

Description

The `etable` command allows you to easily create a table of estimation results and export it to a variety of file types. You can create a table complete with a title, notes, stars for indicating significant results, and more.

Quick start

Create a table from the active estimation results, reporting the coefficients, standard errors, and number of observations

```
etable
```

Same as above, and display stars for significant results and a note indicating what the stars represent

```
etable, showstars showstarsnote
```

Create a table with title “My title” and note “My note”

```
etable, title("My title") note("My note")
```

Create a table with stored estimates `model1` and `model2`, along with the number of observations and adjusted R^2 value for each model

```
etable, estimates(model1 model2) mstat(N) mstat(r2_a)
```

Same as above, and export the table to `myfile.tex`

```
etable, estimates(model1 model2) mstat(N) mstat(r2_a) export(myfile.tex)
```

Menu

Statistics > Summaries, tables, and tests > Table of estimation results

Syntax

etable [, options]

options	Description
Main	
<code>estimates(<i>namelist</i>)</code>	work with previously stored estimation results
<code>margins</code>	consume results from margins
<code>replay</code>	report table without consuming results
<code>column(<i>column_header</i>)</code>	select column header
<code>name(<i>cname</i>)</code>	work with collection <i>cname</i> ; default is name(ETable)
<code>append</code>	append to the collection
<code>replace</code>	replace the collection
Coefficients	
<code>keep(<i>coeflist</i>)</code>	report coefficients in order specified
<code>cstat(<i>cstat</i> [, <i>cstat_opts</i>])</code>	report coefficient statistic
Model	
<code>mstat(<i>mstat</i> [, <i>mstat_opts</i>])</code>	report model statistic
Equations	
<code>equations(<i>eqlist</i>)</code>	report equations in order specified
<code>eqrcode(<i>oldeq</i> = <i>neweq</i>)</code>	recode equation
<code>[no]showeq</code>	display or suppress equations
Stars	
<code>stars([<i>starspec</i>] [, <i>stars_opts</i>])</code>	customize rules for star labels
<code>[no]showstars</code>	display or suppress star labels
<code>[no]showstarsnote</code>	display or suppress note explaining star labels
Title	
<code>title(<i>string</i>)</code>	add table title
<code>titlestyles(<i>text_styles</i>)</code>	change table title styles
Notes	
<code>note(<i>string</i>)</code>	add table note
<code>notestyles(<i>text_styles</i>)</code>	change table note styles
Export	
<code>export(<i>filename.suffix</i> [, <i>export_opts</i>])</code>	export table
Options	
<code>[no]varlabel</code>	display or suppress variable names or labels
<code>[no]fvlabel</code>	display or suppress factor values or labels
<code>[no]center</code>	center or right-align item cells
<code>label(<i>filename</i> [, replace])</code>	specify the collection labels
<code>style(<i>filename</i> [, override])</code>	specify the collection style
<code>warn</code>	show collect warnings

warn does not appear in the dialog box.

<i>column_header</i>	Description
<code>depvar</code>	show dependent variable name; the default
<code>dvlabel</code>	show variable label for dependent variable
<code>command</code>	show command name
<code>title</code>	show command title
<code>estimates</code>	show estimates name
<code>index</code>	show result set index
<i>cstat_opts</i>	Description
<code>label(<i>string</i>)</code>	specify coefficient statistic label
<code>font(<i>[fontfamily]</i> [<i>, font_opts</i>])</code>	specify font style
<code>smcl(<i>smcl</i>)</code>	specify formatting for SMCL files
<code>latex(<i>latex</i>)</code>	specify L ^A T _E X macro
<code>shading(<i>sspec</i>)</code>	set background color, foreground color, and fill pattern
<code>nformat(<i>%fmt</i>)</code>	specify numeric format
<code>sformat(<i>sfmt</i>)</code>	specify string format
<code>cidelimiter(<i>char</i>)</code>	use character as delimiter for confidence interval limits
<code>cridelimiter(<i>char</i>)</code>	use character as delimiter for credible interval limits
<i>mstat_opts</i>	Description
<code>label(<i>string</i>)</code>	specify model statistic label
<code>font(<i>[fontfamily]</i> [<i>, font_opts</i>])</code>	specify font style
<code>smcl(<i>smcl</i>)</code>	specify formatting for SMCL files
<code>latex(<i>latex</i>)</code>	specify L ^A T _E X macro
<code>shading(<i>sspec</i>)</code>	set background color, foreground color, and fill pattern
<code>nformat(<i>%fmt</i>)</code>	specify numeric format
<code>sformat(<i>sfmt</i>)</code>	specify string format
<i>font_opts</i>	Description
<code>size(<i>#</i> [<i>unit</i>])</code>	specify font size
<code>color(<i>color</i>)</code>	specify font color
<code>variant(<i>variant</i>)</code>	specify font variant and capitalization
<code>[no]bold</code>	specify whether to format statistic as bold
<code>[no]italic</code>	specify whether to format statistic as italic
<code>[no]strikeout</code>	specify whether to strike out statistic
<code>underline(<i>upattern</i>)</code>	specify whether to underline statistic

<i>stars_opts</i>	Description
<code>attach(<i>cstat</i>)</code>	attach star labels to coefficient statistic <i>cstat</i>
<code>increasing</code>	compose stars note with increasing <i>p</i> -values; the default
<code>decreasing</code>	compose stars note with decreasing <i>p</i> -values
<code>pvalue(<i>string</i>)</code>	<i>p</i> -value name for stars note
<code>delimiter(<i>char</i>)</code>	use character as delimiter for labels in stars note
<code>nformat(<i>%fmt</i>)</code>	numeric format for stars note
<code>prefix(<i>string</i>)</code>	prefix for stars note
<code>suffix(<i>string</i>)</code>	suffix for stars note
<code>clear</code>	remove previous star properties

