The Stata Journal

The *Stata Journal* publishes reviewed papers together with shorter notes or comments, regular columns, book reviews, and other material of interest to Stata users. Examples of the types of papers include 1) expository papers that link the use of Stata commands or programs to associated principles, such as those that will serve as tutorials for users first encountering a new field of statistics or a major new technique; 2) papers that go "beyond the Stata manual" in explaining key features or uses of Stata that are of interest to intermediate or advanced users of Stata; 3) papers that discuss new commands or Stata programs of interest either to a wide spectrum of users (e.g., in data management or graphics) or to some large segment of Stata users (e.g., in survey statistics, survival analysis, panel analysis, or limited dependent variable modeling); 4) papers analyzing the statistical properties of new or existing estimators and tests in Stata; 5) papers that could be of interest or usefulness to researchers, especially in fields that are of practical importance but are not often included in texts or other journals, such as the use of Stata in managing datasets, especially large datasets, with advice from hard-won experience; and 6) papers of interest to those who teach, including Stata with topics such as extended examples of techniques and interpretation of results, simulations of statistical concepts, and overviews of subject areas.

For more information on the *Stata Journal*, including information for authors, see the webpage

http://www.stata-journal.com

The *Stata Journal* is indexed and abstracted in the following:

- CompuMath Citation Index®
- Current Contents/Social and Behavioral Sciences®
- RePEc: Research Papers in Economics
- Science Citation Index Expanded (also known as SciSearch®)
- Scopus™
- Social Sciences Citation Index®

The *Stata Journal*, electronic version (ISSN 1536-8734) is a publication of Stata Press. Stata, Mata, NetCourse, and Stata Press are registered trademarks of StataCorp LP.

The Stata Journal (2012)
12, Number 1, pp. 3–28

A new system for formatting estimation tables

John Luke Gallup
Portland State University
Portland, OR
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 Microsoft Word or LaTeX files. My objective is to provide as complete control as is practical over the layout and formatting of the estimation tables 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. I demonstrate in examples the numerous formatting options for the `outreg` command.

Keywords: sg97_4, outreg, tables, estimation, formatting, Microsoft Word, TeX, LaTeX

1 Introduction

My goal in creating a new version of `outreg` is to make it possible to create a fully formatted statistical appendix (that would not require any tweaking after creation) in a Microsoft Word or LaTeX document from a single do-file. To this end, `outreg` has a number of capabilities that I do not think exist yet in Stata programs.

- `outreg` can create any (rectangular) arrangement of 22 possible statistics based on coefficient or marginal effect estimates.
- Any summary statistics can be added.
- 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.
- Wide tables can be placed in landscape orientation.
- 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.
- `outreg` can be used in loops, including writing to two or more tables each iteration (for example, saving multiple estimates of first- and second-stage estimation to separate tables).

Although this version shares the basic command syntax of previous versions of `outreg` (Gallup 1998, 1999, 2000, 2001), the code has been rewritten from scratch, 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 it runs fast.

Most of the capabilities of `outreg` come from `frmttable`, a command for programmers that takes a generic Stata matrix of statistics and converts it, along with numerous formatting options and accompanying text, into a Microsoft Word[1] or LaTeX table. Other tables created by `frmttable` 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 large number of options to enable users to make fine adjustments to layout and formatting (analogous, on a smaller scale, to the plethora of `graph` options in Stata). Because `outreg` has so many options, I leave the option descriptions out of this article. Users can find the options and descriptions in the `outreg` help file in Stata.

Several other user-written commands are available for rearranging and formatting Stata estimation results. This version of `outreg` is the only one that offers the features listed in the bullet points above. It also has one of the simplest syntaxes (to my mind). Other programs offer distinctive characteristics. `outreg2` (Wada 2008) is currently the most popular command; it is unique in offering native Excel table output along with Microsoft Word and LaTeX tables. `outreg2` was the outgrowth of earlier work on `outreg`, which was taken over and expanded by Roy Wada. `estout` and related programs (Jann 2005, 2007) are also very popular, especially with serious Stata users, offering numerous options with which to create Microsoft Word and LaTeX tables.

Other estimation formatting programs are simpler. `modltbl` (Tyler 1997) displays reorganized estimation results in the Stata Results window. `mktab` (Winter 2005), `outtex` (Terracol 2001), and `est2tex` (Muendler 2005) all produce LaTeX tables. `xml_tab` (Lokshin and Sajaia 2008) produces XML tables that are openable in Excel and related spreadsheet programs.

