

Title

`estout` — Making regression tables from stored estimates

Table of contents

[Syntax](#)
[Description](#)
[Options](#)
[Examples](#)
[Remarks](#)
[Backmatter](#)

Syntax

```
estout [ namelist ] [ using filename ] [ , options ]
```

where `namelist` is `_all` | * | `name` [`name` ...]
`name` is `identifier` | .

<i>options</i>	Description
<hr/>	
<u>Parameter statistics</u>	
<code>cells(<i>elements and subopts</i>)</code>	contents of the table cells, where an <i>element's subopts</i> are in parentheses, i.e. <code>element[(<i>subopts</i>)]</code>
<i>elements:</i>	
<code>b</code>	raw coefficients
<code>se</code>	standard errors
<code>t</code>	t-statistics
<code>p</code>	p-values
<code>ci</code>	confidence intervals
<code>ci_l</code>	lower bounds of confidence intervals
<code>ci_u</code>	upper bounds of confidence intervals
<code>myel</code>	results from <code>e(myel)</code>
<i>subopts (for each element):</i>	
<code>[no]star</code>	attach "significance stars"
<code>fmt(<i>fmt</i> [<i>fmt</i> ...])</code>	set the display format(s)
<code>label(<i>string</i>)</code>	define a label for <i>element</i>
<code>par[(<i>l r</i>)] nopar</code>	place results in parentheses
<code>keep(<i>keeplist</i>)</code>	drop certain individual results
<code>drop(<i>droplist</i>)</code>	keep certain individual results
<code>pattern(<i>pattern</i>)</code>	model selection
<code>[no]abs</code>	use absolute t-statistics

drop (<i>droplist</i>)	drop individual parameters
keep (<i>keeplist</i>)	keep individual parameters
order (<i>orderlist</i>)	change order of parameters
indicate (<i>groups</i> [, <i>subopt</i>])	indicate presence of parameters
<i>subopt</i> : labels (<i>yes no</i>)	redefine "Yes" and "No" labels
equations (<i>eqmatchlist</i>)	match the models' equations
eform [(<i>pattern</i>)] noeform	report exponentiated coefficients
transform (<i>list</i> [, <i>subopt</i>])	apply transformations to coefficients
<i>subopt</i> : pattern (<i>pattern</i>)	select models
margin [(<i>u c p</i>)] nomargin	report marginal effects/elasticities
discrete (<i>string</i>) nodiscrete	identify 0/1 variables (if margin)
meqs (<i>eq_list</i>)	select equations for marginal effects
level (#)	set level for confidence intervals

Summary statistics

stats (<i>scalarlist</i> [, <i>subopts</i>])	display summary statistics at the bottom of the table
<i>subopts</i> :	
fmt (<i>fmt</i> [<i>fmt</i> ...])	set the display formats
labels (<i>strlist</i> [, <i>label_subopts</i>])	label the summary statistics
star [(<i>sca'list</i>)] nostar	denote the model significance

Significance stars

starlevels (<i>levelslist</i>)	define thresholds and symbols, where where ' <i>levelslist</i> ' is ' <i>symbol #</i> [<i>symbol # ...</i>]' with # in (0,1] and listed in descending order
[no]stardetach	display the stars in their own column

Layout

varwidth (#)	set width of the table's left stub
modelwidth (#)	set width of the results columns
[no]abbrev	abbreviate long names and labels
[no]unstack	place equations from multiple- equation models in separate columns
begin (<i>string</i>)	specify the beginning of the rows
delimiter (<i>string</i>)	specify the column delimiter
end (<i>string</i>)	specify the ending of the table rows
dmarker (<i>string</i>)	define the decimal marker
msign (<i>string</i>)	define the minus sign
[no]lz	print the leading zero of fixed format numbers in (-1,1)
extracols (<i>numlist</i>)	add empty column to the table
substitute (<i>subst</i>)	apply end-of-pipe substitutions, where ' <i>subst</i> ' is ' <i>from to [from to ...]</i> '

Labeling

title (string)	specify a title for the table
[no] legend	add a significance symbols legend
prehead (strlist)	add text before the table heading
posthead (strlist)	add text after the table heading
prefoot (strlist)	add text before the table footer
postfoot (strlist)	add text after the table footer
hlinechar (string)	specify look of @hline
[no] label	make use of variable labels
varlabels (matchlist[, sub.])	relabel the parameters
subopts:	
blist (matchlist)	assign prefixes to certain rows
elist (matchlist)	assign suffixes to certain rows
label_subopts	
refcat (matchlist[, subopts])	add reference category information
subopts:	
label (string)	redefine the "ref." label
below	change positioning of refcat
mlabels (strlist[, subopts])	label the models
subopts:	
[no] depvvars	use the name/label of the dependent variable as model label
[no] numbers	number models labels consecutively
label_subopts	
collabels (strlist[, label_subopts])	label the columns within models
eqlabels (strlist[, label_subopts])	label the equations
mgroups (strlist[, subopts])	define and label groups of models
subopts:	
pattern (pattern)	define the grouping of the models
label_subopts	
numbers [(l r)] nonumbers	add a row containing model numbers

Output

[no] replace	overwrite an existing file
[no] append	append the output to an existing file
[no] type	print the table in the results window
[no] showtabs	display tabs as $\langle T \rangle$ s

Defaults

style (style)	specify a style for the output table
styles:	
tab	tab delimited table
fixed	fixed format table
tex	table for use with LaTeX
html	table for use with HTML
mystyle	user defined addition

<i>label_subopts</i>	Description
[no]none	suppress the labels
prefix(string)	add a common prefix
suffix(string)	add a common suffix
begin(strlist)	add an overall prefix
end(strlist)	add an overall suffix
[no]last	print the last occurrence of end
[no]span	span columns if appropriate
erepeat(string)	add a "span" suffix
lhs(string)	label the table's left stub

Description

estout assembles a table of coefficients, "significance stars", summary statistics, standard errors, t- or z-statistics, p-values, confidence intervals, and other statistics for one or more models previously fitted and stored by **estimates store** or **esto**. It then writes the table to the Stata log and/or to a text file specified by **using**. *namelist* provides the names of the stored estimation sets to be tabulated. You may use the * and ? wildcards in *namelist*. The results estimated last may be indicated by a period (.), even if they have not yet been stored. If no model is specified, **estout** tabulates the estimation sets stored by **esto** (see help **esto**) or, if no such estimates are present, the currently active estimates (i.e. the model fit last). **estout** may be used after any estimation command that correctly returns its results in **e()**.

See the Introduction in the Examples section for an introduction on using **estout**. See help **estimates** for general information about managing estimation results. Furthermore, see help **esto** for an alternative to the **estimates store** command.

