

Keeping Up to Date with ODS Graphics

Warren F. Kuhfeld and Dan Heath, SAS Institute Inc.

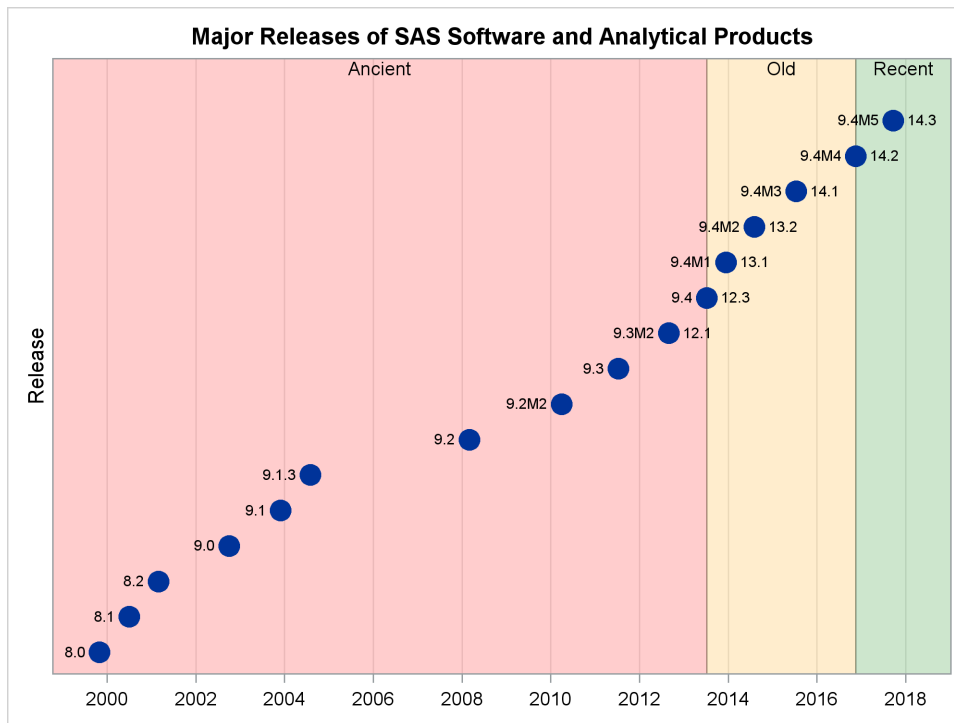
ABSTRACT

SAS® 9.4M5 provides you with several enhancements to ODS Graphics, including the new SGMAP procedure. You can use PROC SGMAP to create maps and superimpose graphs such as bubble plots. Bar charts in existing procedures such as PROC SGPLOT have new options for fill patterns and fill types. New options for box plots enable you to display statistics and control the caps on whiskers. Other options enable you to modify tick labels, tick styles, legends, baselines, and reference line thickness. You can also control the image names when there are BY groups. This paper describes and illustrates these recent updates.

INTRODUCTION

Graphics have a long history in SAS® software. In the 1970s, procedures such as PROC PLOT and PROC CHART produced character-based graphical displays that could be rendered on a line printer. Then SAS/GRAPH® software appeared in the 1980s and provided true computerized graphics. These graphics, while modern in the 1980s, were no longer modern by the 2000s. In 2008, a new and modern graphical method appeared in SAS: ODS Graphics. Figure 1 presents a time line of SAS releases over the past 20 years.¹ If you are using an old release, you will not have all the newest features, including many that appear in this paper.

Figure 1 SAS Releases over the Past 20 Years



ODS Graphics was originally available only if you licensed SAS/GRAPH software. In 2011, ODS Graphics became part of Base SAS® software. Since 2011, all SAS users have been able to make great graphs without paying for an extra license.

Procedures such as PLOT, CHART, GPLOT, and GCHART are still in use, even though they are based on methods that are now decades old. However, you will find that ODS Graphics is easier to use than the legacy methods, and the graphs are more modern.

¹The data and the original code for the graph are courtesy of Rick Wicklin and the [DO Loop](#) blog.

Development in ODS Graphics has never stopped, and ODS Graphics continues to evolve. Some new features give you new functionality. Other new features make it easier to do things that were harder to do in previous releases. Many new features were added because of a customer request. New features in SAS 9.4M5 include these:

- PROC SGMAP
- bar fill patterns
- bar fill types
- box plot statistics displays
- image names created from BY-group values
- axis value fitting
- linear and proportional bubbles
- legend sorting
- bar label fit policy
- box plot cap scale
- baseline display
- tick style, inside or outside
- legend items
- reference line thickness on discrete axes

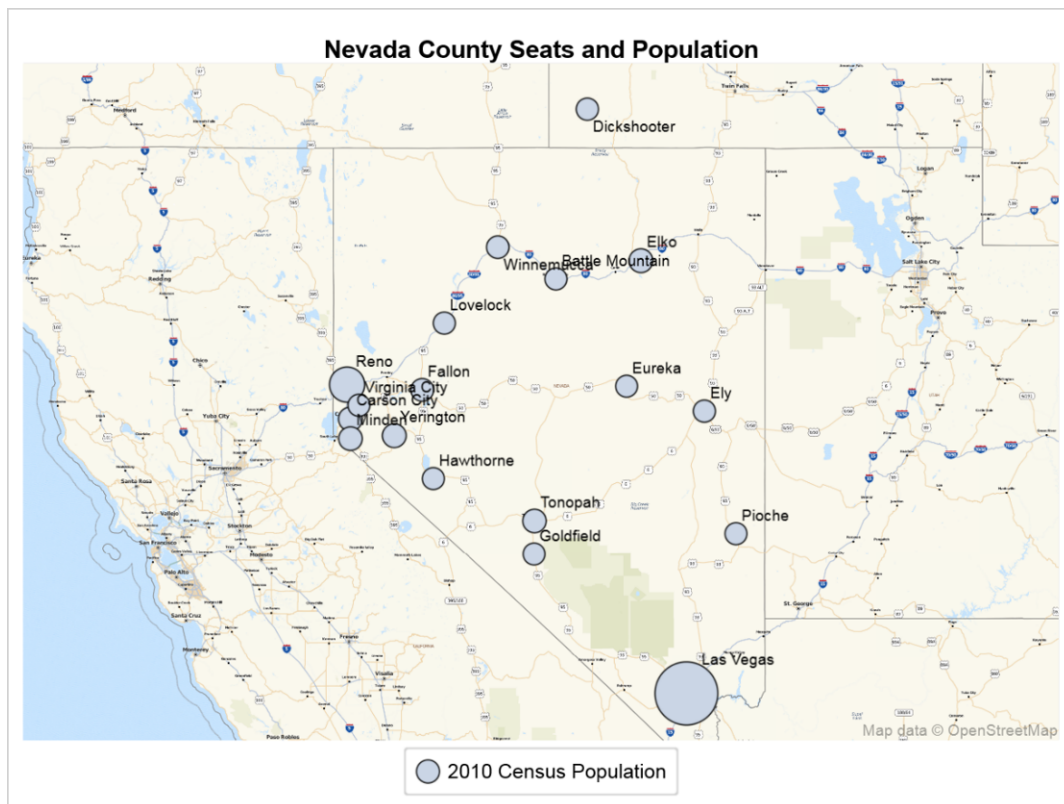
NEW FEATURES OF SAS 9.4M5

PROC SGMAP

ODS Graphics has a new procedure in SAS 9.4M5, PROC SGMAP. Like PROC GMAP in SAS/GRAPH software, you can use PROC SGMAP to produce maps. You have long been able to produce maps by using PROC SGPLOT and the POLYGON statement, but PROC SGMAP is specifically designed for maps. (For more information, see the [Graphically Speaking](#) blog.) PROC SGMAP is documented along with PROC GMAP and the other mapping procedures, not with the other SG procedures. The [PROC SGMAP](#) documentation and the Graphically Speaking blog show how to create data sets suitable for input into PROC SGMAP. After creating the data sets, you can submit the following code to produce the map in Figure 2:

```
proc sgmmap mapdata=sasuser.Nevada plotdata=resp_loc;
  title 'Nevada County Seats and Population';
  openstreetmap;
  bubble x=x y=y size=population /
    name='cty_pts' datalabel=city;
  keylegend 'cty_pts';
run;
```

