AgEcon SEARCH
RESEARCH IN AGRICULTURAL & APPLIED ECONOMICS

The World's Largest Open Access Agricultural & Applied Economics Digital Library

This document is discoverable and free to researchers across the globe due to the work of AgEcon Search.

Help ensure our sustainability.

Give to AgEcon Search

AgEcon Search
http://ageconsearch.umn.edu
aesearch@umn.edu

*Papers downloaded from **AgEcon Search** may be used for non-commercial purposes and personal study only. No other use, including posting to another Internet site, is permitted without permission from the copyright owner (not AgEcon Search), or as allowed under the provisions of Fair Use, U.S. Copyright Act, Title 17 U.S.C.*

The Stata Journal

Editors

H. Joseph Newton
Department of Statistics
Texas A&M University
College Station, Texas
editors@stata-journal.com

Nicholas J. Cox
Department of Geography
Durham University
Durham, UK
editors@stata-journal.com

Associate Editors

Christopher F. Baum, Boston College
Nathaniel Beck, New York University
Rino Bellocco, Karolinska Institutet, Sweden, and University of Milano-Bicocca, Italy
Maarten L. Buis, University of Konstanz, Germany
A. Colin Cameron, University of California–Davis
Mario A. Cleves, University of Arkansas for Medical Sciences
William D. Dupont, Vanderbilt University
Philip Ender, University of California–Los Angeles
David Epstein, Columbia University
Allan Gregory, Queen's University
James Hardin, University of South Carolina
Ben Jann, University of Bern, Switzerland
Stephen Jenkins, London School of Economics and Political Science
Ulrich Kohler, University of Potsdam, Germany
Frauke Kreuter, Univ. of Maryland–College Park
Peter A. Lachenbruch, Oregon State University
Jens Lauritsen, Odense University Hospital
Stanley Lemeshow, Ohio State University
J. Scott Long, Indiana University
Roger Newson, Imperial College, London
Austin Nichols, Urban Institute, Washington DC
Marcello Pagano, Harvard School of Public Health
Sophia Rabe-Hesketh, Univ. of California–Berkeley
J. Patrick Royston, MRC Clinical Trials Unit, London
Philip Ryan, University of Adelaide
Mark E. Schaffer, Heriot-Watt Univ., Edinburgh
Jeroen Weesie, Utrecht University
Ian White, MRC Biostatistics Unit, Cambridge
Nicholas J. G. Winter, University of Virginia
Jeffrey Wooldridge, Michigan State University

Stata Press Editorial Manager

Lisa Gilmore

Stata Press Copy Editors

David Culwell, Shelbi Seiner, and Deirdre Skaggs

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.

The *Stata Journal* is indexed and abstracted by *CompuMath Citation Index*, *Current Contents/Social and Behavioral Sciences*, *RePEc: Research Papers in Economics*, *Science Citation Index Expanded* (also known as *SciSearch*), *Scopus*, and *Social Sciences Citation Index*.

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

http://www.stata-journal.com

Subscriptions are available from StataCorp, 4905 Lakeway Drive, College Station, Texas 77845, telephone 979-696-4600 or 800-STATA-PC, fax 979-696-4601, or online at

http://www.stata.com/bookstore/sj.html

Subscription rates listed below include both a printed and an electronic copy unless otherwise mentioned.

U.S. and Canada		Elsewhere	
Printed & electronic		**Printed & electronic**	
1-year subscription	$115	1-year subscription	$145
2-year subscription	$210	2-year subscription	$270
3-year subscription	$285	3-year subscription	$375
1-year student subscription	$ 85	1-year student subscription	$115
1-year institutional subscription	$345	1-year institutional subscription	$375
2-year institutional subscription	$625	2-year institutional subscription	$685
3-year institutional subscription	$875	3-year institutional subscription	$965
Electronic only		**Electronic only**	
1-year subscription	$ 85	1-year subscription	$ 85
2-year subscription	$155	2-year subscription	$155
3-year subscription	$215	3-year subscription	$215
1-year student subscription	$ 55	1-year student subscription	$ 55

Back issues of the *Stata Journal* may be ordered online at

http://www.stata.com/bookstore/sjj.html

Individual articles three or more years old may be accessed online without charge. More recent articles may be ordered online.

http://www.stata-journal.com/archives.html

The *Stata Journal* is published quarterly by the Stata Press, College Station, Texas, USA.

