PharmaSUG China 2023 - Paper PO-129

Common Methods to Plot Multiple Graphs on One Page Using SAS/GRAPH® V9 with Examples

Jieqiong (Jay) Tang, Everest Clinical Research

ABSTRACT

To facilitate effective analysis and comparisons of clinical trial data, it often becomes necessary to plot multiple graphs on a single page. Recognizing the availability of various viable approaches, this paper focuses on exploring common methods utilized in SAS/GRAPH® V9 to achieve this goal, accompanied by practical examples and insights. These techniques span from older approaches, such as employing PROC GREPLAY to reorganize plots stored in graph catalog, to newer methods, such as utilizing the PROC SGRENDER procedures to generate graphical output from templates created with the Graph Template Language (GTL). By implementing these common methods in SAS/GRAPH V9, the intended audience can be equipped with the necessary tools to optimize their data visualizations, ultimately leading to increased efficiency and effectiveness of their data analysis processes.

INTRODUCTION

This paper will focus on the following four methods of plotting multiple graphs on one page two different kinds of practical examples to compare these techniques and vividly illustrate their weaknesses and strengths.

Methods	Description
ODS Layout - Absolute Layout	Create expected ODS layout using absolute values
ODS Layout - Relative Layout	Create expected ODS layout using relative values
Proc SGRENDER with Proc TEMPLATE	The SGRENDER procedure produces graphical output from templates that are created with the Graph Template Language (GTL).
Proc GREPLAY	Create old-style graph catalog using Proc GPLOT and then use PROC GREPLAY.

Table 1. Methods and Description Used in This Paper

The reasoning behind plotting multiple graphs on a single page encompasses two primary approaches: creating individual graphs for each frame and subsequently reorganizing them on the page; or creating a page with multiple frames and plotting a graph within each frame. The first approach is facilitated by Proc GREPLAY and ODS Absolute Layout, while the second approach is supported by ODS Relative Layout and Proc SGRENDER.

MULTIPLE GRAPHS ON ONE PAGE

The Bland-Altman plot (Figure 1) will be used as an example in this paper to compare these four methods. Bland-Altman plots, also known as difference plots, are effective graphical tools for comparing two measurement techniques and evaluating the agreement between two datasets (in this case, a test for the two eyes). These plots visually represent the difference (referred to as DIFF in the example codes) between two measurements on the y-axis and the average (referred to as MEAN in the example codes) of the two measurements on the x-axis. Additionally, the plot will include a side-by-side BOXPLOT of the DIFF and MEAN values to show patterns. PARAMCD represents the eye tested, and the values are OD or OS.

