

Get a Quick Start with SAS® ODS Graphics By Teaching Yourself

Roger D. Muller, Ph.D., Data To Events, Inc.

ABSTRACT

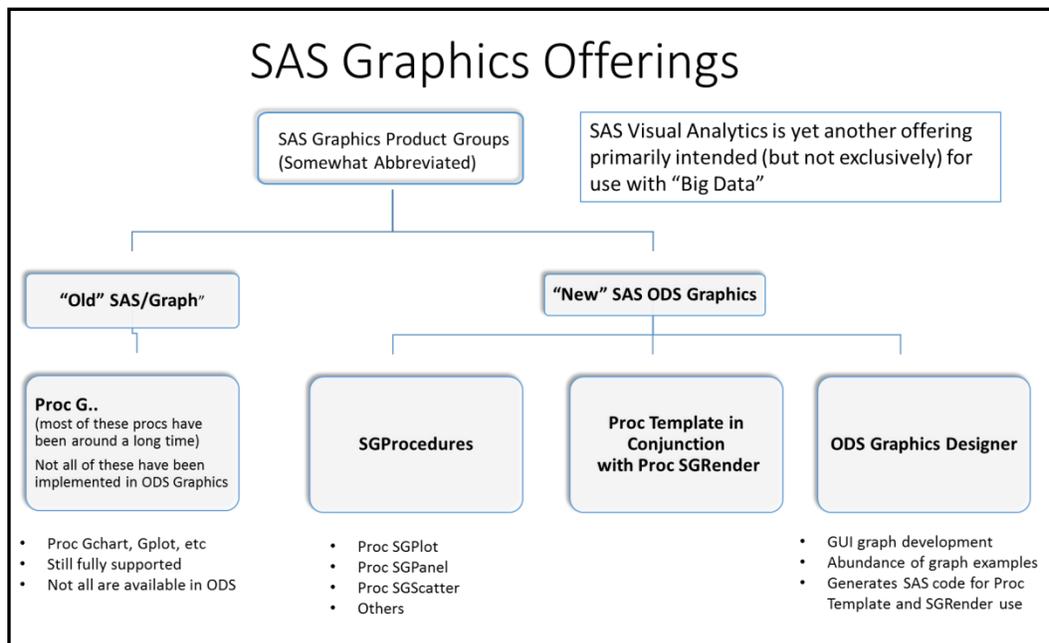
SAS® Output Delivery System (ODS) Graphics started appearing in SAS® 9.2. When first starting to use these tools, the traditional SAS/GRAPH® software user might come upon some very significant challenges in learning the new way to do things. This is further complicated by the lack of simple demonstrations of capabilities. Most graphs in training materials and publications are rather complicated graphs that, while useful, are not good teaching examples. This paper contains many examples of very simple ways to get very simple things accomplished. Over 20 different graphs are developed using only a few lines of code each, using data from the SASHELP data sets. The usage of the SGPLOT, SGPANEL, and SGSCATTER procedures are shown. In addition, the paper addresses those situations in which the user must alternatively use a combination of the TEMPLATE and SGRENDER procedures to accomplish the task at hand. Most importantly, the use of ODS Graphics Designer as a teaching tool and a generator of sample graphs and code are covered. The emphasis in this paper is the simplicity of the learning process. Users will be able to take the included code and run it immediately on their personal machines to achieve an instant sense of gratification..

INTRODUCTION

This paper is oriented to users of the original SAS/Graph software (now included in the base product, formerly marketed as a separate product) who are now wanting to move into SAS ODS Graphics (also referred to as SG graphics). The original intention of this paper was to present a number of sample graphs with the “SG” procedures. This has been expanded to include the Proc Template/Proc SGrender approaches, ODS Graphics Designer, development of styles via Proc Template and an introduction to the routing of output, using a complex example of outputting to multiple frame HTML files. The electronic poster accompanying this paper is on YouTube (Google search for “SAS Video Playlists by Roger Muller” without the quotes for a dynamic display of the ODS Graphics Designer.

I. WHERE TO START?

What a question! The beginning user can get frustrated with all of the options available and may very easily go down a path that may not get them where they want as fast as they want. First, there are many different SAS graphics offerings. The following diagram summarizes SAS offerings (with some liberties taken for brevity):



Several points need to be made before we start:

- The ODS Graphics Designer is clearly the easiest way to teach yourself the code needed to generate many, but not all, ODS graphics FAST. We are not extensively covering the ODS Graphics Designer in this written paper. Rather, it will be shown in the electronic poster and accompanying video segments on YouTube as it is highly interactive. It is awful easy for serious programmers to casually overlook this tool. You’ve heard it before: “real programmers write code, they don’t drag-and-drop”. This is best described as point-and-choose software (GUI – Graphical User Interface) that is highly interactive and can make graphics in SAS Enterprise Guide environment and in PC SAS.

ODS Graphics Designer is often used by occasional users who do not want to get into extensive programming. The Graphics Designer can be used to generate graphic template code that can be copied and pasted in the SAS environments (PC, EG, batch), and will greatly speed up the development process when learning the new procedures.

ODS Graphics Designer can most definitely be used as the “Training Program for Learning How to Code SAS ODS Graphics”. It will often get you 90% of the way there in 10% of the time. That is not a bad tradeoff. Bottom line: **START HERE!** Look at the YouTube videos. In Google, search for “SAS Video Playlists by Roger Muller” without the quotes.

- We are not addressing the assorted SAS ODS graphics that are produced as supplementary output by numerous SAS statistical procedures. These are usually documented with the procedures.
- The “SG” procedures (SGPLOT, SGPanel, and SGSCATTER) collectively referred to as “the SG procedures” are addressed in this paper.
- Another major point of emphasis is the combined use of Proc Template to create “skeleton template” files for graphic development with Proc SGRender. The stored templates are merged against SAS datasets to develop graphics with Proc SGRender.
- The use of SAS/Graph, the original SAS graphics development tool, is not the subject of this presentation. SAS/Graph is still fully supported and may occasionally be preferred for certain graphics development projects.
- SAS Visual Analytics is yet another SAS offering that will not be covered here. It is primarily marketed for use with “Big Data” and is not really used much by users of the traditional SAS/Graph product. This could change as there are pricing options that would allow for its use in environments that are less than “big data”.

SOURCES OF INFORMATION ON ODS GRAPHICS

There are many sources of information on the “new” SAS graphics. So many in fact, that it can be overwhelming. The use of internet search engines such as Google will return so many links that it can be counterproductive to achieving a quick fast start. If you are fortunate enough to have source code from previous programs developed by other competent programmers that you can modify, this will obviously get you off to a good start. There are classes available. The problem with classes is that they may not necessarily be well synchronized with the software availability at your institution and the timing of the projects you are working on. While the SAS programming manuals are available on line, the somewhat condensed version available in html form is not always helpful. The full PDF versions of the manual are on line, but can be unwieldy to use because of their sheer size. I have found that the following manuals purchased as book editions are very useful. They are not necessarily long on examples, but they do contain in-depth information on the numerous programming options (complete information in the Additional Reading section at the end of this paper):

- SAS 9.4 ODS Graphics Procedures Guide 4th edition, 2013. Total pages is 959. There are 361 pages on Proc SGPlot, 291 pages on Proc SGPanel. These procs are heavily used. This has been superseded by the 5th edition.
- SAS 9.4 ODS Graphics Procedures Guide 5th edition, September, 2015. . See special section near the end of this paper regarding use of this document including downloading and using a PDF version. Expanded to 1624 pages.
- SAS 9.4 Graph Template Language User’s Guide, Third Edition, 2014. Total pages is 633. This book covers many uses of Proc Template for creating templates that are then rendered to SAS ODS graphics with Proc SGRender. This method of operation is preferred by many users as their requirements and skills grow and advance.
- SAS 9.4 Output Delivery System User’s Guide, Third Edition, 2013. 1,130 pages. While much broader in scope than just ODS graphics, this manual addresses many issues in handling and understanding ODS (Output Deliver System) usage that programmers will eventually need.

SOME QUESTIONS TO ANSWER BEFORE YOU START

- What version of SAS do you have? If you are creating a graphics system, what version will eventual users of the system have access to? SAS ODS graphics start with version 9.2. If you are earlier than that, you must use the original SAS/Graph software.
- What is my operating environment? Likewise, what is the environment of the eventual users?
 - PC, with or without Enterprise Guide
 - Unix (or some other remote batch facility)
- What is the desired output and where does it get placed?
 - To a graphics file?
 - Just to a screen?
 - Inside a document such as an html page, a Microsoft Word RTF file, or an Adobe Acrobat PDF?
- Do you like to use Graphical User Interfaces or do you want to hardcode programs with an editor?
- Is this a “one-time deal” for creating graphics for a one person project? Or is this a system to be used by others over an extended period?

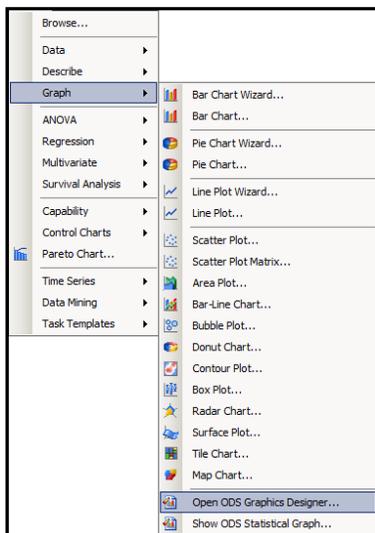
This paper does not address all of these issues. It is intended to get a single SAS programmer rapidly developing graphics by programming with a code editor with the “SG” procedures along with Proc Template and its associated use of Proc SGRender.

