Paper 193-2019

Doing More with the SGPLOT Procedure

Joshua M. Horstman, Nested Loop Consulting

ABSTRACT

Once you've mastered the fundamentals of using the SGPLOT procedure to generate high-quality graphics, you'll certainly want to delve in to the extensive array of customizations available. This workshop will move beyond the basic techniques covered in the introductory workshop. We'll go through more complex examples such as combining multiple plots, modifying various plot attributes, customizing legends, and adding axis tables.

INTRODUCTION

The SGPLOT procedure is the workhorse for producing single-cell plots in modern SAS environments. It produces dozens of types of plots and allows for comprehensive customization of nearly every visual feature of those plots. The basic functionality and features of SGPLOT are covered in *Getting Started with the SGPLOT Procedure* (Horstman 2019). Readers unfamiliar with the procedure should begin with that paper.

This paper builds on that knowledge and digs deeper into the procedure. Topics include more complex ways to combine multiple plots, optional SGPLOT statements that allow for customization of graph features such as axes and legends, and advanced features such as axis tables and custom plot symbols. This paper is intended as a companion to a hands-on workshop taught in a live classroom setting, but it can be used on its own for independent study.

REVIEW OF THE SGPLOT PROCEDURE

THE SGPLOT PROCEDURE

The SGPLOT procedure is one of the SG procedures that comprise the ODS Statistical Graphics package. It is used to create single-cell plots of many different types. These include scatter plots, bar charts, box plots, bubble plots, line charts, heat maps, histograms, and many more.

Here is the basic syntax of the SGPLOT procedure:

```
proc sgplot data=<input-data-set> <options>;
  <one or more plot requests>
  <other optional statements>
run;
```

We start with the SGPLOT statement itself. This allows us to specify an input data set as well as numerous other procedure options.

Next, we include one or more plot request statements. There are dozens of plot request statements available. Some of these include SCATTER, SERIES, VBOX, VBAR, HIGHLOW, and BUBBLE. Several of these were discussed in detail in *Getting Started with the SGPLOT Procedure* (Horstman 2019).

Finally, there are several optional statements that control certain plot features such as XAXIS, YAXIS, REFLINE, INSET, and KEYLEGEND. We'll examine some of these and others as we progress through the exercises.

COMBINATION PLOTS

The SGPLOT procedure can be used to create combination plots by simply including multiple plot request statements. All plots will be overlaid atop one another in the same graph space and using the same axis

system. Plots are drawn in the order listed within the procedure call, with subsequent plot requests drawn over earlier plots.

```
proc sgplot data=<input-data-set> <options>;
  <plot-request-statement-1>
  <plot-request-statement-2>
  <plot-request-statement-3>
  ...
  <other optional statements>
run;
```

MULTIPLE AXIS SYSTEMS IN COMBINATION PLOTS

By default, all plot requests in a combination plot use the same axes. This can cause undesired results when the plots have different ranges of values. Using the secondary axis system can help in this situation. The secondary X axis is located along the top of the plot area, and the secondary Y axis is to the right-hand side. To specify that one or both secondary axes should be used for a plot, simply include the X2AXIS and/or Y2AXIS options on the corresponding plot request statement.

ODS DESTINATIONS

To create ODS graphs, a valid ODS destination must be open when the graph procedure is executed. For example, to invoke the SGPLOT procedure and direct the output to a PDF file, the ODS PDF statement is used to open and close the file as follows:

```
ods pdf file="c:\example.pdf";
  <SG procedure code goes here...>;
ods pdf close;
```

There are similar statements associated with other ODS destinations such as ODS HTML and ODS RTF. You can also have multiple destinations open simultaneously if you wish.

ABOUT THE EXERCISES

USING THE EXERCISES

These exercises were created as part of a hands-on workshop to be presented in a classroom setting. If you are using them on your own, it is recommended that you progress through them sequentially as they build on each other. To maximize your learning, try to complete each exercise on your own before looking at the solution provided. Also, keep in mind that there are often multiple ways to perform a task in SAS, so the code provided may not be the only correct solution.

EXAMPLE DATA SETS

Throughout this workshop, we will make use of several data sets from the SASHELP library. These data sets are included with SAS, which means these exercises should work anywhere you have SAS installed.

We will use the following data sets:

- SASHELP.CLASS – demographics on 19 students in a grade school classroom
- SASHELP.CARS – technical data about 428 car models

Take a few moments to familiarize yourself with these data sets before proceeding with the exercises.

EXERCISE 1: LINE CHART WITH DUAL AXES

THE VLINE STATEMENT

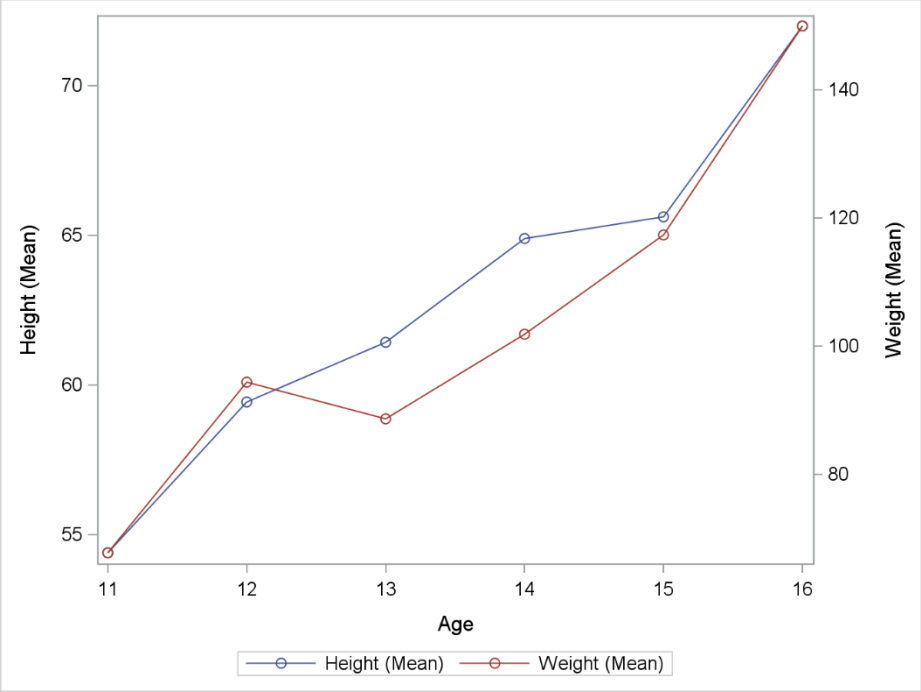
The VLINE statement is used to create a vertical line chart (which consists of horizontal lines). The endpoints of the line segments are statistics based on a categorical variable as opposed to raw data values.

```
proc sgplot data=<input-data-set> <options>;
  vline categorical-variable < / options>;
run;
```

The optional RESPONSE= and STAT= arguments can be used to specify a response variable and statistics, respectively, that will determine the coordinates of the endpoints of the line segments. The default statistic is the sum when a response variable is specified, or a frequency count otherwise. To add plot markers, use the MARKERS option.

