

## Stata tip 72: Using the Graph Recorder to create a pseudograph scheme

Kevin Crow  
StataCorp  
College Station, TX  
kcrow@stata.com



Following the update of 25 February 2008, the Graph Editor can now record a series of edits, name the recording, and apply the edits from the recording to other graphs. You can apply the recorded edits from the Graph Editor or from the command line. The edits can be applied from the command line when a graph is created, when it is used from disk, or whenever it is the active graph. See *Graph Recorder* in `help graph editor` for creating and playing recordings in the Graph Editor. For applying edits from the command line, see `help graph play` and the option `play(recordingname)` in `help std_options` and `help graph use`.



In this tip, I focus on the use of the Graph Recorder to create a graph scheme. A graph scheme specifies the overall look of the graph. If you want to create your own look for your graphs, you will want to create a scheme file. There is a problem with scheme files, however, because unless you know exactly how you want your scheme to be set up, creating a scheme can be very time consuming.

A shortcut to creating a scheme file is saving a graph recording to disk and replaying that recording on your graphs by using the `play()` option of the `graph` command. Using the Graph Recorder to create your graph recording also allows you to tinker with your graph's look on the fly without having to edit a scheme file. Let's walk through an example.

Suppose you want to create several graphs for a report and you want those graphs to have a specific look. To create your recording, you first need to draw the first graph of the report. Try

```
. sysuse auto  
(1978 automobile data)  
. scatter mpg weight
```

Now that your graph is in the Graph window, you can right-click on the window and select **Start Graph Editor** to start the Graph Editor. Next start the Graph Recorder by clicking on the **Start Recording** button, , so that you save your changes to memory. Once the graph looks the way you want, you then click on the same button,  (which now has the tool tip **End Recording**), to save your changes to a `.grec` file. By default, Stata saves `.grec` files to your `PERSONAL/grec` directory.

If you want to tinker with the graph during a recording, but you do not want the changes to be saved to the `.grec` file, click on the **Pause** button, , to temporarily stop saving the changes. To unpause the Recorder, click on the **Pause** button, , again.

Now that you have a recorder file saved to disk, you can type your next **graph** command in your do-file or from the Command window and apply your scheme to the graph with the **play()** option. For example,

```
. scatter mpg turn, play("test.grec") saving(test1, replace)
```

Also, if you have graphs already created and saved to disk, you can apply your recording to those graphs by using the **play()** option of **graph use**. For example,

```
. graph use "oldfile.gph", play("test.grec") saving("new_file", replace)
```

## Software Updates

dm89\_1: Dropping variables or observations with missing values. N. J. Cox. *Stata Technical Bulletin* 60: 7–8. Reprinted in *Stata Technical Bulletin Reprints* vol. 10, pp. 44–46.

`dropmiss` has been updated in style and content. A new `force` option has been added, which must be specified if the data in memory have been changed.

gr0024\_1: Graphical representation of interactions. F. M.-S. Barthel and P. Royston. *Stata Journal* 6: 348–363.

A bug caused an error in the calculation of the hazard ratio or relative risk of treatment for the second level of the covariate alone. This has now been fixed.

st0015\_5: Concordance correlation coefficient and associated measures, tests, and graphs. T. J. Steichen and N. J. Cox. *Stata Journal* 7: 444; 6: 284; 5: 471; 4: 491; 2: 183–189. *Stata Technical Bulletin* 58: 9; 54: 25–26; 45: 21–23; 43: 35–39. Reprinted in *Stata Technical Bulletin Reprints* vol.10, p.137; vol.9, pp.169–170; vol.8, pp.137–145.

The dialog box has been modified to correct visual layout problems. Its functionality has not changed.

st0100\_1: Decomposing inequality and obtaining marginal effects. A. López-Feldman. *Stata Journal* 6: 106–111.

Bugs have been fixed to avoid problems in the estimation of marginal effects as well as the unintended deletion of variables from the users' data. Some typos in the help and the output have also been corrected.

st0150\_1: A Stata package for the estimation of the dose–response function through adjustment for the generalized propensity score. M. Bia and A. Mattei. *Stata Journal* 8: 354–373.

Some references in `doseresponse` to variables named `treatment_level_plus` or similar were incorrect in certain circumstances. These have been fixed. The handling of predicted probabilities in `dosereponse` when fitting `mlogit`, `mprobit`, `ologit`, or `oprobit` models has been improved.

sxd1\_4: Random allocation of treatments in blocks. P. Ryan. *Stata Journal* 8: 146; *Stata Technical Bulletin* 54: 49–53; 50: 36–37; 41: 43–46. Reprinted in *Stata Technical Bulletin Reprints*, vol. 9, pp.353–358, vol. 9, pp.352–353; vol.7, pp.297–300.

A small bug causing graceless exits for Stata/SE and Stata/MP users has been fixed.