

## PROC SGPLOT over PROC GPLOT

Shruthi Amruthnath, Experis Business Analytics, Portage, MI

### ABSTRACT

SAS® offers different statistical graphic procedures for data visualization and presentation. SAS® 9.2 brought out a new family of template-based graphical procedures to create high-quality graphics called Statistical Graphics (SG) procedures. These procedures are so powerful that more complex information can be presented effectively with minimal coding.

This paper will focus on employing SGPLOT versus GPLOT; applying SGPLOT to financial data to create reports for trending data, consolidated reports and yearly reports; and managing, displaying and styling procedural output using ODS PDF. The SGPLOT procedure has different types of graphical figures like bar charts, line graphs and scatter plots. This paper explains how to produce line graphs like line plot and area plot by setting different options. In this presentation, each of these topics is illustrated using different techniques with an example.

### INTRODUCTION

SAS® 9.3 offers a wide choice of procedures to create graphical outputs. Several changes were made in SAS/GRAPH 9.3; in particular ODS GRAPHICS has been transferred from SAS/GRAPH to Base SAS.

After ODS GRAPHICS was moved, there were many changes made to the procedures. Most importantly ODS HTML and ODS GRAPHICS are automatically turned on, and the ODS GRAPHICS is set as default output destination. The output destination ODS LISTING has been turned off. This is applicable for Microsoft Windows and UNIX operating systems. In addition SAS/GRAPH also supports RGBA for image compositing to provide effects like translucency. Additionally links to graphical output have been simplified.

### COMPARING PROC SGPLOT VERSUS PROC GPLOT

#### THE SGPLOT PROCEDURE

The SGPLOT procedure produces single cell plots. By using PROC SGPLOT, different types of plots and charts can be produced like basic plots (scatter, series, needle etc.), fit and confidence plots (loess, regression, etc.), distribution plots (box plots, histograms, etc.) and categorization plots (dot plots, bar charts etc.). The SGPLOT procedure has the capability to produce sophisticated graphs more effectively with regression lines, B-Splines and Loess lines. By using SAS ODS facility, the graphs can be presented quickly and attractively.

#### THE GPLOT PROCEDURE

The GPLOT procedure generates a two-dimensional graph which plots two or more variables on a set of horizontal (*x-axis*) and vertical (*y-axis*) axes. The GPLOT procedure also creates different kinds of graphs, but it is limited to producing only scatter plots, overlay plots, logarithmic plots and bubble plots.

#### A SIMPLE EXAMPLE

Let's see a basic example of SGPLOT and GPLOT. Both of the procedures begin with a PROC statement with the name of the procedure and name of the dataset.

```
proc <name of the procedure> data = <name of the dataset> <option(s)>;  
<sas statements>  
run;
```

The example shown below produces a simple scatter plot. For comparison purposes, both PROC SGPLOT and PROC GPLOT are using the same dataset to produce similar graphical output. The SGPLOT procedure provides built-in statements to produce specific types of output. In this example, the scatter statement is used in the SGPLOT procedure to produce a scatter plot output.

On the other hand GPLOT procedure uses a plot statement with plot requests implied in the form y\*x to generate a scatter plot. In both of the procedures, x represents the horizontal axis and y represents the vertical axis. While using this procedure, at least one plot statement must be specified.

In PROC SGPLOT, more than one plot can be combined together whereas in PROC GPLOT, no more than two plot statements are allowed.

Here's an example of a scatter plot using PROC SGPLOT :

```
proc sgplot data=sashelp.shoes;
  scatter x=subsidiary y=stores / group=region;
  where region in ("Africa","Asia","Canada");
run;
```

