

Using SAS SG Procedures to Create and Enhance Figures in Pharmaceutical Industry

Shuo Chi & Huashan Huo, PPD Inc., Beijing

ABSTRACT

SAS/GRAPH statistical graphics (SG) procedures provide a simple syntax for creating graphics commonly used in exploratory data analysis and for creating customized statistical displays. This paper will focus on 1) how to use SAS SG Procedures to create pharmaceutical figures, 2) Introduce the advantages and enhancements of SAS SG Procedures, 3) New features in SAS SG 9.3 version to enhance both graphic visualization and printer-friendly result, 4) How to prepare statistics for better and flexible graphic visualization using SAS SG.

INTRODUCTION

From SAS 9.2, “statistical graphics procedures” (SG) has been widely used in programming figures in pharmaceutical area. This article mainly discusses how to use SG procedures, the advantage compared to traditional SAS/GRAPH, new features in SAS 9.3 and above and how to prepare statistics for better and flexible graphic visualization using SAS SG.

From SAS 9.2, a new model called “statistical graphics procedures” (SG) has been introduced to design a variety of statistical plots. These plots vary from the simple kind, such as scatter plot, to complicate kind, such as overlaid panel plots. In SG, all procedures’ name start with “SG” to distinguish the traditional SAS/GRAPH procedures. The procedures include SGPLOT, SGSCATTER, SGPANEL, SGRENDER, SGDESIGN and they are well designed to present figures in pharmaceutical industry.

HOW TO USE SAS SG PROCEDURES

To whom are new to SAS SG procedures and would like to get familiar in a short time, it is a optimal choice to start with using a %sgdesign to enter a GUI interface, drag in the ideal graph model from menu bar, and automatically generate the corresponding code. (Figure 1.)

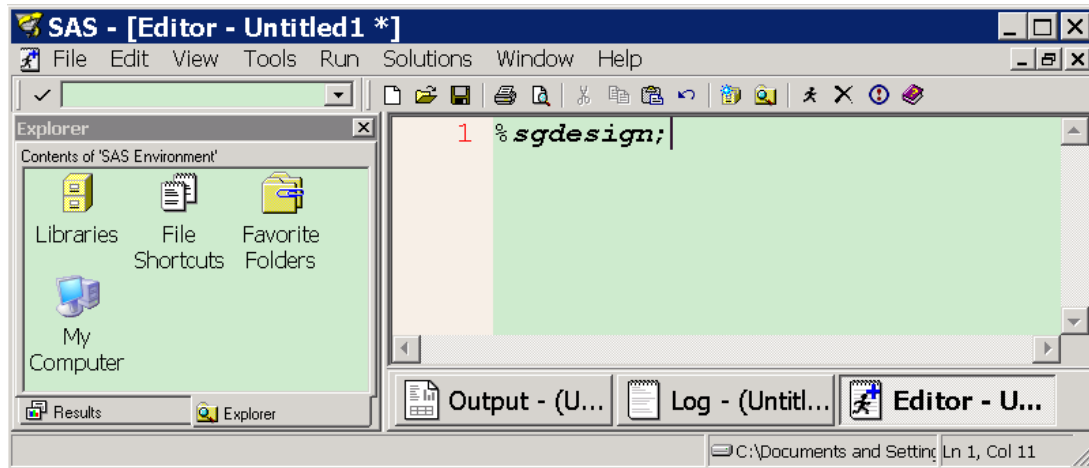


Figure 1. Start a GUI interface with a %sgdesign statement

After run the statement, an interface from ODS Graphics Designer will shown up as Figure 2

