

The Graph Template Language: Beyond the SAS/GRAPH® Procedures

Jesse M. Pratt, Cincinnati Children's Hospital Medical Center, Cincinnati, OH

ABSTRACT

The SGPLOT and SGPANEL procedures are powerful tools that are capable of producing many types of high quality graphs; however, these procedures have some limitations. What happens when one is asked to specifically produce a graph that these procedures cannot create? The Graph Template Language (GTL) is much more flexible when it comes to creating customized displays. This paper presents situations where the SGPLOT and SGPANEL procedures break down, then briefly introduces GTL, and finally uses GTL to generate the displays not possible in PROC SGPLOT and PROC SGPANEL.

INTRODUCTION

"Can I see all of the observations with that boxplot instead of just the outliers? How about making the labels blue instead of black? Can you put graphs of these variables on the same display, even though they have dramatically different ranges?" Requests such as these can cause head-scratching moments for a SAS programmer. Why? Even though the SGPLOT and SGPANEL procedures can create many versatile displays, they do have limitations that can cause difficulty when trying to create certain kinds of graphs. Examples of such tasks include overlaying a scatter plot or smoothing curve over a set of boxplots, adding color to labels and axis values, displaying a lattice of two or more different types of graphs, and having a lattice of graphs with different scales on the axes. The Graph Template Language (GTL), introduced in SAS 9.2, can perform tasks that are not possible using the SGPLOT and SGPANEL procedures. This paper will go into more detail about the examples previously listed, introduce GTL and explain basic syntax, and then show how GTL can succeed where these procedures fail. Knowledge of PROC SGPLOT and PROC SGPANEL is assumed.

SAS/GRAPH® SHORTCOMINGS

Incompatible Graphs

The SGPLOT and SGPANEL procedures are incredibly powerful and efficient tools for creating high quality graphics within SAS; however, there are certain limitations. First, consider overlaying a scatter plot and a boxplot. The SCATTER and VBOX statements in PROC SGPLOT are not compatible; therefore they cannot be used for the same graph, as the following code and error message shows:

```
proc sgplot data = sasdata.cars;
  vbox MPG_CITY / category = DRIVETRAIN;
  scatter x = DRIVETRAIN y = MPG_CITY;
run;
ERROR: Attempting to overlay incompatible plot or chart types
NOTE: The SAS System stopped processing this step because of errors.
NOTE: PROCEDURE SGPLOT used (Total process time):
    real time          1.31 seconds
    cpu time           0.65 seconds
```

A similar message would be generated when trying to overlay a smoothing curve over the boxplots.

Label Attributes

What if one wanted to produce a graph with blue axis labels instead of the default black? To change characteristics of markers, there is a MARKERATTRS= statement; to change the pattern and color of lines, there is a LINEATTRS= statement. Perhaps a LABELATTRS= statement exists in PROC SGPLOT within the axis statements:

```
proc sgplot data = sasdata.cars;
  vbox MPG_CITY / category = DRIVETRAIN;
  xaxis label = "Drive Train" / labelattrs = (color = blue);
run;
```

Running this code returns the following:

```

ERROR 22-322: Syntax error, expecting one of the following: ;, DISCRETEORDER, DISPLAY, FITPOLICY,
GRID, INTEGER, INTERVAL, LABEL, LOGBASE, LOGSTYLE, MAX, MIN, MINOR, NOTIMESPLIT,
OFFSETMAX, OFFSETMIN, REFTICKS, TICKVALUEFORMAT, TYPE, VALUES, VALUESHINT.
ERROR 76-322: Syntax error, statement will be ignored.
16 run;

NOTE: The SAS System stopped processing this step because of errors.
NOTE: PROCEDURE SGPLOT used (Total process time):
      real time           0.01 seconds
      cpu time            0.01 seconds

```

Multiple Graphs with Different Axes

When using PROC SGPANEL, each variable is graphed with the same scale on the y-axis. What happens when some of the variables range from 50 to 100 and others from -1 to 1 as in Figure 1 below?