Figure 2 PROC SGMAP Output



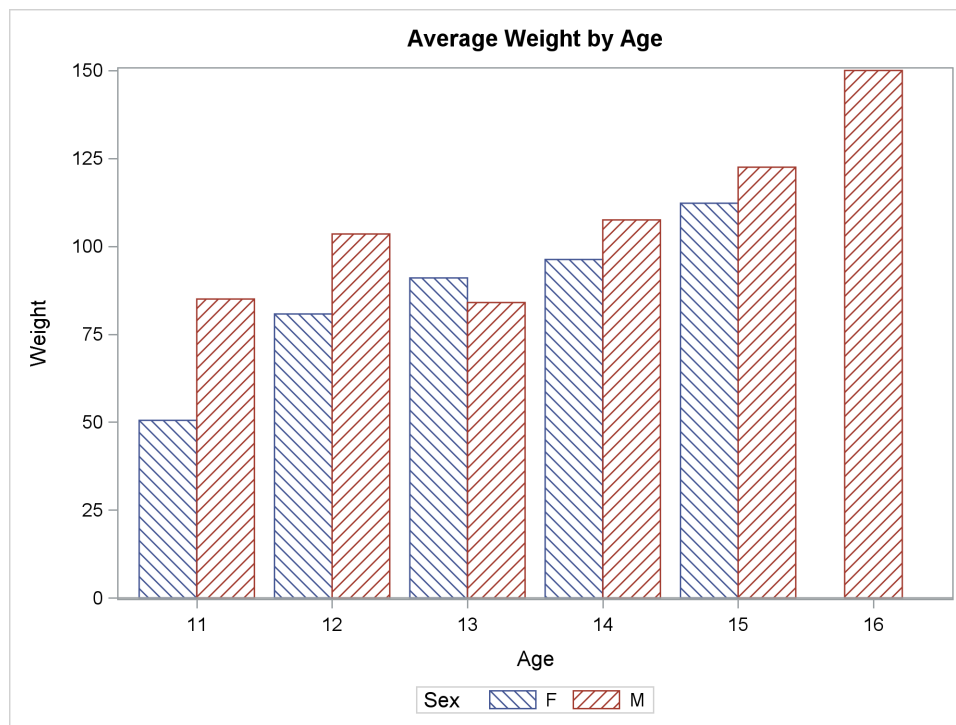
FILL PATTERNS

There are new options in SAS 9.4M5 for fill patterns. Fill patterns provide a way to distinguish groups in black-and-white graphs (such as those created by using one of the Journal family of styles). They can also be used with colors, as is shown here. The following step creates the graph in [Figure 3](#):

```
ods graphics on / attrpriority=None;
proc sgplot data=sashelp.class;
  title "Average Weight by Age";
  styleattrs datafillpatterns=(L1 R1);
  vbar age / response=weight stat=mean nofill fillpattern group=sex
    groupdisplay=cluster nostatlabel;
run;
```

The ODS GRAPHICS statement option ATTRPRIORITY=NONE is often required when you use both a STYLEATTRS statement and an ATTRPRIORITY=COLOR style such as HTMLBlue. Group colors, contrast colors, marker symbols, marker sizes, line styles, line thicknesses, and fill patterns are all defined in the **GraphData1** through **GraphData12** style elements. The values that you specify in the STYLEATTRS statement override those style definitions, but specifying a STYLEATTRS statement does not force your specifications to be used. You will see the second fill pattern (or the second of any of the other attributes, excluding color) only when the attribute priority is NONE. The attribute priority can be set to NONE in three ways. 1) You can use an ATTRPRIORITY=NONE style because that is the default type of style for your destination. 2) You can specify an ATTRPRIORITY=NONE style in your ODS destination statement. 3) You can override the attribute priority by specifying the ATTRPRIORITY=NONE option in the ODS GRAPHICS statement. The styles HTMLBlue, Pearl, PearlJ, and Sapphire are ATTRPRIORITY=COLOR styles. Most other styles are ATTRPRIORITY=NONE styles.

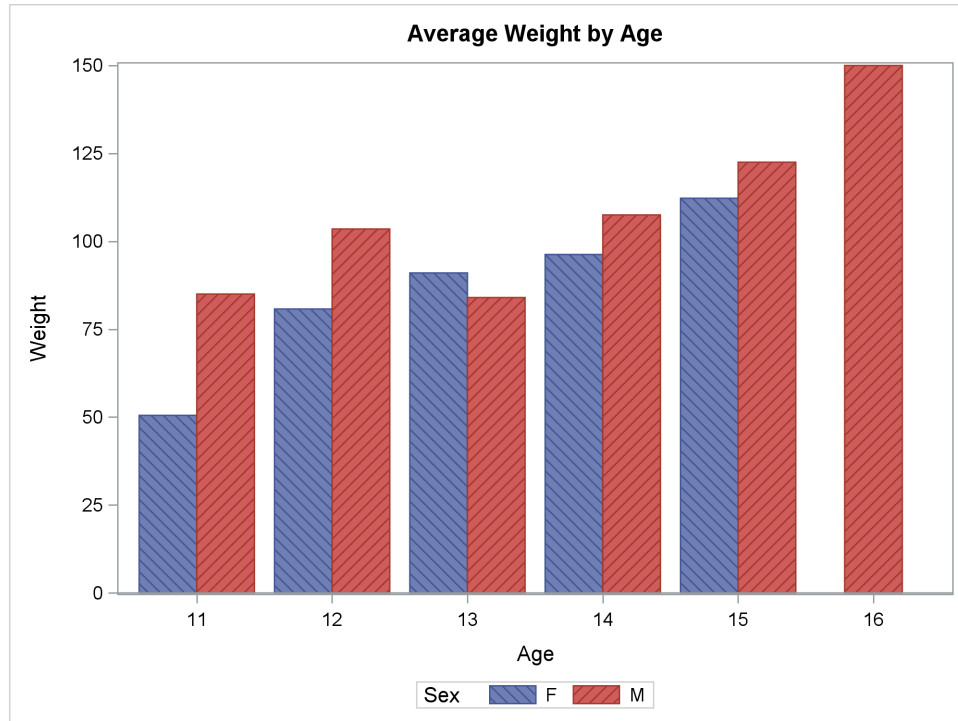
Figure 3 Fill Patterns



You can combine colored fills and fill patterns. The following step creates the graph in [Figure 4](#):

```
proc sgplot data=sashelp.class;
  styleattrs datafillpatterns=(L1 R1);
  vbar age / response=weight stat=mean fillpattern group=sex
    groupdisplay=cluster;
run;
ods graphics on / attrpriority=Color;
```

Figure 4 Both Fill Colors and Fill Patterns



The available fill patterns along with their names are displayed in [Figure 5](#).

Figure 5 Fill Patterns

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	

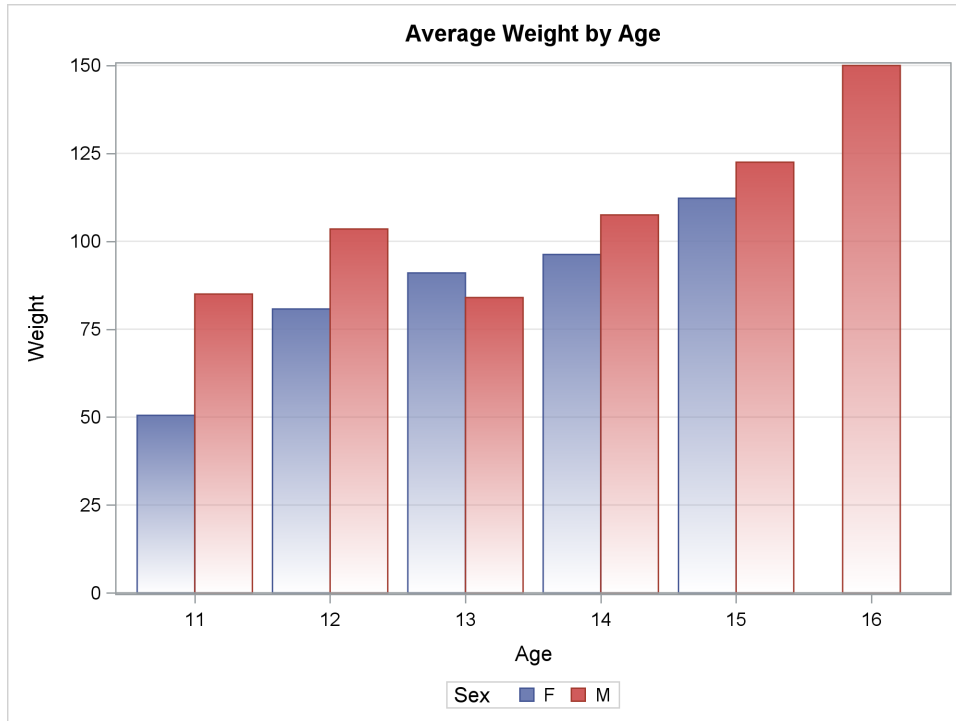
FILL TYPES

You can use a gradient along with fill colors. The following step creates the graph in [Figure 6](#):

```
proc sgplot data=sashelp.class;
  yaxis grid;
  vbar age / response=weight stat=mean filltype=alphagradient
    group=sex groupdisplay=cluster nostatlabel;
run;
```

You can control the starting color by specifying the FILLATTRS=(COLOR=*color*) option (not shown). For nongrouped plots, the default starting color comes from the `GraphDataDefault` style element. You can control the end color by specifying the FILLENDCOLOR= option.

