The Stata Journal (2009)
9, Number 4, pp. 643–647

Stata tip 81: A table of graphs

Maarten L. Buis
Department of Sociology
Tübingen University
Tübingen, Germany
maarten.buis@uni-tuebingen.de

Martin Weiss
Department of Economics
Tübingen University
Tübingen, Germany
martin.weiss@uni-tuebingen.de

A useful option within the `graph` command is the `by()` option ([G] ***by_option***). This option allows us to explore graphs across one or more additional categorical variables. This Stata tip will discuss how to fine-tune the display of such a graph when we want to compare graphs across two variables. Consider, for example, the graph created with the following set of commands and displayed in figure 1:

```
. sysuse auto
(1978 Automobile Data)
. fillin rep78 foreign
. twoway scatter price mpg, by(foreign rep78, cols(5) compact)
```

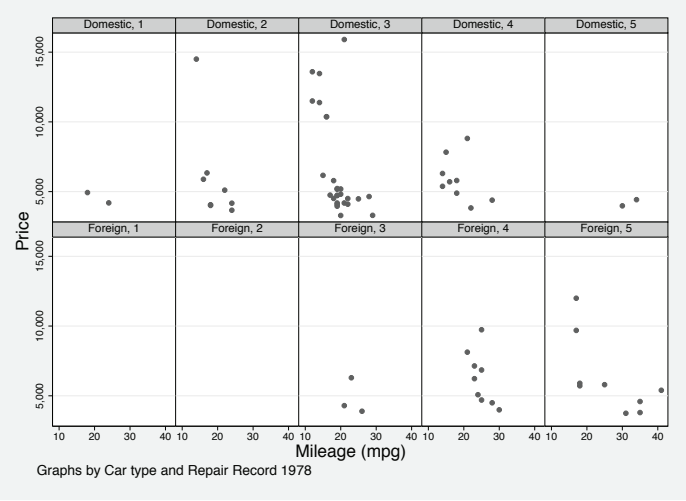

Figure 1. Comparing the relationship between price and mileage across repair status and car type (with a title for every subgraph)

There is a tabular look to this graph, with the rows representing the origin of the cars and the columns representing their repair status. This design helps us identify the relevant comparisons in a way similar to so-called trellis graphics (Cleveland 1993; Becker, Cleveland, and Shyu 1996) and lattice graphics (Sarkar 2008).

The first step in creating such a graph is to make sure that the columns and rows are properly aligned. We did that by using the `cols()` suboption of the `by()` option to make sure that there are as many columns as there are categories of `rep78`. We also used

the `fillin` command ([D] **fillin**) to make sure that there is at least one observation for every combination of `rep78` and `foreign`, whereby those combinations that did not exist in the original data contain only missing values—here foreign cars with a repair status of poor (1) or fair (2). This has the effect of creating empty graphs for those nonexisting combinations. An alternative would be to use the `holes()` suboption within the `by()` option, which would leave the area otherwise occupied by a subgraph completely blank. Within the table-like logic of this graph, we like to use the empty graphs for combinations that in principle could occur but did not, while reserving those completely blank spaces for combinations that for logical reasons cannot occur. In our example, there is no reason why foreign cars could not have a poor or fair repair status, so we used the empty graphs. If we had compared graphs across gender and pregnancy status, we would instead have left a hole for the combination "pregnant men".

There is still a problem with this graph: the subgraph labels contain superfluous information and distract from the tabular design. Ideally, we would like to have titles at the top representing the columns and titles on the right representing the rows. To get the column titles, all we would need to do is keep only the top titles and suppress all the others. In the example below, this is done in the following way: the variables `rep78` and `foreign` are combined into one variable with the `egen` function `group()` ([D] **egen** and Cox [2007]). Then value labels that correspond to the column titles are assigned to the first five categories (`rep78` contains five categories, so in this graph there will be five columns). The remaining values of this new variable are assigned the value label `" "`. This will suppress the remaining titles. This example also adds titles to the column axis and the row axis. The resulting graph is shown in figure 2.

```
. egen group = group(foreign rep78)
(5 missing values generated)
. label define group 1  "Poor"
>                    2  "Fair"
>                    3  "Average"
>                    4  "Good"
>                    5  "Excellent"
>                    6  " "
>                    7  " "
>                    8  " "
>                    9  " "
>                    10 " "
. label value group group
. twoway scatter price mpg,
> by(group, cols(5)
>    r1title("Car type",
>            orientation(rvertical)
>            size(medsmall))
>    t1title("Repair status",
>            size(medsmall))
>    note("") compact)
```