II. THE ODS GRAPHICS DESIGNER AS A CODE TRAINER AND “PROTOTYPER”

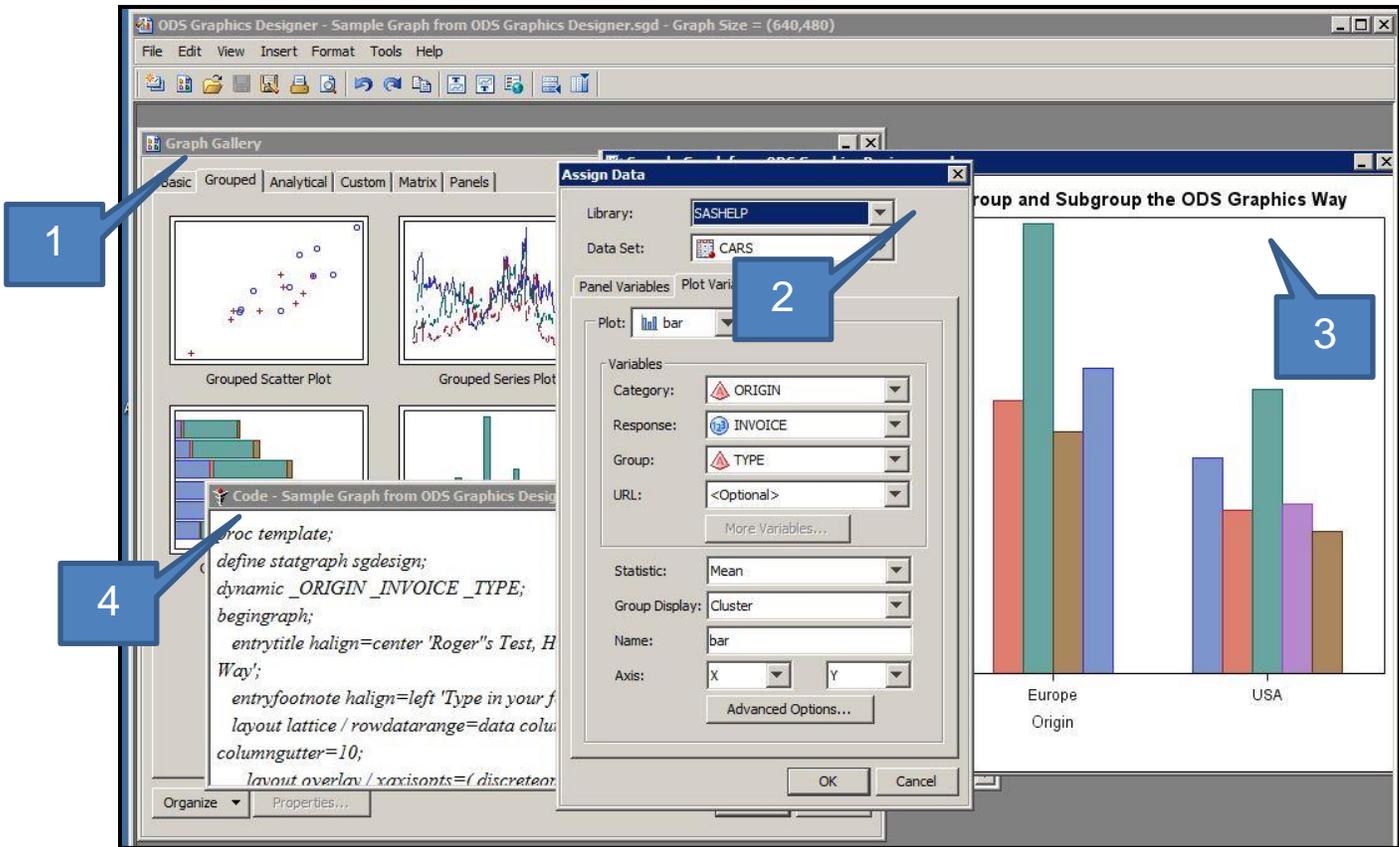
Do yourself a big favor and forget any preconceived notions about the use of GUI tools with their drag-and-drop features. Use ODS Graphics Designer as your initial code prototype tool and your personal trainer. There is no better set of examples of potential SAS ODS Graphs that can be generated here. While it does take the approach of using Proc Template with Proc SGRender, it will also benefit users of SGPlot, SGPanel and SGScatter as there are lots of similarity in the code.

Getting ODS Graphics Designer Started. This depends a little bit on your environment. It should be installed as the software has been included with the Base SAS installation since version 9.3.

In SAS Enterprise Guide 7.1, ODS Graphics Designer is somewhat hidden. From the Enterprise Guide menu bar select “Tasks” and proceed to select “Graph”. Then select “Open ODS Graphics Designer” as shown in the figure to the left and it will open into the “Graph Gallery” window. Find a sample that more-or-less matches what you want to do.



An Important Note: When pulling down the menu “Tasks” – “Graphics” in SAS Enterprise Guide, many other graphics options are available besides “Open ODS Graphics Designer”. These are for generating graphs and SAS code with the older SAS/Graph product. This is not SAS ODS graphics. These procedures are fully supported by SAS Institute and do contain graphics capabilities that are not yet available under the SAS “ODS umbrella of graphics tools”. Do not hesitate to use them. These assorted icons are designed to be dragged into the Process Flow diagrams of SAS Enterprise Guide. **For assistance in doing this see the papers or video by Muller, 2013.**



In the figure above, 4 major sub windows are overlapped. In the Graph Gallery of samples(1), a “Cluster Grouped Vertical Bar” was selected. The “filled-in” data panel (2) is partially shown and the graph as currently specified is shown at the right (3). Most importantly, from the Menu at the top of the screen “View” – “Code” was selected and the SAS code to do this graph via Proc Template and Proc SGRender is shown in the code window (4).

Author’s personal opinion: I am repeating a position stated earlier. This tool is the best training tool out there to learn what SAS code it takes to accomplish programming in SAS “SG” graphics. This is particularly true in regards to understanding the Proc Template with Proc SGRender approach. There have not been a lot of examples published illustrating capabilities of “SG” graphics and the underlying code. This is “do-it-yourself” training and “do-it-yourself” example building via the “look-see” approach. **Start Here!**

III. THE “SG” PROCEDURES VS THE COMBINATION OF PROC TEMPLATE AND PROC SGRENDER

The three procedures to be covered are the primary ones for creating graphics:

1. Proc SGPlot will have the most examples as it the proc most often used.
2. Proc SGPanel can do most of the graphics that are done in SGPlot, but also provides an interesting “panelby” option to place graphics into a lattice grid.
3. Proc SG Scatter is used to develop scatter plots of data points.
4. Numerous references have been made earlier to the combined used of Proc Template and Proc SGRender. It has been suggested to start with the ODS Graphics Designer and use that code to learn ODS graphics techniques and programming. A lot of the coding that was developed there is similar to Proc SGPLOT, SGPanel and SGScatter.

Are there overlaps in the functionality of these proc’s? Yes. Therefore it is recommended to start standalone coding with Proc SGPlot and move on from there to Proc SGPanel, and SGScatter.

If the 3 “SG” procs don’t get the job done, then where do you go? Back to Proc Template. Nearly all of the functionality of the “SG” procs and MORE is available. The Graphics Template Language (GTL) is more robust.

IV. BEFORE GETTING STARTED, A DISCUSSION ON THE ODS STATEMENTS THAT “WRAP THE GRAPHICS CODE”

There is a wrapper around the graphics code that is developed. This wrapper usually contains a statement on some specific graphics options and then some code concerning the routing of output. There may more.

Typical order (very primitive and simplified):

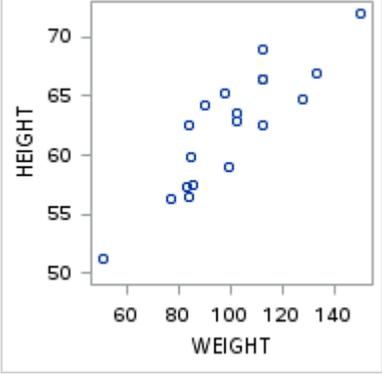
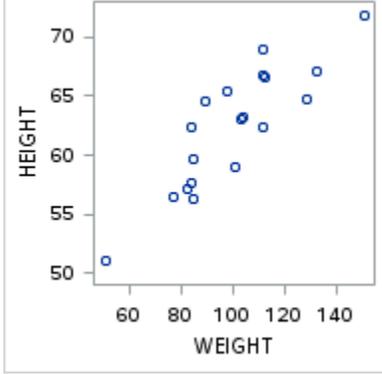
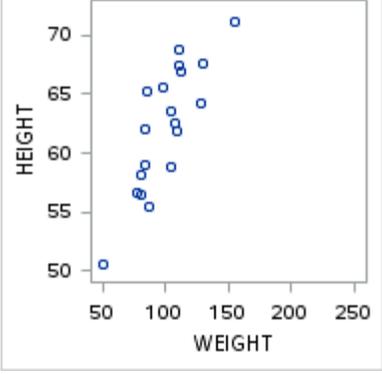
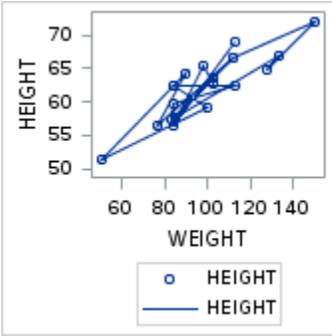
```
ODS graphics on;
ODS HTML ...etc.    ;
Lots of graphics code (Proc SG..whatevers);
ODS HTML close;
```

