Paper 095-2011

The Power to Plot: Three Ways to Enhance SG Graphical Outputs

Xiangxiang Meng, University of Cincinnati, Cincinnati, OH

ABSTRACT

In SAS® 9.2, a set of statistical graphics (SG) procedures, including the SGPLOT, SGPANEL, SGCATTER, and SGRENDER procedures, are introduced to create high-quality template-based graphs using the Graph Template Language (GTL). This paper discusses three different ways to enhance the statistical graphics output generated by SG procedures. The cooperation of SG procedures, ODS styles and GTL language is the focus of this paper, and the TEMPLATE procedure is highly involved to control and modify ODS style or GTL program.

INTRODUCTION

In SAS® 9.2, a set of statistical graphics (SG) procedures, including the SGPLOT, SGPANEL, SGCATTER, and SGRENDER procedures, are introduced to create high-quality template-based graphs using the Graph Template Language (GTL). Compared with graphs produced by traditional SAS/GRAPH® procedures, the graphs created by SG procedures are saved in standard image formats (BMP, PNG, etc.) instead of in the SAS® graphics catalog, and displayed in the Result Window instead of in the Graphics Window. Moreover, some SAS/GRAPH® global statements, such as the GOPTION, SYMBOL and AXIS statements, no longer affect SG graphical output. Their roles have been replaced by more powerful and flexible options within the SG procedures, or attributes in the ODS graphical templates.

This paper focuses on how to customize the SG graphical output in the following ways 1) using options and statements within SG procedures; 2) defining or modifying graphical attributes of an ODS graphical template in the TEMPLATE procedure, and then applying the modified ODS style onto the SG output; 3) modifying the GTL code generated by SG procedures and then re-generate the graph by using PROC SGRENDER. Basic introduction to the SG procedures and the GTL framework is included but the focus of this paper is more on how the SG procedures interact with other procedures and modules within the whole ODS Graphics framework. The author refers to some recent publications in SAS® Global Forum proceedings for introductions to the SG procedures. See *Delwiche and Slaugther 2009* for PROC SGPLOT, *Kincaid 2010* for PROC SGPANEL, *Meng 2010* for SGSCATTER, and *Schwartz 2009* for using SG procedures for clinical trial reporting.

Two datasets from SASHELP.CARS and the ODS style DEFAULT are used throughout the examples in this paper. The DEFAULT style is by default the ODS style used by the SG procedures in SAS® 9.2.

```
ods html style=default;
data cars_small;                                        /* For Scatter Plots*/
  set sashelp.cars;
  where make in ('Jeep' 'Chevrolet' 'Ford' 'Chrysler');
data cars_large;                                        /* For Bar Charts*/
  set sashelp.cars;
  n1=(cylinders le 4)*-1;
  n2=(cylinders gt 4)*1;
  label n1='<= 4 Cylinders' n2='>4 Cylinders';
run;
```

Variables In Use	Description
MSRP	Manufacturer's suggested retail price
Type	The type of a car. (Hybrid, SUV, Sedan, Sports, Truck Wagon)
Origin	Takes value "Asia", "America", or "Europe"
Invoice	Invoice price of a car.
Weight	The weight of a car
Length	The length of a car.
n1	Indicator for (Cylinders < = 4). Takes value -1 (Yes) and 0 (No).
n2	Indicator for (Cylinders > 4). Takes value 1 (Yes) and 0 (No).

Table 1. List of Variables used in examples

GETTING TO KNOW THE SG PROCEDURES

The basic elements underlying the SG procedures are *Graphs*, *Cells*, and *Plots*. A *Graph* consists of one or more *Cells*, and a *Cell* consists of one or more overlaid *Plots*. Each *Plot* is controlled by a statement in the SG procedure. A variety of *Plot*s can be produced in this statistical graphical framework, such as scatter plots, bar charts and regression fits. More specifically, PROC SGPLOT is designed to generate single-cell graphs. PROC SGPANEL and PROC SGSCATTER are designed to generate multi-cell graphs which are also called panels. The first example gives an introduction to the syntax of PROC SGPLOT and PROC SGPANEL.

```
* [Example 1] simple examples of SGPLOT and SGPANEL;
* scatter plot of weight vs. MSRP with quadratic fits and confid. intervals;
* and scatter plot of weight vs. invoice price;
proc sgplot data=cars_small;
  reg x=weight y=MSRP / degree=2 clm;
  scatter x=weight y=invoice;
run; quit;

* A panel of three cells for Asian, European, and American cars.
* Each cell is a butterfly plot containing two bar charts;
proc sgpanel data=cars_large;
  panelby origin / rows=1 columns=3;
  hbar type / response=n1;
  hbar type / response=n2;
run;
```