The default for **estout** is to produce a plain, tab-separated table containing point estimates. Producing a fully formatted end-product may involve specifying many options. However, note that a simple-to-use **estout** wrapper producing pre-formatted publication style tables is available as **esta**. Furthermore, use **estadd** to make additional results available for tabulation such as the standardized coefficients or the means and standard deviations of the regressors.

Programms similar to **estout** include **outreg** by John Luke Gallup, **outreg2** by Roy Wada, **modltbl** by John H. Tyler, **mktab** by Nicholas Winter, **outtex** by Antoine Terracol, or **est2tex** by Marc Muendler. Also see Newson (2003) for a very appealing approach.

Options

Contents

Parameter statistics
Summary statistics
Significance stars
Layout
Labeling
Output
Defaults
label_subopts

Parameter statistics

cells(array) specifies the parameter statistics to be reported and how they are to be arranged. The default is for cells to report raw coefficients only, i.e. **cells(b)**. **cells(none)** may be used to completely suppress the printing of parameter statistics. Alternatively, **cells(b se)** would result in the reporting of raw coefficients and standard errors. Multiple statistics are placed in separate rows beneath one another by default. However, elements of array that are listed in quotes, e.g. **cells("b se")**, are placed beside one another. For example, **cells("b p" se)** would produce a table with raw coefficients and standard errors beneath one another in the first column and p-values in the top row of the second column for each model.

The parameter statistics available are **b** (coefficients), **se** (standard errors), **t** (t/z-statistics), **p** (p-values), and **ci** (confidence intervals; to display the lower and upper bounds in separate cells use **ci_l** and **ci_u**). Any additional parameter statistics included in the **e()**-returns for the models can be tabulated as well. If, for example, **e(beta)** contains the standardized coefficients, type **cells(beta)** to tabulate them. Use **estadd** to add statistics such as the standardized coefficients to the **e()**-returns of a model. Also see the **eform** and **margin** options for more information on the kinds of statistics that can be displayed.

A set of suboptions may be specified in parentheses for each statistic named in array. For example, to add significance stars to the coefficients and place the standard errors in parentheses, specify **cells(b(star) se(par))**. The following suboptions are available. Use:

star to specify that stars denoting the significance of the coefficients be attached to the statistic: * for p<.05, ** for p<.01, and *** for p<.001. The symbols and the values for the thresholds and the number of levels are fully customizable (see the Significance stars options).

fmt(fmt [fmt ...]) to specify the display format(s) of a statistic. It defaults to the display format for raw coefficients (**b**), or **%9.0g**. If only one format is specified, it is used for all occurrences of the statistic. For example, type

```
. estout ..., cells("b(fmt(3)) t(fmt(2))")
```

to print coefficients and t-values beside one another using three decimal places for coefficients and two decimal places for t-values. If multiple formats are specified, the first format is used for the first regressor in the estimates table, the second format for the second regressor, and so on. The last format is used for the remaining regressors if the number of regressors in the table is greater than the number of specified formats. For instance, type

```
. estout ..., cells(b(fmt(3 4 2)))
```

to use three decimal places for the first coefficient, four decimal places for the second, and two decimal places for all remaining coefficients. Note that, regardless of the display format chosen, leading and trailing blanks are removed from the numbers. White space can be added by specifying a `modelwidth()` (see the [Layout](#) options). `fmt` may be any of Stata's numerical display formats, e.g., `%9.0g` or `%8.2f`, an integer `#` such as `1` or `3` to use a fixed format with `#` decimal places, or `a#` such as `a1` or `a3` to use `estout`'s automatic display format (see [Numerical formats](#) in the [Remarks](#) section for details).

`label(string)` to specify a label to appear in the column heading. The default is the name of the statistic.

`par[(l r)]` to specify that the statistic in question be placed in parentheses. It is also possible to specify custom "parentheses". For example, `se(par({ }))` would display the standard errors in curly brackets. Or, `se(par(`"="(' `)"'))` will write parentheses in a way that Excel can recognize. For `ci` the syntax is:

```
ci(par[(l m r)])
```

`drop(droplist)` to cause certain individual statistics to be dropped. For example, specifying `t(drop(_cons))` suppresses the t-statistics for the constants. A `droplist` comprises one or more specifications, separated by white space. A specification can be either a parameter name (e.g. `price`), an equation name followed by a colon (e.g. `mean:`), or a full name (e.g. `mean:price`). You may use the `*` and `?` wildcards in equation names and parameter names. Be sure to refer to the matched equation names, and not to the original equation names in the models, when using the `equations()` option to match equations.

`keep(keeplist)` to cause certain individual statistics to be kept. For example, the specification `t(keep(mpg))` would display the t-statistics exclusively for the variable `mpg`. See the `drop()` suboption above for further details.

pattern(*pattern*) to designate a pattern of models for which the statistics are to be reported, where the *pattern* consists of zeros and ones. A 1 indicates that the statistic be printed; 0 indicates that it be suppressed. For example **beta(pattern(1 0 1))** would result in **beta** being reported for the first and third models, but not for the second.

abs to specify that absolute t-statistics be used instead of regular t-statistics (relevant only if used with **t()**).

drop(*droplist*) identifies the coefficients to be dropped from the table. A *droplist* comprises one or more specifications, separated by white space. A specification can be either a parameter name (e.g. **price**), an equation name followed by a colon (e.g. **mean:**), or a full name (e.g. **mean:price**). You may use the * and ? wildcards in equation names and parameter names. Be sure to refer to the matched equation names, and not to the original equation names in the models, when using the **equations()** option to match equations.

keep(*keeplist*) selects the coefficients to be included in the table. *keeplist* is specified analogous to *droplist* in **drop()** (see above). Note that **keep()** does *not* change the the order of the coefficients. Use **order()** to change the order of coefficients.

order(*orderlist*) changes the order of the coefficients and equations within the table. *orderlist* is specified analogous to *droplist* in **drop()** (see above). Reordering of coefficients is performed equation by equation, unless equations are explicitly specified. Coefficients and equations that do not appear in *orderlist* are placed last (in their original order).

indicate(*groups* [, **labels**(*yes no*)]) indicates for each model (or, if **unstack** is specified, for each equation) the presence of certain groups of coefficients at the end of the table body. The syntax for *groups* is

```
group [ group ... ]
```

where a *group* is

```
[name = ] list
```

and *list* is a list of coefficient specifications as defined in **drop()** above. The single groups should be enclosed in quotes unless there is only one group and "name =" is specified. If "name =" is omitted, the first element of *list* is used as name. Note that *name* may contain spaces.

For example, if some of the models contain a set of year dummies, say **y1 y2 y3**, specify

```
estout ..., indicate(year effects = y1 y2 y3)
```

to drop the dummies from the table and add a "year effects" row containing "Yes" for models in which *at least one* of the dummies is present, and "No" for the other models.

