

Comparison of the different methods to create multiple graphs on a page

Minming Chen, Sanofi Corporation.

ABSTRACT

In the clinical study report, we usually need to put multiple graphs on a single page. With the new SAS version 9, there are many ways for the graph combination, such as STARTPAGE=NO option in the ODS PDF statement, ODS LAYOUT, PROC GREPLAY and Graph Template Language (GTL). Each method has its own advantage and limitation. For example, STARTPAGE=NO or ODS LAYOUT statements enable you to create reports that easily mix SAS graphics and arrange them on a page for PDF destination. If we want to arrange multiple graphs on the same RTF page, GREPLAY or GTL are recommended. GTL requires more time to prepare the appropriate source data and template to create sophisticated graphics. If we only need to provide some simple layouts and single graphics can be written to the SAS/GRAPH catalog, GREPLAY is more straightforward to render the multiple graphs per page by calling SAS default template which is stored in SASHELP.TEMPLT. This paper will discuss and compare the advantages and limitations of each method to help users to choose the appropriate way for the graph combination under different circumstances.

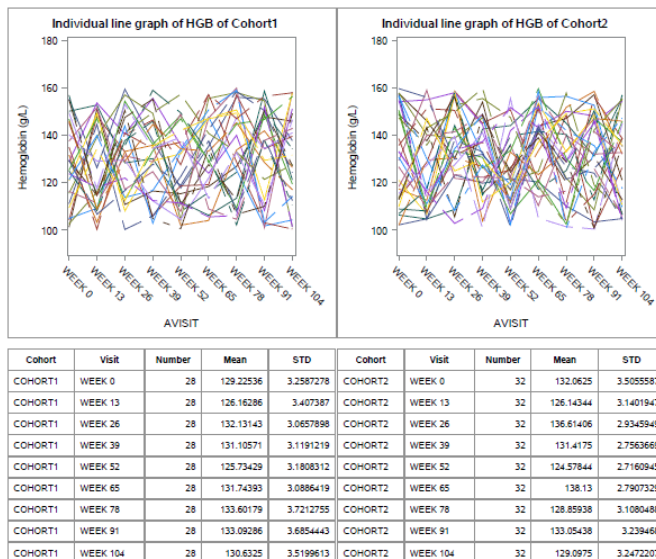
INTRODUCTION

In clinical analysis, we may receive some requests to provide multiple graphs per page. With the new SAS version 9, there are many ways to achieve this goal. In case of pressing time, it's necessary to choose one simple solution under different circumstances. To achieve this goal, we must consider two elements in priority:

- 1) The complexity of the graphic. (How many individual items need to include? Need to overlap the items?)
- 2) The destination for the graphic. (PDF or RTF?)

This paper will mainly discuss the different methods of combination without modification the original code of each item. To help user better understand the advantage and limitation of each method, one example will be created by different methods. The syntax and code example will be provided.

Figure 1



*All data in this paper has been simulated

Assume you have already created 4 items using with PROC SGPLOT and PROC REPORT (Shown in

Table 1), and you need to keep the original code and combine them on PDF format with 2 plots across and 2 plots down (Graph 1). In this situation, two methods are recommend for the PDF destination:

- STATPAGE=NO in ODS PDF statement
- ODS LAYOUT

Table 1. Four individual items which need to combine within one page

Figure 1	<pre>proc sgplot data=adlb01(where=(cohort='COHORT1')) noautolegend; series x=avisit y=aval/group=usubjid; yaxis min=90 max=180; run;</pre>
Figure 2	<pre>proc sgplot data=adlb01(where=(cohort='COHORT2')) noautolegend; series x=avisit y=aval/group=usubjid; yaxis min=90 max=180; run;</pre>
Table 1	<pre>proc report data=ads01(where=(COHORT='COHORT1')) style(report)={borderwidth=1 cellwidth=3in fontsize=4pt} style(column)={borderwidth=0} nocenter nowindows; column cohort avisit n1 mean1 stderr1; define cohort /'Cohort'; define avisit /'Visit'; define n1 /'Number'; define mean1 /'Mean'; define stderr1 /'STD'; run;</pre>
Table 2	<pre>proc report data=ads01(where=(COHORT='COHORT2')) style(report)={borderwidth=1 cellwidth=3in fontsize=4pt} style(column)={borderwidth=0} nocenter nowindows; column cohort avisit n1 mean1 stderr1; define cohort /'Cohort'; define avisit /'Visit'; define n1 /'Number'; define mean1 /'Mean'; define stderr1 /'STD'; run;</pre>

