

Working in SGPLOT: Understanding the General Logic of Attributes

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

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 or control 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 focus on the commonalities present across a wide variety of plotting statements and options within them. Some non-standard style elements are also considered.

Introduction

For those who are relatively new to the SG graphing procedures (SGPLOT, SGPANEL, SGSCATTER), 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 capabilities of the procedure. Fortunately, the logical structure for these attributes and the options that control them is very clear and reliably consistent throughout the set of plotting statements available. In this paper, examples only involve the HBAR and SCATTER plotting statements; however, this review of attribute structure translates well to any plotting statement chosen in the SGPLOT Procedure.

Graphical Elements and Their Style Options

Here, the term graphical element is used to refer to various components SAS uses to construct a graph or plot. This is distinct from a style element, which is a graphical style defined in a template. The four graphical elements are: text, fills, lines, and markers; and each has a set of attributes available for modification. Below is a listing of the attributes, and details on possible settings, available for each of the graphical elements.

Text

Text attributes are:

- *Family (Font)*: any TrueType font available in the SAS 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 models or naming schemes.
- *Symbol*: see the SAS Documentation for a table of names and corresponding symbols.

Any of these choices are also controllable by referencing a style element from a template, but for this discussion attributes are altered manually.

Examples for Default Graphical Elements in HBAR

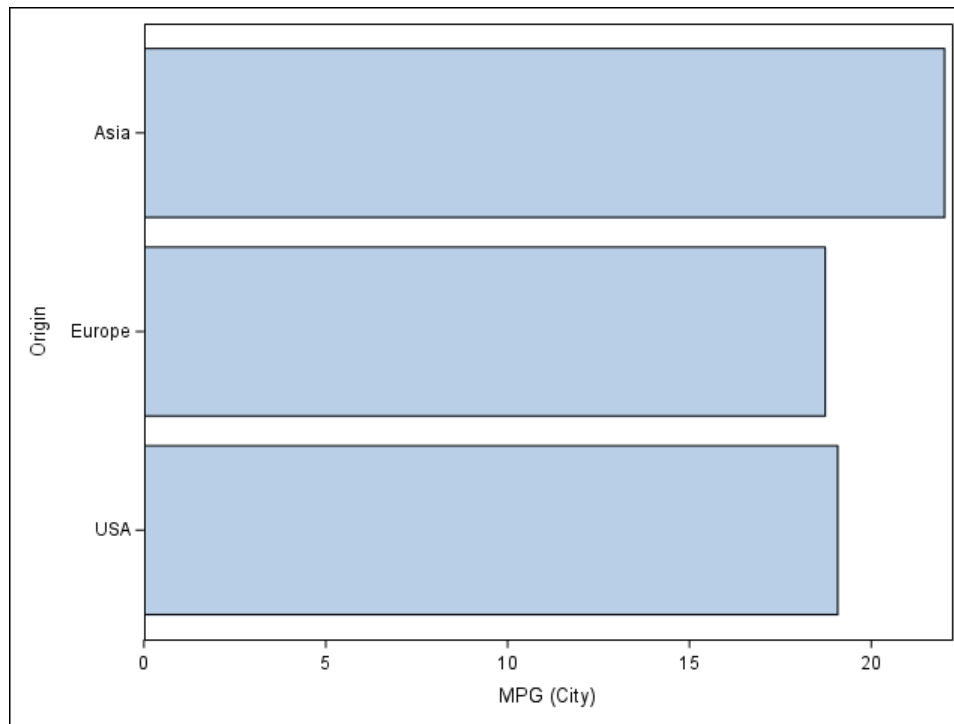
In order to effectively use attributes to modify styles it is important to know the name SAS gives to the graphical element being modified. If the graphical element is created by a specified option, the attribute option is typically that same keyword with the suffix ATTRS attached. This suffix is also used for default graphical elements – while those names are often easy to deduce, there are times where they are not obvious.

The initial examples are based on a simple bar chart drawn from the Cars data, a data set that is shipped as part of the standard SASHELP library. This chart, by default, contains every graphical element except markers. Program 1 shows an SGPLOT step to generate the sample bar graph shown in Output 1.

Program 1: A Sample Bar Graph

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

Output 1: A Sample Bar Chart



Output 1 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. Alterations to the code that allow for restyling of some of these elements are considered below.

The bar fills, being an internal part of the graph area generated by the HBAR statement, 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 is built by attaching the ATTRS suffix and becomes FILLATTRS. A modification of the bar color is given in Program 2 and Output 2 shows the resulting orange fill for the bars.