<i>text_styles</i>	Description
<code>font(<i>[fontfamily]</i> [, <i>text_opts</i>])</code>	font style
<code>smcl(<i>smcl</i>)</code>	specify formatting for SMCL files
<code>latex(<i>latex</i>)</code>	specify L ^A T _E X macro
<code>shading(<i>sspec</i>)</code>	set background color, foreground color, and fill pattern

<i>text_opts</i>	Description
<code>size(# [<i>unit</i>])</code>	specify font size
<code>color(<i>color</i>)</code>	specify font color
<code>variant(<i>variant</i>)</code>	specify font variant and capitalization
<code>[no]bold</code>	specify whether to format statistic as bold
<code>[no]italic</code>	specify whether to format statistic as italic
<code>[no]strikeout</code>	specify whether to strike out statistic
<code>[no]underline</code>	specify whether to underline statistic

<i>suffix</i>	<i>fileformat</i>	Output format
<code>docx</code>	<code>as(docx)</code>	Microsoft Word
<code>html</code>	<code>as(html)</code>	HTML 5 with CSS
<code>pdf</code>	<code>as(pdf)</code>	PDF
<code>xlsx</code>	<code>as(xlsx)</code>	Microsoft Excel 2007/2010 or newer
<code>xls</code>	<code>as(xls)</code>	Microsoft Excel 1997/2003
<code>tex</code>	<code>as(tex)</code>	L ^A T _E X
<code>smcl</code>	<code>as(smcl)</code>	SMCL
<code>txt</code>	<code>as(txt)</code>	plain text
<code>markdown</code>	<code>as(markdown)</code>	Markdown
<code>md</code>	<code>as(md)</code>	Markdown

<i>export_opts</i>	Description
<code>as(<i>fileformat</i>)</code>	specify document type
<code>replace</code>	overwrite existing file
<i>docx_options</i>	available when exporting to .docx files
<i>html_options</i>	available when exporting to .html files
<i>pdf_options</i>	available when exporting to .pdf files
<i>excel_options</i>	available when exporting to .xls and .xlsx files
<i>tex_options</i>	available when exporting to .tex files
<i>smcl_option</i>	available when exporting to .smcl files
<i>txt_option</i>	available when exporting to .txt files
<i>md_option</i>	available when exporting to .markdown and .md files

<i>docx_options</i>	Description
<code>noisily</code>	show the putdocx commands used to export to the Microsoft Word file
<code>dofile(<i>filename</i>[, replace])</code>	save the putdocx commands used for exporting to the named do-file

<i>html_options</i>	Description
<code>append</code>	append to an existing file
<code>tableonly</code>	export only the table to the specified file
<code>cssfile(<i>cssfile</i>)</code>	define the styles in <i>cssfile</i> instead of <i>filename</i>
<code>prefix(<i>prefix</i>)</code>	use <i>prefix</i> to identify style classes

<i>pdf_options</i>	Description
<code>noisily</code>	show the putpdf commands used to export to the PDF file
<code>dofile(<i>filename</i>[, replace])</code>	save the putpdf commands used for exporting to the named do-file

<i>excel_options</i>	Description
<code>noisily</code>	show the putexcel commands used to export to the Excel file
<code>dofile(<i>filename</i>[, replace])</code>	save the putexcel commands used for exporting to the named do-file
<code>sheet(<i>sheetname</i>[, replace])</code>	specify the worksheet to use; the default sheet name is Sheet1
<code>cell(<i>cell</i>)</code>	specify the Excel upper-left cell as the starting position to export the table; the default is cell (A1)
<code>modify</code>	modify Excel file
<code>noopen</code>	do not open Excel file in memory

`noopen` does not appear in the dialog box.

<i>tex_options</i>	Description
<code>append</code>	append to an existing file
<code>tableonly</code>	export only the table to the specified file

<i>smcl_option</i>	Description
append	append to an existing file
<i>txt_option</i>	Description
append	append to an existing file
<i>md_option</i>	Description
append	append to an existing file

fontfamily specifies a valid font family.

unit may be in (inch), pt (point), or cm (centimeter). An inch is equivalent to 72 points and 2.54 centimeters. The default is pt.

variant may be allcaps, smallcaps, or normal.

`variant(allcaps)` changes the text to all uppercase letters; applicable when publishing items from a collection to Microsoft Word, PDF, L^AT_EX, and HTML files.

`variant(smallcaps)` changes the text to use large capitals for uppercase letters and smaller capitals for lowercase letters; applicable when publishing items from a collection to Microsoft Word, L^AT_EX, and HTML files.

