

## Working in SGPLOT: Understanding the General Logic of Attributes

James Blum, University of North Carolina Wilmington;  
Jonathan Duggins, North Carolina State University

### ABSTRACT

ODS Graphics use common style elements for distinct graphical entities—text, symbols, fills and lines—through attribute options. Most graphical entities produced, either by default or requested via a specific option, can have their styles modified with an ATTRS-type option. The syntax of such options is built to be consistent across instances of these graphical entities for most plotting statements that generate them. Knowing what graphical entities are generated by various statements and options, along with understanding how SAS categorizes graphical entities and their style elements, can help make style modification much easier to understand. Examples will focus on the commonalities present across a wide variety of plotting statements and options within them. Some non-standard style elements will be considered.

### INTRODUCTION

For those who are relatively new to SGPLOT, particularly when transitioning from using SAS/GRAPH procedures and associated statements, learning to set styles for graphical elements is an important component to getting the most out of the procedure's capabilities. Fortunately, the logical structure for these attributes and the options that control them is very clear and reliable throughout the set of plotting tools available. In this paper, we will work through examples using only the HBAR and SCATTER plotting statements; however, this review of attribute structure should translate well to any plotting tool you choose to use.

### GRAPHICAL ELEMENTS AND THEIR STYLE OPTIONS

Here, we use the term *graphical element* to refer to various components that a graph or plot is constructed from. This should not be confused with a *style element*, which are graphical styles defined in a template. The four graphical elements are: text, fills, lines and markers; and each has a set of attributes you can modify. Here is a listing of the attributes and information on possible settings for each of the graphical elements:

#### Text

Text attributes are:

- Family (Font): any TrueType font available in your session, specified as a literal.
- Size: expressed in various units: cm, mm, in, pct, pt, px.
- Color: expressed using any of the standard SAS color models or naming schemes.
- Style: normal or italic.
- Weight: normal or bold.

#### Fills

Fill attributes are:

- Color: expressed using any of the standard SAS color models or naming schemes.
- Transparency: a number (proportion) between 0 and 1, 0 is opaque.

## Lines

Line attributes are:

- Thickness: expressed in various units: cm, mm, in, pct, pt, px.
- Color: expressed using any of the standard SAS color models or naming schemes.
- Pattern: a number corresponding to a line pattern, see the SAS Documentation for a table of possible values.

## Markers

Marker attributes are:

- Size: expressed in various units: cm, mm, in, pct, pt, px.
- Color: expressed using any of the standard SAS color colors or naming schemes.
- Symbol: see the SAS Documentation for a table of names and corresponding symbols.

Again, any of these choices could be controlled by referencing a style element from a template, but for this discussion we will alter attributes manually.

## EXAMPLES FOR DEFAULT GRAPHICAL ELEMENTS IN HBAR

In order to effectively use attributes to modify styles, it is important to know the name given to the graphical element you are trying to modify. If the graphical element is created by a specified option, the attribute option is reliably that same keyword with the suffix **attrs** attached. This suffix is also used for default graphical elements, and their names are usually easy to deduce, but may not be obvious.

Our first examples are based on a simple bar chart drawn from the cars data that is shipped as part of the standard sashelp library. This chart, by default, contains every graphical element except markers.

Here is the unmodified code:

```
ods graphics / reset;  
proc sgplot data=sashelp.cars;  
    hbar origin / response=mpg_city stat=mean;  
run;  
quit;
```

