The Stata Journal (*yyyy*) *vv*, Number *ii*, pp. 1–50

A new system for formatting estimation tables

John Luke Gallup
Portland State University
Portland, Oregon
jlgallup@pdx.edu

Abstract. I present an entirely rewritten version of the `outreg` command, which creates tables from the results of Stata estimation commands and generates formatted Word or LaTeX files. My objective is to provide as complete control over the layout and formatting of the estimation tables as practical in both file formats. `outreg` provides a wide range of estimation statistics, including confidence intervals and marginal effects, can control the number and arrangement of the statistics displayed, and can merge subsequent estimation results into the same table. Users can specify numeric formats, font sizes, and font types at the table cell level, as well as lines in the table and row spacing. Multiple tables can be written to the same document, making it possible to create a fully formatted statistical appendix from a `.do` file. The numerous formatting options for `outreg` command are demonstrated in examples.

Keywords: st0001, `outreg`, tables, estimation, formatting, Word, TeX

1 Introduction

My goal in creating a new version of `outreg` was to make it possible to create a fully formatted statistical appendix in a Word or LaTeXdocument from a single `.do` file, which would not require any tweaking after it was created. To this end, `outreg` has a number of capabilities that I don't think exist in Stata yet.

- `outreg` can create any (rectangular) arrangement of 26 statistics based on coefficient or marginal effect estimates.
- Addition of any summary statistics.
- Users can specify fonts and font sizes for each cell of the table, horizontal and vertical bounding lines, and spacing between table rows, if they wish.
- Cells can span multiple columns.
- Successive tables can be added to the same document, with paragraphs of regular text in between.
- With some effort, users can add footnotes or other symbols to any part of the table.
- Users can include Greek letters and other Unicode symbols in their text.

Although this version shares the basic command syntax of previous versions of `outreg` (Gallup 1998, Gallup 1999, Gallup 2000, Gallup 2001), the code has been entirely rewritten, mostly in Mata. The Mata string matrix is a compact memory structure well suited to holding the contents of the table. Mata is more concise and powerful than the Stata macro language, and runs fast.

Most of the capabilities of `outreg` come from `frmttable`, a programmers' command which takes a generic Stata matrix of statistics and converts it, along with numerous formatting options and accompanying text, into a Word or TEX[1] table. In the future, I plan to complete additional Stata commands based on `frmttable` to create formatted tables of frequencies (`outtab`) and summary statistics (`outstat`). The tables created by these commands can be added to documents containing `outreg` tables, or even merged with `outreg` tables.

This version of `outreg` has a simple syntax for typical regression tables, but it has a very large number of options to enable users to make fine adjustments to the layout and formatting (analogous, on a much smaller scale, to the plethora of `graph` options in Stata). Because the desciption of `outreg`'s options is long, I show a number of examples of its use right after the description of `outreg`'s syntax. Unlike typical Stata documentation, I leave the detailed explanation of all the options to the end, since it goes on so long.

2 Syntax and description

`outreg using` *filename* [`,` *options*]

By default, `outreg` arranges the results of Stata estimation commands in tables as they are typically presented in journal articles, rather than as they are presented in the Stata Results window. By default, t statistics appear in parentheses below the coefficient estimates with asterisks for significance levels.

`outreg` works after any estimation command in Stata (see [R] **estimation commands** for a complete list[2]). Like [R] **predict**, `outreg` makes use of internally saved estimation results, so it should be invoked after the estimation.

`outreg` creates a Microsoft Word file by default, or a TEX file using the `tex` option. In addition, the table created by `outreg` is displayed in the Results window, minus some of the finer formatting destined for the Word or TEX file.

Successive estimation results, which may use different variables, can be combined by

1. For brevity, from here on I will refer to LATEX documents as TEX documents.
2. To be precise, `outreg` can display results after every estimation command which saves both `e(b)` and `e(V)` values. Estimation commands which do not save both `e(b)` and `e(V)` are [R] **ca**, [R] **candisc**, [R] **discrim**, [R] **exlogistic**, [R] **expoisson**, [R] **factor**, [R] **mca**, [R] **mds**, [R] **mfp**, [R] **pca**, [R] **procrustes**, [SVY] **svy:tabulate**. On the other hand, `outreg` *can* display the results of the commands [R] **mean**, [R] **ratio**, [R] **proportion**, and [R] **total** which may not be thought of as estimation commands, and these commands accept the **svy:** prefix.

`outreg` into a single table using the `merge` option. (*n.b.* In previous versions of `outreg`, the `merge` option was called "append".)

Below is a listing of all of `outreg`'s options, with one-sentence descriptions of their action. This is followed by a set of examples of `outreg` in use in Section 3. The full explanation of the options is put off until Section 4 to avoid losing the reader in the details of the numerous options. In the table below, the subsections of Section 4 are listed next to the options categories to help readers find the detailed explanations of options.

Categories	Description	Section
Estimate selection	which statistics are displayed in table	4.1
Estimates formatting	numerical formatting & estimate arrangement	4.2
Text additions	titles, notes, added rows and columns	4.3
Text formatting:		
Column formatting	column widths, justification, etc.	4.4
Fonts	font specifications for table	4.5
Lines & spaces	horizontal and vertical lines, cell spacing	4.6
File & display options	TEX files, merge, replace, etc.	4.7
Stars options	change stars for statistical significance	4.8
Brackets options	change brackets around, e.g., *t* stats	4.9
Summary stats options	summary statistics below estimates	4.10
`frmttable` *options*	technical options passed to `frmttable`	4.11
Inline text formatting	superscripts, italics, Greek characters, etc.	4.12
Notes about specific estimation commands		4.13

4.1 *Estimate selection*

`se`	report standard errors rather than *t* statistics
`marginal`	report marginal effects instead of coefficients
`or` | `hr` | `irr` | `rrr`	odds ratios, that is, `exp(b)` instead of `b`
`stats(`*statname*`)`	select from 26 statistics (not just `b` and *t* stats)
`nocons`	drop constant estimate (don't include `_cons` coefficient)
`keep(`*eqlist* |*varlist*`)`	include only specified coefficients, or reorder them
`drop(`*eqlist* |*varlist*`)`	exclude specified coefficients
`level(#)`	set level for confidence intervals; default is `level(95)`

4.2 *Estimates formatting*

bdec(*numlist*)	decimal places for coefficients
tdec(*#*)	decimal places for *t* statistics
sdec(*numgrid*)	decimal places for all statistics
bfmt(*fmtlist*)	numerical format for coefficients
sfmt(*fmtgrid*)	numerical format for all statistics
nosubstat	don't put *t* statistics (or others) below coefficients
eq_merge	merge multi-equation coefficients into multiple columns

4.3 *Text additions*

varlabels	use variable labels as row titles
title(*textcolumn*)	put title above table
ctitles(*textgrid*)	headings at top of columns
rtitles(*textgrid*)	headings to the left of each row
note(*textcolumn*)	put note below table
pretext(*textcolumn*)	regular text placed before the table
posttext(*textcolumn*)	regular text placed after the table
nocoltitl	no column titles at all
norowtitl	no row titles at all
addrows(*textgrid*)	add rows at bottom of table
addrtc(*#*)	number of row title columns in addrows
addcols(*textgrid*)	add columns to right of table
annotate(*Stata matrix name*)	grid of annotation locations
asymbol(*textrow*)	symbols for annotations

4.4 *Column formatting*

colwidth(*numlist*)†	change column widths
multicol(*numtriple*[*;numtriple...*])	have column titles span multiple columns
coljust(*cjstring*[*;cjstring...*])	column justification: left, center, right, or decimal
nocenter	Don't center table within page

†Word-only option

4.5 *Font specification*

`basefont(`*fontlist*`)`	change the base font for all text
`titlfont(`*fontcolumn*`)`	change font for table title
`ctitlfont(`*fontgrid*`[;`*fontgrid...*`])`	change font for column titles
`rtitlfont(`*fontgrid*`[;`*fontgrid...*`])`	change font for row titles
`statfont(`*fontgrid*`[;`*fontgrid...*`])`	change font for statistics in body of table
`notefont(`*fontcolumn*`)`	change font for notes below table
`addfont(`*fontname*`)`†	add a new font type
`plain`	plain text - one font size, no justification
`outreg` table sections	explanation of `outreg` table sections

†Word-only option

4.6 *Border lines and spacing*

`hlines(`*linestring*`)`	horizontal lines between rows
`vlines(`*linestring*`)`	verticle lines between columns
`hlstyle(`*lstylelist*`)`†	change style of horizontal lines (e.g. double, dashed)
`vlstyle(`*lstylelist*`)`†	change style of verticle lines (e.g. double, dashed)
`spacebef(`*spacestring*`)`	put space above cell contents
`spaceaft(`*spacestring*`)`	put space below cell contents
`spaceht(#)`	change size of `spacebef` & `spaceaft`

†Word-only option

4.7 *File and display options*

`tex`	write a TEX file instead of the default Word file
`merge`	merge as new columns to existing table
`replace`	replace existing file
`addtable`	write a new table below an existing table
`append`	append as new rows below an existing table
`fragment`‡	create TEX code fragment to insert into TEX document
`nodisplay`	don't display table in Results window
`dwide`	display all columns in Results, however wide

‡TEX-only option

4.8 *Stars options*

starlevels(*numlist*)	significance levels for stars
starloc(#)	locate stars next to which statistic (default=2)
margstars	calculate stars from marginal effects, not coefficients
nostars	no stars for significance
nolegend	no legend explaining significance levels
sigsymbols(*textrow*)	symbols for significance (in place of stars)

4.9 *Brackets options*

squarebrack	square brackets instead of parentheses
brackets(*textpair*[*textpair*...])	symbols with which to bracket substatistics
nobrket	put no brackets on substatistics
dbldiv(*text*)	symbol separating double statistics ("-")

4.10 *Summary statistics options*

summstat(*e_values*)	additional summary statistics below coefficients
summdec(*numlist*)	decimal places for summary statistics
summtitles(*textgrid*)	row titles for summary statistics
noautosumm	no automatic summary stats (R^2,N)

4.11 `frmttable` *options*

blankrows	allow (don't drop) blank rows in table
nofindcons	don't assign _cons to separate section of table

Sections 4.12 (Inline formatting) and 4.13 (Command notes) contain no options.

❑ **Technical note**

Differences between the implementation of Word and TEX tables

Almost all formatting capabilities of `outreg` are implemented in a similar way for Word and TEX files, aside from minor variations due to the peculiarities of the file formats. There are two exceptions of capabilities implemented for Word tables alone. The first is decimal justification of numbers, for reasons discussed in the explanation of the `coljust` option. The second is user-specified fonts. Arbitrary fonts appear to be implementable in TEX, but I didn't pursue this because in my experience most TEX users prefer the distinctive TEX fonts.

❑

❑ **Technical note**

Under the hood

`outreg.ado` creates statistics from the `e(b)` and `e(V)` estimation result matrices (or the marginal effects matrices if marginal statistics are specified) and some of the other saved estimation results. The statistics are put in a Stata matrix that is passed to `frmttable.ado` along with information about the dimensions of the statistics: how many different statistics the user has specified and how many are double statistics (like confidence intervals). `outreg` also sends a matrix of indicators for asterisk symbols, and reorganizes numberical formatting options in the more general form accepted by `frmttable`. Decimal place options `bdec` and `tdec` are rearranged into the more general, but less convenient, `sdec` option, and number formats `bfmt` are rearranged as an `sfmt` option.

The summary statistics (either default or specified in the `summstat` option) are converted to set of strings passed as an `addrow` option to `frmttable` (and prepended to any user requested `addrow`).

The heavy lifting of creating the tables is done by `frmttable`. `frmttable` converts the statistics matrix to a Mata string matrix and adds on asterisks, brackets, and summary statistics (as added rows). Row and column titles, either derived from the variable names or user-specified, are added to the Mata string matrix. `frmttable` holds a Mata `struct` of string matrices for the `pretext` (regular text above the table), title rows, table body, notes rows, and `posttext` (regular text below the table). These matrices and some additional information about where column and row titles begin and end remain in memory for future use in the Mata `struct`.

The contents of the table are combined with formatting information (passed by `outreg`) from numerous options specifying fonts, lines, spaces, etc., and written to a Word RTF file (or a LaTeX file with the `tex` option). The table is then displayed in the Stata results window incorporating some, but not all of the formatting specification (e.g. not font sizes).