`variant(normal)` changes the font variant back to normal and leaves the capitalization unchanged from the original text; applicable when publishing items from a collection to Microsoft Word, PDF, L^AT_EX, and HTML files.

upattern may be any of the patterns listed in the [Appendix](#). For example, `underline(none)` removes the underline from the statistic, and `underline(single)` underlines the statistic. All other *upatterns* are available only when publishing items from a collection to Microsoft Word.

smcl specifies the name of the SMCL directive to render text for SMCL output. The supported SMCL directives are `input`, `error`, `result`, and `text`.

latex specifies the name of a L^AT_EX macro to render text for L^AT_EX output. Example L^AT_EX macro names are `textbf`, `textsf`, `textrm`, and `texttt`. Custom L^AT_EX macros are also allowed. If *text* is to be rendered in a cell, title, or note, then *latex* is translated to the following when you export to L^AT_EX:

`\latex {text}`

sspec is

[`background(bgcolor)` `foreground(fgcolor)` `pattern(fpattern)`]

bgcolor specifies the background color.

fgcolor specifies the foreground color.

fpattern specifies the fill pattern. A complete list of fill patterns is shown in the [Appendix](#).

bcolor, *fgcolor*, and *color* may be one of the colors listed in the [Appendix](#); a valid RGB value in the form ### # ##, for example, 171 248 103; or a valid RRGGBB hex value in the form #####, for example, ABF867.

sfmt is the specification for a string format in option `sformat()` and may contain a mix of text and `%s`. Here `%s` refers to the statistic value that is formatted as specified using `nformat()`. The text will be placed around the statistic as it is placed around `%s` in this option. For instance, to place parentheses around the statistic, you can specify `sformat("(%s")`.

Two text characters must be specified using a special character sequence if you want them to be displayed in your collection. To include `%`, type `%%`. To include `\`, type `\\`. For instance, to place a percent sign after a statistic, you can specify `sformat("%s%")`.

Options

Main

`estimates(namelist)` specifies the estimation results to be included in the table. These are the names specified with `estimates store`. By default, `etable` creates a table with the active estimation results.

`margins` creates a table with the results of the immediately preceding `margins` command.

`replay` specifies that `etable` redisplay the table without consuming results.

`column(column_header)` specifies the content to be used in the column headers. *column_header* may be `depvar`, `dlabel`, `command`, `title`, `estimates`, or `index`.

`depvar` specifies that `etable` use the dependent variable name for the column headers. This name is obtained from the `eclass` macro `e(depvar)`. Note that this macro may contain multiple names after fitting a multivariate model.

`dlabel` specifies that `etable` use the variable label for the dependent variable name for the column headers. `etable` uses the variable label associated with the variable name in the `eclass` macro `e(depvar)`. If the variable does not have a variable label, the variable name will be used. `column(dlabel)` will not be helpful when the estimation command stores multiple names in `e(depvar)`.

`command` specifies that `etable` use the command name for the column headers. This name is obtained from the `eclass` macro `e(cmd)`.

`title` specifies that `etable` use the command title for the column headers. This title is obtained from the `eclass` macro `e(title)`.

`estimates` specifies that `etable` use the name given to previously stored estimation results for the column headers; these are the names specified with `estimates store`.

`index` specifies that `etable` use the result set index for the column headers.

`name(cname)` specifies the collection from which estimation results will be obtained, instead of the current collection. The default is `name(ETable)`.

`append` specifies that `etable` append the results into the collection named in `name()`.

`replace` permits `etable` to overwrite the existing collection. This option is implied for `name(ETable)` when `append` and `replay` are not specified.

Coefficients

`keep(coeflist)` specifies the coefficients to be included in the table and the order in which to display them. Names are separated from each other by blanks. The default is to display all coefficients.

`cstat(cstat [, cstat_opts])` specifies the coefficient statistic to be included in the table. Optionally, you may specify the label and style for this statistic. `cstat()` may be repeated to request multiple coefficient statistics.

The default is to display the coefficients (`_r_b`) and their standard errors (`_r_se`), both formatted to three decimals. Standard errors are enclosed in parentheses.

Available coefficient statistics are

<i>cstat</i>	Description
<code>_r_b</code>	coefficients reported by estimation
<code>_r_se</code>	standard errors of <code>_r_b</code>
<code>_r_z</code>	test statistics for <code>_r_b</code>
<code>_r_z_abs</code>	absolute values of <code>_r_z</code>
<code>_r_p</code>	<i>p</i> -values for <code>_r_b</code>
<code>_r_lb</code>	lower bounds of confidence intervals (CIs) for <code>_r_b</code>
<code>_r_ub</code>	upper bounds of CIs for <code>_r_b</code>
<code>_r_ci</code>	CIs for <code>_r_b</code>
<code>_r_crlb</code>	lower bounds of credible intervals for <code>_r_b</code>
<code>_r_crub</code>	upper bounds of credible intervals for <code>_r_b</code>
<code>_r_cri</code>	credible intervals of Bayesian estimates
<code>_hide</code>	hide coefficient statistics

cstat_opts are `label(string)`, `font([fontfamily] [, font_opts])`, `smcl(smcl)`, `latex(latex)`, `shading(string)`, `nformat(%fmt)`, `sformat(sfmt)`, `cidelimiter(char)`, and `cridelimiter(char)`.

`label(string)` is used to modify the label for the specified coefficient statistic.

`font([fontfamily] [, size(# [unit]) color(color) variant(variant) [no]bold [no]italic [no]strikeout [no]underline underline(upattern)])` specifies the font style for the coefficient statistic. These font style properties are applicable when exporting the table to Microsoft Word, Microsoft Excel, PDF, HTML, and L^AT_EX files, unless otherwise specified.

fontfamily specifies a valid font family. This font style property is applicable when publishing items from a collection to Microsoft Word, Microsoft Excel, PDF, and HTML files.

`size(# [unit])` specifies the font size as a number optionally followed by units. This font style property is applicable when publishing items from a collection to Microsoft Word, Microsoft Excel, PDF, and HTML files.

`color(color)` specifies the text color.

`variant(variant)` specifies the font variant and capitalization.