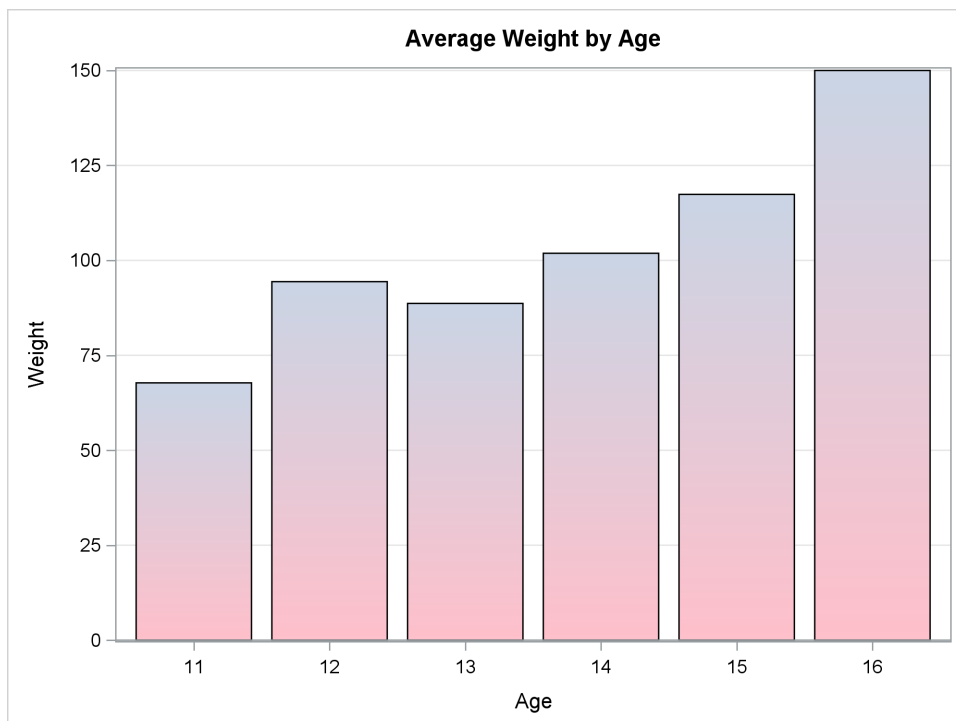
Figure 6 Fill Gradient



The following step sets the fill end color to pink and creates the graph in [Figure 7](#):

```
proc sgplot data=sashelp.class;  
  yaxis grid;  
  vbar age / response=weight stat=mean filltype=gradient  
    fillendcolor=pink groupdisplay=cluster nostatlabel;  
run;
```

Figure 7 Fill Gradient and End Color



DISPLAY STATISTICS IN BOX PLOTS

The DISPLAYSTATS= option enables you to display statistics in an X-axis table along with box plots. Statistics are displayed in the order in which you list them. The following step creates the graph in [Figure 8](#):

```
proc sgplot data=sashelp.heart tplout='t';  
  title "Cholesterol Distribution by Weight Status";  
  vbox cholesterol / category=weight_status displaystats=(q1 median mean q3);  
run;
```

Figure 8 Box Plot Statistics

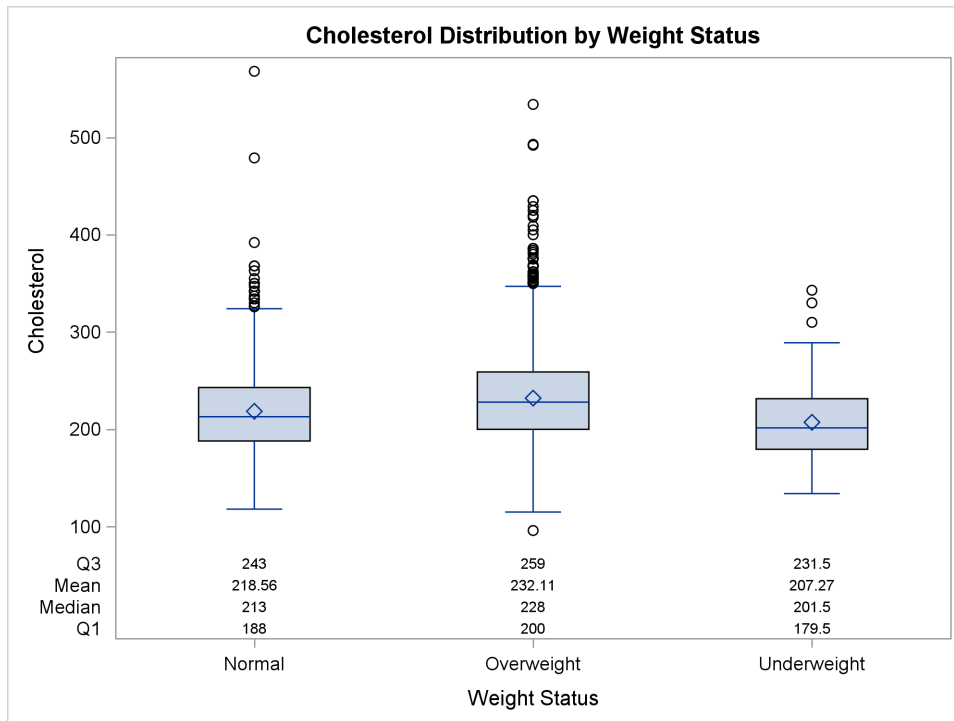
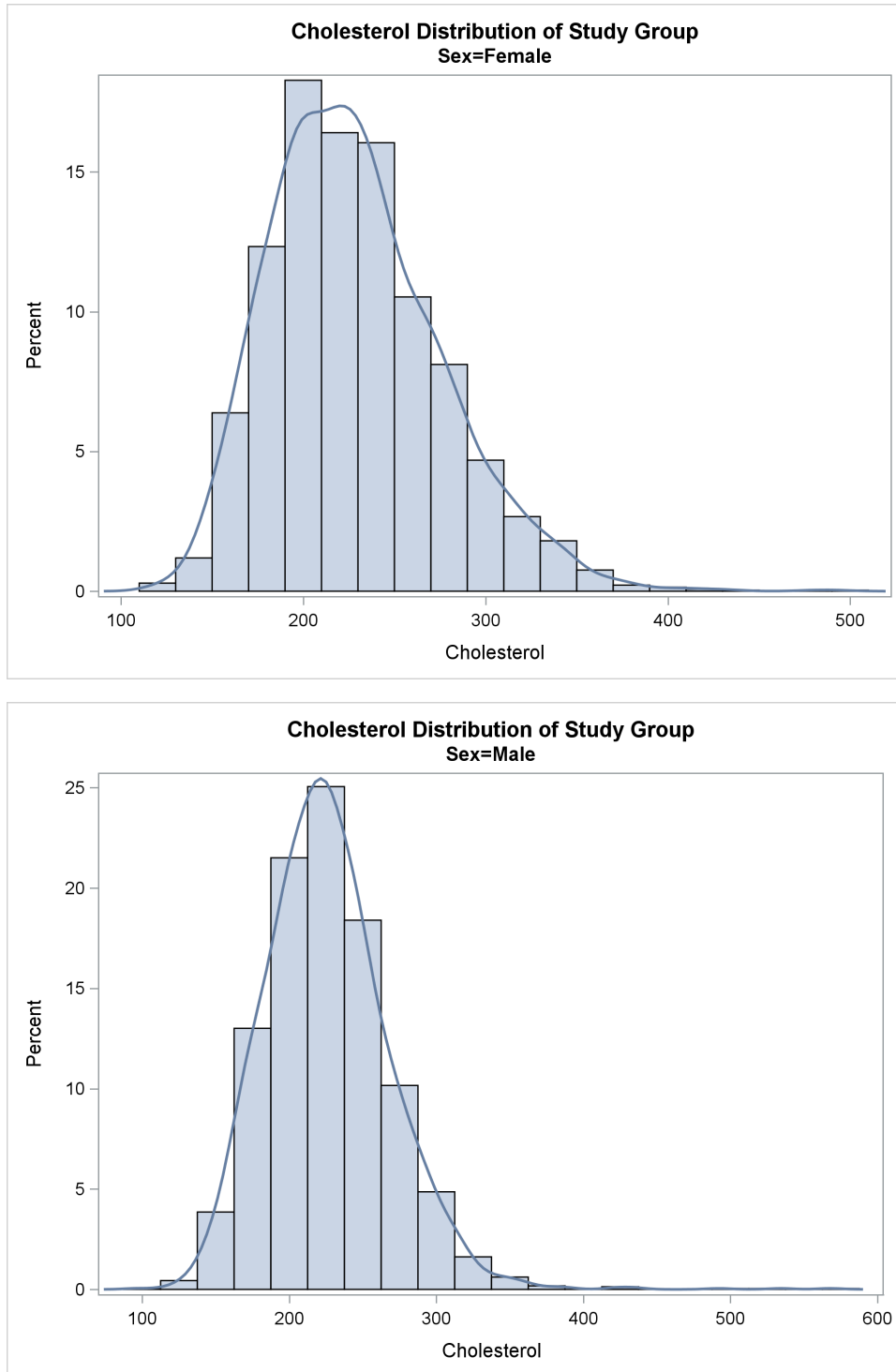