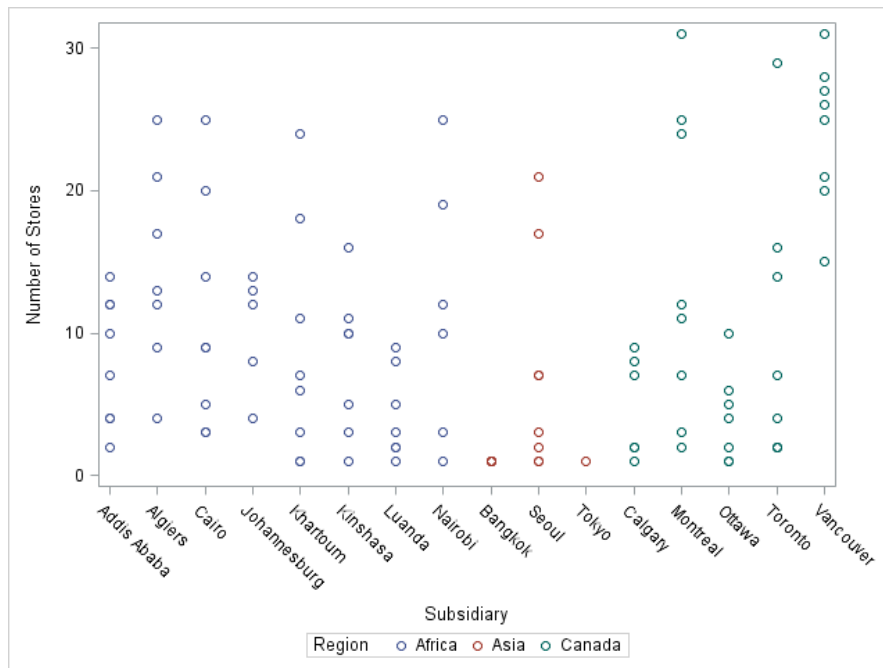
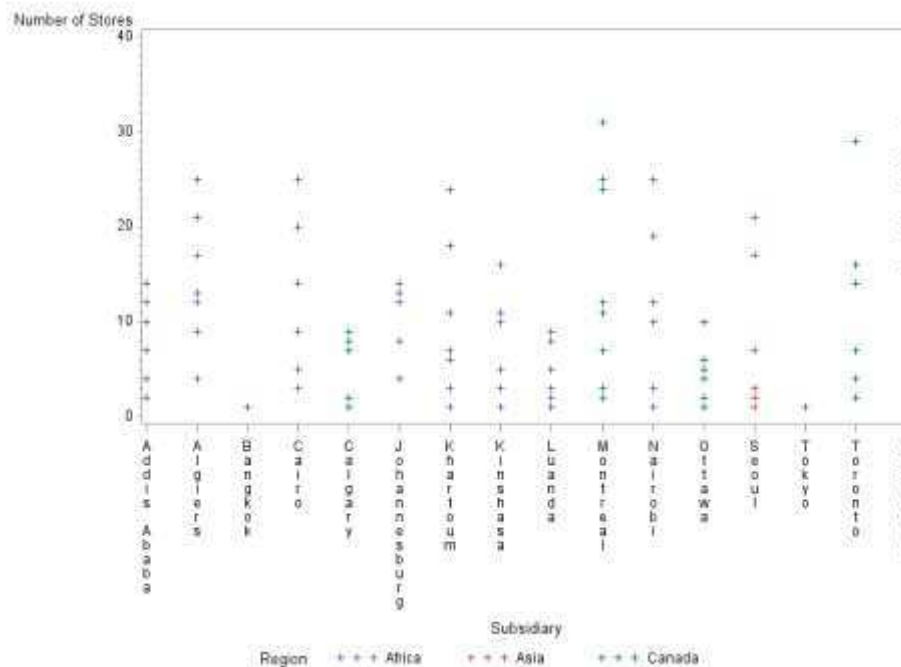


Figure 1. Example of scatter plot using SGPLOT

Here's an example of a scatter plot using PROC GPLOT:

```
proc gplot data=sashelp.shoes;
  plot stores * subsidiary = region;
  where region in ("Africa","Asia","Canada");
run;
```



**Figure 2. Example of scatter plot using GPLOT**

In the above example, the graph which is produced by PROC SGPLOT has a better quality and appearance than the graph produced by PROC GPLOT. The marker shapes and colors are clearer in PROC SGPLOT than in PROC GPLOT. These attributes are easily customizable with PROC SGPLOT, whereas in PROC GPLOT, the attributes for the marker shapes and colors are controlled from outside PROC GPLOT in global SYMBOL statements. Also, as you see in Figure 1, the orientation of x-axis and y-axis are automatically angled in PROC SGPLOT to produce the desired graph. The orientation of axes can be controlled by specifying the options in XAXIS and YAXIS statement. In Figure 2, the orientation of the x-axis is angled to 90 degrees and the y-axis label is specified in the top-left corner. In order to style the axes (e.g. scale, appearance, tick marks and text options), you can declare them using the global AXIS statement in PROC GPLOT.