Program 2: A Filled Bar Chart

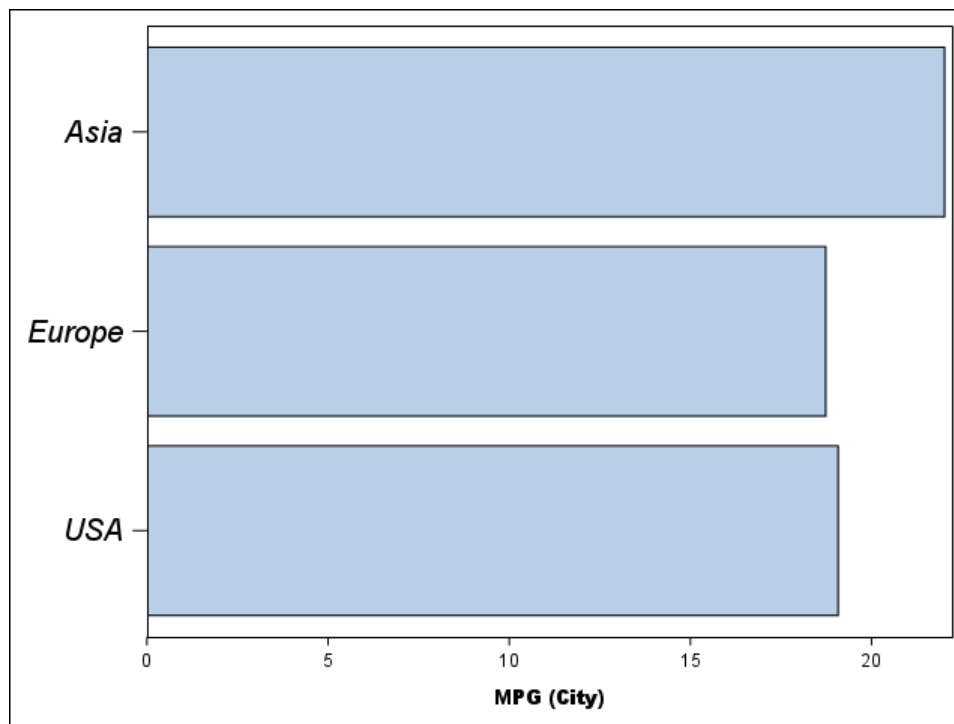
```
ods graphics / reset imagename="Bar with Fill";
proc sgplot data=sashelp.cars;
  hbar origin / response=mpg_city stat=mean fillattrs=(color=cxFFAA00);
run;
quit;
```


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 available are LABELATTRS and VALUEATTRS. Program 4 demonstrates some modifications of those text elements. Output 4 includes the adjusted text elements: the vertical axis no longer has a label and has a larger, italic font while the horizontal axis now uses the Arial Black font.

Program 4: Modifying the Axis Text Elements

```
ods graphics / reset imagename="Modify Axis Text";
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;
```

Output 4: Modifying the Axis Text Elements

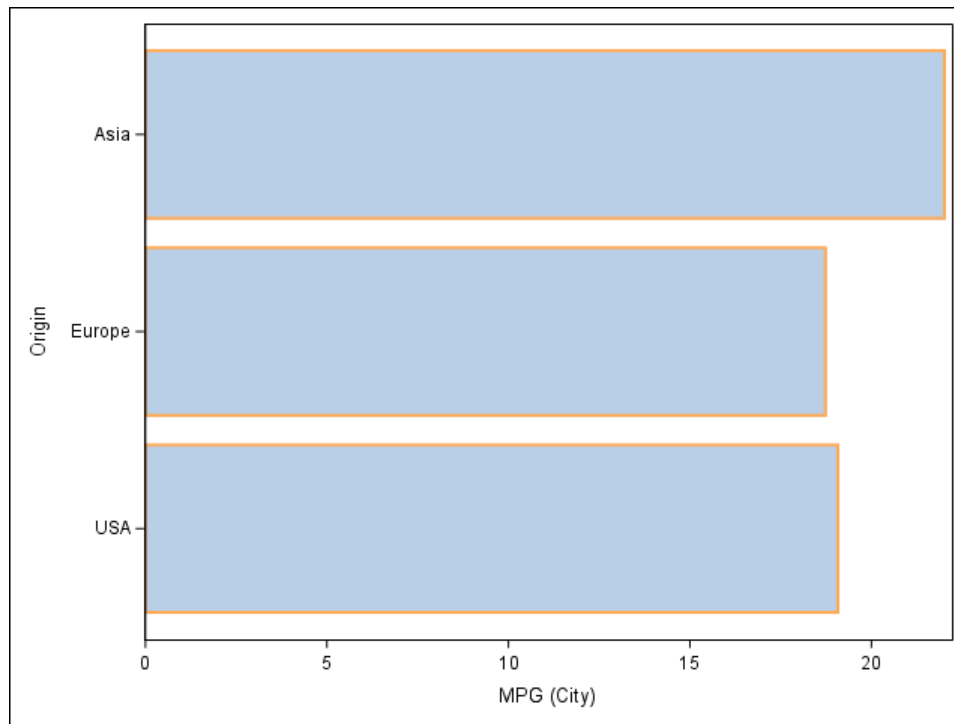


The HBAR statement not only generates the bars, it also generates the bar outlines; thus, their attribute modifications are specified via an option in the HBAR statement. These are line graphical elements and, since SAS refers to these as outlines, the correct option request is OUTLINEATTRS. As Program 5 shows, this affects all the outlines drawn by this specific HBAR statement. Output 5 shows the resulting thicker, orange outlines on each bar.

Program 5: Modifying Attributes for Bar Outlines

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

Output 5: Modifying Attributes for Bar Outlines



At this point, the basic pattern for working with these attributes becomes clear: as long as the graphical element type and its name are known, the correct option for setting its attributes can be readily deduced.