Address changes should be sent to the *Stata Journal*, StataCorp, 4905 Lakeway Drive, College Station, TX 77845, USA, or emailed to sj@stata.com.

Copyright © 2014 by StataCorp LP

Copyright Statement: The *Stata Journal* and the contents of the supporting files (programs, datasets, and help files) are copyright © by StataCorp LP. The contents of the supporting files (programs, datasets, and help files) may be copied or reproduced by any means whatsoever, in whole or in part, as long as any copy or reproduction includes attribution to both (1) the author and (2) the *Stata Journal*.

The articles appearing in the *Stata Journal* may be copied or reproduced as printed copies, in whole or in part, as long as any copy or reproduction includes attribution to both (1) the author and (2) the *Stata Journal*.

Written permission must be obtained from StataCorp if you wish to make electronic copies of the insertions. This precludes placing electronic copies of the *Stata Journal*, in whole or in part, on publicly accessible websites, fileservers, or other locations where the copy may be accessed by anyone other than the subscriber.

Users of any of the software, ideas, data, or other materials published in the *Stata Journal* or the supporting files understand that such use is made without warranty of any kind, by either the *Stata Journal*, the author, or StataCorp. In particular, there is no warranty of fitness of purpose or merchantability, nor for special, incidental, or consequential damages such as loss of profits. The purpose of the *Stata Journal* is to promote free communication among Stata users.

The *Stata Journal* (ISSN 1536-867X) is a publication of Stata Press. Stata, Stata, Stata Press, Mata, Mata, and NetCourse are registered trademarks of StataCorp LP.

The Stata Journal (2014)
14, Number 4, pp. 965–974

Collecting and organizing Stata graphs

Joseph D. Wolfe
University of Alabama at Birmingham
Birmingham, AL
jdwolfe@uab.edu

Shawn Bauldry
University of Alabama at Birmingham
Birmingham, AL
sbauldry@uab.edu

Abstract. Stata includes a powerful set of tools for constructing a wide array of graphs. Stata's graphing capabilities are well suited for describing exploratory or preliminary data analyses as well as producing publication-quality graphics. Currently, however, Stata does not have a built-in suite of commands for constructing various types of files (for example, HTML, TEX, or RTF files) to display multiple graphs. Such files can be invaluable for organizing and facilitating the interpretation of the numerous graphs needed throughout an analysis or in the final stage of a project. In this article, we provide an overview of two commands, `graphsto` and `graphout`, designed to organize and process multiple graphs across various file types.

Keywords: gr0060, graphsto, graphout, graph, graph export, HTML, RTF, TEX

1 Introduction

For many Stata users, graphs play an essential role in all aspects of an analysis. Graphs are often used to explore univariate and bivariate distributions of data, to display postestimation results for diagnostics, to explore the implications of models, and to efficiently and elegantly display the results of an analysis. Stata provides a powerful set of tools for constructing a variety of graphs for all of these uses (Mitchell 2012a,b).

Stata, however, does not currently have a built-in suite of commands for managing and reviewing the many graphs that emerge at various phases of a research project. It would be helpful, for instance, to have one file that contains all the preliminary descriptive graphs (for example, univariate or bivariate distributions or matrix scatterplots) so that users can refer to a single location to easily access, compare, and interpret the graphs. Similarly, it would be helpful to maintain a file that includes a series of graphs displaying postestimation diagnostics (for example, residual plots) or collects all the graphs to be included in an article.

Stata allows users to save a graph as one file in various formats (for example, PNG, EPS, WMF, and PDF) via the `graph export` command. However, to review and compare graphs outside Stata, users must manually move saved graphs to an external word processor like Microsoft Word or create links to the graphs in programs like LATEX. When a user is interested in only a few graphs, this is not a difficult procedure. However, when users want to review more than a few graphs, the process can become cumbersome and tedious.