## TRANSITION FROM PROC GPLOT TO SGPLOT

Many SAS programmers still use the traditional SAS/GRAPH procedures to present graphical outputs. But, it is always a good practice to check the SAS documentation to see if there are any new procedures available before writing a program. SAS/GRAPH offers reusability of programs, but certain options may not be applicable to all outputs. If you are not a SAS/GRAPH user and are trying to create a graph using the GPLOT procedure, then you need to understand several options used in this procedure; or even if you are a SAS/GRAPH user and are trying to reuse code that uses GPLOT, it is may not be easy to interpret it in one step. This is because in the GPLOT procedure, only the PLOT statement is used to produce different kinds of output. If you are not aware of the different options used in the PLOT statement, then it may take a longer to understand it. This results in a delay in coding.

The graphical outputs from GPLOT can be created efficiently using SGPLOT procedure. The transition from using the GPLOT procedure to the SGPLOT procedure can be easily understood because SGPLOT offers built-in statements to create different kinds of graphs. For example, the Figure 1 output generated by SGPLOT used the SCATTER statement to create a scatter plot. Similarly, there are other built in statements like HBAR, VBAR, HBOX, VBOX, and SERIES to present horizontal bar charts, vertical bar charts, box plots or series plots respectively. Also, more sophisticated graphs can be easily created and customized within the SGPLOT procedure.

## USING BUILT-IN OPTIONS IN SGPLOT

It is easy to control the appearance of graphs using PROC SGPLOT (unlike GPLOT) because the options are readily available and can be easily added with PROC SGPLOT and ODS GRAPHICS statements. You don't need to specify the ODS GRAPHICS statement unless and until you want to change the default settings. Whereas in the GPLOT procedure, the appearance of graphical output is controlled each time using the global GOPTIONS statement. Once

the GOPTIONS statement is added in your program it remains in effect until you change, modify or cancel them. In SAS/GRAPH, the options like AXIS, PATTERN, SYMBOLS, NOTE and LEGEND which are used in the GOPTION STATEMENT have been replaced using LINEATTRS=, MARKERATTRS=, and FILLATTRS= options with in each Statistical Graphics procedure.

The examples below each create a three-series plot to present financial data. Consider that you want to observe the pattern of a market share for the year 2005. Here both the SGPLOT and GPLOT procedure use the `stocks` dataset to produce same results. The first step is data preparation where you subset the data from the master dataset. In this case it is `sashelp.stocks`

```

** Data preparation **;
proc sort data=sashelp.stocks out=stocks;
  by date;
  where stock = "IBM" and "01DEC04"d <= date <= "01DEC05"d;
  format date date9.;
run;

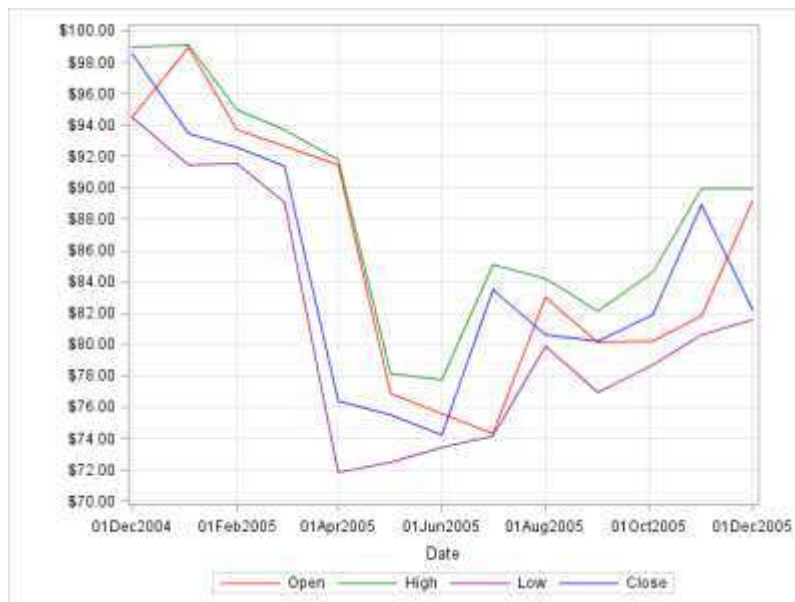
```