EXERCISE

Using the SASHELP.CLASS data set, create a line plot of mean WEIGHT and mean HEIGHT by AGE with plot markers. You'll need two VLINE statements, each with a categorical variable and the RESPONSE=, STAT=, and MARKERS options. Add the Y2AXIS option to one VLINE statement to specify the secondary Y axis for that variable.



Exercise 1. Line Chart with Dual Axes

SOLUTION

```
proc sgplot data=sashelp.class;
  vline age / response=height stat=mean markers;
  vline age / response=weight stat=mean markers y2axis;
run;
```

EXERCISE 2: DUAL OFFSET BOX PLOT

THE VBOX STATEMENT

The VBOX statement is used to create a vertical box plot. The single unnamed required argument is the name of a numeric analysis variable.

```
proc sgplot data=<input-data-set> <options>;
  vbox variable < / options>;
run;
```

The CATEGORY= option can be used to create a separate box for each distinct value of a categorical variable. This can also be combined with the GROUP= option to add groups within each category.

THE BOXWIDTH= AND DISCRETEOFFSET= OPTIONS

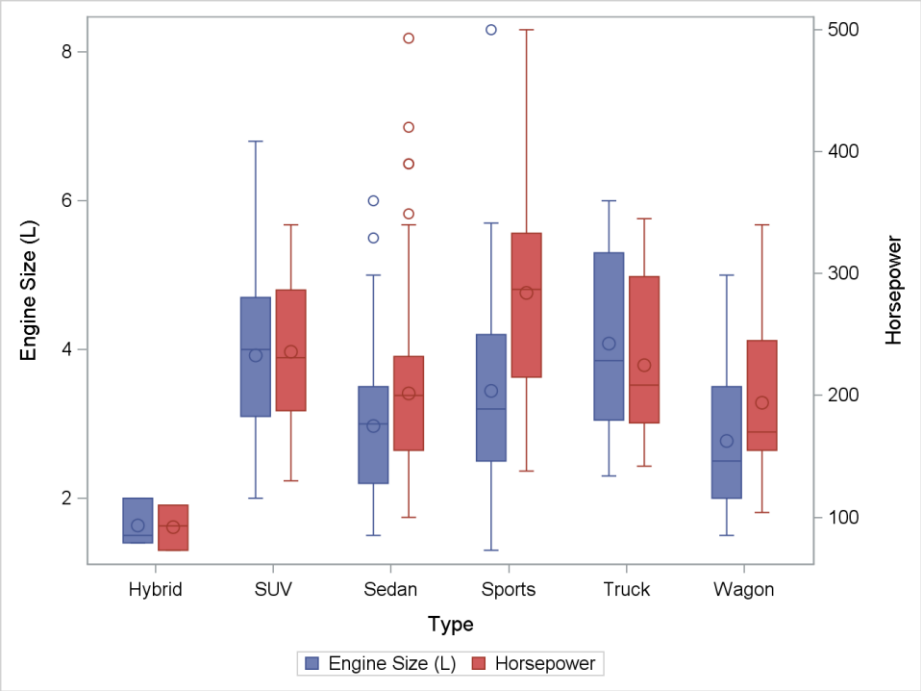
Two options that can be useful when combining multiple box plots are BOXWIDTH= and DISCRETEOFFSET=. These options are available on both the VBOX and HBOX statements.

The BOXWIDTH= option controls the width of the boxes. Valid values range from 0 to 1 and represent a proportion of the available width. The default value is 0.4.

The DISCRETEOFFSET= option specifies an amount by which to offset the box from the tick marks. Valid values range from -0.5 to 0.5. In the case of vertical boxes, -0.5 represents a left offset and 0.5 represents a right offset. The default value is 0, which corresponds to no offset at all.

EXERCISE

Using the SASHELP.CARS data set, create a vertical box plot of ENGINESIZE and HORSEPOWER for each TYPE of vehicle. You'll need two VBOX statements, one per analysis variable, along with the CATEGORY= option. Adjust the BOXWIDTH= and DISCRETEOFFSET= options so the boxes don't collide. Add the Y2AXIS option to one VLINE statement to utilize the secondary Y axis.



Exercise 2. Offset Dual Box Plot

SOLUTION

```
proc sgplot data=sashelp.cars;
  vbox enginesize / category=type boxwidth=0.25 discreteoffset=-0.15;
  vbox horsepower / category=type boxwidth=0.25 discreteoffset=0.15
       y2axis;
run;
```

EXERCISE #3: MODIFYING THE PLOT MARKERS

THE SCATTER STATEMENT

The SCATTER statement is used to create a scatter plot. It has two required arguments, X= and Y=, which specify the variables to plot. Here is the syntax:

```
proc sgplot data=<input-data-set> <options>;
  scatter x=variable y=variable < / options>;
run;
```

MARKERATTRS OPTION

The MARKERATTRS option allows us to specify marker attributes such as the marker symbol, size, and color. It can be used with any plot request statement that creates plot markers. The syntax consists of pairs of attribute names and values enclosed in parentheses as follows:

```
markerattrs=(symbol=symbol-name size=n <units> color=line-color)
```

