Rapid Formation of Regression Tables for Research Purposes

Roy Wada[1]
The RAND Corporation
rwada@rand.org

Abstract. The ostensible reason for a preparation of regression tables is to have them submitted to journals for publication purposes. Contrary to this professed view, regression tables are used during research and not after. Regression tables are information management tools that concentrate information from various sources for immediate consumption by researchers. A next logical step in the development of statistical packages is to be able to produce regression tables as fast and as naturally as performing regressions themselves. Regression tables ought to be produced easily, rapidly, and sequentially; they need to be displayed immediately on the computer screen. The utility of regression tables is much reduced if waited until the end of research. **outreg**, a program by John Gallup, has been modified and augmented extensively for this purpose. **outreg2** will immediately produce formatted regression tables. **shellout** opens them directly in programs associated with LaTex, Word, or Excel files. **seeout** will immediately display a regression table in the browser view.

Keywords: outreg, estimates table, regression tables, Excel, xml, Word, rtf

[1] Roy Wada is an AHRQ postdoctoral fellow in Health Service Research at UCLA and RAND.

1. Introduction

The ostensible reason for preparation of regression tables is to have them submitted to journals for publication. This is the position taken by John Gallup (1998, 1999, and 2000) and Ben Jann (2005, 2007) who presented their programs as a way to produce publication-ready tables. The implication is that regression tables are produced at the end of research. I propose to the contrary that regression tables are being used during research and not after. An informal survey would show that researchers are increasingly producing and using regression tables during their own research. Journals themselves require because they provide rapid access to the presented information for the readers.[2]

Regression tables are information management tools that concentrate information from various sources for immediate consumption by researchers. Important analytical tasks such as specification checks are performed in serial steps by including or omitting variables. The serial nature of these analytical tasks causes information to be spread out into a log file, which makes them difficult to compare them. Researchers have traditionally overcome this difficulty by printing hardcopies of log files and manually comparing them by flipping the pages back and forth.[3] The need for manual comparison is obviated by regression tables that gather the necessary information into one place. Hence regression tables are technological substitute for log files. Regression tables are information management tool that facilitates between-specification comparisons by extracting relevant information from regression outputs and placing them next to each other.

[2] Given the fact that journal publication is actually an infrequent event for researchers (perhaps less than twice a year for social scientists), it is highly doubtful that the ubiquity of user-written programs for producing regression tables is owed to journals alone.

[3] SAS users, for example, are known for printing hundreds of pages. Administrators will be pleased to learn that a significant cost saving can be realized with Stata, whose users print significantly less for this reason.

Unfortunately, the technology for making regression table has lagged considerably behind the technology for producing regression outputs. Production of regression outputs has been automated but not that of regression tables. Given the competitive market for statistical packages, it seems unlikely that their makers would neglect to make regression tables a prominent feature of their product. It is more likely that the difficulty of making of regression tables owes itself to the difficulty of information management, which makes them vital for research.

Computational information management requires information to be held together in memory for it to be processed prior to delivery to the end-user. Until very recently, however, most computers rarely possessed memory in abundance. The traditional solution was to write regression outputs to the hard-drive with little or no processing in the form of log files. As implied by its name, log files represent unprocessed information that is not readily read back into the computer. Hence the information contained in the log file could not be used for the purposes of making regression tables.

Stored files that can be read back by computer are known as data files. Unlike log files, data files possess standardized structure suitable for decoding. This was the innovation that was implemented by John Gallup (1998, 1999, 2000) in his program, **outreg**. The text files produced by **outreg** possess a standardized structure, effectively making them into data file. The insight behind **outreg** was to overcome the hardware difficulty with respect to the lack of sufficient memory by using the hard-drive as an extra form of storage. The standardized structure of regression tables is akin to that of data files, which makes it possible for a program like **outreg** to repeatedly code and decode the information contained therein.

With the growing availability of memory, however, it has become possible to store all necessary information together. A straightforward solution is to hold previous estimation results in the data file as they are produced. This approach is implemented by **estimates store** (see [R] **estimates**). The stored information may then be converted into regression tables with **estimates table**. Jann (2005) presented **estout** as wrappers for **estimates table** that will produce superior publication-quality regression tables and **eststo/esttab** as a solution to the "cumbersome" process of storing regression outputs.