Use the **labels()** suboption to redefine the indication labels to be printed in the table. The default is **labels(Yes No)**. Use quotes if the labels include spaces, e.g. **labels("in model" "not in model")**.

equations(eqmatchlist) specifies how the models' equations are to be matched. This option is passed to the internal call of **estimates table**. See help **estimates** on how to specify this option. Note that **estout** automatically matches the models' first equations into an equation called **main** if the equations have different names. Specify **equations("")** to suppress this behavior.

eform[*pattern*] displays the coefficient table in exponentiated form. The exponent of **b** is displayed in lieu of the untransformed coefficient; standard errors and confidence intervals are transformed as well. Specify a *pattern* if the exponentiation is to be applied only for certain models. For instance, **eform(1 0 1)** would transform the statistics for Models 1 and 3, but not for Model 2. Note that, unlike **regress** and **estimates table**, **estout** in eform-mode does not suppress the display of the intercept. To drop the intercept in eform-mode, specify **drop(_cons)**. Note: **eform** is implemented via the **transform()** option. If both options are specified, **transform()** takes precedence over **eform**.

transform(list [, *pattern(pattern)*]) displays transformed coefficients, standard errors and confidence intervals. *list* may be

fx dfx

where *fx* is the transformation function and *dfx* is its first derivative. *fx* is applied to coefficients and confidence intervals, that is, *fx(b)* and *fx(ci)* is displayed instead of **b** and **ci**. *dfx* is used to delta transform standard errors, i.e. *se*dfx(b)* is displayed instead of **se**. Use **@** as a placeholder for the function's argument in *fx* and *dfx*. For example, type

```
estout ..., transform(exp(@) exp(@))
```

to report exponentiated results (this is equivalent to specifying the **eform** option).

Alternatively, *list* may be specified as

```
coefs fx dfx [ ... [coefs] fx dfx ]
```

where *coefs* identifies the coefficients to be transformed. Syntax for *coefs* is as explained above in the description of the **drop()** option (however, include *coefs* in quotes if it contains multiple elements). Say, a model has two equations, **price** and **select**, and you want to exponentiate the **price** equation but not the **select** equation. You could then type


```
estout ..., transform(price: exp(@) exp(@))
```

Note that omitting *coef* in the last transformation specification causes the last transformation to be applied to all remaining coefficients.

Specify the **pattern()** suboption if the transformations are to be applied only for certain models. For instance, **pattern(1 0 1)** would apply the transformation to Models 1 and 3, but not Model 2.

margin[u|c|p] indicates that the marginal effects or elasticities be reported instead of the raw coefficients. This option has an effect only if **mfx** has been applied to a model before its results were stored (see help **mfx**) or if a **dprobit** (see help **probit**), **truncreg,marginal** (help **truncreg**), or **dtobit** (Cong 2000) model is estimated. One of the parameters **u**, **c**, or **p**, corresponding to the unconditional, conditional, and probability marginal effects, respectively, is required for **dtobit**. Note that the standard errors, confidence intervals, t-statistics, and p-values are transformed as well.

Using the **margin** option with multiple-equation models can be tricky. The marginal effects of variables that are used in several equations are printed repeatedly for each equation because the equations per se are meaningless for **mfx**. To display the effects for certain equations only, specify the **meqs()** option. Alternatively, use the **keep()** and **drop()** options to eliminate redundant rows. The **equations()** option might also be of help here.

discrete(string) may be used to override the default symbol and explanatory text used to identify dummy variables when reporting marginal effects. The first token in *string* is used as the symbol. The default is:

```
discrete(" (d)" for discrete change of dummy variable from 0 to 1)
```

To display explanatory text, specify either the **legend** option or use the **@discrete** variable.

Use **nodiscrete** to disable the identification of dummy variables as such. The default is to indicate the dummy variables unless they have been interpreted as continuous variables in all of the models for which results are reported (for **dprobit** and **dtobit**, however, dummy variables will always be listed as discrete variables unless **nodiscrete** is specified).

meqs(eq_list) specifies that marginals be printed only for the equations in *eq_list*. Specifying this option does not affect how the marginals are calculated. An *eq_list* comprises one or more equation names (without colons) separated by white space. If you use the **equations()** option to match equations, be sure to refer to the matched equation names and not to the original equation names in the models.

level(#) assigns the confidence level, in percent, for the confidence intervals of the coefficients (see help **level**).

stats(*scalarlist* [, *stats_subopts*]) specifies one or more scalar statistics - separated by white space - to be displayed at the bottom of the table. The *scalarlist* may contain **e()**-scalars (such as, e.g., **N**, **r2**, or **chi2**) and the following statistics:

aic	Akaike's information criterion
bic	Schwarz's information criterion
rank	rank of e(V) , i.e. the number of free parameters in model
p	the p-value of the model (overall model significance)

See help [estimates](#) for details on the **aic** and **bic** statistics. The rules for the determination of **p** are as follows (note that although the procedure outlined below is appropriate for most models, there might be some models for which it is not):

- 1) p-value provided: If the **e(p)** scalar is provided by the estimation command, it will be interpreted as indicating the p-value of the model.
- 2) F test: If **e(p)** is not provided, **estout** checks for the presence of the **e(df_m)**, **e(df_r)**, and **e(F)** scalars and, if they are present, the p-value of the model will be calculated as **Ftail(df_m,df_r,F)**. This p-value corresponds to the standard overall F test of linear regression.
- 3) chi2 test: Otherwise, if neither **e(p)** nor **e(F)** is provided, **estout** checks for the presence of **e(df_m)** and **e(chi2)** and, if they are present, calculates the p-value as **chi2tail(df_m,chi2)**. This p-value corresponds to the Likelihood-Ratio or Wald chi2 test.
- 4) If neither **e(p)**, **e(F)**, nor **e(chi2)** is available, no p-value will be reported.

Type **ereturn list** after estimating a model to see a list of the returned **e()**-scalars (see help [ereturn](#)). Use [estadd](#) (available from the SSC Archive) to add extra statistics to the **e()**-returns.