Next, the program below generates a three-series plot using the SGPLOT procedure. If you observe, this procedure uses built-in SERIES statements to create the series plot. The properties and styles of graphical output like colors, symbols, thickness and patterns are controlled using the option LINEATTRS= with in this SG procedure . The structure and features the of axes like axis label, grid lines, data type, data range and tick marks are controlled using XAXIS, YAXIS, X2AXIS and Y2AXIS statements.

```

** Create 3 series plot using SGPLOT procedure**;
proc sgplot data=stocks;
  series x = date y = open / lineattrs = (color = red thickness = 1
    pattern=solid);
  series x = date y = high / lineattrs = (color = green thickness = 1
    pattern=solid);
  series x = date y = low / lineattrs = (color = purple thickness = 1
    pattern=solid);
  series x = date y = close / lineattrs = (color = blue thickness = 1
    pattern=solid);
  xaxis label = 'Date' values=("01DEC04"d to "01DEC05"d by month) grid;
  yaxis label = ' ' grid values = (70 to 100 by 2.00);
run;

```



**Figure 3. Example of three series plot using SGPLOT**

The program below generates a three-series plot using the GPLOT procedure. If you observe, this GPLOT procedure produces the same output as the SGPLOT procedure. But the clarity in presentation is much better in the SGPLOT versus the GPLOT. In this example, you can see that the features such as style, color, and patterns are defined in the global GOPTIONS statement. If there are more than one PROC GPLOT, then each time the GOPTIONS statement must be redefined if the PROC GPLOT is not using the options that is specified in previous GOPTIONS statement.

```

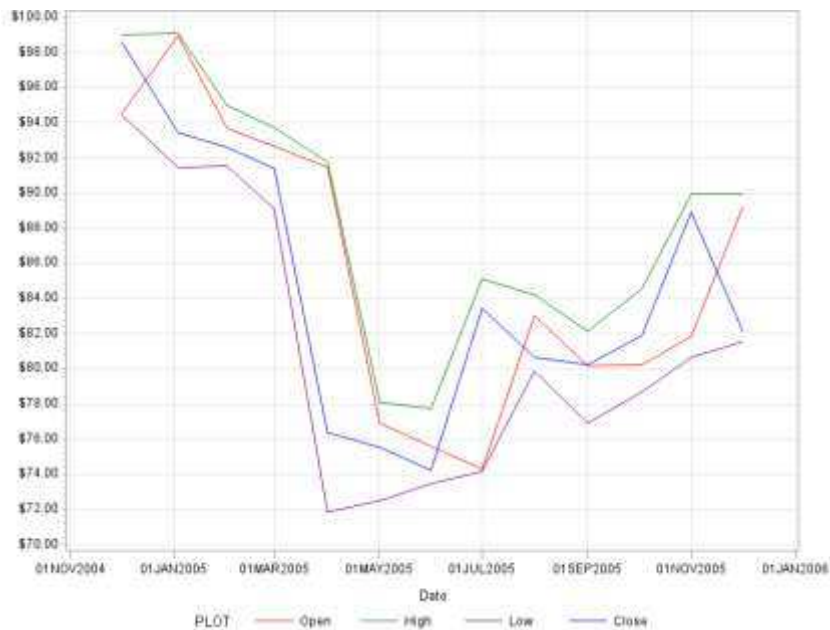
** Create 3 series plot using GPLOT procedure** ;
options reset=all
      border
      colors=(red green purple blue)
      interpol=join ;

proc gplot data= stocks;
  plot open * date high * date low * date close * date /
  overlay
  grid
  VAXIS=AXIS2
  hminor=0
  legend=legend1;
run;

axis2 label= none order=(70 to 100 by 2.00);

run;
quit;