Because the contents of the table are kept in the Mata `struct _FrmtT` which persists in memory (until the Stata session ends), they are available for merging with additions to the table, such as subsequent estimations.

3 Examples of `outreg` in use

The table below lists the topics covered by the examples.

Section	Example
3.1	Basic usage and variable labels
3.2	Decimal places for coefficients and titles
3.3	Merging estimation tables together
3.4	Standard errors, no stars, and square brackets in a TEX file
3.5	10% significance level and summary statistics
3.6	Display some but not all coefficients
3.7	Add statistics not in summstat
3.8	Multi-equation models
3.9	Marginal effects and star options
3.10	Multi-column `ctitles`; merge variable means to estimation results
3.11	Specifying fonts
3.12	Superscripts, italics, and Greek characters
3.13	Place additional tables in same document
3.14	Place footnotes among coefficients
3.15	Show statistics side-by-side, like Stata estimation results

3.1 Basic usage and variable labels

`outreg` is used after an estimation command, like [R] **predict**, because it needs the saved estimation results to construct a formatted table. Consider a regression using Stata's `auto.dta` dataset:

```
. sysuse auto, clear
(1978 Automobile Data)
. reg mpg foreign weight
```

Source	SS	df	MS		
Model	1619.2877	2	809.643849	Number of obs	= 74
Residual	824.171761	71	11.608053	F(2, 71)	= 69.75
				Prob > F	= 0.0000
Total	2443.45946	73	33.4720474	R-squared	= 0.6627
				Adj R-squared	= 0.6532
				Root MSE	= 3.4071

mpg	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
foreign	-1.650029	1.075994	-1.53	0.130	-3.7955	.4954422
weight	-.0065879	.0006371	-10.34	0.000	-.0078583	-.005317
_cons	41.6797	2.165547	19.25	0.000	37.36172	45.99768

The simplest form of `outreg` just requires an output file name, in this case "`auto`".

```
. outreg using auto
```

	mpg
foreign	-1.650
	(1.53)
weight	-0.007

```
                (10.34)**
_cons            41.680
                (19.25)**
R2                0.66
N                   74
------------------------------
  * p<0.05; ** p<0.01
```

The `outreg` command creates a new Word file named `auto.doc` and displays an approximation of the Word table in the Stata Results window (which can be turned off with `nodisplay`).

`outreg` can also create tables in TeX format with the `tex` option.

The option `varlabels` replaces variable names with their labels, so that the independent variable `mpg` listed above the column of regression coefficients uses the label "`Mileage (mpg)`", the variable `foreign` uses its label "`Car type`", etc. The user can customize the variable labels before invoking `outreg` to provide the desired captions in the `outreg` table. Alternatively, the user can specify column and row titles directly with `ctitles` and `rtitles`.

Since the file `auto.doc` already exists from the previous `outreg` command, we must include the `replace` option as well.

```
. outreg using auto, varlabels replace
```
(*output omitted*)

The Results window display of table is omitted here. Instead, we show the resulting Word table in `auto.doc`:

	Mileage (mpg)
Car type	-1.650
	(1.53)
Weight (lbs.)	-0.007
	(10.34)**
Constant	41.680
	(19.25)**
R^2	0.66
N	74

* $p<0.05$; ** $p<0.01$

3.2 Decimal places for coefficients and titles

The regression table in the previous example would be improved by formatting the coefficient values and adding informative titles. By default the regression coefficients are shown with three decimal places in `outreg` tables, but this isn't very satisfactory for the `weight` variable in the regression above. The `weight` coefficient is statistically significant, but only one non-zero digit is displayed. We could use the option `bdec(5)` to display 5 decimal places for all the coefficients, but we can do better. To display five decimal places of the `weight` coefficient only and two decimal places of the other coefficients, we use `bdec(2 5 2)`.

We can add a title to the table with the `title` option. As long as the title text contains no backspaces (which indicate multiple lines of title) or commas, no quotation marks are required, so we add the option `title(What kind of cars have low mileage?)`. We also change the column heading of the estimates from the name of the independent variable to "`Base case`" with `ctitle("",Base case)`. We need the `""` to indicate that there is no `ctitle` in the left-most column of the table. We can get away with no parentheses around "`Base case`" because there is no "`,`" or "`\`" in the title, which are interpreted by `ctitles` as column and row delimiters.

```
. outreg using auto, bdec(2 5 2) varlabels replace ///
    title(What cars have low mileage?) ctitle("",Base case)
```
(*output omitted*)

If you run the commands above and open the resulting file `auto.doc` in Word or most other word-processing software, you can see the following formatted table:

What cars have low mileage?

	Base case
Car type	-1.65
	(1.53)
Weight (lbs.)	-0.00659
	(10.34)**
Constant	41.68
	(19.25)**
R^2	0.66
N	74

* $p<0.05$; ** $p<0.01$

3.3 Merging estimation tables together

Users often want to include several related estimations in the same table. `outreg` can combines multiple estimation results with the `merge` option.

We create a new variable `weightsq` for the second regression.

```
. gen weightsq = weight^2
. label var weightsq "Weight squared"
```

Then we run the second regression with the quadratic `weightsq` term.

```
. regress mpg foreign weight weightsq
```
(*output omitted*)

We add the second regression results to the regression table in Example 2 with the `merge` option. In the second regression, the `weightsq` term is statistically significant but very small due to the small units used for `weight` (pounds). We can avoid displaying a very large number of decimal places by formatting the `weightsq` coefficient in scientific notation with the option `bfmt(f f e f)`. We also specify the number of decimal places for each coefficient as in the first regression, and add an informative column title with the options `bdec(2 5 2)` and `ctitle("",Quadratic mpg)`. Note that although there are four coefficients (counting the constant), there are only three numbers in `bdec(2 5 2)`. The last number in `bdec`, 2, applies to all the remaining coefficients.

```
. outreg using auto, bdec(2 5 2) bfmt(f f e f) ctitle("",Quadratic mpg) ///
    varlabels merge
```
(*output omitted*)

The coefficients and t statistics for the variables are aligned correctly in the merged table, and the scientific notation applied to the `weightsq` variable.

What cars have low mileage?

	Base case	Quadratic mpg
Car type	-1.65	-2.20
	(1.53)	(2.08)*
Weight (lbs.)	-0.00659	-0.01657
	(10.34)**	(4.18)**
Weight squared		1.59e-06
		(2.55)*
Constant	41.68	56.54
	(19.25)**	(9.12)**
R^2	0.66	0.69
N	74	74

* $p<0.05$; ** $p<0.01$

Note that since the first `outreg` table from Example 2 used `varlabels`, we need

to use `varlabels` in the `outreg` command that merges the second regression. If not, the row titles would differ between the original table and the new results being merged and the coefficients would not be aligned correctly. For example, the label for the first coefficient in the original table is "`Car type`". Without the `varlabels` option in the `outreg` commmand above merging the new results, the first coefficient of the second regression would be labeled "`foreign`" and would be treated as new variable instead of being aligned in the first row with "`Car type`".

3.4 Standard errors, brackets, and no asterisks in a TEX table

Economics journals often prefer standard errors to `t` statistics and don't use asterisks to denote statistical significance. The `se` option replaces `t` statistics with standard errors, and the `nostar` option suppresses asterisks. We will also replace the parentheses around the standard errors with square brackets using the `squarebrack` option, and save the document as a TeX file with the `tex` option. Note that the decimal places specified by the `bdec` option apply to both the coefficients and the standard errors.

```
. regress mpg foreign weight
  (output omitted)
. outreg using auto, se bdec(2 5 2) squarebrack nostars replace tex ///
     varlabels title("No t statistics, please"\"We´re economists")
  (output omitted)
```

No t statistics, please
We're economists

	Mileage (mpg)
Car type	-1.65
	[1.08]
Weight (lbs.)	-0.00659
	[0.00064]
Constant	41.68
	[2.17]
R^2	0.66
N	74

3.5 10% significance level and summary statistics

The cutoff levels for stars indicating statistical significance can be modified with the `starlevels` option. The default levels are one star for 5% significance and two stars for 1% significance (i.e. `starlevels(5 1)`). To add a symbol for 10% significance, we use the option `starlevels(10 5 1)`. This would display 1 star for 10%, 2 for 5%, and 3 for 1%. To retain the original number of stars for 5% and 1% levels, but add a cross for the 10% level, we can use the option `sigsymbols(+,*,**)` with the symbols corresponding

to the significance levels in `starlevels`. The legend at the bottom of the `outreg` table is modified to reflect these options.

The default summary statistics are the R^2 (if it's defined, that is, only for linear regressions) and the number of observations. Instead, we display the F statistic and the adjusted R^2 using the `summstat` option. The symbols used for these statistics in the estimates return values are "`F`" and "`r2_a`". All available return values after an estimation can be seen with the command `ereturn list`. The `summstat(F\r2_a)` option is specified with a backslash separating the statistics because we want them to be on different rows in the same column (if we used a comma to separate the values, they would be on the same row in different columns, making the table one column wider). We also specify the names of the statistics in `summtitle(F statistic\Adjusted R-squared)`, similarly to `rtitles`. To give the `F` statistic one decimal place and the adjusted R^2 two decimal places, we use the option `summdec(1 2)`.

```
. reg mpg foreign weight turn
  (output omitted)
. outreg using auto, bdec(2 5 3 2) varlabels replace        ///
     starlevels(10 5 1) sigsymbols(+,*,**) summstat(F\r2_a) ///
     summtitle(F statistic\Adjusted R-squared) summdec(1 2) ///
     title(Summary statistics and\10% significance level)
  (output omitted)
```

Summary statistics and
10% significance level

	Mileage (mpg)
Car type	-2.08
	(1.85)+
Weight (lbs.)	-0.00560
	(5.59)**
Turn Circle (ft.)	-0.235
	(1.28)
Constant	48.13
	(8.78)**
F statistic	47.5
Adjusted R-squared	0.66

+ p<0.1; * p<0.05; ** p<0.01

3.6 Display some but not all coefficients

The options `keep` and `drop` allow you to display some but not all coefficients in the estimation. `keep` also allows you to change the order in which the coefficient estimates are displayed. To `keep` or `drop` the constant term, include "`_cons`" in the list of coefficients.

The first example removes dummy variable coefficients and reorders the coefficients with `keep(weight foreign)`:

```
. tab rep78, gen(repair)
  (output omitted)
. regress mpg foreign weight repair1-repair4
  (output omitted)
. outreg using auto, keep(weight foreign) varlabels replace ///
    note(Coefficients for repair dummy variables not shown)
```

	Mileage (mpg)
Weight (lbs.)	-0.006
	(9.16)**
Car type	-2.923
	(2.18)*
R2	0.69
N	69

* p<0.05; ** p<0.01
Coefficients for repair dummy variables not shown

The second example uses `keep` to remove from the table the auxiliary parameters included in `e(b)` by Stata. The [R] **tobit** command estimates a `sigma` parameter. The main coefficient estimates are included in the `e(b)` vector with the equation name "`model`" and the sigma parameter is given the equation name "`sigma`".

When in doubt about which equation names are included in the `e(b)` vector after an estimation, you can view the matrix and its names with the `matrix list e(b)` command. By default, `outreg` includes the sigma parameter and the equation names in the estimates table.

```
. gen wgt = weight/100
. label var wgt "Weight (lbs/100)"
. tobit mpg wgt, ll(17)
  (output omitted)
. outreg using auto, replace
```

model	wgt	-0.687
		(9.82)**
	_cons	41.499
		(20.16)**
sigma	_cons	3.846
		(10.50)**
N		74

* p<0.05; ** p<0.01

To limit the table to the coefficient estimates alone, we can use the option `keep(model:)`. The colon after "`model`" indicates that it is an equation name, not a coefficient name, and all estimates in the "`model`" equation are kept.

```
. outreg using auto, keep(model:) varlabel replace
```

	Mileage (mpg)
Weight (lbs/100)	-0.687
	(9.82)**
Constant	41.499
	(20.16)**
N	74

* p<0.05; ** p<0.01

3.7 Add statistics not in `summstat`

There are many statistics, particularly test statistics, which we may want to report in estimation tables but are not available in the `summstat` option. The statistics available in `summstat` are limited to the `e( )` scalar values that can be viewed after an estimation command with `ereturn list`.