1. The Microsoft Word file format is Rich Text Format, an ASCII file format that should be readable by any software that can open Microsoft Word files.

2 Syntax and description

The syntax of `outreg` is

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

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

`outreg` works after any estimation command[2] in Stata (for a complete list, see `help estimation commands`). Similarly to `predict`, `outreg` uses internally saved estimation results, so it should be invoked after the estimation.

If `using` *filename* is specified, `outreg` creates a Microsoft Word file by default or creates a TEX[3] file with 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 Microsoft Word or TEX file.

Successive estimation results, which may use different variables, can be combined by `outreg` into a single table by using the `merge` option (in previous versions of `outreg`, the `merge` option was called `append`).

Because there are so many additional options for `outreg`, descriptions of the options are not included in this article. A list of options and their descriptions can be found in the Stata help file for `outreg`.

❑ Technical note

Differences between the implementation of Microsoft Word tables and TEX tables

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

❑

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

❑ **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 numerical formatting options in the more general form accepted by `frmttable`. `bdec()` and `tdec()` (decimal place options) are rearranged into the more general but less convenient `sdec()` option, and `bfmt()` (number format option) is rearranged into an `sfmt()` option.

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

The heavy lifting of creating tables is done by `frmttable`, which converts the Stata statistics matrix to a Mata string matrix and adds 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 Microsoft Word Rich Text Format file (or a LaTeX file with the `tex` option). The table is then displayed in the Stata Results window, incorporating some of but not all the formatting specifications (for example, 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.

3.1 Basic usage and variable labels

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

```
. sysuse auto
(1978 Automobile Data)
. regress mpg foreign weight
      Source |       SS       df       MS              Number of obs =      74
-------------+------------------------------           F(  2,    71) =   69.75
       Model |  1619.2877     2  809.643849            Prob > F      =  0.0000
    Residual |  824.171761    71   11.608053           R-squared     =  0.6627
-------------+------------------------------           Adj R-squared =  0.6532
       Total |  2443.45946    73  33.4720474           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   -.0053175
       _cons |    41.6797   2.165547    19.25   0.000     37.36172    45.99768
------------------------------------------------------------------------------
```

The simplest form of `outreg` displays a reformatted estimation table in the Stata Results window.

```
. outreg
```

	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

If you use the `outreg using auto` command, it will create a new Microsoft Word file named `auto.doc` and it will display the table in the Results window (a feature that can be turned off with the `nodisplay` option). `outreg` can also create tables in TeX format with the `tex` option.

The option `varlabels` replaces variable names with their labels; in our example, 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. You can customize the variable labels before invoking `outreg` to provide the desired captions in the `outreg` table. Alternatively, you can specify column and row titles directly with the `ctitles()` and `rtitles()` options.

If the file `auto.doc` already exists from a previous `outreg` command, then we must also include the `replace` option.

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

The Results window display of the table is omitted here. Instead, I show the resulting Microsoft 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 is not very satisfactory for the `weight` variable in the regression above. The `weight` coefficient is statistically significant, but only one nonzero digit is displayed. We could use the option `bdec(5)` to display five decimal places for all the coefficients, but we can do even 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 backslashes (which indicate multiple lines of title) or commas, no quotation marks are required. So we add the option `title(What cars have low mileage?)`. We also change the column heading of the estimates from the name of the independent variable to `Base case` with the option `ctitles("", Base case)`. We need the `""` to indicate that there is no title heading in the left-most column of the table. We can get away with no quotes around `Base case` because it contains no backslashes or commas, 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?) ctitles("", Base case)
```

(*output omitted*)

If you run the commands above and open the resulting file, `auto.doc`, in Microsoft Word or most other word-processing softwares, 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 combine multiple estimation results with the `merge` option.

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

```
. generate 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 section 3.2 above by using the `merge` option. In the second regression, the `weightsq` term is statistically significant but very small because of 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 `bfmt(f f e f)` option. We also specify the number of decimal places for each coefficient as we did in the first regression with the `bdec(2 5 2)` option. We add an informative column title with the option `ctitles("", 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()` applies to all the remaining coefficients.

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

The coefficients and t statistics for the variables are aligned correctly in the merged table, and the scientific notation is 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

Because the first `outreg` table from section 3.2 used `varlabels`, we need to use `varlabels` in the `outreg` command that merges the second regression. If we did not, the row titles would differ between the original table and the new results being merged, causing the coefficients to be aligned incorrectly. For example, the label for the first coefficient in the original table is `Car type`. Without the `varlabels` option in the `outreg` command above, the first coefficient of the second regression would be labeled `foreign` and would be treated as a new variable instead of being aligned in the first row with `Car type`.