```



**Figure 4. Example of three series plot using GPLOT**

#### REDUCED CODING EFFORT USING SGPLOT

Imagine that you want to create a band plot using the `stocks` dataset. Using SGPLOT you can write this code with just 8 lines of code. You can reduce and control the program length using SGPLOT. By using SGPLOT you can save a lot of valuable time.

```

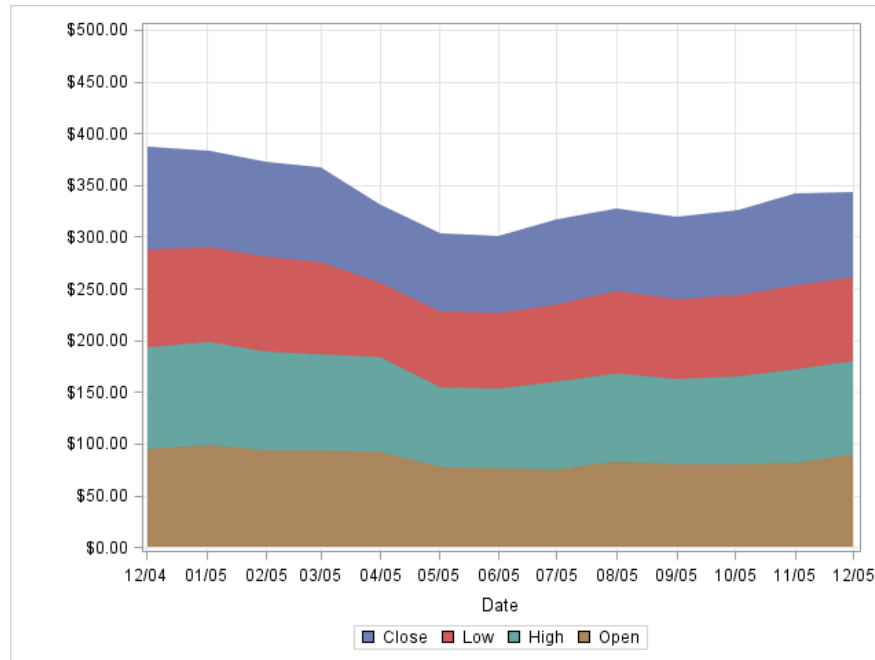
** Create band plot using SGPLOT ** ;
proc sgplot data=stocks;
  band x=date lower=0 upper=close / legendlabel="Close" ;
  band x=date lower=0 upper=low / legendlabel="Low";
  band x=date lower=0 upper=high / legendlabel="High" ;
  band x=date lower=0 upper=open / legendlabel="Open" ;

```

```

axis label = 'Date' values=("01DEC04"d to "01DEC05"d by month) grid;
yaxis label = ' ' grid values = (0 to 500 by 50);
run;

```



**Figure 5. Example of band graph using SGPLOT**

By comparison, the same graph is illustrated using GPLOT. As you can see, there is a lot of effort in coding with the GPLOT procedure. Imagine that you are creating a report that has lot of graphs. If you are using GPLOT, then it takes a lot more time to code. The options defined in global statements such as GOPTIONS, AXIS, and SYMBOL statements must be reset for each PROC GPLOT even when they are different kinds of graphs using different options. Also, it is very difficult to maintain especially when you expect a situation where it needs changes, modifications or any updates.

```

** Create band plot using GPLOT **;
goptions reset=all device=ACTXIMG hsize=6.5 in vsize=4 in
        colors=(brown green orange blue) ftext=arial ftitle=arial;

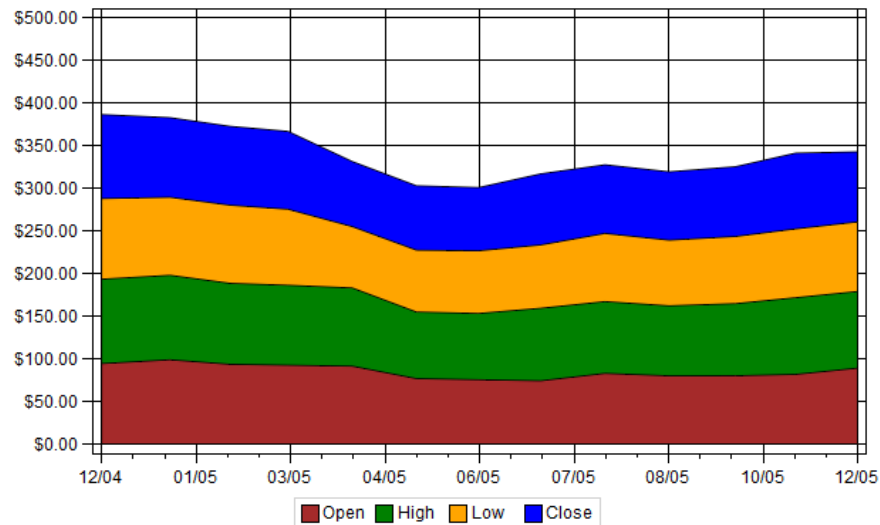
legend1 frame ;
symbol1 interpol=join;
pattern1 value=MS;
axis1
    style=1
    color=black
    width=1
    order=(0 to 500 by 50)
    minor=none
    label= none
;
axis2
    style=1
    color=black
    width=1
    major = (number=9)
    minor = (number=2)
    label=none
;