The `addrows` option can add additional rows of text below the coefficient estimates and summary statistics. This example shows how to display the results of the [R] **test** command as addeds rows of the `outreg` table.

Below we test whether the coefficient on the variable `foreign` is equal to the negative of the coefficient on `goodrep` with `test foreign = -goodrep`. The command [R] **test** saves the F statistic in the return value `r(F)` and its p value in the return value `r(p)`. If we include `r(F)` and `r(p)` in `addrows` directly, they are reported with seven or eight decimal places. To control the numerical formatting of the return values `F` and `p`, we use the local macro directive `display`. `local F : display %5.2f ‘r(F)’` takes the value in `r(F)` and puts it in the local macro "`F`" displayed with two decimal places and a width of 5. Similarly, the local macro "`p`" has three decimal places.

```
. gen goodrep = rep78==5
. reg mpg weight foreign goodrep
  (output omitted)
. test foreign = -goodrep
  (output omitted)
. local F : display %5.2f `r(F)´
. local p : display %4.3f `r(p)´
```

We are now ready to add the test statistics to the `outreg` table. The `addrows` option below adds two rows, one for the F test and one for its p value, and two columns, one for the text in the left column and one for the test values. As usual, columns of text are separated with a comma, and rows of text are separated with the backslash.

```
. outreg using auto, replace ///
    addrows("F test: foreign = -goodrep","`F´"\"p value","`p´")
```

	mpg
weight	-0.006
	(10.40)**
foreign	-2.745
	(2.53)*

goodrep	3.613
	(2.98)**
_cons	40.733
	(19.59)**
R2	0.70
N	74
F test: foreign = -goodrep	0.43
p value	0.515

* p<0.05; ** p<0.01

If we wanted to report the F test statistics above the summary statistics (R^2 and N), then we would need to use the option `noautosumm` to suppress the default summary statistics, and instead include them in the `addrows` option below the F test statistics. The values of R^2 and N are available in the scalars `e(r2)` and `e(N)`.

3.8 Multi-equation models

`outreg` displays estimation results in a single column even for multi-equation models unless the user chooses the `eq_merge` option (for "equation merge"). When different equations in the estimation model share many of the same covariates, users may prefer to display the results like the merged results of separate estimations. `eq_merge` puts each equation is a separate column and any common variables are displayed the same row. Using an example of seemingly unrelated regression estimation with the three equations each sharing two covariates, `outreg` organizes the table as shown below.

```
. sureg (price foreign weight length) (mpg displ = foreign weight)
  (output omitted)
. outreg using auto, varlabels eq_merge replace ///
    ctitles("",Price Equation,Mileage Equation,Engine Size Equation) ///
    summstat(r2_1,r2_2,r2_3\N,N,N) summtitle(R2\N)
```

	Price Equation	Mileage Equation	Engine Size Equation
Car type	3,575.260	-1.650	-25.613
	(5.75)**	(1.57)	(2.05)*
Weight (lbs.)	5.691	-0.007	0.097
	(6.18)**	(10.56)**	(13.07)**
Length (in.)	-88.271		
	(2.81)**		
Constant	4,506.212	41.680	-87.235
	(1.26)	(19.65)**	(3.47)**
R2	0.55	0.66	0.81
N	74	74	74

* p<0.05; ** p<0.01

Each of the equations in `sureg` has an R^2 statistic. The `summstat` option places them below the coefficient estimates along with the number of observations. The `summstat` option has three columns and two rows.

3.9 Marginal effects and star options

`outreg` can display marginal effects estimates calculated by the [R] **mfx** command instead of coefficient estimates. This requires that the user run `mfx compute` after the estimation in question before using `outreg`.

The simplest way to substitute marginal effects for coefficient estimates is with the `marginal` option. This replaces the statistic `dydx` for `b` and `t_abs_dfdx` for `t_abs` (or `se_dfdx` for `se` if the *option* `se` is in effect). The asterisks for significance now refer to the marginal effects rather than the underlying coefficients.

```
. logit foreign wgt mpg
  (output omitted)
. mfx compute
  (output omitted)
. outreg using auto, marginal replace
```

	foreign
wgt	-0.052
	(3.73)**
mpg	-0.022
	(1.76)
N	74

* p<0.05; ** p<0.01

Marginal effects can also be combined with regression coefficients or other statistics in the `outreg` table. Below, the table displays each coefficient estimate with the marginal effect below it, and the 95% confidence interval of the marginal effect below that, because of the `stats(b dydx ci_dfdx)` option. The `margstar` option specifies that the asterisks refer to the significance of the hypothesis that the marginal effects are zero, rather than the coefficients being zero. The `starloc(3)` option places the asterisks next to the third statistic (the marginal effect confidence intervals) instead of the default, next to the second statistic.

```
. outreg using auto, stat(b dydx ci_dfdx) replace     ///
    title("Marginal Effects & Confidence Intervals"\ ///
          "Below Coefficients") margstar starloc(3)
```

Marginal Effects & Confidence Intervals
Below Coefficients

	foreign
wgt	-0.391
	(-0.052)
	[-0.079 - -0.025]**
mpg	-0.169
	(-0.022)
	[-0.047 - 0.003]
_cons	13.708
N	74

* p<0.05; ** p<0.01

3.10 Multi-column ctitles; merge variable means with estimation results

Empirical papers commonly report summary statistics for the variables used in estimations, usually their means and standard deviations. This example shows how to merge variable means onto an estimation table, and how to make column titles that span multiple columns.

First we create an `outreg` table which merges two simple regressions as was done in section 3.3. The `nodisplay` option suppresses display of the intermediate `outreg` tables we are creating, which normally appear in the Stata results window. The `ctitles` have been specified to have two rows, with a supertitle on the first two columns of "`_____{\ul Regressions}_____`". The reason for the "`{\ul }`" code is to underline "`Regressions`" (inline formatting codes are explained in more detail in sections 3.12 and 4.12). Word does not underline space characters, so the underlining is extended with the underscore ("`_`") characters. The underlining causes the "`Regressions`" supertitle to span from above the first regression column's title to the second regression column's title.

```
. reg mpg foreign weight
  (output omitted)
. outreg using auto, bdec(2 5 2) replace varlabels nodisplay   ///
     ctitles("","_______{\ul Regressions}________"¨,"Base case")
. reg mpg foreign weight weightsq
  (output omitted)
. outreg using auto, bdec(2 5 2) bfmt(f f e f) varlabels merge ///
     ctitles("",""\"","Quadratic mpg") nodisplay
```

Then we run the [R] **mean** command, which calculates variable means and their standard errors. [R] **mean** is an estimation command, so it stores its results in `e(b)` and `e(V)` and they can be displayed and merged using `outreg`. We `merge` the variable means to the `outreg` table already created above. The `ctitles` in this `outreg` command have two rows, aligning them with the previous `ctitles`. The `multicol(1,2,2)` option causes the cell in the first row, second column, to span two cells horizontally so that the title `Regressions` is centered over both the `"Base case"` and `"Quadratic mpg"` columns. See the result in the Word table below. Note that the `multicol` option must be used in the third and last `outreg` command, because it is a formatting characteristic that is not retained from an earlier `outreg` table that is `merged` with a new one. It is also necessary to specify the `colwidth` option because the column width algorithm does not account for the fact that the wide column title "`_____{\ul Regressions}_____`" spans two columns. Without the `colwidth` option, the `"Base case"` column would be too wide due to the wide supertitle.

```
. mean mpg foreign weight
  (output omitted)
. outreg using auto, bdec(1 3 0) nostar varlabels merge    ///
     ctitles("","Means &"\"","Std Errors") multicol(1,2,2) ///
     title(Multi-column ctitles) colwidth(12 7 11 8)
```

The [R] **mean** command calculates the variable means and their standard *errors*.

Multi-column ctitles

	Regressions		Means &
	Base case	Quadratic mpg	Std Errors
Car type	-1.65	-2.20	0.297
	(1.53)	(2.08)*	(0.053)
Weight (lbs.)	-0.00659	-0.01657	3,019
	(10.34)**	(4.18)**	(90)
Weight squared		1.59e-06	
		(2.55)*	
Mileage (mpg)			21.3
			(0.7)
Constant	41.68	56.54	
	(19.25)**	(9.12)**	
R^2	0.66	0.69	
N	74	74	74

* $p<0.05$; ** $p<0.01$

More typically, summary statistic tables report the variable means and their standard deviations (which differ from the standard errors of the mean by a factor of the square root of *N*). To report the standard deviations of the variables, I use the as yet unreleased command `outstat` which, since it is also based on the underlying formatting engine `frmttable`, can be appended to an `outreg` table:

```
. reg mpg foreign weight
  (output omitted)
. outreg using auto, varlabels replace
  (output omitted)
. outstat mpg foreign weight using auto, merge sdec(2\2\4\4\0\0) ///
     title(Merge summary statistics with regression results)     ///
     varlabels
(note: tables being merged have different numbers of row sections)
```

The warning message "`tables being merged have different numbers of row sections`" is displayed because the differing structure of the `outreg` table and the `outstat` table mean that the `merge` process may not align rows the way the user intended, but in this case it does.

3.11 Specifying fonts

One of the objectives of this version of `outreg` is to have as complete control of the layout and appearance of estimates tables as possible. An important element of this is fine control of fonts. `outreg` now enables users to specify fonts down to the table cell level, although this is needed only rarely. Users can specify font sizes, font types (such

Merge summary statistics with regression results

	Mileage (mpg)	Means
Car type	-1.650	0.2973
	(1.53)	(0.4602)
Weight (lbs.)	-0.007	3,019
	(10.34)**	(777)
Mileage (mpg)		21.30
		(5.79)
Constant	41.680	
	(19.25)**	
R^2	0.66	
N	74	

* $p<0.05$; ** $p<0.01$

as Times Roman or Arial), and font styles (such as bold or italic). For Word files, users can apply any font type installed on their computers by adding the font name in the `addfont` option.

This example prepares a table for a presentation as an overhead slide with special fonts that are displayed much larger than usual (the resulting Word table below is shrunk to not take up too much of the page). Two specialized fonts are added to the document with the `addfont(Futura,Didot Bold)` command. These fonts can then be applied to different parts of the table as "`fnew1`" for the first added font, or "`fnew2`", the second added font. We set the default font of the table to be Futura ("`fnew1`") in the `basefont(fs32 fnew1)`. This `basefont` option also sets the font size to 32 points to make the table fill the whole overhead slide. The title is assigned the second added font, Didot Bold, with a 40 point size in `titlfont(fs40 fnew2)`. The statistics in the table are displayed in the Arial font for readability with the `statfont(arial)` option. (Times Roman, Arial, and Courier fonts are predefined in Word and TEX documents and don't need to be added.) The `basefont` font characteristics apply to all parts of the table, unless otherwise specified, so the Arial font in `statfont` has a point size of 32.

Font specifications do not change the appearance of the table displayed in the Stata results window so the output is omitted. The Word table in `auto.doc` is displayed in shrunken form.

```
. reg mpg foreign weight
  (output omitted)
. outreg using auto, replace varlabels ///
     title("New Fonts for Overhead Slides") addfont(Futura,Didot Bold) ///
     basefont(fs32 fnew1) titlfont(fs40 fnew2) statfont(arial)
  (output omitted)
```

New Fonts for Overhead Slides

	Mileage (mpg)
Car type	-1.650
	(1.53)
Weight (lbs.)	-0.007
	(10.34)**
Constant	41.680
	(19.25)**
R^2	0.66
N	74

* $p<0.05$; ** $p<0.01$

3.12 Superscripts, italics, and Greek characters

This example uses some of the methods of inline formatting explained in section 4.12 to apply superscripts, italic text, and Greek characters. It is helpful to review those methods to understand the codes used here.

This example is similar to section 3.7 in that the results of a test of coefficient equality are displayed in the estimation table. However, since the estimation is nonlinear, the test statistic is a χ^2 rather than an F statistic. We will write χ^2 with the Greek character chi and a superscripted "2" in the Word file generated by `outreg`. A different set of codes can produce the same formatting in TEX files, also discussed in section 4.12.