`bold` and `nobold` specify the font weight. `bold` changes the font weight to bold; `nobold` changes the font weight back to normal.

`italic` and `noitalic` specify the font style. `italic` changes the font style to italic; `noitalic` changes the font style back to normal.

`strikeout` and `nostrikeout` specify whether to add a strikeout mark to the coefficient statistic. `strikeout` adds a strikeout mark to the statistic; `nostrikeout` changes the statistic back to normal.

`underline(upattern)`, `underline`, and `nounderline` specify how to underline the coefficient statistic.

Only one of `strikeout` or `underline` is allowed when publishing to HTML files.

`smcl(smcl)` specifies how to render the statistic value for SMCL output. This style property is applicable only when publishing items from a collection to a SMCL file.

`latex(latex)` specifies how to render the statistic value for L^AT_EX output. This style property is applicable only when publishing items from a collection to a L^AT_EX file.

`shading(sspec)` sets the background color, foreground color, and fill pattern. The background color is applicable when exporting the table to Microsoft Word, Microsoft Excel, PDF, HTML, and L^AT_EX files. The foreground color and fill pattern are applicable when exporting the table to Microsoft Word and Microsoft Excel.

`nformat(%fmt)` applies the Stata numeric format *%fmt* to the coefficient statistic.

`sformat(sfmt)` applies a string format to the coefficient statistic.

`cidelimiter(char)` changes the delimiter between confidence interval limits for coefficient statistic `_r_ci`. The default is `cidelimiter(" ")`.

`cridelimiter(char)` changes the delimiter between credible interval limits for coefficient statistic `_r_cri`. The default is `cridelimiter(" ")`.

Model

`mstat(mstat [, mstat_opts])` specifies the model statistics to be included in the table.

mstat may be a *result identifier* or a *named expression*. `mstat()` may be repeated to request multiple model statistics. The default is to display the number of observations, with zero decimal digits.

result identifier is one of the following:

Identifier	Result
N	number of observations
aic	Akaike's information criteria
bic	Schwarz's Bayesian information criteria
F	<i>F</i> statistic
chi2	χ^2
ll	log likelihood of fitted model
r2	R^2
r2_a	adjusted R^2
rank	rank of fitted model
escalar	any <code>e()</code> scalar, by default
rscalar	any <code>r()</code> scalar with option margins
_hide	hide model statistics

named expression is specified as *name* = *exp*, where *name* may be any valid Stata name and *exp* is a scalar expression, typically an expression that involves one or more scalars in `e()`. For example, `aic=(-2*e(ll) + 2*e(rank))`.

With option `margin`s, or if you ran a postestimation command like `estat, exp` may also involve one or more scalars in `r()`. For example, suppose you fit a logistic model and also called `estat gof`, then you can show its goodness-of-fit statistic with `chi2_gof=r(chi2)`.

`mstat_opts` are `label(string)`, `font([fontfamily] [, font_opts])`, `smcl(smcl)`, `latex(latex)`, `shading(sspec)`, `nformat(%fmt)`, and `sformat(sfmt)`.

`label(string)` is used to modify the label for the specified model statistic.

`font([fontfamily] [, size(# [unit]) color(color) variant(variant) [no]bold [no]italic [no]strikeout [no]underline underline(upattern)])` specifies the font style for the model statistic. These font style properties are applicable when exporting the table to Microsoft Word, Microsoft Excel, PDF, HTML, and L^AT_EX files, unless otherwise specified.

`fontfamily` specifies a valid font family. This font style property is applicable when publishing items from a collection to Microsoft Word, Microsoft Excel, PDF, and HTML files.

`size(# [unit])` specifies the font size as a number optionally followed by units. This font style property is applicable when publishing items from a collection to Microsoft Word, Microsoft Excel, PDF, and HTML files.

`color(color)` specifies the text color.

`variant(variant)` specifies the font variant and capitalization.

`bold` and `nobold` specify the font weight. `bold` changes the font weight to bold; `nobold` changes the font weight back to normal.

`italic` and `noitalic` specify the font style. `italic` changes the font style to italic; `noitalic` changes the font style back to normal.

`strikeout` and `nostrikeout` specify whether to add a strikeout mark to the model statistic. `strikeout` adds a strikeout mark to the statistic; `nostrikeout` changes the text back to normal.

`underline(upattern)`, `underline`, and `nounderline` specify how to underline the model statistic.

Only one of `strikeout` or `underline` is allowed when publishing to HTML files.

`smcl(smcl)` specifies how to render the statistic value for SMCL output. This style property is applicable only when publishing items from a collection to a SMCL file.

`latex(latex)` specifies how to render the statistic value for L^AT_EX output. This style property is applicable only when publishing items from a collection to a L^AT_EX file.

`shading(sspec)` sets the background color, foreground color, and fill pattern.

`nformat(%fmt)` applies the Stata numeric format `%fmt` to the model statistic.

`sformat(sfmt)` applies a string format to the model statistic.

Equations

`equations(eqlist)` specifies the equations to be included in the table and the order in which they are reported.

`eqrcode(oldeq = neweq)` changes the equation name from `oldeq` to `neweq`. `eqrcode()` may be repeated to recode multiple equations.

`showeq` and `nshoweq` specify whether equations should be displayed. `showeq` displays the equations; `nshoweq` suppresses the equations.

Stars

`stars([#1 "label1" [#2 "label2" [#3 "label3" [#4 "label4" [#5 "label5"]]]])`
`[, stars_opts]` manages the display of stars for indicating the significance of results.