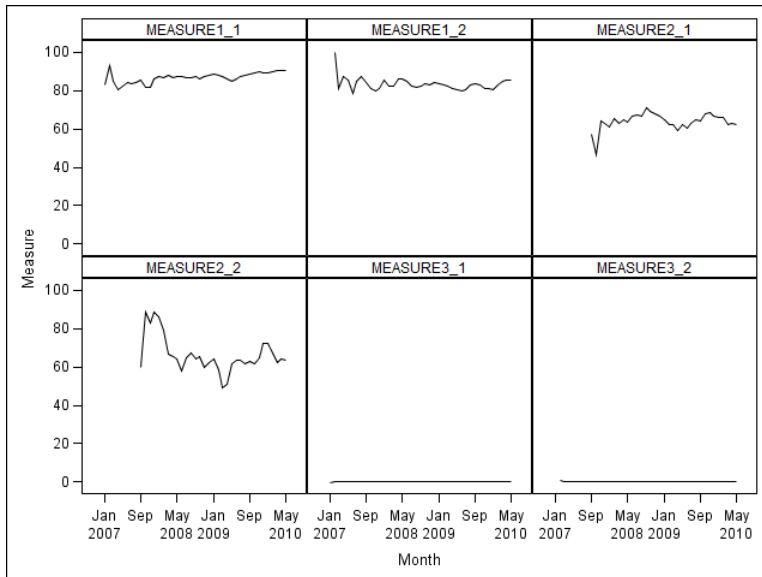


Figure 1: PROC SGPANEL with Variables Having Different Ranges

Note some other limitations in SGPANEL. The grouping variables are displayed alphabetically, and PROC SGPANEL requires each graph to be of the same type.

GETTING STARTED WITH GTL

Using GTL requires the use of the TEMPLATE procedure in order to define the STATGRAPH template. After defining the template, the SGRENDER procedure is used to create the actual graph. Here is the basic code structure used to create a single graph using PROC TEMPLATE:

```

proc template;
  define statgraph TEMPLATE.NAME;
    begingraph;
      entrytitle "TITLE";
      layout statement(s) / options;
      plot statement(s) / options;
      endlayout statement(s);
    endgraph;
  end;
run;

proc sgrender data = <DATA SET> template = "TEMPLATE.NAME";
run;

```

Note that there is no DATA = statement within the PROC TEMPLATE statement.

The DEFINE STATGRAPH statement tells SAS that a graphics template is desired, and the template is named by the user. An END statement goes with the DEFINE statement.

After the DEFINE statement comes the BEGINGRAPH statement, which has an ENDGRAPH statement that goes with it. The optional ENTRYTITLE statement puts a title on the graph. To have multiple titles, simply use multiple ENTRYTITLE statements. Note that, unlike TITLE statements in the rest of SAS, the ENTRYTITLE statements are not numbered.

Next is the LAYOUT statement. This paper will only consider the LAYOUT OVERLAY and LAYOUT LATTICE statements. See the SAS User's Guide for GTL for details on the other LAYOUT statements. LAYOUT OVERLAY is the most basic and common of the LAYOUT statements. It is used to create layered composite plots (for example, a density curve layered over a histogram). An ENDLAYOUT statement is needed for LAYOUT statements.

The LAYOUT LATTICE statement is used to create a series of plots on a single display, arranged on a lattice. Multiple LAYOUT OVERLAY statements are used within the LAYOUT LATTICE statement. Use the COLUMNS= option to determine the size of the lattice. Note that each LAYOUT statement must have a corresponding ENDLAYOUT statement.

Plot statements have very similar syntax in PROC TEMPLATE as they do in PROC SGPLOT, however the individual plot names are usually slightly different (see Table 1 for examples of the differences) and all plots that require both an X and a Y variable use X= and Y= statements after the plot name. Histograms and density curves only require the variable in question to be specified after the plot name.