```

```

proc gplot data=stocks;
  plot open * date high * date low * date close * date /
  overlay
  grid
  areas=4
  frame vaxis=axis1
  haxis=axis2
  legend=legend1;
run;
quit;

```



**Figure 6. Example of band graph using GPLOT**

## ACCESSING SGPLOT OUTPUT

The procedure output from SGPLOT can be managed and accessed individually using the output delivery system. The outcomes of PROC SGPLOT can be saved in a variety of destinations like HTML, LISTING, PDF and RTF. This paper focuses on techniques used to display graphs using ODS PDF destination. The ODS PDF is a part of the PRINTER family of Output Delivery System (ODS) which is used to display high quality procedural output effectively. In ODS PDF, by using ODS GRAPHICS both the tables and graphs can be integrated on the same page. The graphical output can be easily interpreted using ODS GRAPHICS statements which offer more flexibility by coordinating different style elements such as the color, lines, and symbols.

Let's see an example of accommodating both a graph and a table in a PDF file.

```

data stocks;
  set stocks;
  count=1; /* create new variable count for page breaks */
run;

ods graphics / reset width=4in ;

ods pdf file = "MyGraph.pdf"
  pdftoc = 1
  startpage = no
  style = printer
  dpi = 250;

proc sgplot data=stocks;
  band x=date lower=0 upper=close / legendlabel="Close";
  band x=date lower=0 upper=low / legendlabel="Low";

```

```

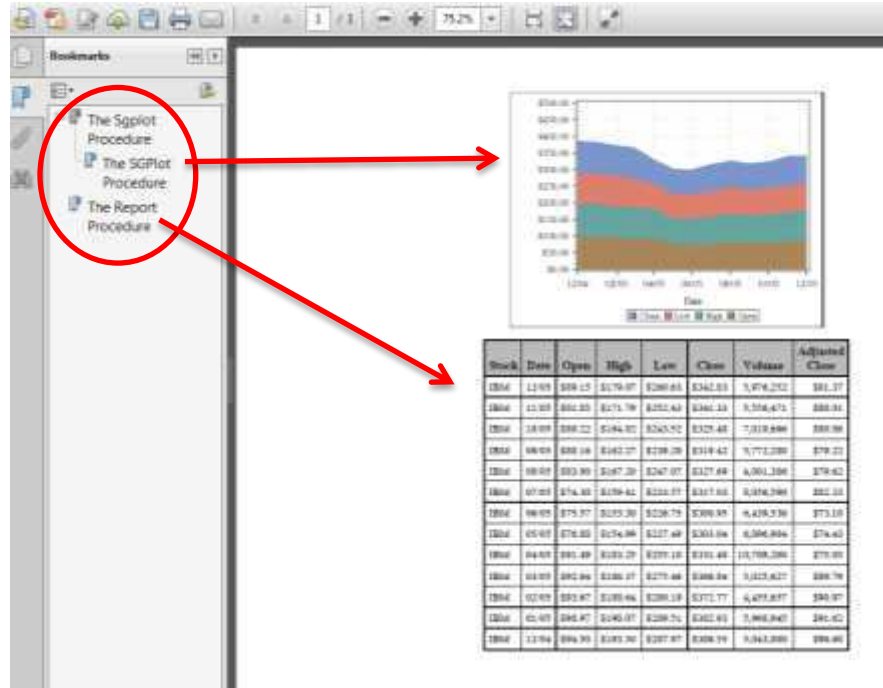
band x=date lower=0 upper=high / legendlabel="High";
band x=date lower=0 upper=open / legendlabel="Open";
xaxis label = 'Date' values=("01DEC04"d to "01DEC05"d by month) grid;
yaxis label = ' ' grid values = (0 to 500 by 50);
run;

proc report data=stocks nowd contents="";
  define count / noprint group;
  break before count / contents="" page;
run;

ods pdf close;

