Title

stata.com

ologit postestimation — Postestimation tools for ologit

Postestimation commands

The following postestimation commands are available after `ologit`:

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)
`estat (svy)`	postestimation statistics for survey data
`estimates`	cataloging estimation results
* `forecast`	dynamic forecasts and simulations
* `hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`linktest`	link test for model specification
* `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
`suest`	seemingly unrelated estimation
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

* `forecast`, `hausman`, and `lrtest` are not appropriate with `svy` estimation results. `forecast` is also not appropriate with `mi` estimation results.

predict

Description for predict

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

Menu for predict

Statistics > Postestimation

Syntax for predict

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

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

statistic	Description
Main	
`pr`	predicted probabilities; the default
`xb`	linear prediction
`stdp`	standard error of the linear prediction

If you do not specify `outcome()`, `pr` (with one new variable specified) assumes `outcome(#1)`.

You specify one or k new variables with `pr`, where k is the number of outcomes.

You specify one new variable with `xb` and `stdp`.

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, calculates the predicted probabilities. If you do not also specify the `outcome()` option, you specify k new variables, where k is the number of categories of the dependent variable. Say that you fit a model by typing `ologit result x1 x2`, and `result` takes on three values. Then you could type `predict p1 p2 p3` to obtain all three predicted probabilities. If you specify the `outcome()` option, you must specify one new variable. Say that `result` takes on the values 1, 2, and 3. Typing `predict p1, outcome(1)` would produce the same `p1`.

`xb` calculates the linear prediction. You specify one new variable, for example, `predict linear, xb`. The linear prediction is defined, ignoring the contribution of the estimated cutpoints.

`stdp` calculates the standard error of the linear prediction. You specify one new variable, for example, `predict se, stdp`.

`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.

`nooffset` is relevant only if you specified `offset(`*varname*`)` for `ologit`. It 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 $\mathbf{x}_j\mathbf{b} + \text{offset}_j$.

`scores` calculates equation-level score variables. The number of score variables created will equal the number of outcomes in the model. If the number of outcomes in the model was k, then

the first new variable will contain $\partial \ln L/\partial(\mathbf{x}_j\mathbf{b})$;

the second new variable will contain $\partial \ln L/\partial\kappa_1$;

the third new variable will contain $\partial \ln L/\partial\kappa_2$;

...

and the kth new variable will contain $\partial \ln L/\partial\kappa_{k-1}$, where κ_i refers to the ith cutpoint.

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*]

statistic	Description
default	probabilities for each outcome
`pr`	probability for a specified outcome
`xb`	linear prediction for a specified outcome
`stdp`	not allowed with `margins`

`pr` defaults to the first outcome.

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

See [U] **20 Estimation and postestimation commands** for instructions on obtaining the variance–covariance matrix of the estimators, predicted values, and hypothesis tests. Also see [R] **lrtest** for performing likelihood-ratio tests.

▷ Example 1

In example 2 of [R] **ologit**, we fit the model `ologit rep77 foreign length mpg`. The `predict` command can be used to obtain the predicted probabilities.

We type `predict` followed by the names of the new variables to hold the predicted probabilities, ordering the names from low to high. In our data, the lowest outcome is "poor", and the highest is "excellent". We have five categories, so we must type five names following `predict`; the choice of names is up to us:

```
. use http://www.stata-press.com/data/r15/fullauto
(Automobile Models)
. ologit rep77 foreign length mpg
  (output omitted)
. predict poor fair avg good exc
(option pr assumed; predicted probabilities)
. list exc good make model rep78 if rep77>=., sep(4) divider
```

	exc	good	make	model	rep78
3.	.0033341	.0393056	AMC	Spirit	.
10.	.0098392	.1070041	Buick	Opel	.
32.	.0023406	.0279497	Ford	Fiesta	Good
44.	.015697	.1594413	Merc.	Monarch	Average
53.	.065272	.4165188	Peugeot	604	.
56.	.005187	.059727	Plym.	Horizon	Average
57.	.0261461	.2371826	Plym.	Sapporo	.
63.	.0294961	.2585825	Pont.	Phoenix	.

The eight cars listed were introduced after 1977, so they do not have 1977 repair records in our data. We predicted what their 1977 repair records might have been using the fitted model. We see that, based on its characteristics, the Peugeot 604 had about a $41.65 + 6.53 \approx 48.2\%$ chance of a good or excellent repair record. The Ford Fiesta, which had only a 3% chance of a good or excellent repair record, in fact, had a good record when it was introduced in the following year.

◁

❑ Technical note

For ordered logit, `predict, xb` produces $S_j = x_{1j}\beta_1 + x_{2j}\beta_2 + \cdots + x_{kj}\beta_k$. The ordered-logit predictions are then the probability that $S_j + u_j$ lies between a pair of cutpoints, κ_{i-1} and κ_i. Some handy formulas are

$$\Pr(S_j + u_j < \kappa) = 1/(1 + e^{S_j - \kappa})$$
$$\Pr(S_j + u_j > \kappa) = 1 - 1/(1 + e^{S_j - \kappa})$$
$$\Pr(\kappa_1 < S_j + u_j < \kappa_2) = 1/(1 + e^{S_j - \kappa_2}) - 1/(1 + e^{S_j - \kappa_1})$$

Rather than using `predict` directly, we could calculate the predicted probabilities by hand. If we wished to obtain the predicted probability that the repair record is excellent and the probability that it is good, we look back at `ologit`'s output to obtain the cutpoints. We find that "good" corresponds to the interval `/cut3` $< S_j + u <$ `/cut4` and "excellent" to the interval $S_j + u >$ `/cut4`:

```
. predict score, xb
. generate probgood = 1/(1+exp(score-_b[/cut4])) - 1/(1+exp(score-_b[/cut3]))
. generate probexc = 1 - 1/(1+exp(score-_b[/cut4]))
```

The results of our calculation will be the same as those produced in the previous example. We refer to the estimated cutpoints just as we would any coefficient, so `_b[/cut3]` refers to the value of the `/cut3` coefficient; see **[U] 13.5 Accessing coefficients and standard errors**.

❑

Also see

[R] **ologit** — Ordered logistic regression

[U] 20 Estimation and postestimation commands