The Word code for the Unicode representation of the Greek lower-case letter chi is "`\u0966?`" (see all Greek letter codes for Word files in section 4.12). The code for chi needs to be placed in quotes in the `addrows` option because otherwise the backslash would be interpreted as a row divider. The superscripted 2 is encoded as "`{\super 2}`". Note the space between the formatting code ("`{\super`") and the regular text ("2"). Without it, Word would try to interpret the code "`\super2`", which doesn't exist. Finally, we italicize the "p" in `p value` like this: "`{\p}`". The full `addrows` option becomes `addrows("\u0966{\super 2}test","`‘`chi2`’`"¨{\i p}value","`‘`p`’`")`. As in section 3.7, 'chi2' and 'p' are the value of local macros containing the numerically formatted values of the χ^2 statistic and its p value.

The `note` option in the `outreg` command below has a couple of tricks in it. The first is a blank row (`""`) to separate the `note` text from the legend for asterisks above it. We also add Stata system macro values for the current time, date, and dataset file name from predefined Stata macros
t $S_TIME, `$S_DATE`, and `$S_FN`, respectively.

```
. logit foreign wgt mpg
```
(*output omitted*)

```
. test wgt = mpg
  (output omitted)
. local chi2 : display %5.2f `r(chi2)´
. local p : display %4.3f `r(p)´
. outreg using auto, replace colwidth(12 10) varlabels               ///
    addrows("\u0966?{\super 2} test","`chi2´"\"{\i p} value","`p´") ///
    note(""\"Run at $S_TIME, $S_DATE"\"Using data from $S_FN")      ///
    title("Greek characters, superscripts, and italics")
```

Greek characters, superscripts, and italics

	Car type
Weight (lbs/100)	-0.391
	(3.86)**
Mileage (mpg)	-0.169
	(1.83)
Constant	13.708
	(3.03)**
N	74
χ^2 test	10.84
p value	0.001

* $p<0.05$; ** $p<0.01$

Run at 16:51:05, 8 Sep 2010
Using data from /Applications/Stata/ado/base/a/auto.dta

3.13 Place additional tables in same document

One of the goals for `outreg` is to create whole documents, such as statistical appendices, from a Stata `.do` file. To do this, one must be able to write multiple tables to the same document, which is possible with the `addtable` option.

The `outreg` command is actually mostly a front end for the programmers' command `frmttable`. Other commands based on the `frmttable` formatting engine that I intend to complete soon are `outtab` which formats [R] **tabulate** output and `outstat` which formats [R] **tabstat** output. The tables created by `outtab` and `outstat` can be added to documents that have `outreg` tables in them.

The command below uses `outtab` to create a frequency table for the variable `foreign` and places it below the table just created in section 3.12 in the Word file `auto.doc`.

```
. outtab foreign using auto, addtable ///
    title("A frequency table for car type")
  (output omitted)
```

The user can add paragraphs of regular text before and after each table with the

Combine two tables in one document

	foreign
wgt	-0.391
	(3.86)**
mpg	-0.169
	(1.83)
_cons	13.708
	(3.03)**
N	74
χ^2 test	10.84
p value	0.001

* $p<0.05$; ** $p<0.01$

Run at 16:51:05, 8 Sep 2010
Using data from /Applications/Stata/ado/base/a/auto.dta

A frequency table for car type

Car type	Frequency
Domestic	52
Foreign	22
Total	74

`pretext` and `posttext` options.

3.14 Place footnotes among coefficients

Placing footnotes in any of the text elements of a `outreg` table is straightforward, such as in `title`, `ctitles`, `rtitles`, or `note`. You can place a footnote number in the text, using a superscript as in section 3.12 if you want, and place the footnote text in the `note` or `posttext`.

Placing a footnote in the body of the `outreg` table is not as straightforward as in the text elements, because the table body is made up of numeric statistics. For this, we use the `annotate` option. First we create a Stata matrix with the footnote locations used by `annotate`, and put the footnote symbols in the text string of `asymbol`. It is helpful to review the entry for the `annotate` option in section 4.3 for details.

Below, we place superscripted footnotes in a regression table. The first footnote is added to the label of the variable `foreign`, which is used by `outreg` because of the `varlabels` option. The next two footnotes are placed among the regression statistics. For this we create a Stata matrix with the `matrix annotmat = J(3,2,0)` command. This creates a 3 by 2 matrix of zeros. The matrix should have the dimension of the number of coefficients (3, including the constant) by the number of statistics (by default, 2: `b` and `t_abs`). All elements of the matrix `annotmat` which are zero are ignored. The locations with a "`1`" have the first `asymbol` appended, "`2`" have the second `asymbol`, etc. Since we want to place a footnote next to the first t statistic, we place a 1 at position (1,2) of `annotmat` for the first coefficient, second statistic of the table. We place another footnote next to the third coefficient estimate, so we place a 2 at position (3,1) of `annotmat`. The 1 and 2 in `annotmat` correspond to the first and second strings

in `asymbol`, which are "`{\super 2}`" and "`{\super 3}`" since these should be footnotes number 2 and 3.

The final footnote, 4, is placed in the text labeling the summary statistic, *N*, using the `summtitle("{\i N}{\super 2}")` which gives us an italicized *N* and a superscripted 4.

The `outreg` command below includes the `colwidth` option because `outreg` is unable to distinguish the long formatting codes (such as "`{\super 2}`") from columns with long text in it. The `colwidth` option is necessary to prevent the columns widths from being excessively wide. This problem does not occur with TEXdocuments because TEXis more sophisticated about fitting column widths.

It is not possible to position a footnote next to the summary statistic in `summstat`. To accomplish this, it is necessary to turn off the automatic summary statistics with `noautosumm` (which `summstat` does by default), and place the statistic and the footnote symbol in `addrows`, which was described in sections 3.7 and 3.12.

The footnote text is added below the table in the `note` option, with superscripts for the footnote numbers.

```
. reg mpg foreign weight
. label var foreign "Car Type {\super 1}"
. matrix annotmat = J(3,2,0)
. matrix annotmat[1,2] = 1
. matrix annotmat[3,1] = 2
. outreg using auto, annotate(annotmat) asymbol("{\super 2}","{\super 3}") ///
    summstat(N) summtitle("{\i N}{\super 4}") ///
    title("Footnotes among the coefficients") ///
    note("{\super 1} First footnote."  ///
         "{\super 2} Second footnote." ///
         "{\super 3} Third footnote."  ///
         "{\super 4} Fourth footnote.") varlabels replace colwidth(10 10)
```

(*output omitted*)

3.15 Show statistics side-by-side, like Stata estimation results

To show statistics side-by-side, such as the *t* statistics next to the coefficient rather than below it, use the `nosubstat` option. The following example creates a table similar to Stata's display of regression results, reporting six statistics using the `stats` option. Asterisks for significance have been turned off with the `nostars` option.

Note the dollar signs and backslash in the `ctitles` option to make sure that the symbols >, |, and % show up correctly in this TEXdocument (*cf* section 4.12).

```
. outreg using auto, nosubstat stats(b se t p ci_l ci_u) ctitles("mpg",  ///
         "Coef.","Std. Err.","t","P$>|$t$|$","[95\% Conf.","Interval]") ///
         title("Horizontal Output like Stata´s -estimates post-")       ///
         bdec(7) nostar replace tex
```

(*output omitted*)

The resulting TEXtable shows all statistics for each coefficient side-by-side:

Footnotes among the coefficients

	Mileage (mpg)
Car Type[1]	-1.650
	(1.53)[2]
Weight (lbs.)	-0.007
	(10.34)**
Constant	41.680[3]
	(19.25)**
N[4]	74

* $p<0.05$; ** $p<0.01$
[1]First footnote.
[2]Second footnote.
[3]Third footnote.
[4]Fourth footnote.

Horizontal Output like Stata's `estimates post`

mpg	Coef.	Std. Err.	t	P> |t|	[95% Conf.	Interval]
foreign	-1.6500291	1.0759941	-1.53	0.13	-3.7955004	0.4954422
weight	-0.0065879	0.0006371	-10.34	0.00	-0.0078583	-0.0053175
_cons	41.6797023	2.1655472	19.25	0.00	37.3617239	45.9976808

4 Options descriptions

4.1 Estimate selection

`se` specifies that standard errors rather than t statistics are reported in parentheses below the coefficient estimates. The decimal places displayed are those set by `bdec`.

`marginal` specifies that marginal effects rather than coefficients are reported. The t statistics are for the hypothesis that the marginal effects, not the coefficients, are equal to zero, and the asterisks report the significance of this hypothesis test. `marginal` is equivalent to `stats(dydx t_abs_dfdx)` (or `stats(dydx se_dfdx)` if the `se` option is used) combined with the `margstars` option.

`or`, `hr`, `irr`, or `rrr` cause the coefficients to be displayed in exponentiated form: for each coefficient, `exp(b)` rather than `b` is displayed. Standard errors and confidence intervals are also transformed. Display of the intercept, if any, is suppressed. These options are identical, but by convention different estimation methods use different names.

Exponentiation option	Name
`or`	odds ratio
`hr`	hazard ratio
`irr`	incidence-rate ratio
`rrr`	relative-risk ratio

Note that after commands such as [R] **stcox**, which report coefficients in exponentiated form by default, you must use one of the exponentiation options for the `outreg` table to display exponentiated coefficients and standard errors as they are displayed in the Results window after `stcox` command.

The exponentiation options are equivalent to the option `stats(e_b t)` (or `stats(e_b e_se)` if the `se` option is in effect).

These options correspond to the `or` option used for [R] **logit**, [R] **clogit**, and [R] **glogit** estimation, `irr` for [R] **poisson** estimation, `rrr` for [R] **mlogit**, `hr` for [R] **stcox** hazard models, and `eform` for [R] **xtgee**, but they can be used to exponentiate the coefficients after any estimation. Exponentiation of coefficients is explained in [R] **maximize - methods and formulas**.

`stats(`*statname* [`...`]`)` specifies the statistics to be displayed; the default is equivalent to specifying `stats(b t_abs)`. Multiple statistics are arranged below each other (unless you use the `nosubstat` option), with varying `brackets`. Available statistics are:

statname	Definition
`b`	coefficient estimates
`se`	standard errors of estimate
`t`	*t* statistics for the test of b=0
`t_abs`	absolute value of *t* statistics
`p`	*p* value of *t* statistics
`ci`	confidence interval of estimates
`ci_l`	lower confidence interval of estimates
`ci_u`	upper confidence interval of estimates
`beta`	normalized beta coefficients (see the beta option of [R] **regress**)
`e_b`	exponentiated form of the coefficients.
`e_se`	exponentiated standard errors
`e_ci`	exponentiated confidence interval
`e_ci_l`	exponentiated lower confidence interval
`e_ci_u`	exponentiated upper confidence interval
`dydx`	marginal effect of the coefficients (requires [R] **mfx**)
`eyex`	elasticities in the form of d(lny)/d(lnx) (see [R] **mfx**)
`eydx`	elasticities in the form of d(lny)/d(x) (see [R] **mfx**)
`dyex`	elasticities in the form of d(y)/d(lnx) (see [R] **mfx**)
`se_dfdx`	standard errors of marginal effects (or elasticities)
`t_dfdx`	*t* statistics of marginal effects (or elasticities)
`t_abs_dfdx`	absolute value of *t* statistics of marginal effects (or elasticities)
`p_dfdx`	*p* values of *t* statistics of marginal effects (or elasticities)
`ci_dfdx`	confidence interval of marginal effects (or elasticities)
`ci_l_dfdx`	lower confidence interval of marginal effects (or elasticities)
`ci_u_dfdx`	upper confidence interval of marginal effects (or elasticities)
`X`	values around which marginal effects (or elasticities) were estimated

`nocons` drops the constant estimate from the table.

`keep(`*eqlist* | *coeflist*`)` includes only the specified coefficients and reorders coefficients.

`drop(`*eqlist* | *coeflist*`)` excludes the specified coefficients.

eqlist consists of *eqname:*[*coeflist*] [*eqname:*[*coeflist*]...]. *coeflist* (coefficient list) is like a *varlist* but can include "`_cons`" for the constant coefficient, or other parameter names. The *coeflist* can include any of the simple column names of the `e(b)` coefficient vector, which forms the basis of the table created by `outreg`. You can see the contents of the `e(b)` vector after an estimation command by typing `matrix list e(b)`. If using marginal effects rather than coefficient estimates, the relevant vector is `e(Xmfx_dydx)` (or `e(Xmfx_eyex)`, etc.).

eqname is a second level column name of the `e(b)` vector used for multi-equation estimation commands, such as [R] **reg3** or [R] **mlogit**. Many Stata estimation commands attach additional parameters to the coefficient vector `e(b)` with a distinct equation name. For instance,the [R] **xtreg, fe** command includes two parameters

in `e(b)` with *eqnames* "`sigma_u:`" and "`sigma_e:`". The *coeflist* for each of these *eqnames* is "`_cons`".