IMAGE NAMES MATCH BY-GROUP VALUES

You can control the output graph image names when you have a BY variable so that the image names match the values of the BY variable. You can do this by specifying the IMAGENAME= option in the ODS GRAPHICS statement. The following step creates the graphs in [Figure 9](#):

```
title 'Cholesterol Distribution of Study Group';  
ods graphics / reset=index imagename="#byval(sex)";  
  
proc sort data=sashelp.heart out=heart; by sex; run;  
  
proc sgplot data=heart noautolegend;  
  histogram cholesterol;  
  density cholesterol / type=kernel;  
  by sex;  
run;  
  
ods graphics / reset=imagename;
```

Figure 9 BY Groups



Each graph contains a BY line that contains the variable name and the BY-group value but not the full image name. The BY lines are "Sex=Female" and "Sex=Male". The image names are *Female.png* and *Male.png*. By default, each image name has the form *SGPlotn.png*. The two values of *n* are 1 and 2 the first time PROC SGPLOT is run, 3 and 4 the second time PROC SGPLOT is run, and so on. The values of *n* are different if there are multiple open output destinations.

X-AXIS VALUE ROTATIONS

The VALUESROTATE= option enables you to rotate X-axis text in several ways. The following steps create the graphs in Figure 10:

```
%let rot = ValuesRotate=Diagonal;
proc sgplot data=sashelp.class;
  title "&rot";
  vbar name / response=weight nostatlabel;
  xaxis &rot;
run;
```

```
%let rot = ValuesRotate=Diagonal2;
proc sgplot data=sashelp.class;
  title "&rot";
  vbar name / response=weight nostatlabel;
  xaxis &rot;
run;
```

```
%let rot = ValuesRotate=Vertical;
proc sgplot data=sashelp.class;
  title "&rot";
  vbar name / response=weight nostatlabel;
  xaxis &rot;
run;
```

```
%let rot = FitPolicy=StackedAlways;
proc sgplot data=sashelp.class;
  title "&rot";
  vbar name / response=weight nostatlabel;
  xaxis &rot;
run;
title;
```

Figure 10 X-Axis Value Rotation

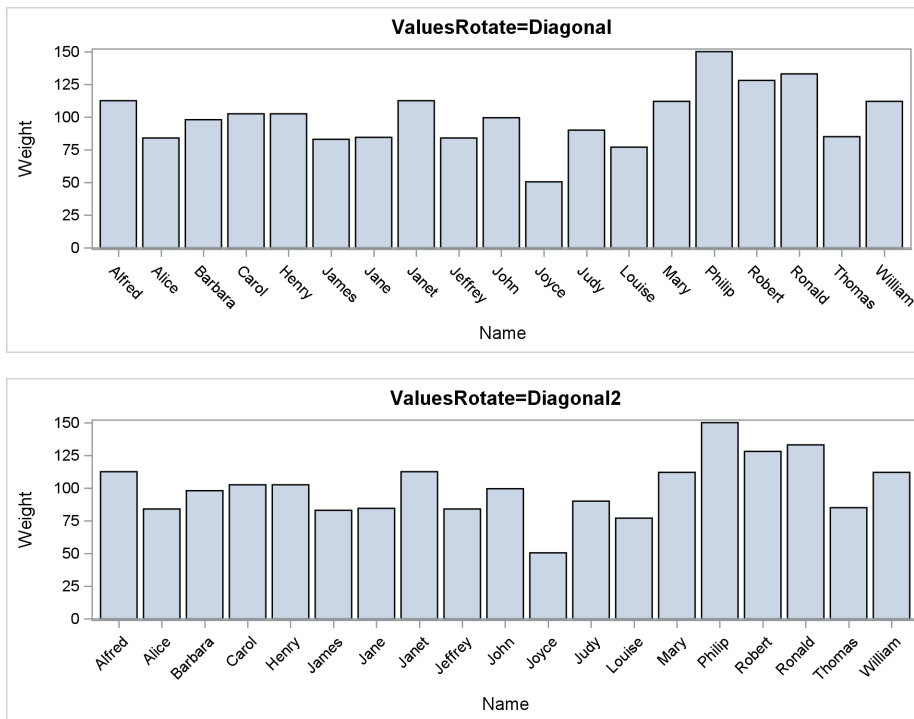
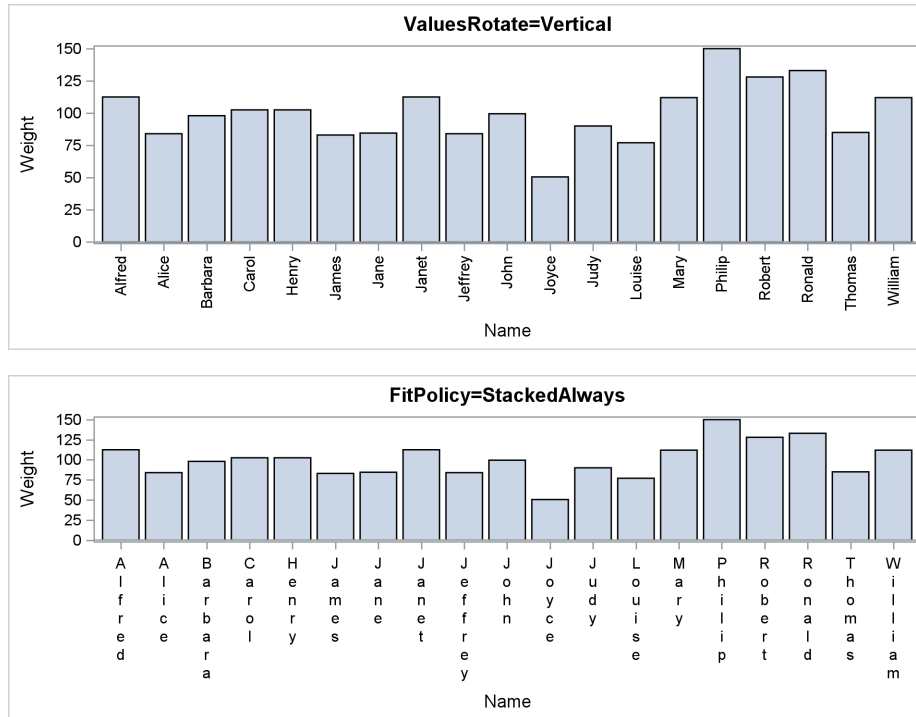