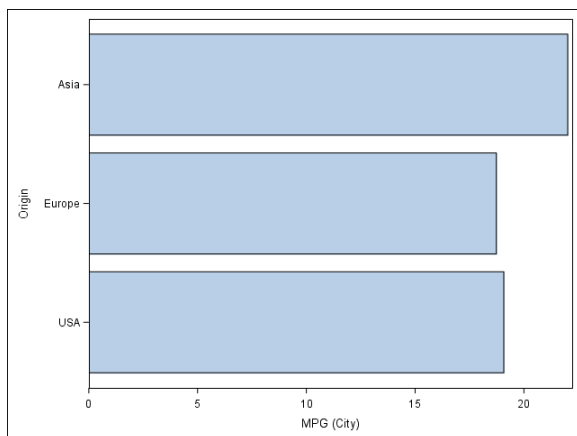


Figure 1: Sample Bar Graph

This graph contains text on each axis in the role of labels and values (familiar nomenclature for SAS/GRAPH) users, fills for the bars, and lines as bar outlines and axes. We will consider alterations to our code that allow us to restyle some of these.

The bar fills, being an internal part of the graph area generated by the HBAR statement, will have their attributes altered with an option in the HBAR statement. Given that each of those graphical elements are simply referred to as a fill, the appropriate option attaches the **attrs** suffix and becomes FILLATTRS. A modification of the bar color is given below:

```
ods graphics / reset;
proc sgplot data=sashelp.cars;
    hbar origin / response=mpg_city stat=mean fillattrs=(color=cxFFAA00);
run;
quit;
```

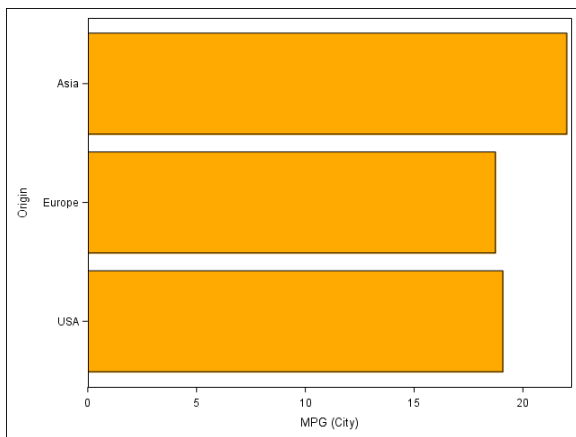


Figure 2: Bar Fill Attribute Modification

If you are a regular user of SAS/GRAPH, avoid the temptation to use aliases for these attributes, they are not legal in this context:

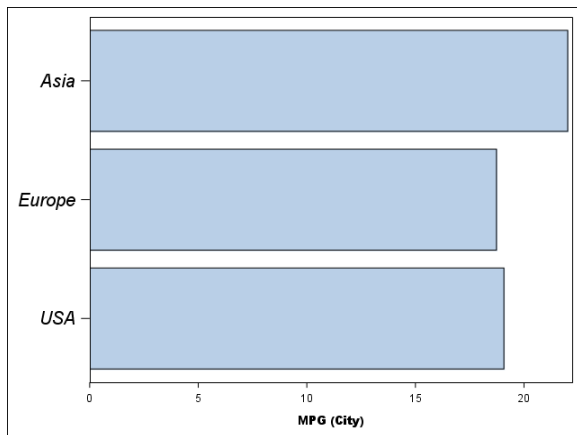
```
ods graphics / reset;
proc sgplot data=sashelp.cars;
    hbar origin / response=mpg_city stat=mean fillattrs=(c=cxFFAA00);
run;
quit;
```

```
Log - (Untitled)
55 proc sgplot data=sashelp.cars;
56     hbar origin / response=mpg_city stat=mean
57         fillattrs=(c=cxFFAA00);
                    -
                    22
                    202
ERROR 22-322: Syntax error, expecting one of the following: COLOR,
TRANSPARENCY.
ERROR 202-322: The option or parameter is not recognized and will be
ignored.
58 run;
NOTE: The SAS System stopped processing this step because of errors.
NOTE: PROCEDURE SGPLOT used (Total process time):
      real time    0.12 seconds
      cpu time     0.01 seconds
59 quit;
```