The default is `stars(.05 "*" .01 "**", attach(_r_b))`, which will display one star (*) for p -values less than 0.05 and two stars (**) for p -values less than 0.01; the stars will be attached to the coefficients (`_r_b`).

`stars_opts` are `attach(cstat)`, `increasing`, `decreasing`, `pvname(string)`, `delimiter(char)`, `nformat(%fmt)`, `prefix(string)`, `suffix(string)`, and `clear`.

`attach(cstat)` specifies that the star labels be attached to coefficient statistic *cstat*. The default is `attach(_r_b)`.

`increasing` and `decreasing` control the order of p -values in the stars note.

`increasing` specifies that the stars note be composed with increasing p -values. This is the default.

`decreasing` specifies that the stars note be composed with decreasing p -values.

`pvname(string)` specifies the name for the p -value in the stars note that is displayed with `showstarsnote`. The default is `pvname(p)`.

`delimiter(char)` changes the delimiter between labels in the stars note. The default is `cridelimiter(",")`.

`nformat(%fmt)` specifies the numeric format for the numbers displayed in the stars note. The default is `nformat(%9.0g)`.

`prefix(string)` adds a prefix to the stars note.

`suffix(string)` adds a suffix to the stars note.

`clear` removes existing star properties.

`showstars` and `nshowstars` specify whether star labels should be displayed. `showstars` displays star labels; `nshowstars` suppresses the star labels.

`showstarsnote` and `nshowstarsnote` specify whether to display the note explaining what the star labels represent. `showstarsnote` displays the note; `nshowstarsnote` suppresses the note.

`showstarsnote` is ignored if `nshowstars` is in effect.

Title

`title(string)` adds the text *string* as a title to the table.

`titlestyles(text_styles)` changes the style for the table title. *text_styles* are the following:

`font([fontfamily] [, size(# [unit]) color(color) variant(variant) [no]bold [no]italic [no]strikeout [no]underline])` specifies the font style. These font style properties are applicable when exporting the table to Microsoft Word, Microsoft Excel, PDF, HTML, and L^AT_EX files, unless otherwise specified.

fontfamily specifies a valid font family. This font style property is applicable when publishing items from a collection to Microsoft Word, Microsoft Excel, PDF, and HTML files.

`size(# [unit])` specifies the font size as a number optionally followed by units. This font style property is applicable when publishing items from a collection to Microsoft Word, Microsoft Excel, PDF, and HTML files.

`color(color)` specifies the text color.

`variant(variant)` specifies the font variant and capitalization.

`bold` and `nobold` specify the font weight. `bold` changes the font weight to bold; `nobold` changes the font weight back to normal.

`italic` and `noitalic` specify the font style. `italic` changes the font style to italic; `noitalic` changes the font style back to normal.

`strikeout` and `nostrikeout` specify whether to add a strikeout mark to the title. `strikeout` adds a strikeout mark to the title; `nostrikeout` changes the title back to normal.

`underline` and `nounderline` specify whether to underline the table title. `underline` adds a single line under the title; `nounderline` removes the underline.

Only one of `strikeout` or `underline` is allowed when publishing to HTML files.

`smcl(smcl)` specifies how to render the table title for SMCL output. This style property is applicable only when publishing items from a collection to a SMCL file.

`latex(latex)` specifies how to render the table title for L^AT_EX output. This style property is applicable only when publishing items from a collection to a L^AT_EX file.

`shading(sspec)` sets the background color, foreground color, and fill pattern. The background color is applicable when exporting the table to Microsoft Word, Microsoft Excel, PDF, HTML, and L^AT_EX files. The foreground color and fill pattern are applicable when exporting the table to Microsoft Word and Microsoft Excel.

Notes

`note(string)` adds the text *string* as a note to the table. `note()` may be specified multiple times to add multiple notes. Each note is placed on a new line.

`notestyles(text_styles)` changes the style for the table notes. *text_styles* are the following:

`font([fontfamily] [, size(# [unit]) color(color) variant(variant) [no]bold [no]italic [no]strikeout [no]underline])` specifies the font style. These font style properties are applicable when exporting the table to Microsoft Word, Microsoft Excel, PDF, HTML, and L^AT_EX files, unless otherwise specified.

fontfamily specifies a valid font family. This font style property is applicable when publishing items from a collection to Microsoft Word, Microsoft Excel, PDF, and HTML files.

`size(# [unit])` specifies the font size as a number optionally followed by units. This font style property is applicable when publishing items from a collection to Microsoft Word, Microsoft Excel, PDF, and HTML files.

`color(color)` specifies the text color.

`variant(variant)` specifies the font variant and capitalization.

`bold` and `nobold` specify the font weight. `bold` changes the font weight to bold; `nobold` changes the font weight back to normal.

`italic` and `noitalic` specify the font style. `italic` changes the font style to italic; `noitalic` changes the font style back to normal.

`strikeout` and `nostrikeout` specify whether to add a strikeout mark to the notes. `strikeout` adds a strikeout mark to the note; `nostrikeout` changes the note back to normal.

`underline` and `nounderline` specify whether to underline the table notes. `underline` adds a single line under the notes; `nounderline` removes the underline.

Only one of `strikeout` or `underline` is allowed when publishing to HTML files.

`smcl` (*smcl*) specifies how to render the table notes for SMCL output. This style property is applicable only when publishing items from a collection to a SMCL file.

`latex` (*latex*) specifies how to render the table notes for L^AT_EX output. This style property is applicable only when publishing items from a collection to a L^AT_EX file.