Figure 10 continued



LINEAR AND PROPORTIONAL BUBBLES

By default, bubble sizes are linearly scaled to correspond to the range of the sizes. The new PROPORTIONAL option enables you to scale the bubble size to correspond to the actual size value. The following steps illustrate. The resulting graphs are displayed in Figure 11.

```

data b;
  y = 'Proportional';
  do Size = 1 to 6;
    output;
  end;
  y = 'Linear';
  size = .;
  do S2 = 1 to 6;
    output;
  end;
run;

proc sgplot data=b noautolegend nocycleattrs;
  bubble y=y x=size size=size / proportional;
  bubble y=y x=s2 size=s2;
  yaxis display=(nolabel) offsetmin=0.2 offsetmax=0.2 reverse;
  xaxis offsetmin=0.05 offsetmax=0.05;
run;

data b2;
  set b;
  size = size + 5;
  s2 = s2 + 5;
run;

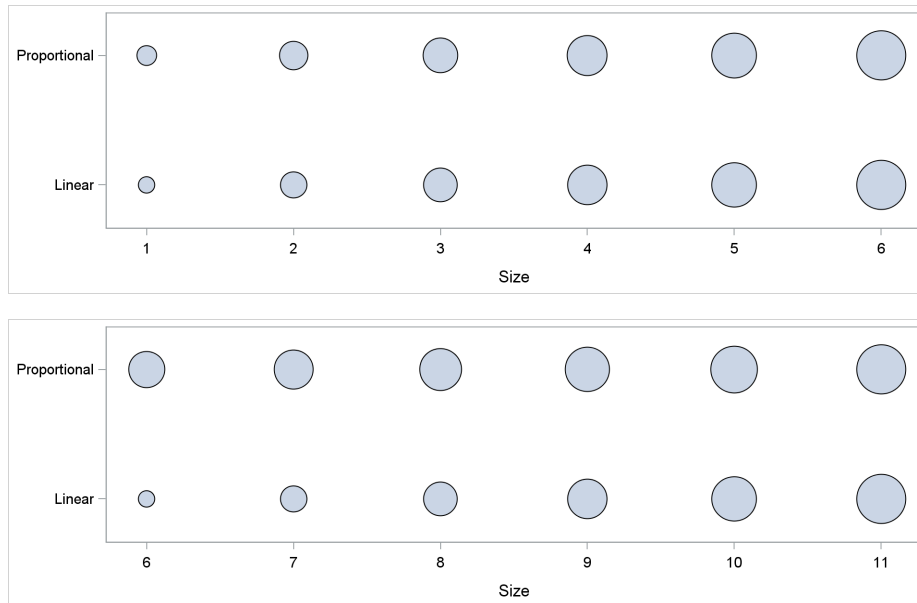
```

```

proc sgplot data=b2 noautolegend nocycleattrs;
  bubble y=y x=size size=size / proportional;
  bubble y=y x=s2 size=s2;
  yaxis display=(nolabel) offsetmin=0.2 offsetmax=0.2 reverse;
  xaxis offsetmin=0.05 offsetmax=0.05;
run;

```

Figure 11 Bubble Size Comparisons



Each graph is constructed from two BUBBLE statements. The data and graphs are constructed so that the Y axis displays the type of bubble scaling. The PROPORTIONAL option is specified for the six bubbles in the top row of each graph, and the default linear scaling is used for the six bubbles in the bottom row of each graph. In the first graph, the bubble sizes are similar when you use the two scaling methods. The proportional bubbles are slightly larger because their sizes are 1 to 6, whereas the linear bubble sizes are based on the relative values of 0 to 5. The linear bubbles in the first graph are identical to the corresponding bubbles in the second graph because the relative sizes are identical. The proportional bubbles in the second graph are larger than the linear bubbles. The sizes of the proportional bubbles in the second graph do not vary as much as the sizes of the proportional bubbles in the first graph, because the relative sizes do not vary as much ($1/5$ to $5/5$ in the first graph and $6/6$ to $11/6$ in the second graph).

SORTING A DISCRETE LEGEND

The SORTORDER= option enables you to sort the legend entries into ascending or descending order. You can now reverse the default sort order of the legend entries by using the SORTORDER=REVERSEAUTO option. The following steps create the graphs in [Figure 12](#):

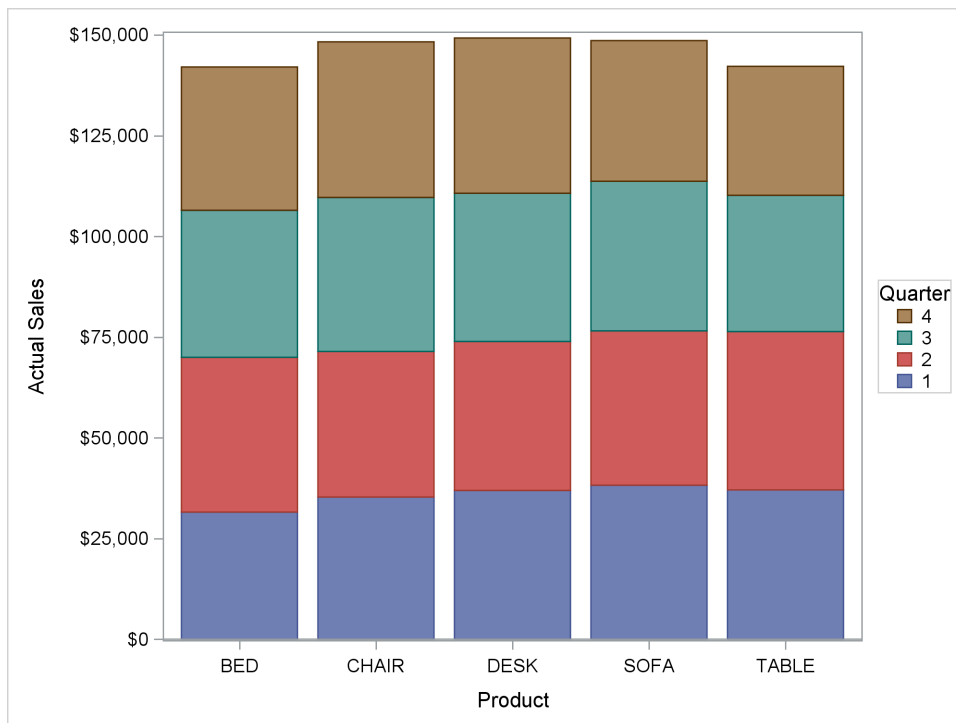
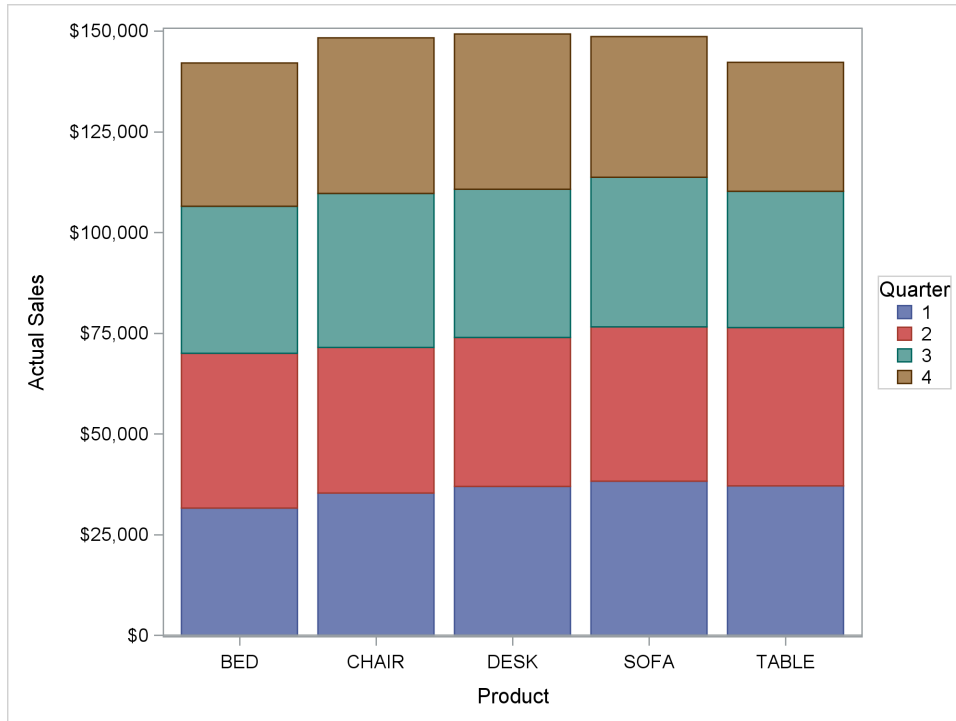
```

proc sgplot data=sashelp.prdsale;
  vbar product / response=actual group=quarter;
  keylegend / position=right;
run;

proc sgplot data=sashelp.prdsale;
  vbar product / response=actual group=quarter;
  keylegend / position=right sortorder=reverseauto;
run;

