Title

stata.com

intreg postestimation — Postestimation tools for intreg

Postestimation commands

The following postestimation commands are available after `intreg`:

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
* `hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
* `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`	linear, censored, and truncated predictions
`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

*`hausman` and `lrtest` are not appropriate with `svy` estimation results.

predict

Description for predict

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

Menu for predict

Statistics > Postestimation

Syntax for predict

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

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

statistic	Description
Main	
`xb`	linear prediction; the default
`stdp`	standard error of the prediction
`stdf`	standard error of the forecast
`pr(`*a*`,`*b*`)`	$\Pr(a < y_j < b)$
`e(`*a*`,`*b*`)`	$E(y_j \mid a < y_j < b)$
`ystar(`*a*`,`*b*`)`	$E(y_j^*)$, $y_j^* = \max\{a, \min(y_j, b)\}$

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

`stdf` is not allowed with `svy` estimation results.

where a and b may be numbers or variables; a missing ($a \geq .$) means $-\infty$, and b missing ($b \geq .$) means $+\infty$; see **[U] 12.2.1 Missing values**.

Options for predict

Main

`xb`, the default, calculates the linear prediction.

`stdp` calculates the standard error of the prediction, which can be thought of as the standard error of the predicted expected value or mean for the observation's covariate pattern. The standard error of the prediction is also referred to as the standard error of the fitted value.

`stdf` calculates the standard error of the forecast, which is the standard error of the point prediction for 1 observation. It is commonly referred to as the standard error of the future or forecast value. By construction, the standard errors produced by `stdf` are always larger than those produced by `stdp`; see *Methods and formulas* in [R] **regress postestimation**.

`pr(`*a*`,`*b*`)` calculates $\Pr(a < \mathbf{x}_j\boldsymbol{\beta} + \epsilon_j < b)$, the probability that $y_j|\mathbf{x}_j$ would be observed in the interval (a, b).

a and *b* may be specified as numbers or variable names; *lb* and *ub* are variable names;
`pr(20,30)` calculates $\Pr(20 < \mathbf{x}_j\boldsymbol{\beta} + \epsilon_j < 30)$;
`pr(`*lb*`,`*ub*`)` calculates $\Pr(lb < \mathbf{x}_j\boldsymbol{\beta} + \epsilon_j < ub)$; and
`pr(20,`*ub*`)` calculates $\Pr(20 < \mathbf{x}_j\boldsymbol{\beta} + \epsilon_j < ub)$.

a missing ($a \geq .$) means $-\infty$; `pr(.,30)` calculates $\Pr(-\infty < \mathbf{x}_j\boldsymbol{\beta} + \epsilon_j < 30)$;
`pr(`*lb*`,30)` calculates $\Pr(-\infty < \mathbf{x}_j\boldsymbol{\beta} + \epsilon_j < 30)$ in observations for which $lb \geq .$
and calculates $\Pr(lb < \mathbf{x}_j\boldsymbol{\beta} + \epsilon_j < 30)$ elsewhere.

b missing ($b \geq .$) means $+\infty$; `pr(20,.)` calculates $\Pr(+\infty > \mathbf{x}_j\boldsymbol{\beta} + \epsilon_j > 20)$;
`pr(20,`*ub*`)` calculates $\Pr(+\infty > \mathbf{x}_j\boldsymbol{\beta} + \epsilon_j > 20)$ in observations for which $ub \geq .$
and calculates $\Pr(20 < \mathbf{x}_j\boldsymbol{\beta} + \epsilon_j < ub)$ elsewhere.

`e(`*a*`,`*b*`)` calculates $E(\mathbf{x}_j\boldsymbol{\beta} + \epsilon_j \mid a < \mathbf{x}_j\boldsymbol{\beta} + \epsilon_j < b)$, the expected value of $y_j|\mathbf{x}_j$ conditional on $y_j|\mathbf{x}_j$ being in the interval (a, b), meaning that $y_j|\mathbf{x}_j$ is truncated.
a and *b* are specified as they are for `pr()`.

