Title

stata.com

asroprobit postestimation — Postestimation tools for asroprobit

Description

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

Command	Description
`estat alternatives`	alternative summary statistics
`estat covariance`	covariance matrix of the latent-variable errors for the alternatives
`estat correlation`	correlation matrix of the latent-variable errors for the alternatives
`estat facweights`	covariance factor weights matrix
`estat mfx`	marginal effects

The following standard postestimation commands are also available:

Command	Description
`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)
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`lrtest`	likelihood-ratio test
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
`predict`	predicted probabilities, estimated linear predictor and its standard error
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

Special-interest postestimation commands

`estat alternatives` displays summary statistics about the alternatives in the estimation sample. The command also provides a mapping between the index numbers that label the covariance parameters of the model and their associated values and labels for the alternative variable.

`estat covariance` computes the estimated variance–covariance matrix of the latent-variable errors for the alternatives. The estimates are displayed, and the variance–covariance matrix is stored in `r(cov)`.

`estat correlation` computes the estimated correlation matrix of the latent-variable errors for the alternatives. The estimates are displayed, and the correlation matrix is stored in `r(cor)`.

`estat facweights` displays the covariance factor weights matrix and stores it in `r(C)`.

`estat mfx` computes marginal effects of a simulated probability of a set of ranked alternatives. The probability is stored in `r(pr)`, the matrix of rankings is stored in `r(ranks)`, and the matrix of marginal-effect statistics is stored in `r(mfx)`.

Syntax for predict

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

`predict` [*type*] { *stub*`*` | *newvarlist* } [*if*] [*in*] `, scores`

statistic	Description
Main	
`pr`	probability of each ranking, by case; the default
`pr1`	probability that each alternative is preferred
`xb`	linear prediction
`stdp`	standard error of the linear prediction

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

Menu for predict

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

`pr`, the default, calculates the probability of each ranking. For each case, one probability is computed for the ranks in `e(depvar)`.

`pr1` calculates the probability that each alternative is preferred.

`xb` calculates the linear prediction $\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_i\boldsymbol{\alpha}_j$ for alternative j and case i.

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

`altwise` specifies that alternativewise deletion be used when marking out observations due to missing values in your variables. The default is to use casewise deletion. The `xb` and `stdp` options always use alternativewise deletion.

`scores` calculates the scores for each coefficient in `e(b)`. This option requires a new variable list of length equal to the number of columns in `e(b)`. Otherwise, use the *stub*`*` option to have `predict` generate enumerated variables with prefix *stub*.

Syntax for estat

Alternative summary statistics

```
estat alternatives
```

Covariance matrix of the latent-variable errors for the alternatives

```
estat covariance [ , format(%fmt) border(bspec) left(#) ]
```

Correlation matrix of the latent-variable errors for the alternatives

```
estat correlation [ , format(%fmt) border(bspec) left(#) ]
```

Covariance factor weights matrix

```
estat facweights [ , format(%fmt) border(bspec) left(#) ]
```

Marginal effects

```
estat mfx [ if ] [ in ] [ , estat_mfx_options ]
```

estat_mfx_options	Description
Main	
`varlist(`*varlist*`)`	display marginal effects for *varlist*
`at(median [` *atlist* `])`	calculate marginal effects at these values
`rank(`*ranklist*`)`	calculate marginal effects for the simulated probability of these ranked alternatives
Options	
`level(#)`	set confidence interval level; default is `level(95)`
`nodiscrete`	treat indicator variables as continuous
`noesample`	do not restrict calculation of the medians to the estimation sample
`nowght`	ignore weights when calculating medians

Menu for estat

Statistics > Postestimation > Reports and statistics

Options for estat

Options for `estat` are presented under the following headings:

Options for estat covariance, estat correlation, and estat facweights
Options for estat mfx

Options for estat covariance, estat correlation, and estat facweights

`format(%fmt)` sets the matrix display format. The default for `estat covariance` and `estat facweights` is `format(%9.0g)`. The default for `estat correlation` is `format(%9.4f)`.

`border(bspec)` sets the matrix display border style. The default is `border(all)`. See [P] **matlist**.

`left(#)` sets the matrix display left indent. The default is `left(2)`. See [P] **matlist**.

Options for estat mfx

Main

`varlist(varlist)` specifies the variables for which to display marginal effects. The default is all variables.

`at(median` $[$ *atlist* $]$`)` specifies the values at which the marginal effects are to be calculated. *atlist* is

$[\,[\,$ *alternative*`:`*variable* `=` *#* $]\,[\,$ *variable* `=` *#* $]\,[\,\ldots\,]\,]$`)`

The marginal effects are calculated at the medians of the independent variables.