```

Figure 12 Discrete Legend



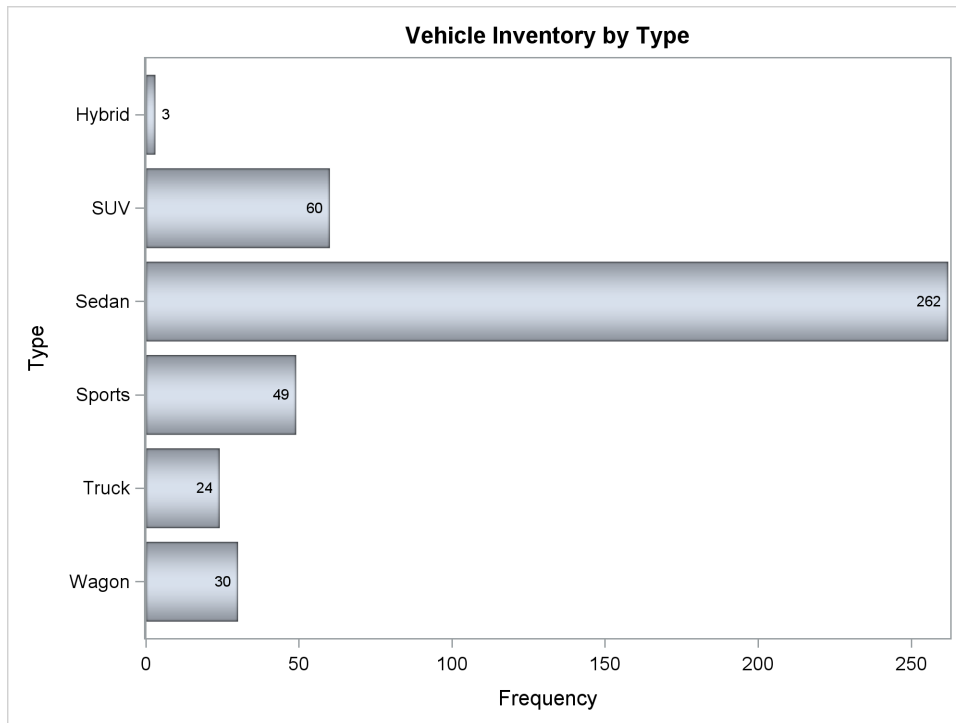
DATA LABEL FIT POLICY

A new option, `DATALABELFITPOLICY=INSIDEPREFERRED`, enables you to specify that the preferred location for a label is inside the bar. The following step creates the graph in Figure 13:

```
proc sgplot data=sashelp.cars;  
  title "Vehicle Inventory by Type";  
  hbar type / dataskin=pressed datalabel datalabelfitpolicy=insidepreferred;  
run;
```

All bars are labeled inside the bar except the top bar, Hybrid, which is too short to have enough room for the label.

Figure 13 Data Label Fit Policy



BOX PLOT CAP SCALE

You can control the width of the cap at the end of each whisker of a box plot by specifying the CAPSCALE= option. The following steps create the graphs in Figure 14:

```
proc sgplot data=sashelp.heart;
  title 'No CapScale';
  hbox cholesterol / category=weight_status;
run;

proc sgplot data=sashelp.heart;
  title 'CapScale=0.5';
  hbox cholesterol / category=weight_status capscale=0.5;
run;
```

Figure 14 Box Plot Cap Scale

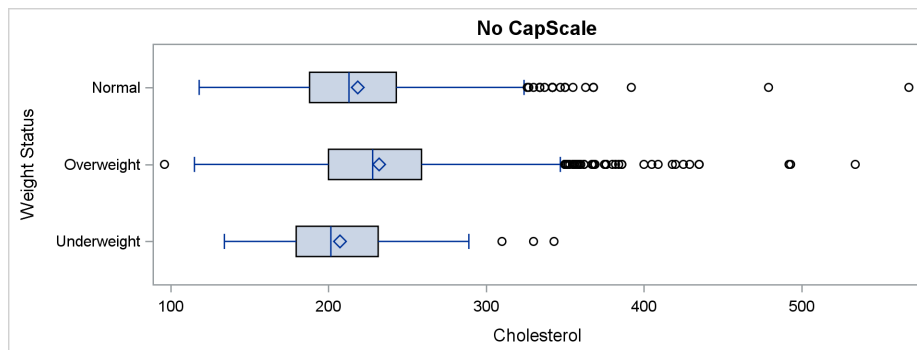
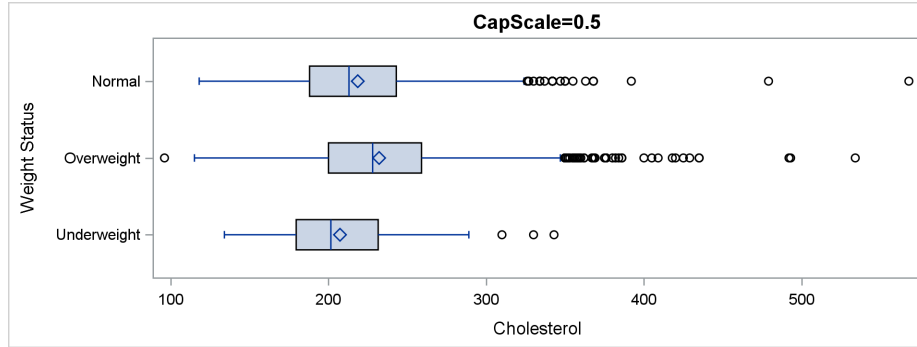


Figure 14 continued



In the second graph, the cap is 50% of the size of the default cap in the first graph.

DISPLAY BASELINE

You can use the DISPLAYBASELINE= option to display or suppress the baseline. The following steps create the graphs in Figure 15:

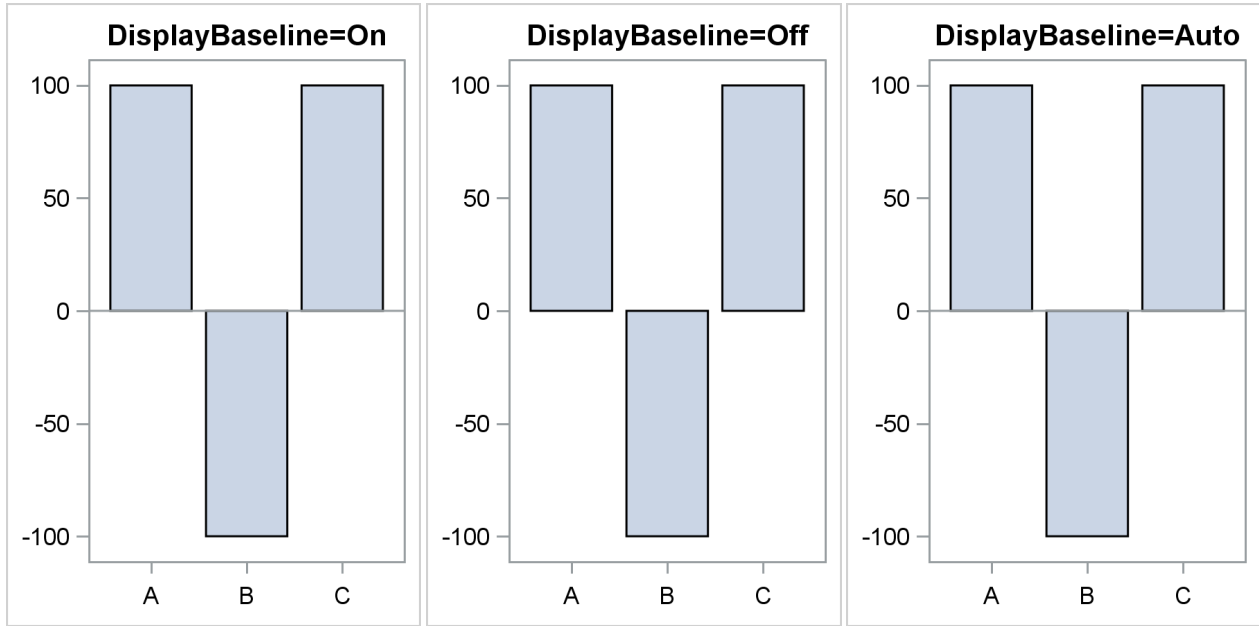
```
data r;
  input Y X $;
  datalines;
  100 A
 -100 B
  100 C
 ;

%let bl=DisplayBaseline=On;
proc sgplot;
  title "&bl";
  vbar x / response=y nostatlabel &bl;
  yaxis offsetmin=0.05 offsetmax=0.05 display=(nolabel);
  xaxis display=(nolabel);
run;

%let bl=DisplayBaseline=Off;
proc sgplot;
  title "&bl";
  vbar x / response=y nostatlabel &bl;
  yaxis offsetmin=0.05 offsetmax=0.05 display=(nolabel);
  xaxis display=(nolabel);
run;

%let bl=DisplayBaseline=Auto;
proc sgplot;
  title "&bl";
  vbar x / response=y nostatlabel &bl;
  yaxis offsetmin=0.05 offsetmax=0.05 display=(nolabel);
  xaxis display=(nolabel);
run;
```