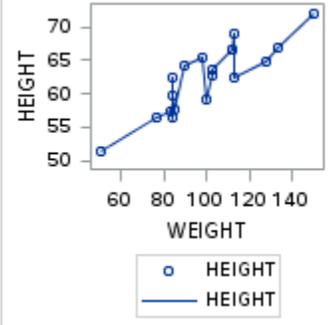
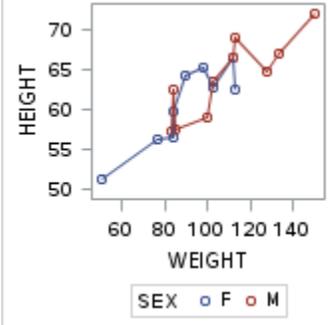
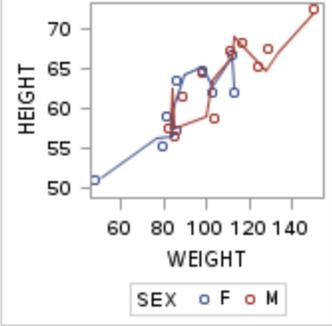
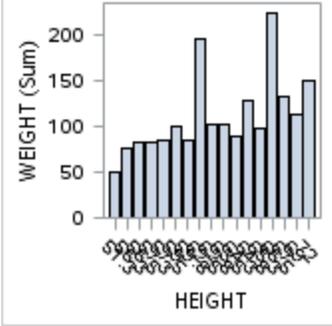
The following are a few comments on these statements and the wrapper. An intense HTML example will be presented at the end of the paper.

<code>ODS Graphics On /Reset;</code>	Turns on ODS graphics (or Off if desired). Reset will set other options to default
<code>ODS Graphics On /imagemap</code>	Enables display of data tips in HTML graphics. These are not displayed in other output destinations such as PDF, RTF, etc. Data tips are mouse overs that display underlying data values and more (see intense HTML example).
<code>ODS Graphics Height=3in width=3in;</code>	Set graphics size.
<code>ODS Graphics imagefmt=jpeg ; /* png bmp etc (15 are available)*/</code>	Set format for image to be produced. If that format is not compatible with the current active output destination, this will be ignored and the default for destination will be used
<code>ODS Graphics /* any or all of 25 plus settings -*/</code>	Control a wide variety of graphics options too numerous to list here. See ODS Graphics Statement (abbreviated) , ODS Graphics Statement (more complete documentation)

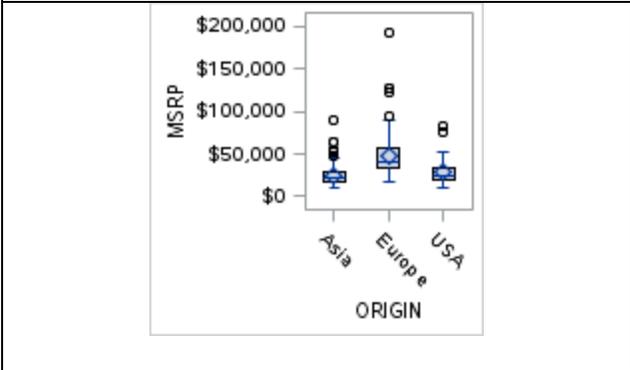
V. SAMPLES, PROC SG PLOT

A series of graphics generated by Proc Sgplot are in the following table. Most are intended to be “getting started examples”. There are very limited comments are at the top of the graph and the code is at the bottom. The code can be copied via Windows copy command and pasted into a SAS editor. The data is available in your SAS installation in the SASHELP directory. This is a look-see approach and is by no means intended to be complete documentation or example of good graphics technique, but rather give you some idea of where to start.

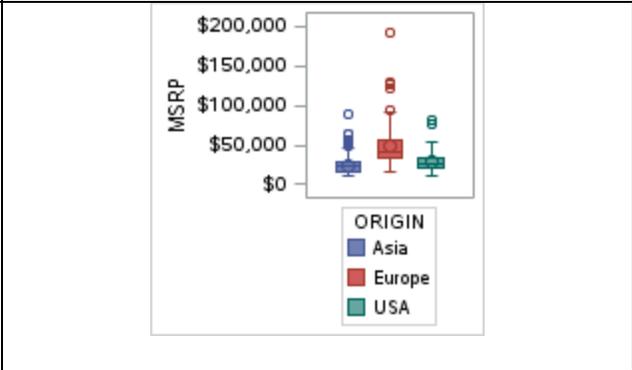
<p>Simple plot of points</p>	<p>Only 18 of 19 points showed up in the previous graph, add jitter option</p>
	
<pre>proc sgplot data=sashelp.class; scatter x=weight y=height;run;</pre>	<pre>proc sgplot data=sashelp.class; scatter x=weight y=height/jitter jitterwidth=5; run;</pre>
<p>Tweak axis scale, expand jittering.</p>	<p>Simple line connecting points</p>
	
<pre>proc sgplot data=sashelp.class; scatter x=weight y=height/jitter jitterwidth=20; xaxis value=(50 to 250 by 25); run;</pre>	<pre>proc sgplot data=sashelp.class; scatter x=weight y=height; series x=weight y=height; run; *not so great, how can it be improved?;</pre>

<p style="text-align: center;">Sorting before drawing lines and points</p>	<p style="text-align: center;">Separate lines for males and females</p>
	
<pre> proc sort data=sashelp.class out=work.class; by weight; run; proc sgplot data=work.class; scatter x=weight y=height; series x=weight y=height; run; </pre>	<pre> proc sgplot data=work.class; scatter x=weight y=height/group=sex; series x=weight y=height/group=sex; run; </pre>
<p style="text-align: center;">Let's add jitter to the points, not to the line</p>	<p style="text-align: center;">Let's do a vbar and see what happens</p>
	
<pre> proc sgplot data=work.class; scatter x=weight y=height/group=sex jitter jitterwidth=20; series x=weight y=height/group=sex ; run;*NOT a good idea!; </pre>	<pre> proc sgplot data=work.class; vbar height ; run; *these defaults don't do much for this data, may be ok for others; </pre>

Simple Vbox plot, specify category



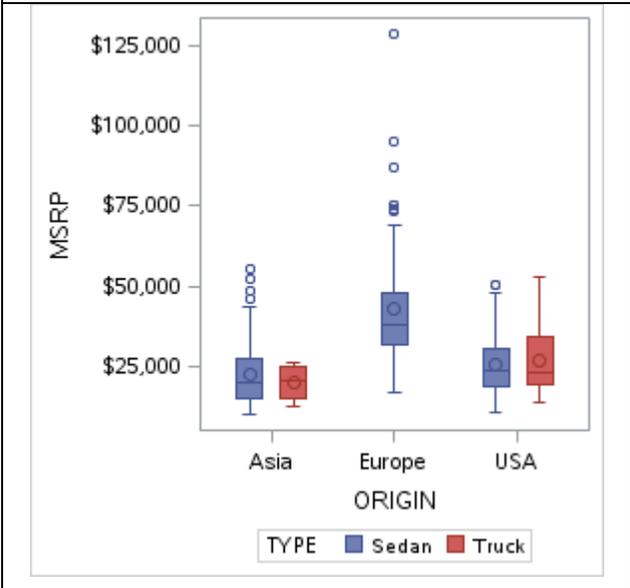
Simple Vbox, specify group



```
proc sgplot data=sashelp.cars;
vbox msrp /category=origin;;
```

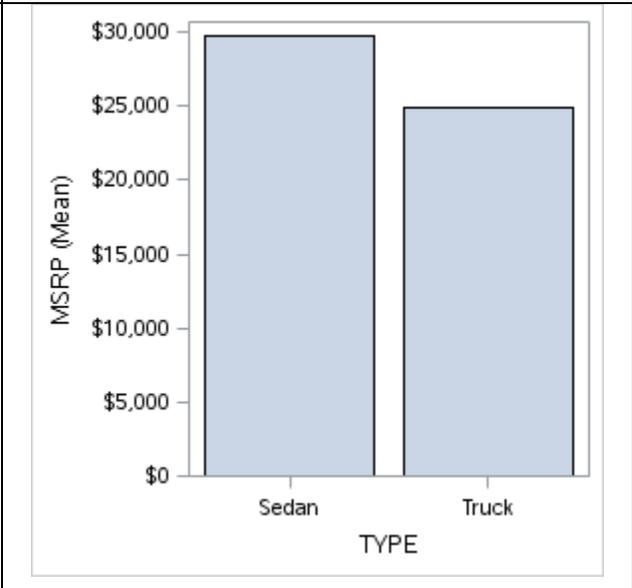
```
proc sgplot data=sashelp.cars;
vbox msrp/group=origin;
run;
```

Let's change size of graph and do a vbox with groups



```
ods graphics /height=3in width=3in;
proc sgplot data=sashelp.cars;
```