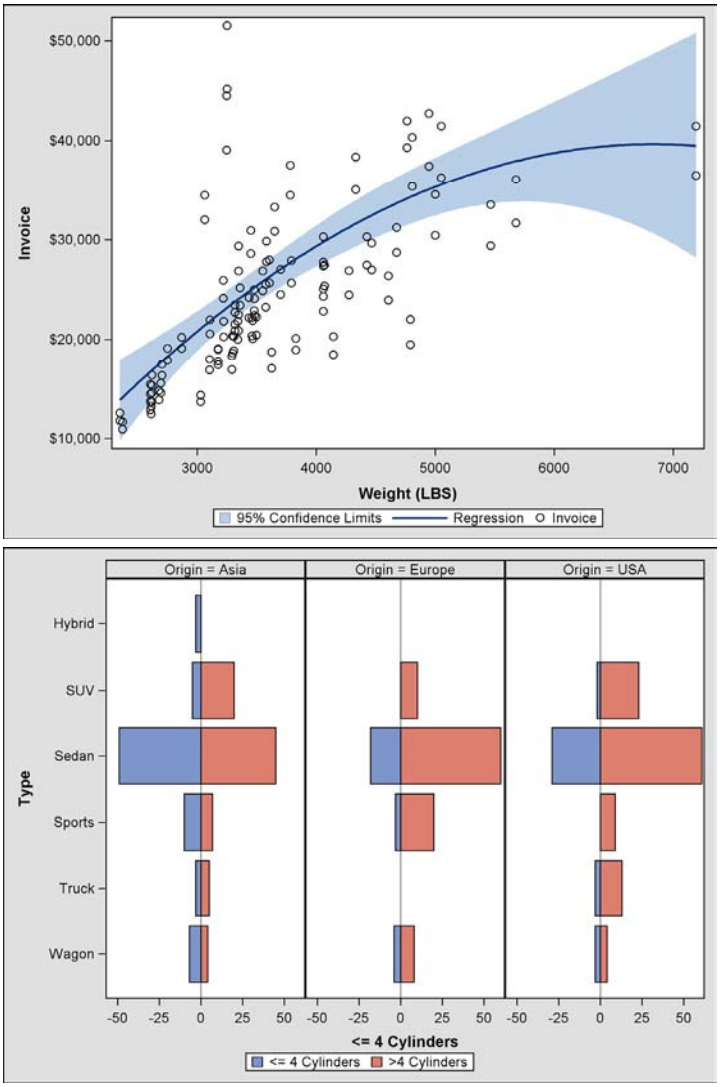
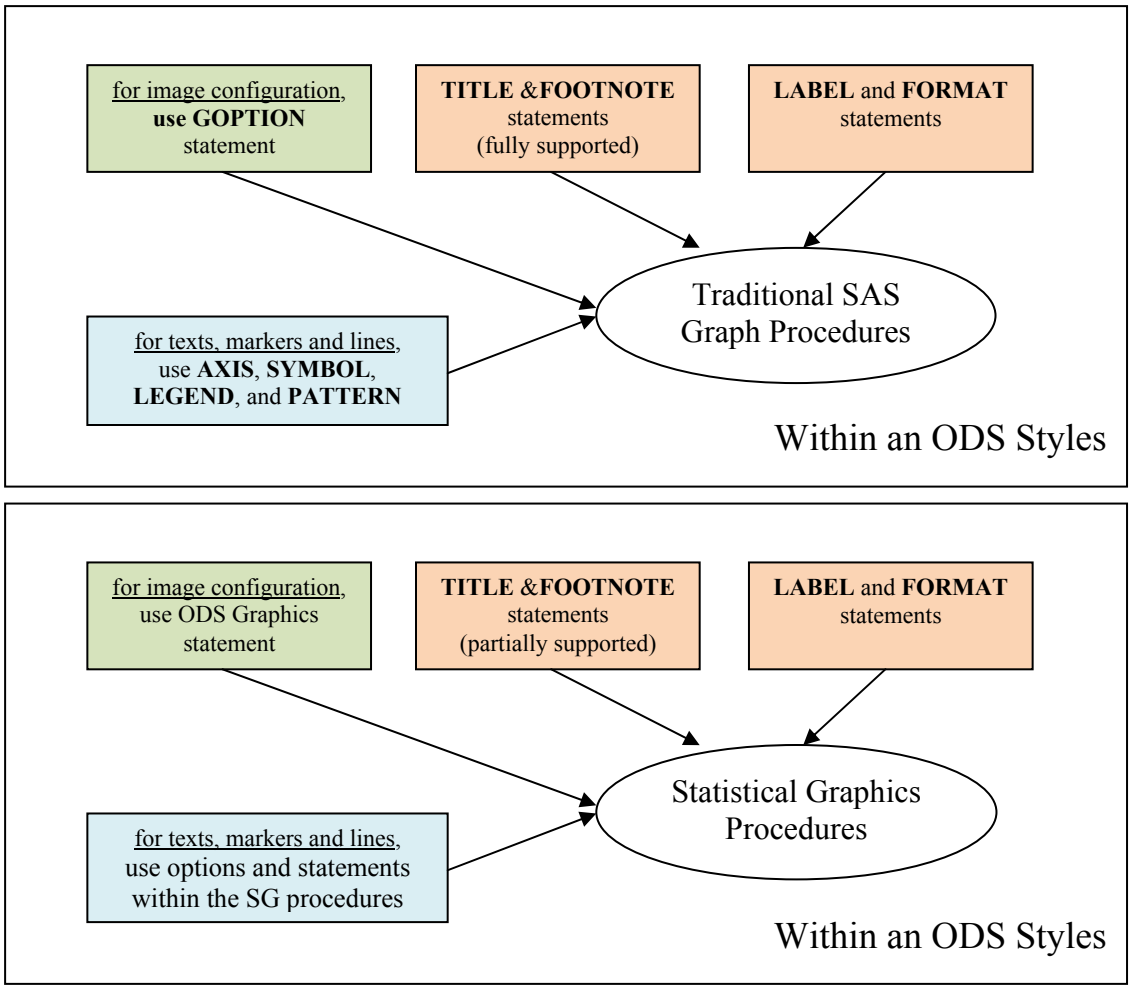


Figure 1. Introduction to SG graphs. Top: single-cell graph with three overlaid plots generated by PROC SGPLOT. Bottom: multi-cell graph with two bar charts generated by PROC SGPANEL.

HOW TO EHHANCE SG GRAPHICS?

The first example shows the easy syntax of the SG procedures for creating sophisticated statistical graphs (14 words in the first PROC and 17 words in the second PROC!). However, both graphs need to be further customized. For example, the two scatter plots cannot be distinguished since the same symbols and colors are applied to two set of data points (Weight vs. MSRP, Weight vs. Invoice). Adjusting the position, label, and font size of the axes are also necessary. As part of the new ODS graphics framework released with SAS® 9.2, the SG procedures cooperate with SAS/GRAPH® and SAS ODS in a very different way from other SAS/GRAPH® procedures such as the GPLOT and GCHART procedures. The following two charts show the difference:



SOLUTION I, USING BUILT-IN OPTIONS & STATEMENTS

In the new SG setting, the role of SAS/GRAPH® global statements, including AXIS, SYMBOL, LEGEND, PATTERN and NOTE, has been replaced by options or statements within each SG procedure. This section highlights how to enhance the two graphs in the first example step by step using built-in options and statements. First, the alternatives to SYMBOL and PATTERN statements are three options: LINEATTRS=, MARKERATTRS, and FILLATTRS for the attributes of lines, markers and fills.

```
* [Example 2.1] enhanced graphs with LINEATTRS=, MARKERATTRS=;
proc sgplot data=cars_small;
  reg x=weight y=MSRP
    / degree=2 clm
      markerattrs=(symbol=circleFilled size=2pct color=CX16A629)
      lineattrs=(color=CXFF6060 pattern=1 thickness=1pct)
;
  scatter x=weight y=invoice
    / markerattrs=(symbol=star size=2pct color=CX13478C)
;
run; quit;
```

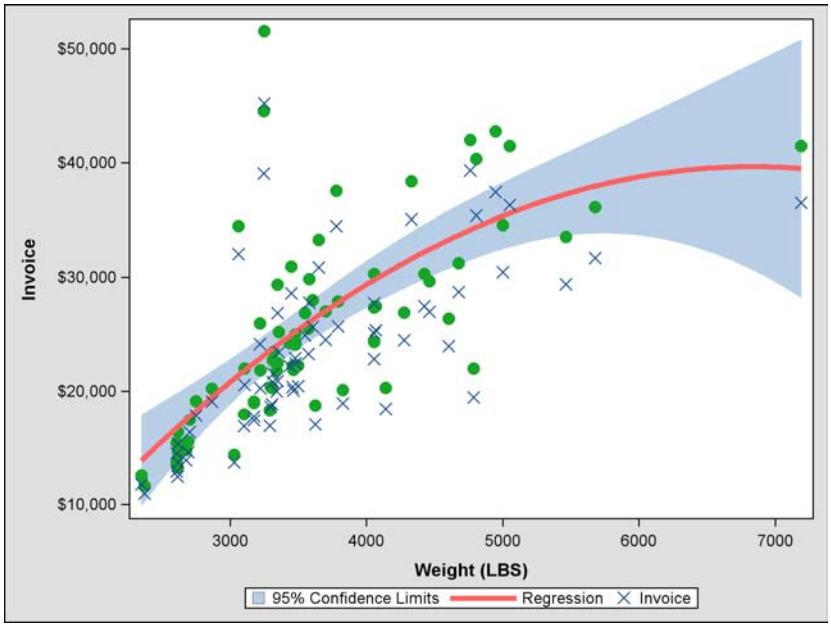


Figure 2. Enhanced Scatter Plots by LINEATTRS= and MARKERATTRS= Options

The LINEATTRS= and MARKERATTRS= options above only affect the graphical properties of the plot created by their hosting statement. Since different SG statements create different types of statistical graphs, each of these three attribute options is not compatible with all SG statements. For example, there is no LINEATTRS= or MARKERATTRS= allowed in VBAR and HBAR statement for bar charts. Furthermore, besides the graph attributes such as COLOR, SIZE, and THICKNESS used in Example 2.1, a general style can also be assigned. Example 2.2 shows how to assign GRAPHDATA style to adjust the fill colors of the bars charts.

```
* [Example 2.2] enhanced graphs with FILLATTRS= and new GRPAHDATA style;
* NOTE: GRAPHDATA5 and GRAPHDATA3 are defined in ODS style DEFAULT.;
proc sgpanel data=cars_large;
  panelby origin / rows=1 columns=3;
  hbar type / response=n1 fillattrs=graphdata5
;
  hbar type / response=n2 fillattrs=graphdata3
;
run;
```

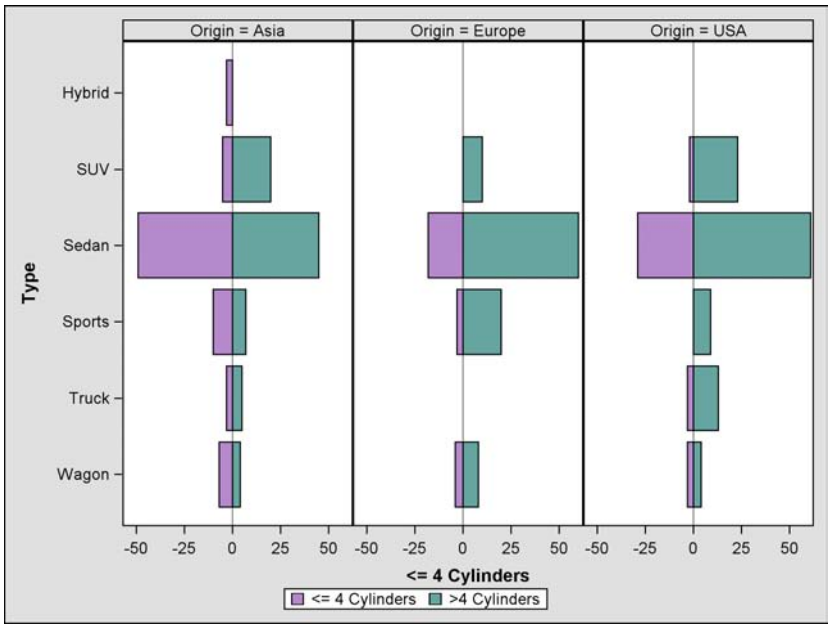


Figure 3. Changing Bar Colors by FILLATTRS= Option and GRAPHDATA Styles

Next, let us adjust the legends and axes in both plots:

```
* [Example 3] adjust axes and legends;
* Note: a duplicate scatter plot of MSRP vs. Weight;
* is included for better legend adjustment;
proc sgplot data=cars_small;
  reg x=weight y=MSRP
    / degree=2 clm
      markerattrs=(symbol=circleFilled size=2pct color= CX16A629)
      lineattrs=(color= CXFF6060 pattern=1 thickness=1pct)
      name='sg1'
;
  scatter x=weight y=invoice
    / markerattrs=(symbol=x size=2pct color= CX13478C)
      name='sg2'
      legendlabel='Invoice Price'
;
  scatter x=weight y=MSRP
    / markerattrs=(symbol=circleFilled size=2pct color= CX16A629)
      name='sg3'
      legendlabel='MSRP w. 95% CLM'
;
  xaxis label='Weight of Car (Lbs)';
  yaxis display=(nolabel);
  keylegend 'sg2' 'sg3' / location=inside position=top down=2;
run; quit;

proc sgpanel data=cars_large;
    panelby origin / rows=1 columns=3 spacing=10 ;
    hbar type / response=n1 fillattrs=graphdata5;
    hbar type / response=n2 fillattrs=graphdata3;
    rowaxis label=' ';
    colaxis label='Number of Car Models' values=(-60 to 60 by 20) grid;
run; quit;
```