`shading` (*spec*) sets the background color, foreground color, and fill pattern. The background color is applicable when exporting the table to Microsoft Word, Microsoft Excel, PDF, HTML, and L^AT_EX files. The foreground color and fill pattern are applicable when exporting the table to Microsoft Word and Microsoft Excel.

Export

`export` (*filename*.*suffix*[, *export_opts*]) exports the table to the specified file. *export_opts* are the following:

`as` (*fileformat*) specifies the file format to which the table is to be exported. This option is rarely specified because, by default, `etable` determines the format from the suffix of the file being created.

`replace` permits `etable` to overwrite an existing file.

`noisily` specifies that `etable` show the commands used to export the table to Microsoft Word, Microsoft Excel, and PDF files. The `putdocx`, `putexcel`, or `putpdf` command used to export the table will be displayed.

`dofile` (*filename*[, `replace`]) specifies that `etable` save to *filename* the commands used to export the table to Microsoft Word, Microsoft Excel, and PDF files.

If *filename* already exists, it can be overwritten by specifying `replace`. If *filename* is specified without an extension, `.do` is assumed.

`append` specifies that `etable` append the table to an existing file.

This option is applicable when you export the table to an HTML, a L^AT_EX, a SMCL, a `txt`, or a Markdown file. When you export to HTML and L^AT_EX files, the `append` option implies the `tableonly` option. Furthermore, when you export to HTML files, if the target CSS file already exists, `etable` will also append to it.

`tableonly` specifies that only the table be exported to the specified HTML or L^AT_EX document. With this option, the produced file may be included in other HTML or L^AT_EX documents. By default, `etable` produces complete HTML and L^AT_EX documents.

When you export to an HTML file, if the `cssfile()` option is not specified, a CSS filename is constructed from *filename*, with the extension replaced with `.css`.

`cssfile(cssfile)` specifies that etable define the styles in *cssfile* instead of *filename* when you export to HTML.

`prefix(prefix)` specifies that etable use *prefix* to identify style classes when you export to HTML.

`sheet(sheetname [, replace])` saves to the worksheet named *sheetname*. For more information about this option, see [RPT] [putexcel](#).

`cell(cell)` specifies an Excel upper-left cell as the starting position to publish the table. The default is cell(A1).

`modify` permits `putexcel` set to modify an Excel file. For more information about this option, see [RPT] [putexcel](#).

`noopen` prevents `putexcel` from opening the Excel file in memory for modification. It does not appear in the dialog box. For more information about this option, see [RPT] [putexcel](#). This option is necessary only when you need to force etable to produce do-files as it did when etable was first introduced in Stata 17.

Options

`varlabel` and `novarlabel` specify whether variable labels should be displayed. `varlabel` displays variable labels; `novarlabel` displays variable names.

`fvlabel` and `nofvlabel` specify whether value labels should be displayed. `fvlabel` displays value labels; `nofvlabel` displays the values of the factor variable.

`center` and `nocenter` specify how item cells are horizontally aligned. `center` specifies that item cells are centered; `nocenter` specifies that item cells are right-aligned.

`label(filename [, replace])` specifies the *filename* containing the collection labels to use for your table. Labels in *filename* will be loaded for the table, and default labels will be used for any labels not specified in *filename*.

If you prefer to replace the labels used by etable with those specified in *filename*, specify `replace`. The etable labels will be discarded, and only the labels in *filename* will be applied.

`style(filename [, override])` specifies the *filename* containing the collection styles to use for your table. This might be a style you saved with [collect style save](#) or a [predefined style](#) shipped with Stata. The etable collection styles will be discarded, and only the collection styles in *filename* will be applied. Note that the layout specification saved in *filename* will not be applied; etable will always use its predefined layout.

If you prefer the etable collection styles but also want to apply any styles in *filename*, specify `override`. If there are conflicts between the default collection styles and those in *filename*, the ones in *filename* will take precedence.

The default is to use only the collection styles set in `c(etable_style)`; see [TABLES] [set etable_style](#).

The following option is available with etable but is not shown in the dialog box:

`warn` specifies that etable display warnings from `collect`. By default, these warnings are suppressed.

Remarks and examples

Remarks are presented under the following headings:

Introduction

A first example

Table comparing regression results

Multiple-equation models

Introduction

`etable` allows you to easily create a table of estimation results and export it to a variety of file types, without any knowledge of the collection system. You can make a standard estimation table with the active estimation results, results from a `margins` command, or with stored estimates. You can also customize the table by formatting the results, adding model statistics and coefficient-specific statistics, labeling statistically significant results, adding a title and notes, and more.

In most cases, you will use `etable` to easily create a table and export it to another format. However, you can customize the table beyond the options that are available with `etable`. When you issue an `etable` command, the results are stored in a collection called `ETable`. This collection is replaced with each new `etable` command, unless you specify the `append` or `replay` option. You can make additional changes to the collection with the `collect` suite of commands. To learn more about the `collect` commands, see [TABLES] [Intro](#) and the entries discussed therein.

A first example

In its simplest specification, you type `etable` after fitting a model, and you get a table with coefficients, standard errors, and the number of observations. For example, below, we use data from the Second National Health and Nutrition Examination Survey (NHANES II) (McDowell et al. 1981). We fit a simple model for systolic blood pressure and then create our table of estimation results:

```
. use https://www.stata-press.com/data/r19/nhanes2l
(Second National Health and Nutrition Examination Survey)
. quietly: regress bpsystol age weight i.region
. etable
```

	bpsystol
Age (years)	0.638 (0.011)
Weight (kg)	0.407 (0.012)
Region	
MW	-0.240 (0.564)
S	-0.619 (0.560)
W	-0.862 (0.570)
Intercept	71.708 (1.108)
Number of observations	10351

You can also include statistics that pertain to the coefficients, such as test statistics and confidence intervals, and model statistics, such as the R^2 value and the F statistic. You can look at the `cstat()` and `mstat()` options for additional statistics.