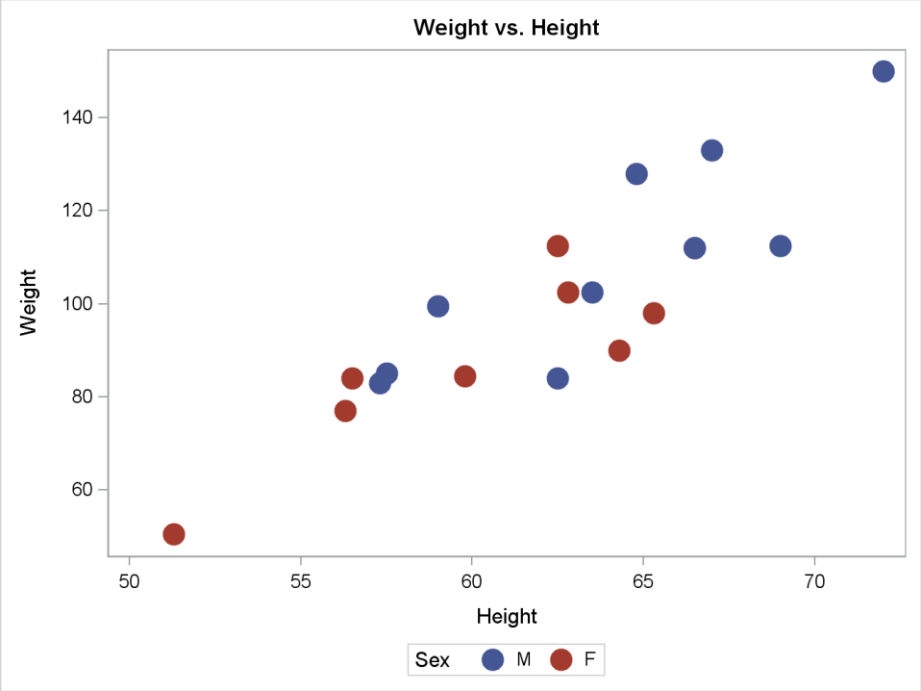
Marker Attribute Name	Sample Values
SYMBOL	Circle, CircleFilled, Square, Star, Plus, X
SIZE	0.2in, 3mm, 10pt, 5px, 25pct
COLOR	red, blue, lightgreen, aquamarine, CXFFFFFF

Table 1. Marker Attributes

For more detailed information about specifying marker attribute values, refer to *SAS 9.4 ODS Graphics: Procedures Guide* (SAS Institute, 2016).

EXERCISE

Using SASHELP.CLASS, create a scatter plot of WEIGHT vs HEIGHT grouped by SEX. Modify the plot markers to use filled circles 15 pixels in size. Add a title to the plot.



Exercise 3. Modifying the Plot Markers

SOLUTION

```
proc sgplot data=sashelp.class;
  title "Weight vs. Height";
  scatter x=height y=weight / group=sex
          markerattrs=(symbol=CircleFilled size=15px);
run;
```

EXERCISE 4: ADDING STYLE ATTRIBUTES

STYLEATTRS STATEMENT

In the previous example, we used the MARKERATTRS option to apply attributes to all markers. Had we chosen to apply a color in this manner, it would have affected all markers in the plot. This would make it impossible to distinguish which markers correspond with each value of the grouping variable.

Prior to SAS 9.4, specifying custom plot attributes by group required using PROC TEMPLATE to modify the style definition. Starting in SAS 9.4, the STYLEATTRS statement can be used for this purpose. The STYLEATTRS statement allows for the specification of a list of attribute values for each attribute.

```
styleattrs attr1=(value1 value2 …) attr2=(value1 value2 …)  … ;
```

The attribute names are not identical to those we used on the MARKERATTRS option in the previous example. For instance, instead of referring to SYMBOL and COLOR, we use DATASYMBOLS and DATACONTRASTCOLOR, respectively.

ATTRIBUTE CYCLING

When multiple lists of attributes are specified on the STYLEATTRS statement (for example, a list of marker shapes and a list of marker colors), there are two different methods of applying these attributes to groups. These two methods are color priority and no priority.

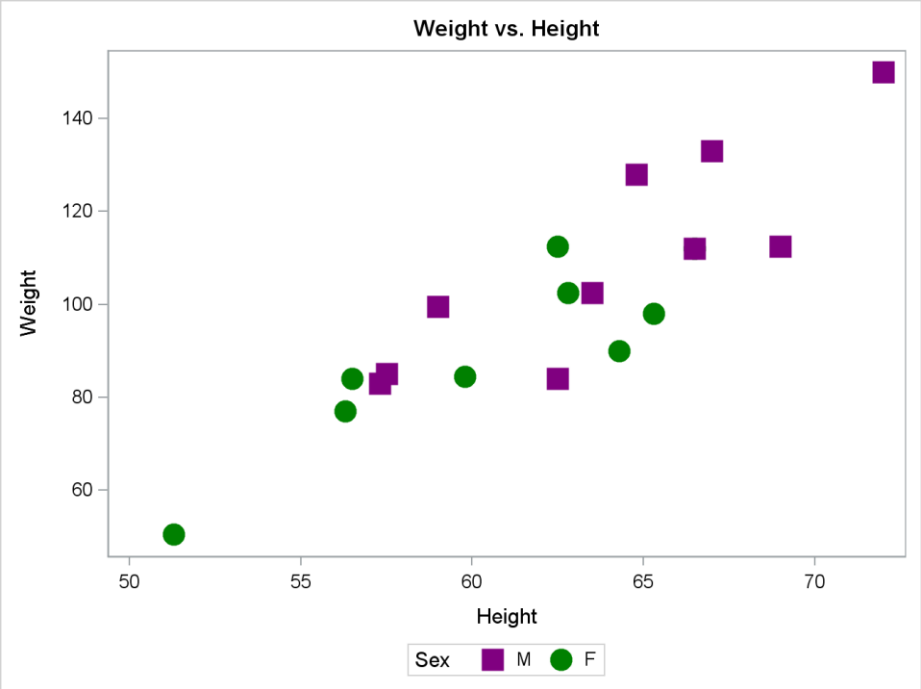
Under color priority, attributes are assigned to groups by cycling through all colors while holding the other attributes fixed before advancing to other values of the other attributes. Using no priority, the attributes are taken pairwise.

For example, if the list of colors includes red and blue and the list of symbols consists of a square followed by a circle, then the groups would be assigned attributes in this order under color priority: red square, blue square, red circle, blue circle. Specifying no priority would result in this ordering: red square, blue circle. Under both methods, values are recycled until attributes are assigned to all groups.

Use the ODS GRAPHICS statement with the option ATTRPRIORITY=COLOR to specify color priority. Similarly, ATTRPRIORITY=NONE is used to select no priority.

EXERCISE

Using the SASHELP.CLASS data set, create a scatter plot of WEIGHT vs. HEIGHT grouped by SEX. Use purple filled squares for males and filled green circles for females, all 15 pixels in size. You'll need to use a SCATTER statement with the X= and Y= arguments and the GROUP= and MARKERATTRS= options. You will also need a STYLEATTRS statement with the DATASYMBOLS= and DATACONTRASTCOLORS= options.