A few ad-hoc procedures for constructing files with multiple Stata graphs are currently available online. In addition, Lo Magno (2013) developed a set of procedures that

© 2014 StataCorp LP

allows users to transfer graphs obtained in Stata directly into Word documents, but these procedures are limited to Word running on Windows-based systems. Finally, users can also write their own Stata code to produce markup language code (Gini and Pasquini 2006). None of these approaches, however, help users to collect multiple Stata graphs in varying document types.

In this article, we present the two commands that are capable of quickly organizing Stata graphs and saving them to a single file for viewing. The first command, `graphsto`, saves users' graphs and marks them for `graphout`. After users have created all their graphs, they can use `graphout` to create a new HTML, RTF, or TEX file containing links to the graphs. When users open this file, the graphs will be displayed and numbered. In the remainder of this article, we provide an overview of these two commands along with examples illustrating their use.

2 Description and syntax

Together, `graphsto` and `graphout` allow users to quickly create an RTF, an HTML, or a TEX file that can display a variety of graph formats. For exploratory research, for example, users often create HTML files that display PNG graphs. To share graphs with colleagues, on the other hand, users tend to use PNG graphs to create an RTF file, which can then be converted to a PDF file to embed graphs. Note that `graphout` will create HTML files with PNG graph formats only; RTF files on a personal computer can display TIF, EMF, WMF, PNG, and EPS files. RTF files on Macs can create RTF files that can display EPS, PNG, and PDF graph formats. TEX files can display multiple graph formats depending on the TEX package used.

The `graphsto` command is a wrapper for the `graph export` command, which outputs graphs as a variety of file types, for example, PNG, EPS, and PDF (see [G-2] **graph export** for a full list). `graphsto` exports graphs as the output format specified by *.suffix* and collects the names of the graphs as specified by *graphname*. The user can also use `graphsto` to add a title or note. The syntax for `graphsto` is as follows:

`graphsto` *graphname*`.`*suffix* [`,` *options*]

Table 1 describes the optional arguments for `graphsto`.

Table 1. Options for `graphsto`

options	Description
`title(`*string*`)`	add a title above graph
`note(`*string*`)`	add a note below graph
`replace`	overwrite an existing file
`directory`	list the graph names stored in the global macro
`clear`	clear the graph names stored in the global macro

Users can also clear stored graph names from memory by typing `graphsto clear`, and they can obtain a list of the stored graph names by typing `graphsto dir`.

The `graphout` command constructs an RTF, an HTML, or a TEX file (as specified by *.suffix*) that contains links to the graphs stored by `graphsto`. The new file will display the specified graphs as long as the links are not broken. To embed the graphs, the user should create a PDF copy of the file containing the links to the graphs. The syntax for `graphout` is as follows,

`graphout using` *filename.suffix* [`,` *options*]

where *filename* is the name of the file that will contain links to the graphs that have been stored with `graphsto`. Table 2 describes the options for `graphout`.

Table 2. Options for `graphout`

options	Description
Output (HTML, RTF, and TEX)	
`replace`	overwrite an existing file
`append`	append the output to an existing file
Size (HTML and TEX only)	
`height`(*string*)	adjust the height of graphs
`width`(*string*)	adjust the width of graphs
`scale`(*numlist*)	adjust the scale of graphs
Layout (RTF, HTML, and TEX)	
`alignment`(*string*)	center, right-align, or left-align the graphs
Text (RTF, HTML, and TEX)	
`nocount`	do not add "`Figure` #" to each graph
`basecount`(*numlist*)	specify the starting # for "`Figure` #"
Advanced (TEX only)	
`placement`(*string*)	specify the placement of a float in TEX file
`fbarrier(#)`	add "\`FloatBarrier`" to every graph in TEX file
`document`	make a stand-alone TEX document
`label`(*string*)	add "\`label`{*string*}" to TEX files