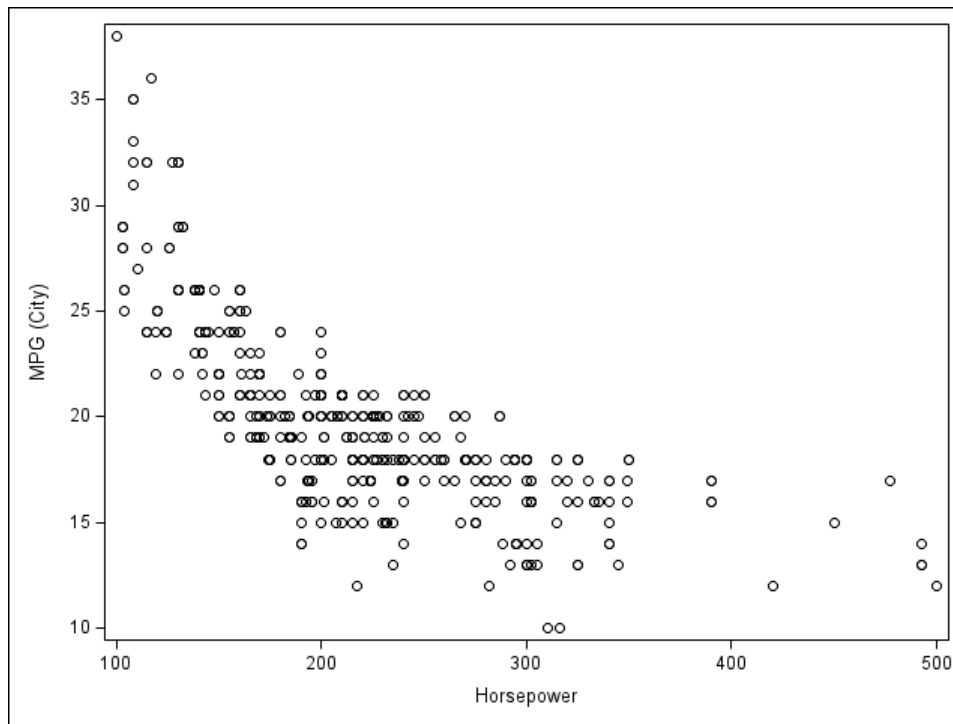
Examples for the Marker Graphical Element in Scatter

The one graphical element not present in the bar chart examples is the marker – bar graphs generally do not produce any marker elements. Program 6 is based on a simple scatter plot drawn again from the Cars data. By default, this graph – shown in Output 6 – contains every graphical element except the fill; however, fill attributes are included in the next section.

Program 6: A Sample Scatterplot

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

Output 6: A Sample Scatterplot



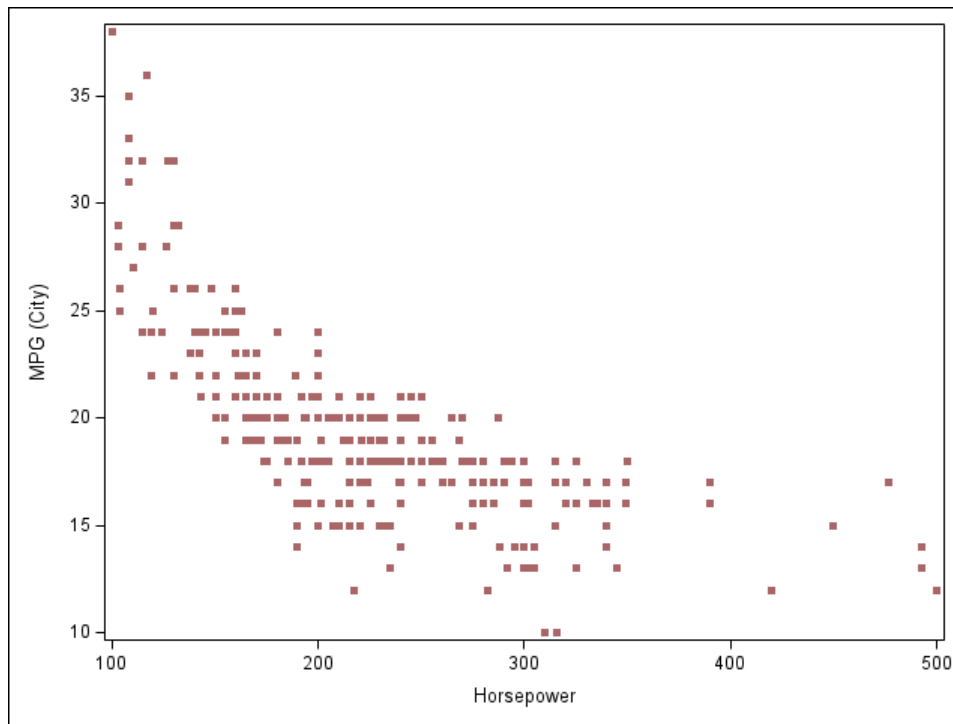
The logic of attributes in SGPLOT dictates that the keyword to adjust the markers is `MARKERATTRS`. Further, it follows that the `MARKERATTRS` option should appear in the `SCATTER` statement itself since it generates the markers and they appear in the data area of the graph (rather than outside like the axes which are modified in their own statements). One item that is a bit different for regular users of SAS/GRAPH is that `SYMBOL` is no longer a statement, but is an attribute that controls the shape of the marker. Program 7 demonstrates adjusting several marker attributes simultaneously. Output 7 shows the scatter plot now contains 4 point, light red, filled squares as the markers.