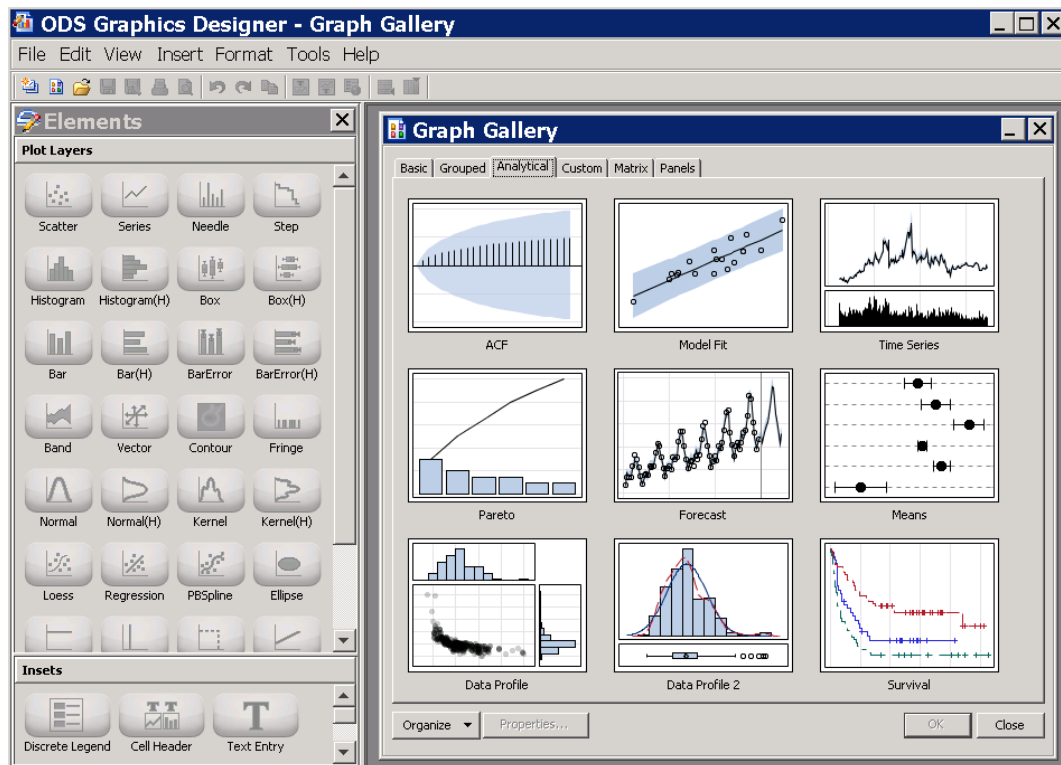


Figure 2. ODS Graphics Designer interface

And then a simple model could be chosen and after fill in some simple parameters, a sample code can be automatically derived.