Vertical bar with stat capability

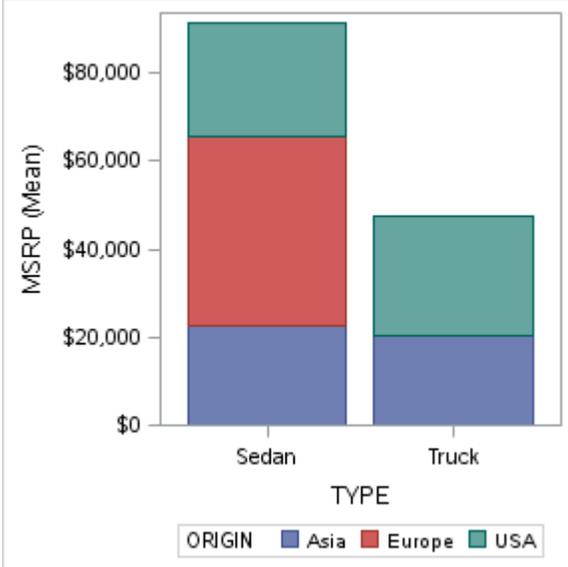


```
proc sgplot data=sashelp.cars;
where type in ('Sedan' 'Truck');
```

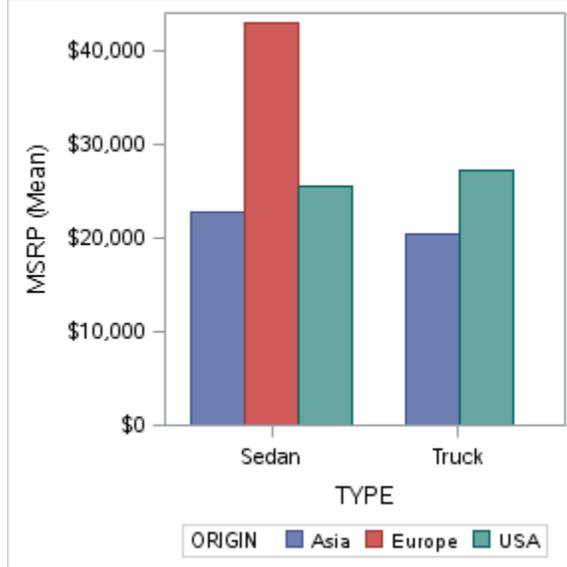
```
where type in ('Sedan' 'Truck');
vbox msrp/group=type category=origin ;
run; *looks good!;
```

```
vbar type/ stat=mean response=msrp ;
run;;
*other stats include sum, freq, median,
percent and more that you could specify;
```

Calculate means and make bar chart



Add in subcategory to make side by side



```
proc sgplot data=sashelp.cars;
where type in ('Sedan' 'Truck');
vbar type/group=origin stat=mean
response=msrp ;
run; *NOT SO GREAT. Not what we wanted.
Makes no sense, fix it on next graph;
```

```
Proc sgplot data=sashelp.cars;
where type in ('Sedan' 'Truck');
vbar type/group=origin stat=mean
response=msrp groupdisplay=cluster ;
run; *THIS IS WAY BETTER. How do you
learn to do this? Copy an old program,
read the User's Guide, search on-line.
This is not intuitive. /* A better way
to start is let ODS Graphics Designer
write your initial prototypes. Nobody
ever told me that! The code generated is
for Proc Template, not SGPlot. But it is
good enough to give an idea where to go.
*/
```

And something far more complex with a table at the right side

```

proc sort data=sashelp.cars out=cars;by
origin;run;

proc means data=cars noprint;var msrp;output
out=car_range min=Minimum max=Maximum;
by origin;run;

data car_range;set car_range; x1="Lowest Price
$"; x2="Highest Price $";run;

title "Price Range by Area of Origin";

proc sgplot data=car_range noautolegend;

band y=origin lower=Minimum
upper=Maximum/transparency=0.6;

scatter y=origin x=Maximum/
markerattrs=graphdata2(symbol=circlefilled)

transparency=0.4;

scatter y=Origin x=Minimum
/markerattrs=graphdata1(symbol=trianglefilled)

transparency=0.4;

yaxistable minimum maximum /location=inside
position=right;

xaxis display=(nolabel) grid;

yaxis display=(nolabel noticks) reverse
colorbands=odd

colorbandsattrs=(color=grey
transparency=0.9);

run;

```



The above graph shows the overlaying of a band upper and lower curves and scatter points for the data. It also shows the building of a table with column headings at the right side with the yaxistable statement. This statement is available starting with SAS 9.4.

SUMMARY OF CODING EXAMPLES

Let's quit. This could go on forever. Look in the References section for additional examples. What should you have learned here?

1. This is not SAS/Graph. The syntax is not the same. This is the "new" ODS SG graphics.
2. There are many capabilities; we have only touched upon them.
3. Look carefully at the output in simple test plots as you develop code. It may not be doing exactly what you thought.
4. Maybe we should look at the ODS Graphics Designer as a tool for developing initial code. Even though the code generated with Designer is for a Proc Template with Proc Greplay approach, it will give you some direction as to what you might try in SGPlot and other SG procedures.
5. How do you keep track of all of the little things you learn while you are coding? You will need to make some sort of scrapbook. Some of the little nuisances you learn will be used very infrequently. Look at how this panel of examples was assembled. Copy this PDF out and convert to RTF and open in Microsoft Word. Or setup your own table structure in Microsoft Excel. Put a title in the top cell, the graph in the middle cell and the code in the bottom cell. Develop the code in you SAS session (EG, PC SAS, batch). Put in the following statement to control size: ODS graphics/height=3in width=3in; You can copy and paste this size

graphic nicely into the cells in Word or Excel with the Windows Cut and Paste shortcut keys (Ctrl-C and Ctrl-V)..

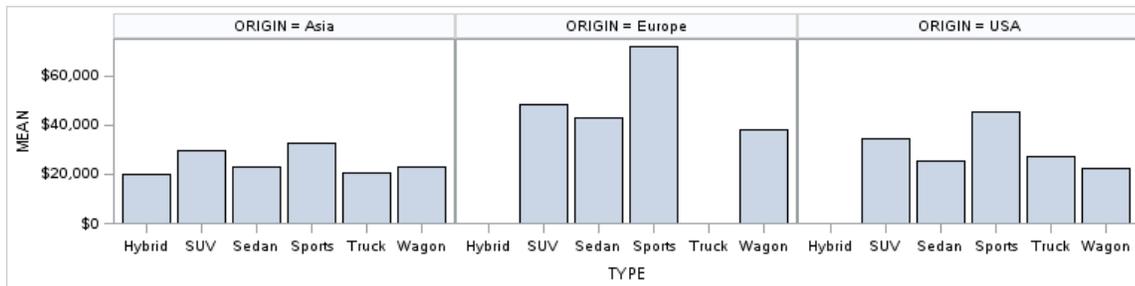
VI. SAMPLES, MOVING ON TO PROC SG PANEL

Most, but not all, of the graphics capabilities of Proc SGPlot are available in Proc SGPanel. There might be some differences in syntax on occasion. What is different is the ability to lay out a lattice panel grid to place graphics into.

The variables that build this grid must be specified with the “Panelby var” statement. It is very important to note that this capability can place only SAS variables into a given grid. Simply stated, one “Result” or “Dependent” variable can be plotted and the “panelby” variable must be in the SAS dataset.

If the task at hand is to place different variables in a series of panels in a grid on a page, this must be done with the more flexible capabilities of Proc Template and Proc SGRender which are covered later in this paper. An example of such a graph might be a 3 x 2 lattice grid where measurements for Systolic Blood Pressure over time, diastolic blood pressure and heart rate are on the top row as actual values while the %changes from baseline are on the bottom row of cells. Proc SGPanel cannot do this graphic.

The following graphic was created by Proc SGPanel.



```
proc means data=sashelp.cars alpha=.05 clm mean std;
class origin type;
var msrp;
output out=carsMean ucl=uclm lclm=lclm mean=mean;run;

ods graphics /height=2in width=8in;
proc sgpanel data=carsMean;
panelby origin/layout=columnlattice;
vbarparm category=type response=mean;run;
```

What have we learned in the above example?

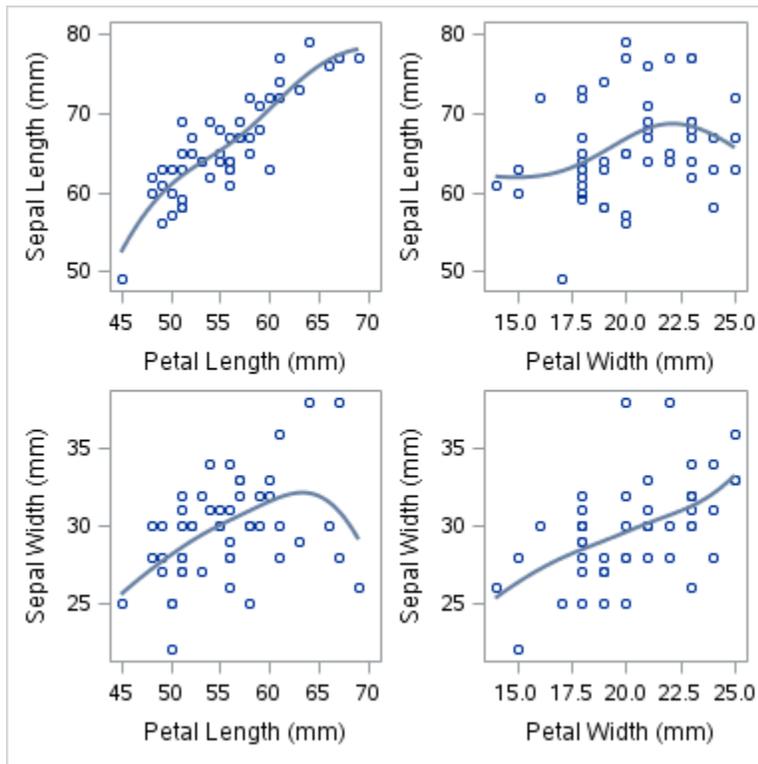
- We had to preprocess the data with proc means
- We learned about SGpanel and the panelby statement. Pretty cool when you need it. What if we had a 3 across by 4 up and down lattice chart? This would do it. It will do anything of this nature as long as you are plotting the results of only one variable and the “panelby” is a variable (or variables) on the dataset.