The following *stats_subopts* are available. Use:

fmt(*fmt* [*fmt* ...]) to set the display formats for the scalar statistics in *scalarlist*. *fmt* may be any of Stata's numerical display formats, e.g., **%9.0g** or **%8.2f**, an integer # such as **1** or **3** to use a fixed format with # decimal places, or **a#** such as **a1** or **a3** to use **estout**'s automatic display format (see [Numerical formats](#) in the [Remarks](#) section for details). For example, **fmt(3 0)** would be suitable for **stats(r2_a N)**. Note that the last specified format is used for the remaining scalars if the list of scalars is longer than the list of formats. Thus, only one format need be specified if all scalars are to be displayed in the same format. If no format is specified, the default format is the display format of the coefficients.

labels(*strlist*[, *label_subopts*]) to specify labels for the scalars in *scalarlist*. If specified, the labels are used instead of the scalar names. For example:

```
. estout ..., stats(r2_a N, labels("Adj. R-Square" "Number of Cases"))
```

Note that names like **r2_a** produce an error in LaTeX because the underscore character has a special meaning in LaTeX (to print the underscore in LaTeX, type `_`). Use the **label()** suboption to rename such statistics, e.g. **stats(r2_a, labels(r2_a))**. An alternative approach is to use **estout**'s **substitute()** option (see the [Layout](#) options).

star(*scalarlist*) to specify that the overall significance of the model be denoted by stars. The stars are attached to the scalar statistics specified in *scalarlist*. If *scalarlist* is omitted, the stars are attached to the first reported scalar statistic. The printing of the stars is suppressed in empty results cells (i.e. if the scalar statistic in question is missing for a certain model). The determination of the model significance is based on the p-value of the model (see above).

Hint: It is possible to attach the stars to different scalar statistics within the same table. For example, specify **stats(,star(r2_a r2_p))** when tabulating OLS estimates and, say, probit estimates. For the OLS models, the F test will be carried out and the significance stars will be attached to the **r2_a**; for the probit models, the chi2 test will be used and the stars will appear next to the **r2_p**.

Significance stars

starlevels(*levelslist*) overrides the default thresholds and symbols for "significance stars". For instance, **starlevels(+ 0.10 * 0.05)** sets the following thresholds: + for $p < .10$ and * for $p < .05$. Note that the thresholds must lie in the (0,1] interval and must be specified in descending order. To, for example, denote insignificant results, type **starlevels(* 1 "" 0.05)**.

stardetach specifies that a delimiter be placed between the statistics and the significance stars (i.e. that the stars are to be displayed in their own column).

Layout

varwidth(#) specifies the number of characters used to display the names (labels) of regressors and statistics (i.e. **varwidth** specifies the width of the table's left stub). Long names (labels) are abbreviated (depending on the **abbrev** option) and short or empty cells are padded out with blanks to fit the width specified by the user. **varwidth** defaults to 0, which means that the names are not abbreviated and no white space is added. Specifying low values may cause misalignment.

modelwidth(#) designates the number of characters used to display the results columns. If a non-zero **modelwidth** is specified, model names are abbreviated if necessary (depending on the **abbrev** option) and short or empty results cells are padded out with blanks. In contrast, **modelwidth** does not shorten or truncate the display of the results themselves (coefficients, t-statistics, summary statistics, etc.) although it may add blanks if needed. **modelwidth** defaults to 0, which means that the model names are not abbreviated and no white space is added. Specifying low values may cause misalignment.

The purpose of **modelwidth** is to be able to construct a fixed-format table and thus make the raw table more readable. Be aware, however, that the added blanks may cause problems with the conversion to a table in word processors or spreadsheets.

abbrev specifies that long names and labels be abbreviated if a **modelwidth()** and/or a **varwidth()** is specified.

unstack specifies that the individual equations from multiple-equation models (e.g. **mlogit**, **reg3**, **heckman**) be placed in separate columns. The default is to place the equations below one another in a single column. Summary statistics will be reported for each equation if **unstack** is specified and the estimation command is either **reg3**, **sureg**, or **mvreg** (see help **reg3**, help **sureg**, help **mvreg**).

begin(string) specifies a string to be printed at the beginning of every table row. The default is an empty string. It is possible to use special functions such as **_tab** or **_skip** in **begin()**. For more information on using such functions, see the description of the functions in help **file**.

delimiter(string) designates the delimiter used between the table columns. The default is a tab character. See the **begin** option above for further details.

end(string) specifies a string to be printed at the end of every table row. The default is an empty string. See the **begin** option above for further details.

dmarker(string) specifies the form of the decimal marker. The standard decimal symbol (a period or a comma, depending on the input provided to **set dp**; see help **format**) is replaced by **string**.

msign(string) determines the form of the minus sign. The standard minus sign (-) is replaced by **string**.

lz specifies that the leading zero of fixed format numbers in the interval (-1,1) be printed. This is the default. Use **no1z** to advise **estout** to omit the leading zeros (that is, to print numbers like **0.021** or **-0.33** as **.021** and **-.33**).

extracols(*numlist*) inserts empty table columns at the indicated positions. For example, **extracols(1)** adds an extra column between the left stub of the table and the first column.

substitute(*subst_list*) specifies that the substitutions specified in *subst_list* be applied to the estimates table after it has been created. Specify *subst_list* as a list of substitution pairs, that is:

from to [from to ...]

For example, specify **substitute(_ _)** to replace the underscore character (as in **_cons** or **F_p**) with it's LaTeX equivalent **_**.

Labeling

title(*string*) may be used to specify a title for the table. The *string* is printed at the top of the table unless **prehead()**, **posthead()**, **prefoot()**, or **postfoot()** is specified. In the latter case, the variable **@title** can be used to insert the title.

legend adds a legend explaining the significance symbols and thresholds.

prehead(*strlist*), **posthead**(*strlist*), **prefoot**(*strlist*), and **postfoot**(*strlist*) may be used to define lists of text lines to appear before and after the table heading or the table footer. For example, the specification

```
. estout ..., prehead("\S_DATE \S_TIME" "")
```

would add a line containing the current date and time followed by an empty line before the table. Various substitution functions can be used as part of the text lines specified in *strlist*, including **@span** to print the total number of physical columns in the table (including the left stub that holds the variable names), **@M** to print the number of models included, **@discrete** to print the contents of the **discrete()** option, **@starlegend** to print a legend explaining the significance symbols, **@title** to print the contents of the **title()** option, and **@hline** to plot a horizontal "line" (series of dashes, by default; see the **hlinechar()** option). For example, a table heading to be used with LaTeX might be formatted as follows:

```
. estout ..., prehead(\begin{tabular}{l*{@M}{r}})
```

hlinechar(*string*) specifies the character(s) to be used in **@hline**. The default is **hlinechar(-)**, resulting in a dashed line. To produce a solid line, specify **hlinechar(`=char(151)')** (Windows only; other systems may use other codes).

label specifies that variable labels be displayed instead of variable names in the left stub of the table.

varlabels(*matchlist*[, *suboptions*]) may be used to relabel the regressors from the models, where *matchlist* is

```
name label [name label ...]
```