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

Economics journals often prefer standard errors to t statistics and do not use asterisks to denote statistical significance. The `se` option replaces t statistics with standard errors, and the `nostars` option suppresses asterisks. We will also replace the parentheses around the standard errors with square brackets by using the `squarebrack` option, and we will 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 asterisks indicating statistical significance can be modified with the `starlevels()` option. The default levels are one asterisk for 5% significance and two asterisks for 1% significance (that is, `starlevels(5 1)`). To add a symbol for 10% significance, we use the `starlevels(10 5 1)` option; this would display one asterisk

for 10%, two for 5%, and three for 1%. To retain the original number of asterisks for 5% and 1% levels but add a cross symbol 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 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 by 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 `summtitles(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 `summdec(1 2)` option.

```
. regress 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)
> summtitles(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.

This 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, below, uses `keep` to remove from the table the auxiliary parameters included in `e(b)` by Stata. The `tobit` command (see [R] **tobit**) 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 by using the `matrix list e(b)` command. By default, `outreg` includes the sigma parameter and the equation names in the estimates table.

```
. generate 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 `keep(model:)` option. 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, that we may want to report in estimation tables but that 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 `test` command (see [R] **test**) as added 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 `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 `r(F)` and `r(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, we request that the local macro `p` have three decimal places.

```
. generate goodrep = rep78==5
. regress 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 a 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 Multiequation models

`outreg` displays estimation results in a single column even for multiequation models unless you choose the `eq_merge` option (for "equation merge"). When different equations in the estimation model share many of the same covariates, you may prefer to display the results like the merged results of separate estimations. `eq_merge` puts each equation in a separate column, and any common variables are displayed in 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 these statistics below the coefficient estimates along with the number of observations. The `summstat()` option has three columns and two rows.

3.9 Marginal effects and asterisk options

`outreg` can display marginal effects estimates calculated by the `margins` command (see [R] **margins**) instead of displaying coefficient estimates. `outreg` can also display marginal effects calculated by the `mfx` and `dprobit` commands that were part of Stata 10 and earlier. Displaying marginal effects requires that you run `margins, dydx(*)` or a similar command 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 `b_dfdx` 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 to the underlying coefficients.

```
. logit foreign wgt mpg
  (output omitted)
. margins, dydx(*)
  (output omitted)
. outreg using auto, marginal replace
```

	foreign
wgt	-0.046
	(8.01)**
mpg	-0.020
	(2.03)
N	74

* p<0.05; ** p<0.01

Marginal effects can also be combined with regression coefficients or other statistics in the `outreg` table. The table below 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 b_dfdx ci_dfdx)` option. Note that the statistics `b_dfdx` and `ci_dfdx` refer to whichever marginal effects were specified in the `margins` command. This could be `dydx()`, `eydx()`, `dyex()`, or `eyex()` depending on the `margins` option used.

The `margstars` option specifies that the asterisks refer to the significance of the hypothesis that the marginal effects are zero rather than the hypothesis that the coefficients are zero. The `starloc(3)` option places the asterisks next to the third statistic (the marginal effect confidence intervals) instead of the default position next to the second statistic.

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

Marginal Effects & Confidence Intervals
Below Coefficients

	foreign
wgt	-0.391
	(-0.046)
	[-0.057 - -0.035]**
mpg	-0.169
	(-0.020)
	[-0.039 - 0.001]
_cons	13.708
N	74

* p<0.05; ** p<0.01

3.10 Multicolumn ctitles(); merge variable means with estimation results

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

First, we create an `outreg` table that 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 of column titles, with a supertitle over the first two columns of `Regressions`.

Notice that the two `outreg` commands below do not include a `using` statement. This means that the results are not written as Microsoft Word files. Saving the files right now is not necessary because we will merge more estimation results below and do not need to save the intermediate files. The contents of the table are saved in Stata's memory in the meantime.