VII. SAMPLES, PROC SGSCATTER

This proc is used to produce assorted variations of scatter plots to study the distribution and relationships of data values. While it does have some overlap with capabilities of Proc SGplot and Proc SGPanel, it has its own unique capabilities of showing distributions and summarizing them with such features as histograms, normal distribution curves, mathematical lines etc in a paneled environment. If your project entails studying distributions of data, relationships of variables, etc., this is extremely useful.

A simple example of the flower data (Iris, Fisher) that is in the sashelp dataset shows Sgscatter is a great tool. Relationships are graphed in a multiple dimensional graph grid. This proc is very useful for data exploration and is probably not really intended to be a presentation tool. There are numerous options to add mathematical

summarizations. The amount of code used is minimal. Could the following plots be developed with Proc SGPlot and/or Proc SGPanel? Probably yes, but not this simply.



Look at the minimal amount of code required:

```
Proc sgscatter data=sashelp.iris(where=(species="Virginica"));
plot (sepallength sepalwidth)*(petallength petalwidth) /pbspline;
run;
```

VIII. PROC TEMPLATE WITH PROC SGRENDER, AN ALTERNATIVE POWERFUL COMBINATION

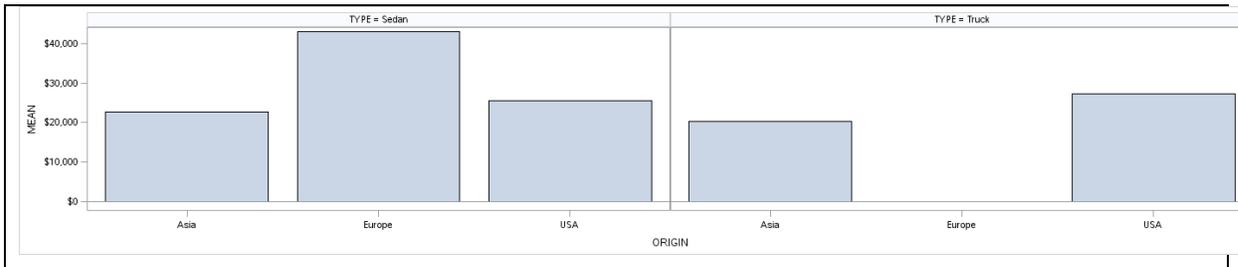
Last to be discussed in this presentation as an SG Graphic tool is the combination of Proc Template and Proc SGRender. This combination can arrive at most graphics that are done with the three earlier procedures, and a lot more. If you are running into problems readily accomplishing your projects requirements with the 3 earlier procedures, move on this combination sooner rather than later. At the end of the day, it is considered by most graph developers to be more robust.

Proc Template creates "skeleton" files that are used as a pattern or template. When these are merged with the data and specific inputs in Proc SGRender, the final graph is produced. This facility becomes even more powerful and flexible when the SAS macro facility is incorporated into its use. Rather than go into an extended discussion, two graphs will be presented. One comes from existing code which had originally been based on code found in a SAS manual. The other one was designed "from scratch" with the initial code being developed with the ODS Graphics Designer

1. Designed with Existing Code or Programmer-Developed Code

As you can see in the graph below, this graph could very easily have been developed in Proc SGPanel. If you look at the code below the graph, there are some distinct differences between this code and that used in Proc SGPanel. What is so special about this technique? Look at the Proc SGRender code. This could be reused with many diverse graphics, recalling the template. Simply specify the dataset name, the subsetting where statement and the dynamic variables and you can be using this another day on an entirely different subject.

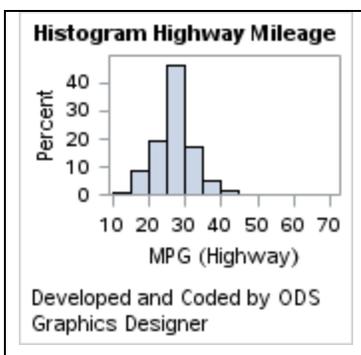
The proc template is somewhat similar to a macro, and the remaining elements in the Proc SGRender code are similar to macro variables in the sense of how they are being used.



```
proc template;
  define statgraph panels;
    dynamic cat resp col;
    begingraph;
    layout datalattice columnvar=col;
    layout prototype;
      barchartparm category=cat response=resp;
      endlayout;
    endlayout;
  endgraph;
end;
ods graphics / width=15in;
proc sgrender data=carsmean template=panels;
  where type in ('Sedan' 'Truck');
  dynamic cat='origin' resp='mean' col='type';
run;
```

2. Designed with ODS Graphics Designer-Generated Code

Remember that the ODS Graphics Designer software will generate prototypes for you. It is an excellent approach for getting started and for those situations where you are “stumped” as to how to proceed. Here is a graphic and the generated code that produced it is below;



```
ods graphics on /width=3in height=3in imagemap;
proc template;
```

```

define statgraph sgdesign;
dynamic _MPG_HIGHWAY;
begingraph;
  entrytitle halign=center 'Histogram Highway Mileage';
  entryfootnote halign=left 'Developed and Coded by ODS Graphics
Designer';
  layout lattice / rowdatarange=data columndatarange=data rowgutter=10
columngutter=10;
  layout overlay;
  histogram _MPG_HIGHWAY / name='Histogram of Highway Mileage'
binaxis=false;
  endlayout;
endgraph;
end;
run;

proc sgrender data=SASHELP.CARS template=sgdesign;
dynamic _MPG_HIGHWAY="'MPG_HIGHWAY'n";
run;

```

IX. THE STYLES AND ATTRIBUTES ASSOCIATED WITH GRAPHICS – THE SO-CALLED “ATTERS”.

Author’s note: Discussing this is huge complexity to address all aspects. This discussion is a simple overview and introduction to get you started.

What about all of the enhancements to graphics – the labels, scales, colors, legends, symbols, fills, typefaces and sizes etc. that must nearly always be used to customize graphics?

Some of these can be addressed by what is called “Styles”. These are catalogued libraries of content styles that will cover many of these needs. After specifying a style, the programmer can go in and assign all kinds of attributes to different graphics items to modify (override) current style-assigned values and thus alter their appearance. There are so many of these that SAS programmers frequently refer to them collectively as “ATTERS” as a shortcut for the word “attributes”. There are so many styles and attributes that this is beyond the scope of this paper. We will show one example of developing a custom style for the type fonts and sizes associated with graphics.

But wait, there is also your own personal trainer out there to show you examples of the behind the scenes coding of many of these items. It’s your own personal trainer “ ODS Graphics Designer”. Once you have started ODS Graphics Designer, you will start the Style Editor by selecting “Tools” – “Style Editor” from the menu bar, a window showing how to set or change some of the settings comes up. By looking at the code that is generated, you get a feel how individual elements are changed. To cover all of the elements and attributes is beyond the scope of the document. Here is an example of generated code out of the ODS Graphics Designer where Style Editor was used:

```

proc template;
define statgraph sgdesign;
dynamic _SEX_HEIGHT;
begingraph / backgroundcolor=CXE8E6E8 dataskin=matte;
  entrytitle halign=center 'This is Roger's Title' / textattrs=(color=CX0000FF
family='Brush Script MT' size=24 );
  entryfootnote halign=center 'This is Roger's Footnote';
  layout lattice / rowdatarange=data columndatarange=data rowgutter=10
columngutter=10;
  layout overlay / xaxisopts=( discreteopts=( tickvaluefitpolicy=splitrotate));
  barchart category=_SEX response=_HEIGHT / name='bar' stat=mean
barwidth=0.85 groupdisplay=Cluster clusterwidth=1.0 grouporder=data;
  endlayout;
endgraph;
end;
run;

```

```
proc sgrender data=SASHELP.CLASS template=sgdesign;  
dynamic _SEX="SEX" _HEIGHT="HEIGHT";  
run; .
```

MORE INFORMATION ON STYLES AND ELEMENTS

Understanding styles and their elements and attributes is a huge undertaking. The following link may prove useful: <http://support.sas.com/documentation/cdl/en/grstatproc/67909/HTML/default/p16cypqe7k15dbn1537m2rwlvzx9.htm>

For the listing of Styles currently available in your environment, run the following code:

```
Proc Template;  
List Styles;  
Run;
```

A list of 54 styles appeared in the output window. Most of these are well described by their name. The default for HTML output is Styles.HTMLBlue. Something like "Styles.FestivalPrinter" is likely colorful while "Styles.GrayscalePrinter" is more likely more subdued. The easiest way to learn about these is to use them in your projects and look at the output. Once you find something that suites your tastes, move on to modifying the attributes of specific elements. Look at the generated code in the example above to get an overall feel for these.

How do you specify a style?