After specifying the summary statistic, you can specify specific values for variables. You can specify values for alternative-specific variables by alternative, or you can specify one value for all alternatives. You can specify only one value for case-specific variables. For example, in the `wlsrank` dataset, `female` and `score` are case-specific variables, whereas `high` and `low` are alternative-specific variables. The following would be a legal syntax for `estat mfx`:

```
. estat mfx, at(median high=0 esteem:high=1 low=0 security:low=1 female=1)
```

When `nodiscrete` is not specified, `at(median` $[$ *atlist* $]$`)` has no effect on computing marginal effects for indicator variables, which are calculated as the discrete change in the simulated probability as the indicator variable changes from 0 to 1.

The median computations respect any `if` or `in` qualifiers, so you can restrict the data over which the medians are computed. You can even restrict the values to a specific case, for example,

```
. estat mfx if case==13
```

`rank(ranklist)` specifies the ranks for the alternatives. *ranklist* is

alternative `=` *#* *alternative* `=` *#* $[\,\ldots\,]$`)`

The default is to rank the calculated latent variables. Alternatives excluded from `rank()` are omitted from the analysis. You must therefore specify at least two alternatives in `rank()`. You may have tied ranks in the rank specification. Only the order in the ranks is relevant.

Options

`level(#)` specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`; see **[U] 20.7 Specifying the width of confidence intervals**.

`nodiscrete` specifies that indicator variables be treated as continuous variables. An indicator variable is one that takes on the value 0 or 1 in the estimation sample. By default, the discrete change in the simulated probability is computed as the indicator variable changes from 0 to 1.

`noesample` specifies that the whole dataset be considered instead of only those marked in the `e(sample)` defined by the `asroprobit` command.

`nowght` specifies that weights be ignored when calculating the medians.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Predicted probabilities

After fitting an alternative-specific rank-ordered probit model, you can use `predict` to obtain the probabilities of alternative rankings or the probabilities of each alternative being preferred. When evaluating the multivariate normal probabilities via (quasi) Monte Carlo, `predict` uses the same method to generate the (quasi) random sequence of numbers as the previous call to `asroprobit`. For example, if you specified `intmethod(halton)` when fitting the model, `predict` also uses Halton sequences.

▷ Example 1

In example 1 of [R] **asroprobit**, we fit a model of job characteristic preferences. This is a study of 1957 Wisconsin high school graduates that were asked to rate their relative preference of four job characteristics: esteem, a job other people regard highly; variety, a job that is not repetitive and allows you to do a variety of things; autonomy, a job where your supervisor does not check on you frequently; and security, a job with a low risk of being laid off. The case-specific covariates are gender, `female`, an indicator variable for females, and `score`, a score on a general mental ability test measured in standard deviations. The alternative-specific variables are `high` and `low`, which indicate whether the respondent's current job is high or low in esteem, variety, autonomy, or security. This approach provides three states for a respondent's current job status for each alternative, $(1,0)$, $(0,1)$, and $(0,0)$, using the notation (`high`, `low`). The score $(1,1)$ is omitted because the respondent's current job cannot be considered both high and low in one of the job characteristics. The $(0,0)$ score would indicate that the respondent's current job does not rank high or low (is neutral) in a job characteristic. The alternatives are ranked such that 1 is the preferred alternative and 4 is the least preferred.

We can obtain the probabilities of the observed alternative rankings, the `pr` option, and the probability of each alternative being preferred, the `pr1` option, by using `predict`:

```
. use http://www.stata-press.com/data/r13/wlsrank
(1992 Wisconsin Longitudinal Study data on job values)
. asroprobit rank high low if noties, case(id) alternatives(jobchar)
> casevars(female score) reverse
  (output omitted)
. keep if e(sample)
(11244 observations deleted)
. predict prob, pr
. predict prob1, pr1
. list id jobchar prob prob1 rank female score high low in 1/12
```

	id	jobchar	prob	prob1	rank	female	score	high	low
1.	13	security	.0421807	.2784269	3	0	.3246512	0	1
2.	13	autonomy	.0421807	.1029036	1	0	.3246512	0	0
3.	13	variety	.0421807	.6026725	2	0	.3246512	1	0
4.	13	esteem	.0421807	.0160111	4	0	.3246512	0	1
5.	19	autonomy	.0942025	.1232488	4	1	.0492111	0	0
6.	19	esteem	.0942025	.0140261	3	1	.0492111	0	0
7.	19	security	.0942025	.4601368	1	1	.0492111	1	0
8.	19	variety	.0942025	.4025715	2	1	.0492111	0	0
9.	22	esteem	.1414177	.0255264	4	1	1.426412	1	0
10.	22	variety	.1414177	.4549441	1	1	1.426412	0	0
11.	22	security	.1414177	.2629494	3	1	1.426412	0	0
12.	22	autonomy	.1414177	.2566032	2	1	1.426412	1	0

The `prob` variable is constant for each case because it contains the probability of the ranking in the `rank` variable. On the other hand, the `prob1` variable contains the estimated probability of each alternative being preferred. For each case, the sum of the values in `prob1` will be approximately 1.0. They do not add up to exactly 1.0 because of approximations due to the GHK algorithm.