PROC SGPLOT	PROC TEMPLATE
SCATTER	SCATTERPLOT
SERIES	SERIESPLOT
BAND	BANDPLOT
NEEDLE	NEEDLEPLOT
LOESS	LOESSPLOT
REG	REGRESSIONPLOT
PBSPLINE	PBSPLINEPLOT
ELLIPSE	ELLIPSEPLOT
HISTOGRAM	HISTOGRAM
VBOX	BOXPLOT
VBAR	BARCHART

Table 1: Comparing Plot Statements in PROC SGPLOT and PROC TEMPLATE

XAXIS and YAXIS statements are replaced with XAXISOPTS=() and YAXISOPTS=(), respectively, in the OVERLAY statement in PROC TEMPLATE. See Table 2 for details. For more details on all syntax, see the SAS User's Guide for GTL.

PROC TEMPLATE Statement	Description
XAXISOPTS = () (or YAXISOPTS)	Axis options. Used in the OVERLAY statement.
LABEL =	Creates an axis label. Used within the XAXISOPTS option.
LABELATTRS = (COLOR = SIZE =)	Modifies the label attributes, such as color and size. Used within the XAXISOPTS option.
TICKVALUEATTRS = (COLOR = SIZE =)	Modifies the tick attributes on the axis, such as color and size. Used within the XAXISOPTS option.
LINEAROPTS = ()	Sub-option within XAXISOPTS. Controls the values on the axes.
TICKVALUESEQUENCE = (START = END = INCREMENT =)	Values given will be the values plotted as long as they are in the actual data range. Used within the LINEAROPTS sub-option.
VIEWMIN = VIEWMAX =	Used to reduce or extend the axis data range. Used within the LINEAROPTS sub-option.

Table 2: Description of the Axis Options within PROC TEMPLATE

AN EXAMPLE

To get a feel for GTL in action, let's take a look at an example. The data used is the "Cars" data set from the SASHELP library, and we shall consider a scatter plot of city miles per gallon and weight in pounds of the car.

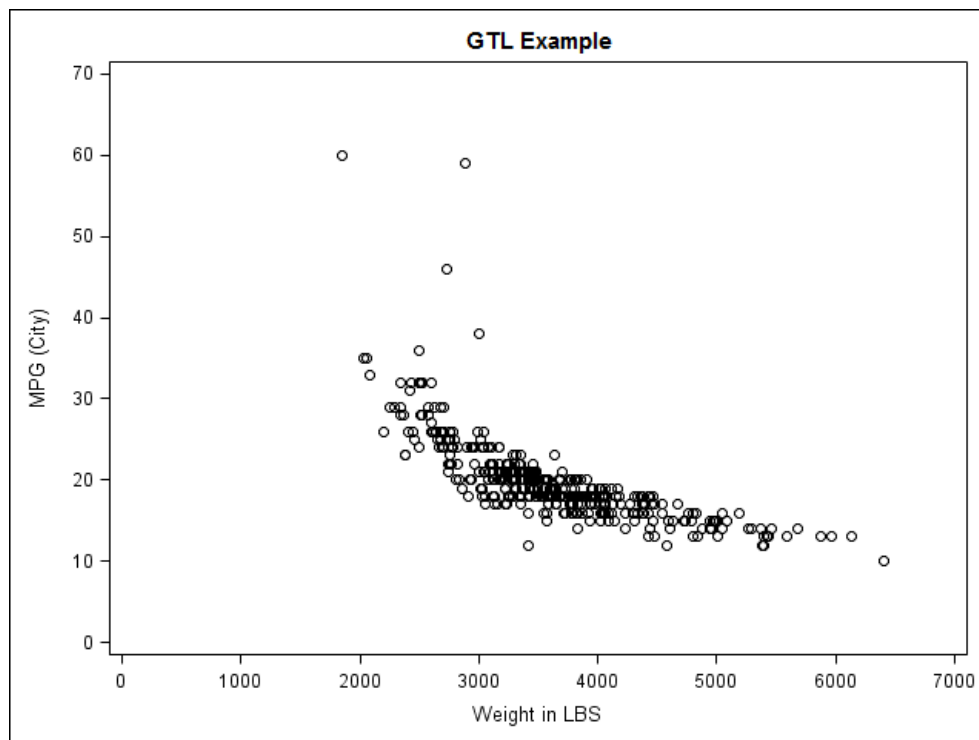


Figure 2a

```

proc template;
  define statgraph mygraph.example1;
    begingraph;
      entrytitle "GTL Example"
      layout overlay / yaxisopts=(label="MPG (City)"
        linearopts=(tickvaluesequence=(start=0 end=70
          increment=10) viewmin=0 viewmax=70))
        xaxisopts=(label="Weight in LBS"
          linearopts=(tickvaluesequence=(start=0 end=7000
            increment=1000) viewmin=0 viewmax=7000));
      scatterplot x=WEIGHT y=MPG_CITY;
    endlayout;
  endgraph;
end;
run;

proc sgrender data=sashelp.cars template="mygraph.example1";
run;