Exercise 4. Adding Style Attributes

SOLUTION

```
proc sgplot data=sashelp.class;
  title "Weight vs. Height";
  styleattrs datasymbols=(SquareFilled CircleFilled)
      datacontrastcolors=(purple green);
  scatter x=height y=weight / group=sex markerattrs=(size=15px);
run;
```

EXERCISE 5: MODIFYING LINE ATTRIBUTES

LINEATTRS OPTION

The LINEATTRS option allows us to specify line attributes such as the line pattern, thickness, and color. It can be used with any plot request statement that creates lines. Like the MARKERATTRS option, the syntax consists of pairs of attribute names and values enclosed in parentheses as follows:

```
lineattrs=(pattern=line-pattern thickness=n <units> color=line-color)
```

Marker Attribute Name	Sample Values
PATTERN	Solid, Dash, Dot, DashDashDot, LongDash
THICKNESS	0.2in, 3mm, 10pt, 5px, 25pct
COLOR	red, blue, lightgreen, aquamarine, CXFFFFFF

Table 2. Line Attributes

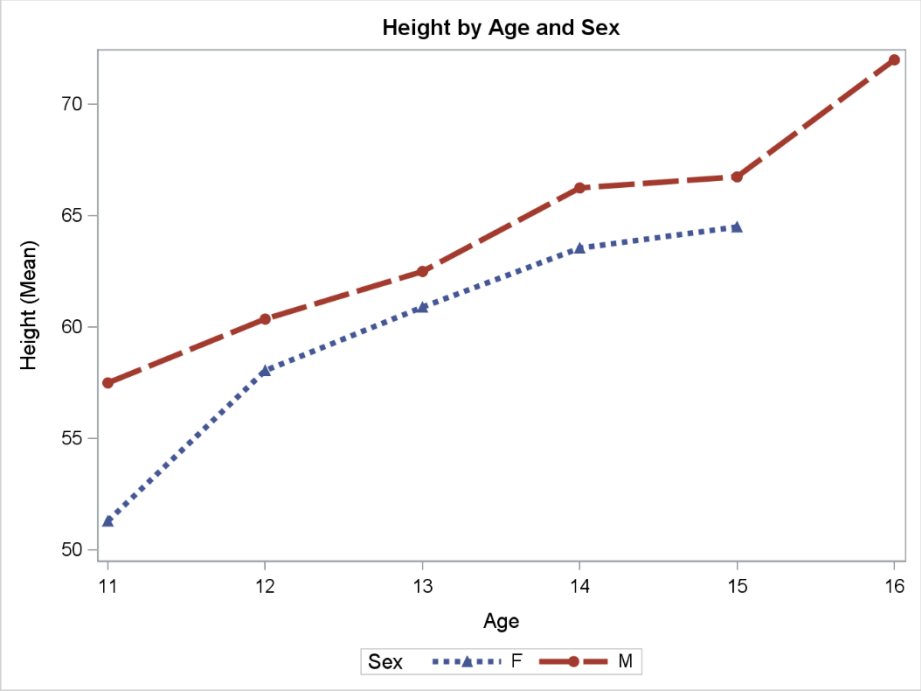
Just like with plot markers, we can use the STYLEATTRS statement to specify line attributes that vary by group. The DATALINEPATTERNS= option is used to specify a list of line patterns, and DATACONTRASTCOLORS= is used for line colors. For more detailed information about specifying line attribute values, refer to *SAS 9.4 ODS Graphics: Procedures Guide* (SAS Institute, 2016).

EXERCISE

Using the SASHELP.CLASS data set, create a vertical line chart of mean HEIGHT by AGE grouped by SEX. Modify the lines and plot markers as follows:

Males: ShortDash line pattern, 4 pixels thick, TriangleFilled marker symbol
Females: LongDash line pattern, 4 pixels thick, CircleFilled marker symbol

Use the VLINE statement with a categorical variable and the RESPONSE=, STAT=, GROUP=, and MARKERS options. Add the LINEATTRS= option to control the line thickness. In addition, use a STYLEATTRS statement to specify the line patterns and marker symbols with the DATASYMBOLS= and DATALINEPATTERNS= options.



Exercise 5. Modifying Line Attributes

SOLUTION

```
proc sgplot data=sashelp.class;
  title "Height by Age and Sex";
  vline age / response=height stat=mean markers
              group=sex lineattrs=(thickness=4px);
  styleattrs datasymbols=(TriangleFilled CircleFilled)
             datalinepatterns=(ShortDash LongDash);
run;
```

EXERCISE #6: MODIFYING THE LEGEND

KEYLEGEND STATEMENT

In several exercises, we've seen the SGPLOT procedure automatically add a legend. The KEYLEGEND statement allows us to customize many aspects of the legend. The following table summarizes selected options provided by the KEYLEGEND statement.

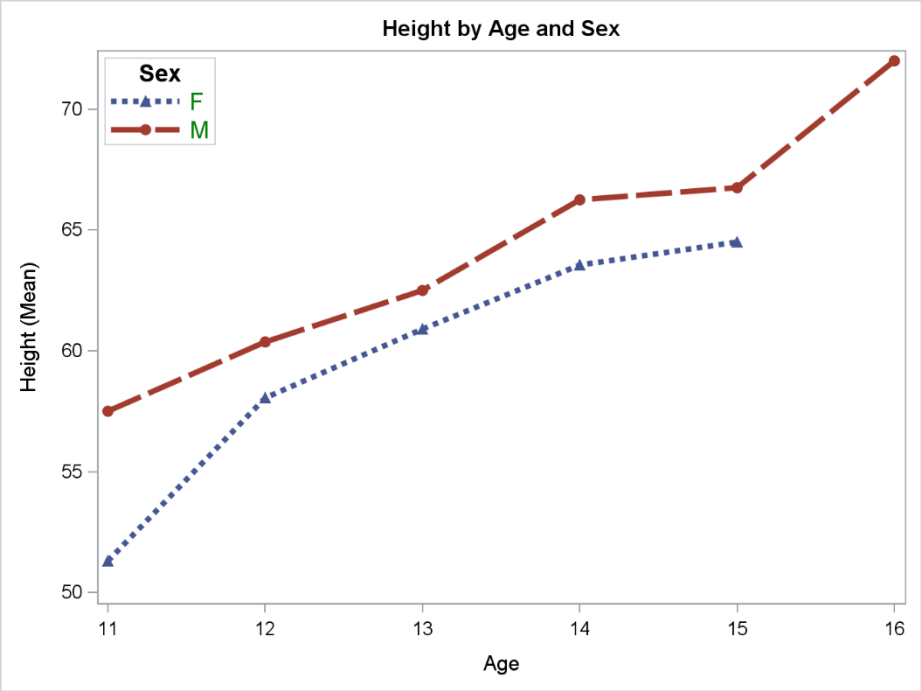
Legend Option	Description
LOCATION=	Specifies whether legend will appear INSIDE or OUTSIDE (default) the axis area.
POSITION=	Specifies the position of the legend: TOP, BOTTOM (default), LEFT, RIGHT, TOPLEFT, TOPRIGHT, BOTTOMLEFT, BOTTOMRIGHT
ACROSS=	Specifies number of columns in legend
DOWN=	Specifies number of rows in legend
TITLEATTRS=	Specifies text attributes of legend title
VALUEATTRS=	Specifies text attributes of legend values