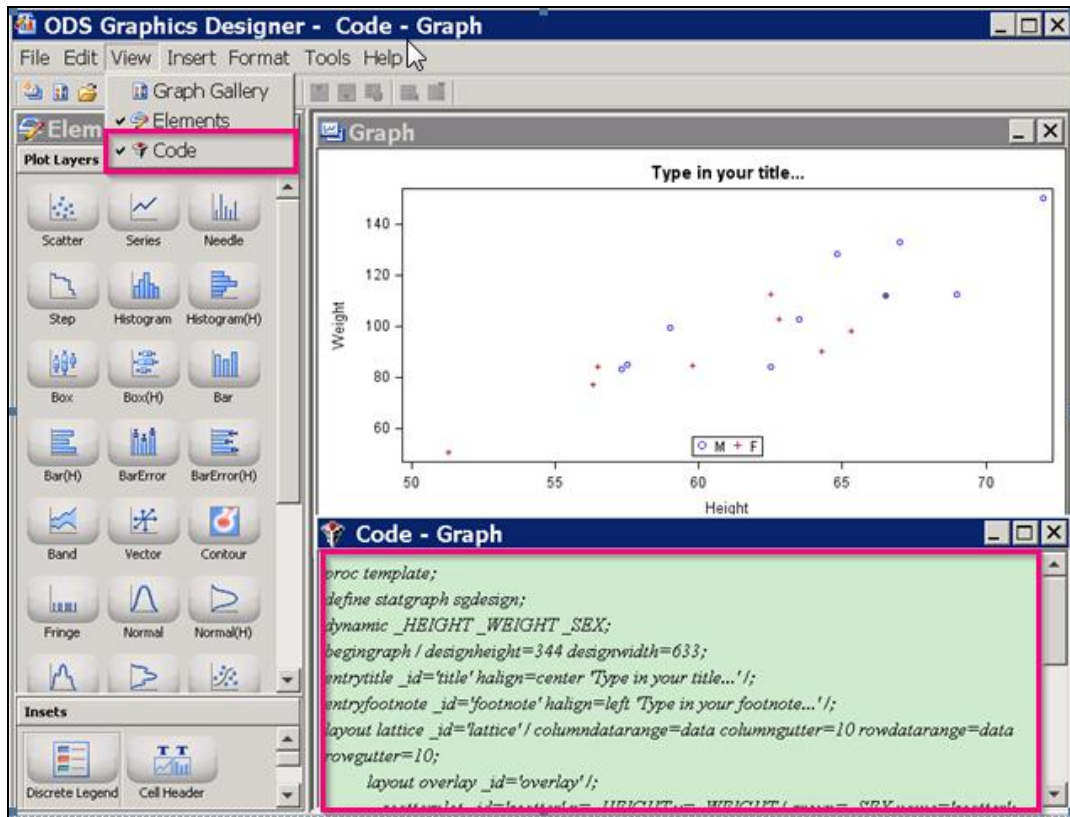


Figure 3. Sample Code generated by ODS Graphics Designer

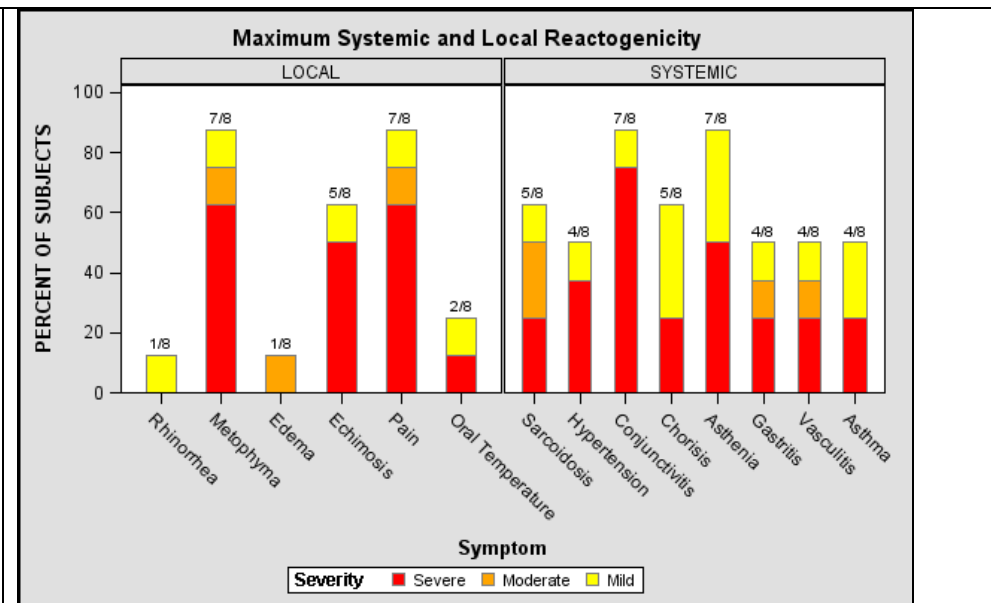
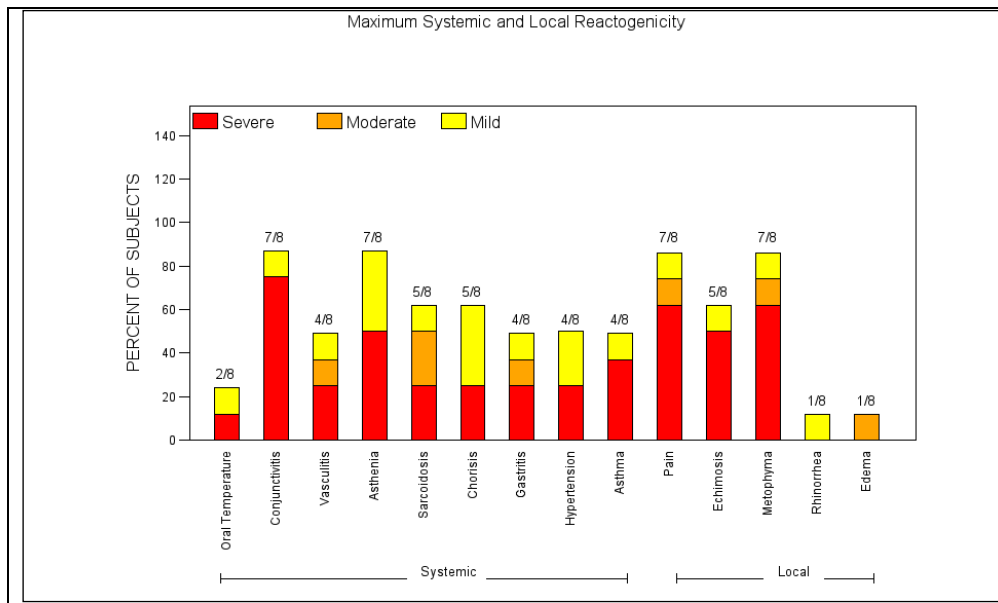
ADVANTAGES TO USE SAS SG PROCEDURES

Compare to the traditional SAS/GRAPH procedures, SG procedures require minimum effort on optimizing the visual appearance, give more flexibility on the layout for either single or multiple plots and make it more doable to generate the more customized combination of figures.