Figure 15 Baseline Options

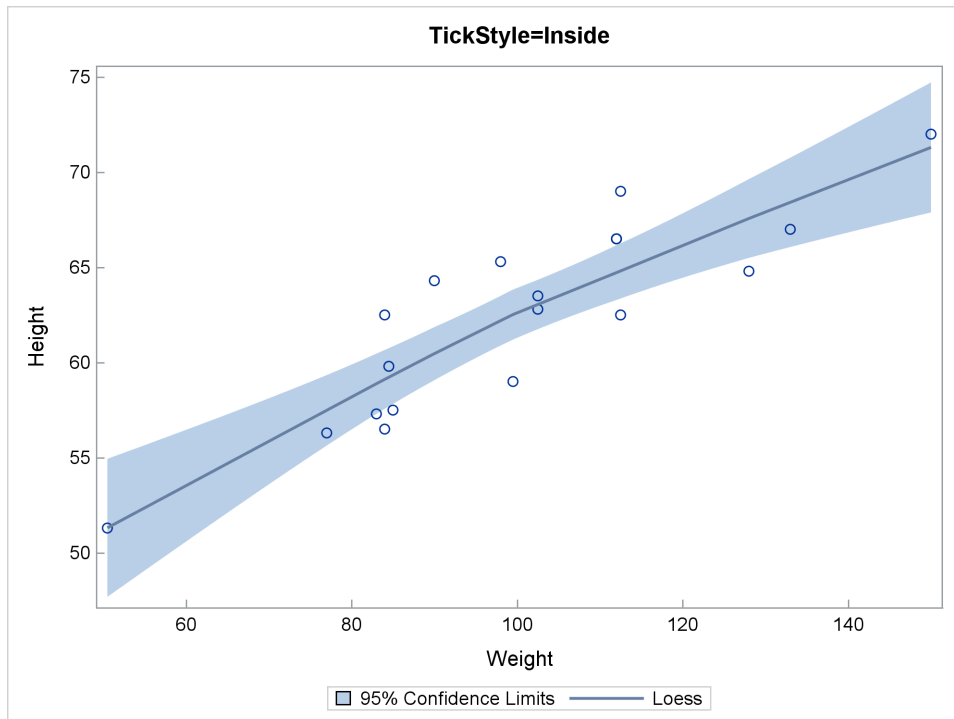


TICK STYLE

You can use the `TICKSTYLE=INSIDE` option to display ticks inside the axes. All previous graphs in this paper that display ticks display them outside the axes. The following steps create the graph in [Figure 16](#):

```
proc sgplot data=sashelp.class;  
  title "TickStyle=Inside";  
  xaxis tickstyle=inside;  
  yaxis tickstyle=inside;  
  loess x=weight y=height / clm;  
run;
```

Figure 16 Ticks Displayed inside the Axes

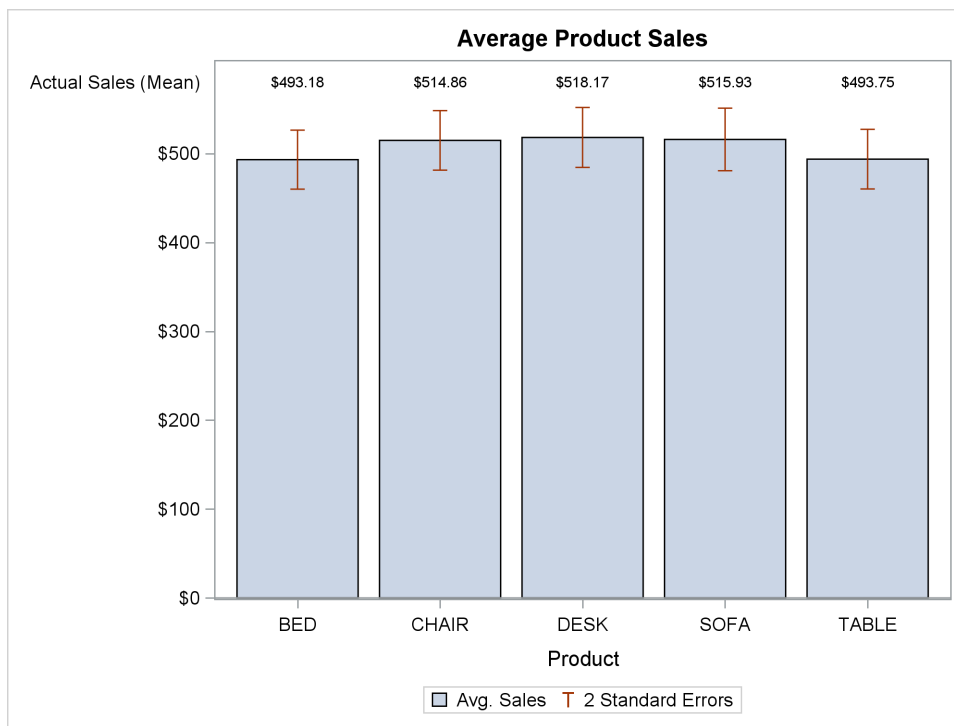


LEGEND ITEM

You can add items to the legend, even if they do not appear in any direct way in the graph. Usually, legend entries are automatically generated by SCATTER, SERIES, and other plotting statements, and you do not need to do anything special to display them. But there are times when you might want to add text or a symbol to the graph and add an entry in the legend to explain it. You might want to put something in the legend that perhaps does not appear in this particular data set but might appear in other comparable graphs that are made from other data. Or you might simply want to have more control over your legend. The following example uses the LEGENDITEM statement and Unicode to provide a legend for error bars. This step creates the graph in [Figure 17](#).

```
proc sgplot data=sashelp.prdsale;
  title "Average Product Sales";
  yaxis display=(nolabel) offsetmax=0.01;
  vbar product / response=actual stat=mean
    limitstat=stderr numstd=2 name="bar" legendlabel="Avg. Sales";
  xaxistable actual / stat=mean position=top pad=(top=8px)
    location=inside statlabel;
  legenditem name="limits" type=text / text="(*ESC*){unicode TAU_U}"
    textattrs=GraphError(size=10pt) label="2 Standard Errors";
  keylegend "bar" "limits";
run;
title;
```