Program 7: Attribute Modifications for Markers

```
ods graphics / reset imagename="Adjust Markers";
proc sgplot data=sashelp.cars;
  scatter x=horsepower y=mpg_city / markerattrs=(symbol=squarefilled
                                                color=cxAA6666
                                                size=4pt);

  where type ne 'Hybrid';
run;
quit;
```

Output 7: Attribute Modifications for Markers



Some options in the SCATTER statement can modify the role of the markers themselves. In instances where modifications to the plotting statement produce new graphical elements or alter the way SAS uses the graphical elements, the attribute settings need to be done through a different option. Such scenarios are considered in the next section.

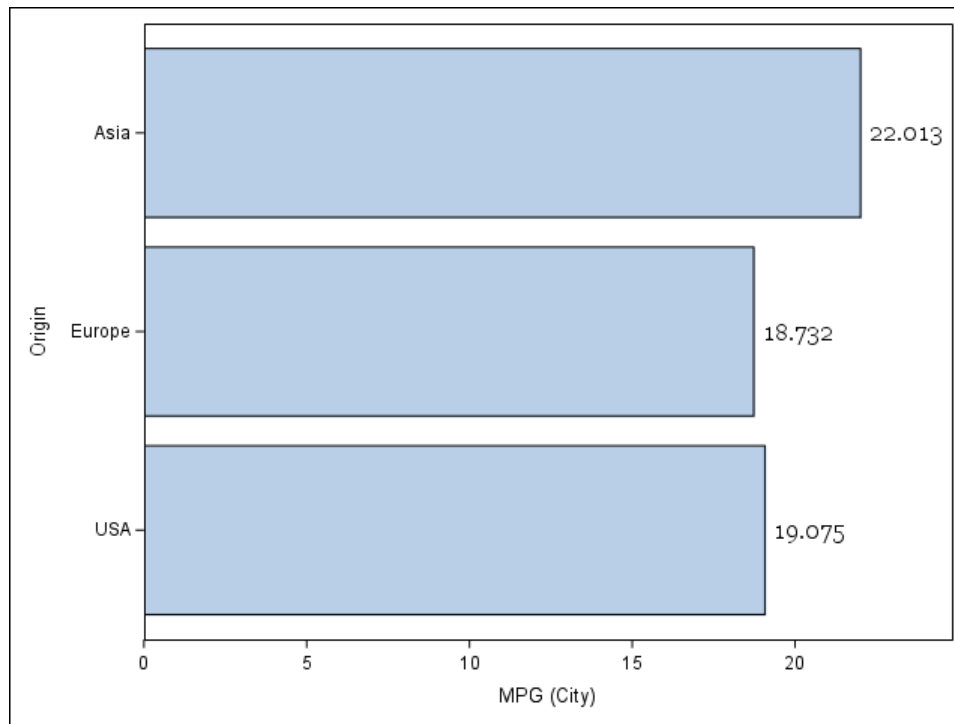
Setting Attributes When Options Add Graphical Elements

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 can place the summary statistic at the end of the bars, introducing a text element inside the graph area. Therefore, you can modify the associated attributes by also including DATALABELATTRS. Program 8 demonstrates this particular customization by adjusting the font family and font size of these new text elements.

Program 8: Adding Data Labels and Setting Their Attributes

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


Output 8: Adding Data Labels and Setting Their Attributes



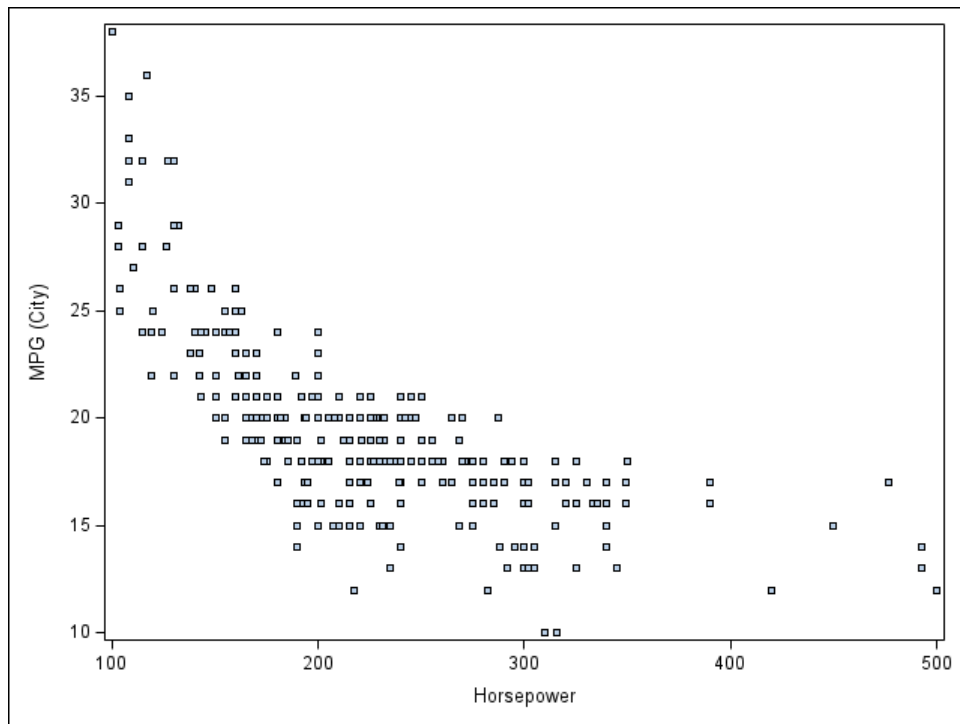
In scatterplots, a similar situation occurs if you use `MARKERCHAR=` and pick a variable – this replaces the marker with text and the attributes are controlled by `MARKERCHARATTRS`. Another interesting case with markers occurs when the `FILLEDOUTLINEDMARKERS` option is utilized. These option names can become rather long so it is not uncommon to see strategies that take advantage of SAS being a case-insensitive language. For example, camel case (`filledOutlineMarkers`) or Pascal Case (`FilledOutlineMarkers`) both increase readability. Program 9 applies the `FILLEDOUTLINEDMARKERS` option and attempts to modify several of the marker attributes.

