Multiple Graphs on One Page
(Step-by-step approach)
Yogesh Pande, Schering-Plough Corporation, Summit NJ

ABSTRACT

In statistical analysis and reporting, it is essential to provide a clear presentation of analysis results. To achieve this goal, the appearance of figures plays a very important role in data presentation. The most common SAS procedure for generating figures is PROC GPLOT. However, it is necessary to provide a simple solution for the high demand of presenting multiple graphs on one page due to the limitation of GPLOT procedure. The purpose of this paper is to provide user friendly step-by-step instructions in generating multiple graphs on one page by using SAS procedure, PROC GREPLAY. This paper uses features of SAS/GRAPH and its procedures. The explanation is good for O.S. UNIX, Windows Xp. The skill set of the intended audience should be from intermediate to advance level.

Key Words: graph, figure, PROC GPLOT, PROC BOXPLOT, PROC GREPLAY, PROC MEANS

1. DEFINITIONS AND KEY WORDS IN SAS GRAPHS

Catalog: While creating SAS graphs using PROC GXXXX or PROC BOXPLOT if the programmer uses GOUT option, he/she is creating a graphic catalog.

Example: proc gplot data=sashelp.test gout=<catalog-name>;

Catalog-name: This catalog-name can be stored in which ever library the user wants to put the catalog. If the library name is not specified it will store those in temporary library “WORK”.

Example: gout=test.graph1 (the graph1 catalog will be stored in library test, incase “test.” is not written then graph1 catalog will be stored in temporary library “work”)

Goptions: When the user makes graphical catalog, it becomes necessary to use the Goptions “nodisplay”, but CAUTION, if the user has made graphical catalog and if using goptions nodisplay, it becomes necessary for the user to make use of proper “Device”.

Example: goptions nodisplay device=WIN

Note: If you write all these examples together, the initial step to create SAS GRAPH catalog will be seen as

```
Goptions nodisplay device=WIN;
Proc gplot data=sashelp.test gout=test.graph;
```

This is how the graphical catalog is created. One more important thing about catalog is, the file stored in the catalog “Graph” cannot be overwritten by another catalog file having same name.

For example: PROC G PLOT DATA=INPUT.TEST GOUT=TEST.GRAPH;
    Where order=1;
    Plot xxx*yyy ;
    RUN;
    PROC G PLOT DATA=INPUT.TEST GOUT=TEST.GRAPH;
    Where order=2;
    Plot xxx*yyy ;
    RUN;

Explanation: In this example even if there are two files of similar names but different plots (as order is different in the where clause) are stored in graph graphic catalog. The catalog will not overwrite these files but the catalog will store this into two files namely gplot and gplot1.

Note: The example shown above and the dataset shown above “INPUT.TEST” is just to simplify the explanation of how catalog files are stored in graphical catalog. The “order”, “xxx” and “yyy” in the where clause are the variables in INPUT.TEST.

2. HOW GREPLAY HELPS IN CREATING MULTIPLE PLOTS?

First step: The very first step before we start thinking of how to make 2 or 3 or 4 plots on one page is to check whether the graphical catalog is created properly. The user should also make sure, by what names and how the catalog files are stored in the graphic catalog.
For example: Catalog "graph (of above example)" has graphical catalog files as `gplot`, `gplot1`, `gplot2` and `gplot3`. The user has to be sure of the names, because while using PROC GREPLAY user needs to specify the exact catalog file name, just to replay those specific graphs.

**Second step:** The user has to decide whether he/she wants to create his own template or he/she can use available graphical templates in SASHELP.TEMPLT.

For example: There are many available templates, few are as given in the Table 1

<table>
<thead>
<tr>
<th>Templates in SASHELP.TEMPLT</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>H2</td>
<td>Horizontal (1 box left, 1 box right)</td>
</tr>
<tr>
<td>H3</td>
<td>Horizontal (3 box across)</td>
</tr>
<tr>
<td>L1R2</td>
<td>1 box left 2 boxes right</td>
</tr>
<tr>
<td>L2R2S</td>
<td>2 boxes left and 2 boxes right with space</td>
</tr>
<tr>
<td>L2R2</td>
<td>2 boxes left and 2 boxes right without space</td>
</tr>
</tbody>
</table>

Table 1: Available templates in SASHELP.TEMPLT.

