

heckpoisson postestimation — Postestimation tools for heckpoisson

[Postestimation commands](#)
[predict](#)
[margins](#)
[Remarks and examples](#)
[Methods and formulas](#)
[Also see](#)

Postestimation commands

The following postestimation commands are available after `heckpoisson`:

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>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 <code>margins</code> (profile plots, interaction plots, etc.)
<code>nlcom</code>	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
<code>predict</code>	number of events, incidence rates, probabilities, 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 new variables containing predictions such as number of events, incidence rates, conditional predicted number of events, probabilities, linear predictions, and equation-level scores.

Menu for predict

Statistics > Postestimation

Syntax for predict

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

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

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

Main

<code>n</code>	number of events; the default
<code>ir</code>	incidence rate
<code>ncond</code>	predicted number of events conditional on y_j being observed
<code>pr(n)</code>	$\Pr(y_j = n)$
<code>pr(a, b)</code>	$\Pr(a \leq y_j \leq b)$
<code>pselect</code>	$\Pr(y_j \text{ observed})$
<code>xb</code>	linear prediction
<code>xbselect</code>	linear prediction for selection equation

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 $\exp(\mathbf{x}_j\beta + \sigma^2/2)$ if neither `offset()` nor `exposure()` was specified when the model was fit; is $\exp(\mathbf{x}_j\beta + \sigma^2/2 + \text{offset}_j)$ if `offset()` was specified; or is $\exp(\mathbf{x}_j\beta + \sigma^2/2) \times \text{exposure}_i$ if `exposure()` was specified.

`ir` calculates the incidence rate $\exp(\mathbf{x}_j\beta + \sigma^2/2)$, which is the predicted number of events when exposure is 1. Specifying `ir` is equivalent to specifying `n` when neither `offset()` nor `exposure()` was specified when the model was fit.

`ncond` calculates the predicted number of events conditional on y_j being observed, which is $\exp(\mathbf{x}_j\beta + \sigma^2/2)\Phi(\mathbf{w}_j\gamma + \rho\sigma)/\Phi(\mathbf{w}_j\gamma)$.

`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)`.

`pselect` calculates the probability of selection (or being observed):

$\Pr(y_j \text{ observed}) = \Pr(\mathbf{w}_j\gamma + \epsilon_{2j} > 0)$

`xb` calculates the linear prediction for the dependent count variable, which is $\mathbf{x}_j\beta$ if neither `offset()` nor `exposure()` was specified; $\mathbf{x}_j\beta + \text{offset}_j^\beta$ if `offset()` was specified; or $\mathbf{x}_j\beta + \ln(\text{exposure}_j)$ if `exposure()` was specified.

`xbselect` calculates the linear prediction for the selection equation, which is $\mathbf{w}_j\gamma$ if `offset()` was not specified in `select()` and is $\mathbf{w}_j\gamma + \text{offset}_j^\gamma$ if `offset()` was specified in `select()`.

`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 $\mathbf{x}_j\beta$ rather than as $\mathbf{x}_j\beta + \text{offset}_j$ or $\mathbf{x}_j\beta + \ln(\text{exposure}_j)$.

`scores` calculates equation-level score variables.

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

The second new variable will contain $\partial \ln L / \partial (\mathbf{w}_j\gamma)$.

The third new variable will contain $\partial \ln L / \partial \text{atanh } \rho$.

The fourth new variable will contain $\partial \ln L / \partial \ln \sigma$.

margins

Description for margins

`margins` estimates margins of response for number of events, incidence rates, conditional predicted number of events, 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>n</code>	number of events; the default
<code>ir</code>	incidence rate
<code>ncond</code>	predicted number of events conditional on y_j being observed
<code>pr(n)</code>	$\Pr(y_j = n)$
<code>pr(a, b)</code>	$\Pr(a \leq y_j \leq b)$
<code>psel</code>	$\Pr(y_j \text{ observed})$
<code>xb</code>	linear prediction
<code>xbsel</code>	linear prediction for selection equation

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)

► Example 1: Obtaining margins for a count model with selection

In [example 1](#) of [R] [heckpoisson](#), we fit a model for the number of patents. In that example, we are interested in the effect of R&D expenditures on the number of patents received by a firm. We continue that example to determine the magnitude of the effect of R&D expenditures on the number of patents and compare this effect for IT and non-IT sectors.

After reading in the data and fitting the model, we use `margins` to estimate the effect of an increase of a million dollars in R&D expenditures (`expenditure`) on the number of patents (`npatents`) for firms in the IT and non-IT sectors (`tech`).