```
. regress mpg foreign weight
```
(*output omitted*)
```
. outreg, bdec(2 5 2) varlabels nodisplay
> ctitles("", "Regressions" \ "", "Base case")
. regress mpg foreign weight weightsq
```
(*output omitted*)
```
. outreg, bdec(2 5 2) bfmt(f f e f) varlabels merge
> ctitles("", "" \ "", "Quadratic mpg") nodisplay
```

Then we run the `mean` command, which calculates variable means and their standard errors. `mean` is an estimation command, so it stores its results in `e(b)` and `e(V)`, which 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 the `Quadratic mpg` columns. See the result in the Microsoft 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 `merge`d with a new one. It is also often necessary to specify the `colwidth()` option because the column-width algorithm does not account for the fact that the column supertitle spans two columns. Without the `colwidth()` option, the `Base case` column would be too wide because of the wide supertitle.

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

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

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

We could embellish the `Regressions` supertitle by underlining it. In Microsoft Word files, this is accomplished with the formatting code `{\ul Regressions}`. If we want the underline to span more widely than the word `Regressions`, one approach is to place tab characters before and after the word. Spaces do not do the job because Microsoft Word does not underline spaces. To place one tab character on either side of the supertitle, we would use `{\ul\tab Regressions\tab}` in the `ctitles()` option. Another option is to use underscore characters, although the line they create is offset slightly below the underlining. See the *Inline text formatting* section from `help outreg_complete` within Stata for more information about underlining and other within-string formatting issues.

3.11 Specifying fonts

One of the objectives of this version of `outreg` is to have as complete control as is possible of the layout and appearance of estimates tables. An important element of control relates to fonts. `outreg` now enables users to specify fonts down to the table cell level, although this is rarely needed. You can specify font sizes, font types (such as Times Roman or Arial), and font styles (such as bold or italic). For Microsoft Word files, you can apply any font type installed on your computer by adding the font name in the `addfont()` option.

In this example, we prepare a table for a presentation as an overhead slide with special fonts that are displayed much larger than usual (the resulting Microsoft Word table below is shrunk so as not to take up too much of the page).

Two specialized fonts are added to the document with the `addfont(Futura, Didot Bold)` option. These fonts can then be applied to different parts of the table as "`fnew1`" for the first added font (Futura) or "`fnew2`" for the second added font (Didot Bold). We set the default font of the table to be Futura in the `basefont(fs32 fnew1)` option. This `basefont()` option also sets the font size to 32 points. 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 Microsoft Word and TEX documents and do not need to be included in `addfont()`.) 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 Microsoft Word table in `auto.doc` is displayed in shrunken form below.

```
. regress 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 more detail in the `outreg` help file (type `help outreg` in Stata).

This example is similar to the one in section 3.7 in that the results of a test of coefficient equality are displayed in the estimation table. However, because 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 Microsoft Word file generated by `outreg`. (A different set of codes can produce the same formatting in TEX files, which is also discussed in the `outreg` help file.)

The Microsoft Word code for the Unicode representation of the Greek lowercase letter chi is `\u0966?` (see all Greek letter codes for Microsoft Word files in the `greek_in_word` help file). 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, Microsoft Word would try to interpret the code `\super2`, which does not exist. Finally, we italicize the p in p-value with `{\i 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 values of local macros containing the numerically formatted values of the χ^2 statistic and its p-value.

The `outreg` command requires the `colwidth()` option, as is usually the case with inline formatting, because `outreg` cannot distinguish the formatting codes from the width of the displayed characters.

For some reason, Stata gives the `logit` regression output an equation name for the coefficients even though there is only one equation, which would show the variable titles in two columns. The `keep(foreign:)` option selects this equation, which effectively strips off the equation name from the variable titles.

The `note()` option in the `outreg` command below has a couple of tricks to 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 filename from predefined Stata macros `$S_TIME`, `$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 keep(foreign:)
> 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 2011
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, you must be able to write multiple tables to the same document, which is possible with the `addtable` option.

The `mean` command below creates summary statistics for the variables. `outreg` with the `addtable` option places a summary statistics table below the table just created in section 3.12. The option `nostars` turns off asterisks for significance tests, and `nosubstat` puts the standard errors side by side with the means, as explained in section 3.15 below. The formatted Microsoft Word table is shown in shrunken form below the command.

```
. mean foreign wgt mpg
```

(*output omitted*)

```
. outreg using auto, addtable ctitle(Variables, Means, Std Errors)
> nostars nosubstat title(Summary statistics)
```

(*output omitted*)

Greek characters, superscripts, and italics

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 02:14:55, 9 Sep2011
Using data from /Applications/STATAMP/Stata/ado/base/a/auto.dta

Summary statistics

Variables	Means	Std Errors
foreign	0.297	0.053
wgt	30.195	0.903
mpg	21.297	0.673

You can use the `pretext()` and `posttext()` options to add paragraphs of regular text before and after each table.