**Third and final Step:** Once the template is decided, if it's the available template then the user needs to just call the template within PROC GREPLAY and make multiple plots on one page (that's our goal). The example below elaborates how to create user-defined graphical template and how multiple plots are constructed on one page.

### 3. EXAMPLES

**Example-1:** The user is trying to plot four plots (line plots) on one page. He/she doesn't want to use available templates, but wants to create user-defined template.
- Create 4 frames (in this example I am creating 4 plots on one page) on one page, these 4 frames will display 4 graphs in a very organized manner.
- These frames can be created by PROC GREPLAY. Please see the code at Fig. 1.

```sas
GOPTIONS DISPLAY;
proc greplay nofs igout=work.TEST tc=tempcat;
tdef spec4 des='four panel template'
  1/ LLX=0      LLY=0
      ULX=50     ULY=50
      URX=50     URY=50
      LRX=50     LRY=0
      COLOR=black
   Frame 1
  2/ LLX=0      LLY=50
      ULX=0      ULY=100
      URX=50     URY=100
      LRX=50     LRY=50
      COLOR=black
   Frame 2
  3/ LLX=50     LLY=50
      ULX=50     ULY=100
      URX=100    URY=100
      LRX=100    LRY=50
      COLOR=black
   Frame 3
  4/ LLX=50     LLY=0
      ULX=50     ULY=50
      URX=100    URY=50
      LRX=100    LRY=0
      COLOR=black
   Frame 4
;template spec4;
treplay 1:grep1 2:grep1t 3:grep2 4:grep3
Delete _ALL_;
Run;
Quit;
```

Figure 1: PROC GREPLAY user defined template code
Explanation:

```plaintext
proc greplay nofs igout=work.TEST tc=tempcat;
   tdef spec4 des='four panel template'
```

NOFS = Suppress the PROC GREPLAY screen to appear.
IGOUT = The graphic catalog, where the graphs are stored.
TC = The name where user defined template "spec4" will be stored.
DES = The description of the "spec 4" created template < optional>

The user defined Template "tdef", has four frames and PROC GREPLAY enables the user to define those four frames. The frames are defined by the parameters as explained below. (Please look at the above code while reading the definition of these parameters)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLX</td>
<td>Lower Left X-axis</td>
</tr>
<tr>
<td>LLY</td>
<td>Lower Left Y-axis</td>
</tr>
<tr>
<td>ULX</td>
<td>Upper Left X-axis</td>
</tr>
<tr>
<td>ULY</td>
<td>Upper Left Y-axis</td>
</tr>
<tr>
<td>URX</td>
<td>Upper Right X-axis</td>
</tr>
<tr>
<td>URY</td>
<td>Upper right Y-axis</td>
</tr>
<tr>
<td>LRX</td>
<td>Lower Right X-axis</td>
</tr>
<tr>
<td>LRY</td>
<td>Lowe Right Y-axis</td>
</tr>
</tbody>
</table>

I) The value written in front of these parameters LLX, LLY is in accordance with the A4 page size. This means that LLX=0 will be the point at the origin of the X-axis and LLY=0 will also be the point at the origin of the Y-axis. Thus the Frame 1 first point will be (0, 0). Similarly other points makes FRAME 1 complete (please see figure 2)

![Frame 1 of PROC GREPLAY](image)

**Figure 2: Frame construction of PROC GREPLAY**

II) The "Frame 1" of PROC GREPLAY covers the 50% of x-axis and 50% of Y-axis.

III) Similarly all other frames (Frame 2, Frame 3 and Frame 4) are constructed by the user defined template "spec4", as seen below.
The example show below uses the sample dataset; please click the link to download the dataset, in order to better understand the code written below.

http://rs301.rapidshare.com/files/202402733/data.txt

Example:

PROC FORMAT;
  value days
    0='Base'
    1='Day 1'
    2='Day 2'
    3='Day 3'
    4='Day 4'
    5='Day 5'
    6='Day 6'
    7='Day 7'
;
RUN;

*------------------------------------------------------------------------
Note: Removing all the formats (if any) associated with the variables assigning format to the 'Days' variable which will be x-axis for the plots-------------------------------------------------------------;

DATA sample;
  SET current.show; *--Download this dataset to better understand the code;
  FORMAT _ALL_;
  FORMAT days days.;;
RUN;

*-----Sorting the data set by sequence, days and group each player belongs to------;