To do this, we use the `at()` option of `margins`. We use the observed values in our first scenario, so we tell `margins` to set `expenditure` equal to itself. For our second scenario, we tell `margins` to set `expenditure` equal to the observed value plus 1 because expenditures are measured in millions of dollars. We include the `post` option so that we can perform additional calculations later.

```

. use https://www.stata-press.com/data/r18/patent
(Fictional data on patents and R&D)
. quietly heckpoisson npatents expenditure i.tech,
> select(applied = expenditure size i.tech)
. margins i.tech, at(expenditure = generate(expenditure))
> at(expenditure = generate(expenditure+1)) post
Predictive margins                                Number of obs = 10,000
Model VCE: OIM
Expression: Predicted number of events, predict()
1._at: expenditure = expenditure
2._at: expenditure = expenditure+1

```

	Delta-method		z	P> z	[95% conf. interval]	
	Margin	std. err.				
<hr/>						
_at#tech						
1 #						
Non-IT se..	1.276213	.0556644	22.93	0.000	1.167112	1.385313
1#IT sector	2.287013	.080119	28.55	0.000	2.129983	2.444044
2 #						
Non-IT se..	2.099539	.131364	15.98	0.000	1.84207	2.357007
2#IT sector	3.76244	.2226221	16.90	0.000	3.326109	4.198771

The output indicates that the expected number of patents for non-IT firms is about 1.28 compared with 2.29 for firms in the IT sector.

The second scenario shows the expected number of patents after our hypothetical increase in R&D expenditures. In the non-IT sector, the expected number of patents received would be about 2.10 compared with 3.76 in the IT sector. It appears that increasing expenditures may have a larger effect for IT firms—the difference between the two scenarios is 1.47 for IT firms and only 0.82 for non-IT firms. We can test whether the effect of increasing expenditures is different for IT and non-IT firms. We use `lincom` to obtain an estimate of the difference in the differences between scenarios for the two sectors and a test of its significance. We ask for the differences by referring to the scenarios as 1._at and 2._at and by referring to the sector using the value that corresponds to the IT sector indicator, 1.tech for IT firms and 0.tech otherwise.

```

. lincom (_b[2._at#1.tech] - _b[1._at#1.tech]) -
> (_b[2._at#0.tech] - _b[1._at#0.tech])
( 1) 1bn._at#0bn.tech - 1bn._at#1.tech - 2._at#0bn.tech + 2._at#1.tech = 0

```

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
(1)	.6521006	.0917299	7.11	0.000	.4723134	.8318878

We find that the expected effect of increasing R&D expenditures by one million dollars is 0.65 patents larger for IT firms than for non-IT firms, and this difference is significantly different from 0. ◀

Methods and formulas

Suppose that the count outcome y_j has covariates \mathbf{x}_j and the selection outcome s_j has covariates \mathbf{w}_j . y_j is assumed to have a Poisson distribution, conditional on \mathbf{x}_j , with conditional mean

$$E(y_j | \mathbf{x}_j, \epsilon_{1j}) = \mu_j = \exp(\mathbf{x}_j \boldsymbol{\beta} + \epsilon_{1j})$$

s_j is a binary outcome from a latent-variable model:

$$s_j = \begin{cases} 1, & \text{if } \mathbf{w}_j\gamma + \epsilon_{2j} > 0 \\ 0, & \text{otherwise} \end{cases}$$

The expectation of y_j conditional on covariates \mathbf{x}_j for the whole population is

$$E(y_j|\mathbf{x}_j) = \exp(\mathbf{x}_j\boldsymbol{\beta} + \sigma^2/2)$$

Furthermore, if we want the expectation of y_j only if it was observed, then the formula is

$$E(y_j|\mathbf{x}_j, \mathbf{w}_j, s_j = 1) = \exp(\mathbf{x}_j\boldsymbol{\beta} + \sigma^2/2) \frac{\Phi(\mathbf{w}_j\gamma + \rho\sigma)}{\Phi(\mathbf{w}_j\gamma)}$$

We note that if $\rho = 0$, this expectation is the same as its population version.

We can also predict the probability of y_j conditional on \mathbf{x}_j . Note that although y_j is Poisson-distributed conditional on ϵ_1 and \mathbf{x}_j , the distribution of y_j is unknown unconditional on ϵ_1 .

$$\Pr(y_j = n|\mathbf{x}_j) = \int_{-\infty}^{\infty} \Pr(y_j = n|\mathbf{x}_j, \epsilon_1) \phi(\epsilon_1/\sigma) d\epsilon_1$$

As in the implementation of log likelihood, we approximate this integral by Gauss–Hermite quadrature.

Also see

[R] [heckpoisson](#) — Poisson regression with sample selection

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

