use the `predict` command to obtain the predicted probabilities for both the estimation sample and other samples. With the `pr` option, `predict` calculates the predicted probability of one or all ordinal value outcomes. With the `xb` option, `predict` calculates the linear prediction, $\mathbf{x}_j\mathbf{b}$, where \mathbf{x}_j are the independent variables in the j th observation and \mathbf{b} is the estimated parameter vector. The linear prediction is defined ignoring the contribution of the cutpoints. With the `sigma` option, `predict` calculates the predicted standard deviations of the error term, $\sigma_j = \exp(\mathbf{z}_j\mathbf{g})$, where \mathbf{g} is the estimated coefficient vector for the variance model.

See [example 2](#) in [R] [hetoprobit](#) for an example of `predict` after `hetoprobit`. In [example 3](#) and [example 4](#), we demonstrate how to use `margins` to obtain marginal effects, to compute expected probabilities of outcome levels across values of covariates, and to characterize the variance as a function of covariates.

Methods and formulas

For definitions of the terminology used in this section, see *Remarks and examples* in [R] **hetoprobit**.

For outcome h , the statistic `pr` is the estimated predicted probability

$$\widehat{\Pr}(y_j = h) = \Phi \left\{ \frac{\widehat{\kappa}_{h+1} - \mathbf{x}_j \mathbf{b}}{\exp(\mathbf{z}_j \mathbf{g})} \right\} - \Phi \left\{ \frac{\widehat{\kappa}_h - \mathbf{x}_j \mathbf{b}}{\exp(\mathbf{z}_j \mathbf{g})} \right\}$$

where $\mathbf{x}_j \mathbf{b}$ and $\mathbf{z}_j \mathbf{g}$ are the linear predictions of the regression and variance models, respectively, for the j th subject.

The statistic `sigma` is the estimated standard deviation of the modeled heteroskedastic error, namely,

$$\widehat{\sigma}_j = \exp(\mathbf{z}_j \mathbf{g})$$

If you specified `offset(varname)` with **hetoprobit** or within the `het()` option (and if you do not specify option `nooffset` with `predict`), then the specified offsets are applied by `predict`. Namely, the linear prediction is computed as $\mathbf{x}_j \mathbf{b} + \text{offset}_j^b$; the prediction of $\ln(\sigma)$ is computed as $\mathbf{z}_j \mathbf{g} + \text{offset}_j^g$; and all other statistics are based on the resulting predictions. If you specify `nooffset` with `predict`, then the linear prediction is $\mathbf{x}_j \mathbf{b}$ and the prediction of $\ln(\sigma)$ is $\mathbf{z}_j \mathbf{g}$, regardless of whether you specified the `offset()` option with **hetoprobit** or within `het()`.

Also see

[R] **hetoprobit** — Heteroskedastic ordered probit regression

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

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2023 StataCorp LLC, College Station, TX, USA. All rights reserved.

