cpoisson postestimation — Postestimation tools for cpoisson

Postestimation commands

The following postestimation commands are available after `cpoisson`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of parameters
`estat ic`	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC, respectively)
`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
`etable`	table of estimation results
* `forecast`	dynamic forecasts and simulations
* `hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
* `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 parameters
`predict`	number of events, incidence rates, probabilities, etc.
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of parameters
`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.

predict

Description for predict

`predict` creates a new variable containing predictions such as numbers of events, incidence rates, conditional means, probabilities, conditional probabilities, linear predictions, standard errors, and the equation-level score.

Menu for predict

Statistics > Postestimation

Syntax for predict

`predict` [*type*] *newvar* [*if*] [*in*] [, *statistic* `nooffset`]

statistic	Description
Main	
`n`	number of events; the default
`ir`	incidence rate
`cm`	conditional mean, $E(y_j \mid y_j > L_j)$, $E(y_j \mid y_j < U_j)$, or $E(y_j \mid L_j < y_j < U_j)$
`pr(`*n*`)`	probability $\Pr(y_j = n)$
`pr(`*a*`,`*b*`)`	probability $\Pr(a \le y_j \le b)$
`cpr(`*n*`)`	conditional probability $\Pr(y_j = n \mid y_j > L_j)$, $\Pr(y_j = n \mid y_j < U_j)$, or $\Pr(y_j = n \mid L_j < y_j < U_j)$
`cpr(`*a*`,`*b*`)`	conditional probability $\Pr(a \le y_j \le b \mid y_j > L_j)$, $\Pr(a \le y_j \le b \mid y_j < U_j)$, or $\Pr(a \le y_j \le b \mid L_j < y_j < U_j)$
`xb`	linear prediction
`stdp`	standard error of the linear prediction
`score`	first derivative of the log likelihood with respect to $\mathbf{x}_j\boldsymbol{\beta}$

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

`n`, the default, calculates the predicted number of events, which is $\exp(\mathbf{x}_j\boldsymbol{\beta})$ if neither `offset()` nor `exposure()` was specified when the model was fit; $\exp(\mathbf{x}_j\boldsymbol{\beta} + \text{offset}_j)$ if `offset()` was specified; or $\exp(\mathbf{x}_j\boldsymbol{\beta}) \times \text{exposure}_j$ if `exposure()` was specified.

`ir` calculates the incidence rate $\exp(\mathbf{x}_j\boldsymbol{\beta})$, which is the predicted number of events when exposure is 1. This is equivalent to specifying both the `n` and the `nooffset` options.

`cm` calculates the conditional mean,

$$E(y_j \mid \Omega_j) = \frac{E(y_j)}{\Pr(\Omega_j)}$$

where Ω_j represents $y_j > L_j$ for a left-censored model, $y_j < U_j$ for a right-censored model, and $L_j < y_j < U_j$ for an interval-censored model. L_j is the left-censoring point found in `e(llopt)`, and U_j is the right-censoring point found in `e(ulopt)`.

`pr(`*n*`)` calculates the probability $\Pr(y_j = n)$, where *n* is a nonnegative integer that may be specified as a number or a variable.

`pr(`*a*`,`*b*`)` calculates the probability $\Pr(a \leq y_j \leq b)$, where *a* and *b* are nonnegative integers that may be specified as numbers or variables;

b missing ($b \geq .$) means $+\infty$;
`pr(20,.)` calculates $\Pr(y_j \geq 20)$;
`pr(20,`*b*`)` calculates $\Pr(y_j \geq 20)$ in observations for which $b \geq .$ and calculates $\Pr(20 \leq y_j \leq b)$ elsewhere.

`pr(.,`*b*`)` produces a syntax error. A missing value in an observation of the variable *a* causes a missing value in that observation for `pr(`*a*`,`*b*`)`.

`cpr(`*n*`)` calculates the conditional probability $\Pr(y_j = n \mid \Omega_j)$, where Ω_j represents $y_j > L_j$ for a left-censored model, $y_j < U_j$ for a right-censored model, and $L_j < y_j < U_j$ for an interval-censored model. L_j is the left-censoring point found in `e(llopt)`, and U_j is the right-censoring point found in `e(ulopt)`. *n* is an integer in the noncensored range.

`cpr(`*a*`,`*b*`)` calculates the conditional probability $\Pr(a \leq y_j \leq b \mid \Omega_j)$, where Ω_j represents $y_j > L_j$ for a left-censored model, $y_j < U_j$ for a right-censored model, and $L_j < y_j < U_j$ for an interval-censored model. L_j is the left-censoring point found in `e(llopt)`, and U_j is the right-censoring point found in `e(ulopt)`. *a* and *b* must fall in the noncensored range if they are not missing. A missing value in an observation of the variable *a* causes a missing value in that observation for `cpr(`*a*`,`*b*`)`.