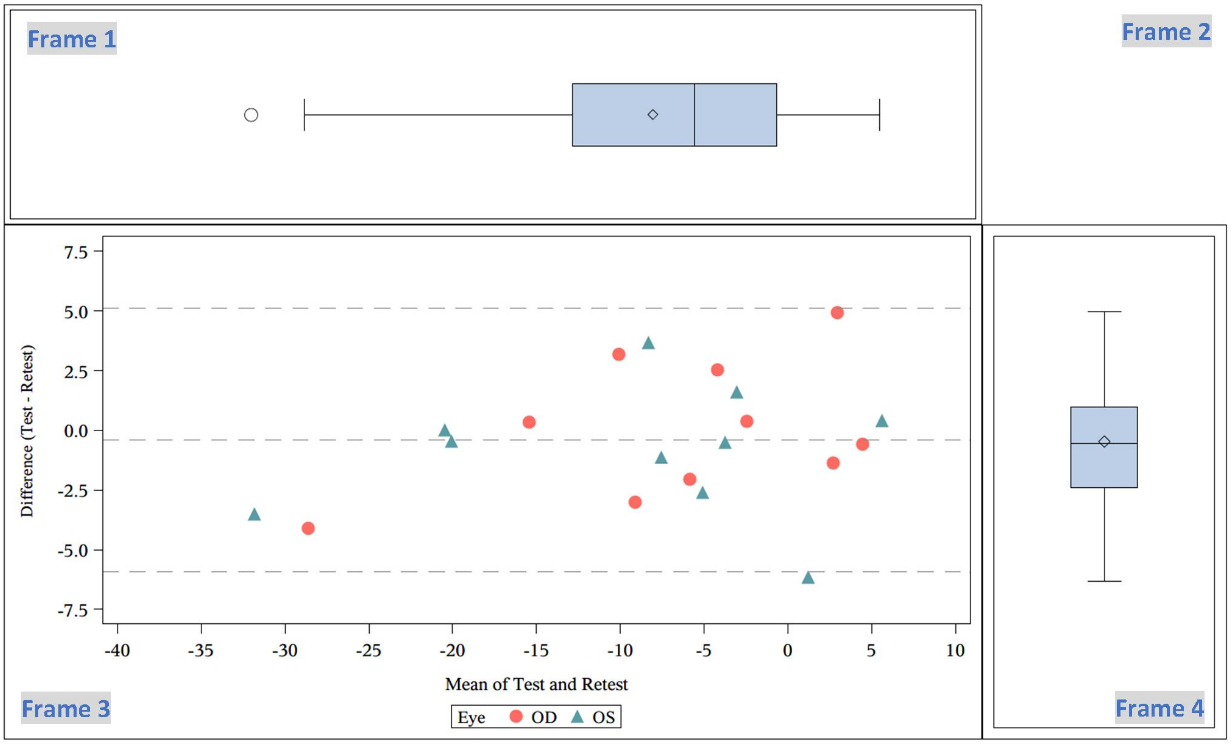


Figure 1. Bland-Altman Plot for Test-Retest Average

ODS LAYOUT

ODS LAYOUT can be utilized in two ways: Absolute Layout and Relative Layout. Both types of LAYOUT involve dividing a page into regions where SAS generates output. The Absolute Layout operates similarly to PROC GREPLAY, where regions are defined by specifying the starting coordinates (X and Y) and the size of the region as percentages of the total horizontal and vertical space on the page. While the Relative Layout operates in a manner similar to PROC SGRENDER, where regions are divided into columns and rows, and then graph is placed in the specific Column/Row locations.

EXAMPLE 1. ABSOLUTE LAYOUT

```
ods pdf file="./absolute.pdf" style=newstyle;
ods graphics on/width=100% height=100%;

ods layout start width=10in height=6in;

ods region x=0 y=0 width=80% height=30%;                    Frame 1
proc sgplot data=FINAL;
        hbox MEAN;
        xaxis display=none offsetmin=0.25 offsetmax=0.1;
run;

ods region x=0 y=30% width=80% height=70%;                  Frame 3
proc sgplot data=FINAL;
        scatter x=MEAN y=DIFF/group=PARAMCD name='S' JITTER;
        xaxis label="Mean of Test and Retest" values=(-40 to 10 by 5);
        yaxis label="Difference (Test - Retest)" values=(-7.5 to 7.5 by 2.5);
        discretelegend 'S' / title="Eye";
        refline &MDIFF. / axis=Y;
```

```
        refline &UPPER. / axis=Y;
        refline &LOWER. / axis=Y;
run;
```

Frame 4

```
ods region x=80% y=0 width=20% height=100%;
proc sgplot data=FINAL noborder;
        vbox DIFF;
        yaxis display=none offsetmin=0.2 offsetmax=0.4;
run;

ods layout end;
ods pdf close;
ods graphics off;
```

In this example, a PDF document with three frames on one page is created, following the ODS Layout Absolute format. The process begins with an ODS LAYOUT START statement, where the page size is defined using the WIDTH and HEIGHT options. ODS REGION statements are then used before each output to position it within a specific section of the page. The X and Y coordinates determine the upper-left corner of the region, corresponding to ULX and ULY in PROC GREPLAY. The size of the region is specified as a percentage of the total width and height. Finally, the ODS layout definition is concluded with an ODS LAYOUT END and ODS PDF CLOSE statement.

EXAMPLE 2. RELATIVE LAYOUT

ODS Relative layout functions similarly, with the distinction that frame number, size and gutter are assigned immediately after the ODS LAYOUT START statement. In the given example, four frames are created, but only three of them are occupied. It's important to note that the number of frames corresponds to the total number of columns multiplied by the total number of rows.

```
ods pdf file="./relative.pdf" style=newstyle;
ods graphics on/width=100% height=100% ;

ods layout start width=10in height=6in columns=2 rows=2
column_widths=(8in 2in) row_heights=(1.8in 4.2in)
column_gutter=0 row_gutter=0 ;
```

Frame 1

```
ods region row=1 column=1;
proc sgplot data=FINAL;
        hbox MEAN;
        xaxis display=none offsetmin=0.25 offsetmax=0.1;
run;
```

Frame 3

```
ods region row=2 column=1;
proc sgplot data=FINAL;
        scatter x=MEAN y=DIFF/group=PARAMCD name='S' JITTER;
        xaxis label="Mean of Test and Retest"  values=(-40 to 10 by 5);
        yaxis label="Difference (Test - Retest)" values=(-7.5 to 7.5 by 2.5);
        discretelegend 'S' / title="Eye";
        refline &MDIFF. / axis=Y;
        refline &UPPER. / axis=Y;
        refline &LOWER. / axis=Y;
run;
```

Frame 4

```
ods region row=2 column=2;
proc sgplot data=FINAL;
        vbox DIFF;
        yaxis display=none  offsetmin=0.3 offsetmax=0.15;
```

```
run;
ods layout end;
ods pdf close;
ods graphics off;
```

PROC SGRENDER

In the aforementioned methods (Example 1 and Example 2), the graphs are placed on a single page without strict alignment (OFFSETMIN and OFFSETMAX is used to make it look like alignment) . However, when it is desired to align the corresponding axes of each graph, it is recommended to use the combination of Proc SGRENDER with Proc TEMPLATE (Example 3). The SGRENDER procedure generates graphical output using templates created with the Graph Template Language (GTL). In other words, Proc TEMPLATE is used to create the graph template (referred to as the StatGraph template), which includes the frame and then Proc SGRENDER renders the data based on that template.

