Title

poisson postestimation — Postestimation tools for poisson

Description

The following postestimation command is of special interest after `poisson`:

command	description
`estat gof`	goodness-of-fit test

`estat gof` is not appropriate after the `svy` prefix. For information about `estat gof`, see below.

The following standard postestimation commands are also available:

command	description
`estat`	AIC, BIC, VCE, and estimation sample summary
`estat` (svy)	postestimation statistics for survey data
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`linktest`	link test for model specification
`lrtest`[1]	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`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
`suest`	seemingly unrelated estimation
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

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

See the corresponding entries in the *Base Reference Manual* for details, but see [SVY] **estat** for details about `estat` (svy).

Special-interest postestimation command

`estat gof` performs a goodness-of-fit test of the model. The default is the deviance statistic; specifying option `pearson` will give the Pearson statistic. If the test is significant, the Poisson regression model is inappropriate. Then you could try a negative binomial model; see [R] **nbreg**.

Syntax for predict

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

statistic	description
Main	
n	number of events; the default
ir	incidence rate
pr(*n*)	probability $\Pr(y_j = n)$
pr(*a*,*b*)	probability $\Pr(a \le y_j \le b)$
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.

Menu

Statistics > Postestimation > Predictions, residuals, etc.

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. Specifying ir is equivalent to specifying n when neither offset() nor exposure() was specified when the model was fit.

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 $\Pr(a \le y_j \le b)$, where *a* and *b* are nonnegative integers that may be specified as numbers or variables;

b missing ($b \ge .$) means $+\infty$;
pr(20,.) calculates $\Pr(y_j \ge 20)$;
pr(20,*b*) calculates $\Pr(y_j \ge 20)$ in observations for which $b \ge .$ and calculates $\Pr(20 \le y_j \le b)$ 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*).

xb calculates the linear prediction, which is $\mathbf{x}_j\boldsymbol{\beta}$ if neither offset() nor exposure() was specified; $\mathbf{x}_j\boldsymbol{\beta} + \text{offset}_j$ 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`.

Syntax for estat gof

```
estat gof [, pearson ]
```

Menu

Statistics > Postestimation > Reports and statistics

Option for estat gof

`pearson` requests that `estat gof` calculate the Pearson statistic rather than the deviance statistic.

Remarks

▷ Example 1

Continuing with example 2 of [R] **poisson**, we use `estat gof` to determine whether the model fits the data well.

```
. use http://www.stata-press.com/data/r11/dollhill3
. poisson deaths smokes i.agecat, exp(pyears) irr
  (output omitted)
. estat gof

         Goodness-of-fit chi2  =  12.13244
         Prob > chi2(4)        =    0.0164
```

The goodness-of-fit χ^2 tells us that, given the model, we can reject the hypothesis that these data are Poisson distributed at the 1.64% significance level.

So let us now back up and be more careful. We can most easily obtain the incidence-rate ratios within age categories by using `ir`; see [ST] **epitab**:

```
. ir deaths smokes pyears, by(agecat) nohet
```

agecat	IRR	[95% Conf.	Interval]	M-H Weight	
1	5.736638	1.463557	49.40468	1.472169	(exact)
2	2.138812	1.173714	4.272545	9.624747	(exact)
3	1.46824	.9863624	2.264107	23.34176	(exact)
4	1.35606	.9081925	2.096412	23.25315	(exact)
5	.9047304	.6000757	1.399687	24.31435	(exact)
Crude	1.719823	1.391992	2.14353		(exact)
M-H combined	1.424682	1.154703	1.757784		

We find that the mortality incidence ratios are greatly different within age category, being highest for the youngest categories and actually dropping below 1 for the oldest. (In the last case, we might argue that those who smoke and who have not died by age 75 are self-selected to be particularly robust.)

Seeing this, we will now parameterize the smoking effects separately for each age category, although we will begin by constraining the smoking effects on age categories 3 and 4 to be equivalent:

```
. constraint 1 smokes#3.agecat = smokes#4.agecat
. poisson deaths c.smokes#agecat i.agecat, exposure(pyears) irr constraints(1)
Iteration 0:   log likelihood =  -31.95424
Iteration 1:   log likelihood = -27.796801
Iteration 2:   log likelihood = -27.574177
Iteration 3:   log likelihood = -27.572645
Iteration 4:   log likelihood = -27.572645
Poisson regression                                Number of obs   =         10
                                                  Wald chi2(8)    =     632.14
Log likelihood = -27.572645                       Prob > chi2     =     0.0000
 ( 1)  [deaths]3.agecat#c.smokes - [deaths]4.agecat#c.smokes = 0
```

deaths	IRR	Std. Err.	z	P>|z|	[95% Conf.	Interval]
agecat# c.smokes						
1	5.736637	4.181256	2.40	0.017	1.374811	23.93711
2	2.138812	.6520701	2.49	0.013	1.176691	3.887609
3	1.412229	.2017485	2.42	0.016	1.067343	1.868557
4	1.412229	.2017485	2.42	0.016	1.067343	1.868557
5	.9047304	.1855513	-0.49	0.625	.6052658	1.35236
agecat						
2	10.5631	8.067701	3.09	0.002	2.364153	47.19623
3	47.671	34.37409	5.36	0.000	11.60056	195.8978
4	98.22765	70.85012	6.36	0.000	23.89324	403.8244
5	199.2099	145.3356	7.26	0.000	47.67693	832.3648
pyears	(exposure)					

```
. estat gof
         Goodness-of-fit chi2  =  .0774185
         Prob > chi2(1)        =    0.7808
```