Program 9: Filled Outlined Markers' Attributes I

```
ods graphics / reset imagename="Filled Outlined";
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;
```

Output 9: Filled Outlined Markers' Attributes I



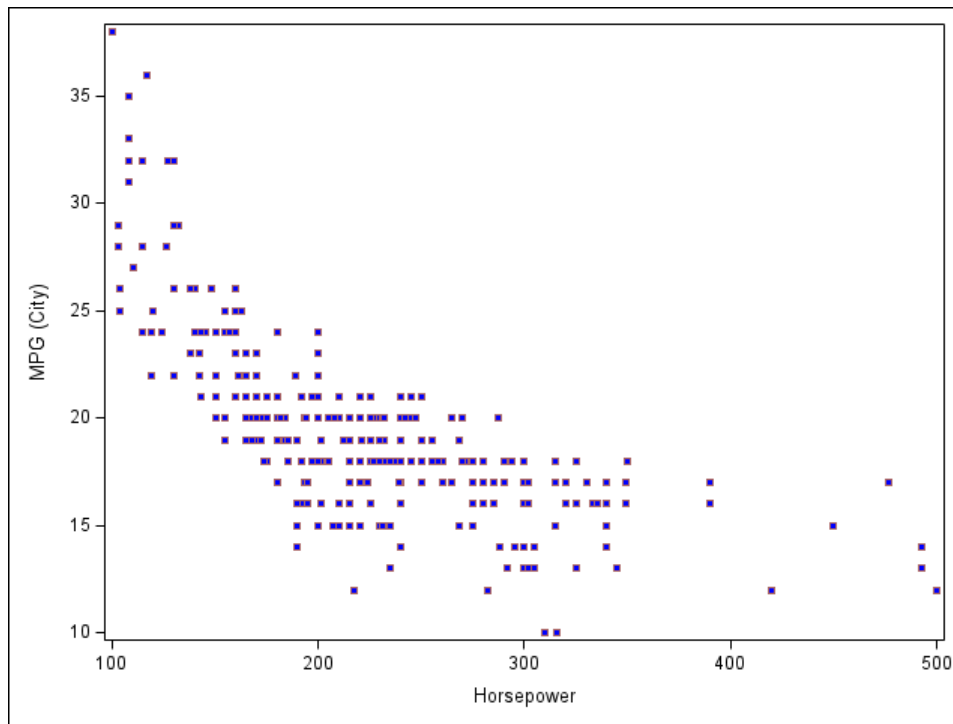
Output 9 shows that SAS applies the chosen symbol attribute (in fact, a filled symbol must be used in order for FILLEDOUTLINEDMARKERS to have any effect), but the color is ignored. This result may be initially unexpected but is explained by the fact that the FILLEDOUTLINEDMARKERS option splits the marker into two separate graphical elements: its fill and its outline. With this in mind, when using FILLEDOUTLINEDMARKERS there are now two other options for setting attributes: MARKERFILLATTRS and MARKEROUTLINEATTRS. Program 10 uses these new options in addition to the MARKERATTRS option to control the full set of attributes.

Program 10: Filled Outlined Markers' Attributes II

```
ods graphics / reset imagename="Filled Outlined 2";
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;
```

Output 10: Filled Outlined Markers' Attributes II



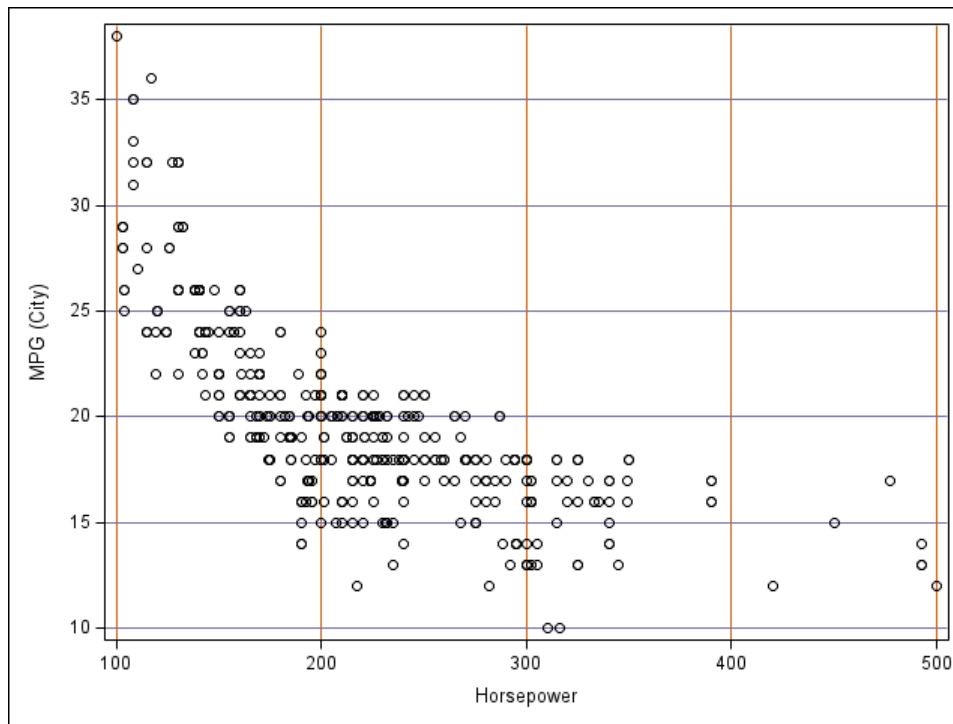
Output 10 shows the desired result: 4 point squares with a light red border and a blue fill. The `MARKERATTRS` option is necessary to select the type of marker – `SQUAREFILLED` – and set the size. Given that the `FILLEDOUTLINEDMARKERS` option is in effect, the `MARKERFILLATTRS` and `MARKEROUTLINEDATTRS` are then used to separately control the fill and outline attributes, respectively. It is important to note that this level of customization is not required – it is available when requested by the user and allows for fine-grained control over graphs generated by `SGPLOT` (and `SGPANEL` and `SGSCATTER`).