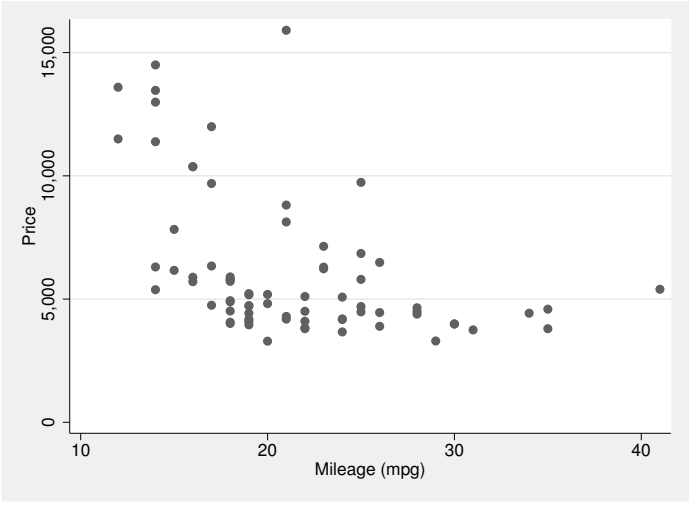
3 Examples

3.1 Basic example

The following example illustrates using `graphsto` and `graphout` to create an RTF document containing one graph. For this example, we create a scatterplot of `price` and `mpg` using Stata's built-in dataset `auto.dta`. We use `graphsto` to title the graph and export it to a PNG format. This command also stores the name of the graph, which we specified here as `Ex1`. Because we use `graphsto`, it is not necessary to specify the title of the graph in `graphout`.

```
. graphsto, clear
. sysuse auto
. graph twoway scatter price mpg, scheme(sj)
. graphsto Ex1.png, replace title(Price by MPG)
. graphout using Ex1.rtf, replace
```

Executing this code produces an RTF file with the specified title and a link to the following graph.



For the links to the graphs in the RTF file (and in all the other files formats) to work, the graphs must remain in the same file directory as the RTF file. If a user moves the RTF file to a different directory without moving the graphs as well, then the links will fail and no graphs will appear. This is important to remember when working with collaborators. It is possible, however, to save a copy of the original RTF, HTML, or TEX file with the links to the graphs as a PDF file that contains all the graphs and does not require the original graph files to be in the same directory.

3.2 Collecting graphs for exploratory analysis

During the preliminary phase of a project, we often find ourselves constructing numerous graphs to explore the data. For this type of exploratory analysis, creating HTML files can be especially useful. The files can be opened in the user's favorite web browser, and the user can easily review updates to the graphs by clicking on the refresh button. For the following example, we again use Stata's `auto.dta`. We construct histograms by `price` and `mpg` and then a scatterplot by `price` and `mpg`. We also illustrate two formatting features that are options for `graphout`: centering the graphs on the page and resizing the graphs while maintaining the same scale. There are also options to adjust the height and width of the graphs directly.

```
. graphsto, clear
. sysuse auto
. histogram price, scheme(sj)
. graphsto Ex2_1.png, replace title(Histogram for price)
. histogram mpg, scheme(sj)
. graphsto Ex2_2.png, replace title(Histogram for mpg)
. graph twoway scatter price mpg, scheme(sj)
. graphsto Ex2_3.png, replace title(Scatterplot price by mpg)
. graphout using Ex2.html, replace alignment(center) scale(0.5)
```

Executing this code produces an HTML file with links to the following graphs.