3.14 Place footnotes among coefficients

Placing footnotes in any of the text elements of an `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()` option.

Placing a footnote in the body of the `outreg` table is not as straightforward because the table body is made up of numeric statistics. To place a footnote in the body of the table, we use the `annotate()` option. First, we create a Stata matrix with the footnote

locations used by `annotate()`, and we put the footnote symbols in the text string of `asymbol()`.

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 3×2 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` that are zero are ignored. The locations with a `1` have the first `asymbol()` appended, `2` have the second `asymbol()` appended, etc. Because 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 want 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}` because these should be footnote numbers 2 and 3.

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

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

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

```
. regress 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*)

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.

It is not possible to position a footnote next to the summary statistic in `summstat()`. To accomplish this, use the `addrows()` option instead of the `summstat()` option. First, you must turn off the automatic summary statistics by using the `noautosumm` option and then place the statistic and the footnote symbol in `addrows()`, which was described in sections 3.7 and 3.12.

3.15 Show statistics side by side, like Stata estimation results

To show statistics side by side, such as *t* statistics next to the coefficients rather than below them, 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 TEX document (See the section *Inline text formatting* in the `outreg` help file).

```
. 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 {\tt ereturn display}")
> bdec(7) nostar replace tex
```

(*output omitted*)

The resulting TEX table shows all statistics for each coefficient side by side:

Horizontal Output like Stata's `ereturn display`

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

3.16 Merge multiple estimation results in a loop

If you want to run the same estimation on different datasets or on different groups within a dataset, it is often efficient to create a loop using the `forvalues` or `foreach` commands (see [P] **forvalues** or [P] **foreach**). This example first shows how to merge the results of each estimation in the loop into a single `outreg` table, and second shows how to merge sequential estimations in a loop into two separate tables.

Say that we want to run separate regressions by groups that are indexed by the categorical variable `rep78` in the `auto.dta` dataset. We use the `forvalues` command to create a loop that steps through the values of `rep78` from 2 to 5. For each value of `rep78`, we run a regression of the variable `mpg` on covariates, restricting the sample to the current value of `rep78` with the statement `if rep78==‘r’`. (`r` is a local macro containing the current value of the loop indicator.)

Following each regression, the `outreg, merge` command merges successive regression results into a single table. The first time that `outreg, merge` is executed after the first regression, we actually do not want it to merge with anything. The `merge` option allows merging without an existing table precisely to enable its use in loops, although `outreg` does produce the warning message that no existing `outreg` table was found.

To ensure that there is no preexisting table before the first `outreg, merge` command, we precede the `forvalues` loop with the command `outreg, clear`. The `clear` option removes any `outreg` table in memory; `outreg` tables persist until cleared or replaced by a new table. Even if no previous `outreg` command has been run, if the commands in this example are rerun, the `outreg, clear` command is necessary to clear out the previous version of the table.

```
. outreg, clear
. forvalues r = 2/5 {
2.   quietly regress mpg price weight if rep78==`r´
3.   outreg, merge varlabels ctitle("", "`r´") nodisplay
4. }
warning: no existing table found for merge or append
```

The `outreg` command in the loop does not need any `using` statement because we do not need to save the table as a Microsoft Word document (or TeX document) until we have merged all the regressions together. Once we have, and the loop is complete, we

save the table as a Microsoft Word document with the `outreg using auto, replay` command. (If we had left out the `replay` option, we would have created a new table from the existing `e(b)` matrix, which holds the results of the last regression in the loop.)