```

Figure 2b

For a great introduction to GTL, see *Matange, 313-2008*.

THE POWER OF GTL

The graph presented in Figure 2a can be easily reproduced using PROC SGPLOT. This final section will illustrate some displays that cannot be produced using PROC SGPLOT or PROC SGPANEL. Specifically, it will show:

- A lattice showing a graph of a kernel density estimator on one plot and an empirical CDF on the other.
- A set of boxplots with a smoothing curve through them.
- A lattice showing different axes for some of the plots, as well as coloring the graph, axis labels, and axis values.
- A set of boxplots with a jittered scatterplot overlaid.

The plots will be shown first, with code and explanation to follow.

DIFFERENT GRAPH TYPES ON THE SAME LATTICE

This is an example of putting two different types of graphs on the same lattice, something not possible in PROC SGPANEL. In particular, the plot on the left of Figure 3a shows a kernel density estimator and the plot on the right shows an empirical CDF with a smoothing curve through it.

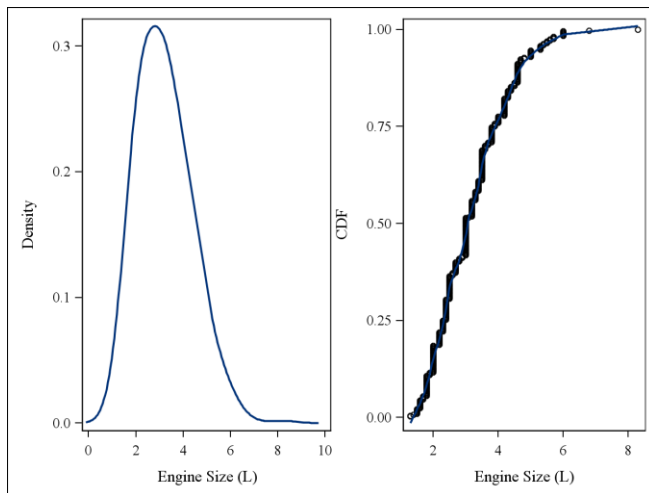


Figure 3a

```

proc sort data = sashelp.cars;
  by ENGINE SIZE;
run;

data cars;
  set sashelp.cars nobs=TOTALOBS;
  ECDF=_N_/TOTALOBS;
run;

proc template;
  define statgraph mygraphs.example2;
    beginngraph;
      layout lattice / columns=2;

      layout overlay;
        densityplot ENGINE SIZE / kernel(c=1.2);
      endlayout;
      layout overlay / yaxisopts=(label="CDF");
        scatterplot x=ENGINE SIZE y=ECDF;
        loessplot x=ENGINE SIZE y=ECDF;
      endlayout;

    endlayout;
  endgraph;
end;
run;

proc sgrender data=sashelp.cars template="mygraphs.example2";
run;

```

Figure 3b

- The initial SORT procedure and DATA step create the cumulative probabilities for the empirical CDF.
- The first LAYOUT LATTICE statement is used to create multiple plots on the same display, and the COLUMNS= option determines how many columns will be used in the layout.
- Note that each plot in the display needs its own LAYOUT LATTICE statement.

A SMOOTHING CURVE OVERLAID ON BOXPLOTS

The data set used contains reproductive data on water fleas exposed to different concentrations of nitrofen, an herbicide. A boxplot of the total number of offspring for each concentration of nitrofen is desired, along with a smoothing curve overlaid on top.