Figure 1: Histogram for price

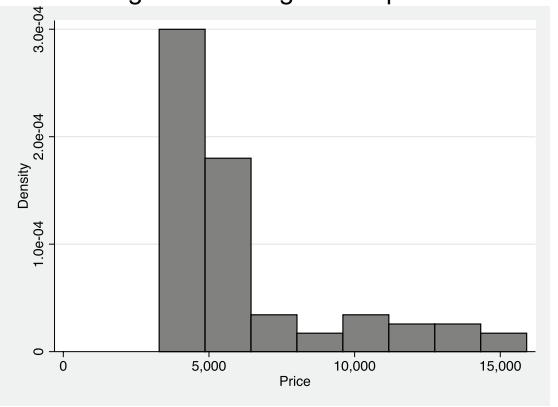


Figure 2: Histogram for mpg

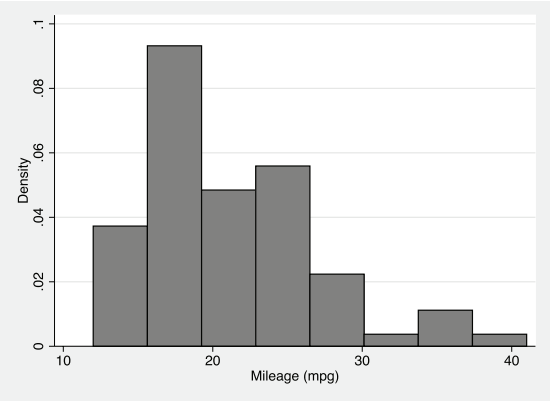
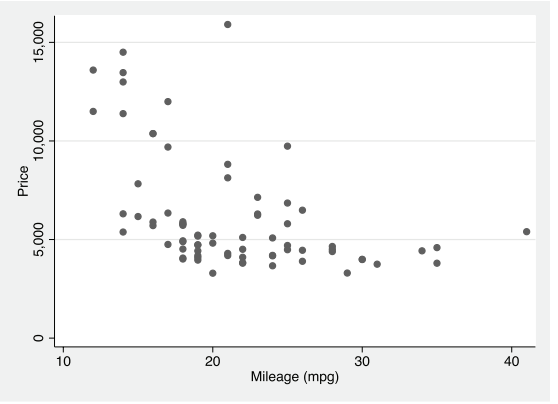


Figure 3: Scatterplot price by mpg



3.3 Constructing graphs in loops

Here we illustrate a more advanced use of `graphsto` and `graphout`. Our third example involves quickly constructing numerous graphs using loops. This procedure can make exploratory or diagnostic analyses, for instance, much more efficient than examining the graphs one by one. For this example, we examine scatterplots that include linear fits for `price` and six different variables potentially associated with `price`. After we construct each graph, we use `graphsto` to export the graph along with a title. After the loop, we use `graphout` to save all the collected graphs in one HTML file with the graphs left-aligned and resized to the specified height. For HTML files, the `height()` and `width()` options are automatically specified by pixels. For TEX files, users can specify the units for height or width (for example, `2in` for 2 inches or `5cm` for 5 centimeters).

```
. graphsto, clear
. sysuse auto
. foreach var of varlist mpg rep78 headroom trunk weight length {
  2.         graph twoway scatter price `var´ || lfit price `var´, scheme(sj)
  3.         graphsto Ex3_`var´.png, replace title(Scatterplot price by `var´)
  4. }
. graphout using Ex3.html, replace alignment(left) height(500)
```

3.4 Adding a graph to a table of regression estimates

Our fourth example involves appending a graph to a set of model estimates stored using Jann's `eststo` and `esttab` suite of commands (Jann 2005, 2007). The append feature of `graphout` allows a user to add a graph (or set of graphs) to an already existing file. This feature introduces many interesting possibilities. For instance, a user can add a diagnostic graph to a TEX file that contains a table of regression estimates. The following example illustrates this using Stata's `auto.dta`.

We estimate a regression model with the mile per gallon (`mpg`), repair record in 1978 (`rep78`), weight (`weight`), and length (`length`) predicting the price of a car, and we store the results using the `eststo` command. We are interested in examining the Studentized residuals, a common diagnostic for identifying potential outliers. Following our regression, we use `predict` to obtain the Studentized residuals, and we then create a histogram and box plot to examine the distribution. We combine the histogram and box plot into one graph with the two plots side by side. Finally, we use the `esttab` command to create a TEX file with the regression results and then use `graphout` to append the Studentized residual plots to the regression table.

```
. graphsto, clear
. sysuse auto
. qui regress price mpg rep78 weight length
. eststo m1
. predict rstu, rstu
. tempfile g1 g2
. histogram rstu, scheme(sj) nodraw saving(`g1´)
. graph box rstu, scheme(sj) nodraw saving(`g2´)
. graph combine "`g1´" "`g2´", scheme(sj)
. graphsto Ex4.png, replace title(Studentized residuals)
. esttab m1 using Ex4.tex, replace b(%9.3f) se(%9.3f) nogap
>   compress title("Price regression") alignment(center)
. graphout using Ex4.tex, append scale(0.25)
```