A *name* is a parameter name (e.g. **price**) or a full name (e.g. **mean:price**) (abbreviation and wildcards not allowed). For example, specify **varlabels(_cons Constant)** to replace each occurrence of **_cons** with **Constant**. (Note that, in LaTeX, the underscore character produces an error unless it is specified as `_`. Thus, names such as **_cons** should always be changed if the estimates table is to be used with LaTeX. The **substitute()** may also be helpful; see the [Layout](#) options.) The *suboptions* are:

blist(*matchlist*) to assign specific prefixes to certain rows in the table body. Specify the *matchlist* as pairs of regressors and prefixes, that is:

```
name prefix [name prefix ...]
```

A *name* is a parameter name (e.g. **price**), an equation name followed by a colon (e.g. **mean:**), or a full name (e.g. **mean:price**) (abbreviation and wildcards not allowed). Note that equation names cannot be used if the **unstack** option is specified. The prefix will include the total number of physical columns in the table if the **@span** token is used in its definition.

elist(*matchlist*) to assign specific suffixes to certain rows in the table body (see the analogous **blist()** option above). This option may, for example, be useful for separating thematic blocks of variables by adding vertical space at the end of each block. A LaTeX example:

```
. estout ..., varlabels(,elist(price \addlinespace mpg
\addlinespace))
```

(the macro `\addlinespace` is provided by the **booktabs** package in LaTeX)

[label_subopts](#), which are explained in their own section.

refcat(*matchlist*[, *suboptions*]) may be used to insert a row containing information on the reference category of a categorical variable in the model. *matchlist* is

```
name refcat [name refcat ...]
```

A *name* is a parameter name (e.g. `_Irep78_2`) (abbreviation and wildcards not allowed). For example, assume that you include the categorical variable `rep78` ("Repair Record 1978" from the auto dataset) in some of your models using `xi` (see help `xi`). Since `rep78` has five levels, 1 through 5, `xi` will create 4 dummy variables, `_Irep78_2` through `_Irep78_5`. You can now type

```
. estout ..., refcat(_Irep78_2 _Irep78_1)
```

to add a table row containing "`_Irep78_1`" in the left stub and "ref." in each column in which the `_Irep78_2` dummy appears. The *suboptions* are:

label(*string*) to specify the label that is printed in the table columns. The default is **label(ref.)**

below to position the reference category row below the specified coefficient's row. The default is above. For example, if the 5th category of `rep78` is used as reference category, i.e. if `_Irep78_1` through `_Irep78_4` are included in the models, you might want to type **refcat(_Irep78_4 _Irep78_5, below)**.

mlabels(*strlist[, suboptions]*) determines the model captions printed in the table heading. The default is to use the names of the stored estimation sets (or their titles, if the **label** option is specified and titles are available). The *suboptions* for use with **mlabels** are:

depvars to specify that the name (or label) of the (first) dependent variable of the model be used as model label.

numbers to cause the model labels to be numbered consecutively.

label_subopts, which are explained in their own section.

collabels(*strlist[, label_subopts]*) specifies labels for the columns within models or equations. The default is to compose a label from the names or labels of the statistics printed in the cells of that column. The *label_subopts* are explained in their own section below.

eqlabels(*strlist[, label_subopts]*) labels the equations. The default is to use the equation names as stored by the estimation command, or to use the variable labels if the equation names correspond to individual variables and the **label** option is specified. The *label_subopts* are explained in their own section below. Note that **eqlabels(, none)** causes **_cons** to be replaced with the equation name or label, if **_cons** is the only parameter in an equation. This is useful, e.g., for tabulating **ologit** or **oprobit** results in Stata 9. Specify **eqlabels("", none)** to not replace **_cons**.

mgroups(*strlist[, suboptions]*) may be used to label groups of (consecutive) models at the top of the table heading. The labels are placed in the first physical column of the output for the group of models to which they apply. The *suboptions* for use with **mgroups** are:

pattern(*pattern*) to establish how the models are to be grouped. *pattern* should be a list of zeros and ones, with ones indicating the start of a new group of models. For example,

```
. estout ..., mgroups("Group 1" "Group 2", pattern(1 0 0 1
0))
```

would group Models 1, 2, and 3 together and then groups Models 4 and 5 together as well. Note that the first group will always start with the first model regardless of whether the first token of *pattern* is a one or a zero.

label_subopts, which are explained in their own section. In particular, the **span** suboption might be of interest here.

numbers[*l r*] adds a row to the table header displaying consecutive model numbers. The default is to enclose the numbers in parentheses, i.e. (1), (2), etc. Alternatively, specify *l* and *r* to change the tokens on the left and right of each number. For example, **numbers**(" ") would result in 1), 2), etc.

Output

replace permits **estout** to overwrite an existing file.

append specifies that the output be appended to an existing file. It may be used even if the file does not yet exist.

type specifies that the assembled estimates table be printed in the results window and the log file. This is the default. Use **notype** to suppress the display of the table.

showtabs requests that tabs be displayed as `<T>`s in both the results window and the log file instead of in expanded form. This option does not affect how tabs are written to the text file specified by **using**.

Defaults

style(*style*) specifies a "style" for the output table. **defaults**(*style*) is a synonym for **style**(*style*). A "style" is a named combination of options that is saved in an auxiliary file called **estout_style.def**. In addition, there are four internal styles called **tab**, **fixed**, **tex**, and **html**. Their particulars are:

settings	styles			
	tab	fixed	tex	html
begin				<tr><td>
delimiter	_tab	" "	&	</td><td>
end			\\	</td></tr>
varwidth	0	12/20*	12/20*	12/20*
modelwidth	0	12	12	12
abbrev	off	on	off	off
			(* if label is on)	

The **tab** style is the default. See [Defaults files](#) in the [Remarks](#) section to make available your own style.

Note that explicitly specified options take precedence over settings provided by a style. For example, if you type

```
. estout, delimiter("") style(tab)
```

then the column delimiter will be set to empty string since the **delimiter()** option overwrites the default from the **tab** style. Similarly, specifying **noabbrev** will turn abbreviation off if using the **fixed** style.

label_subopts

The following suboptions may be used within the **mgroups()**, **mlabels()**, **collabels()**, **eqlabels()**, **varlabels()**, and **stats(, labels())** options:

none suppresses the printing of the labels or drops the part of the table heading to which it applies.

prefix(string) specifies a common prefix to be added to each label.

suffix(string) specifies a common suffix to be added to each label.

begin(strlist) specifies a prefix to be printed at the beginning of the part of the table to which it applies. If **begin** is specified in **varlabels()** or **stats(, labels())**, the prefix will be repeated for each regressor or summary statistic.

end(strlist) specifies a suffix to be printed at the end of the part of the table to which it applies. If **end** is specified in **varlabels()** or **stats(, labels())**, the suffix will be repeated for each regressor or summary statistic.

last specifies that the last occurrence of the **end()**-suffix in **varlabels()** or **stats(, labels())** be printed. This is the default. Use **no last** to suppress the last occurrence of the suffix.