Table 3. Selected KEYLEGEND Options

Those options which specify text attributes, such as TITLEATTRS and VALUEATTRS, will themselves consist of lists of attribute names and values (much like MARKERATTRS and LINEATTRS). Text attributes include COLOR, FAMILY (font), SIZE, STYLE (italic or normal), and WEIGHT (bold or normal).

EXERCISE

Modify the plot created in Exercise 5 to move the legend to the inside top left and place the values in a single column. Use a bold 12-point legend title and green 12-point values. Use the KEYLEGEND statement with the LOCATION=, POSITION=, ACROSS=, TITLEATTRS=, and VALUEATTRS= options.



Exercise 6. Modifying the Legend

SOLUTION

```
proc sgplot data=sashelp.class;
  title "Height by Age and Sex";
  vline age / response=height stat=mean markers
              group=sex lineattrs=(thickness=4px);
  styleattrs datasymbols=(TriangleFilled CircleFilled)
             datalinepatterns=(ShortDash LongDash);
  keylegend / location=inside position=topleft across=1
              titleattrs=(weight=bold size=12pt)
              valueattrs=(color=green size=12pt);
run;
```

EXERCISE #7: ADDING A REFERENCE LINE

REFLINE STATEMENT

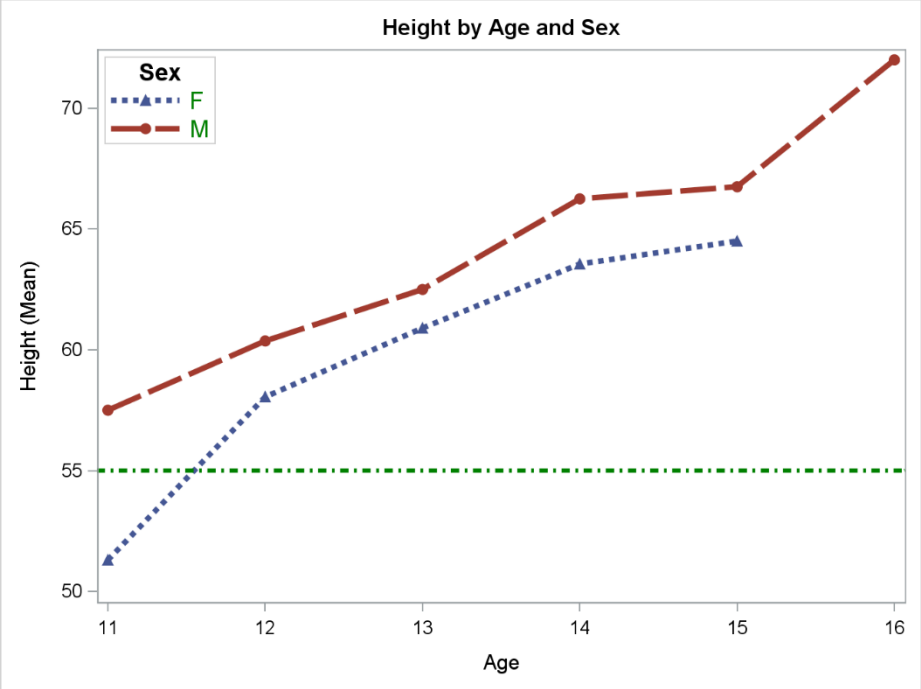
The REFLINE statement adds horizontal or vertical reference lines to a plot. Its unnamed required argument is a numeric variable, value, or list of values. A reference line will be added for each value listed or for each value of the variable specified.

```
refline <value(s) or var> / <options>;
```

The AXIS= option is used to specify which axis contains the reference line value(s). Valid values are X, Y, X2, and Y2. By default, the Y axis is used. The LINEATTRS= option specifies line attributes. The attributes and values are the same described in Table 2 in Exercise 5.

EXERCISE

Modify the plot created in Exercise 6 to add a horizontal reference line corresponding to 55 inches. Use a green line, 3 pixels thick, with the ShortDashDot line pattern. Add a REFLINE statement with a numeric constant and use the AXIS= and LINEATTRS= options.



Exercise 7. Adding a Reference Line

SOLUTION

```
proc sgplot data=sashelp.class;
  title "Height by Age and Sex";
  vline age / response=height stat=mean markers
              group=sex lineattrs=(thickness=4px);
  styleattrs datasymbols=(TriangleFilled CircleFilled)
             datalinepatterns=(ShortDash LongDash);
  keylegend / location=inside position=topleft across=1
              titleattrs=(weight=bold size=12pt)
              valueattrs=(color=green size=12pt);
  refline 55 / axis=y lineattrs=(color=green thickness=3px
                                 pattern=ShortDashDot);
run;
```

EXERCISE #8: MODIFYING THE AXES

XAXIS AND YAXIS STATEMENTS

The XAXIS and YAXIS statements are used to control the features and structure of the X and Y axes, respectively. There are no required arguments, but dozens of options provide extensive customization.