To report only the coefficient estimates without additional parameters in the `e(b)` vector, it usually works to use the `keep(`*depvar*`:)` option, since the coefficients are given an *eqname* of the dependent variable.

You can use the `keep` option to reorder variables for the formatted `outreg` table. The estimation coefficients will be displayed in the order specified in `keep`. Don't forget to include "`_cons`" in the reordered *coeflist* if you want the constant coefficient term to be included in the formatted table.

If in doubt about what variable names, or especially equation names, to include in keep or drop, use `matrix list e(b)` (or `matrix list e(Xmfx_dydx)` for marginal effects) to see what names are assigned to saved estimation results.

You may have problems with `keep` and `drop` if you have chosen both coefficients and marginal effects as statistics, since they usually do not have same *coeflist* in both cases due to lack of a constant estimate in the marginal effects. A `keep` option that included "`_cons`" would result in an error message after because no constant could be found in the marginal effects. In this case, you could only `keep` or `drop` variables occurring in both vectors. However, if you are using `drop`, you can still eliminate the constant term with the `nocons` option.

`level(#)` sets the significance level for confidence intervals, which are included in the `outreg` table using the `stats(ci)` option. The default is level(95) for a 95% confidence level. Note that `level` has no impact on the asterisks for the statistical significance of coefficients (for this, see `starlevels`). For more information about `level` see [R] **estimation options - level**. The default `level` can be set for all Stata commands, including `outreg` using the [R] **set level** command.

4.2 Estimates formatting

`bdec(`*numlist*`)` specifies the number of decimal places reported for coefficient estimates (the `b`'s). It also specifies the decimal places reported for standard errors if the `se` option is in effect. The default value for `bdec` is 3. The minimum value is 0 and the maximum value is 15. If one number is specified in `bdec`, it will apply to all coefficients. If multiple numbers are specified in `bdec`, the first number will determine the decimals reported for the first coefficient, the second number, the decimals for the second coefficient, etc. If there are fewer numbers in `bdec` than coefficients, the last number in `bdec` will apply to all the remaining coefficients.

The decimal places applied to each coefficient are also applied to the corresponding standard errors, confidence intervals, beta coefficients, and marginal effects, if they are included with the `se` or `stats` options.

`tdec(#)` specifies the number of decimal places reported for t statistics. The default value for `tdec` is 2. The minimum value is 0 and the maximum value is 15.

`sdec(`*numgrid*`)` is for finer control of the decimal places of estimates than is possible with `bdec` and `tdec`, but is rarely needed. The `sdec` *numgrid* corresponds to the decimal places for each of the statistics in the table. It can be used, for instance, to specify different decimal places for coefficients versus standard errors (`bdec` applies to both), or to allow varying decimal places for *t* statistics.

numgrid is a grid of intergers 0-15 in the form used by [R] **matrix define**. Commas separate elements along a row, and backslashes ("\") separate rows: *numgrid* has the form #[,#...][\ #[,#...][\ [...]]]. For example, if the table of statistics has three rows and two columns, the `sdec(`*numgrid*`)` could be `sdec(1,2 \ 2,2 \ 1,3 \ 2,2)`. If you specify a grid smaller than the table of statistics created by `outreg`, the last rows and columns of the `sdec` *numgrid* will be repeated to cover the whole table. Unbalanced rows or columns will not cause an error. They will be filled in, and `outreg` will display a warning message.

`bfmt(`*fmtlist*`)` specifies the numerical format for coefficients. The possible format types are:

fmt code	Format type
`e`	exponential (scientific) notation
`f`	fixed number of decimals
`fc`	fixed with commas for thousands, etc. - the default for `outreg`
`g`	"general" format (see [R] **format**)
`gc`	"general" format with commas for thousands, etc.

Format type `e`, scientific notation, is the format most likely to be useful for `outreg` tables. The `g` formats do not allow the user to control the number of decimal places displayed.

frmtlist consists of *fmt* [*fmt* ...] where *fmt* is either `e`, `f`, `fc`, `g`, or `gc`. Like `bdec`, if one format is specified in `bfmt`, it will apply to all coefficients. If multiple format codes are specified in `bfmt`, the first format will apply to the first coefficient, the second format, the second coefficient, etc. If there are fewer formats in *fmt* than coefficients, the last format in `bfmt` will apply to all the remaining coefficients. The format applied to each coefficient is also applied to the corresponding standard errors, confidence intervals, beta coefficients, and marginal effects, if they are specified in `se` or `stats`.

`sfmt(`*fmtgrid*`)` is for finer control of the numerical formats of estimates than is possible with `bfmt`, but is rarely needed. The `sfmt` *fmtgrid* is a grid of the format types (`e`, `f`, `g`, `fc`, or `gc`) for each statistic in the table. For example, `sfmt` could be used to assign different numerical formats for the coefficients in different columns of a multi-equation estimation, or to change the format for *t* statistics.

The *fmtgrid* in `sfmt` has the same form as the *numgrid* of the `sdec` option above.

`nosubstat` puts additional statistics, like *t* statistics or other "sub-statistics", in columns to the right of coefficients, rather than below them. Applying the `nosubstat` with

the default statistics of `b` and `t_abs`, the `outreg` table would have one only row, but two columns, for each coefficient. For example, the command `outreg using test, nosubstat stats(b,se,t,p,ci_l,ci_u)` will arrange regression output the way it is displayed in the Stata Results window after the [R] **regress** command, with each statistic in a separate column. In this case, for each variable in the regression, there is one row of results, but six columns, of statistics (see section 3.15).

`eq_merge` merges multi-equation estimation results into multiple columns, one column per equation. By default, `outreg` displays the equations one below the other in a single column. `eq_merge` is most useful after estimation commands like [R] **reg3**, [R] **sureg**, [R] **mlogit**, and [R] **mprobit**, where many or all of the variables recur in each equation. The coefficients are merged as if the equations were estimated one at a time, and the results were sequentially combined with the `merge` option.

4.3 Text additions

`varlabels` replaces variable names with variable labels (see [R] **label**), if they exist. For example, if using the `auto.dta` data set, `varlabel` gives a coefficient for the `mpg` variable the row title `"Mileage (mpg)"` instead of `"mpg"`. `varlabels` also replaces "`_cons`" with `"Constant"` for constant coefficients.

Text structures used for titles

textcolumn is `"`*string*`"` [`\"`*string*`"`...]

textrow is `"`*string*`"` [`,"`*string*`"`...]

textgrid is `"`*string*`"` [`,"`*string*`"`...] [`\ "`*string*`"`[`,"`*string*`"`...] [`\ `[...]]] or a *textrow* or a *textcolumn* as a special case

`"`*string*`"` [`"`*string*`" `...] will often work in place of a *textrow* or a *textcolumn* when the user's intent is clear, but if in doubt use the proper *textrow* or *textcolumn* syntax above.

`title(`*textcolumn*`)` specifies a title or titles above the regression table. Subtitles should be separated from the primary titles by backslashes ("\"), like this: `title("Main Title"\"First Sub-Title"\"Second Sub-Title")`. By default, titles are set in a larger font than the body of the table. If title text does not contain backslashes, you can dispense with the quotation marks, but if in doubt, include them.

`ctitles(`*textgrid*`)` specifies the column titles above the estimates. By default if no `ctitles` are specified, the name of the dependent variable is displayed. A simple form of `ctitles` is, for example, `ctitles("Variables","First Regression")`. Note that the first title in `ctitles` goes above the variable name column and the second title goes above the estimates column. If you want no heading above the variable name column, specify `ctitles("","First Regression")`.

Fancier titles in `ctitles` can have multiple rows. These are specified as a *textgrid*. For example, to put a number above the title for the estimation method (in prepara-

tion for merging additional estimation results), one could use `ctitles("","Regression 1" \ "Independent Variables","OLS")`. The table would now have a first column title of `"Regression 1"` above the coefficients estimates, and a second column title of `"OLS"` in the row below.

See section 3.10 for an application of multi-row `ctitles`.

The option `nocoltitl` removes even the default column titles.

`rtitles(`*textgrid*`)` replaces the leftmost column of the table with new row titles for the coefficient estimates. By default (with no `rtitles` option), the row titles are variable names. Multiple titles for the leftmost column in `rtitles` should be separated by "\" since they are placed below one another (if the titles are separated with commas, they will all be placed in the first row of the estimates). An example of `rtitles` is `rtitles("Variable 1"\""\"Variable 2"\""\"Constant")`. The empty titles `""` are to account for the t statistics below the coefficients.

Multicolumn `rtitles` are possible, and will be merged correctly with other estimation results. Multicolumn `rtitles` occur by default, without a specified `rtitle`, after multi-equation estimations, where the first `rtitle` column is the equation name, and the second `rtitle` column is the variable name within the equation. See the second part of section 3.6 for a `outreg` table showing this.

The option `norowtitl` removes even the default `rtitles`.

`note(`*textcolumn*`)` specifies a note to be displayed below the `outreg` table. Multiple lines of a note should be separated by backslashes ("\"), like this: `note("First note line."\"Second note line."\"Third note line.")`. Notes are centered immediately below the table. By default, they are set in a smaller font than the body of the table. Blank note lines (`""`) are possible to insert space between `note` rows.

`pretext(`*textcolumn*`)` regular text placed before the table.

`posttext(`*textcolumn*`)` regular text placed after the table.

`pretext` and `posttext` contains regular paragraphs of text to be placed before or after the `outreg` table in the document created. This allows a document to be created with regular paragraphs between the tables. The default font is applied, which can be changed with the `basefont` option. Text is left justified and spans the whole page.

Multiple paragraphs can be separated by the backslash character: `pretext("Paragraph 1" \ "Paragraph 2")`.

When creating a Word document, you can create blank lines with empty paragraphs: e.g. `posttext(""\""\"This is text")` would create two blank lines before the paragraph `"This is text"`.

For Word documents, you can also use the code "`\line`" for blank lines. You can insert page breaks between tables with the Word code "`\page`" in `pretext("\page")`,

which is useful when placing multiple tables within one document with the `addtable` option. The page break or line break codes can be used within a text string, but they must have a space between the codes and the subsequent text: e.g. `pretext("\page\line This is text")`. Without the space, in `pretext("\page\lineThis is text")`, Word would try to interpret the code "`\lineThis`" which is not defined.

When creating a TEX document (using option `tex`), you can insert blank lines using the code "`\bigskip`" (the trick used above for Word files of inserting blank paragraphs does not work for TEX files). You can insert page breaks between tables with the code "`\pagebreak`", as in `pretext("\pagebreak")`, which is useful with the `addtable` option to put each table on a separate page. The page break or line break codes must be in separate rows from text: e.g. `pretext("\pagebreak\bigskip" \"This is text")`.

`nocoltitl` ensures that there are no column titles - the default column title of the dependent variable name is not used. To replace the column headings, instead of eliminate them, use `ctitles`.

`norowtitl` ensures that there are no row titles - the default row titles of the coefficient variable names are not used. It is unlikely that you will want to eliminate row titles for an `outreg` table, because it will be difficult to know which coefficient is which. To replace the row headings, instead of eliminate them, use `rtitles`.

`addrows(`*textgrid*`)` adds rows of text to the bottom of the `outreg` table (above the notes). All elements of the rows must be converted from numbers to text before including in the *textgrid*. For example, to include the test results of coefficient equality, you could use `addrows("t test of b1=b2","`‘`ttest`’ `**")` where "`ttest`" is the name of a [R] **local macro** with the value of the t test of ceofficient equality. The asterisks are included because the t test was significant at the 5% level.

See section 3.7 for an application of `addrows`.

`addrtc(#)` is a rarely used option to specify the number of row title columns in addrows. It is only needed when either the default row titles, `rtitles`, or `addrows` has more than one column to ensure that the row titles are lined up correctly vis-a-vis the data. Multi-equation results have two row title columns because of the equation names. By default, `addrtc` is equal to 1.