PROC SORT;
  BY sequence days group;
RUN;

*------------------------------------------------------------------------
Note: Taking the median of the 'Scores' given by the trainer during the training (This median of the 'Scores' variable will be plotted)
------------------------------------------------------------------------;

PROC MEANS DATA=sample NOPRINT;
  BY sequence days group;
  VAR score;
  OUTPUT OUT=testing1 (DROP=_TYPE_ _FREQ_) MEDIAN=MEDIAN;
RUN;
PROC MEANS DATA=testing1 NOPRINT;
BY sequence;
VAR MEDIAN days;
OUTPUT OUT=testing2 MIN= Y_MIN X_MIN MAX= Y_MAX X_MAX;
RUN;

*------------------------------------------------------------------------
Creating MACRO variables for ranges of various 'Sequence'. As the plots are by 'Sequence' these macro variables will give an idea about the minimum and maximum values that are seen on X-axis and Y-axis
------------------------------------------------------------------------;

data _null_;   
set testing2;   
call symputx("X_MIN"||CATC(sequence),X_MIN);   
call symputx("X_MAX"||CATC(sequence),X_MAX);   
call symputx("Y_MIN"||CATC(sequence),CEIL(Y_MIN-1));   
call symputx("Y_MAX"||CATC(sequence),CEIL(Y_MAX+1));   
RUN;

%put &x_min1 &x_max1 &y_min1 &y_max1;   
%put &x_min2 &x_max2 &y_min2 &y_max2;   
%put &x_min3 &x_max3 &y_min3 &y_max3;   
%put &x_min4 &x_max4 &y_min4 &y_max4;

%Multi_GRAPH (SEQ1=1, SEQ2=2, SEQ3=3, SEQ4=4, LABEL1=Sequence 1, LABEL2=Sequence 2, LABEL3=Sequence 3, LABEL4=Sequence 4, N2=4, Y_SPACE1=2, Y_SPACE2=4, Y_SPACE3=4, Y_SPACE4=4)

*------------------------------------------------------------------------
About %MULTI_GRAPH macro
Purpose: To create 4 plots on one page by PROC GREPLAY
Note : This macro uses PROC GLOT features to create graphs.

Parameter Explanation:
SEQ1-SEQ4 = The sequence numbers in the dataset, each sequence number will constitute one plot. So here in this macro we need to declare 4 sequence numbers so that 4 graphs are plotted.
Labell-Labe14 = The title that each graph should display on the top.
N2 = As we need to use PROC GLOT 4 times, this key macro parameter will iterate 4 times PROC GLOT as there are four sequence. (Basically used in %DO %END statement (see the macro for further details)
YSPACE1-YSPACE4 = These key parameters is to define the gap between each point on Y-axis

*------------------------------------------------------------------------;

%MACRO Multi_GRAPH(SEQ1=,SEQ2=,SEQ3=,SEQ4=,LABEL1=,LABEL2=,LABEL3=,LABEL4=, N2=, Y_SPACE1=, Y_SPACE2=, Y_SPACE3=, Y_SPACE4=);   
symbol1 i = join c = red  l=1 w = 1.75 value=1 h=2;   
symbol2 i = join c = blue l=2 w = 1.75 value=2 h=2;   
symbol3 i = join c = green l=3 w = 1.75 value=3 h=2;   
symbol4 i = join c = black l=4 w = 1.75 value=4 h=2;
The result of the above code is as shown in Figure 3.
4. CONCLUSION

This paper has examined and explained in a simplistic step-by-step approach towards creating multiple plots on one page. The basic idea about what goes behind PROC GREPLAY has been explained in detail. This paper has also explained a macro that can be used to generate four plots for similar type of data. I hope that after going through this paper user will be more comfortable in using SAS/GRAPH when it comes to plotting multiple plots on one page.

5. REFERENCE

SAS/GRAPH Software: version 9, SAS® Institute Inc., Cary NC

SAS/Graph help online documentation: Version 9, SAS® Institute Inc., Cary NC


Acknowledgement:
I would like to thank Ms. Annpey Pong for her valuable comments.

Contact Information: Your comment and questions are valued and encouraged.
Yogesh Pande
Sr. Clinical Programmer
Schering-Plough Clinical Programming Dept.
556 Morris Ave., Summit NJ 07901
Phone: 908-473-5574
Email: yogesh.pande@spcorp.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.