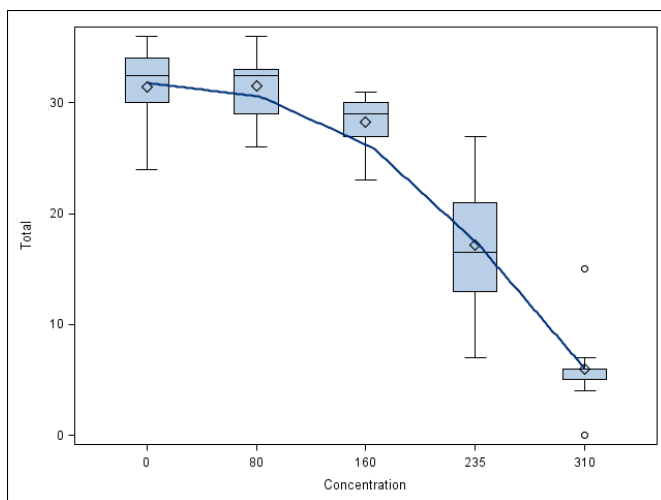


Figure 4a

```

proc template;
  define statgraph mygraphs.example3;
    begingraph;
      layout overlay / x2axisopts=(display=(line));
        boxplot x=CONC y=TOTAL;
        loessplot x=CONC y=TOTAL / xaxis=x2 smooth=0.7;
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=nitrofen template="mygraphs.example3";
run;

```

Figure 4b

The main technique to get these normally incompatible graphs overlaid on each other involves:

- Plot the boxplots on the X-axis and the smoothing curve on the X2-axis. Even within PROC TEMPLATE, two incompatible graph types cannot be on the same axis. They can, however, be on different x-axes within PROC TEMPLATE, but not in PROC SGPLOT or SGPANEL.
- To put a plot on the X2-axis, use the XAXIS=X2 option in the plot statement.
- The X2-axis needs to be blank for the aesthetics of the graph. To make the axis totally blank, use the X2AXISOPTS=(DISPLAY=(LINE)) option in the LAYOUT OVERLAY statement.

COLORED GRAPHS WITH DIFFERENT AXES

This graph illustrates how to change the colors of the values on the axes, making the axes different lengths. Since each graph needs its own LAYOUT OVERLAY statement, manually coding this graph could take some time. This example will use a macro to help shorten that time.