`addcols(`*textgrid*`)` adds columns to the right of table. The contents of the new columns are not merged - it is the user's responsibility to ensure that the new columns line up in the appropriate way.

`annotate(`*Stata matrix name*`)` passes a matrix of annotation locations.

`asymbol(`*textrow*`)` provides symbols for each annotation location in `annotate`.

`annotate` and `asymbol` (always specified together) are useful for placing footnotes or other annotations next to statistics in the `outreg` table, but they are not the most user-friendly. (Footnotes or annotations in any of the title regions, including row and column titles, can be included directly in the title text with options like

`rtitles` and `ctitles`.)

The values in `annotate` range from 0 to the number of symbols in `asymbols`. The dimensions of the matrix in `annotate` has rows equal to the number of coefficients in the estimation, and columns equal to the number of statistics displayed (2, by default). Whenever the `annotate` matrix has a value of zero, no symbol is appended to the statistic in the corresponding cell of the table. Where the `annotate` matrix has a value of 1, the first `asymbol` symbol is added on the left of the statistic, where there's a value of 2, the second symbol is added, etc.

The *textrow* in `asymbols` has the syntax "*text*"[,"*text*" ...]]. If you want to have a space between the statistic in the table and the `asymbol` *text*, make sure to include it in the *text*, e.g. `asymbols(" 1"," 2")`. Superscripts for the symbols in a Word file can be included as follows: enclose the symbol with curly brackets "`{}`" and prepend the superscript code "`\super`". So for a superscript one ([1]), the *text* in `asymbols` would be "`{\super 1}`". Make sure to include the space after "`\super`". For TEX files, "`1`" can be superscripted either with the code "`$^1$`" or "`\textsuperscript{1}`". See the discussion about inline formatting in section 4.12.

To understand the correspondence between the locations in the `annotate` matrix and the final `outreg` table, it helps to know how `outreg` uses the `frmttable` program to create tables. `outreg` sends the different estimation statistics in separate columns, so for the default statistics of `b` and `t_abs`, `outreg` sends a K by 2 matrix to `frmttable`, where K is the number of coefficients. The nonzero locations of `annotate` that indicate a symbol should correspond to the locations of the K by 2 matrix passed to `frmttable`, not the $2K$ by 1 table of statistics created by `frmttable`. Perhaps a simpler way of saying this is that `annotate` positions correspond to the final table positions when you use the `nosubstat` option. If there are S statistics (2 by default), the `annotate` matrix should be a K by S Stata matrix where K is the number of columns in `e(b)`. This can be created in a Stata for a regression with 5 coefficients and the default of 2 statistics like this:

```
. matrix annotmat = J(5,2,0)
. matrix annotmat[1,1] = 1
. matrix annotmat[3,2] = 2
. outreg ... , annotate(annotmat) asymbol(" (1)"," (2)")
```

This will assign the first `asymbol(" (1)")` to the first coefficient, and the second `asymbol(" (2)")` to the third t statistic.

In fact, the `annotate` matrix can be smaller than K by S if there are rows at the bottom of the table or columns on the right of the table that don't need any symbols. In other words, if the `annotate` matrix is not the same size as the statistics, the missing, or too large, parts of it are ignored.

If `annotate` and `asymbol` are used to create footnote references, the footnotes themselves can be included in the `note` option.

See section 3.14 for an example of `annotate` and `asymbol`.

4.4 Column formatting

`colwidth(`*numlist*`)` assigns column widths. By default, `outreg` makes its best guess of the appropriate column width, but Word RTF files have no algorithm to ensure that the column width exactly fits the maximum width of the contents of its cells, the way TEX files do. In particular, when special non-printing formatting codes (such as superscript codes) are included in `ctitles` and `rtitles`, `outreg` will probably get the width wrong, and `colwidth` will be needed. This option is only allowed for Word files, not TEX files, which automatically determine column widths.

If colwidth has fewer widths than the number of columns, `outreg` will guess the best width for the remaining columns. Specifying `colwidth(10)` will assign a width of 10 characters to the first column in the table, but not change the width of other columns. To assign a width of 10 to all columns in a five column table, use `colwidth(10 10 10 10 10)`. The width of the column using `colwidth(1)` is equal to the width of one "n" of the currently assigned point size, with the addition of the default buffers on either side of the cell.

`multicol(`*numtriple*`[; `*numtriple*`...])` combines table cells into one cell that spans multiple columns. This is mainly used for column titles that apply to more than one column. A *numtriple* means three numbers, separated by commas. Each *numtriple* consists of the row of the first cell to be combined, the column of the first cell, and the number of cells to be combined (≥2).

For example, to combine the heading for the first two statistics columns in a table (with only one `rtitles` column), the option would be `multicol(1,2,2)`. That is, the combined cells start in the first row of the table (below the title) and the second column of the table (the start of the statistics columns), and two cells are to be combined. See an example of this in section 3.10.

It often looks good to underline the `ctitles` in the combined cell to make clear that the column title applies to both columns below it. In Word RTF files, underlining does not apply to blank spaces, so to extend the underline to either side of the text in the `ctitle`, you can insert tab characters, which will be underlined. For example, for the `ctitle` text "`First 2`", you could apply codes for underlining and tabs like this: `ctitle("","{\ul\tab First 2\tab\tab")`. Note the obligatory space between RTF code (`"\tab"`) and the text. Underscore characters "_" can also be used to extend underlining where there is no text, although they create a line that is slightly lower than the underlining line.

`coljust(`*cjstring*`[; `*cjstring*`...])` specifies whether the table columns are left, center, or right justified (that is, the text in each row is flush with the left, center, or right side of the column) or centered on the decimal point (for Word files only). By default, the `rtitles` columns are left justified, and the rest of the columns are decimal justified for Word files. For TEX files, `rtitles` columns are left justified, and the rest of the columns are center justified.

cjstring is a string made up of:

Element	Action
`l`	left justification
`c`	center justification
`r`	right justification
`.`	decimal justification (Word only)
`{}`	repetition

Left, center, and right justification are self-explanatory, but decimal justification requires some elaboration. Decimal justification lines up all of the numbers in the column so that the decimal points are in a verticle line. Whole numbers are justified to the left of the decimal point. Text in the `ctitles` is not decimal justified — otherwise the whole `ctitle` for the column would be to the left of the decimal point, like whole numbers. Instead, in columns with decimal justification `ctitles` are center justified.

Decimal justification works with comma decimal points used in many European languages (to set comma decimal points in Stata, see `set dp comma` in [R] **format**). However, the Microsoft Word application will recognize the comma decimal points correctly only if the operating system has been changed to specify comma decimal points. In the Windows operating system, this can be done in the Control Panel under Regional and Language Options. In the OSX operating system, this is done in System Preferences under Language and Text: Formats.

Each letter in *cjstring* indicates the column justification for one column. For example, "`lccr`" left justifies the first column, center justifies the second and third column, and right justifies the fourth column. If there are more than four columns, the remaining columns will be right justified, since the last element in the string is applied repeatedly. If there are fewer than four columns, the extra justification characters are ignored.

The curly brackets "`{}`" repeat the middle of *cjstring*. For example, "`l{c}rr`" left justifies the first column, center justifies all the subsequent columns up to the next to last column, and right justifies the last two columns.

The semi-colon "`;`" applies column justification to separate sections of the `outreg` table, but is not needed by most users. `outreg` tables have two column sections: the columns of `rtitles` (typically one column), and the columns of estimation statistics.

The section divider allows you to specify the column justification without knowing how many columns are in each section. Hence, the default `coljust` parameters for Word files are `coljust(l;.)`, which applies left justification to all the columns in the first (`rtitles`) section of the table and decimal justification to the remaining column sections of the table.

For example, `coljust(l{c}r;r{c}l)` would apply "`l{c}r`" only to the first column section, and "`r{c}l`" to the second (or more) column sections.

❑ **Technical note**

TEX has the capability for decimal justification using the `dcolumn` package or the `{r@{.}l}` column justification syntax. However, both these methods conflict with other capabilities of `outreg` in ways that make them very difficult to implement. The dcolumn package imposes math mode for the decimal justified columns, which is inconsistent with `outreg` formatting, and also interferes with the `multicol` option. The `{r@{.}l}` syntax splits the column in question into two columns, which would require workarounds for many `outreg` options. Users who do not care to have their t statstics displayed in a smaller font than the coefficient estimations (as is the default in `outreg`), can modify their TEX tables manually to implement decimal justification using the `dcolumn` package.

❑

`nocenter`: Don't center the `outreg` table within the document page. This does not apply to the display of the `outreg` table in the Stata Results window, which is always centered.

4.5 Font specification

`basefont(`*fontlist*`)` changes the base font for all text in the `outreg` table, as well as `pretext` and `posttext`. The default font specification is 12 point Times New Roman for Word documents, and is left unspecified for TEX documents (which normally means it is 10 point Times New Roman).

The *fontlist* is made up of elements in the tables below (different for Word and TEX files), separated by spaces. The elements of the *fontlist* can specify font size, font type (e.g. Times Roman, Arial, or a new font from `addfont`), and font style (like *italic* or **bold**).

If you specify more than one font type (`roman`, `arial`, `courier`, and perhaps `fnew#`), only the last choice in the *fontlist* will be in effect.

See section 3.11 for an application of `basefont`.

A *fontlist* for Word files is made up of:

Element	Action
`fs#`	font size in points
`arial`	Arial font
`roman`	Times New Roman font
`courier`	Courier New font
`fnew#`	font specified in `addfont`
`plain`	no special font effects
`b`	bold text
`i`	italize text
`scaps`	small caps: capitalize lower case letters
`ul`	underline text
`uldb`	underline text with a double line
`ulw`	underline words only (not spaces between words)

A *fontlist* for TEX files is made up of:

Element	Action
`fs#`	font size in points (10, 11, or 12)†
`Huge`	bigger than huge
`huge`	bigger than LARGE
`LARGE`	bigger than Large
`Large`	bigger than large
`large`	bigger than normalsize
`normalsize`	default font size
`small`	smaller than normalsize
`footnotesize`	smaller than small
`scriptsize`	smaller than footnotesize
`tiny`	smaller than scriptsize
`rm`	Times Roman font
`it`	italic text
`bf`	bold face text
`em`	emphasize text (same as `bf`)
`sl`	slanted text
`sf`	sans-serif font, i.e. Arial
`sc`	small caps
`tt`	teletype, i.e. Courier
`underline`	underline text

† `fs#` can only be specified in the `basefont` option for TEX files, not in other font specification options.

`titlfont(`*fontcolumn*`)` changes the font for the table's title.

`notefont(`*fontcolumn*`)` changes the font for notes below the table.

`titlfont` and `notefont` take a *fontcolumn* rather than a *fontlist* to allow for different fonts on different rows of titles or notes, such as a smaller font for the subtitle than the main title.

A *fontcolumn* consists of *fontlist* [\ *fontlist* ...], where *fontlist* is defined above for Word files or for TEX files.

For example, to make the title font large and small caps, and the subtitles still larger than regular text, without small caps, you could use `titlfont(fs17 scaps\fs14)` for a Word file, or `titlfont(Large sc\large)` for a TEX file.

`ctitlfont(`*fontgrid*[;*fontgrid*[...]]`)` changes the fonts for column titles.

`rtitlfont(`*fontgrid*[;*fontgrid*[...]]`)` changes the fonts for row titles.

`statfont(`*fontgrid*[;*fontgrid*[...]]`)` changes the fonts for statistics in the body of the table.

`ctitlfont`'s, `rtitlfont`'s, and `statfont`'s arguments are *fontgrids* to allow a different font specification for each cell of the `ctitles`, `rtitles`, or the table statistics, respectively. By default, all of these areas of the table have the same font as the `basefont`, which by default is Times Roman, 12 point for Word files.

A *fontgrid* consists of *fontrow*[\ *fontrow* ...], where *fontrow* is *fontlist*[, *fontlist* ...]and where *fontlist* is defined above for Word files or for TEX files under the `basefont` option.

For example, to make the font for the first row of `ctitles` bold and the second (and subsequent) rows of `ctitles` italic, you could use `ctitlfont(b\i)` for a Word file, or `ctitlfont(bf\it)` for a TEX file.