Additionally, you can complete your table with a title, notes, and labels for significant results. For example, below, we add a title to our table, and we display stars for statistically significant results:

```
. etable, title(Model for systolic blood pressure) showstars showstarsnote
Model for systolic blood pressure
```

	bpsystol
Age (years)	0.638 ** (0.011)
Weight (kg)	0.407 ** (0.012)
Region	
MW	-0.240 (0.564)
S	-0.619 (0.560)
W	-0.862 (0.570)
Intercept	71.708 ** (1.108)
Number of observations	10351

```
** p<.01, * p<.05
```

The `showstars` option displays stars next to the coefficients that are significant either at the 1% or 5% levels, and `showstarsnote` adds the note we see at the bottom, explaining what the stars represent. You can look at the `stars()` option to create your own rules for displaying stars or to specify your own labels for significance.

Suppose we have finalized our table and we are ready to export it to another format. Below, we export our table to the file `mytable.html`:

```
. etable, title(Model for systolic blood pressure)
> showstars showstarsnote export(mytable.html)
Model for systolic blood pressure
```

	bpsystol
Age (years)	0.638 ** (0.011)
Weight (kg)	0.407 ** (0.012)
Region	
MW	-0.240 (0.564)
S	-0.619 (0.560)
W	-0.862 (0.570)
Intercept	71.708 ** (1.108)
Number of observations	10351

```
** p<.01, * p<.05
(collection ETable exported to file mytable.html)
```

We could also export this table to a Microsoft Word, Microsoft Excel, \LaTeX , Markdown, SMCL, PDF, or plain text file by specifying the appropriate file extension.

Table comparing regression results

If your goal is to create a table comparing regression results, you can store the results from each model with `estimates store` and then specify which of those models you want to include in your table with the `estimates()` option.

For example, below, we fit two different models for systolic blood pressure and store them under the names `model1` and `model2`.

```
. quietly: regress bpsystol i.sex weight
. estimates store model1
. quietly: regress bpsystol i.sex i.agegrp weight
. estimates store model2
```

To include results from both of these models in our table, we specify `estimates(model1 model2)`; the models are presented in the order we list them. Additionally, we report the number of observations and the R^2 adjusted for degrees of freedom.

```
. etable, estimates(model1 model2) mstat(N) mstat(r2_a)
```

	bpsystol bpsystol	
Sex		
Female	1.420 (0.475)	1.041 (0.415)
Weight (kg)	0.452 (0.015)	0.436 (0.014)
Age group		
30-39		1.195 (0.633)
40-49		7.252 (0.684)
50-59		15.942 (0.681)
60-69		22.839 (0.546)
70+		30.466 (0.741)
Intercept	97.634 (1.246)	86.710 (1.116)
Number of observations	10351	10351
Adjusted R-squared	0.08	0.30

We would like to make a few changes to finalize this table. First, because both models have the same dependent variable, we want to display the index of result sets instead of the variable name. Second, we add stars for significance and a note explaining what the stars represent. Third, instead of reporting standard errors, we want to report confidence intervals (`_r_ci`). We format the intervals with one decimal place and use a comma as the delimiter. `etable` will automatically report coefficients, unless you specify `cstat()`, in which case it will report only the coefficient statistics you specify; therefore, we add `cstat(_r_b)`. Finally, we add a title to our table:

```
. etable, estimates(model1 model2) mstat(N) mstat(r2_a) column(index)
> showstars showstarsnote cstat(_r_b)
> title("Models for systolic blood pressure")
> cstat(_r_ci, nformat(%6.1f) cidelimiter(","))
Models for systolic blood pressure
```

	1	2
Sex		
Female	1.420 **	1.041 *
	[0.5, 2.4]	[0.2, 1.9]
Weight (kg)	0.452 **	0.436 **
	[0.4, 0.5]	[0.4, 0.5]
Age group		
30-39		1.195
		[-0.0, 2.4]
40-49		7.252 **
		[5.9, 8.6]
50-59		15.942 **
		[14.6, 17.3]
60-69		22.839 **
		[21.8, 23.9]
70+		30.466 **
		[29.0, 31.9]
Intercept	97.634 **	86.710 **
	[95.2, 100.1]	[84.5, 88.9]
Number of observations	10351	10351
Adjusted R-squared	0.08	0.30

** p<.01, * p<.05

Now our table is complete.