Figure 3: No Aliases for Attribute Names

The only text on this graph is that which appears on the axes, so if those attributes are to be changed, it will have to be done as part of the appropriate axis statement. Just like the nomenclature in SAS/GRAPH, the text elements here are the label and the value at each major tick. So, following the attribute option logic, the options we will use are LABELATTRS and VALUEATTRS:

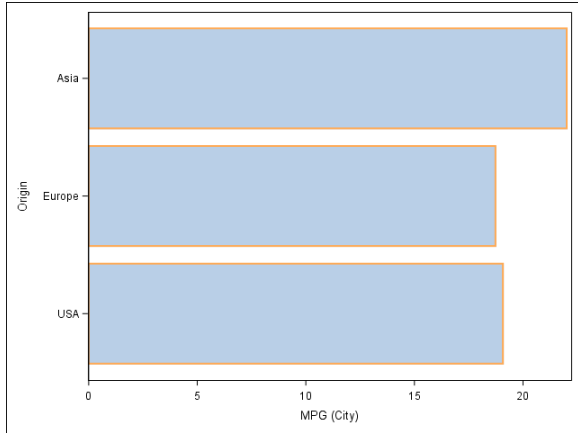
```
ods graphics / reset;
proc sgplot data=sashelp.cars;
  hbar origin / response=mpg_city stat=mean;
  yaxis display=(nolabel) valueattrs=(size=14pt style=italic);
  xaxis labelattrs=(family='Arial Black');
run;
quit;
```



**Figure 4: Modifying Text Attributes on the Axes**

The bar outlines are line graphical elements and are generated by the HBAR request, so the attribute modifications for those will be specified in an HBAR option. Since each of these is given the name outline, the correct option request is OUTLINEATTRS:

```
ods graphics / reset;
proc sgplot data=sashelp.cars;
  hbar origin / response=mpg_city stat=mean
  outlineattrs=(color=cxFFAA55 thickness=2pt);
run;
quit;
```



**Figure 5: Modifying Attributes for Bar Outlines**

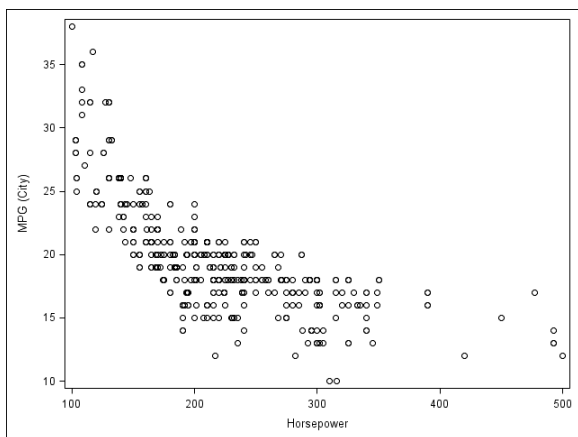
Hopefully, the basic pattern for working with these attributes is becoming clear: as long as you know what the graphical element type and name are, you should be able to deduce the correct option for setting its attributes.

## EXAMPLES FOR THE MARKER GRAPHICAL ELEMENT IN SCATTER

The one graphical element not present in the previous example is the marker—bar graphs generally do not produce any marker elements. This next example is based on a simple scatter plot drawn from the cars data that is shipped as part of the standard SASHELP library. This graph, by default, contains every graphical element except the fill; however, we will include it later.

Here is the unmodified code:

```
ods graphics / reset;
proc sgplot data=sashelp.cars;
  scatter x=horsepower y=mpg_city;
  where type ne 'Hybrid';
run;
quit;
```



**Figure 6: Sample Scatterplot**

Given that the name of the graphical element for each plotted point is marker, and they are generated by the SCATTER request and are internal to the data area of the graph, it follows that the MARKERATTRS option will be used in the SCATTER statement. One item that will be a bit different for regular users of SAS/GRAPH is that SYMBOL is no longer a statement, but is an attribute for the marker that controls its shape. Here is one example:

```
ods graphics / reset;
proc sgplot data=sashelp.cars;
  scatter x=horsepower y=mpg_city /
    markerattrs=(symbol=squarefilled color=cxAA6666 size=4pt);
  where type ne 'Hybrid';
run;
quit;
```