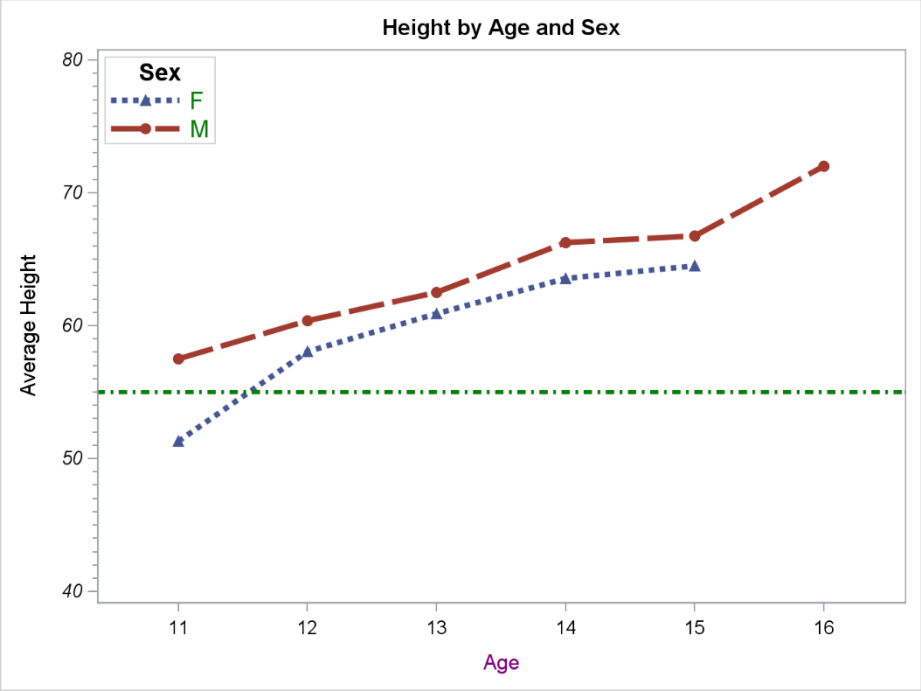
Axis Option	Description
LABEL=	Specifies axis label
LABELATTRS=	Specifies text attributes of axis label
VALUEATTRS=	Specifies text attributes of tick value labels
MIN= and MAX=	Specify minimum and maximum data values to include in display
MINOR	Adds minor tick marks to the axis
MINORCOUNT=	Specifies the number of minor tick marks between each major tick mark
OFFSETMIN=	Specifies an offset below the lowest data value on the axis
OFFSETMAX=	Specifies an offset after the highest data value on the axis

Table 4. Selected Axis Options

Valid values for OFFSETMIN= and OFFSETMAX= range from 0 to 1 and represent a proportion to the total length of the axis. As before, those options which pertain to text attributes, such as LABELATTRS and VALUEATTRS, will themselves consist of lists of text attribute names and values. Text attributes include COLOR, FAMILY (font), SIZE, STYLE (italic or normal), and WEIGHT (bold or normal).

EXERCISE

Modify the plot created in Exercise 7. Add a 10% offset at each end of the X axis and make the X axis label purple. Set the Y axis range to 40 through 80 with tick labels in italics and minor ticks every inch. Label the Y axis “Average Height”. Add an XAXIS statement with the OFFSETMIN=, OFFSETMAX=, and LABELATTRS= options. Add a YAXIS statement with the MIN=, MAX=, MINOR, MINORCOUNT=, VALUEATTRS=, and LABEL= options.



Exercise 8. Modifying the Axes

SOLUTION

```
proc sgplot data=sashelp.class;
  title "Height by Age and Sex";
  vline age / response=height stat=mean markers
              group=sex lineattrs=(thickness=4px);
  styleattrs datasymbols=(TriangleFilled CircleFilled)
             datalinepatterns=(ShortDash LongDash);
  keylegend / location=inside position=topleft across=1
              titleattrs=(weight=bold size=12pt)
              valueattrs=(color=green size=12pt);
  refline 55 / axis=y lineattrs=(color=green thickness=3px
                                 pattern=ShortDashDot);
  xaxis offsetmin=0.1 offsetmax=0.1 labelattrs=(color=purple);
  yaxis min=40 max=80 minor minorcount=9
        valueattrs=(style=italic)
        label="Average Height";
run;
```

EXERCISE #9: AXIS TABLES

XAXISTABLE AND YAXISTABLE STATEMENTS

The XAXISTABLE and YAXISTABLES statements create axis tables which display data values at specific locations along an axis. The only required argument is a list of one or more variables to be displayed.

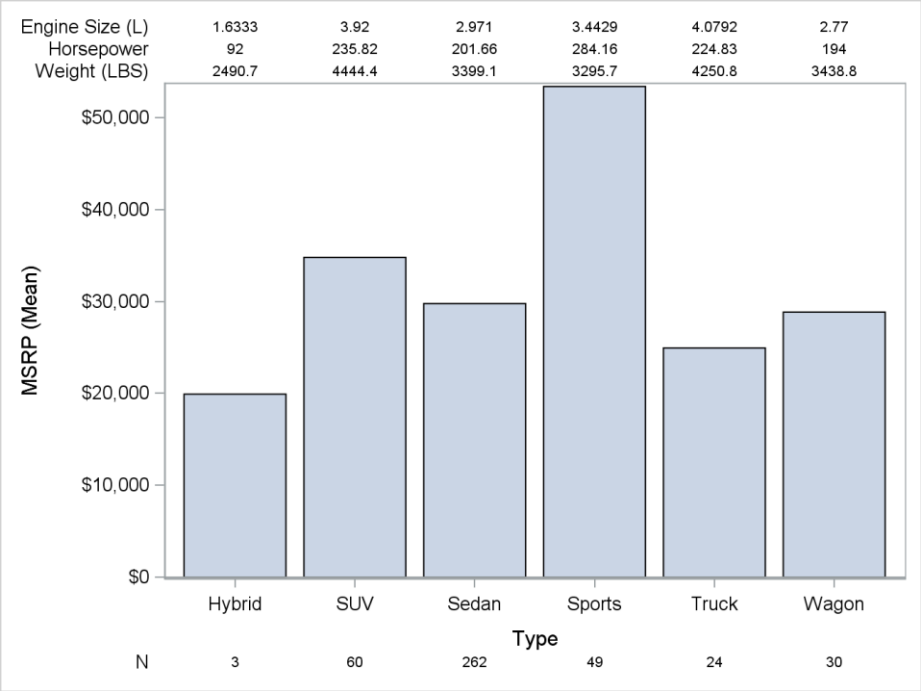
```
xaxistable variable <…variable-n> / <options>;
```

The POSITION= option specifies where the table will be displayed. For X axis tables, valid values are BOTTOM (the default) and TOP, while Y axis tables can be positioned to the LEFT (default) or RIGHT.