```

The global ODS GRAPHICS statement is used to control many features of graphs. By default the ODS GRAPHICS statement is automatically turned on. But graph features like the size, scaling, aspect ratio and borders can be reset in ODS GRAPHICS statements. In the above example, the ODS GRAPHICS statement sets the width of the graph to 4 inches using WIDTH= option. So the widths of the graphs produced by all PROC SGPlot are changed to 4 inches. Similarly, the height of the graph can be adjusted using HEIGHT= option, borders can be turned ON or OFF using BORDER= option, the image names can be specified using the IMAGENAME= option, data tips can be generated using the IMAGEMAP= option and the image format can be specified using the IMAGEFMT= option.



**Figure 7. Accessing graphs and tables**

In Figure 7, the output produced by PROC SGPlot and PROC REPORT are integrated successfully on the same page. This is possible because of following actions-

- Explicitly adjust width of the graph in the ODS GRAPHICS statement using the option WIDTH=
- In the ODS PDF Statement, specify node expansion using the PDFTOC= option
- Specify page breaks in the ODF PDF statement using the STARTPAGE= option

In Figure 7, the ODS PDF has generated a table of contents for each procedural output. There are two bookmarks linked to the same page - one for the graph and the other for the table; and it looks very confusing. So it's a good



practice to arrange bookmarks neatly for easy navigation especially when you have a report that needs to be submitted to senior management. In order to serve this purpose, the bookmarks generation can be controlled by using the option BOOKMARKGEN= in the ODS PDF statement. You can also use the option NOBOOKMARKGEN if you don't want to generate bookmarks.

The program is modified by specifying options such as BOOKMARKGEN= and NOBOOKMARKGEN= as shown below.

```
data stocks;
  set stocks;
  count=1;
run;

ods graphics / reset width=4in ;
ods pdf file = "Mygraph.pdf"
  pdftoc = 1
  startpage = no
  style = printer
  dpi=250;

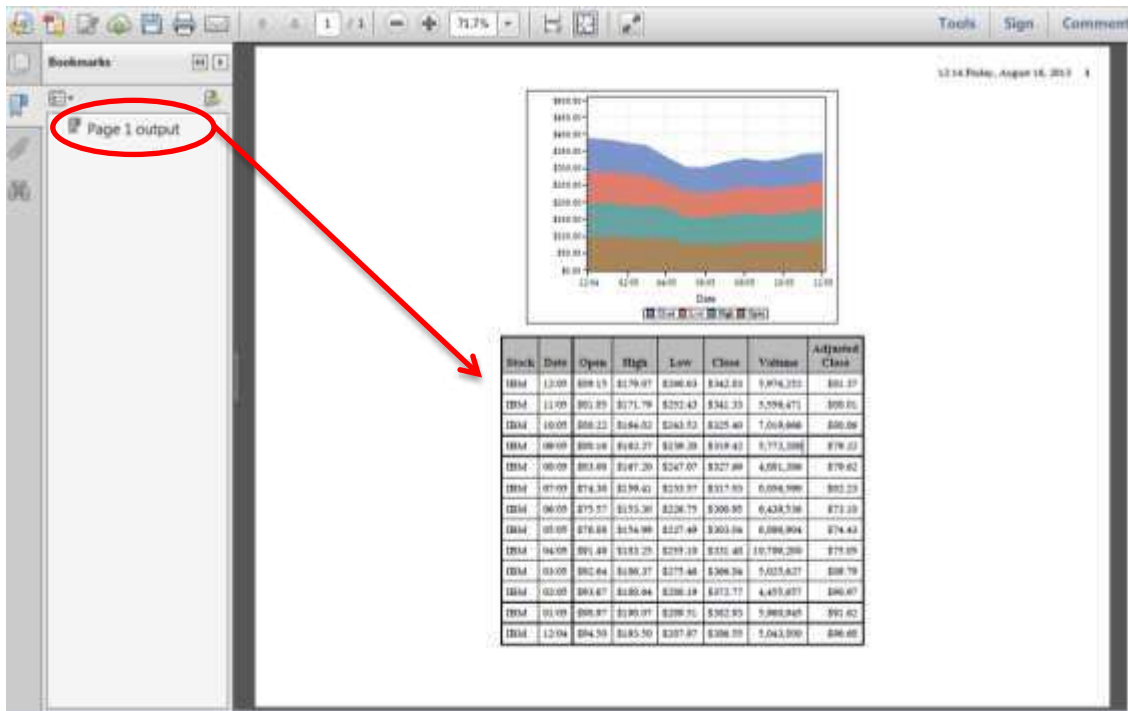
ods pdf nobookmarkgen;

proc sgplot data=stocks;
  band x=date lower=0 upper=close / legendlabel="Close" name="band1" ;
  band x=date lower=0 upper=low / legendlabel="Low" name="band2";
  band x=date lower=0 upper=high / legendlabel="High" name="band3" ;
  band x=date lower=0 upper=open / legendlabel="Open" name="band4" ;
  xaxis label = 'Date' values=("01DEC04"d to "01DEC05"d by month) grid;
  yaxis label = ' ' grid values = (0 to 500 by 50);
run;

ods pdf bookmarkgen;
ods proclabel "Page 1 output";

proc report data=stocks nowd contents="";
  define count / noprint group;
  break before count / contents="" page;
run;

ods pdf close;
```