First of all, SG procedures need minimum effort on visual appearance. In traditional SAS/Graph language, title/ footnote/ axis/symbols/ legend and pattern should be pre-defined in GOPTION statement. However, without coding the specific appearance attributes, Graph Template Language (GTL) in SG procedures can provide a sufficient visibility and contrast. GTL stores many plot template with clear visualization

Second, some of the SG procedures support to create multiple plots with a single proc step. In SAS Graphic language, this feature can be achieved by: 1. individually generate each graph 2. use PROC GSLIDE language if any incompatibility between procedures, 3. use a PROC GRREPLY to overlay the graphs in a suitable proportion. Or if any customized annotation needed, this should go through a GANNO procedure. In SG procedures, if the same this procedure can be solved in a more flexible path. If the graph cells are displayed by one or more classification variables, it could be simply carried out by PROC SGPANEL with a Panelby Statement. Or if any annotation needed, it could be accomplished by using an anno dataset or just simple add a variable format. Figure 4 gives an illustration on comparison between a SAS/GRAPH and a SG Procedure on producing a plot with annotation. [1] Mekhala Acharya.

SAS/GRAPH	SG Procedure
<pre> DATA _anno; set class; xsys = '2'; ysys= '2'; x = height; y = weight; function='label'; text=name; position='2'; PROC PRINT; ***GOPTION set up in Appendix 1.*****; PROC GCHART data=; vbar / discrete anno=_anno subgroup=severity type=freq freq=pct space=4 raxis=axis1 maxis=axis2 Legend=legend1; </pre>	<pre> DATA work.sevfmt; length label \$ 8; set work.tmp (keep=pct text rename=(pct=start text=label)) end=__last; retain fmtname "st" type "n"; end=start; output; if __last = 1 then do; hlo = "o"; label = "other"; output; end; run; PROC FORMAT library=work cntlin=work.sevfmt; run; *****GTL language in Appendix 2*****; PROC SGPANEL data=; panelby reacto /novarname uniscale=row; vbar symptom/stat=sum group=severity response=pct datalabel; rowaxis tickvalueformat=best12. values=(0 to 100 by 20) label='PERCENT OF SUBJECTS'; label severity='Severity' symptom='Symptom'; format symptom \$symp. severity \$sev. pct st.; run; </pre>



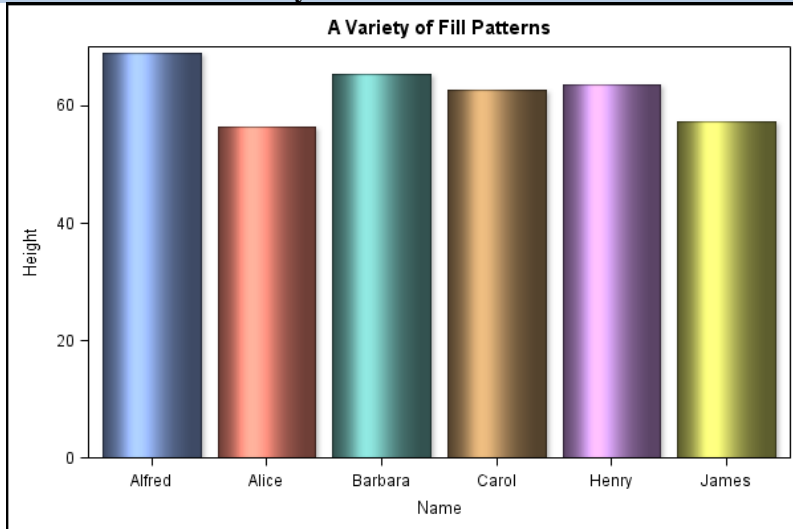
Third, SAS SG procedures are more convenient to produce the customized combination of statistical plots. This feature relies on its GTL language to provide a customized template and SGRENDER to display the template on output.