Simple. Do something like this: ODS RTF FILE='C:\SAS_Final_Output\whatever.rtf' style= GrayscalePrinter; This will assign where the output will go and will change the appearance of the output as directed by the style.

The following link is to a document containing all of the style elements and their attributes. This is page 1357 in the PDF version of the 5th edition of the ODS 9.4 Graphics Procedures Users Guide .

<http://support.sas.com/documentation/cdl/en/grstatproc/67909/HTML/default/viewer.htm#n02bywwj20lelzn15x30wucq9pn8.htm>

Assigned attributes are not the same across all ODS destinations. Therefore, that information is not documented in publicly available information. You have to generate it yourself.

To get this information, you would need to look at the style definitions. Read this carefully, it gets complicated in a hurry. The default style used for the HTML Destination is HTMLBLUE. The parent for this style is STATISTICAL and the parent for this style in turn is DEFAULT. So you would need to look at all three of these styles in order to get a complete picture of what the output will look like by default for the HTML destination.

You can do this interactively by opening up each style or you can submit the following code:

```
Proc template;  
Source styles.htmlblue;  
Source styles.statistical;  
Source style.default;  
Run;
```

The style definitions are written to the SAS log. This is not easy to wade through. It is not pretty, many of the values are returned in hex code. Perhaps more than anything else, this log gives you the names of elements that can be changed by simple programming changes in your code without trying to go in and create new styles.

In general, it is probably best to pick a style, test it on representative output, and if you like the output, stick with it. If there are a few elements whose attributes you wish to change, do that in the plot via changes in your code.

For yet another discussion on plot elements that can be changed see the following chapters in the SAS 9.4 ODS Graphics Procedure Guide, 4th edition:

- Chapter 10, Common Concepts
- Chapter 11 Commonly Used Attribute Options.

Most of this can also be found on line at:

<http://support.sas.com/documentation/cdl/en/grstatproc/67909/HTML/default/viewer.htm#p15hx003imnhqzn1gtn4j8376gel.htm>

Google searches that might be of help with symbols, lines etc.:

- SAS ods graphics line attributes and patterns
- SAS ods graphics markerattrs
- SAS ods graphics marker attributes and symbols

X. LET'S BUILD A CUSTOM STYLE WITH PROC TEMPLATE

The following example shows how to modify some typefaces for the X axis labels and values in a simple vertical bar chart via making programming statements in the code. This can get very tedious in a hurry. We will then show how to do this by building a custom style named "Roger". This is actually not that difficult if you have an example to follow.

Run a simple graph changing some attributes for the type on the x axis label and the x axis values. See how tedious this is.

```
goptions hsize=3in vsize=3in;
title "Label on X axis modified by code"; footnote;
ods html path="/          /perftest/rmuller" file="default_type.html";
proc sgplot data=sashelp.class;
vbar sex ;
xaxis label="Sex" labelattrs=(family="timesroman" size=28pt)
valueattrs=(family="timesroman" size=28pt);
run;
quit;
ods html close;
```

Label on X axis modified by code.

Only the coded changes for the X axis label and values are made. Others are default style.

Let's find out what the default type is for graphs. Submit this code and look at the SAS log:

```
proc template;
source styles.default;
run;
```

This generates output to the saslog file, 100's of lines. There are many areas addressed and all can be changed. We are looking for things affecting Graph Fonts.

Let's modify the style, kick up the point sizes:

```

proc template;
define style roger;
parent = styles.default;

class GraphFonts

  "Fonts used in graph styles" /
  'GraphDataFont' = ("

```

Go to the top of the log, scroll to a section that begins with “class Graph Fonts”. Select all of this and paste into the program editor as shown in the left hand column. Copied out code is hilited in grey. Add the 3 lines of code at the top and an “end;” and “run;” at the bottom.

All code was modified by adding 10 points to the size specification (i.e. 7 became 17, 9 became 19, etc). This file was then submitted and the “roger” style was built in a work area. Remember that everything else in the default style is now in the “Roger” style.

Then submit this code to run the graph:

```

ods html close;
goptions hsize=3in vsize=3in;
title "Label on X axis modified by Style"; footnote;
ods html path='/          /perfctest/rmuller' file='roger_type.html' style=roger;

proc sgplot data=sashelp.class;
vbar sex ;
xaxis label="Sex" ;
run;
quit;
ods html close; Label on X and Y axis modified by Style

```

All point sizes for labels and values have been increased by 10 points. These changes are made by altering the style, not coding each and every attribute. SLICK!

Finally a word to the wise. Don't get too bogged down in this. Allow your skills with style elements and their attributes to develop over time. Contact SAS technical support at support.sas.com and submit a question. The answer will come back to you faster than you suspect and will greatly relieve any anxiety you had about ODS graphics.

XI. WHERE DOES THE OUTPUT GO? GENERATING AN HTML WITH FRAMES

Quite frankly almost anyplace. It was not the original intent of this paper to get into this discussion. If you don't specify an output location, it will most likely end up in the Results screen of your interactive session. What about other options. Here are some: PDF, RTF, HTML and more. Rather than try to cover all of these, let's look at one

that is very complex --- outputting to HTML with frames. This creates a very dramatic output for use with browsers, and again, once you have an example to follow is not that difficult. After this, other outputs are simple.

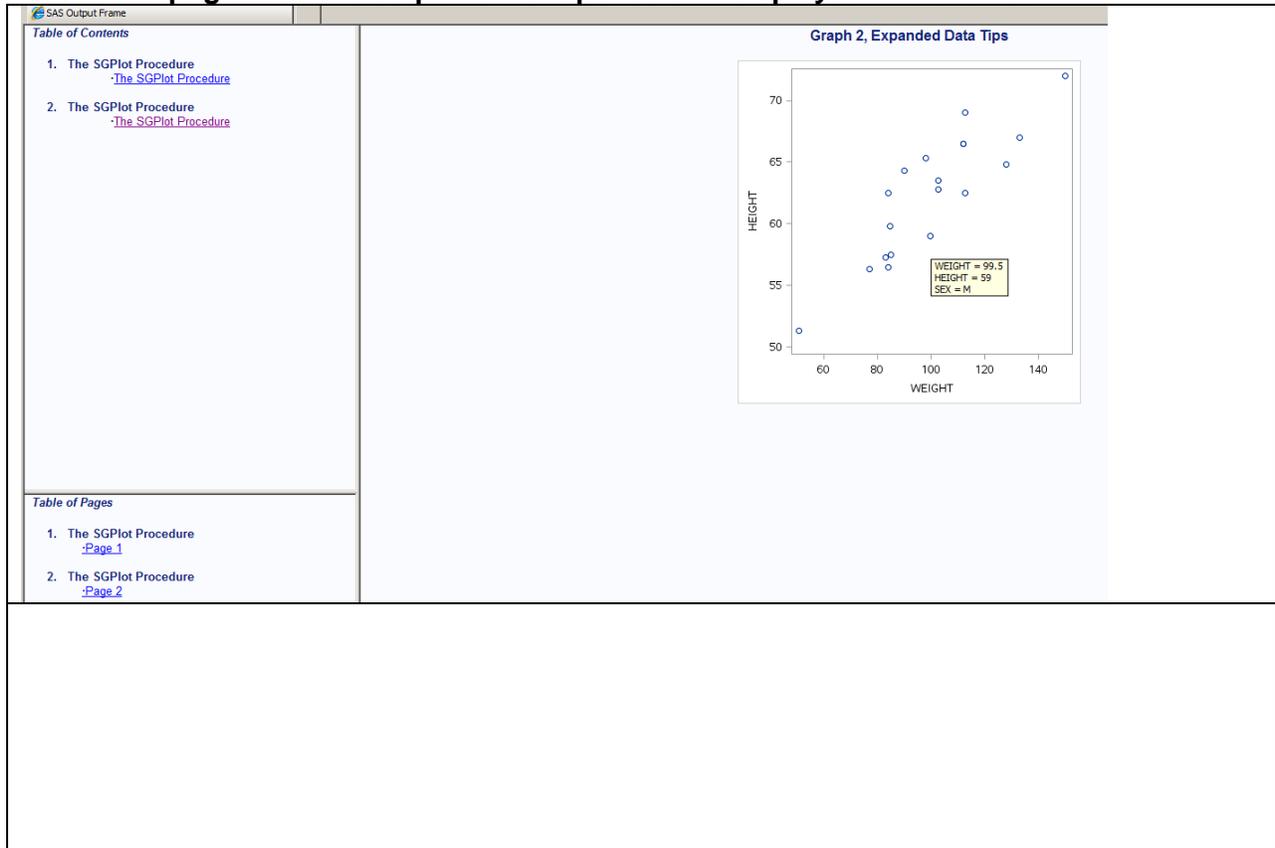
To make the example even more complex, we'll pass some of the destination information as macro variables and will put the files out to Linux. There are 4 file types associated with this: body, contents page and frame. All of these are placed in one directory. Alternatively they could be mapped to various locations. The entire output is opened by opening the frame file with a browser.