The STAT= option is used to specify a statistic to be applied to numeric variables. The default is SUM. The LABEL= option can be used to provide a text string to label the axis table.

EXERCISE

Using the SASHELP.CARS data set, create a vertical bar chart of mean MSRP by vehicle TYPE with two axis tables: one at the top showing mean ENGINESIZE, HORSEPOWER, and WEIGHT, and one at the bottom showing N. You'll need a VBAR statement and two XAXISTABLE statements. To get N, use the FREQ statistic on any numeric variable in the data set and relabel it using the LABEL= option.



Exercise 9. Axis Tables

SOLUTION

```
proc sgplot data=sashelp.cars;
  vbar type / response=msrp stat=mean;
  xaxistable enginesize horsepower weight / stat=mean position=top;
  xaxistable enginesize / stat=freq label="N";
run;
```

In the second XAXISTABLE statement, the ENGINESIZE variable was chosen arbitrarily in order to obtain frequency counts of the number of records in each category.

You can also try using the TITLEATTRS=, LABELATTRS=, and VALUEATTRS= options on the XAXISTABLE statement to modify the text attributes of the various axis table components.

EXERCISE #10: USING A SYMBOL CHARACTER

THE SYMBOLCHAR STATEMENT

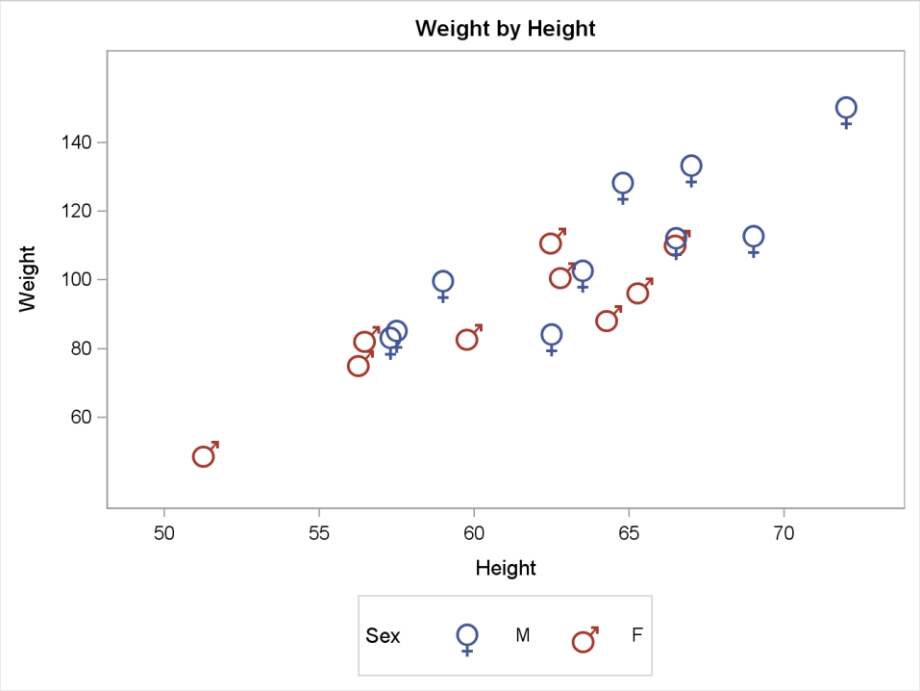
The SYMBOLCHAR statement is used to define a marker symbol from a Unicode value. The required argument NAME= allows the user to specify a unique identifier to use when referring to the symbol later. The required argument CHAR= provides the Unicode value for the desired character. It is common practice to refer to Unicode characters by their hexadecimal value. In SAS, hexadecimal constants can be specified by enclosing the hexadecimal string in double quotes followed immediately by an x.

```
symbolchar name=id char="hex-value"x </ options>;
```

Additional options are available to allow for the rotation, scaling, and offset of the character as well as modifications to its text attributes.

EXERCISE

Using the SASHELP.CLASS data set, create a scatter plot of WEIGHT by HEIGHT grouped by SEX, using the male and female symbols as the plot characters. Use two SYMBOLCHAR statements with the NAME= and CHAR= options to define the marker symbols. Then, use a STYLEATTRS statement with the DATASYMBOLS= option to apply these symbols to the data groups. Of course, you'll also need a SCATTER statement. Add a title as well.



Exercise 10. Using a Symbol Character

SOLUTION

```
proc sgplot data=sashelp.class;
  title "Weight by Height";
  scatter x=height y=weight / group=sex markerattrs=(size=40);
  symbolchar name=female_sign char="2640"x;
  symbolchar name=male_sign   char="2642"x;
  styleattrs datasymbols=(female_sign male_sign);
run;
```

Note the identifiers "female_sign" and "male_sign" are arbitrary names we assigned here.

EXERCISE #11: USING DATA AS A SYMBOL MARKER

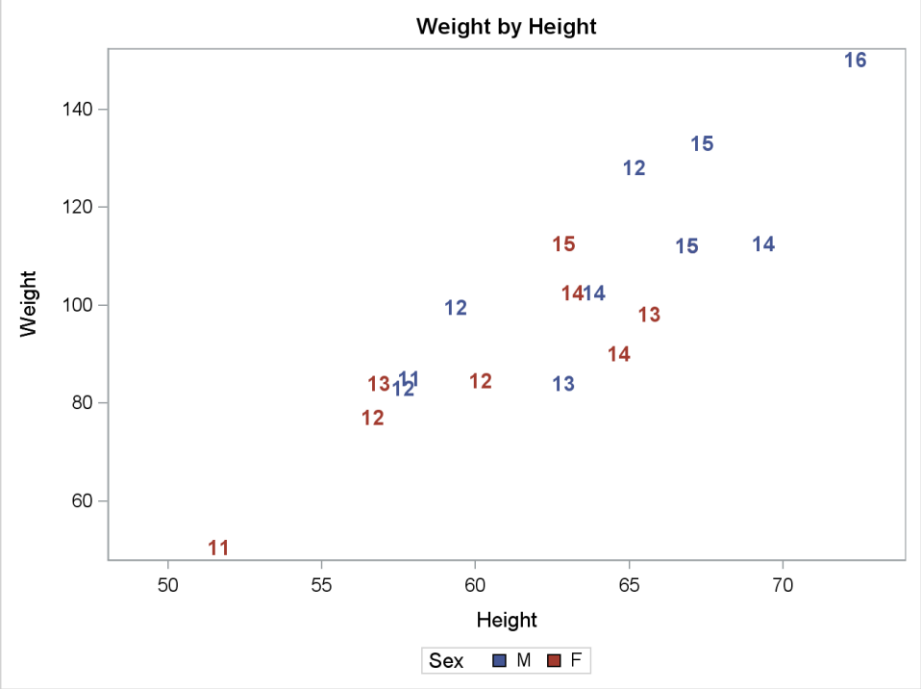
THE MARKERCHAR= OPTION

The MARKERCHAR= option specifies a variable who values are used as plot symbols. It applies only to the SCATTER statement. The MARKERCHARATTRS= option can be used in conjunction with MARKERCHAR= to modify its text attributes (color, font family, size, style, and weight).