The statements for axis adjustment in PROC SGPLOT and PROC SGPANEL are different. For SGPLOT, use XAXIS and YAXIS. For SGPANEL, use ROWAXIS and COLAXIS. Another new feature is that the legend appearance must be controlled by both the KEYLEGENG statement and some options in each plot statements. In the above example, NAME= options assign to each plot statement a "plot ID", which is then used in the KEYLEGEND statement to identity which plots to be included in the legend (Only sg2 and sg3 in this example).

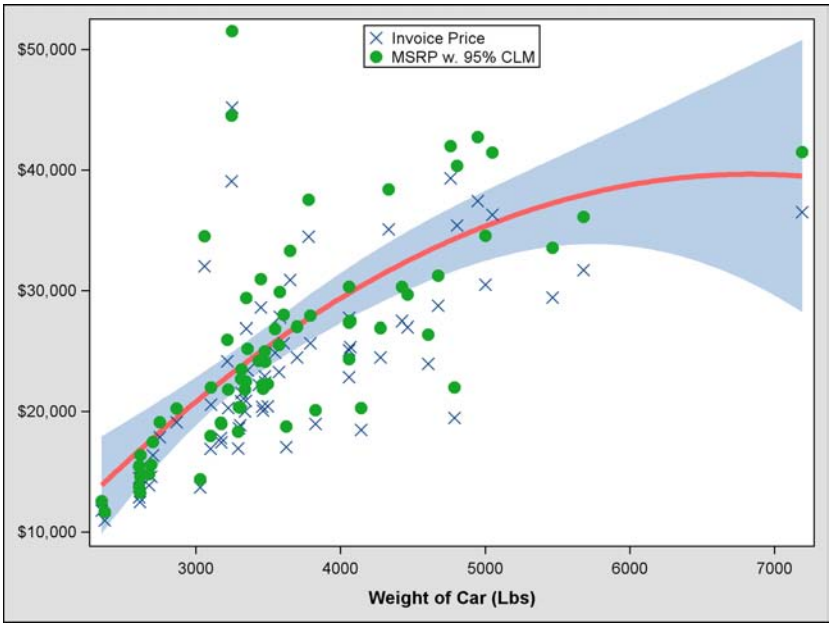


Figure 4. Scatter Plots with Adjusted Axes and Legends

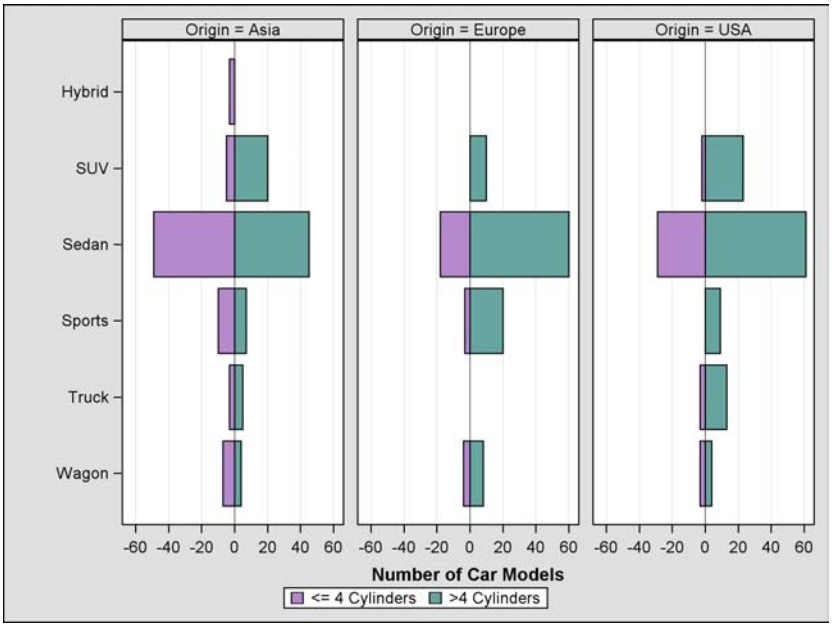


Figure 5. Bar Charts with Adjusted Axes and Legends

The following table summarize s the function and compatibility of the options and statements discussed in this section:

To Control...	Statement / Option	Available in PROC:		
		SGPLOT	SGPANEL	SGSCATTER
Markers	MARKERATTRS=	Y	Y	Y
Lines	LINEATTRS=	Y	Y	N
Fills	FILLATTRS=	Y	Y	N
Axes	XAXIS,YAXIS,X2AXIS,Y2AXIS	Y	N	N
	COLAXIS, ROWAXIS	N	Y	N
Legends	KEYLEGEND (statement)	Y	Y	N
	NAME=, LEGENDLABEL= (options)	Y	Y	N
	LEGEND=, NOLEGEND (options)	N	N	Y
Reference Lines	REFLINE	Y	Y	N
Titles, Footnotes, Formats, Labels	Traditional global statements	Y	Y	Y

Table 2. Compatibility of Options and Statements

Note: many other plot-specific options are also available. For example, OUTLIER and NOUUTLIER options are available in the HBOX and VBOX statements to control the display of outliers in a box plot. The author refers to the manual of statistical graphics procedures for more details.

SOLUTION 2, COOPERATING WITH ODS STYLE, ODS GRAPHICS, AND PROC TEMPLATE

In the above section, we go through some common options and statements in the SG procedures for adjusting the graphical output as a replacement of the global statements in SAS/GRPAH. However, this adjustment is limited by the definitions of these options and statement. In some situations not all the graphical elements you want to customize is available as an option or sub-option in SG procedures. For example, there are no direct options in XAXIS, YAXIS, COLAXIS, and ROWAXIS statement to control the size and type of fonts for axis label. Even worse, no axis-control statement is supported by PROC SGSCATTER. What should we do to get larger fonts in the Figure 4 and Figure 5?