An alternative way to create this table is to build up the estimation table with the `append` option. For example, we can fit the first model and create the table. Then, after fitting the second model, we append the results as follows:

```
. quietly: regress bpsystol i.sex weight
. etable
. quietly: regress bpsystol i.sex i.agegrp weight
. etable, append mstat(N) mstat(r2_a) column(index)
> showstars showstarsnote cstat(_r_b)
> title("Models for systolic blood pressure")
> cstat(_r_ci, nformat(%6.1f) cidelimiter(","))
(output omitted)
```

Multiple-equation models

When you work with multiple-equation models, there is an additional option that will prove useful when creating the table of estimation results. For example, below, we fit a multivariate regression with `mvreg`:

```
. mvreg bpsystol bpdiastr = age weight
```

Equation	Obs	Parms	RMSE	"R-sq"	F	P>F
bpsystol	10,351	3	19.48051	0.3031	2250	0.0000
bpdiastr	10,351	3	11.51474	0.2067	1348.469	0.0000

	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
bpsystol						
age	.6379892	.0111315	57.31	0.000	.6161692	.6598091
weight	.4069041	.0124786	32.61	0.000	.3824435	.4313646
_cons	71.27096	1.041742	68.42	0.000	69.22894	73.31297
bpdiastr						
age	.187733	.0065797	28.53	0.000	.1748355	.2006306
weight	.3116502	.007376	42.25	0.000	.2971918	.3261086
_cons	50.37585	.615764	81.81	0.000	49.16884	51.58287

Next, we create our table of estimation results:

```
. etable
```

	bpsystol	bpdiastr
Age (years)	0.638	
	(0.011)	
Weight (kg)	0.407	
	(0.012)	
Intercept	71.271	
	(1.042)	
Age (years)	0.188	
	(0.007)	
Weight (kg)	0.312	
	(0.007)	
Intercept	50.376	
	(0.616)	
Number of observations	10351	

The results for both models are placed in a single column, so below we add the `showeq` option to display the equation names (`bpsystol` and `bpdiaast`). This will help us identify which results correspond to each model. Note that there are two dependent variables in this model, and both variable names are displayed in the column header. These names are collected from the returned result `e(depvar)`. Instead of displaying both names, we will display the index of result sets we have collected for our table by typing `command(index)`.

```
. etable, showeq column(index)
```

	1
Systolic blood pressure	
Age (years)	0.638 (0.011)
Weight (kg)	0.407 (0.012)
Intercept	71.271 (1.042)
Diastolic blood pressure	
Age (years)	0.188 (0.007)
Weight (kg)	0.312 (0.007)
Intercept	50.376 (0.616)
Number of observations	10351

Appendix

Colors

bgcolor, *fgcolor*, and *color*

aliceblue	darkslategray	lightsalmon	palevioletred
antiquewhite	darkturquoise	lightseagreen	papayawhip
aqua	darkviolet	lightskyblue	peachpuff
aquamarine	deeppink	lightslategray	peru
azure	deepskyblue	lightsteelblue	pink
beige	dimgray	lightyellow	plum
bisque	dodgerblue	lime	powderblue
black	firebrick	limegreen	purple
blanchedalmond	floralwhite	linen	red
blue	forestgreen	magenta	rosybrown
blueviolet	fuchsia	maroon	royalblue
brown	gainsboro	mediumaquamarine	saddlebrown
burlywood	ghostwhite	mediumblue	salmon
cadetblue	gold	mediumorchid	sandybrown
chartreuse	goldenrod	mediumpurple	seagreen
chocolate	gray	mediumseagreen	seashell
coral	green	mediumslateblue	sienna
cornflowerblue	greenyellow	mediumspringgreen	silver
cornsilk	honeydew	mediumturquoise	skyblue
crimson	hotpink	mediumvioletred	slateblue
cyan	indianred	midnightblue	slategray
darkblue	indigo	mintcream	snow
darkcyan	ivory	mistyrose	springgreen
darkgoldenrod	khaki	moccasin	steelblue
darkgray	lavender	navajowhite	tan
darkgreen	lavenderblush	navy	teal
darkkhaki	lawngreen	oldlace	thistle
darkmagenta	lemonchiffon	olive	tomato
darkolivegreen	lightblue	olivedrab	turquoise
darkorange	lightcoral	orange	violet
darkorchid	lightcyan	orangered	wheat
darkred	lightgoldenrodyellow	orchid	white
darksalmon	lightgray	palegoldenrod	whitesmoke
darkseagreen	lightgreen	palegreen	yellow
darkslateblue	lightpink	paleturquoise	yellowgreen

Underline patterns

upattern

none	dashLong
single	dashLongHeavy
words	dotDash
double	dashDotHeavy
thick	dotDotDash
dotted	dashDotDotHeavy
dottedHeavy	wave
dash	wavyHeavy
dashedHeavy	wavyDouble

Shading patterns

fpattern

nil	pct20
clear	pct25
solid	pct30
horzStripe	pct35
vertStripe	pct37
reverseDiagStripe	pct40
diagStripe	pct45
horzCross	pct50
diagCross	pct55
thinHorzStripe	pct60
thinVertStripe	pct62
thinReverseDiagStripe	pct65
thinDiagStripe	pct70
thinHorzCross	pct75
thinDiagCross	pct80
pct5	pct85
pct10	pct87
pct12	pct90
pct15	pct95

Acknowledgments

We thank Roger Newson of the King's College London, UK, Ben Jann of the Institute of Sociology at the University of Bern, Switzerland, and John Luke Gallup of Portland State University for writing similar commands for exporting tables of estimation results.

References

- McDowell, A., A. Engel, J. T. Massey, and K. Maurer. 1981. “Plan and operation of the Second National Health and Nutrition Examination Survey, 1976–1980”. In *Vital and Health Statistics*, ser. 1, no. 15. Hyattsville, MD: National Center for Health Statistics.
- Mitchell, M. N. 2025. *Create and Export Tables Using Stata*. College Station, TX: Stata Press.

Also see

- [\[R\] table intro](#) — Introduction to tables of frequencies, summaries, and command results
- [\[TABLES\] Intro](#) — Introduction

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow and NetCourseNow are trademarks of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.

For suggested citations, see the FAQ on [citing Stata documentation](#).