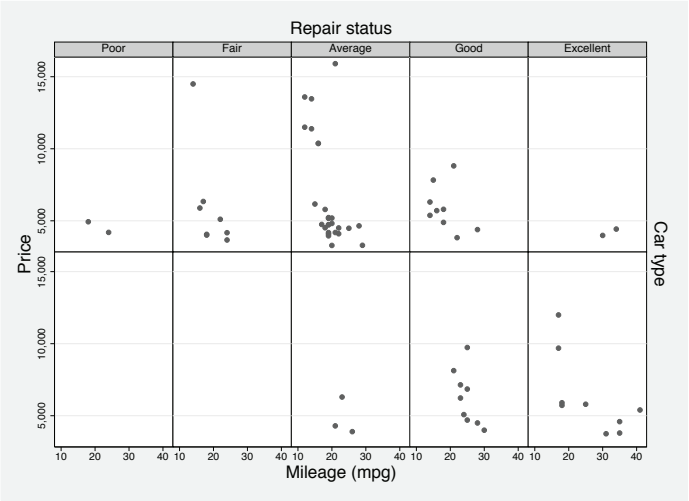

Figure 2. Comparing the relationship between price and mileage across repair status and car type (with column titles)

The row titles can now be added using the Graph Editor ([G] **graph editor** and Mitchell [2008, chapter 2]). The Graph Editor can be started within the graph window by going to the `file` menu and clicking on `Start Graph Editor`. Within the Graph Editor, the window of immediate interest is the `Object Browser` on the right-hand side of the Graph Editor. If we expand `plotregion1` (see figure 3) by clicking on the `+`, we can see that the characteristics of the individual subgraphs are identified by adding a `[1]` for the first (top left) subgraph, `[2]` for the second subgraph (to the right of the first subgraph), etc.

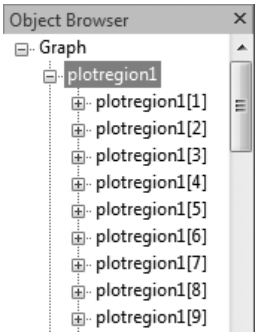

Figure 3. The object browser

In our example, we want to add a title to the right of the fifth and tenth subgraphs. A title to the right of a graph is an `r1title`, so within the `Object Browser`, we are looking for the elements `r1title[5]` for the top row title and `r1title[10]` for the

bottom row title. This is specific to this example, because we wanted to create a graph with five columns and two rows. If, for example, we wanted to make a graph with three rows and three columns, we would be looking for `r1title[3]`, `r1title[6]`, and `r1title[9]`, for the top, middle, and bottom row title, respectively.

When you double-click on `r1title[5]`, you will get the dialog box shown in figure 4. In this dialog box, we can type the title. The remaining options govern how this row title will look. If we use the `sj` scheme ([G] **schemes intro**) for our graph, then the options shown in figure 4 will make sure that the row titles will look the same as the column titles. If you use another scheme, you can find out how to set those options by double-clicking on one of the column titles and looking in the resulting dialog box at how the options are set. The title for the bottom row is created in the same way by typing the title and setting the options for `r1title[10]`. The resulting graph is shown in figure 5.