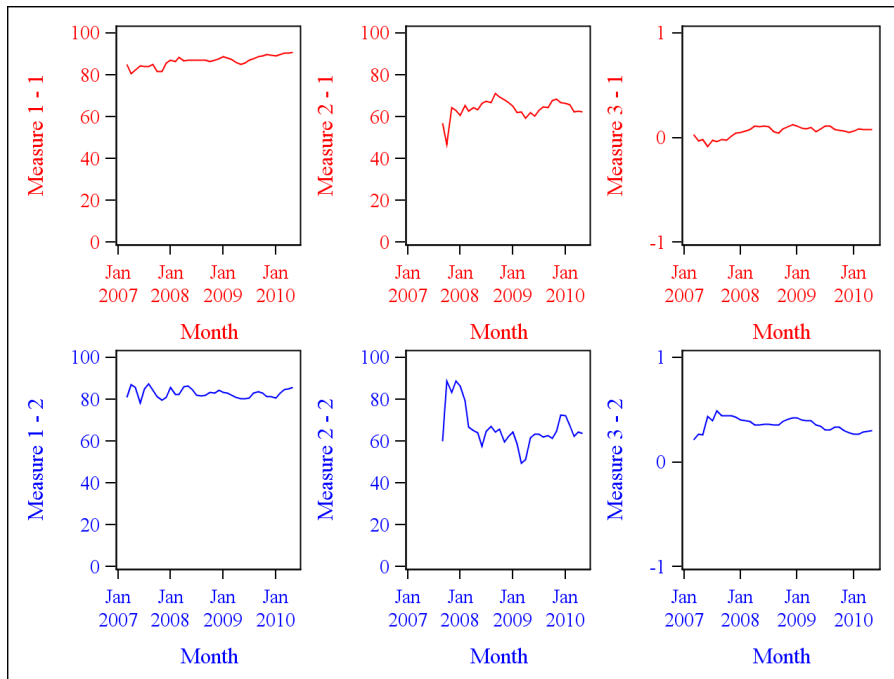


Figure 5a

```

%macro template(YVAR,COLR,LO,HI,INC,LAB);
  layout overlay / yaxisopts=(label="&LAB" labelattrs=(color=&COLR)
                        tickvalueattrs=(color=&COLR)
                        linearopts=(tickvaluesequence=(start=&LO end=&HI
                        increment=&INC) viewmin=&LO viewmax=&HI))
  xaxisopts=(label="Month" labelattrs=(color=&COLR)
            tickvalueattrs=(color=&COLR));
  seriesplot x=MONTH y=&YVAR / linearattrs=(color=&COLR);
endlayout;
%mend;

proc template;
  define statgraph mygraphs.example4;
    begingraph;
      layout lattice / columns=3;
      %template(MEASURE1_1,red,0,100,20,Measure 1 - 1)
      %template(MEASURE2_1,red,0,100,20,Measure 2 - 1)
      %template(MEASURE3_1,red,-1,1,1,Measure 3 - 1)
      %template(MEASURE1_2,blue,0,100,20,Measure 1 - 2)
      %template(MEASURE2_2,blue,0,100,20,Measure 2 - 2)
      %template(MEASURE3_2,blue,-1,1,1,Measure 3 - 2)
    endlayout;
  endgraph;
end;
run;

proc sgrender data=monthly_measures template="mygraphs.example4";
run;

```

Figure 5b

By noting what changes in each of the six needed LAYOUT OVERLAY statements, we can use a macro with the changing items as macro variables. In this example:

- YVAR corresponds to the outcome variable for each graph.
- COLR to the desired color.
- LO to the minimum value on the Y-axis.
- HI to the maximum value on the Y-axis.
- INC to the increment used for the Y-axis.
- LAB to the desired label for the Y-axis.

A JITTERED SCATTER PLOT OVERLAID ON BOXPLOTS

Scatter plots and boxplots are incompatible in the SAS/GRAPH procedures, however using the technique employed in Figures 4a and 4b will allow the two to be overlaid in PROC TEMPLATE. Scatter plots where one variable is categorical can obscure data, so to remedy this, a technique known as "jittering" is used. Jittering involves generating a random quantity (usually from either a normal or uniform distribution) and adding this to each category. Note that it may be necessary to create a new variable if the original one is a character variable. The final example will illustrate a set of boxplots with a jittered scatter plot overlaid.