◁

Obtaining estimation statistics

For examples of the specialized `estat` subcommands `covariance` and `correlation`, see [R] **asmprobit postestimation**. The entry also has a good example of computing marginal effects after `asmprobit` that is applicable to `asroprobit`. Below we will elaborate further on marginal effects after `asroprobit` where we manipulate the `rank()` option.

▷ Example 2

We will continue with the preferred job characteristics example where we first compute the marginal effects for case `id` = 13.

```
. estat mfx if id==13, rank(security=3 autonomy=1 variety=2 esteem=4)

Pr(esteem=4 variety=2 autonomy=1 security=3) = .04218068
```

variable	dp/dx	Std. Err.	z	P>|z|	[	95% C.I.	]	X
high*								
esteem	-.008713	.001964	-4.44	0.000	-.012562	-.004864		0
variety	-.009102	.003127	-2.91	0.004	-.015231	-.002973		1
autonomy	.025535	.007029	3.63	0.000	.011758	.039313		0
security	-.003745	.001394	-2.69	0.007	-.006477	-.001013		0
low*								
esteem	.001614	.002646	0.61	0.542	-.003572	.0068		1
variety	.001809	.003012	0.60	0.548	-.004094	.007712		0
autonomy	-.003849	.006104	-0.63	0.528	-.015813	.008115		0
security	.000582	.000985	0.59	0.554	-.001348	.002513		1
casevars								
female*	.009767	.009064	1.08	0.281	-.007998	.027533		0
score	.008587	.004488	1.91	0.056	-.00021	.017384		.32465

(*) dp/dx is for discrete change of indicator variable from 0 to 1

Next we compute the marginal effects for the probability that autonomy is preferred given the profile of case `id` = 13.

```
. estat mfx if id==13, rank(security=2 autonomy=1 variety=2 esteem=2)

Pr(esteem=3 variety=4 autonomy=1 security=2) +
Pr(esteem=4 variety=3 autonomy=1 security=2) +
Pr(esteem=2 variety=4 autonomy=1 security=3) +
Pr(esteem=4 variety=2 autonomy=1 security=3) +
Pr(esteem=2 variety=3 autonomy=1 security=4) +
Pr(esteem=3 variety=2 autonomy=1 security=4) = .10276103
```

variable	dp/dx	Std. Err.	z	P>|z|	[	95% C.I.	]	X
high*								
esteem	-.003524	.001258	-2.80	0.005	-.005989	-.001059		0
variety	-.036203	.00894	-4.05	0.000	-.053724	-.018681		1
autonomy	.057279	.013801	4.15	0.000	.030231	.084328		0
security	-.0128	.002665	-4.80	0.000	-.018024	-.007576		0
low*								
esteem	.000518	.000833	0.62	0.534	-.001116	.002151		1
variety	.006409	.010588	0.61	0.545	-.014343	.027161		0
autonomy	-.008818	.013766	-0.64	0.522	-.035799	.018163		0
security	.002314	.003697	0.63	0.531	-.004932	.009561		1
casevars								
female*	.013839	.021607	0.64	0.522	-.028509	.056188		0
score	.017917	.011062	1.62	0.105	-.003764	.039598		.32465

(*) dp/dx is for discrete change of indicator variable from 0 to 1

The probability computed by `estat mfx` matches the probability computed by `predict, pr1` only within three digits. This outcome is because of how the computation is carried out and the numeric inaccuracy of the GHK simulator using a Hammersley point set of length 200. The computation carried out by `estat mfx` literally computes all six probabilities listed in the header of the MFX table and sums them. The computation by `predict, pr1` is the same as `predict` after `asmprobit` (multinomial probit): it computes the probability that `autonomy` is chosen, thus requiring only one

call to the GHK simulator. Hence, there is a difference in the reported values even though the two probability statements are equivalent.

◁

Stored results

`estat mfx` stores the following in `r()`:

Scalars

`r(pr)`	scalar containing the computed probability of the ranked alternatives.

Matrices

`r(ranks)`	column vector containing the alternative ranks. The rownames identify the alternatives.
`r(mfx)`	matrix containing the computed marginal effects and associated statistics. Column 1 of the matrix contains the marginal effects; column 2, their standard errors; column 3, their z statistics; and columns 4 and 5, the confidence intervals. Column 6 contains the values of the independent variables used to compute the probabilities `r(pr)`.

Also see

[R] **asroprobit** — Alternative-specific rank-ordered probit regression

[R] **asmprobit** — Alternative-specific multinomial probit regression

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