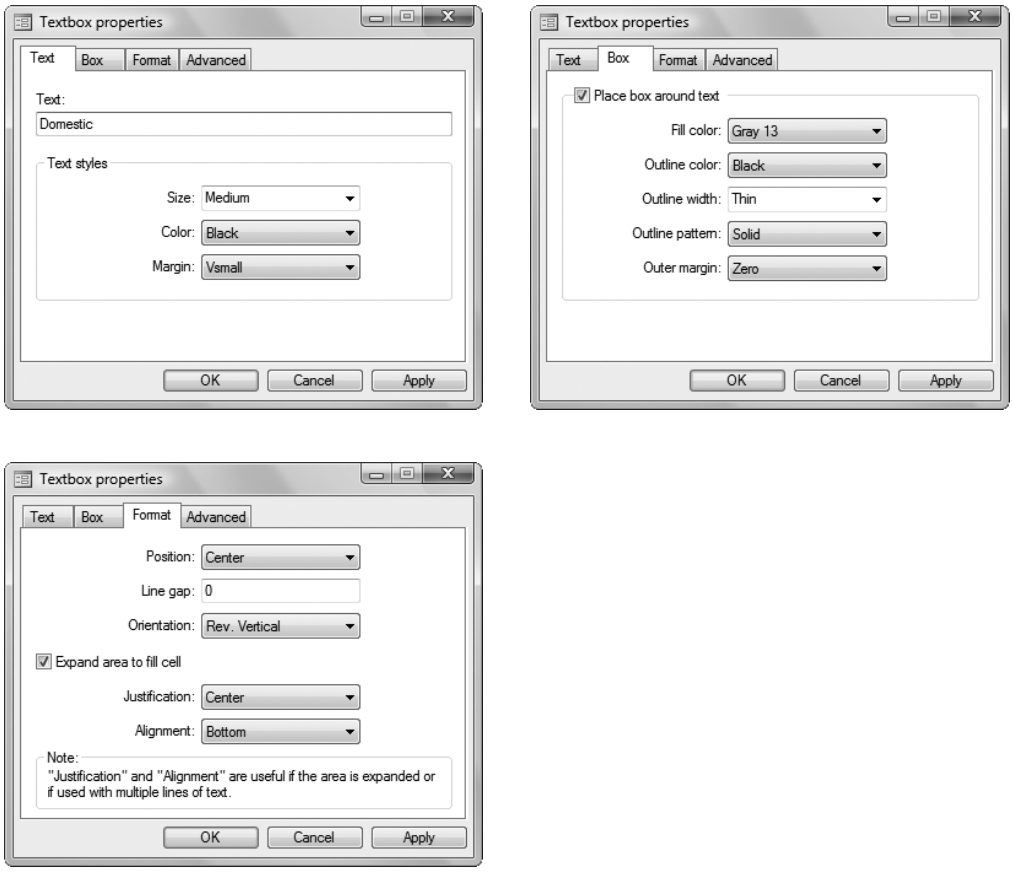

Figure 4. Properties specified in the Graph Editor for `r1title[5]`

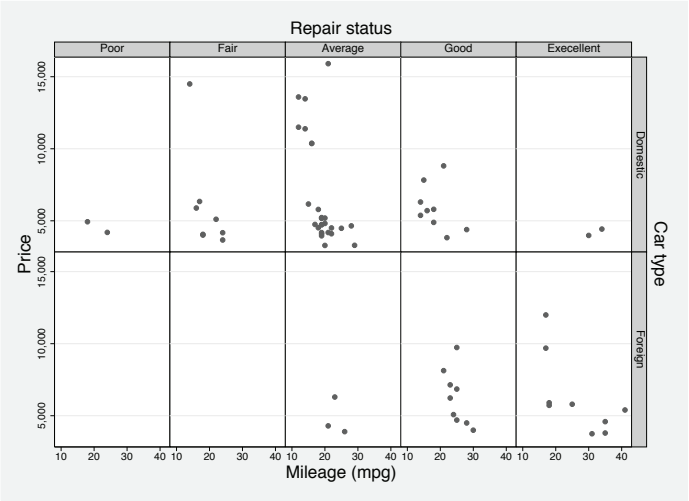

Figure 5. Comparing the relationship between price and mileage across repair status and car type (with column and row titles)

References

Becker, R. A., W. S. Cleveland, and M.-J. Shyu. 1996. The visual design and control of trellis display. *Journal of Computational and Graphical Statistics* 5: 123–155.

Cleveland, W. S. 1993. *Visualizing Data*. Summit, NJ: Hobart.

Cox, N. J. 2007. Stata tip 52: Generating composite categorical variables. *Stata Journal* 7: 582–583.

Mitchell, M. N. 2008. *A Visual Guide to Stata Graphics*. 2nd ed. College Station, TX: Stata Press.

Sarkar, D. 2008. *Lattice: Multivariate Data Visualization with R*. New York: Springer.