It has been presented earlier in this paper, however, that regression tables are information management tools to be used during research for delivery of formatted information to the end-user for analytical purposes. For regression tables to accomplish this task, they need to be produced rapidly, easily, and sequentially; they need to be displayed immediately on the computer screen. The utility of regression tables would be greatly reduced if they were to be produced solely at the end of research.

A next logical step in the development of statistical packages is to be able to produce regression tables as fast and as naturally as performing regressions themselves. Such a program should require little or no learning and require the bare minimum of user involvement. Mindful of these issues, this paper presents **outreg2**, which is based on John Gallup's **outreg**. **outreg2** has several advantages. First, **outreg2** is invariant to the type of regression command.[4] **outreg2** will handle any regression output produced by Stata that meets the minimum convention for e-class returns (see [P] **ereturn**). Second, the programming codes have been updated from Stata 6 to Stata 8.2, taking advantage of expanded system limits. Third, new features have been added with the goal of lowering the hurdle against using regression tables, thus making it ideal for frequent use. Added

[4] The invariance is simply due to the standardized matrix nomenclature implemented by Stata Corporation.

features include automatic formatting of digits, an immediate access to regression table, automatic conversions to LaTeX, MS Word, and MS Excel formats, user-specified levels of significance, the ability to make tables from stored estimates, and syntaxes that allow shorthand and pre-command forms.

Companion commands make the implementation of these new features easy. **shellout** opens the produced regression table directly in programs associated with LaTex, Word, or Excel files. **seeout** will immediately display a regression table in the browser view. Hypertexts for **shellout** and **seeout** are placed on the results window, further lowering the hurdle against their use. The spreadsheet format presented **seeout** and Excel (with **shellout**) makes it very easy to read the results by providing horizontal and vertical cells that can be transverse by cursor keys and also avoiding the "wrapping text" problem inherent to results window. The rest of paper will detail these new functions in more detail.[5]

2 Basic Syntax and Main Options

An effort has been made to keep syntax simple. The most popular options, including using asterisks to indicate the levels of significance, have been made the default so that users are spared from specifying them. The full syntax for **outreg2** is the following. The advanced syntax employing shorthand or pre-command form will be discussed later.

. **outreg2** [*varlist*] [**[***estlist***]**] **using *filename*** [**,** *options*]

where *estlist* refers the list of stored estimates encased with the square brackets **[ ]**. Both the *varlist* and *estlist* will take the wildcard abbreviation with asterisks. The *options* are roughly categorized as Main Options, Output Files, Decimals and Displays, Auxiliary

[5] While they are visually attractive, non-spreadsheet tables such as those produced LaTex sometimes require "finger pointing" in which a finger is used as a visual aid. It is much easier to user the cursor on the screen.

Statistics (associated with each estimated coefficient), and Ancillary Statistics (associated with each regression output). This paper will focus on the core options and unique features that make it easy to use during research. The complete list of change is available in Appendix.

Basic Example

The basic plan for making a regression table with **outreg2** is to run it after each regression. The following example makes this concept clear.

```
. sysuse auto, clear
(1978 Automobile Data)

. regress mpg foreign weight headroom trunk length turn displacement
(regression output omitted)

. outreg2 using myfile
(note: file myfile.txt not found)
seeout
```

In the display window, **seeout** is colored blue, indicating that it is a clickable hypertext containing a Stata command. Clicking it will implement the **seeout** command using the created tab-delimited text file called *myfile.txt*, which has been stored in the current directory.

```
. seeout
  seeout using "myfile.txt"
seeout no change made
```

The data browser will pop open with the produced regression table that should look like this.

v1	v2	Notes_Titles
COEFFICIENT	mpg	
		Standard errors in parentheses
foreign	-1.967	*** p<.01,** p<.05, * p<.10
	(1.181)	
Weight	-0.00420**	

	(0.00202)
headroom	-0.0592
	(0.645)
trunk	-0.0122
	(0.159)
length	-0.0631
	(0.0644)
turn	-0.165
	(0.198)
displacement	0.000792
	(0.0103)
Constant	53.14***
	(7.584)
Observations	74
R-squared	0.68

seeout option thus provides a fast and easy access to the produced table without having to go through the time-consuming process of importing the created file into non-Stata software in order to see it.