The goodness-of-fit χ^2 is now small; we are no longer running roughshod over the data. Let us now consider simplifying the model. The point estimate of the incidence-rate ratio for smoking in age category 1 is much larger than that for smoking in age category 2, but the confidence interval for `smokes#1.agecat` is similarly wide. Is the difference real?

```
. test smokes#1.agecat = smokes#2.agecat
 ( 1)  [deaths]1b.agecat#c.smokes - [deaths]2.agecat#c.smokes = 0
           chi2(  1) =    1.56
         Prob > chi2 =    0.2117
```

The point estimates may be far apart, but there is insufficient data, and we may be observing random differences. With that success, might we also combine the smokers in age categories 3 and 4 with those in 1 and 2?

```
. test smokes#2.agecat = smokes#3.agecat, accum
 ( 1)  [deaths]1b.agecat#c.smokes - [deaths]2.agecat#c.smokes = 0
 ( 2)  [deaths]2.agecat#c.smokes - [deaths]3.agecat#c.smokes = 0
           chi2(  2) =    4.73
         Prob > chi2 =    0.0938
```

Combining age categories 1–4 may be overdoing it—the 9.38% significance level is enough to stop us, although others may disagree.

Thus we now fit our final model:

```
. constraint 2 smokes#1.agecat = smokes#2.agecat
. poisson deaths c.smokes#agecat i.agecat, exposure(pyears) irr constraints(1/2)
Iteration 0:   log likelihood = -31.550722
Iteration 1:   log likelihood = -28.525057
Iteration 2:   log likelihood = -28.514535
Iteration 3:   log likelihood = -28.514535
Poisson regression                              Number of obs   =         10
                                                Wald chi2(7)    =     642.25
Log likelihood = -28.514535                     Prob > chi2     =     0.0000
 ( 1)  [deaths]3.agecat#c.smokes - [deaths]4.agecat#c.smokes = 0
 ( 2)  [deaths]1b.agecat#c.smokes - [deaths]2.agecat#c.smokes = 0
------------------------------------------------------------------------------
      deaths |        IRR   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     agecat#
    c.smokes |
          1  |   2.636259   .7408403     3.45   0.001     1.519791    4.572907
          2  |   2.636259   .7408403     3.45   0.001     1.519791    4.572907
          3  |   1.412229   .2017485     2.42   0.016     1.067343    1.868557
          4  |   1.412229   .2017485     2.42   0.016     1.067343    1.868557
          5  |   .9047304   .1855513    -0.49   0.625     .6052658     1.35236
             |
      agecat |
          2  |   4.294559   .8385329     7.46   0.000     2.928987    6.296797
          3  |   23.42263   7.787716     9.49   0.000     12.20738    44.94164
          4  |   48.26309   16.06939    11.64   0.000     25.13068    92.68856
          5  |   97.87965   34.30881    13.08   0.000     49.24123     194.561
      pyears |  (exposure)
------------------------------------------------------------------------------
```

The above strikes us as a fair representation of the data. The probabilities of observing the deaths seen in these data are estimated using the following `predict` command:

```
. predict p, pr(0, deaths)
. list deaths p

     +-------------------+
     | deaths          p |
     |-------------------|
  1. |     32   .6891766 |
  2. |    104   .4456625 |
  3. |    206   .5455328 |
  4. |    186   .4910622 |
  5. |    102   .5263011 |
     |-------------------|
  6. |      2    .227953 |
  7. |     12   .7981917 |
  8. |     28   .4772961 |
  9. |     28   .6227565 |
 10. |     31   .5475718 |
     +-------------------+
```

The probability $\Pr(y \leq \texttt{deaths})$ ranges from 0.23 to 0.80.

◁

Methods and formulas

All postestimation commands listed above are implemented as ado-files.

In the following, we use the same notation as in [R] **poisson**.

The equation-level scores are given by

$$\text{score}(\mathbf{x}\boldsymbol{\beta})_j = y_j - e^{\xi_j}$$

The deviance (D) and Pearson (P) goodness-of-fit statistics are given by

$$\begin{aligned}
\ln L_{\max} &= \sum_{j=1}^{n} w_j \left[-y_j \{ \ln(y_j) - 1 \} - \ln(y_j!) \right] \\
\chi^2_D &= -2\{ \ln L - \ln L_{\max} \} \\
\chi^2_P &= \sum_{j=1}^{n} \frac{w_j (y_j - e^{\xi_j})^2}{e^{\xi_j}}
\end{aligned}$$

Also see

[R] **poisson** — Poisson regression

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