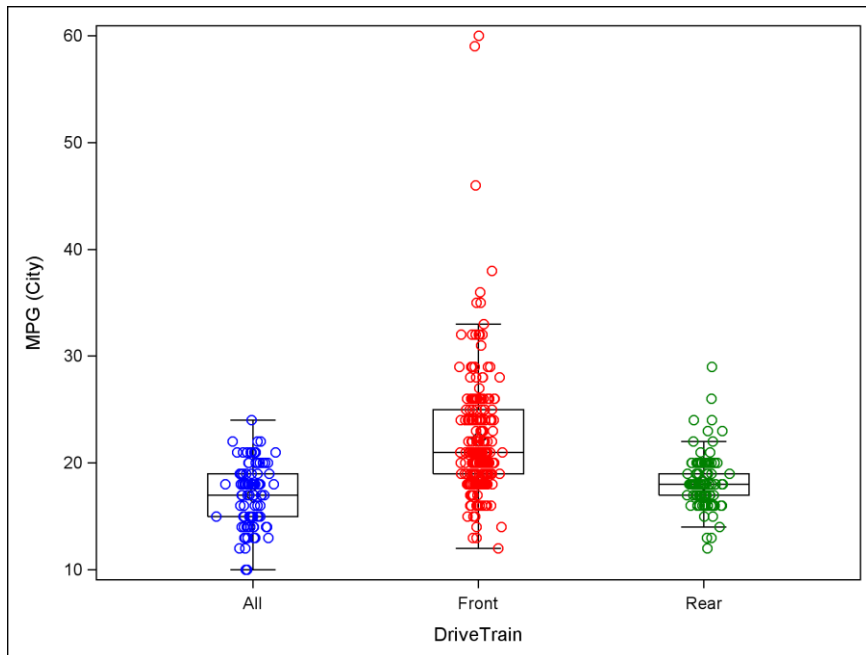


Figure 6a

```

proc sort data=sashelp.cars;
  by DRIVETRAIN;
run;

data cars2;
  set sashelp.cars;
  if DRIVETRAIN="All" then ALL=-1+0.1*rannor(0);
  if DRIVETRAIN="Front" then FRONT=1.5+0.1*rannor(0);
  if DRIVETRAIN="Rear" then REAR=4+0.1*rannor(0);
run;

proc template;
  define statgraph mygraphs.example5;
    begingraph;
      layout overlay / x2axisopts=(display=(line) linearopts=(viewmin=-1
        viewmax=4));
      boxplot x=DRIVETRAIN y=MPG_CITY / display=(caps mean median)
        fillattrs=(color=white);
      scatterplot x=ALL y=MPG_CITY / xaxis=x2 markerattrs=(color=blue);
      scatterplot x=FRONT y=MPG_CITY / xaxis=x2 markerattrs=(color=red);
      scatterplot x=REAR y=MPG_CITY / xaxis=x2 markerattrs=(color=green);
    endlayout;
  endgraph;
end;
run;

```

Figure 6b

A similar technique is used to construct Figure 6a as was used to construct Figure 4a. Some things to note:

- First create a separate numeric variable for each category, with some randomness added for the jittering effect. The choice of values of the numeric variables may need to be experimented with in order to get the scatter plots to line up properly with the boxplots.
- The BOXPLOT statement in PROC TEMPLATE will order the boxplots by the order the categories are first encountered in the data set. Therefore sorting the initial data set in the SET statement in the above data step by the X variable is highly recommended.

- Just as in Figure 4b, the X2AXISOPTS=(DISPLAY=(LINE)) option is used within the LAYOUT OVERLAY statement so that the X2-axis appears blank. The additional LINEAROPTS=() statement is also used to help line up the scatter plots and boxplots.
- Outliers for the boxplots will be produced along with the points from the scatter plot. To avoid duplication of these values, use the DISPLAY=(CAPS MEAN MEDIAN) option to only display these items.
- Attributes for boxplots can be changed using the options OUTLIERATTRS=(), MEDIANATTRS=(), WHISKERATTRS=(), MEANATTRS=() after the BOXPLOT statement.
- Not available in PROC SGPLOT, the FILLATTRS=() option can be used to change the color of the boxplots.
- Each category will need its own SCATTERPLOT statement, with each X variable being the one created in the previous DATA step.
- Remember the XAXIS=X2 option in each SCATTERPLOT statement.

CONCLUSION

This paper has highlighted limitations to the SGPLOT and SGPANEL procedures, introduced the Graph Template Language, and shown how to overcome the previous limitations by using GTL. It has also used a technique within GTL to put previously incompatible types of plots on the same graph by utilizing the X2 axis. So, in answer to the three questions posed in the introduction, “Yes, yes, and yes!”

ACKNOWLEDGMENTS

I would like to thank the Division of Biostatistics and Epidemiology at Cincinnati Children’s Hospital Medical Center and specifically Eileen King and Patricia Herbers for their comments, edits, and support.

REFERENCES

- Bailer, A. John. 2010. Statistical Programming in SAS. p 24, 25, 205. Cary, NC: SAS Institute, Inc.
- Delwiche, Lora D., Slaughter, Susan J. “Using PROC SGPLOT for Quick High Quality Graphs.” SAS Global Forum Paper Number 158-2009.
- Matange, Sanjay. “Introduction to the Graph Template Language.” SAS Global Forum Paper Number 313-2008.

CONTACT INFORMATION

Jesse Pratt
 Cincinnati Children’s Hospital Medical Center
 3333 Burnet Ave., Cincinnati, OH, 45229
 Phone: 513-803-3084
 Email: jesse.pratt_2@cchmc.org

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