span causes labels to span columns, i.e. extends the labels across several columns, if appropriate. This suboption is relevant only for the **mgroups()**, **mlabels()**, **eqlabels()**, and **collabels()** options. The **@span** string returns the number of spanned columns if it is included in the label, prefix, or suffix. A LaTeX example:

```
. estout ..., mlabels(, span prefix{\multicolumn{@span}{c}{}} suffix({}))
```

erepeat(string) specifies a string that is repeated for each group of spanned columns at the very end of the row if the **span** suboption is specified. This suboption is relevant only for the **mgroups()**, **mlabels()**, **eqlabels()**, and **collabels()** options. If the **@span** string is included in *string* it will be replaced by the range of columns spanned. A LaTeX example:

```
. estout ..., mlabels(, span erepeat(\cline{@span}))
```

lhs(string) inserts *string* into the otherwise empty cell in the left stub of the row of the table heading to which it applies. This suboption is relevant only for the **mgroups()**, **mlabels()**, **eqlabels()**, and **collabels()** options.

Examples

Contents

[Introduction](#)
[Publication style table](#)
[t-statistics for selected variables only](#)
[Summary statistics only](#)
[Table of descriptives](#)
[Unstack multiple equations](#)
[Marginal effects](#)

Please first read the [Introduction](#). The other examples are more advanced and intended for users already familiar with the basic features of **estout**. Additional examples can be found in Jann (2005).

Introduction

The full syntax of **estout** is rather complex and is to be found above. However, consider the following basic syntax, which includes only the most important options:

```
estout [ namelist ] [ using filename ] [ , cells(array) stats(scalarlist)  
      style(style) more_options ]
```

where *namelist* is a list of the names of stored estimation sets (the name list can be entered as * to refer to all stored estimates). The **cells()** and **stats()** options determine the primary contents of the table. The **style()** option determines the basic formatting of the table.

Basic usage

The general procedure for using **estout** is to first store several models using the **estimates store** or the **esto** command and then apply **estout** to save and/or display a table of the estimates. By default, **estout** produces a plain, tab-separated table of the coefficients of the models indicated by the command:

```
. sysuse auto
(1978 Automobile Data)

. replace price = price / 1000
price was int now float
(74 real changes made)

. replace weight = weight / 1000
weight was int now float
(74 real changes made)

. regress price weight mpg
(output omitted)

. estimates store m1, title(Model 1)

. generate forXmpg = foreign * mpg

. regress price weight mpg forXmpg foreign
(output omitted)

. estimates store m2, title(Model 2)

. estout * using example.txt
```

	m1	m2
	b	b
weight	1.746559	4.613589
mpg	-.0495122	.2631875
forXmpg		-.3072165
foreign		11.24033
_cons	1.946068	-14.44958

The table produced by the **estout** command looks messy in the Stata results window or the Stata log because the columns are tab-separated (note that tab characters are not preserved in the results window or the log). However, the stored **example.txt** would look tidy if it were opened, for example, in a spreadsheet program.

Choosing a style

To align the columns, fixed widths can be specified for the columns and tab characters can be removed. This is most easily done via the **style()** option, which provides a style called **fixed**:

```
. estout *, style(fixed)
```

	m1	m2
	b	b
weight	1.746559	4.613589
mpg	-.0495122	.2631875
forXmpg		-.3072165
foreign		11.24033
_cons	1.946068	-14.44958

Other predefined styles are **tab** (the default), **tex**, and **html**, but it is also possible to define one's own styles (see [Defaults files](#) in the [Remarks](#) section). The **tex** style, for example, modifies the output table for use with LaTeX's tabular environment:

```
. estout *, style(tex) varlabels(_cons \_cons)
```

	&	m1&	m2\\
	&	b&	b\\
weight	&	1.746559&	4.613589\\
mpg	&	-.0495122&	.2631875\\
forXmpg	&	&	-.3072165\\
foreign	&	&	11.24033\\
_cons	&	1.946068&	-14.44958\\

Note that **_cons** has been replaced by its LaTeX equivalent in the example above using the **varlabels()** option (the underscore character produces an error in LaTeX unless it is preceded by a backslash). For more information on the **varlabels()** option, see **estout**'s [Labeling](#) options.

The cells option

Use the **cells()** option to specify the parameter statistics to be tabulated and how they are to be arranged. The parameter statistics available are **b** (coefficients; the default), **se** (standard errors), **t** (t-/z-statistics), **p** (p-values), **ci** (confidence intervals; to display the lower and upper bounds in separate cells use **ci_l** and **ci_u**), as well as any additional parameter statistics included in the **e()**-returns for the models (see **estout**'s [Parameter Statistics](#) options). For example, **cells(b se)** results in the reporting of raw coefficients and standard errors:

```
. estout *, cells(b se) style(fixed)
```

	m1	m2
	b/se	b/se
weight	1.746559	4.613589
	.6413538	.7254961
mpg	-.0495122	.2631875
	.086156	.1107961
forXmpg		-.3072165
		.1085307
foreign		11.24033
		2.751681
_cons	1.946068	-14.44958
	3.59705	4.42572

Multiple statistics are placed in separate rows beneath one another by default as in the example above. However, elements that are listed in quotes are placed beside one another. For example, specifying `cells("b se t p")` produces the following table:

```
. estout m2, cells("b se t p") style(fixed)
```

	m2			
	b	se	t	p
weight	4.613589	.7254961	6.359219	1.89e-08
mpg	.2631875	.1107961	2.375421	.0203122
forXmpg	-.3072165	.1085307	-2.830687	.0060799
foreign	11.24033	2.751681	4.084896	.0001171
_cons	-14.44958	4.42572	-3.26491	.0017061

The two approaches can be combined. For example, `cells("b p" se)` would produce a table with raw coefficients and standard errors beneath one another in the first column and p-values in the top row of the second column for each model.

Note that for each statistic named in the `cells()` option a set of suboptions may be specified in parentheses. For example, in social sciences it is common to report standard errors or t-statistics in parentheses beneath the coefficients and to indicate the significance of individual coefficients with stars. Furthermore, the results are rounded. Just such a table can be created using the following procedure:

```
. estout *, cells(b(star fmt(3)) t(par fmt(2))) style(fixed)
```

	m1	m2
	b/t	b/t
weight	1.747** (2.72)	4.614*** (6.36)
mpg	-0.050 (-0.57)	0.263* (2.38)
forXmpg		-0.307** (-2.83)
foreign		11.240*** (4.08)
_cons	1.946 (0.54)	-14.450** (-3.26)

The **estout** default is to display * for $p < .05$, ** for $p < .01$, and *** for $p < .001$. However, note that the significance thresholds and symbols are fully customizable (see **estout**'s [Significance stars](#) options).