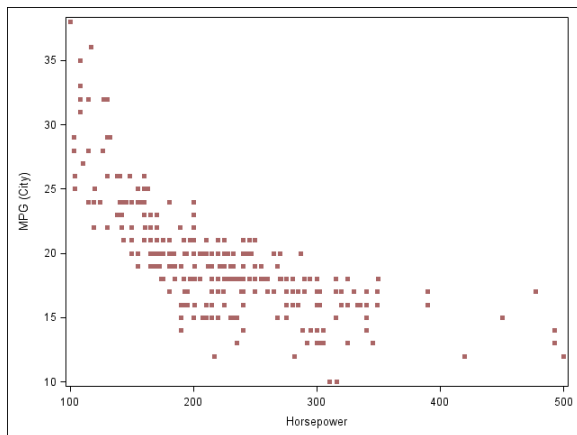


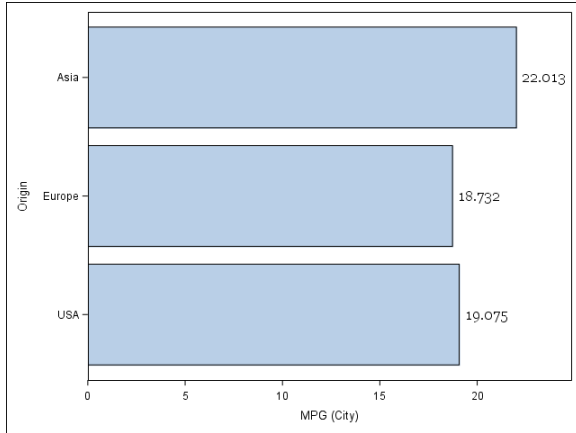
Figure 7: Attribute Modifications for Markers

Some options in the SCATTER request modify the role of the markers themselves, in those instances the attribute settings will need to be done through a different option. Such scenarios will be considered in the next section.

## SETTING ATTRIBUTES WHEN OPTIONS ADD GRAPHICAL ELEMENTS

Various options in plot requests or axis statements or elsewhere cause additional graphical elements to be created. Typically, the attributes for these are controlled by an option whose name combines the original option name with the **attrs** suffix. For example, in an HBAR request, the DATALABEL option places the summary statistic at the end of the bars, introducing a text element inside the graph area. Therefore, you can modify the associate attributes by also including DATALABELATTRS (yes, these option names can become rather long). Here is a sample:

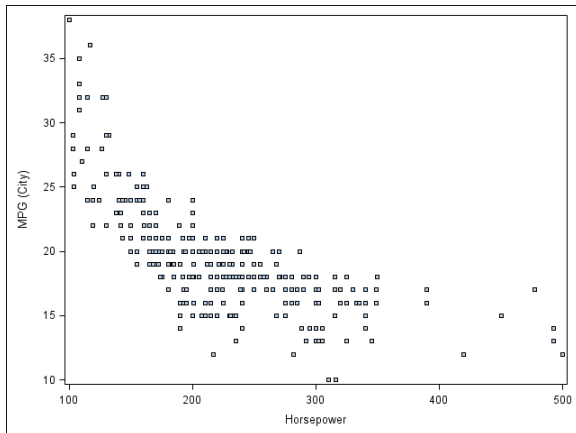
```
ods graphics / reset;
proc sgplot data=sashelp.cars;
  hbar origin / response=mpg_city stat=mean datalabel
    datalabelattrs=(family='Georgia' size=12pt);
run;
quit;
```



**Figure 8: Adding Data Labels and Setting Their Attributes**

In scatterplots, a similar situation occurs if you use `MARKERCHAR=` and pick a variable—the marker is replaced with text and the attributes are controlled by `MARKERCHARATTRS`. Another interesting case with markers occurs when the `FILLEDOUTLINEDMARKERS` option is utilized. Consider the following code and result:

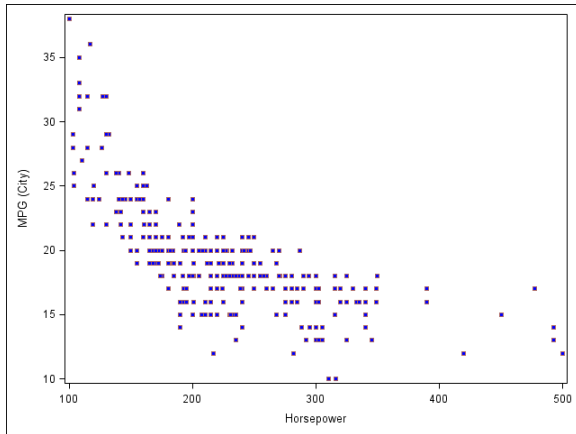
```
ods graphics / reset;
proc sgplot data=sashelp.cars;
  scatter x=horsepower y=mpg_city/filledoutlinedmarkers
    markerattrs=(symbol=squarefilled color=cxAA6666 size=4pt);
  where type ne 'Hybrid';
run;
quit;
```



**Figure 9: Filled Outlined Markers' Attributes I**

The symbol attribute chosen is utilized (in fact, you must use a filled symbol in order for `FILLEDOUTLINEDMARKERS` to have any effect), but the color is ignored. This unexpected result is explained by the fact that the `FILLEDOUTLINEDMARKERS` option splits the marker into its fill and its outline. With this in mind, there are now two new options for setting attributes: `MARKERFILLATTRS` and `MARKEROUTLINEATTRS`. Here is code where attributes for each of these are set:

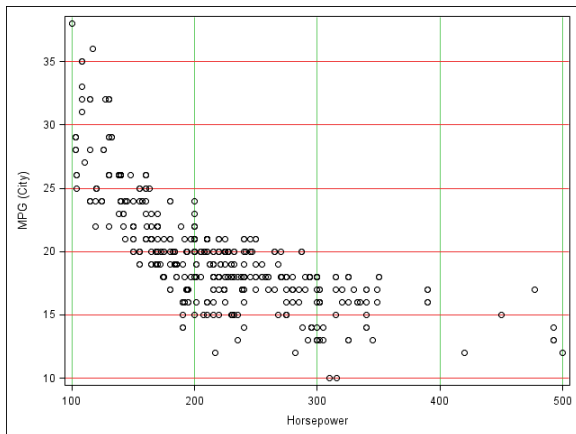
```
ods graphics / reset;
proc sgplot data=sashelp.cars;
  scatter x=horsepower y=mpg_city/filledoutlinedmarkers
    markerattrs=(symbol=squarefilled size=4pt)
    markerfillattrs=(color=blue)
    markeroutlineattrs=(color=cxAA6666);
  where type ne 'Hybrid';
run;
quit;
```



**Figure 10: Filled Outlined Markers Attributes II**

Axis statements can generate new elements as well; the GRID option puts reference lines at all major ticks, so these line element attributes are controlled by GRIDATTRS:

```
ods graphics / reset imagename='Figure 11';
proc sgplot data=sashelp.cars;
  scatter x=horsepower y=mpg_city;
  xaxis grid gridattrs=(color=cx66CC66);
  yaxis grid gridattrs=(color=cxEE3333);
  where type ne 'Hybrid';
run;
quit;
```



**Figure 11: Grid Line Attributes**



Again, for any option you might choose in any particular SGLOT statement, if it creates a graphical element it should be relatively straightforward to figure out how you can set the attributes for it.