The semi-colon "`;`" in the argument list applies different fonts to separate sections of the `outreg` table. This is more likely to be useful for row sections than column sections. See the entry `outreg` *table sections* at the end of this section (4.5) for a diagram of `outreg` row and column sections. `outreg` tables have two column sections: the columns of `rtitles` (typically one column), and the columns of estimation statistics. `outreg` tables have four row sections: the rows of `ctitles` (often one row), and three sections for the `rtitles` and statistics: the rows of regular coefficients, the rows of constant coefficients, and the rows of summary statistics below the coefficients.

The section divider allows you to specify the column or row fonts without knowing for a particular table how many columns or rows are in each section. To italicize the *t* statistics below coefficient estimates for the coefficients, but not italicize the summary statistics rows, you could use `statfont(plain\i;plain\i;plain)` for a Word file, or `statfont(rm\it;rm\it;rm)` for a TEX file.

Note that if you specify a new font type or a single font point size in `titlfont` or `statfont`, this is applied to all rows of the `title` or estimation statistics, removing

the default behavior of making the subtitles smaller than the first row of `title`, and the "substatistics" like the t statistic smaller than the coefficient estimates. To retain this behavior, specify two rows of font sizes in `titlfont` or `statfont`, with the second being smaller than the first. Changing the `basefont` does not have any effect on the differing font sizes in the rows of `title` and estimation statistics.

`addfont(`*textrow*`)` adds a new font type, making it available for use in the font specifications for various parts of the `outreg` table. This option is available only for Word files, not TEX files.

By default, only Times Roman ("`roman`"), Arial ("`arial`"), and Courier New ("`courier`") are available for use in Word RTF documents. `addfont` makes it possible to make additional fonts available for use in the Word documents created by `outreg`.

textrow is a sequence of font names in quotation marks, separated by commas.

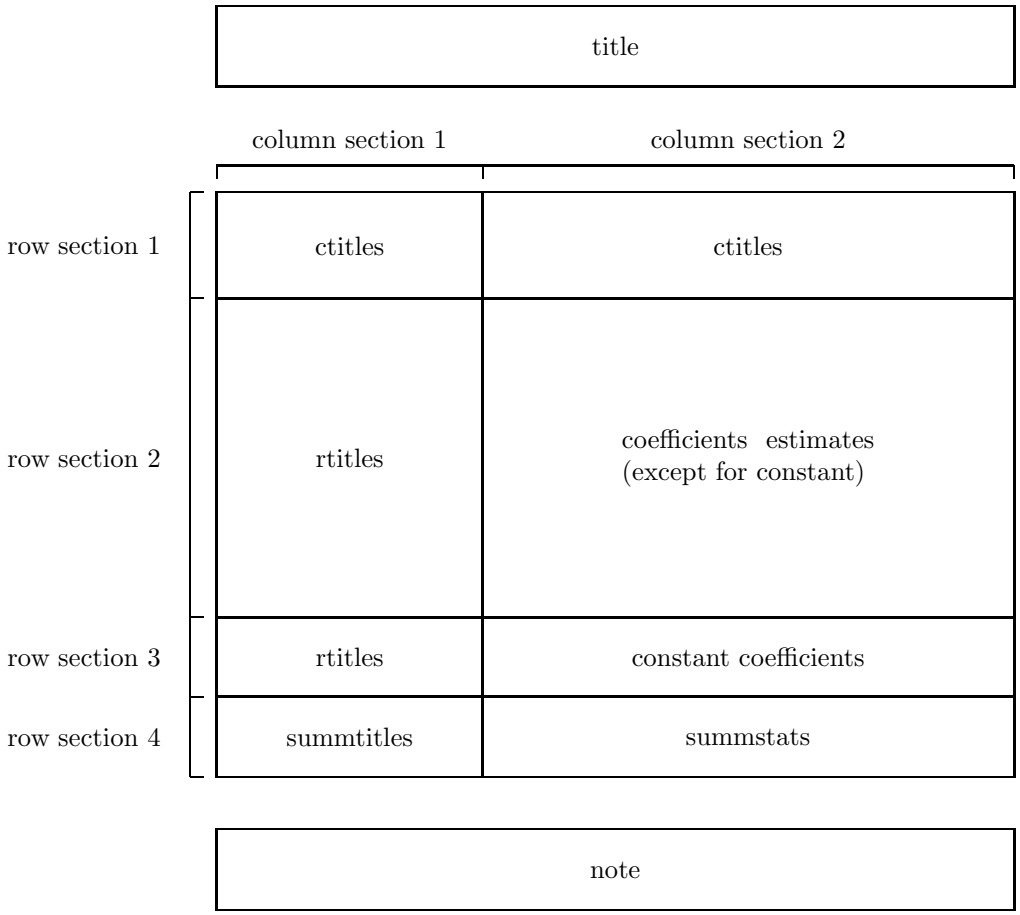
The new font in `addfont` can be referenced in the various font specification options, like `basefont` and `titlfont` with the code "`fnew1`" for the first new font in `addfont` and increments of it ("`fnew2`", "`fnew3`", etc.) for each additional font.

If the font specified in `addfont` is not available on your computer when using the Word file created by `outreg`, the new font will not display correctly - another font will be substituted. You can find the correct name of each available font in Word by scrolling through the font selection window on the toolbar of the Word application.

See section 3.11 for an example of `addfont`.

`plain` eliminates default formatting, reverting to plain text: only one font size for the whole table, no column justification, and no added space above and below the horizontal border lines. Instead of using `plain`, the default formatting can also be reversed feature by feature with `titlfont`, `notefont`, `coljust`, `spacebef`, and `spaceaft`. The `plain` option does this all at once.

`outreg` *table sections*: It can be helpful for specifying fonts and other formatting to understand how `outreg` divides the table into sections. The following diagram illustrates the section divisions:



4.6 Border lines and spacing

`hlines(`*linestring*[`;`*linestring*...]`)` draws horizontal lines between rows.

`vlines(`*linestring*[`;`*linestring*...]`)` draws verticle lines between columns.

`hlines` and `vlines` designate where horizontal and verticle lines will be placed to

delineate parts of the table. By default `outreg` draws horizontal lines above and below the `ctitle` header rows and at the bottom of the table above the notes, if any. There are no verticle lines by default.

linestring is a string made up of:

Element	Action
`1`	add a line
`0`	no line
`{}`	repetition

Each "`1`" in *linestring* indicates a line and a "`0`" indicates no line. For example, `hlines(110001)` would draw a line above and below the first row of the table and below the fifth row (above the sixth row). There is one more possible horizontal line than row (and one more verticle line than column). That is, for a five row table, to put a line above and below every row one would specify *six* `hlines`: `hlines(111111)`.

`hlines` and `vlines` are not displayed correctly in the Stata Results window. They only apply to the final Word or TEX document.

Curly brackets "`{}`" repeat the middle of *linestring*. For example, `hlines(11{0}1)` puts a horizontal line above and below the first row, and another below the last row.

The semi-colon "`;`" applies line designations to separate sections of the `outreg` table. `outreg` tables have two column sections and four row sections. The column sections are made up of the columns of `rtitles` (typically one column), and the columns of the estimation statistics. The row sections are made up of the rows of `ctitles` (often one row), the rows of the coefficient estimates (except the constant), the rows of the constant coefficients, and the rows of the summary statistics below the coefficients.

The section divider allows you to specify the `hlines` and `vlines` without knowing how many rows and columns are in each section. Hence, the default `hlines` elements are `hlines(10;101)`, which puts a horizontal line above the header rows, a line above the statistics rows, and a line below the last statistics row. By default, there are no `vlines`, which some graphic designers think are best avoided.

`hlstyle(`*lstylestring*`[;`*lstylestring*`...])` changes the style of horizontal lines.

`vlstyle(`*lstylestring*`[;`*lstylestring*`...])` changes the style of verticle lines.

`hlstyle` and `vlstyle` options are only available for Word files. By default, all lines are solid single lines.

lstylestring is a string made up of:

Element	Action
`s`	Single line
`d`	Double line
`o`	Dotted line
`a`	Dashed line
`S`	Heavy weight single line
`D`	Heavy weight double line
`O`	Heavy weight dotted line
`A`	Heavy weight dashed line
`{}`	repetition

Repetition using curly brackets "`{}`" and semi-colons "`;`" for section dividers are used in the same way they are for `hlines` and `vlines`.

Some word processing applications, like OpenOffice or Pages (for the Mac) do not display all Word RTF line styles correctly.

`spacebef(`*spacestring*`)` puts space above cell contents.

`spaceaft(`*spacestring*`)` puts space below cell contents.

`spaceht(#)` changes the size of the space above and below cell contents in `spacebef` and `spaceaft`.

`spacebef` and `spaceaft` are options to make picky changes in the appearance of the table. They increase the height of the cells in particular rows so that there is more space above and below the contents of the cell. They are used by default to put space between the horizontal line at the top of the table and the first header row, above and below the line separating the header row from the statistics, and put space below the last row of the table, above the horizontal line.

spacestring has the same form as *linestring* above. A "`1`" indicates a extra space (above the cell if in `spacebef` and below the cell if in `spaceaft`), and a "`0`" indicates no extra space. "`{}`" repeats indicators and "`;`" separates row sections.

`spaceht` controls how big the extra space is in `spacebef` and `spaceaft`. Each one unit increase in `spaceht` increases the space by about a third of the height of a capital letter. The default is `spaceht(1)`. `spaceht` is scaled proportionally to the base font size for the table. For example `spaceht(2)` makes the extra spacing 100% larger than it is by default.

For TEX files (using the `tex` option), `spaceht` can only take the values 2 or 3. The default corresponds to the LATEX code `\smallskip`. Values 2 and 3 for `spaceht` correspond to the LATEX codes `\medskip` and `\bigskip`, respectively.

4.7 File and display options

`tex` specifies that `outreg` writes a TEX output file rather than a Word file. The output is suitable for including in a TEX document (see the `fragment` option) or loading into a TEX typesetting program such as Scientific Word.

`merge` specifies that new estimation output be merged to an existing table in a file of the same name specified in `using`. The coefficient estimates are lined up according to the appropriate variable name (or `rtitles`), with the coefficients for new variables introduced in the merged estimation placed below the original variables, but above the constant term. Note that in previous versions of `outreg`, the `merge` option was called *append*. Users will usually want to specify `ctitles` when using `merge`.

`merge` can be used even if a previous table with the same file name does not exist for merging. This is to enable `merge` to be used in loops. `outreg` issues a warning message if no existing table is found.

`replace` specifies that it is okay to overwrite an existing file.

`addtable` places the estimation results as a new table below an existing table in the same document (rather than combining the tables as with `merge`). This makes it possible to build up a document with multiple tables in it.

Options `pretext` and `posttext` can add accompanying text between the tables. To put a page break between successive tables, so that each table is on its own page, see the discussion for options `pretext and posttext`.

See section 3.13 for an example of `addtable`.

`append` combines the estimation results as new rows below an existing table. **Warning: this is not the append option from previous versions of outreg - use** `merge`.

`append` does not match column headings. The column headings of the new table being appended are ignored unless the new table has more columns than the original table, in which case only the headings of the new columns are used.

`fragment` creates a TEX code fragment for inclusion in a larger TEX document instead of a stand-along TEX document. A TEX fragment saved to the file auto.tex can then be included in the following TEX document with the TEX `\input{auto}` command:

```
\ documentclass[]article
\ begin{document}
    ... text before inclusion of table  auto.tex
\ input{auto}
    ... text after inclusion of table  auto.tex
\ end{document}
```

Including TEX fragments with the TEX `\input{}` command allows the table created by `outreg` to be updated without having to change the TEX code for the document itself. This is convenient because estimation tables often require small modifications which can be made without having to reinsert a new table manually. Creating TEX fragments for inclusion in larger TEX documents is especially useful when there are

many tables in a single document (see also the `addtable` option).

An alternative to the TEX `\input{}` command is the TEX `\include{}` command which inserts page breaks before and after the included table.

`nodisplay` suppresses displaying the table in the Stata Results window.

`dwide` displays all columns in the Stata Results window, however wide the table is. This is mainly useful if you want to copy the table to paste it into another document (which hopefully is not necessary). Without the `dwide` option, very wide tables are displayed in the Results window in sections containing as many columns as will fit given the current width of the Results window.