METHOD 1: STARTPAGE=NO OR NEVER

The default behavior in ODS PDF that forces a new page after a graphics procedure. To control page breaks, we can use STARTPAGE=NEVER or NO to turn off that behavior and put the multiple outputs on the same page. See detail options of STARTPAGE below:

Table 2. OPTIONS OF STARTPAGE

STARTPAGE=NEVER NO NOW YES	
NO OR OFF	No new pages be inserted at the beginning of each procedure New pages will be created only when necessary to avoid overflowing pages
NEVER	No new pages be inserted at the beginning of each procedure. New pages never to be created so this might cause graphics to overprint
YES ON	Insert a new page at the start of each procedure
NOW	Force a new page right now

EXAMPLE 1: STARTPAGE=NO

The following codes demonstrate how to combine the multiple items by using STATPAGE.

- STATPAGE=NO tells SAS not to go to a new page at the beginning of each procedure. (New pages will be created only when necessary to avoid overflowing pages).
- COLUMNS=2 to specify the number of columns to create on each page of output. Each individual Graph (PROC SGPLOT) and table (PROC REPORT) can be inserted between ODS PDF FILE and ODS PDF CLOSE.

```
ods pdf file="xxxx.xxxx.pdf" startpage=no columns=2 ;
ods graphics on /reset=all width=5in height=5in ;
    [Figure1 Cohort1 : proc sgplot data=...]
ods graphics off;

    [Table1 Cohort1 : proc report data=.. run;]

ods graphics on /reset=all width=5in height=5in ;
    [Figure2 Cohort2: proc sgplot data=...]
ods graphics off;

    [Table2 Cohort2: proc report data=.. run;]
ods pdf close;
```

STATPAGE=NO is the simplest way to combine these 4 individual items. While if you need to create a PDF page with three plots across and three plots down or even more, STATPAGE will have limit options to control the position of each item and it's risky to have an unintended overlap. In such scenario, ODS LAYOUT will be more appropriate.

METHOD 2: ODS LAYOUT GRIDDED

The gridded layout can arrange output in a two-dimensional gridded structure. It supports for HTML and printer destinations (PDF, PS and PCL) as well as the PowerPoint destination. ODS REGION can control the position of each item. And ODS REGION must be used with ODS LAYOUT. A simple syntax is summarized as below:

SYNTAX OF ODS LAYOUT GRIDDED

```
ODS LAYOUT GRIDDED <options>;
    ODS REGION <options>;
        [Output generating procedure code]
    ODS REGION <options>;
        [Output generating procedure code]
    ODS REGION <options>;
```

[Output generating procedure code]

ODS LAYOUT END;

EXAMPLE 2: ODS LAYOUT GRIDDED

```
ods layout gridded height=13cm width=21cm rows=2 columns=2
      row_heights=(6cm 6cm) column_widths=(10cm 10cm)
      order_type=row_major
column_gutter=1cm row_gutter=1cm;
ods region row=1 column=1 ;
  [Figure1 Cohort1: proc sgplot code.. ]
ods region row=1 column=2 ;
  [Figure2 Cohort2: proc sgplot code..]
ods region row=2 column=1 ;
  [Table1 Cohort1: proc report code...]
ods region row=2 column=2;
  [Table2 Cohort2: proc report code...]
ods layout end;
```