```
. outreg using auto, replay replace title(Regressions by repair record)
```

Regressions by repair record

	2	3	4	5
Price	-0.000	0.000	0.000	0.001
	(0.61)	(0.07)	(0.71)	(0.98)
Weight (lbs.)	-0.008	-0.004	-0.005	-0.025
	(5.40)**	(4.74)**	(8.47)**	(3.10)*
Constant	44.953	34.052	34.918	78.648
	(10.91)**	(14.40)**	(15.96)**	(6.17)**
R2	0.92	0.64	0.84	0.76
N	8	30	18	11

The `replay` option tells `outreg` to use the existing `outreg` table in memory instead of creating a new one. With the `replay` option, it is possible to make text additions (except for `varlabels`) such as new titles or even `addrows()`, but it is not possible to change the numerical contents or numerical formatting of the statistics in the table (options for estimates selection, estimates formatting, asterisks, brackets, and summary statistics will be ignored). When using the `replay` option, it is possible to specify all the text formatting options (such as fonts, lines, and spacing) and the relevant file options (such as `replace` or `tex`).

There are some contexts in which it is helpful to merge the estimation results in a loop into two separate `outreg` tables, such as when for each iteration of the loop, the results of the first estimation are used in the second estimation, and we want to record the results of both estimations. In this example, we run instrumental-variables estimation in a loop, and we record both the first- and the second-stage regressions. To merge the regression results into two separate tables, we need to give the tables separate names. Each time the `merge` option is used, it will refer to either the "first" table (for the first-stage regression results) or the "iv" table (for the second-stage results). These table-specific `merge` options become `merge(first)` and `merge(iv)`.

As before, we precede the `forvalues` loop with `outreg, clear` to clear out any `outreg` table in memory. In this case, though, we need to refer to the named tables, so we have two commands: `outreg, clear(first)` and `outreg, clear(iv)`. The built-in Stata command for instrumental-variables estimation, `ivregress` does not have the capability of saving the first-stage results (although they can be displayed). Instead, we use `ivreg2`, which saves the first-stage results with the `savefirst` option. The `ivreg2` command is preceded by the `quietly` command to suppress the display of its output. We then add the instrumental-variables estimates to the "iv" table with the `outreg, merge(iv)` command. The `estimates restore _ivreg2_hsngval` command puts the first-stage estimates into the `e(b)` and `e(V)` vectors. The second `outreg` command, `outreg, merge(first)`, saves the first-stage regression results in the "first" table.

```
. outreg using auto, replay replace title(Regressions by Repair Record)
. webuse hsng2, clear
(1980 Census housing data)
. outreg, clear(iv)
. outreg, clear(first)
. forvalues r = 1/4 {
  2.   quietly ivreg2 rent pcturban (hsngval = faminc) if reg`r´, savefirst
  3.   outreg, merge(iv) varlabels ctitle("", "Region `r´") nodisplay
  4.   quietly estimates restore _ivreg2_hsngval
  5.   outreg, merge(first) varlabels ctitle("", "Region `r´") nodisplay
  6. }
warning: no existing table found for merge or append
warning: no existing table found for merge or append
```

We now save the two tables with two `outreg, replay` commands. To replay the table of first-stage estimates, we use the `replay(first)` option; to replay the second-stage estimates, we use the `replay(iv)` option. By using the `addtable` option in the second `outreg, replay` command (and `using` the same filename), we combine both tables into the file `iv.doc`.

```
. outreg using iv, replay(first) replace title(First Stage Regressions by Region)
```
(*output omitted*)
```
. outreg using iv, replay(iv) addtable
> title(Instrumental variables regression by region)
```
(*output omitted*)

4 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.

Jann, B. 2005. Making regression tables from stored estimates. *Stata Journal* 5: 288–308.

———. 2007. Making regression tables simplified. *Stata Journal* 7: 227–244.

Lokshin, M., and Z. Sajaia. 2008. Creating print-ready tables in Stata. *Stata Journal* 8: 374–389.

Muendler, M.-A. 2005. est2tex: Stata module to create LaTeX tables from estimation results. Statistical Software Components S428402, Department of Economics, Boston College. http://ideas.repec.org/c/boc/bocode/s428402.html.

Terracol, A. 2001. outtex: Stata module to LaTeX code for result tables after any estimation command. Statistical Software Components S420101, Department of Economics, Boston College. http://ideas.repec.org/c/boc/bocode/s420101.html.

Tyler, J. H. 1997. sg73: Table making programs. *Stata Technical Bulletin* 40: 18–23. Reprinted in *Stata Technical Bulletin Reprints*, vol. 7, pp. 186–192. College Station, TX: Stata Press.

Wada, R. 2008. outreg2: Stata module to arrange regression outputs into an illustrative table. Statistical Software Components S456416, Department of Economics, Boston College. http://ideas.repec.org/c/boc/bocode/s456416.html.

Winter, N. 2005. mktab: Stata module to print table of estimates in delimited or screen-presentation format. Statistical Software Components S416801, Department of Economics, Boston College. http://ideas.repec.org/c/boc/bocode/s416801.html.

About the author

John Luke Gallup is an assistant professor in the Department of Economics at Portland State University.