The stats option

Finally, use the **stats()** option to specify scalar statistics to be displayed in the last rows of each model's table. The available scalar statistics are **aic** (Akaike's information criterion), **bic** (Schwarz's information criterion), **rank** (the rank of $e(V)$, i.e. the number of free parameters in model), **p** (the p-value of the model), as well as any scalar contained in the **e()**-returns for the models (see **estout**'s [Summary statistics](#) options). For example, specify **stats(r2 bic N)** to add the R-squared, BIC, and the number of cases to the bottom of the table:

```
. estout *, stats(r2 bic N) style(fixed)
```

	m1	m2
	b	b
weight	1.746559	4.613589
mpg	-.0495122	.2631875
forXmpg		-.3072165
foreign		11.24033
_cons	1.946068	-14.44958
r2	.2933891	.5516277
bic	356.2918	331.2406
N	74	74

Publication style table

```

. label variable foreign "Foreign car type"

. label variable forXmpg "Foreign*Mileage"

. estout *, cells(b(star fmt(%9.3f)) se(par))
> stats(r2_a N, fmt(%9.3f %9.0g) labels(R-squared))
> legend label collabels(, none)
> varlabels(_cons Constant) posthead("")
> prefoot("") postfoot("")
> varwidth(16) modelwidth(12) delimiter("")

```

	Model 1	Model 2
Weight (lbs.)	1.747**	4.614***
	(0.641)	(0.725)
Mileage (mpg)	-0.050	0.263*
	(0.086)	(0.111)
Foreign*Mileage		-0.307**
		(0.109)
Foreign car type		11.240***
		(2.752)
Constant	1.946	-14.450**
	(3.597)	(4.426)
R-squared	0.273	0.526
N	74	74

* p<0.05, ** p<0.01, *** p<0.001

t-statistics for selected variables only

```

. estout *, cells(b(star) t(par keep(mpg)))
> style(fixed)

```

	m1	m2
	b/t	b/t
weight	1.746559**	4.613589***
mpg	-.0495122	.2631875*
	(-.5746806)	(2.375421)
forXmpg		-.3072165**
foreign		11.24033***
_cons	1.946068	-14.44958**

Summary statistics only

```

. estout *, cells(none) stats(r2_a bic N, star)
> style(fixed)

```

	m1	m2
r2_a	.2734846***	.5256351***
bic	356.2918	331.2406
N	74	74

Table of descriptives

```

. quietly generate x = uniform()

. quietly regress x price weight mpg foreign

. estadd mean

. estadd sd, nobinary

. estimates store m3

. estout m3, cells("mean sd") stats(N) mlabels(,none)
> drop(_cons) style(fixed)

```

	mean	sd
price	6.165257	2.949496
weight	3.019459	.7771936
mpg	21.2973	5.785503
foreign	.2972973	
N	74	

Unstack multiple equations

```

. sureg (price foreign weight length)
> (mpg displ = foreign weight)
(output omitted)

. estimates store m4

. estout m4, cells(b t(par)) unstack
> stats(r2 chi2 p) style(fixed)

```

	m4		mpg displacement	
	price		mpg	displacement
	b/t		b/t	b/t
foreign	3.57526	-1.650029	-25.6127	
	(5.749891)	(-1.565555)	(-2.047999)	
weight	5.691462	-6.587886	96.75485	
	(6.182983)	(-10.55641)	(13.06594)	
length	-.0882711			
	(-2.809689)			
_cons	4.506212	41.6797	-87.23547	
	(1.255897)	(19.64914)	(-3.46585)	
r2	.548808	.6627029	.8115213	
chi2	89.73586	145.3912	318.6174	
p	2.50e-19	2.68e-32	6.50e-70	

Marginal effects

```
. generate record = 0

. replace record = 1 if rep > 3
(34 real changes made)

. logit foreign mpg record
(output omitted)

. estimates store raw

. mfx
(output omitted)

. estimates store mfx

. estout raw mfx, cells("b XmfX_X(pattern(0 1))" se(par))
> margin legend style(fixed)
```

	raw	mfx	
	b/se	b/se	XmfX_X
mpg	.1079219 (.0565077)	.0184528 (.0101674)	21.2973
record (d)	2.435068 (.7128444)	.4271707 (.1043178)	.4594595
_cons	-4.689347 (1.326547)		

(d) marginals for discrete change of dummy variable from 0 to 1

Remarks

Contents

[Numerical formats](#)
[Special characters](#)
[Using @-variables](#)
[Defaults files](#)

Numerical formats

Numerical display formats may be specified in **estout** as follows:

1. Official Stata's display formats: You may specify formats, such as **%9.0g** or **%8.2f**. See help [format](#) for a list of available formats.

2. Fixed format: You may specify an integer value such as **0**, **1**, **2**, etc. to request a display format with a fixed number of decimal places. For example, `cells(t(fmt(3)))` would display t-statistics with three decimal places.
3. Automatic format: You may specify **a1**, **a2**, ..., or **a9** to cause `esta` to choose a reasonable display format for each number depending on the number's value. The **#** in **a#** determines the minimum precision according to the following rules:
 - o Absolute numbers smaller than 1 are displayed with **#** significant decimal places (i.e. with **#** decimal places ignoring any leading zeros after the decimal point). For example, **0.00123456** is displayed as **0.00123** if the format is **a3**.
 - o Absolute numbers greater than 1 are displayed with as many digits required to retain at least one decimal place and are displayed with a minimum of (**#** + 1) digits. For example, if the format is **a3**, **1.23456** is displayed as **1.235**, **12.3456** is displayed as **12.35**, and **1234.56** is displayed as **1234.6**.
 - o In any case, integers are displayed with zero decimal places, and very large or very small absolute numbers are displayed in exponential format.

Special characters

The `\` and `$` characters and quotation marks have special meanings in Stata. You should therefore consider the following instructions if you, for example, intend to specify awkward delimiters or specify special characters in labels:

- Strings containing unmatched quotes should be enclosed in compound double quotes (thus, `delimiter("`"'"')` results in columns delimited by ", while `delimiter(")` produces an error).
- The backslash character is used to delay macro expansion in Stata. Specifying `\\` just results in the printing of `\`. To get a double backslash (the `\newline` command in TeX), type `\\`.
- The dollar sign is used for global macro expansion in Stata. Thus, `$x` would result in the display of the contents of global macro `x` (or nothing, if the macro is empty). Therefore, use `\$` to produce `$` in the output.

Stata's `char()` function may also be used to specify odd characters (see help [strfun](#)). In particular, `"`=char(9)'"` results in a tab character and `"`=char(13)'"` results in a carriage return. For example, `delimiter(" `=char(9)' ")` specifies that a tab character with a leading and a trailing blank be used as delimiter.