Two things are notable in **seeout** view of the table. First, the notes at the bottom of table have been temporarily shunted to “Notes_Titles” columns at the right side of the table. The shunting prevents a potentially long note or titles from distorting the viewing widths. Second, all the digits have been automatically formatted, including the smaller numbers that normally would be either cut off or cause the rest of numbers to be trailed by an excessive number of digits. The automatic formatting of digits eliminates the need for such detailed management of the regression output that has been the bane of automated software. Customizing digits by hand is not only time-consuming, it also requires the user to look up the option to implement it and thus further adding to the administrative cost. The default setting for the auto-digits is 3 plus 1 for the coefficients. This means at least 3 significant digits not counting the ones (i.e. the first digit before the decimal point) are to be displayed. The numbers 1.234 and 0.123 are acceptable; 12.345

or 0.01234 will be trimmed. The auxiliary statistics are to display one less than for the coefficient and the ancillary statistics. This means the default is 2 plus one. It is possible to disable auto-digits by invoking the customary fixed or floating digits options. See Appendix for these options.

The following codes will append another regression. It is not necessary to specify **append**, which has been made the default. To write over an existing file, **replace** option should be used.

```
. regress mpg foreign weight headroom
(regression output omitted)

. outreg2 using myfile, seeout
seeout
```

By specifying **seeout** command, the browser window automatically pops open.[6] The contents of the browser window (not shown) are very similar to the last one except the latest regression out has been added in a column next to the previous regression output. By placing the two specifications side-by-side in an organized table, the impact of excluding few variables has been made clear. The data browser should be manually closed before proceeding. A clickable **seeout** hypertext is still available on the display screen whenever one wished to see the produced regression table again.

Conversion of Output Files

The produced table is physically stored on the current directory or the file location specified directly into **outreg2** command. Unless otherwise indicated by the user, the produced table is a tab-delimited text file with the .txt extension. Traditionally, the conversion of tab-delimited text file required careful management of import/export

[6] When specified this way, **outreg2** command is still running and will not stop until the browser window has been closed. **seeout** option is internally implemented at the very end of program, which makes it robust to possible disruptions such as the sudden loss of power to computer.

functions in non-Stata software.[7] This process has been automated in **outreg2** in two ways. First, the conversion process can be implemented from inside **outreg2** by using **word**, **excel**, and **tex** options. Appropriate extensions will be automatically assigned; it should not be assigned manually by the user. Second, the converted document can be made to open from inside Stata using the document's assigned software. For example, *myfile.doc* can be opened from inside Stata using MS Word. This feature is implemented by using **shellout** command.[8] A clickable hypertext is automatically left on the results window to implement it.

3 Advanced Syntax and Stored Estimates

You can automatically recall the stored estimates by specifying them from within **outreg2**. To distinguish stored regression outputs from *varlist*, the *estlist* should be placed within a pair of **[**square brackets**]**. The following series of command demonstrate the use of producing regression tables from stored regression estimates

```
. sysuse auto,clear
. regress mpg foreign weight headroom trunk length turn displacement
. estimates store Full
. regress mpg foreign weight headroom
. estimates store Restricted1
. regress mpg foreign weight
. estimates store Restricted2
. outreg2 [Full Restricted] using myfile, replace
```

outreg2 will take the stored estimates as wildcards (*). Try this:

```
. outreg2 [*] using myfile, replace
. outreg2 [R*] using myfile, replace
```

[7] The user could directly import them into the word processor or the spreadsheet of their choice. Another way is to make use of the Stat/Transfer, which does an excellent job of converting the tab-delimit file into a spreadsheet of choice.

[8] **shellout** is a wrapper for shell. Unlike **shell**, **shellout** does not require the pathway for non-Stata program to be specified and it will automatically close the DOS window. **shellout** only works with Window XP/NT platforms.

The *varlist* may be combined with the *estlist*. The *varlist* will take the wildcards as well, provided they exist in the stored regression estimates like this.

. outreg2 foreign weight [*] using myfile, replace

The use of shorthand syntax is allowed. **outreg2** will remember the last set of options you specified until the end of the day. The complete syntax for the shorthand is this:

. **outreg2** [*varlist*] [[*estlist*]] [, **replace seeout**] [: **command**]

The stored command will expire at mid-night to prevent the possible loss of completed table by inadvertently writing over a finished file. The following shorthand should be tried separately.

. outreg2, replace
. outreg2, seeout
. outreg2

The one-word syntax is the logical conclusion of the decentralized scheme for creating regression tables. Regression estimates do not need to be named. The following two options are excluded from the stored command: **seeout** and **replace**. These two must be specified each time you invoke **outreg2** through the shorthand. To change the stored options, you must invoke the full syntax with the specified using file. The *varlist* and *estlist* not stored with the command.