NEW FEATURES IN SAS 9.3 AND ABOVE

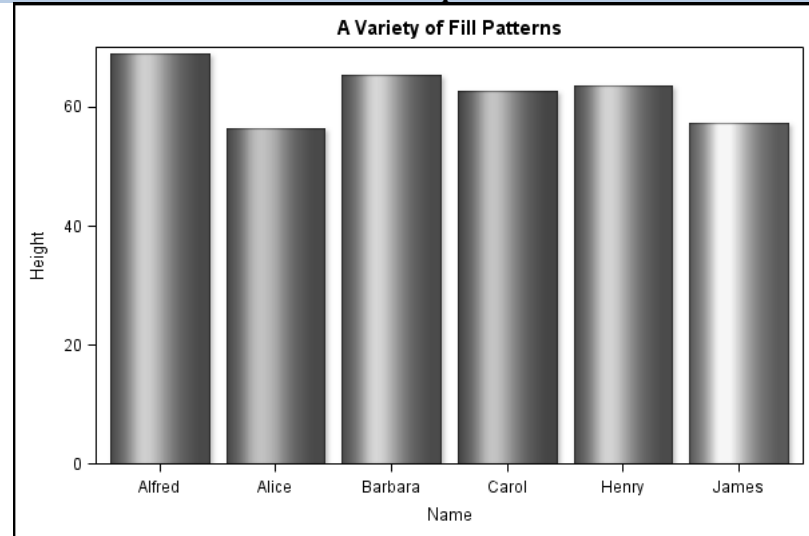
After SAS 9.2, every edition provides many new features. These features are added to not only simplify the statistical figure generation, but also enhance both graphic visualization and printer-friendly result. BUBBLE/ WATERFALL statements are added to create bubble plot and waterfall chart easier. HIGHLOW/ LINEPARM statement are introduced to implement more statistical requirements. Now we take FILLPATTERN option as a specific example to show the enhancement of SAS 9.3 in graphic visualization.

In SAS 9.2, SG procedures allows to produce a distinguished group chart by using different colors. This method make the chart quite clear and uncluttered when you reviewing on desktop layout.(Figure 5.) However, the visualization seems poor when printed out in black and white. The fill patterns appear for grouped data in the following types of plots, whether generated using the SGPLOT or the SGPANEL procedure: HBAR and VBAR / HBARPARM and VBARPARM statement. Code for FILLPATTERN is attached in Appendix 3.

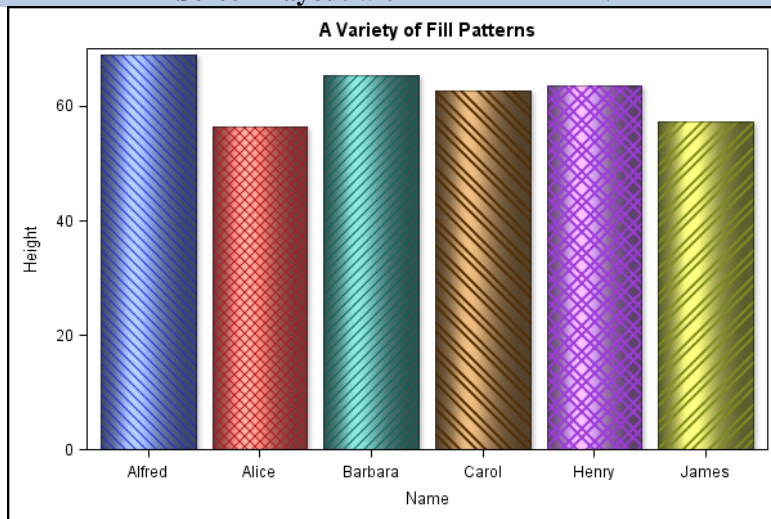
Screen Layout without FILLPATTERN



Black and White Print Mock-up without FILLPATTERN



Screen Layout with FILLPATTERN



Black and White Print Mock-up with FILLPATTERN

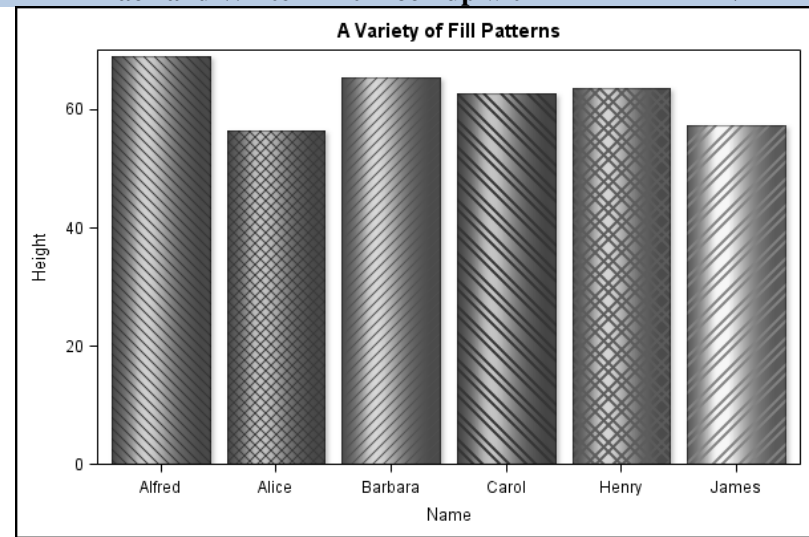


Figure 4. FILLPATTERN Features

HOW TO PREPARE STATISTICS FOR BETTER AND FLEXIBLE GRAPHIC VISUALIZATION USING SAS SG

SG procedures and some of SAS/GRAPH procedures support statistical analysis in some statement.