The solution is ODS Style.

In SAS® 9.2, SAS/GRAPH starts to use ODS styles by default to control graphical outputs. This type of graphics is usually referred to as “template-based" graphics in order to distinguish from the traditional "driver-based" graphics which is generated by a default or user-specified device (For instance, GOPTIONS DEVICE=PNG) and displayed in

the Graphics window. New in SAS® 9.2, all the graphics produced by the SG procedures are template-based and by default displayed in the Result window. In addition, a variety of new ODS styles are introduced in SAS 9.2, which provides a large number of choices for displaying statistical graphics in different industry or domain standards yet without sophisticated coding. The next example simply applies three different ODS styles to the same output and changes the output dramatically.

```
*[Example 4] Changing ODS Styles;
%macro ChangeStyle(Style);
  ods html style=&style;
  proc sgplot data=cars_large;
    hbar type / group=origin;
  run; quit;
%mend;
%ChangeStyle(Default);
%ChangeStyle(Solutions);
%ChangeStyle(Astronomy);
```

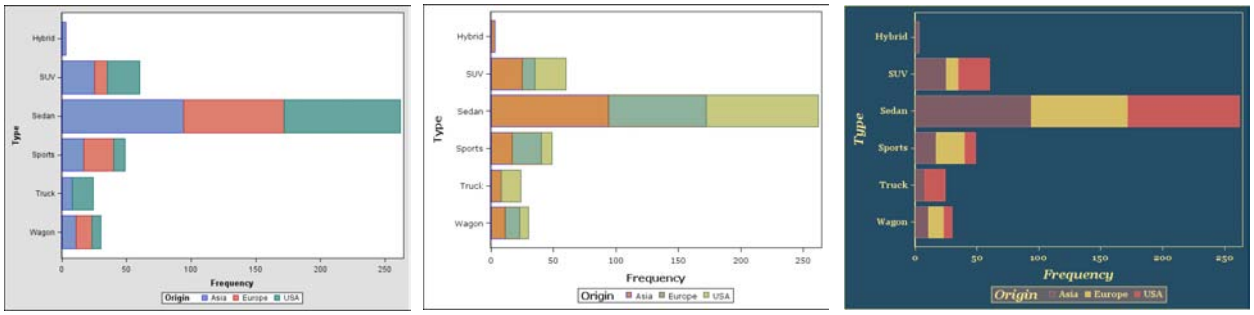


Figure 6. The Same Bar Charts with Different ODS Styles

PROC TEMPLATE is your personal assistant in SAS® for managing ODS styles. The following example shows how to list all available styles in your version of SAS®, print the complete definition of a style, and create a style by inheritance.

```
proc template;
  list styles;
  source styles.default;
  define style Styles.Custom;
    parent = Styles.Default;
  end;
run;
```

In general, graphical outputs are controlled by the style elements with prefix "Graph". For instance, GraphConfidence is for the confidence area, and GraphLegendBackground is for the background of the legend. The objected-oriented nature of ODS style definition makes it very easy and flexible to add or adjust graphical elements in a ODS style:

```
*[Example 4.1] Simple examples for applying modified ODS style;
proc template;
  define style styles.LasVegas1;
    parent = styles.default;
    style GraphConfidence from GraphConfidence  / color=CXE8D898;
    style GraphLegendBackground                 / transparency=1;
  end;
run;

ods html style=LasVegas1;
proc sgplot data=cars_small;
    reg x=weight y=MSRP / degree=2 clm;
    scatter x=weight y=invoice;
run; quit;
```

This example changes the color of the confidence area and makes the legend transparent. Note that the second class `GraphLegendBackground` is created instead of from inheritance since no such class has been defined in the ODS style DEFAULT. The results are shown below.

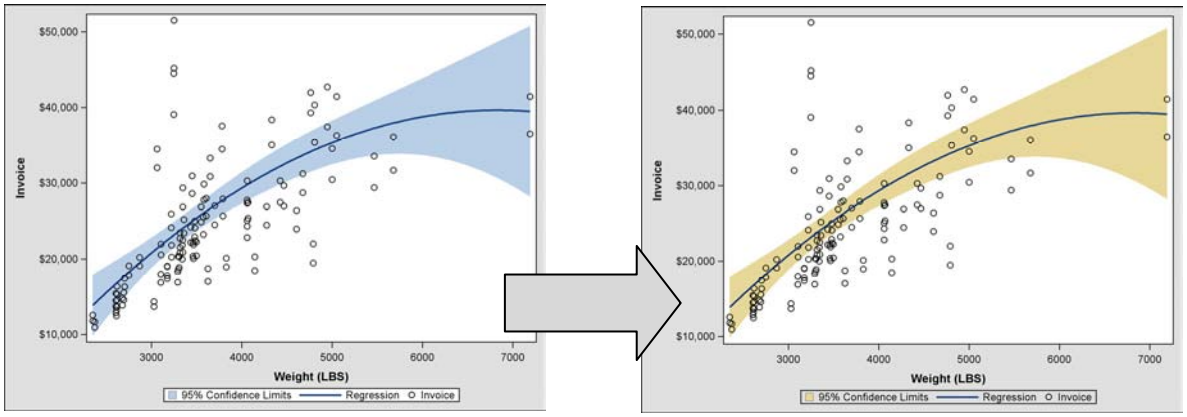


Figure 7. Changing Color of the Conf. Interval and Transparency of the Legend Block

