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, 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.

<table>
<thead>
<tr>
<th>U.S. and Canada</th>
<th>Elsewhere</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Printed &amp; electronic</strong></td>
<td><strong>Printed &amp; electronic</strong></td>
</tr>
<tr>
<td>1-year subscription</td>
<td>$115</td>
</tr>
<tr>
<td>2-year subscription</td>
<td>$210</td>
</tr>
<tr>
<td>3-year subscription</td>
<td>$285</td>
</tr>
<tr>
<td>1-year student subscription</td>
<td>$85</td>
</tr>
<tr>
<td>1-year institutional subscription</td>
<td>$345</td>
</tr>
<tr>
<td>2-year institutional subscription</td>
<td>$625</td>
</tr>
<tr>
<td>3-year institutional subscription</td>
<td>$875</td>
</tr>
</tbody>
</table>

**Electronic only** | **Electronic only**
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1-year subscription</td>
<td>$85</td>
</tr>
<tr>
<td>2-year subscription</td>
<td>$155</td>
</tr>
<tr>
<td>3-year subscription</td>
<td>$215</td>
</tr>
<tr>
<td>1-year student subscription</td>
<td>$55</td>
</tr>
</tbody>
</table>

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 © 2015 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, file servers, 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.
A note on adding objects to an existing twoway graph

Ben Jann
University of Bern
Bern, Switzerland
ben.jann@soz.unibe.ch

Abstract. In Stata, graphs are usually generated by one call to the graph command. Sometimes, however, it would be convenient to be able to add objects to a graph after the graph has been created. In this article, I provide a command called addplot that offers such functionality for twoway graphs, capitalizing on an undocumented feature of Stata’s graphics system.

Keywords: gr0065, addplot, graph twoway, graphics, graph editing

1 Introduction

Even very complex plots with multiple overlays of data, different plot types, and verbose labeling are usually created by one graph command. This works well in most situations, but sometimes, it would be convenient to be able to modify a graph after it has been created. For example, you might need to modify a graph for which you have only the graph file but not the data used to generate the graph. Furthermore, certain modifications are difficult to achieve within a standard graph command.

Ex-post modifications of graphs can be conducted using Stata’s Graph Editor (see [G-1] graph editor), a point-and-click device that lets you add labels and lines or other objects and can also be used to move objects around or delete them. Twoway graphs (see [G-2] graph twoway), however, can also be built by applying multiple graph commands one after another, a feature of Stata’s graphics system that is undocumented and therefore unknown to most users. Here I present the command addplot to facilitate sequential construction of twoway graphs.

Note that many Stata commands feature an addplot() option for adding extra twoway plots to a graph (see [G-3] addplot_option). The addplot command presented here differs from this option in that it is applied as an independent call after the original graph has been created.
A note on adding objects to an existing twoway graph

2 Syntax

`addplot` is a simple wrapper compiling a call to Stata’s graphics system that appends additional twoway commands to an existing graph. Its syntax is as follows:

```stata
addplot [graphname] [numlist] : twoway_plots
```

where `graphname` is the name of the memory graph to be modified. The current (topmost) graph is used if `graphname` is omitted.

`numlist` provides the numbers of the subgraphs to be modified (shorthand conventions as explained in [U] 11.1.8 `numlist` may be applied). This is useful if a graph contains multiple twoway subgraphs that were created using the `by()` option (see [G-3] `by_option`) or `graph combine` (see [G-2] `graph combine`). All twoway subgraphs found in the graph are modified if `numlist` is omitted.

`twoway_plots` is any twoway graph command (without the leading `graph twoway` statement), possibly containing multiple plots as in

```stata
addplot: (plot1) (plot2) ... [, options]
```

or

```stata
addplot: plot1 || plot2 ... [||, options]
```

or possibly containing only twoway options as in

```stata
addplot: , options
```

Technical note

To ensure that the added data points fit in the plot region even if they are outside the range of the axes of the existing plot, `addplot` recompiles the axes of the plot using default settings. To prevent `addplot` from recompiling the axes, you can apply an (undocumented) option called `norescaling` to the added `twoway` command. That is, to preserve the existing axis labels and ranges, type

```stata
addplot: [plots], [options] norescaling
```

Technical note

`addplot` may cause a legend to be added to the graph. To suppress adding a legend, apply option `legend(off)` to the added `twoway` command (see the examples below). Furthermore, `addplot` always re-creates an existing legend using default legend keys. To preserve or create a custom legend, you must specify an appropriate `legend(order())` option with the added `twoway` command (see [G-3] `legend_options`).

```stata
addplot: , legend(order())
```
3 Examples