Pay close attention to (url=none). This allows the frame file to look only in the folder in which it resides to find the assorted pieces when it is opened. Otherwise the content will all be hardcoded via URL links to the specific location where it was first placed. It can't be moved to different locations, the drive mappings (drive letters and more) will hard-coded to the locations where they are created. Use of (nouri) completely does away with this problem and you can move directories around at will. The code follows:

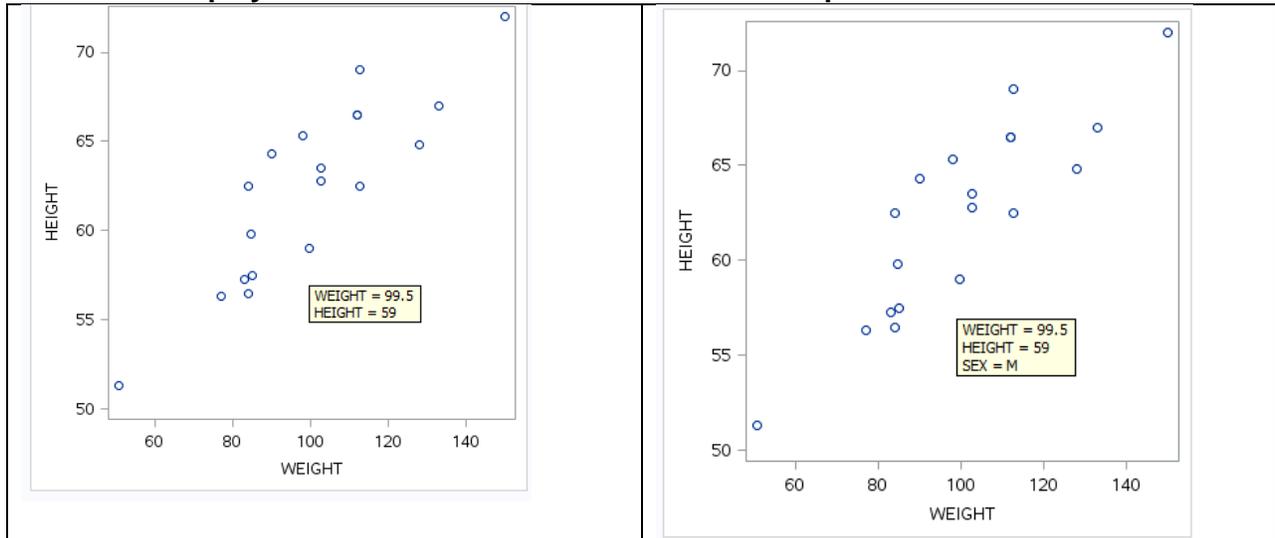
The code:

```
*examples of datatips being used in an html file with frames;
/*following is for a linux system; */
%let outpath=/perftest/rmuller;
/* a pc system might be something like c:\temp\mwsug; */
%let prefixfilenames=MWSUG2015;
ods graphics /imagemap height=4in width=4in;
ods html path="&outpath" (url=none)
nogtitle newfile=page
options(scroll_tables='yes' scroll_control_images='yes')
body="&prefixfilenames._body.html"
contents="&prefixfilenames._contents.html"
page="&prefixfilenames._page.html"
frame="00Frame_&prefixfilenames..html";
Title "Graph 1, Default Data Tips";
proc sgplot data=sashelp.class;
scatter x=weight y=height;
run;
Title "Graph 2, Expanded Data Tips";
proc sgplot data=sashelp.class;
scatter x=weight y=height/tip=(weight height sex);
run;
ods html close;
```

The frame page from the output folder opened and displayed in a browser:



And finally a close-up of the standard data tips (left) and the customized data tips (right). These are displayed when the mouse is brushed-over the points in HTML.



XII. A SPECIAL NOTE ON THE 5TH EDITION OF THE SAS ODS GRAPHICS PROCEDURES GUIDE

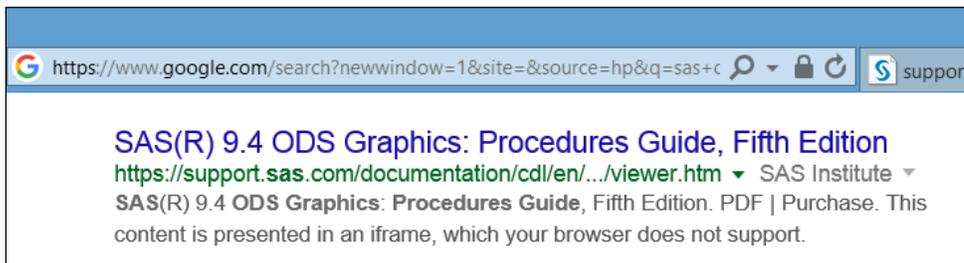
At the time of publication of this paper, the 5th edition of this book was released. Content was increased by a whopping 60%. When search engines such as Google are crawled on this book, only the on-line html versions are returned in the search engine. I find the contents of this HTML approach to often be cryptic at best and very abridged.

The better methodology is to (1) buy the printed version of the book, and (2) download the PDF version of the book to your local machine and used Adobe Acrobat Reader. The PDF version of the book is encrypted and search crawlers are not allowed access to the metadata. Without this, there are no results returned to Google and other engines!

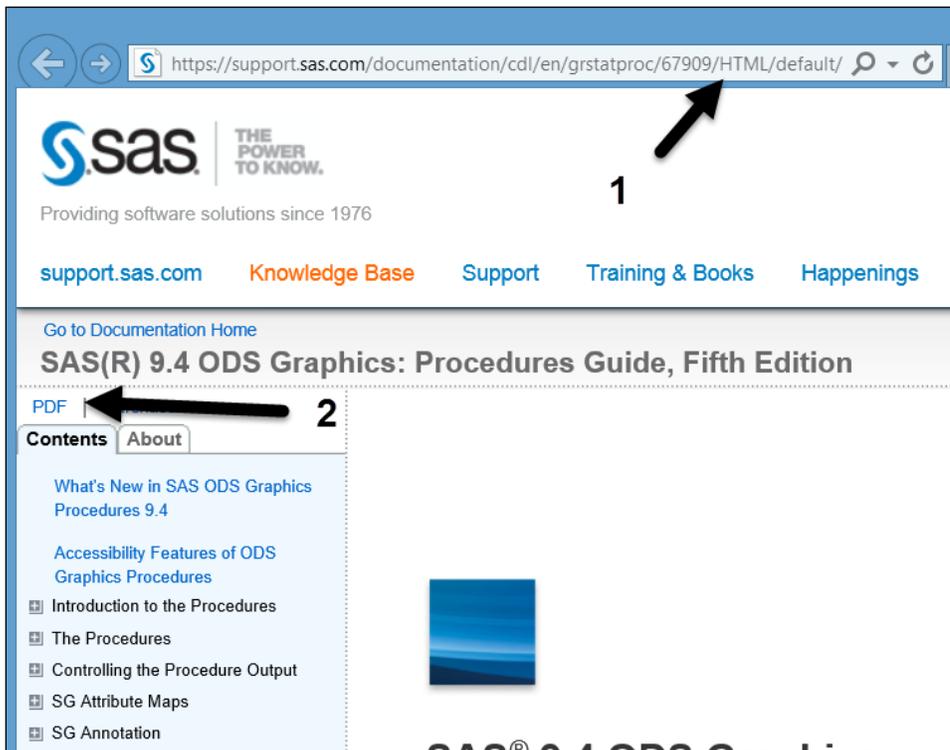
Steps to find, download, save, search and comment with Acrobat Reader are shown in the following figures. Note: Extensive as this book is, it contains only minimal reference to the use of ODS Graphics Designer and the use of Proc Template with Proc SGRend as techniques for graphics development with SAS. These are in other separate books.

To work with this manual in PDF form:

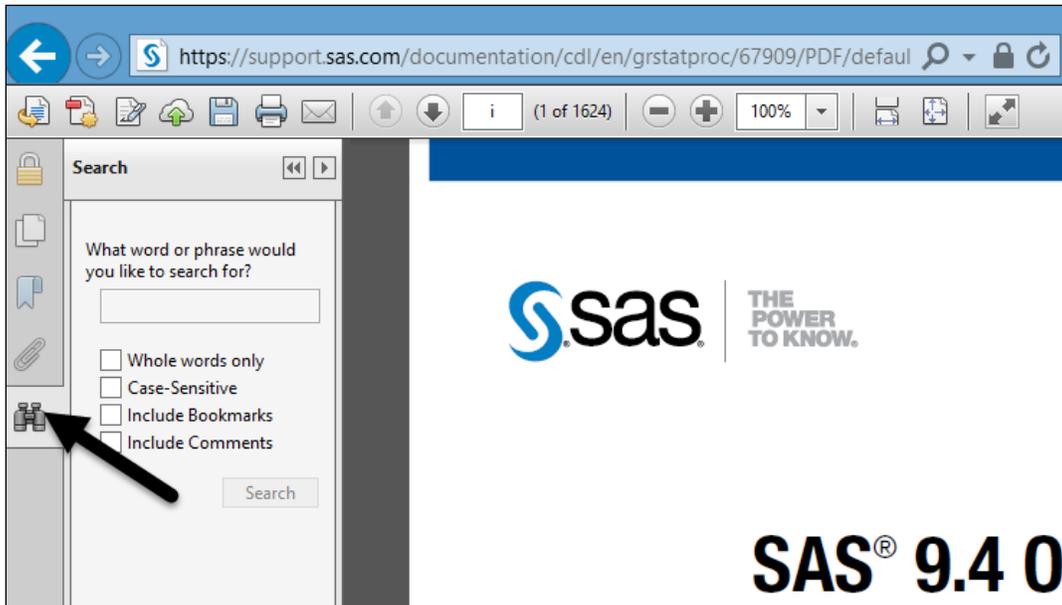
1. Search for the guide in Google, only the HTML version is crawled.



