probit postestimation — Postestimation tools for probit

Description

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

Command	Description
`estat classification`	report various summary statistics, including the classification table
`estat gof`	Pearson or Hosmer–Lemeshow goodness-of-fit test
`lroc`	compute area under ROC curve and graph the curve
`lsens`	graph sensitivity and specificity versus probability cutoff

These commands are not appropriate after the `svy` prefix.

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)
`estat` (svy)	postestimation statistics for survey data
`estimates`	cataloging estimation results
`forecast`[1]	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`[2]	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

[1] `forecast` is not appropriate with `mi` or `svy` estimation results.

[2] `lrtest` is not appropriate with `svy` estimation results.

Syntax for predict

```
predict [type] newvar [if] [in] [, statistic nooffset rules asif]
```

statistic	Description
Main	
`pr`	probability of a positive outcome; the default
`xb`	linear prediction
`stdp`	standard error of the linear prediction
* `deviance`	deviance residual
`score`	first derivative of the log likelihood with respect to $\mathbf{x}_j\boldsymbol{\beta}$

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.

Menu for predict

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

`pr`, the default, calculates the probability of a positive outcome.

`xb` calculates the linear prediction.

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

`deviance` calculates the deviance residual.

`score` calculates the equation-level score, $\partial \ln L/\partial(\mathbf{x}_j\boldsymbol{\beta})$.

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

`rules` requests that Stata use any rules that were used to identify the model when making the prediction. By default, Stata calculates missing for excluded observations.

`asif` requests that Stata ignore the rules and exclusion criteria and calculate predictions for all observations possible using the estimated parameter from the model.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Obtaining predicted values

Once you have fit a probit model, you can obtain the predicted probabilities by using the `predict` command for both the estimation sample and other samples; see **[U] 20 Estimation and postestimation commands** and [R] **predict**. Here we will make only a few additional comments.

`predict` without arguments calculates the predicted probability of a positive outcome. With the `xb` option, `predict` calculates the linear combination $\mathbf{x}_j\mathbf{b}$, where $\mathbf{x}_j$ are the independent variables in the jth observation and $\mathbf{b}$ is the estimated parameter vector. This is known as the index function because the cumulative density indexed at this value is the probability of a positive outcome.

In both cases, Stata remembers any rules used to identify the model and calculates missing for excluded observations unless `rules` or `asif` is specified. This is covered in the following example.

With the `stdp` option, `predict` calculates the standard error of the prediction, which is *not* adjusted for replicated covariate patterns in the data.

You can calculate the unadjusted-for-replicated-covariate-patterns diagonal elements of the hat matrix, or leverage, by typing

```
. predict pred

. predict stdp, stdp

. generate hat = stdp^2*pred*(1-pred)
```

▷ Example 1

In example 4 of [R] **probit**, we fit the probit model `probit foreign b3.repair`. To obtain predicted probabilities, we type

```
. predict p
(option pr assumed; Pr(foreign))
(10 missing values generated)

. summarize foreign p

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
     foreign |         58    .2068966    .4086186          0          1
           p |         48         .25    .1956984         .1         .5
```

Stata remembers any rules used to identify the model and sets predictions to missing for any excluded observations. In example 4 of [R] **probit**, `probit` dropped the variable `1.repair` from our model and excluded 10 observations. When we typed `predict p`, those same 10 observations were again excluded and their predictions set to missing.

`predict`'s `rules` option uses the rules in the prediction. During estimation, we were told, "1.repair != 0 predicts failure perfectly", so the rule is that when `1.repair` is not zero, we should predict 0 probability of success or a positive outcome:

```
. predict p2, rules
(option pr assumed; Pr(foreign))

. summarize foreign p p2

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
     foreign |         58    .2068966    .4086186          0          1
           p |         48         .25    .1956984         .1         .5
          p2 |         58    .2068966    .2016268          0         .5
```

`predict`'s `asif` option ignores the rules and the exclusion criteria and calculates predictions for all observations possible using the estimated parameters from the model:

```
. predict p3, asif
(option pr assumed; Pr(foreign))
. summarize for p p2 p3
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
     foreign |        58    .2068966    .4086186          0          1
           p |        48         .25    .1956984         .1         .5
          p2 |        58    .2068966    .2016268          0         .5
          p3 |        58    .2931034    .2016268         .1         .5
```

Which is right? By default, `predict` uses the most conservative approach. If many observations had been excluded due to a simple rule, we could be reasonably certain that the `rules` prediction is correct. The `asif` prediction is correct only if the exclusion is a fluke and we would be willing to exclude the variable from the analysis, anyway. Then, however, we should refit the model to include the excluded observations.

◁

Performing hypothesis tests

After estimation with `probit`, you can perform hypothesis tests by using the `test` or `testnl` command; see [U] **20 Estimation and postestimation commands**.

Methods and formulas

Let index j be used to index observations. Define M_j for each observation as the total number of observations sharing j's covariate pattern. Define Y_j as the total number of positive responses among observations sharing j's covariate pattern. Define p_j as the predicted probability of a positive outcome for observation j.

For $M_j > 1$, the deviance residual d_j is defined as

$$d_j = \pm\left(2\left[Y_j \ln\left(\frac{Y_j}{M_j p_j}\right) + (M_j - Y_j)\ln\left\{\frac{M_j - Y_j}{M_j(1-p_j)}\right\}\right]\right)^{1/2}$$

where the sign is the same as the sign of $(Y_j - M_j p_j)$. In the limiting cases, the deviance residual is given by

$$d_j = \begin{cases} -\sqrt{2M_j|\ln(1-p_j)|} & \text{if } Y_j = 0 \\ \sqrt{2M_j|\ln p_j|} & \text{if } Y_j = M_j \end{cases}$$

Also see

[R] **probit** — Probit regression

[R] **estat classification** — Classification statistics and table

[R] **estat gof** — Pearson or Hosmer–Lemeshow goodness-of-fit test

[R] **lroc** — Compute area under ROC curve and graph the curve

[R] **lsens** — Graph sensitivity and specificity versus probability cutoff

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