However, in the practice in pharmaceutical programming, instead of using the statistical analysis features in SG statement, we need to calculate the statistics first to give a better and flexible graphic visualization. The most common example in the industry is the plots with mean \pm standard deviation over different time point.

Figure 6 gives an illustration on a series plot over time. In this study, there are 4 treatment groups: Placebo, Drug A, Drug B and Drug C. In case of overlapping the 95%CI line at each time point, the most common method is to pre-process the X values to move a little leftward or rightward. However, in SG procedures, when we define template during SCATTERPLOT statement, the value can be shift 0.15 unit left by using `x=eval(xc-0.15)`, then all lines are clearly displayed without any clutter or overlap. Beside of that, SG procedures can also distinguish the time frames by using a BLOCKPLOT statement, especially for study with many visits. The upper limit and lower limit of the confidence interval can be generated in GTL SCATTERPLOT statement by using parameter YERRORLOWER/YERRORUPPER.

There are advantages of using this method, 1) all statistics calculation are not restricted by the SG procedures' algorithm 2) the statistical analysis and plot generation can be separated, which make it flexible to create more complicated plots. And that will also benefit for the tasks need to be independently validated. These statistics can be output and validated by the corresponding validator, which makes the result more reliable.

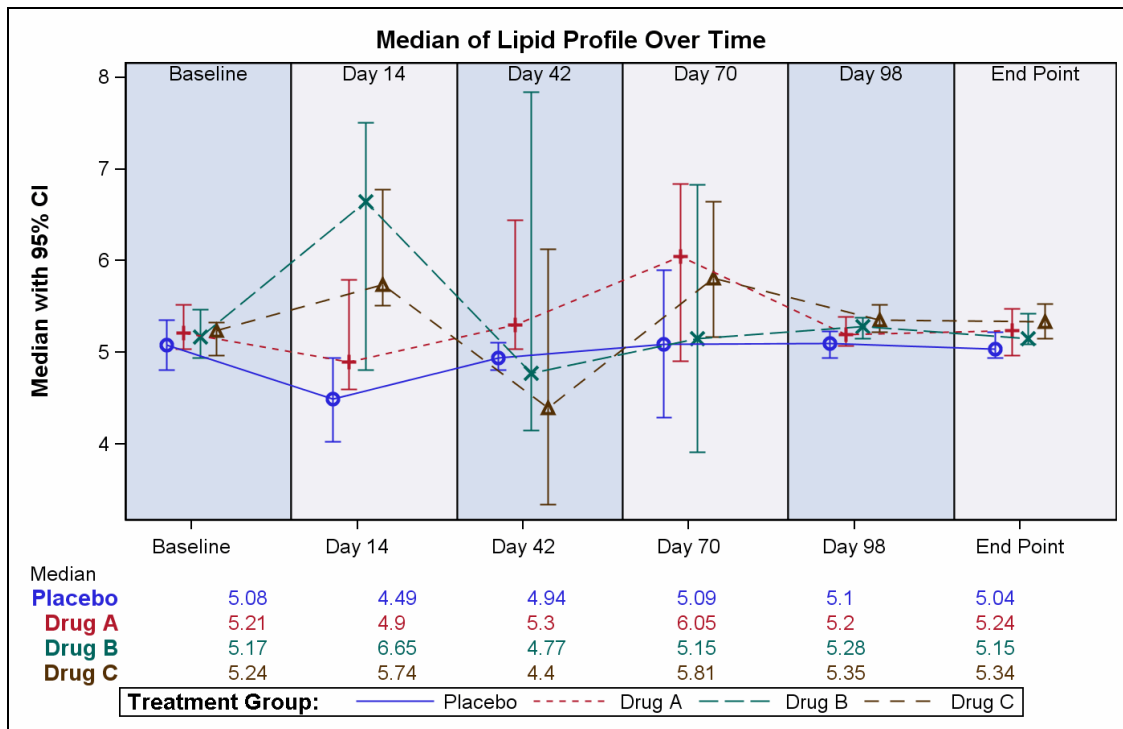


Figure 6. Series Plot over Time

/* Prepare data with statistics values */

data lipid;

label a_med="Drug A" b_med="Drug B" c_med="Drug C" p_med="Placebo";

input by \$ 1-7 x \$ 8-17 xc a_med a_lcl a_ucl b_med b_lcl b_ucl

c_med c_lcl c_ucl p_med p_lcl p_ucl;

datalines;

