Title

stata.com

estimation options — Estimation options

Description Syntax Options Also see

Description

This entry describes the options common to many estimation commands. Not all the options documented here work with all estimation commands. See the documentation for the particular estimation command; if an option is listed there, it is applicable.

Syntax

estimation_cmd ... [, *options*]

options	Description
Model	
noconstant	suppress constant term
offset(*varname*$_o$)	include *varname*$_o$ in model with coefficient constrained to 1
exposure(*varname*$_e$)	include ln(*varname*$_e$) in model with coefficient constrained to 1
constraints(*constraints*)	apply specified linear constraints
collinear	keep collinear variables
Reporting	
level(#)	set confidence level; default is level(95)
lrmodel	perform likelihood-ratio test instead of the default Wald test
nocnsreport	do not display constraints
noci	suppress confidence intervals
nopvalues	suppress p-values and their test statistics
noomitted	do not display omitted collinear variables
vsquish	suppress blank space separating factor variables or time-series variables
noemptycells	do not display empty interaction cells of factor variables
baselevels	report base levels for factor variables and interactions
allbaselevels	display all base levels for factor variables and interactions
nofvlabel	display factor-variable level values rather than value labels
fvwrap(#)	allow # lines when wrapping long value labels
fvwrapon(*style*)	apply *style* for wrapping long value labels; *style* may be word or width
cformat(%*fmt*)	format for coefficients, standard errors, and confidence limits
pformat(%*fmt*)	format for p-values
sformat(%*fmt*)	format for test statistics
nolstretch	do not automatically widen coefficient table for long variable names
Integration	
intmethod(*intmethod*)	integration method for random-effects models
intpoints(#)	use # integration (quadrature) points
coeflegend	display legend instead of statistics

Options

Model

`noconstant` suppresses the constant term (intercept) in the model.

`offset(`*varname*$_o$`)` specifies that *varname*$_o$ be included in the model with the coefficient constrained to be 1.

`exposure(`*varname*$_e$`)` specifies a variable that reflects the amount of exposure over which the *depvar* events were observed for each observation; ln(*varname*$_e$) with coefficient constrained to be 1 is entered into the log-link function.

`constraints(`*numlist* | *matname*`)` specifies the linear constraints to be applied during estimation. The default is to perform unconstrained estimation. See [R] **reg3** for the use of constraints in multiple-equation contexts.

`constraints(`*numlist*`)` specifies the constraints by number after they have been defined by using the `constraint` command; see [R] **constraint**. Some commands (for example, `slogit`) allow only `constraints(`*numlist*`)`.

`constraints(`*matname*`)` specifies a matrix containing the constraints; see [P] **makecns**.

`constraints(`*clist*`)` is used by some estimation commands, such as `mlogit`, where *clist* has the form `#[-#] [, #[-#] ...]`.

`collinear` specifies that the estimation command not omit collinear variables. Usually, there is no reason to leave collinear variables in place, and, in fact, doing so usually causes the estimation to fail because of the matrix singularity caused by the collinearity. However, with certain models, the variables may be collinear, yet the model is fully identified because of constraints or other features of the model. In such cases, using the `collinear` option allows the estimation to take place, leaving the equations with collinear variables intact. This option is seldom used.

Reporting

`level(#)` specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`; see [U] **20.8 Specifying the width of confidence intervals**.

`lrmodel` specifies to conduct a likelihood-ratio test of the full maximum-likelihood model versus the restricted model that includes only a constant term in the regression equation instead of conducting the default Wald test that all coefficients are zero. This option can substantially increase estimation time.

`lrmodel` may not be specified with the `vce(robust)`, `vce(cluster` *clustvar*`)`, `vce(jackknife)`, `vce(bootstrap)`, or `noconstant` option.

`lrmodel` may not be combined with constraints; see `constraints(`*constraints*`)`. In some cases, a likelihood-ratio test is valid for models with constraints. To compute a likelihood-ratio test when constraints have been applied during estimation, use `lrtest`; see [R] **lrtest**.

`lrmodel` may not be specified with the `jackknife`, `bootstrap`, `svy`, or `mi estimate` prefix.

`nocnsreport` specifies that no constraints be reported. The default is to display user-specified constraints above the coefficient table.

`noci` suppresses confidence intervals from being reported in the coefficient table.

`nopvalues` suppresses *p*-values and their test statistics from being reported in the coefficient table.

`noomitted` specifies that variables that were omitted because of collinearity not be displayed. The default is to include in the table any variables omitted because of collinearity and to label them as "(omitted)".

`vsquish` specifies that the blank space separating factor-variable terms or time-series–operated variables from other variables in the model be suppressed.

`noemptycells` specifies that empty cells for interactions of factor variables not be displayed. The default is to include in the table interaction cells that do not occur in the estimation sample and to label them as "(empty)".

`baselevels` and `allbaselevels` control whether the base levels of factor variables and interactions are displayed. The default is to exclude from the table all base categories.

`baselevels` specifies that base levels be reported for factor variables and for interactions whose bases cannot be inferred from their component factor variables.

`allbaselevels` specifies that all base levels of factor variables and interactions be reported.

`nofvlabel` displays factor-variable level values rather than attached value labels. This option overrides the `fvlabel` setting; see [R] **set showbaselevels**.

`fvwrap(#)` specifies how many lines to allow when long value labels must be wrapped. Labels requiring more than # lines are truncated. This option overrides the `fvwrap` setting; see [R] **set showbaselevels**.

`fvwrapon(`*style*`)` specifies whether value labels that wrap will break at word boundaries or break based on available space.

`fvwrapon(word)`, the default, specifies that value labels break at word boundaries.

`fvwrapon(width)` specifies that value labels break based on available space.

This option overrides the `fvwrapon` setting; see [R] **set showbaselevels**.

`cformat(%`*fmt*`)` specifies how to format coefficients, standard errors, and confidence limits in the coefficient table. The maximum format width is 9. See [R] **set cformat**.

`pformat(%`*fmt*`)` specifies how to format *p*-values in the coefficient table. The maximum format width is 5. See [R] **set cformat**.

`sformat(%`*fmt*`)` specifies how to format test statistics in the coefficient table. The maximum format width is 8. See [R] **set cformat**.

`nolstretch` specifies that the width of the coefficient table not be automatically widened to accommodate longer variable names. The default, `lstretch`, is to automatically widen the coefficient table up to the width of the Results window. To change the default, use `set lstretch off`. `nolstretch` is not shown in the dialog box.

Integration

`intmethod(`*intmethod*`)` specifies the integration method to be used for the random-effects model. It accepts one of four arguments: `mvaghermite`, the default for all but a crossed random-effects model, performs mean and variance adaptive Gauss–Hermite quadrature; `mcaghermite` performs mode and curvature adaptive Gauss–Hermite quadrature; `ghermite` performs nonadaptive Gauss–Hermite quadrature; and `laplace`, the default for crossed random-effects models, performs the Laplacian approximation.

`intpoints(#)` specifies the number of integration points to use for integration by quadrature. The default is `intpoints(12)`; the maximum is `intpoints(195)`. Increasing this value improves the accuracy but also increases computation time. Computation time is roughly proportional to its value.

The following option is not shown in the dialog box:

`coeflegend` specifies that the legend of the coefficients and how to specify them in an expression be displayed rather than displaying the statistics for the coefficients.

Also see

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