**Figure 8. Accessing graphs and tables**

In Figure 8, the bookmark is not generated for PROC SGPLOT and it is defined using NOBOOKMARKGEN= in the ODS PDF statement. The bookmark is generated for PROC REPORT and it is defined using BOOKMARKGEN= in the ODS PDF statement. Also, the labels for the procedures can be changed using the ODS PROCLABEL statement. In the figure above, the item name "Page 1 output" in the table of contents is defined in the ODS PROCLABEL statement.

As you see in all of the above examples the output generated from SGPLOT is more captivating and eye catching than the output produced by GPLOT. Because SGPLOT uses ODS GRAPHICS statement, the appearance of these graphs is more uniform than the graphs produced by GPLOT. You can get rid of excessive lines of code by using the SG procedure unless and until you want to explicitly change default settings in your program. By using this method it is easy to maintain the program and convey information more effectively and make necessary changes, modifications or updates with minimal effort.

## CONCLUSION

The techniques which are available in SGPLOT are far more effective than GPLOT. Using this procedure it is easier to locate, update, modify and maintain programs especially when there are several graphs involved. For experienced SAS/GRAPH users the metamorphosis from GPLOT to SGPLOT can be made very easily. For programmers who are new to producing graphs with SAS using the SG procedures will make you look like a pro instantly.

## REFERENCES

- <http://support.sas.com/documentation/cdl/en/grstatproc/62603/HTML/default/viewer.htm#sgplot-ov.htm>
- [http://robslink.com/SAS/papers/sg\\_procs/sgplot.html](http://robslink.com/SAS/papers/sg_procs/sgplot.html)
- Dan Heath, Effective Graphics Made Simple Using SAS/GRAPH® SG Procedures, SAS Global Forum 2008, 255-2008
- Xiangxiang Meng, The Power to Plot: Three Ways to enhance SG Graphical Outputs, SAS Global Forum 2011, 095-2011
- Exploratory and Publication-Ready Graphs Easily Obtained, Using SAS Version 9.2, Agricultural Experiment Station, College of Agricultural, Consumer and Environmental Sciences
- Steven Feder, Integrating Tables and Graphs with ODS LAYOUT, SAS Global Forum 2010, 230-2010

## **ACKNOWLEDGMENTS**

I wish to thank Jack Fuller, Chuck Kincaid, Jared Slain, Scott Davis and Rajesh Lal for their assistance in reviewing this paper and providing valuable comments.

## **RECOMMENDED READING**

SAS/GRAPH ® 9.3 Reference Third Edition

## **CONTACT INFORMATION**

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

Name: Shruthi Amruthnath  
Enterprise: Experis Business Analytics  
Address: 5220 Lovers Lane, Ste 200  
City, State ZIP: Portage, MI 49002  
Work Phone: 269-553-5114  
Fax: 269-553-5101  
E-mail: shruthi.amruthnath@experis.com  
Web: www.experis.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.

Other brand and product names are trademarks of their respective companies.