Test 1 Baseline 1 5.21 5.04 5.52 5.17 4.94 5.47 5.24 4.97 5.33 5.08 4.81 5.35

Test 1 Day 14 2 4.90 4.60 5.79 6.65 4.81 7.51 5.74 5.51 6.78 4.49 4.03 4.94

Test 1 Day 42 3 5.30 5.04 6.44 4.77 4.15 7.84 4.40 3.34 6.13 4.94 4.81 5.11

Test 1 Day 70 4 6.05 4.91 6.84 5.15 3.91 6.83 5.81 5.17 6.65 5.09 4.29 5.90

Test 1 Day 98 5 5.20 5.07 5.39 5.28 5.15 5.38 5.35 5.22 5.52 5.10 4.94 5.23

Test 1 End Point 6 5.24 4.97 5.48 5.15 5.09 5.42 5.34 5.15 5.53 5.04 4.94 5.22

;

run;

/* Create the template for the graph */

proc template;

define statgraph lipid_profile;

dynamic title;

begingraph / designwidth=7in designheight=4.5in backgroundcolor=CXFFFFFF;;

entrytitle title;

layout lattice / columndatarange=union rowweights=(0.8 .04 .04 .04 .04 .04);

layout overlay / cycleattrs=true yaxisopts=(label='Median with 95% CI' griddisplay=on)

xaxisopts=(offsetmin=0.00 offsetmax=0.07 display=(line ticks tickvalues))

linearopts=(tickvaluelist=(1 2 3 4 5 6))

tickdisplaylist=('Baseline' 'Day 14' 'Day 42' 'Day 70' 'Day 98' 'End Point'));

blockplot _id='block' block=x x=eval(xc-0.4)/filltype=alternate name='block' display=(outline fill values)


```

        valuealign=top valuehalign=center labelposition=top;
scatterplot x=eval(xc-0.15) y=p_med / yerrorlower=p_lcl yerrorupper=p_ucl name='ps'
        markerattrs=graphdata1(size=9px weight=bold)
        errorbarattrs=graphdata1(pattern=solid thickness=1);
scatterplot x=eval(xc-0.05) y=a_med / yerrorlower=a_lcl yerrorupper=a_ucl name='as'
        markerattrs=graphdata2(size=9px weight=bold)
        errorbarattrs=graphdata2(pattern=solid thickness=1);
scatterplot x=eval(xc+0.05) y=b_med / yerrorlower=b_lcl yerrorupper=b_ucl name='bs'
        markerattrs=graphdata3(size=9px weight=bold)
        errorbarattrs=graphdata3(pattern=solid thickness=1);
scatterplot x=eval(xc+0.15) y=c_med / yerrorlower=c_lcl yerrorupper=c_ucl name='cs'
        markerattrs=graphdata4(size=9px weight=bold)
        errorbarattrs=graphdata4(pattern=solid thickness=1);
seriesplot x=eval(xc-0.15) y=p_med / lineattrs=graphdata1(pattern=solid thickness=1px) name='pl';
seriesplot x=eval(xc-0.05) y=a_med / lineattrs=graphdata2(pattern=shortdash thickness=1px) name='al';
seriesplot x=eval(xc+0.05) y=b_med / lineattrs=graphdata3(pattern=mediumdash thickness=1px) name='bl';
seriesplot x=eval(xc+0.15) y=c_med / lineattrs=graphdata4(pattern=dash thickness=1px) name='cl';
endlayout;
layout overlay;
    entry halign=left 'Median';
endlayout;
blockplot x=eval(xc-0.35) block=p_med / display=(values label) valuehalign=left
        label='Placebo' repeatedvalues=true valueattrs=graphdata1 labelattrs=graphdata1;
blockplot x=eval(xc-0.35) block=a_med / display=(values label) valuehalign=left
        label='Drug A' repeatedvalues=true valueattrs=graphdata2 labelattrs=graphdata2;
blockplot x=eval(xc-0.35) block=b_med / display=(values label) valuehalign=left
        label='Drug B' repeatedvalues=true valueattrs=graphdata3 labelattrs=graphdata3;
blockplot x=eval(xc-0.35) block=c_med / display=(values label) valuehalign=left
        label='Drug C' repeatedvalues=true valueattrs=graphdata4 labelattrs=graphdata4;
sidebar / spacefill=false;
    discretelegend 'pl' 'al' 'bl' 'cl' / title='Treatment Group: ' across=4;
endsidebar;
endlayout;
endgraph;
end;
run;

/* Create graphs using the template and data. */

ods listing close;
ods html image_dpi=200 file='LipidProfile.html' path='d:\';
ods graphics / reset noborder width=700px height=450px
        imagename='ClinicalHandout_LipidProfile' imagefmt=gif noscale;

proc sgrender data=lipid template=lipid_profile;