For someone who is in a hurry, outreg2 will take the pre-command syntax, provided the desired options have been stored by invoking them in the full syntax. The varlist and the estlist are still allowed under this syntax. This pre-command syntax is made available for the benefit of iterative users.

. outreg2 : reg mpg foreign weight headroom
. outreg2, replace : reg mpg foreign weight headroom

. outreg2
. seeout

4 Appendix

The following options are now implemented as the default: **append**, **3aster**, **coefastr**, **se**, and **nolabel**. **2aster** and **tstat** provide the access to the old options. sigsymb and 10pct are replaced with symbol. **nonobs** is no longer supported. The auxiliary statistics (standard error, etc) are no longer reported in absolute values. The levels of significance are strictly less than the values (used to be less than or equal to). The embedded spaces in the folder names are now accepted. **title**(list) can be added anytime. The .out extension is phased out in favor of .txt extension. A new file will be created if it did not exist. The past restrictions on the number and the size of variable names have been generally expanded to the system limit.

Main Options	
replace	overwrite the existing file; the default is to keep on appending
seeout	display with changes in the brower
label	append labels next to the variable names
onecol	specify one column for display multiple equations; the default is multiple columns. The default for onecol is wide. Also see long
ctitle(list)	choose own column titles instead of estimate names or types
addnote	add your own notes at the bottom

Output Files	
long	produces text file with long format for onecol option; causes word, excel, and tex to adopt the long same format if also specified
word	produces rich text file (rft) compatible with MS Word
excel	produces xml file compabile with MS Excel; only with Stata 9 or higher
tex	produces complete LaTeX file
tex(fragment)	produces a fragment of LaTeX file, may be specified concurrently with two following options
tex(nopretty)	no attempt is made to use italics and fancy display
tex(landscape)	produces a horizontal LaTex
quote	specifies quotation marks to encase each observation
comma	specifies comma-delimited, rather than tabs; consider using

quote as well

Decimals and Displays

auto(integer)	set automatic decimals; the default is auto(3)
bdec(numlist)	set fixed decimals for coefficients
bfmt(list)	set formats for the coefficients and standard errors or confidence intervals
tdec(integer)	set fixed decimals for the auxiliary statistics
rdec(integer)	set fixed decimals for the R-squared
adec(numlist)	set fixed decimals for the user-added statistics in addstat
noparen	no parenthesis
bracket	use brackets instead of parenthesis
noaster	no asterisks attached
2aster	two asterisks instead of three
symbol(list)	specifies symbols for 1% and 5% significance levels, and for 10% if 10pct)
nonotes	specifies that notes and asterisks not be included.

Auxiliary Statistics

eform	report exponentiated coefficients instead of coefficients
tstat	report t-statistics instead of standard errors
pvalue	report p-value of t-statistics instead of standard errors
ci	report confidence intervals instead of standard errors
level(#)	specifies in percent the confidence intervals
beta	report normalized beta coefficients instead of standard errors
nocons	do not report the constant term
noni	do not report groups in a panel estimation

Ancillary Statistics

e(scalars)	report e-class scalars
e(all)	report all available e-class scalars except N
addstat(list)	access e-class, r-class, or s-class scalar statistics
nor2	no R-squared reported
adjr2	report adjusted R-squared
margin	report that the marginal effects instead of the coefficient estimates truncreg, marginal from STB 52, or dtobit from STB 56
margin(u|c|p)	the unconditional, conditional, and probability marginal effects for dtobit
xstats	report the extra statistics included in the e(b) matrix

5 Acknowledgements

This paper benefited from Statalist conversations with Kit Baum and Steve Stillman. Functionality of **outreg2** is based on **outreg**, originally written by John Gallup. The name for **outreg2** was suggested by Richard Williams.

6 References

Gallup, J. L. 1998. sg97: Formatting regression output for published tables. Stata Technical Bulletin 46: 28–30.

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

—. 2000. sg97.2: Update to formatting regression output. Stata Technical Bulletin 58: 9–13.

Jann, Ben. 2005. Making regression tables from stored estimates. Stata Journal 5, Number 3: 288-308.

—. 2007. Making regression tables simplified. Stata Journal 7, Number 2: 227-44.