The MARKERCHAR= option differs from the DATALABEL= option. The latter adds labels to plot markers based on the values of a variable, while MARKERCHAR= actually replaces the plot markers with the values. The two options cannot be used together.

EXERCISE

Using SASHELP.CLASS, create a scatter plot of WEIGHT by HEIGHT grouped by SEX, using the variable AGE in 10-point bold font as the plot marker. Write a SCATTER statement using the X=, Y=, GROUP=, MARKERCHAR= and MARKERCHARATTRS= options.



Exercise 11. Using Data as a Symbol Marker

SOLUTION

```
proc sgplot data=sashelp.class;
  title "Weight by Height";
  scatter x=height y=weight / group=sex markerchar=age
        markercharattrs=(weight=bold size=10pt);
run;
```

EXERCISE #12: USING A SYMBOL IMAGE

SYMBOLIMAGE STATEMENT

The SYMBOLIMAGE statement is used to define a marker symbol from an image file. The required argument NAME= specifies a unique identifier to use when referring to the symbol later. The required argument IMAGE= must contain the name and location of the image file.

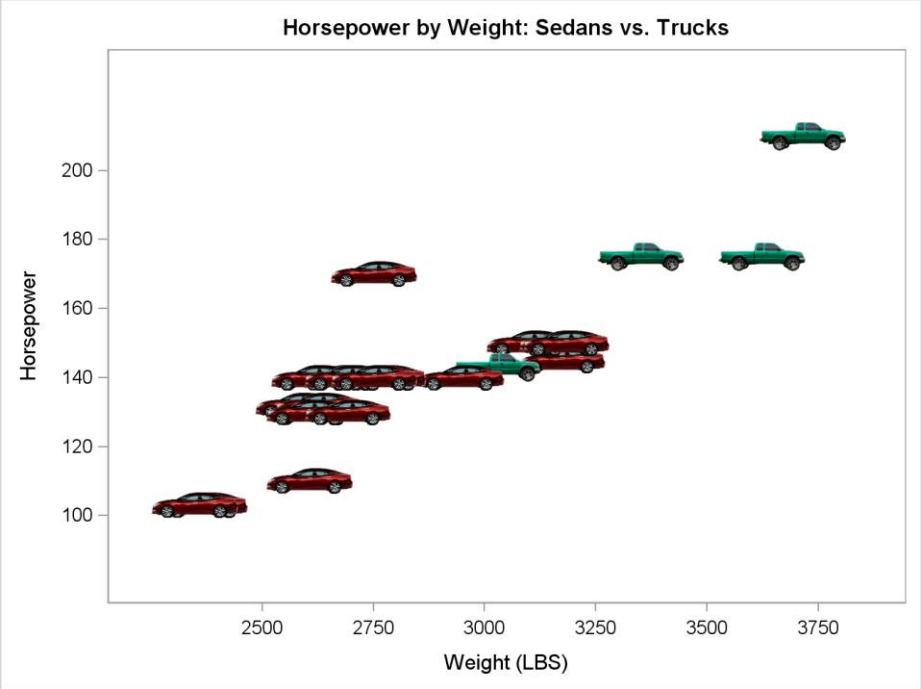
```
symbolimage name=id image="file-spec" </ options>;
```

Additional options are available to allow for the rotation, scaling, and offset of the image file. Supported image formats include PNG, JPG, and GIF.

EXERCISE

Using the SASHELP.CARS data set, create a scatter plot of HORSEPOWER by WEIGHT grouped by TYPE for U.S. trucks and sedans costing less than $20,000. Use images of a truck and a sedan as plot markers. Suppress the automatic legend by including the NOAUTOLEGEND option on the PROC SGPLOT statement. You'll need two SYMBOLIMAGE statements using the NAME= and IMAGE= options and a STYLEATTRS statement with the DATASYMBOLS= option. You'll also need a SCATTER statement as well as a WHERE statement to subset the data. Add a title.

(Note to reader: In the live workshop, image files are supplied to attendees. You'll have to find your own.)



Exercise 12. Using a Symbol Image

SOLUTION

```
proc sgplot data=sashelp.cars noautolegend;
  title "Horsepower by Weight: Sedans vs. Trucks";
  where type in ('Sedan','Truck') and origin='USA' and MSRP < 20000;
  scatter x=weight y=horsepower / group=type markerattrs=(size=60);
  symbolimage name=car image="&srcdir.\car.png";
  symbolimage name=truck image="&srcdir.\truck.png";
  styleattrs datasymbols=(car truck);
run;
```

In the above code, it is assumed that a macro variable called SRCDIR has been defined containing the full path to the location of the image files.

WRAP-UP

The SGPLOT procedure provides for extensive customization of statistical graphs. Its vast catalog of statements and options makes it very versatile. There are many more features beyond those described in this paper. Consult the *SAS 9.4 ODS Graphics: Procedures Guide, Sixth Edition* (SAS Institute, 2016) for complete reference material on all the capabilities of SGPLOT.

REFERENCES

Horstman, Joshua M. "Getting Started with the SGPLOT Procedure." Proceedings of the SAS Global Forum 2019 Conference. Cary, NC: SAS Institute Inc., 2019. https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2019/3167-2019.pdf

SAS 9.4 ODS Graphics: Procedures Guide, Sixth Edition. Cary, NC: SAS Institute, 2016. https://documentation.sas.com/api/docsets/grstatproc/9.4/content/grstatproc.pdf.

CONTACT INFORMATION

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

Joshua M. Horstman
Nested Loop Consulting
317-721-1009
josh@nestedloopconsulting.com