Now, let us complete the graph adjustment by modifying more attributes:

```
*[Example 5.1] Modify and Apply ODS Style to SG graphics;
proc template;
  define style styles.LasVegas2;
    parent = styles.default;
    style GraphConfidence from GraphConfidence  / color= CXE8D898;
    style GraphLegendBackground                 / transparency=1;
    style GraphValueText from GraphValueText / font=('Times New Roman', 10pt, Bold);
    style GraphLabelText from GraphLabelText / font=('Times New Roman', 10pt, Bold);
    style GraphBorderLines from GraphBorderLines / LineThickness=0;
    style GraphWalls from GraphWalls             / color= CXC0C0C0;
    style GraphBackground                        / transparency=1;
  end;
run;
ods html style=LasVegas2;
proc sgplot data=cars_small;
  /* copy the code from  example 3 to here */
run; quit;
```

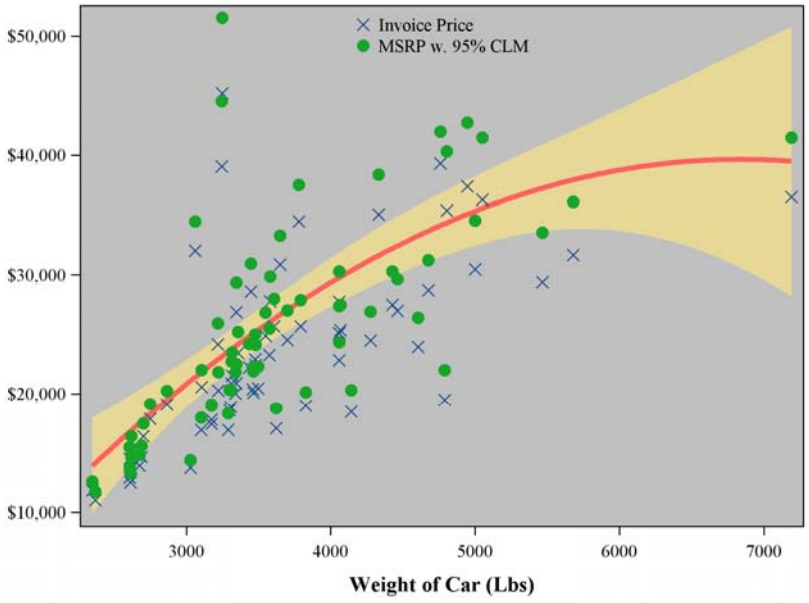


Figure 8. Modifying and Applying ODS Style to customize a graph

Similar to the role of GOPTION statement in controlling driver-based graphics, the ODS GRAPHICS statement cooperates with SG procedures for image configuration such as size, format and resolution. The next example uses the ODS GRAPHICS statement and a PROC TEMPLATE to enhance the bar charts in Figure 5. A new style LasVegas3 is inherited directly from the style LasVegas2 to keep the two graphics in the same fashion and makes the code more efficient.

```
*[Example 5.2] ODS Graphics Statement+ ODS Style;
proc template;
  define style styles.LasVegas3;
    parent = styles.LasVegas2;
    style GraphData1 from GraphData1 / color= CX16A629;
    style GraphData2 from GraphData2 / color= CXFF6060;
    style GraphAxisLines from GraphAxisLines / TickDisplay='inside';
    style GraphGridLines from GraphGridLines / LineThickness=1px
                                               contrastColor=CX595753;
  end;
run;

ods graphics / reset=all imagename='example52' imagefmt=png
               width=12in  height=6in border=off;
ods html gpath='C:\' style=LasVegas3;
proc sgpanel data=cars_large;
    panelby origin / rows=1 columns=3 spacing=10;
    hbar type / response=n1 fillattrs=graphdata1;
    hbar type / response=n2 fillattrs=graphdata2;
    rowaxis label=' ';
    colaxis label='Number of Car Models' values=(-60 to 60 by 20) grid;
run;
```

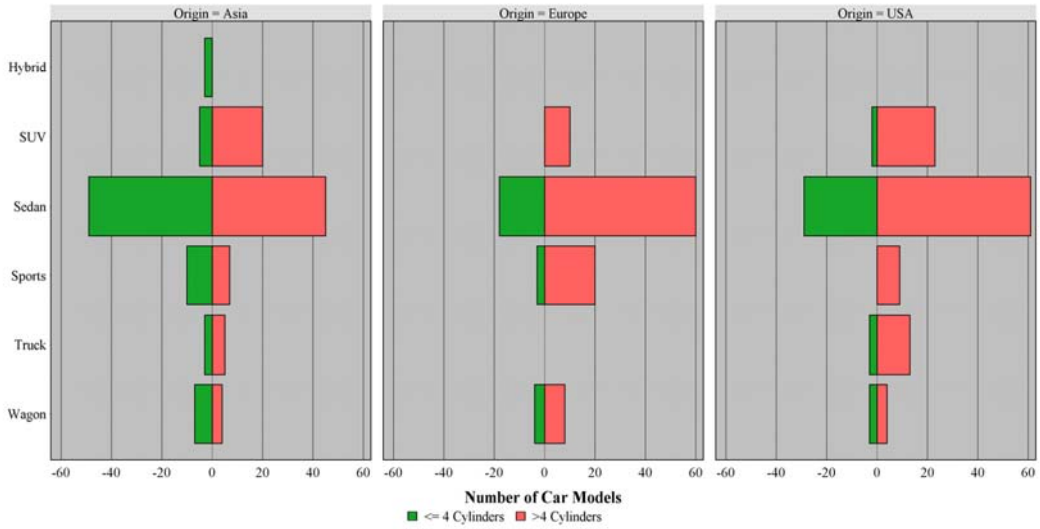


Figure 9. Using ODS GRAPHICS Statement and ODS STYLE for SG graphics

SOLUTION III, DIRECTLY WORKING ON GTL CODE

The ODS Graphics Template Language (GTL) is the foundation underlying the SG procedures. The graphic output of SG procedure is generated by compiled GTL code. In most situations users only need to program within the SG framework without knowing any detail on GTL language. However, the SG procedures also provide an option TMPLOUT= to save the GTL code employed in the current procedure for further use. For example, to see the GTL program employed in Example 5.1:

```
ods html style=LasVegas2;
proc sgplot data=cars tmplout='C:\GTL_1.sas';
  /* copy the code from example 3 to here */
run; quit;
```