4.8 Stars options

`starlevels(`*numlist*`)` indicates significance levels for stars in percent. By default, one star is placed next to coefficients which pass the test for significant difference from zero at the 5% level, and two stars are placed next to coefficients that pass the test for significance at the 1% level, which is equivalent to specifying `starlevels(5 1)`. To place one star for the 10% level, 2 for the 5% level, and 3 for the 1% level, you would specify `starlevels(10 5 1)`. To place one star for the 5% level, 2 for the 1% level, and 3 for the 0.1% level, you would specify `starlevels(5 1 .1)`.

See section 3.5 for an example of the `starlevels` option.

`starloc(#)` put stars next to the statistic indicated. By default, stars are displayed next to the second statistic (`starloc(2)`), but they can be placed next to the first statistic (usually the coefficient estimate) or next to third or higher statistic if they have been specified in the `stats` option.

See section 3.9 for an example of the `starloc` option.

`margstars` calculates stars for significance from marginal effects (and their standard errors), rather than from the coefficients themselves, which is the default.

See section 3.9 for an example of the `margstars` option.

`nostars` suppresses the stars indicating significance levels.

`nolegend` indicates that there will be no legend explaining the stars for significance levels below the table (by default, the legend is "`* p<0.05; ** p<0.01`"). To replace the legend, use the `nolegend` option, and put your own legend in a `note` option.

`sigsymbols(`*textrow*`)` replaces the stars used to indicate statistical significance with other symbols of your choice. For example, to use a plus sign "`+`" to indicate a 10% significance level, you could apply `sigsymbols(+,*,**)` along with `starlevels(10 5 1)`. By default, `outreg` uses one star for the first significance level, and adds an additional star for each additional significance level displayed.

The argument *textrow* consists of text separated by commas.

See section 3.5 for an example of the `sigsymbols` option.

4.9 Brackets options

`squarebrack` substitutes square brackets for parentheses around the statistics placed below the first statistic. For the default statistics, this means that square brackets, rather than parentheses, are placed around t statistics below the coefficient estimates.

`squarebrack` is equivalent to `brackets("","\{,\}\(,\)\<,\>\|,\|)`.

`brackets(`*textpair* [*\textpair* ...]`)` specifies the symbols used to bracket statistics placed below the first statistics. By default, `outreg` places parentheses around the second statistic, the t statistic.

A *textpair* is made up of two elements of text separated by a comma. The default brackets are `brackets("","\(,\){,}\<,\>\|,\|)`.

If there are a sufficient number of statistics for the symbols to be used with the `tex` option, `<,>` and `|,|` are replaced by `$<$,$>$` and `$|$,$|$` so that they show up correctly in TEX documents.

`brackets` has no effect when the `nosubstats` option is in effect.

`nobrket` eliminates the application of `brackets`, so that there would be no brackets around the second or higher statistics.

`dbldiv(`*text*`)` is a rather obscure option that allows you to change the symbol that divides double statistics. Double statistics have both a lower and and upper statistic, like confidence intervals, which are the only double statistics in `outreg`. By default, `outreg` puts a dash "-" between the lower and upper statistics, but `dbldiv` allows you to substitute something else. For example, `dbldiv(:)` would put a colon between the lower and upper statistics.

4.10 Summary statistics options

`summstat(`*evaluegrid*`)` places summary statistics below the coefficient estimates. *evaluegrid* is a grid of the names of different `e()` return values already calculated by the estimation command. The syntax of the *evaluegrid* is the same as the other grids in `outreg`. Elements within a row are separated with commas (`,`), and rows are separated by backslashes (\). The default value of `summstat` is `summstat(r2\N)` (when `e(r2)` is defined), which places the R^2 statistic `e(r2)` below the coefficient estimates, and the number of observations `e(N)` below that.

To replace the R^2 with the adjusted R^2 stored in `e(r2_a)`, you could use the options `summstat(r2_a\N)` and `summtitle("Adjusted R2"\"N")`. You can also specify the decimal places for the summary statistics with the `summdec` option. To see a complete list of the `e()` macro values available after each estimation command, type `ereturn list`.

Statistics not included in the `e()` return values can be added to the table with the `addrows` option, as in section 3.7.

See an example of `summstat` in section 3.5.

`summdec(`*numlist*`)` designates the decimal places displayed for summary statistics in the manner of `bdec`.

`summtitles(`*textgrid*`)` designates row titles for summary statistics in the same manner as `rtitles`.

`noautosumm` eliminates the automatically generated summary stats (R^2, if there is one, and the number of observations) from the `outreg` table.

4.11 `frmttable` options

`blankrows` allows blank rows (across all columns) in the body of the `outreg` table to remain blank without being deleted. By default, `outreg` sweeps out any completely blank rows. This option is useful if you want to use blank rows to separate different parts of the table.

`nofindcons` is a technical option that prevents `frmttable` from finding the constant coefficient "`_cons`" and putting it in a separate row section. Usually finding the constant is needed to ensure that new variables coefficients are `merge`d in correctly, above the constant term, when multiple estimations are merged together. This option is most likely to be useful when merging with a non-`outreg` table that treats constants differently.

4.12 Inline text formatting: superscripts, italics, Greek characters, etc.

`outreg`'s font specification options allow users to control font characteristics at the table cell level, but users often want change the formatting of a word or just a character within a string of text or a table cell. This is true for characteristics like superscripts, subscripts, italics, bold text, and special characters such as Greek letters.

Text strings in the `outreg` table can include inline formatting codes that change the characteristics of just part of a string. These codes are distinct between Word and TEX files, because they are really just Word and TEX formatting codes that are passed directly to the output files, so the codes for the two file types are discussed separately.

Word inline formatting

The Word files created by `outreg` follow the Word Rich Text Format (RTF) specification. A large proportion of the RTF specification codes can be included in `outreg` text (find the full 210 page specification in the links of `en.wikipedia.org/wiki/Rich_Text_Format`). This note will explain a small subset of the most useful codes.

Word RTF codes at enclosed in curly braces "`{`" and "`}`". Codes start with a

backslash character "\" and then the code word. There must be a space after the code word before the text begins so that the text is distinguished from the code. For example, the formatting to italicize the letter "F" is "`{\i F}`", because "i" is the RTF code for italics.

Be very careful to match opening and closing curly brackets because the consistency of the nested curly brackets in a Word file is essential to the file's integrity. If one of the curly brackets is missing, the Word file created by `outreg` may be corrupted and unreadable. You can trace problems of this kind by temporarily removing inline formatting that includes curly braces.

RTF code	Action
`\i`	italic
`\b`	bold
`\ul`	underline
`\scaps`	small capitals
`\sub`	subscript (and shrink point size)
`\super`	superscript (and shrink point size)
`\fs#`	font size (in points * 2; e.g. 12 point is \fs24)

Most of these codes are the same as those used in the font formatting options, but there are some differences, such as the font size code `\fs#` using half points, not points.

Greek and other Unicode characters in Word

Word RTF files can display Greek letters and any other Unicode character (as long as it can be represented by the font type you are using). Greek letters and other Unicode characters can be included in the text as follows: a four digit decimal Unicode code preceded by "`\u`" and followed by "`?`". Complete Unicode code tables are available at `www.unicode.org/charts`. For use in Word RTF files, the hexadecimal codes in the online Unicode tables must be converted to decimal numbers.

Unicode RTF codes are an exception to the rule that RTF codes must be followed by a space before other text. Text can follow immediately after the Unicode code.

For example, for the Greek lowercase character alpha, the hexadecimal Unicode code is 03B1, which is equivalent to 945 decimal, which we make into a four digit number by putting a "0" on the front, so the appropriate RTF code would be "`\u0945?`".

The Unicode characters displayed in the Word file are limited only by which Unicode characters are included in the font used. Fonts like Times New Roman and Arial include a very wide range of characters and symbols.

The following table lists the RTF codes for Greek letters.

Greek letter codes in Word

Letter	RTF codes	Letter	RTF codes
α	\u0945?	A	\u0913?
β	\u0946?	B	\u0914?
γ	\u0947?	Γ	\u0915?
δ	\u0948?	Δ	\u0916?
ϵ	\u0949?	E	\u0917?
ζ	\u0950?	Z	\u0918?
η	\u0951?	H	\u0919?
θ	\u0952?	Θ	\u0920?
ι	\u0953?	I	\u0921?
κ	\u0954?	K	\u0922?
λ	\u0955?	Λ	\u0923?
μ	\u0956?	M	\u0924?
ν	\u0957?	N	\u0925?
ξ	\u0958?	Ξ	\u0926?
o	\u0959?	O	\u0927?
π	\u0960?	Π	\u0928?
ρ	\u0961?	P	\u0929?
σ	\u0963?	Σ	\u0931?
τ	\u0964?	T	\u0932?
υ	\u0965?	Υ	\u0933?
ϕ	\u0966?	Φ	\u0934?
χ	\u0967?	X	\u0935?
ψ	\u0968?	Ψ	\u0936?
ω	\u0969?	Ω	\u0937?

TEX inline formatting

The discussion of TEX inline formatting is brief because TEX users are usually familiar with inserting their own formatting codes into text. Many online references explain how to use TEX formatting codes. A good place to start is the references section of `en.wikipedia.org/wiki/TeX`.

For many formatting effects, TEX can generate inline formatting in two alternative ways: in math mode, which surrounds the formatted text or equation with dollar signs "`$`", or in text mode which uses a backslash followed by formating code and text in curly brackets.

For example, we can create a superscipted number 2 either as "`$^2$`" in math mode or "`\textsuperscript{2}`" in text mode. To display R^2 in a TEX document with the "R" italicized and a superscript "2", one can either use the code "`$ R^2$`" or the code "`\it{R}\textsuperscript{2}`".

Note the space between the "`$`" and "`R`" in "`$ R^2$`", which is a Stata, not a TEX, issue. If we had instead written "`$R^2$`", Stata would have interpreted the `$R` as a global macro, which is probably undefined and empty, so the TEX document would just

contain "`^2`". Whenever using TEX inline formatting in math mode which starts with a letter, make sure to place a space between the "`$`" and the first letter.

Math mode generally italicizes text and is designed for writing formulas. A detailed discussion of its capabilities is beyond the scope of this note. Below is a table of useful text mode formatting codes.

TEX code	Action
`\it`	italic
`\bf`	bold
`\underline`	underline
`\sc`	small capitals
`\textsubscript`	subscript (and shrink point size)
`\textsuperscript`	superscript (and shrink point size)

Keep in mind that many of the non-alphanumeric characters have special meaning in TEX, namely `_`, `%`, `#`, `$`, `&`, `^`, `{`, `}`, `~`, and `\`. If you want these characters to be printed in TeX like any other character, include a `\` in front of the character. The exceptions are the last two, `~` and `\` itself. `~` is represented by `\textasciitilde`, and `\` is represented by either `\textbackslash` or `$\backslash$` to render properly in TeX.

Greek letters in TEX

Greek letters can be coded in TEX documents with a backslash and the name of the letter written in English, surrounded by "`$`". For example, a lowercase delta (δ) can be inserted with the code "`$\delta$`". Upper case Greek letters use the name in English with an initial capital, so an uppercase delta (Δ) is "`$\Delta$`".

4.13 Notes about specific estimation commands

[R] **rocfit** reports a t statistic for the null hypothesis that the slope is equal to 1. `outreg` reports the t statistic for the null hypothesis that the slope is equal to 0.

[R] **stcox** and [R] **streg** report hazard ratios by default, and the coefficients only if the `nohr` option is employed. `outreg` does the reverse. To show the hazard rates in the `outreg` table, use the `hr` option.

5 References

Gallup, J. L. 1998. sg97: Formatting regression output for published tables. *Stata Technical Bulletin* 46: 28–30. Reprinted in *Stata Technical Bulletin Reprints*, vol. 8, pp. 200–202. College Station, TX: Stata Press.

———. 1999. sg97.1: Revision of outreg. *Stata Technical Bulletin* 49: 23. Reprinted

in *Stata Technical Bulletin Reprints*, vol. 9, pp. 170–171. College Station, TX: Stata Press.

———. 2000. sg97.2: Update to formatting regression output. *Stata Technical Bulletin* 58: 9–13. Reprinted in *Stata Technical Bulletin Reprints*, vol. 10, pp. 137–143. College Station, TX: Stata Press.

———. 2001. sg97.3: Update to formatting regression output. *Stata Technical Bulletin* 59: 23. Reprinted in *Stata Technical Bulletin Reprints*, vol. 10, p. 143. College Station, TX: Stata Press.