`xb` calculates the linear prediction, which is $\mathbf{x}_j\boldsymbol{\beta}$ if neither `offset()` nor `exposure()` was specified when the model was fit; $\mathbf{x}_j\boldsymbol{\beta} + \text{offset}_j$ if `offset()` was specified; or $\mathbf{x}_j\boldsymbol{\beta} + \ln(\text{exposure}_j)$ if `exposure()` was specified; see `nooffset` below.

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

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

`nooffset` is relevant only if you specified `offset()` or `exposure()` when you fit the model. It modifies the calculations made by `predict` so that they ignore the offset or exposure variable; the linear prediction is treated as $\mathbf{x}_j\boldsymbol{\beta}$ rather than as $\mathbf{x}_j\boldsymbol{\beta} + \text{offset}_j$ or $\mathbf{x}_j\boldsymbol{\beta} + \ln(\text{exposure}_j)$. Specifying `predict ..., nooffset` is equivalent to specifying `predict ..., ir`.

margins

Description for margins

margins estimates margins of response for numbers of events, incidence rates, conditional means, probabilities, and linear predictions.

Menu for margins

Statistics > Postestimation

Syntax for margins

margins [*marginlist*] [, *options*]

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

statistic	Description
n	number of events; the default
ir	incidence rate
cm	conditional mean, $E(y_j \mid y_j > L_j)$, $E(y_j \mid y_j < U_j)$, or $E(y_j \mid L_j < y_j < U_j)$
pr(*n*)	probability $\Pr(y_j = n)$
pr(*a*,*b*)	probability $\Pr(a \leq y_j \leq b)$
cpr(*n*)	conditional probability $\Pr(y_j = n \mid y_j > L_j)$, $\Pr(y_j = n \mid y_j < U_j)$, or $\Pr(y_j = n \mid L_j < y_j < U_j)$
cpr(*a*,*b*)	conditional probability $\Pr(a \leq y_j \leq b \mid y_j > L_j)$, $\Pr(a \leq y_j \leq b \mid y_j < U_j)$, or $\Pr(a \leq y_j \leq b \mid L_j < y_j < U_j)$
xb	linear prediction
stdp	not allowed with margins
score	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

▷ Example 1: Obtaining marginal effects

Continuing with example 1 of [R] **cpoisson**, we estimate the effect of having another child on the uncensored number of trips to amusement parks. We use margins to estimate the average number of trips when each household has its actual number of children and when each household has one additional child. We include the post option so that we can use the results in subsequent commands.

```
. use https://www.stata-press.com/data/r19/trips
(Visits to the ABC amusement park)

. cpoisson trips income children, ul(3)
  (output omitted)
```

```
. margins, at(children = generate(children))
>          at(children = generate(children+1)) post

Predictive margins                                      Number of obs = 500
Model VCE: OIM

Expression: Predicted number of events, predict()
1._at: children =   children
2._at: children = children+1

------------------------------------------------------------------------------
             |            Delta-method
             |     Margin   std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
         _at |
          1  |   2.525517   .0836237    30.20   0.000     2.361618    2.689417
          2  |   2.889658   .1337997    21.60   0.000     2.627416    3.151901
------------------------------------------------------------------------------
```

An average of 2.53 trips are taken when each household has its observed number of children. If each household has one additional child, then the average number of trips increases to 2.89.

We now use `contrast` to compute the effect of having an additional child. The Wald test, in this case, is superfluous, so we suppress it with the `nowald` option.

```
. contrast r._at, nowald

Contrasts of predictive margins                         Number of obs = 500
Model VCE: OIM

Expression: Predicted number of events, predict()
1._at: children =   children
2._at: children = children+1

--------------------------------------------------------------
             |            Delta-method
             |   Contrast   std. err.     [95% conf. interval]
-------------+------------------------------------------------
         _at |
   (2 vs 1)  |   .3641407   .0849951      .1975535     .530728
--------------------------------------------------------------
```

Adding one child to each household increases the average by 0.36 trips.

◁

Methods and formulas

Using the notation under *Methods and formulas* of [R] **cpoisson**, we see that the equation-level score is given by

$$\begin{aligned}\text{score}(\mathbf{x}\boldsymbol{\beta})_j =& d_j\{-\exp(\xi_j)+y_j\} \\ &+(1-d_j)\frac{\Psi_1(L_j|\mathbf{x}_j)-\Psi_1(U_j-1|\mathbf{x}_j)}{1-F(U_j-1|\mathbf{x}_j)+F(L_j|\mathbf{x}_j)}\end{aligned}$$

where $\Psi_1(C)=\sum_{k=0}^{C} f(k|\mathbf{x}_j)\{k-\exp(\xi_j)\}$; $f(y_j|\mathbf{x}_j)$ and $F(y_j|\mathbf{x}_j)$ denote the probability mass function and the cumulative distribution function of the Poisson, respectively. L_j is the left-censoring point found in `e(llopt)`, and U_j is the right-censoring point found in `e(ulopt)`.

Also see

[R] **cpoisson** — Censored Poisson regression

[U] 20 Estimation and postestimation commands

For suggested citations, see the FAQ on citing Stata documentation.