In this example, the process begins with the DEFINE STATGRAPH statement. A PDF document with four frames on one page is created by COLUMNS and ROWS, following the LAYOUT LATTICE format. The page size is defined using the ROWWEIGHTS and COLUMNWEIGHTS options, while the frame gutter is defined using the ROWGUTTER and COLUMNGUTTER options. The size of each frame is specified as a percentage of the total width and height. Each frame begins with a LAYOUT OVERLAY statement and ends with an ENDLAYOUT statement. The frame on the page is created from left to right, and then from top to bottom (sequentially from Frame 1 to Frame 4). Finally, the ODS layout definition is concluded with an ENDGRAPH and END statement.

Similar to ODS Relative Layout, the number of frames in ODS SGRENDER Layout corresponds to the total number of columns multiplied by the total number of rows. However, unlike ODS Relative Layout, it is not possible to skip any of the frames. Therefore, in order to maintain the desired layout, an invisible boxplot is created in frame 3 (set all colors to white).

EXAMPLE 3. PROC SGRENDER

```
ods pdf file="./sgrender.pdf" style=NEWSTYLE;
ods graphics on/width=100% height=100%;

proc template;
define statgraph BLTMAN;
begingraph;
layout lattice /columns=2 rows=2 rowweights=(.2 .8) columnweights=(.7 .3)
                        rowgutter=0 columngutter=0;

/* define Frame 1 - row 1 column 1 */                    <==========  Frame 1
layout overlay/
        xaxisopts=( display=none type=linear
                linearopts=( tickvaluesequence=(start=-40 end=10 increment=5) tickvaluefitpolicy=none
                tickvaluepriority=true))
        yaxisopts=(display=none);
        boxplot y=MEAN/ orient=horizontal;
endlayout;

/* define Frame 2 - row 1 column 2*/                 <==========  Frame 2
layout overlay/
        xaxisopts=( display=none )
        yaxisopts=( display=none );
    boxplot y=mean/ orient=horizontal DISPLAY=(fill)
                OUTLINEATTRS=(color=white) FILLATTRS=(color=white)
                WHISKERATTRS=(color=white);
endlayout;
```

Frame 3

```
/* define Frame 3 - row 2 column 1*/
layout overlay/
        xaxisopts=(label="Mean of Test and Retest"  type=linear
                linearopts=( tickvaluesequence=(start=-40 end=10 increment=5)
                tickvaluefitpolicy=none tickvaluepriority=true))
        yaxisopts=(label="Difference (Test - Retest)" type=linear
        linearopts=( tickvaluesequence=(start=-7.5 end=7.5 increment=2.5)
                tickvaluefitpolicy=none tickvaluepriority=true));
        scatterplot x=MEAN y=DIFF /group=PARAMCD name='S' jitter=auto;
        referenceline y=&MDIFF. /lineattrs=(pattern=dash);
        referenceline y=&UPPER. /lineattrs=(pattern=dash);
        referenceline y=&LOWER. /lineattrs=(pattern=dash);
        discretelegend 'S'/ title="Eye";
endlayout;
```

Frame 4

```
/* define Frame 4 - row 2 column 2*/
layout overlay/
        xaxisopts=( display=none ) yaxisopts=( display=none type=linear
                linearopts=( tickvaluesequence=(start=-7.5 end=7.5 increment=2.5)
                tickvaluefitpolicy=none tickvaluepriority=true));
        boxplot y=DIFF /orient=vertical;
endlayout;

endlayout;
endgraph;
end;
run;

ods graphics / noborder;
proc sgrender data=FINAL template=BLTMAN;
run;
ods graphics off;
ods pdf close;
```

PROC GREPLAY

If you choose to utilize older style graphics procedures like Proc PLOT or Proc BOXPLOT, which utilize a graph catalog to store and modify the graphs, you can employ PROC GREPLAY to plot multiple graphs on a single page. Once you have created the graph catalog, you have the option to use either the built-in templates or create your own template. SAS provides several built-in templates, and the TC (template catalog) option could refer to the built-in templates found in SASHELP.TEMPLT or your customized templates. The code (Example 4) provided showcases a template that I have built for the given example (Figure 1).