Executing the above code produces a TEX file with code for the regression results and a link to the graph. The following figure illustrates the compiled TEX code for this example:

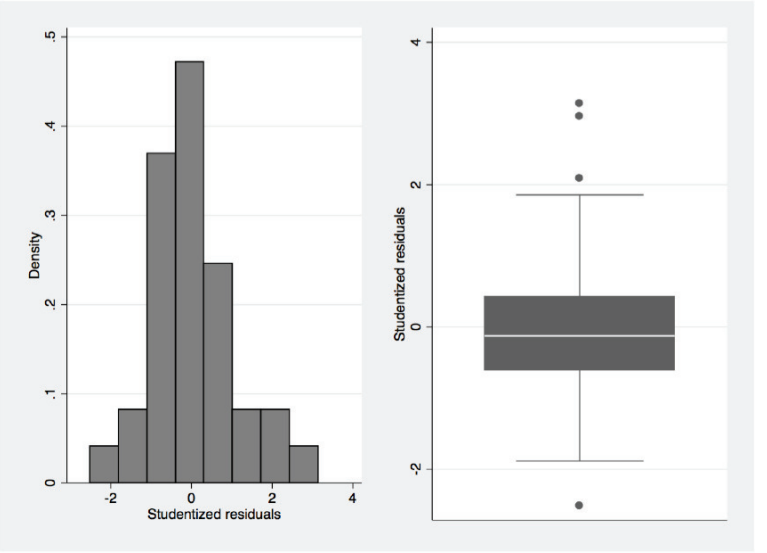
Table 1: Price regression

	(1) price
mpg	-106.712
	(81.158)
rep78	910.986**
	(304.527)
weight	4.960***
	(1.120)
length	-115.018**
	(38.565)
_cons	11934.507*
	(5774.178)
N	69

Standard errors in parentheses

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

Figure 1: Studentized residuals



4 Conclusion

Stata has excellent graphing capabilities for tasks ranging from describing exploratory analysis to producing publication-quality figures. The commands we describe in this article, `graphsto` and `graphout`, are designed to help users collect, organize, and present multiple Stata graphs. The commands allow users to construct or append graphs to three different file types (HTML, RTF, and TEX). The examples illustrate some of the benefits of this capability to manage and view multiple Stata graphs in one file.

5 Acknowledgments

The `graphout` and `graphsto` programs are modeled on and meant to complement Ben Jann's `eststo` and `esttab` packages (Jann 2005, 2007). We are also grateful for Eng's article on file filters (Eng 2007). The idea for `graphout` came from an email on Statalist written by Maartin Buis. His email explained how to add graphs to an HTML file and inspired the `graphout` and `graphsto` programs.

6 References

Eng, J. 2007. File filtering in Stata: Handling complex data formats and navigating log files efficiently. *Stata Journal* 7: 98–105.

Gini, R., and J. Pasquini. 2006. Automatic generation of documents. *Stata Journal* 6: 22–39.

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

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

Lo Magno, G. L. 2013. Sar: Automatic generation of statistical reports using Stata and Microsoft Word for Windows. *Stata Journal* 13: 39–64.

Mitchell, M. N. 2012a. *Interpreting and Visualizing Regression Models Using Stata*. College Station, TX: Stata Press.

———. 2012b. *A Visual Guide to Stata Graphics*. 3rd ed. College Station, TX: Stata Press.

About the authors

Joseph D. Wolfe is an assistant professor in the department of sociology at the University of Alabama at Birmingham.

Shawn Bauldry is an assistant professor in the department of sociology at the University of Alabama at Birmingham. His methodological research interests focus on the development of structural equation models and approaches to handling missing data.