This is the GTL code saved in the file GTL_1.sas:

```
proc template;
  define statgraph my_sgplot;
  begingraph;
    layout overlay / xaxisopts=(Label="Weight of Car (Lbs)" type=linear)
                     yaxisopts=(display=(ticks tickvalues line) type=linear);
      ModelBand "G4BMLFOQ"
        / Name="MODELBAND" LegendLabel="95% Confidence Limits";
      ScatterPlot X=Weight Y=MSRP
```

```
        / primary=true Markerattrs=(Color=CX16A629 Symbol=CIRCLEFILLED Size=2%);
      RegressionPlot X=Weight Y=MSRP
        / Lineattrs=(Color=CXFF6060 Pattern=1 Thickness=1%) NAME="sg1"
          LegendLabel="Regression" clm="G4BMLFOQ" Degree=2;
      ScatterPlot X=Weight Y=Invoice
        / primary=true Markerattrs=(Color=CX13478C Symbol=X Size=2%)
          LegendLabel="Invoice Price" NAME="sg2";
      ScatterPlot X=Weight Y=MSRP
        / Markerattrs=(Color=CX16A629 Symbol=CIRCLEFILLED Size=2%)
          LegendLabel="MSRP w. 95% CLM" NAME="sg3";
      DiscreteLegend "sg2" "sg3"
        / Location=Inside down=2 order=columnMajor valign=top;
    endlayout;
  endgraph;
  end;
run;
```

Similar to ODS styles, ODS GTL programs are managed by PROC TEMPLATE. Modifying the GTL code output by SG procedures, we can add features that are not supported by SG procedures. The next example uses multiple background colors to indentify light and heavy cars. This feature is controlled by the BLOCKPLOT definition in GTL.

```
* [Example 6.1] Modify and Re-render the GTL code;
proc template;
  define statgraph my_sgplot;
  begingraph;
    layout /*...copy from the above example...*/
           /*................................*/
      blockplot x=weight block=weight_cat
        /   datatransparency=0.6              /* Block transparency*/
            display=(fill values)             /* What to be displayed*/
            valueVAlign=top                   /* values of WEIGHT_CAT at top-center*/
            valueHAlign=center                /***see above***/
            filltype=alternate                /* Use two alternative fills*/
            fillattrs=(color= CXC0C0C0)       /* Control the fill of block 1,3,5...*/
            altfillattrs=(color=white) ;      /* Control the fill of block 2,4,6...*/
    endlayout;
  endgraph;
  end;
run;
ods graphics / reset=all imagename='example61' imagefmt=png
               width=12in height=6in border=off ;
ods html gpath='C:\' style=LasVegas2;
proc sgrender data=cars_small template=my_first_GTL;
run; quit;
```

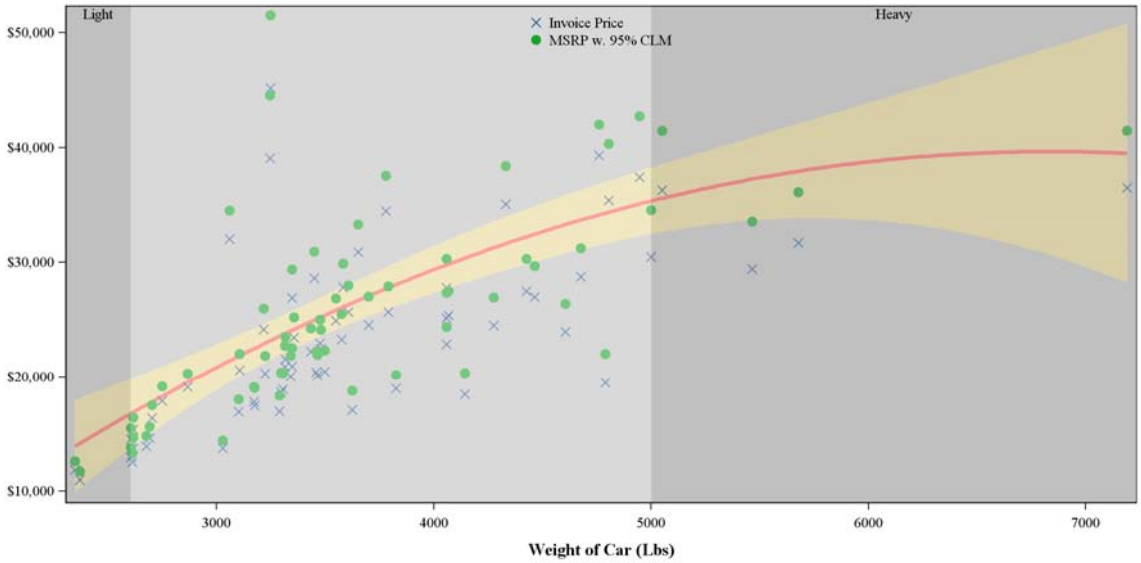


Figure 10. Scatter Plots with Alternative Background created by modifying GTL code

It is also very convenient to create SG graphics by writing GTL program directly. The following example create a panel of one scatter plot and one series plot, and put different marker attributes on them. By default, SG procedures only allow the same plot types and marker attributes across the cells in a panel.

```
* [Example 6.2] Create my own GTL and render it in SGRENDER;
proc template;
  define statgraph my_first_GTL;
    begingraph;
      entryTitle 'This is the Graph Title';
      layout lattice / columns=2;
        layout overlay;                                  Plot 1
          scatterPlot X=Weight Y=invoice
            / Markerattrs=( Color=Red Symbol=CIRCLEFILLED Size=2.5%);
        endlayout;
        layout overlay;                                  Plot 2
          seriesPlot X=Weight Y=MSRP
            / lineattrs=( Color=GREEN Size=1.5%);
        endlayout;
      endlayout;
    endgraph;
  end;
run;

proc sort data=cars_small;
    by weight;
run;

ods graphics / reset=all imagename='example61' imagefmt=png
               width=12in height=6in border=off ;
ods html gpath='C:\' style=LasVegas2;
proc sgrender data=cars_small template=my_first_GTL;
run; quit;
```