In the template, the frames are identified as Frame 1 to 4. Each frame is created by specifying the four corners. For example, in Frame 1, ULX represents the upper-left limit for the X (horizontal) corner. In this case, the frame starts at a horizontal position of 0 (ranging from 0 to 100 from left to right on a page) and a vertical position of 70 (ranging from 0 to 100 from bottom to top on a page), which is the top-left corner of the document. This frame extends from the top-left corner to 80% of the width (X=80) and 30% of the height (Y=30%). The IGOUT option specifies the graphics catalog to be used for the input graphs. The TEMPLATE statement indicates the name of the template to be used. Additionally, the TREPLAY statement plays a crucial role in determining which graph should be placed in each box, based on the graph names. Therefore, it is essential to accurately specify the names of the graphs for proper placement within the template.

EXAMPLE 4. PROC GREYPLAY

```
proc greplay nofs igout=work.temp tc=tempcat;
tdef tpd des='four frames template'
1/      llx=0 lly=70
        ulx=0 uly=100
        urx=80 ury=100
        lrx=80 lry=70
        color=black
```

Frame 1

```
2/      llx=80 lly=70
        ulx=80 uly=100
        urx=100 ury=100
        lrx=100 lry=70
        color=black
```

Frame 2

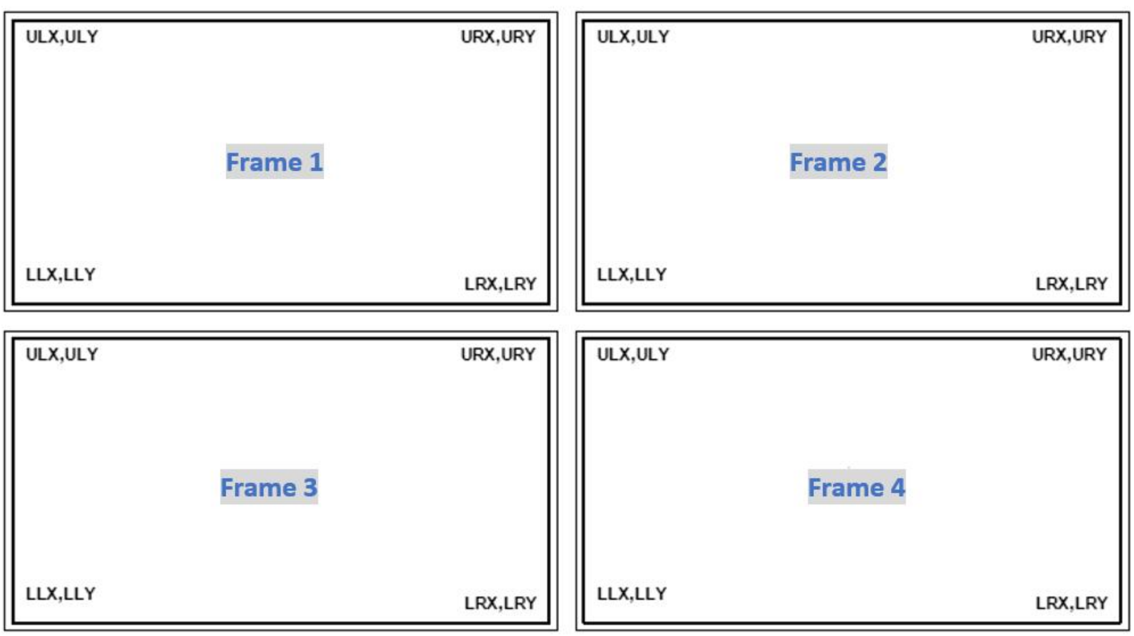
```
3/      llx=0 lly=0
        ulx=0 uly=70
        urx=80 ury=70
        lrx=80 lry=0
        color=black
```

Frame 3

```
4/      llx=80 lly=0
        ulx=80 uly=70
        urx=100 ury=70
        lrx=100 lry=0
        color=black
```

Frame 4

```
;
run;
template tpd;
treplay 1:boxplot1 2:fake 3:scatterplot 4:boxplot2;
DELETE _ALL_;
```




```
run;
```

Figure 2 Frame used for PROC GREPLAY

CONCLUSION

This paper offers readers the common methods to generate multiple graphs on a single page. By utilizing these examples, it is hoped that readers will be able to generate multiple graphs on a single page and devise their own solutions for creating visually appealing pages with multiple graphs.

CONTACT INFORMATION

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

Jieqiong (Jay) Tang
Everest Clinical Research
Unit 115, Building 1, 169 Sheng Xia Lu
Pudong New District, Shanghai, China 201210
Phone: (+86) 173-01895525
E-mail: jay.tang@ecrscorp.com

ACKNOWLEDGEMENT

I would like to thank Mr. Kirk Bateman for his reviewing and valuable comments.