2. Open the returned HTML reference (Arrow 1) and look at the top left corner for the "PDF" link (Arrow 2).

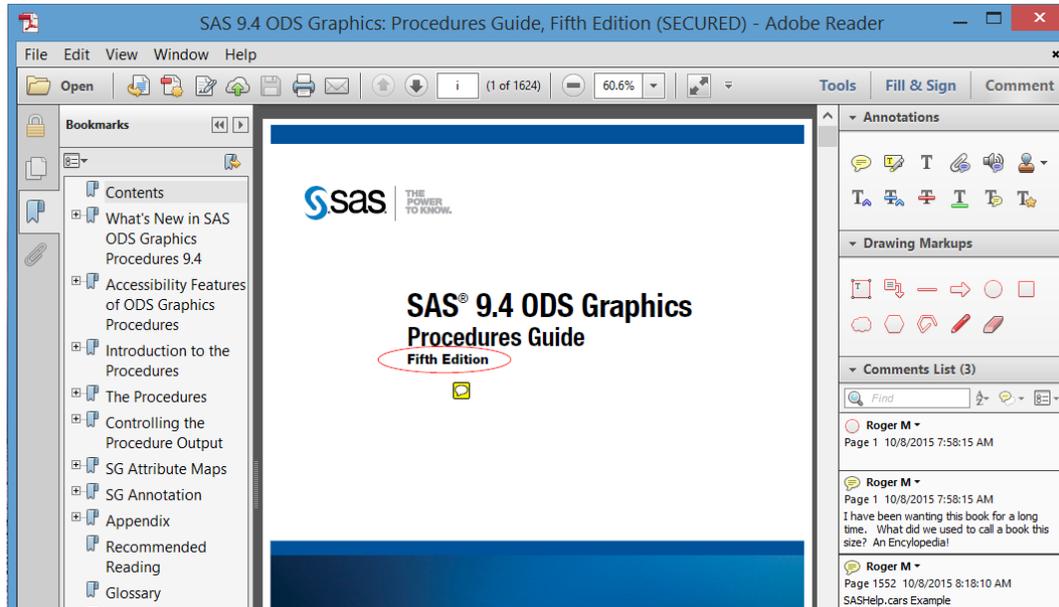


3. **Open the PDF link.** Depending on your browser and how it has been configured by you or your enterprise technicians, some sort of reader will open displaying the document. You may even have some sort of search capability within the document as indicated in this configuration which appears to be a modification of the Adobe Acrobat Reader.



These readers and their configurations which may have their place in certain documents and configurations, can be extremely limited and downright irritating. Download the “real” Adobe Acrobat Reader.

4. **Save the PDF document to your local hard drive.** Remember that you do not have write access to the document when viewing the web version. Yes, SAS allows you to save the PDF to your hard drive and write notes and more in it.
5. **Open with the “Real” Adobe Acrobat Reader.** Note the bookmarks, the comments that you can insert, the highlighting tools, etc. For some reason, “find” does not have an icon, but rather is opened from the Menu with “Edit – Find” or with Ctrl-F. An advanced search is available with “Shift + Ctrl + F”. Your comments, highlights and more are written to the file residing on your hard drive. You can modify the reader to have a find icon.



For more information on using Adobe Acrobat Reader, the use of Adobe search techniques within the document and the insertion of comments, see Muller and Horstman, 2010.
<http://www.mwsug.org/proceedings/2010/resources/MWSUG-2010-120.pdf>

CONCLUSION

Getting off to a good start using SAS ODS Graphics is challenging. Simple examples in this paper will help in that process. The use of the ODS Graphics Designer with its GUI interface will write creative SAS code that will help meet initial needs and will steer you in the right direction. Attribute definitions for things like fonts, fills, colors etc. is tedious when starting. Using Styles as the primary determinant of visual attributes simplifies this process.

ACKNOWLEDGEMENTS

The author would like to thank the following individuals for their efforts: (1) Mr. Donald (D.J.) Penix, President Pinnacle Solutions for guidance in the use of numerous SAS products as they are introduced. (2) All of the behind the scenes programmers at SAS Institute who do a great job bringing these tools forward. (3) Mr. Chris Hemedinger and Mr. Sanjay Matange for the publishing and speaking efforts at meetings, in blogs and in assorted publications to help get users motivated and going in the right direction with new technologies. (4) The outstanding technical support given to users at SAS Institute This paper would not have happened without the guidance from Mr. Martin Mincey and Ms. Lelia McConnell in navigating through issues. (5) All of the volunteers associated with various SAS user groups who make these meetings and publications possible. Having only been distantly involved with the process, there are many hours of time given freely to do this and they are to be admired. They all have real jobs and personal lives to tend to besides this effort.

ADDITIONAL READING

The following are links to books that contain ODS Graphics examples:

- http://www.sas.com/store/books/categories/examples/statistical-graphics-procedures-by-example-effective-graphs-using-sas/prodBK_63855_en.html
- http://www.sas.com/store/books/categories/getting-started/statistical-graphics-in-sas-an-introduction-to-the-graph-template-language-and-the-statistical-graphics-procedures/prodBK_63120_en.html
- http://www.sas.com/store/books/categories/getting-started/getting-started-with-the-graph-template-language-in-sas-examples-tips-and-techniques-for-creating-custom-graphs/prodBK_66550_en.html

Another good resource for examples include:

- <http://support.sas.com/sassamples/graphgallery/index.html>
- <http://support.sas.com/rnd/datavisualization/index.htm>
- <http://blogs.sas.com/content/graphicallyspeaking/>

Attributes and style:

- <http://support.sas.com/documentation/cdl/en/grstatproc/67909/HTML/default/viewer.htm#n02bywwj20lezn15x30wucq9pn8.htm>

REFERENCES

- Delwiche, Lora and Susan Slaughter,. Using PROC SGPLOT for Quick High-Quality Graphs. <http://support.sas.com/resources/papers/proceedings10/154-2010.pdf>
- Welch, Kathy, 2010. Great Graphics Using Proc Sgplot, Proc Sgscatter, and ODS Graphics for SAS®/Stat Procedures. Proceeding MISU Meeting , http://www.misug.org/uploads/8/1/9/1/8191072/kwelch_stat_graphics.pdf
- Schwartz, Susan, 2009. Clinical Trial Reporting Using SAS/GRAPH SG Procedures. 2009 SAS Global Forum. <http://support.sas.com/resources/papers/proceedings09/174-2009.pdf>
- Matange, Sanjay. Google search for “Sanjay Matange SAS ODS SG Graphics” will produce a list of dozens of publications, too numerous to list here.
- Muller, Roger D., Assorted Videos on SAS Topics Including SAS Enterprise Guide.. Available at: SAS Video Playlists by Roger Muller (Google in YouTube).
- Muller, Roger D., 2012. “Program Development with SAS Enterprise Guide® and SAS/Connect® in a Combined PC and Unix Environment” Proceedings of PharmsUG-2012. Available at: <http://pharmasug.org/proceedings/2012/AD/PharmaSUG-2012-AD17.pdf>
- Muller, Roger D. and Donald L Penix, 2012. “SAS Enterprise Guide® - Why and How the Programmer Should Adapt It Now” Proceedings of the 2012 Midwest SAS Users Group. Available at: <http://www.mwsug.org/proceedings/2012/BI/MWSUG-2012-BI08.pdf>
- Muller, Roger D. and Joshua M Horstman, 2010. Custom Google Searches, PDF Sticky Notes and Other Tips for Organizing and Accessing SAS Help Resources. Proceedings 2010 Midwest SAS Users Group, Milwaukee, WI. <http://www.mwsug.org/proceedings/2010/resources/MWSUG-2010-120.pdf>
- Penix, Donald L. and Roger D. Muller, 2012. “Enterprise Guide® – Moving Beyond Your Initial Startup”. Proceedings of the 2012 Midwest SAS Users Group. Available at: <http://www.mwsug.org/proceedings/2012/BI/MWSUG-2012-BI10.pdf>
- Muller, Roger D., 2013a “SAS Enterprise Guide® -- Implementation Hints and Techniques for Insuring Success With the Traditional SAS Programmer”. Proceedings of the 2013 SAS Global Forum, Paper 265 (Video and Proceedings).
- SAS Institute, 2014. SAS 9.4 Graph Template Language User’s Guide, Third Edition, 633 pages. Published by SAS Institute, Cary NC
- SAS Institute, 2013 SAS 9.4 ODS Graphics Procedure Guide, 4th edition 958 pages. Published by SAS Institute, Cary NC
- SAS Institute, 2015 SAS 9.4 ODS Graphics Procedure Guide, 5th edition 1624 pages. Published by SAS Institute, Cary NC

- SAS Institute, 2013. SAS 9.4 Output Delivery System User's Guide, Third Edition. 1129 pages. Published by SAS Institute, Cary NC
- SAS Institute Inc. 2015. SAS® 9.4 ODS Graphics Designer: User's Guide, Third Edition. Cary, NC: SAS Institute Inc.
- SAS Institute Inc. 2015. SAS® 9.4 ODS Graphics Editor: User's Guide, Fourth Edition. Cary, NC: SAS Institute Inc

CONTACT INFORMATION

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

Roger D. Muller, Ph.D.

Data-To-Events, Inc.

14475 Stephanie St.

Carmel, IN USA

317/985-0132 (cell) 317/846-5782

E-mail: rdmuller@hotmail.com

Web: www.data-to-events.com, www.rogermullervideography.com

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.