



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.

No endorsement of AgEcon Search or its fundraising activities by the author(s) of the following work or their employer(s) is intended or implied.

Stata tip 131: Custom legends for graphs that use translucency

Tim P. Morris
MRC Clinical Trials Unit
University College London
London, UK
tim.morris@ucl.ac.uk

As of version 15, Stata graphs permit translucent elements, which are invoked by “%#” following a *colorstyle* (where # is a percentage of opacity). Many `twoway` commands will include a legend. This gives control over the labels but not over the symbols.

I will now demonstrate a trick to produce legend symbols “by hand” for plot types `scatter` and `line`, and I will demonstrate why the arrival of translucency has prompted the need for this. This tip is not well showcased by the `sj` graph scheme; hence, I advise readers to try running the examples using a scheme including colors.

First, we will simulate a dataset with two groups, each containing 1,000 observations per group on `y` and `x` (`drawnorm` sets the dataset size from empty).

```
. clear
. matrix means = (4,1)
. matrix corr = (1, .5 .5, 1)
. drawnorm y1 x1, n(1000) seed(1)
. drawnorm y2 x2, corr(corr) means(means)
```

We begin with a scatterplot:

```
. twoway
> (scatter y1 x1, mcolor(%30) mlcolor(white%1))
> (scatter y2 x2, mcolor(%30) mlcolor(white%1))
> ,
> legend(order(1 "Group 1 (default symbol)" 2 "Group 2 (default symbol)"))
```

The scatterplot above includes translucent symbols with 30% opacity. The translucency inherited by the symbols in the legend is not necessarily desirable, because it matches the palest possible shade seen within the plot, which may make it harder to match the legend to the plotted data. To produce a legend symbol of our own choosing, we can add what I term a “ghost” plot for each symbol we wish to control. This is achieved as follows:

```
. twoway (scatteri . . , mszie(medlarge))
> (scatteri . . , mszie(medlarge))
> (scatter y1 x1, pstyle(p1) mcolor(%30) mlcolor(white%1))
> (scatter y2 x2, pstyle(p2) mcolor(%30) mlcolor(white%1))
> , legend(order(1 " Group 1 (custom symbol)" 2 " Group 2 (custom symbol)"))
```

Figure 1 shows the plot with the first and second versions of the legend, placed in one graph here to emphasize the difference: the top row uses the default translucent symbols, and the bottom row uses the opaque custom symbols produced by the ghost plots.

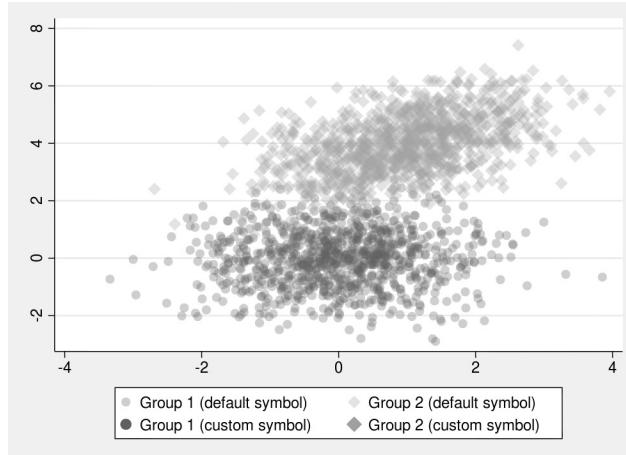


Figure 1. Scatterplot of simulated data with default (upper row) and custom (lower row) legend symbols

The ghost plots here were invoked using `scatteri`, and the points . . ensured nothing was plotted—but the symbol attributes nonetheless appeared in the legend. The option `legend(order(1 "..." 2 "..."))` now refers to our ghost plots (because they appear first and second in the `twoway` command). Our original plots are now in positions 3 and 4, meaning that the symbols they use would have changed. This is remedied by the option `pstyle(p1)` for plot 3 and `pstyle(p2)` for plot 4.

To demonstrate this method using a `line` plot type, we first (twice) `reshape` the simulated data:

```
. generate int id = _n
. reshape long y x, i(id) j(group)
. rename y y1
. rename x y0
. reshape long y, i(id group) j(time)
```

The following `twoway` command then produces a legend with user-manipulated symbols:

```
. twoway
> (function . , lcolor(gs10) lwidth(medthick))
> (function . , lcolor(gs6) lwidth(medthick))
> (line y time if group==1, pstyle(p1line) lcolor(gs10%5))
> (line y time if group==2, pstyle(p2line) lcolor(gs6%5))
> , legend(order(1 "Group 1 (custom symbol)" 2 "Group 2 (custom symbol)") cols(1))
```

Most of this works in the same way as with `scatter`. The ghost plot type is `function`, and again we use it to plot nothing (.¹ denotes missing). The *pstyle*s used by plots 3 and 4 are now `p1line` and `p2line`, respectively.

Using this approach, any options required for the legend can be achieved by manipulating options within the ghost plots, and this manipulation does not have to affect the “living” plots.

The trick outlined in this tip has uses beyond translucency. In presentations, it is sometimes desirable to add elements gradually to build up a graph. For example, the legend that comes with `msymbol(p)` is almost never readable, even when the color of a swarm of points is. Another use is in presentations. One may begin with the graph region with no data but a legend, then introduce the data for group 1 and subsequently group 2, etc.; using this trick, the legend could be present from the beginning.