`ystar(`*a*`,`*b*`)` calculates $E(y_j^*)$, where $y_j^* = a$ if $\mathbf{x}_j\boldsymbol{\beta} + \epsilon_j \leq a$, $y_j^* = b$ if $\mathbf{x}_j\boldsymbol{\beta} + \epsilon_j \geq b$, and $y_j^* = \mathbf{x}_j\boldsymbol{\beta} + \epsilon_j$ otherwise, meaning that y_j^* is censored. *a* and *b* are specified as they are for `pr()`.

`nooffset` is relevant only if you specified `offset(`*varname*`)`. It modifies the calculations made by `predict` so that they ignore the offset variable; the linear prediction is treated as $\mathbf{x}_j\boldsymbol{\beta}$ rather than as $\mathbf{x}_j\boldsymbol{\beta} + \text{offset}_j$.

`scores` calculates equation-level score variables.

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

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

margins

Description for margins

`margins` estimates margins of response for linear predictions, probabilities, and expected values.

Menu for margins

Statistics > Postestimation

Syntax for margins

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

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

statistic	Description
`xb`	linear prediction; the default
`pr(`*a*`,`*b*`)`	$\Pr(a < y_j < b)$
`e(`*a*`,`*b*`)`	$E(y_j \mid a < y_j < b)$
`ystar(`*a*`,`*b*`)`	$E(y_j^*)$, $y_j^* = \max\{a, \min(y_j, b)\}$
`stdp`	not allowed with `margins`
`stdf`	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

stata.com

▷ Example 1: Marginal predictions

Continuing with example 1 of [R] **intreg**, we compute women's expected wages conditional on a woman's wage being higher than $5,000. To do this, we can use the `e(a,b)` option.

```
. use https://www.stata-press.com/data/r17/womenwage2
(Wages of women, fictional data)
. intreg wage1 wage2 age c.age#c.age i.nev_mar i.rural school tenure
  (output omitted)
. predict w1, e(5,.)
. summarize w1
```

Variable	Obs	Mean	Std. dev.	Min	Max
w1	488	18.02362	4.583738	8.717687	35.31161

The predicted wages range from $8,718 to $35,312.

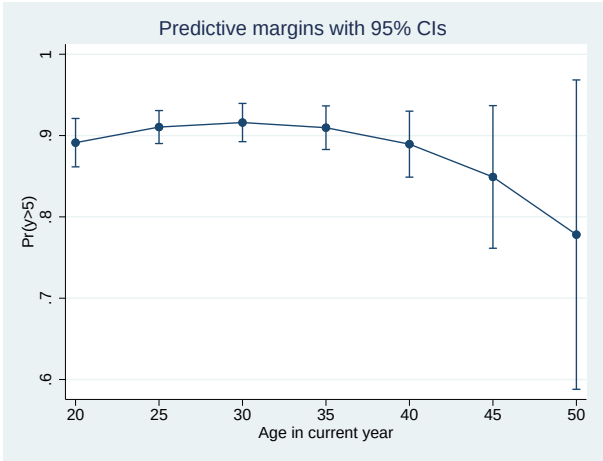
We can also examine whether the probability of earning more than $5,000 varies with age. We can use `margins` to compute the marginal means of the predicted probabilities at different ages.

```
. margins, predict(pr(5,.)) at(age=(20(5)50))
Predictive margins                                      Number of obs = 488
Model VCE: OIM
Expression: Pr(y>5), predict(pr(5,.))
1._at: age = 20
2._at: age = 25
3._at: age = 30
4._at: age = 35
5._at: age = 40
6._at: age = 45
7._at: age = 50
```

	Margin	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
_at						
1	.8912598	.0151773	58.72	0.000	.8615127	.9210068
2	.9104568	.0103467	87.99	0.000	.8901775	.930736
3	.9160005	.0120025	76.32	0.000	.892476	.9395251
4	.9096667	.0136693	66.55	0.000	.8828753	.9364581
5	.8894289	.0206992	42.97	0.000	.8488593	.9299985
6	.8491103	.0447429	18.98	0.000	.7614159	.9368048
7	.7781644	.0970557	8.02	0.000	.5879387	.9683902

We can visualize these results by using `marginsplot`:

```
. marginsplot
Variables that uniquely identify margins: age
```



The probability increases until age 30 and decreases thereafter.

Also see

[R] **intreg** — Interval regression

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