In addition to options within a plotting statement, 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`. Program 11 generates a graph with gridlines and includes modifications to their color – orange for the vertical lines and purple for the horizontal lines.

Program 11: Grid Line Attributes

```
ods graphics / reset imagename="Grid Attributes";
proc sgplot data=sashelp.cars;
  scatter x=horsepower y=mpg_city;
  xaxis grid gridattrs=(color=cxD95F02);
  yaxis grid gridattrs=(color=cx7570B3);
  where type ne 'Hybrid';
run;
quit;
```

Output 11: Grid Line Attributes



Again, for any option chosen in any particular SGPLOT statement that creates a graphical element, it is typically straightforward to figure out how to set the attributes for the element. The general logic is as shown in the above programs: know the name of the element and append ATTRS to it.

The STYLEATTRS Statement

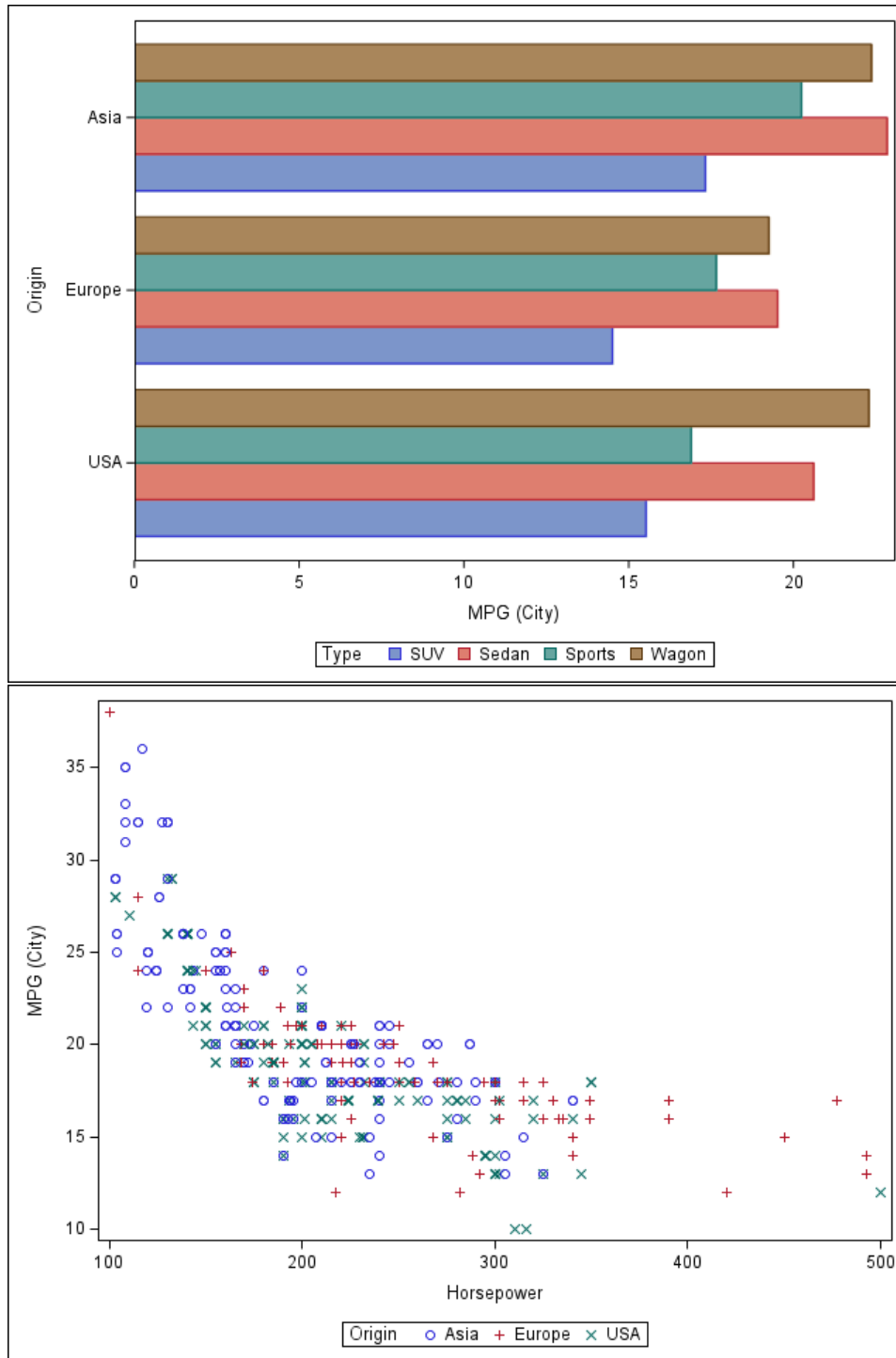
Program 12 includes modified versions of the original bar chart and scatterplot shown above. These new versions include a variable used to create groups within the plots.