The general procedure is to first create a basic twoway graph and then apply `addplot`, possibly multiple times, to complement the graph. The following is a basic example (see figure 1 for the resulting graph):

```stata
. sysuse auto
(1978 Automobile Data)
. twoway scatter price mpg
. addplot: scatter price mpg if price>15000 | mpg>40, msymbol(i)
    > mlabel(make) mlabposition(9) legend(off)
. addplot: pcarrowi 14000 23 15500 21.5 (3) "expensive car"
    > 2000 38 5000 40.7 (6) "high mileage car"
```

Figure 1. Basic example

The first `addplot` command labels some outliers by superimposing a labeled scatterplot with invisible markers. The `legend(off)` option has been added to omit the legend that is added by default in graphs that contain multiple plots.

The second `addplot` command adds the arrows and additional labels using the `pcarrowi` plot type (see [G-2] `graph twoway pcarrowi`).

Note that the two modifications could also have been applied in one step by using one `addplot` call.

```stata
. twoway scatter price mpg
. addplot: (scatter price mpg if price>15000 | mpg>40, msymbol(i)
    > mlabel(make) mlabposition(9)) (pcarrowi 14000 23 15500 21.5 (3)
    > "expensive car" 2000 38 5000 40.7 (6) "high mileage car"),
    > legend(off)
```
A note on adding objects to an existing twoway graph

The previous graph can easily be produced by one graph command, so there is not much advantage of using `addplot`.

```
. twoway (scatter price mpg) (scatter price mpg if price>15000 | mpg>40, >   msymbol(i) mlabel(make) mlabposition(9)) (pcarrowi 14000 23 15500 21.5 (3) >   ”expensive car” 2000 38 5000 40.7 (6) ”high mileage car”), >   legend(off)
```

However, `addplot` can also apply modifications that would be difficult to achieve within one graph command, as the following example illustrates (see figure 2 for the resulting graph):

```
. scatter mpg turn, by(foreign) jitter(2) msymbol(Oh) 
. summarize mpg if foreign==0, meanonly 
. addplot 1: , yline(’r(mean)’) norescaling 
. summarize mpg if foreign==1, meanonly 
. addplot 2: , yline(’r(mean)’) norescaling 
. summarize mpg, meanonly 
. addplot: , yline(’r(mean)’, lpattern(dash)) norescaling
```

In this example, a graph containing separate plots for two subsamples is produced using the `by()` option (see [G-3] `by_option`). Then, two `addplot` calls are used to add subgroup-specific medians as solid reference lines (`addplot 1` modifies the first subgraph; `addplot 2` modifies the second subgraph). In this example, the `norescaling` option is applied to prevent rescaling of the axes of the individual subgraphs (see the technical note above). Finally, a third `addplot` command is used to add the global median as a dashed reference line in both subgraphs. Subgraph-specific reference lines can also be manually added using the graph editor. However, using `addplot` seems
more convenient, especially if additional computations are involved as in the present example.

An interesting aspect of the `norescaling` option is that it allows you to place objects outside the plot region. The following is an example in which nonstandard labeling is added to the right of the plot region (see figure 3 for the resulting graph):

```stata
.scatter mpg turn, jitter(2) msymbol(Oh)
.addplot: (scatteri 10.0 52 10.0 53 19.5 53 19.5 52, recast(line) lp(l))
> (scatteri 20.5 52 20.5 53 29.5 53 29.5 52, recast(line) lp(l))
> (scatteri 30.5 52 30.5 53 40.0 53 40.0 52, recast(line) lp(l)),
> graphregion(margin(r=11)) legend(off) norescaling
> text(15 52.5 "low" 25 52.5 "medium" 35 52.5 "high",
> orientation(rvertical))
```

![Figure 3. Adding objects outside the plot region](image)

Three `scatteri` commands are used to draw the brackets (see [G-2] `graph twoway scatteri`); the `text()` option is used to add the labels (see [G-3] `added_text_options`); and the `graphregion()` option is used to increase the margin on the right of the graph to make space for the brackets and labels (see [G-3] `region_options`).

## 4 Acknowledgment

Vince Wiggins from StataCorp provided critical help and comments.

### About the author

Ben Jann is a professor of sociology at the University of Bern, Switzerland. His research interests include social-science methodology, statistics, social stratification, and labor market sociology. Recent publications include articles in *Sociological Methodology, Sociological Methods and Research, Public Opinion Quarterly*, and the *American Sociological Review*. 