```

```
dynamic title="Median of Lipid Profile Over Time";  
run;  
  
ods html close;  
ods listing;
```

CONCLUSION

SG procedures provide the programmers a different pathway other than traditional SAS/GRAPH language. SG procedure is an easy start-up language and makes the code more efficiency to produce more visualized graphs. SAS/GRAPH language is the extremely useful and versatile annotate system, but SG procedures make the multi-cell combination plots and customized graphs more doable, depending on its powerful GTL and SGRENDER languages. And the new features in SAS 9.3 and above provide the enhancement in layout output.

APPENDIX:

1. Code for GOPTION in Figure 4

```
goptions reset=all;  
goptions reset=goptions ftext="arial" rotate=portrait  
htext=3 hsize=10 in vsize=6 in  
gsfmode=replace prompt gunit=pct gsfname=figs device=png;  
filename figs "figname.png";  
legend1 label=none value=(justify=l h=3.0 'Severe' 'Moderate' 'Mild')  
position=(top left inside) across=3 shape=bar(2.5,2.5) ;  
pattern1 v=solid c=red;  
pattern2 v=solid c=orange;  
pattern3 v=solid c=yellow;  
axis1 order=(0 to 150 by 20) major=(h=1) minor=none  
label=(h=3.0 angle=90 'PERCENT OF SUBJECTS') v=(h=2.0) offset=(, 5 pct);  
axis2 label=none order=(1 to 14) offset=(, 5 pct)  
value=(angle=90 rotate=0 h=2.0  
' Oral Temperature' 'Conjunctivitis' 'Vasculitis' 'Asthenia' 'Sarcoidosis' 'Chorisis'  
'Gastritis' 'Hypertension' 'Asthma' 'Pain' 'Echimosiis' 'Metophyma' 'Rhinorrhea' 'Edema');  
Title1 "Maximum Systemic and Local Reactogenicity";  
Title4 ' ';
```

2. Code for GTL in Figure 4

```
PROC TEMPLATE;  
  define style styles.mystyle;  
    parent=styles.default;  
    style graphdata1 from graphdata1 / contrastcolor=gray color=red;  
    style graphdata2 from graphdata2 / contrastcolor=gray color=orange;  
    style graphdata3 from graphdata3/contrastcolor=gray color=yellow;  
  end;  
run;
```

3. Code for FILLPATTERN in Figure 5.

```
proc template;
  define statgraph regress;
    beginnograph;
      entrytitle 'A Variety of Fill Patterns';
      layout overlay;
        barchart x=name y=height / group=name display=(fillpattern fill outline) dataskin=sheen;
      endlayout;
    endnograph;
  end;

  /* Fill patterns are defined using the FILLPATTERN style element attribute. */
  define style styles.mypatterns;
    parent=styles.listing;
    style GraphData1 from GraphData1 /
      fillpattern = "L1";
    style GraphData2 from GraphData2 /
      fillpattern = "X1";
    style GraphData3 from GraphData3 /
      fillpattern = "R1";
    style GraphData4 from GraphData4 /
      fillpattern = "L5";
    style GraphData5 from GraphData5 /
      fillpattern = "X5";
    style GraphData6 from GraphData6 /
      fillpattern = "R5";
  end;
run;

data class;
  set sashelp.class;
  if _n_ < 7;
run;
ods html style=mypatterns;
ods graphics / reset border width=600px height=400px;

proc sgrender data=class template=regress;
run;
```

REFERENCES

[1] Mekhala Acharya. Annotate your SG Procedure Graphs,2011

[2] SAS online document: SAS National Language Support (NLS): Reference Guide.

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

ACKNOWLEDGEMENT

We would like to thank PPD and PharamSUG Conference Committee for providing us the opportunity to present this paper. Dr. Xianyi Kong and Dr. Jianrong Li kindly read the draft paper and provided valuable comments. Their work is also highly appreciated.

We also wish to thank the staff from SAS Institute Inc. for their best customer support!

CONTACT INFORMATION

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

Name: Shuo Chi, Huashan Huo

Enterprise: PPD Inc.

Address: 8/F, Tower B, Central Point Plaza No 11, Dongzhimen South Ave. Dongcheng District

City, State ZIP: Beijing, 100007

Work Phone: +86 10-57636323 / +86 10-57636250

Fax: +86 10-57636251

E-mail: shuo.chi@ppdi.com huashan.huo@ppdi.com

Web: www.ppdi.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.