Program 12: Grouped Bar Chart and Scatterplot

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

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

Output 12: Grouped Bar Chart and Scatterplot



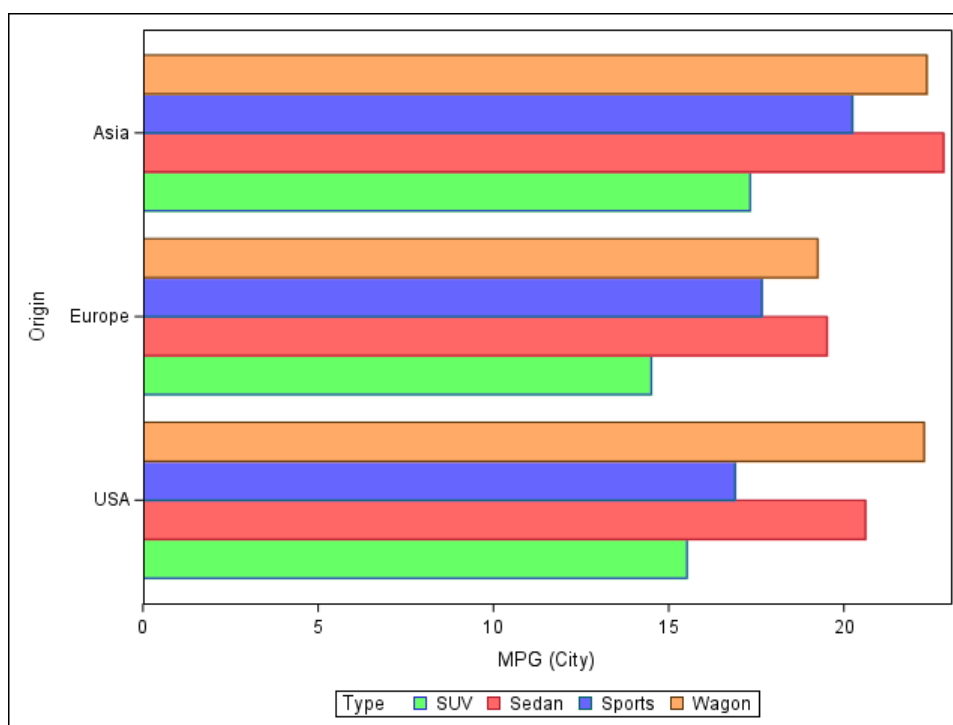
For the bar chart in Output 12, SAS automatically cycles through colors to distinguish the groups. The FILLATTRS option would not be effective for altering fill colors since it only allows for the choice of one color; thus, using it would fill in all bars with the same color. In the scatterplot, SAS cycles through both marker symbols and colors to distinguish the groups. As such, the situation is a bit better since the color or the symbol may be set to a single value and still have distinct groups based only on symbol or only on color, but setting both makes the groups indistinguishable. To alter certain sets of attributes that cycle in grouped or overlaid graphs, SGPlot includes the STYLEATTRS statement. Within this statement, the

DATACOLORS= option allows for the choice of a color list for cycling through fill elements—consider and Program 13 demonstrates its usage.

Program 13: DATACOLORS Applied in the STYLEATTRS Statement

```
ods graphics / reset imagename="Using DATACOLORS";
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;
```