## THE STYLEATTRS STATEMENT

Consider the following modified versions of our original graphs, each of which now includes a group variable:

```
ods graphics / reset;
proc sgplot data=sashelp.cars;
  hbar origin / response=mpg_city stat=mean group=type groupdisplay=cluster;
  where type not in ('Truck','Hybrid');
run;
quit;
```

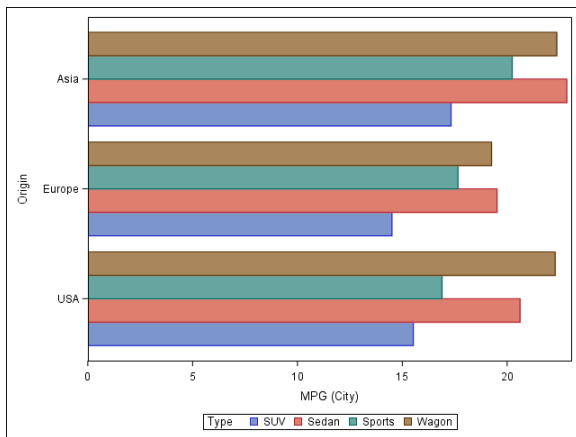


Figure 12: Grouped Bar Chart

```
ods graphics / reset;
proc sgplot data=sashelp.cars;
  scatter x=horsepower y=mpg_city / group=origin;
  where type ne 'Hybrid';
run;
quit;
```

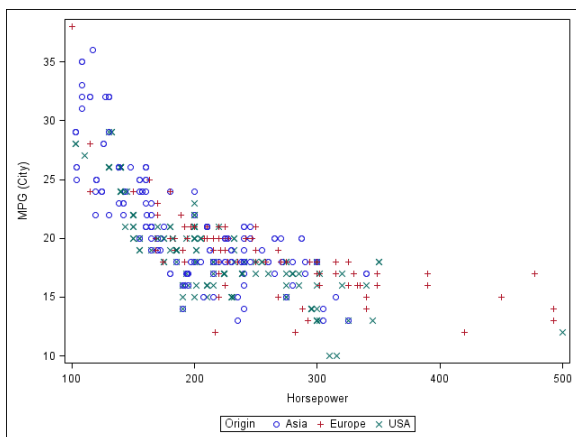
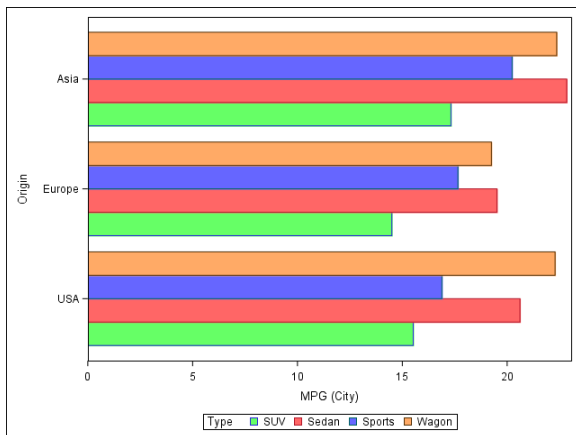


Figure 13: Grouped Scatterplot

For the bar chart, if we wanted to change the fill color, our FILLATTRS option is not going to be effective since we only get to choose one color. In the scatterplot it is a bit better, we could reset the color or the symbol and have distinct groups, but we cannot set both. To alter certain sets of attributes that are cycled through in grouped or overlaid graphs, SGPLOT includes the STYLEATTRS statement. The DATACOLORS= option allows you to pick a color list for cycling through fill elements—consider:

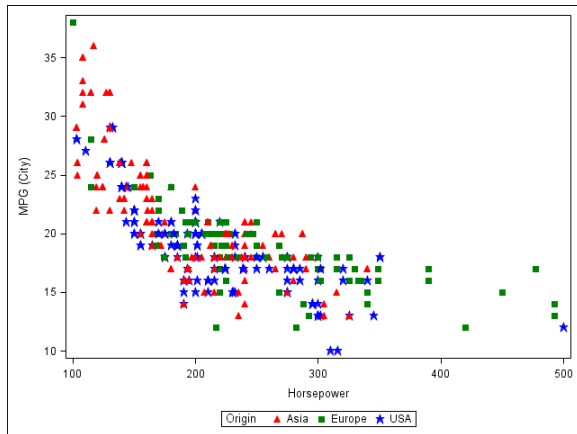
```
ods graphics / reset;
proc sgplot data=sashelp.cars;
  styleattrs datacolors=(cx66FF66 cxFF6666 cx6666FF cxFFAA66);
  hbar origin / response=mpg_city stat=mean group=type groupdisplay=cluster;
  where type not in ('Truck', 'Hybrid');
run;
quit;
```



**Figure 14: DATACOLORS Applied in the STYLEATTRS Statement**

The DATACONTRASTCOLORS= option allows you to pick a color list for cycling through marker (and line) colors and DATASYMBOLS= picks a list for symbol attributes—consider:

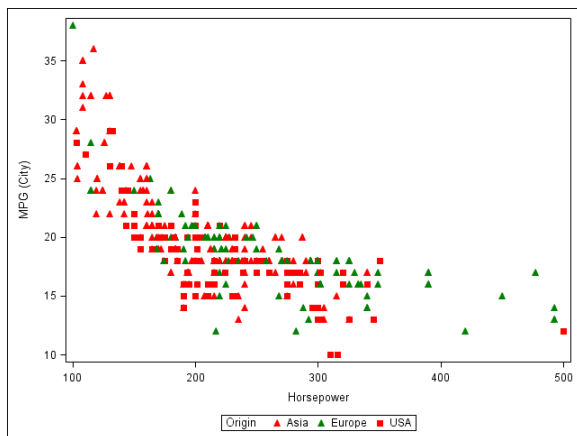
```
ods graphics / reset;
proc sgplot data=sashelp.cars;
  styleattrs datasymbols=(trianglefilled squarefilled starfilled)
  datacontrastcolors=(red green blue);
  scatter x=horsepower y=mpg_city / group=origin;
  where type ne 'Hybrid';
run;
quit;
```



**Figure 15: DATACONTRASTCOLORS and DATASYMBOLS**

DATALINEPATTERNS and DATAFILLPATTERNS (maintenance release 5 and up) are also available. A final note on the colors and symbol lists for markers. By default, SAS simultaneously cycles through both lists for each group it intends to differentiate; however, it is possible to modify this behavior with the ATTRPRIORITY= option in the ODS GRAPHICS statement. At this point, it is only possible to set this to COLOR (or the default of NONE), which causes SAS to cycle through the complete color list before changing symbols, consider:

```
ods graphics / reset attrpriority=color;
proc sgplot data=sashelp.cars;
  styleattrs datasymbols=(trianglefilled squarefilled)
  datacontrastcolors=(red green);
  scatter x=horsepower y=mpg_city/
  group=origin;
  where type ne 'Hybrid';
run;
quit;
```



**Figure 16: Setting ATTRPRIORITY**

By default, the two color and two symbol lists would not be sufficient for three group levels, but with ATTRPRIORITY set to color, there are up to four distinct markers available from these two lists.

## CONCLUSION

With an understanding of what the type and names of various graphical elements are, the attributes you can set for them, and which options generate them; it should be relatively straightforward to determine the right option(s) and where to place them to get the styles you desire. Although we will not claim that attribute options will always be constructed exactly the same way, our experience is that, no matter what type of graph, following the recommendations outlined here should lead you to the correct method to intervene and set attributes correctly.

## RECOMMENDED READING

- *Statistical Graphics Procedures by Example: Effective Graphs Using SAS®*, Sanjay Matange and Dan Heath

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

James E. Blum  
University of North Carolina Wilmington  
blumj@uncw.edu  
<http://people.uncw.edu/blumj>

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.