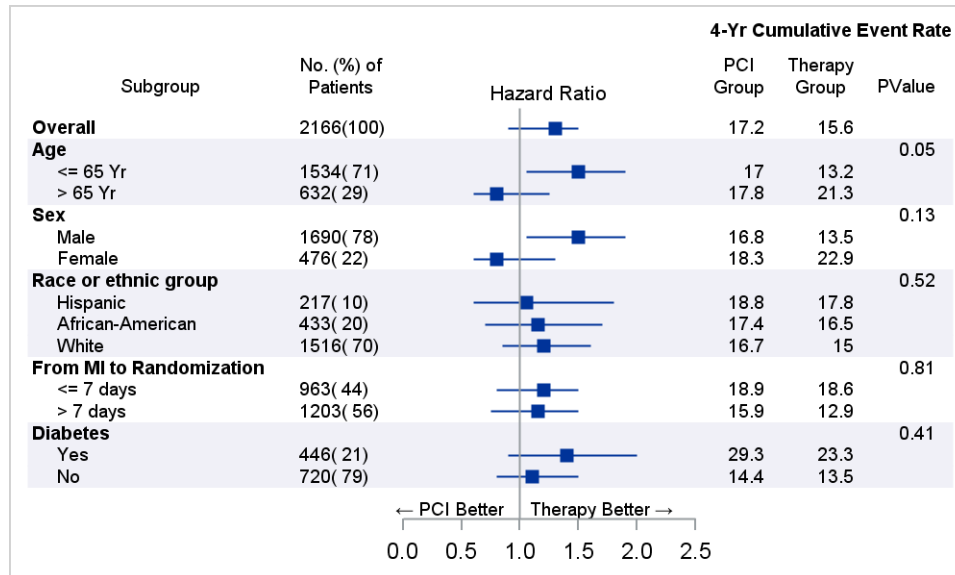
Figure 17 Legend Item



DISCRETE REFERENCE LINE THICKNESS

Figure 18 shows an example of how you might use discrete reference lines.

Figure 18 Discrete Reference Lines in a Forest Plot



The following statement creates gray shading for the rows indicated by the variable **Ref**:

```
refline ref / lineattrs=(color=cxf0f0f7) DiscreteThickness=1;
```

The variable **Ref** is constructed to highlight blocks of three, four, and three lines. You can highlight arbitrary groups of lines. The following steps illustrate the principles that are involved in the context of a simple example:

```
data x;
  do x = 0 to 11;
    y = x;
    ref = ifn(mod(round((x - 1) / 3), 2), y, .);
    output;
  end;
run;

proc print; run;

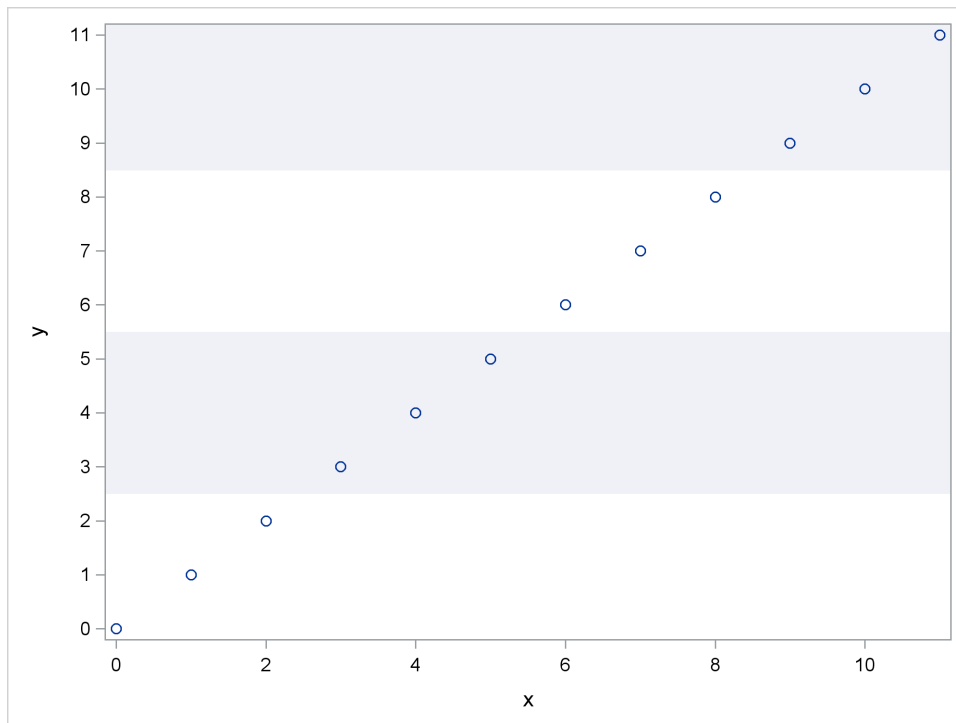
proc sgplot;
  refline ref / lineattrs=(color=cxf0f0f7) discretethickness=1;
  scatter y=y x=x;
  yaxis type=discrete;
run;
```

The data set is displayed in Figure 19, and the graph is displayed in Figure 20. The REFLINE statement highlights lines from 3 to 5 and from 9 to 11.

Figure 19 Data Set with a Reference Line Indicator Variable

Obs	x	y	ref
1	0	0	.
2	1	1	.
3	2	2	.
4	3	3	3
5	4	4	4
6	5	5	5
7	6	6	.
8	7	7	.
9	8	8	.
10	9	9	9
11	10	10	10
12	11	11	11

Figure 20 Discrete Reference Lines



CONCLUSION

ODS Graphics has a number of new features in the SAS 9.4M5 release. If you are using an older release, you might be performing some tasks the hard way. ODS Graphics has continued to evolve, and many great features have been added since the original release. For more information, see the section [“RECOMMENDED READING”](#) on page 18.

ACKNOWLEDGMENTS

The authors are grateful to Ed Huddlestone for his help in preparing this paper.

RECOMMENDED READING

For complete documentation about ODS and ODS Graphics, see SAS Output Delivery System: [User's Guide](#), SAS ODS Graphics: [Procedures Guide](#), SAS Graph Template Language: [User's Guide](#), and SAS Graph Template Language: [Reference](#).

For more documentation and examples, see Chapter 20, [Using the Output Delivery System](#); Chapter 21, [Statistical Graphics Using ODS](#); Chapter 22, [ODS Graphics Template Modification](#); and Chapter 23, [Customizing the Kaplan-Meier Survival Plot](#) in the [SAS/STAT User's Guide](#).

For a gentle and parallel introduction to PROC SGPLOT and the GTL, see the free web book [Basic ODS Graphics Examples](#). For more advanced topics, including an introduction to SG annotation, axis tables, and template modification using the CALL EXECUTE statement, see the free web book [Advanced ODS Graphics Examples](#). For tips, tricks, and the latest developments in ODS Graphics, see Sanjay Matange's [Graphically Speaking](#) blog and his [books](#). The [DO Loop](#) blog by Rick Wicklin is another great source of information about graphs.

A few of the many noteworthy features and tricks from releases that precede SAS 9.4M5 include the following:

- [style attributes](#)
- [range attribute maps](#)
- [highly customized graphs](#)
- [displaying a graph inside a graph](#)
- [draw statements](#)
- [3D waterfall chart](#)
- [bars with data labels](#)
- [processing data sets from PROC SGPLOT](#)
- [animating text](#)
- [long category labels](#)
- [turnip plot](#)
- [multiple blank categories on an axis](#)
- [high-low plot](#)
- [examining and processing templates](#)
- [transparent margins](#)
- [waterfall plot](#)
- [lollipop charts](#)
- [vector plots and labels](#)
- [customizing the Kaplan-Meier plot](#)
- [GTL expressions](#)
- [consistent ordering of graph components](#)
- [butterfly plot](#)
- [stacked band plot](#)
- [discrete attribute maps](#)
- [applying SG annotation to BY groups](#)
- [reformatting correlation matrices](#)
- [axis tables](#)
- [waterfall and swimmer plot](#)
- [series plot with varying attributes](#)
- [graph template and data object in PROC SGPLOT](#)
- [POSITION= option to scatter plots](#)
- [spark table](#)
- [documents, dynamics, and data objects](#)
- [segmented discrete axes](#)
- [axis tables and special characters](#)
- [bar charts with color response](#)
- [long category values](#)
- [axis options](#)
- [equated axes and aspect ratio](#)
- [image backgrounds](#)
- [scatter with mean value](#)
- [multipage adverse event reports](#)
- [category highlighting](#)
- [funnel graph](#)
- [zipper plot](#)
- [jittering](#)

EXAMPLE CODE

[Double-click here](#) for all example code. (The link is supported in Adobe Reader but not in many browsers.) Each example (except for the PROC SGMAP example) is complete.

CONTACT INFORMATION

Warren F. Kuhfeld
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
(919) 531-7922
Warren.Kuhfeld@sas.com

Dan Heath
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
(919) 531-6657
Dan.Heath@sas.com