Output 13: DATACOLORS Applied in the STYLEATTRS Statement

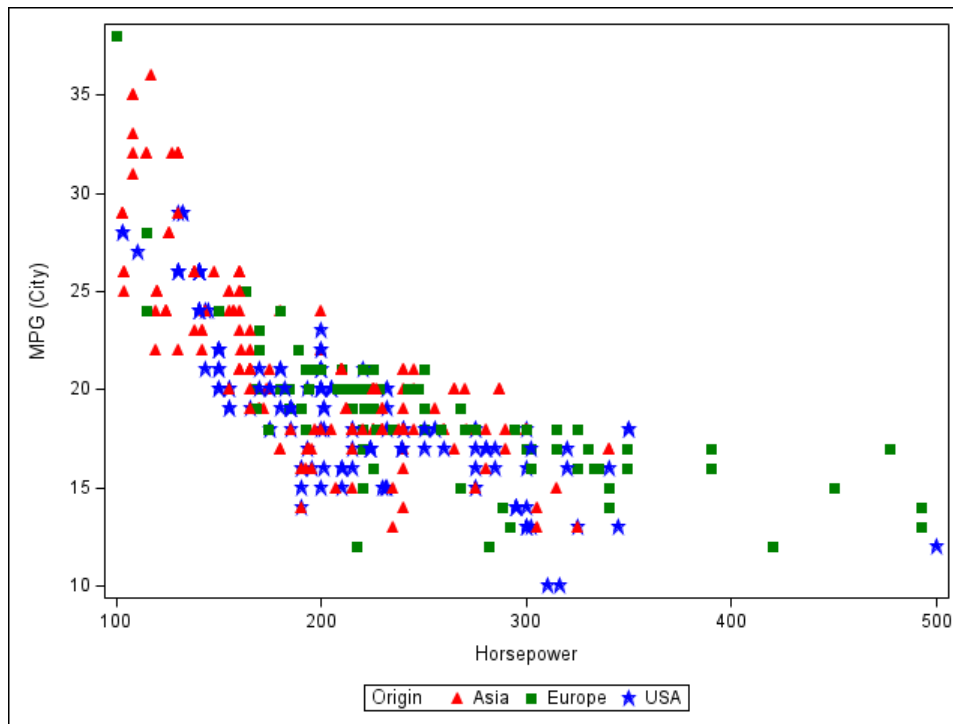


Similarly, the STYLEATTRS statement includes the DATACONTRASTCOLORS= option to cycle through marker (and line) colors and the DATASYMBOLS= option to cycle through marker symbols. Program 14 shows the use of both of these options to set the markers to red filled triangles, green filled squares, and blue filled stars.

Program 14: DATACONTRASTCOLORS and DATASYMBOLS

```
ods graphics / reset imagename="Symbols and Contrast Colors";
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;
```

Output 14: DATACONTRASTCOLORS and DATASYMBOLS



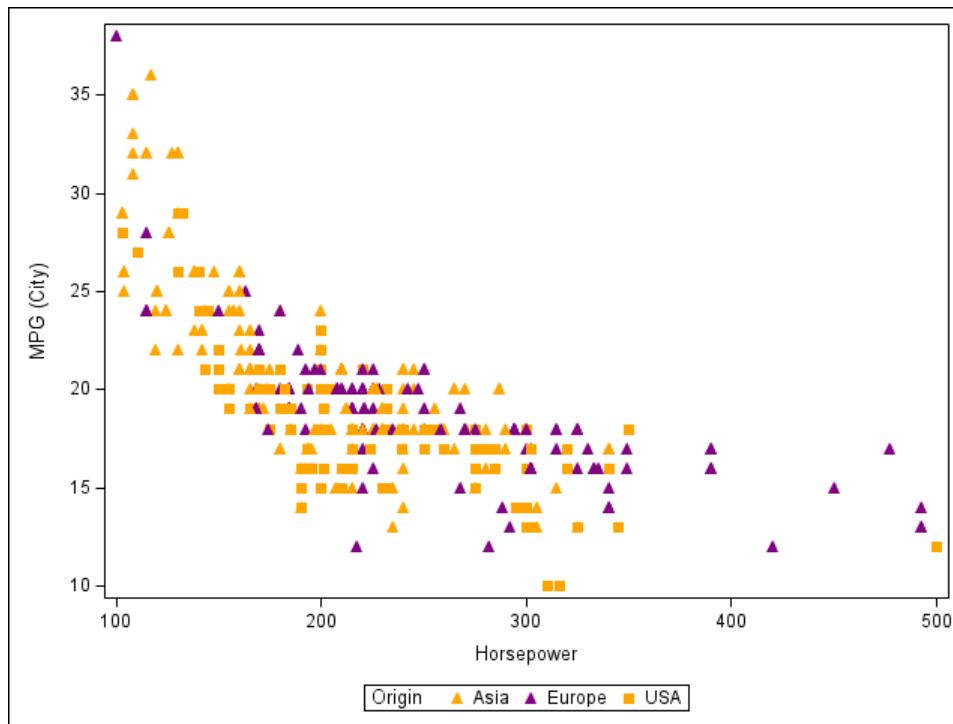
Starting in maintenance release 5, the `DATALINEPATTERNS` and `DATAFILLPATTERNS` are also available to cycle through line patterns and fill patterns, respectively. Note that, by default, SAS simultaneously cycles through the lists it uses to differentiate the groups – that’s why Program 14 produced red filled triangles and green filled squares, for example. However, it is possible to modify this behavior with the `ATTRPRIORITY=` option in the `ODS GRAPHICS` statement.

Currently the default value of `ATTRPRIORITY` is based on the setting inherited from the graph style element; thus, some graphs (like the bar chart) only cycle by color while others (like the scatterplot) cycle by multiple attributes. As of maintenance release 5, the only additional settings are `NONE` or `COLOR`. Setting to `NONE` overrides the default to cycle through all available attribute lists simultaneously. Alternatively, `COLOR` cycles through the list of colors first and then moves on the next element in the other attribute lists. This behavior is demonstrated in Program 15.

Program 15: Setting ATTRPRIORITY

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

Output 15: Setting ATTRPRIORITY



Output 15 shows orange triangles followed by purple triangles due to the color priority. Only after the color list has been exhausted does SAS cycle to the next marker symbol – a filled square – and then begin the color list again resulting in the orange square. By default, using two colors and two symbols in their lists is not sufficient to create three distinguishable groups, but with ATTRPRIORITY set to color, there are up to four distinct markers available from these two lists of length two.

Conclusion

By understanding the type and names of various graphical elements, which options generate them, and the attributes available to set for each type; it is typically relatively straightforward to determine the right option(s) and where to place them to get the styles desired. Although it is not claimed that attribute options are always constructed exactly the same way; in general, no matter what type of graph, following the recommendations outlined here leads 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:

Jonathan Duggins
North Carolina State University
jwduggin@ncsu.edu
<https://jonathanduggins.com/>