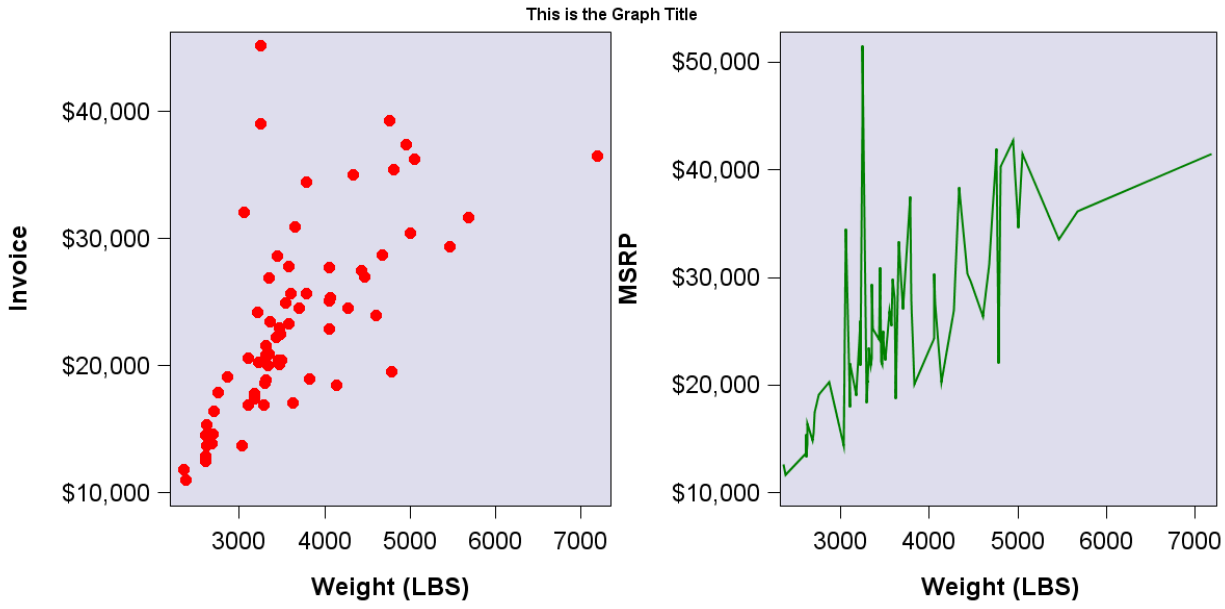
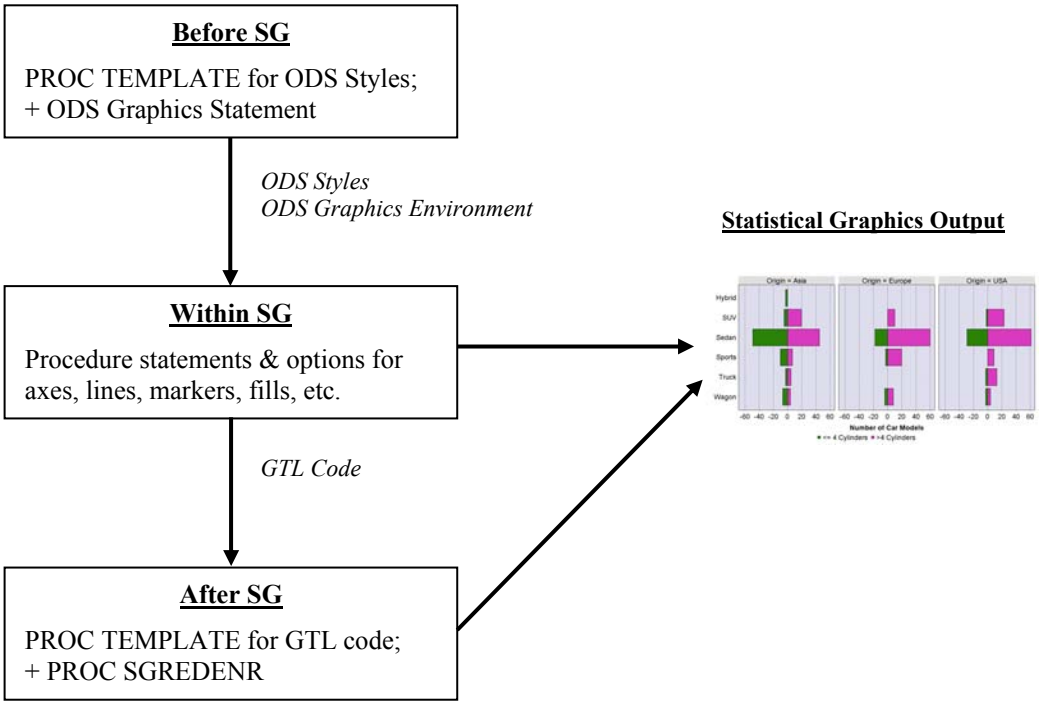


Figure 11. Panel of One Scatter Plot and One Series Plot, Created by GTL and PROC SGREDENR

CONCLUSION

The four SG procedures, SGPLOT, SGPANEL, SGSCATTER and SGRENDER, are new with SAS 9.2 to employ Graphics Template Language to create various statistical graphs. This paper summarizes three possible ways to enhance the SG outputs within the ODS graphics settings before, within, and after SG procedures. The SG graph enhancement is flexible yet the syntax is efficient and easy to understand.



REFERENCES

- Chuck Kincaid. SGPANEL: Telling the Story Better. SAS Global Forum 2010, 234-2010.
- Xiangxiang Meng. Creating High-Quality Scatter Plots: An Old Story Told by the New SGSCATTER Procedure. SAS Global Forum 2010, 057-2010.
- Lora D. Delwiche, Susan J. Slaughter. Using PROC SGPLOT for Quick High-Quality Graphs. SAS Global Forum 2009, 158-2009.
- Susan Schwartz. Clinical Trial Reporting Using SAS/GRAPH SG Procedures. SAS Global Forum 2009, 174-2009.1

CONTACT INFORMATION

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

Xiangxiang Meng
Department of Mathematical Science University of Cincinnati
mengxa@mail.uc.edu