Tip: It is sometimes very useful to set the format of all cells in a spreadsheet to "Text" before pasting the estimates table. This prevents the spreadsheet program from trying to interpret the cells and ensures that the contents of the table remain unchanged.

Using @-variables

estout features several variables that can be used within string specifications. The following list provides an overview of these variables:

@span

Returns the value of a count variable for the total number of physical columns of the table if used in the labels in the **blist()** and **elist()** suboptions of **varlabels()**, in the text specified in **prehead()**, **posthead()**, **prefoot()**, or **postfoot()**, or in the text specified in the **begin()** or **end()** label suboptions.

@span

Returns the number of spanned columns if used in the text specified in the **prefix()** and **suffix()** suboptions of **mgroups()**, **mlabels()**, **eqlabels()**, or **collabels()**, or in the labels specified in these options.

@span

Returns the range of spanned columns (e.g. 2-4 if columns 2, 3 and 4 are spanned) if used in the text specified in the **erepeat()** suboption of **mgroups()**, **mlabels()**, **eqlabels()**, or **collabels()**.

@M

Returns the number of models in the table if used in the text specified in **prehead()**, **posthead()**, **prefoot()**, or **postfoot()**.

@E

Returns the total number columns containing separate equations if used in the text specified in **prehead()**, **posthead()**, **prefoot()**, or **postfoot()**.

@title

Returns the title specified with the **title()** option if used in the text specified in **prehead()**, **posthead()**, **prefoot()**, or **postfoot()**.

@discrete

Returns the explanations provided by the **discrete()** option (provided that the **margin** option is activated) if used in the text specified in **prehead()**, **posthead()**, **prefoot()**, or **postfoot()**.

@starlegend

Returns a legend explaining the significance symbols if used in the text specified in **prehead()**, **posthead()**, **prefoot()**, or **postfoot()**.

@hline

Returns a horizontal line (series of dashes, by default; see the **hlinechar()** option) if used in the text specified in **prehead()**, **posthead()**, **prefoot()**, or **postfoot()**. Note that the length of the line depends on **varwidth()** and **modelwidth()**.

Defaults files

Creating new defaults files:

To make available an own set of default options, proceed as follows:

1. Download "estout_mystyle.def" from the SSC Archive (click [here](#) to copy the file from SSC and store it in the working directory).
2. Open "estout_mystyle.def" in a text editor and make the desired modifications (click [here](#) to open "estout_mystyle.def" in Stata's Do-File Editor).
3. Save the file in the current directory or elsewhere in the ado-file path as **estout_newstyle.def** (see help [sysdir](#)).

To use the new options set in **estout**, then type:

```
. estout ... , style(newstyle)
```

Defaults files syntax:

estout has two main types of options, which are treated differentially in defaults files. On the one hand, there are simple on/off options without arguments, like **legend** or **showtabs**. To turn such an option on, enter the option followed by the options name as an argument, i.e. add the line

```
option option
```

to the defaults file. For example,

```
legend legend
```

specifies that a legend be printed in the table footer. Otherwise, if you want to turn the option off, just delete or comment out the line that contains it (or specify *option* without an argument).

To temporarily turn off an option that has been activated in a defaults file, specify **nooption** in the command line (do not, however, use **nooption** in defaults files). For example, if the legend has been turned on in the defaults file, but you want to suppress it in a specific call of **estout**, type

```
. estout ..., nolegend
```

On the other hand, there are options that take arguments, such as **prehead**(args), **delimiter**(args), or **stats**(args, ...). Such options are specified as

option args

in the defaults file (where *args* must not include suboptions; see below). Specifying an option in the command line overwrites the settings from the defaults file. However, note that a **no** form, which exists for the first options type, is not available here.

Last but not least, there are two options that reflect a combination of the first and second types: **eform**(args) and **margin**(args). These options can be specified as either

option option

or

option args

in the defaults file; the **no** form is allowed.

Many **estout** options have suboptions, i.e., an option might take the form *option(..., suboption)* or *option(..., suboption(args))*. In the defaults file, the suboptions cannot be included in the definition of a higher-level option. Instead, they must be specified in their own lines, as either

optionsuboption suboption

or

optionsuboption args

In the case of a two-level nesting of options, the name used to refer to the suboption is a concatenation of the option's name and the suboption's name, i.e. "*optionsuboption*"="*option*"+"*suboption*". For example, the **labels**() suboption of the **stats**() option would be set by the term **statslabels**. Analogously, the three level nesting in the **stats**() option yields suboption names composed of three names. For instance, the suboption called by the command

```
. estout ..., stats(..., labels(..., prefix(args)))
```

would be referred to as

statslabelsprefix args

in the defaults file. The **cells**() option represents an exception to this rule. It may be defined in the defaults file using only a plain array of cells elements without suboptions, e.g.

cells "b se" p

However, the suboptions of the cells elements may be referred to as *el_suboption*, for example

```
b_star star
```

or

```
se_par [ ]
```

Comments in defaults files:

Be aware that the support for comments in defaults files is limited. In particular, the `/*` and `*/` comment indicators cannot be used. The other comment indicators work (more or less) as usual, that is:

- o Empty lines and lines beginning with `*` (with or without preceding blanks) will be ignored.
- o `//` preceded by one or more blanks indicates that the rest of the line should be ignored. Lines beginning with `//` (with or without preceding blanks) will be ignored.
- o `///` preceded by one or more blanks indicates that the rest of the line should be ignored and the part of the line preceding it should be added to the next line. In other words, `///` can be used to split commands into two or more lines of code.

References

- Cong, R. (2000). sg144: Marginal effects of the tobit model. *Stata Technical Bulletin* 56: 27-34.
- Jann, B. (2005). Making regression tables from stored estimates. *The Stata Journal* 5(3): 288-308.
- Newson, R. (2003). Confidence intervals and p-values for delivery to the end user. *The Stata Journal* 3(3): 245-269.

Acknowledgements

I would like to thank numerous people for their comments and suggestions. Among them are Joao Pedro Azevedo, Kit Baum, Elisabeth Coutts, Henriette Engelhardt, Jonathan Gardnerand, Simone Hirschvogel, Daniel Hoechle, Friedrich Huebler, Maren Kandulla, J. Scott Long, David Newhouse, Clive Nicholas, Fredrik Wallenberg, Ian Watson, and Vince Wiggins.

Author

Ben Jann, ETH Zurich, jann@soz.gess.ethz.ch

Also see

Manual: [R] **estimates**

SJ: SJ5-3 st0085 (Jann 2005)

Online: help for **estimates**, **estcom**, **estimates table**, **ereturn**, **format**, **file**,
mfx, **esto**, **esta**, **estadd**