In the ODS LAYOUT GREIDDED options,

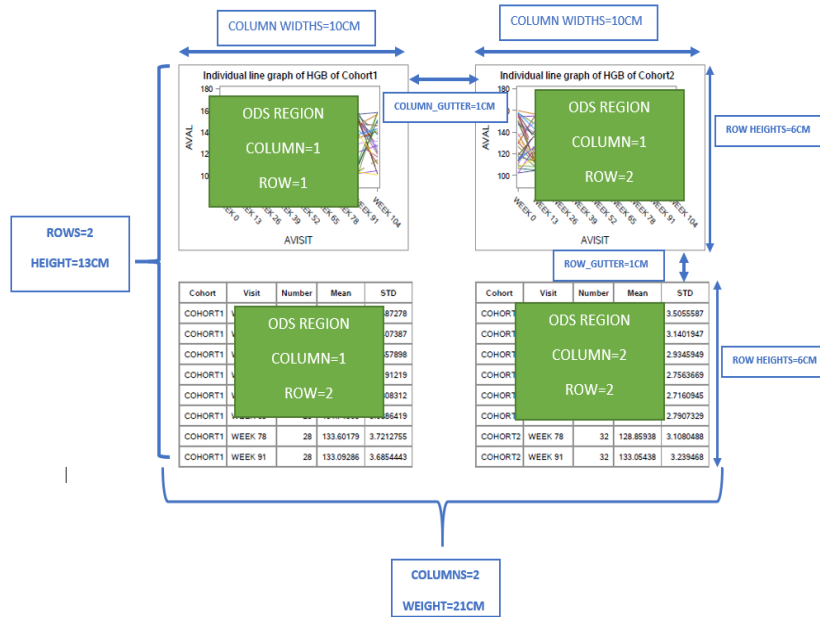
- HEIGHT=13 cm and WIDTH=21 cm to tell SAS the overall dimensions of the layout.
- COLUMNS=2 and ROWS=2 to control the number of columns and rows in the grid.
- COLUMN_WIDTHS=(6cm 6cm) and ROW_HEIGHTS=(6cm 6cm) to control the dimensions of the cells within the grid;
- COLUMN_GUTTER=1cm and ROW_GUTTER=1cm to control the amount of empty space between grid cells.
- The ORDER_TYPE option controls the order in which the layout is filled by default.

In the ODS REGION option,

- COLUMN and ROW specify the exact cell region.

Please see the following diagram which illustrates the main options in ODS LAYOUT GREIDDED (marked in blue) and ODS REGION (marked in green)

Figure 2. OPTIONS OF ODS LAYOUT GREIDDED



With the Gridded layout, you can simply tell ODS how many columns and rows that you need, and it will automatically compute the position of each cell to accommodate the output, which is the greatest advantage of ODS Gridded LAYOUT. It has the following capabilities:

- provide automatic alignment of respective grid cells
- continue the layout onto the next page when necessary
- dynamically compute the size of a grid cell
- makes it easier to maintain the integrity of the report.

However, if you need to create a composite output and overlap some of individual items (**See Graph 3**), ODS Gridded will encounter the limitation. Because the gridded layout may not overlap each other. It will automatically continue the layout onto the next page when necessary. In such case, ODS LAYOUT ABSOLUTE is more appropriate.

METHOD 3: ODS LAYOUT ABSOLUTE

LAYOUT ABSOLUTE enables you to specify an exact page location using x and y coordinates. Same with ODS LAYOUT GRIDDED, it's limited to one page and is supported only by the PRINTER destinations (PDF, PS, and PCL). A simple syntax is summarized as below:

SYNTAX OF ODS LAYOUT ABSOLUTE

```
ODS LAYOUT ABSOLUTE <options>;
    ODS REGION <options>;
    [Output generating procedure code]
ODS LAYOUT END;
```

For example, If you want to overlap the demographic table with the line graph (Graph 3) , the position of each individual item by (X,Y) need to be specified.

EXANPLE 3: ODS LAYOUT ABSOLUTE

```
ods layout absolute;
ods region height=20cm width=20cm x=0% y=0% ;
[Figure1 Cohort1: proc sgplot code.. ]

ods region height=10cm width=10cm x=10% y=4%;
[Demographic Table: proc report data= ..run;]

ods region height=10cm width=10cm x=0% y=50%;
[Table1 Cohort1: proc report code....]
ods layout end;
```

In ODS REGIONS options:

- HEIGHT and WIDTH options in ODS REGIONS determine the dimensions of each cell.
- X and Y options specify the horizontal and vertical starting position of the region. Regions begin from their top left corner. The units of (X,Y) can be 'cm', 'in', 'mm', 'pct'..etc. While I highly recommend using 'pct' to define the position of each cell. This is more practical to adjust the overall output.

Please see the below diagram that illustrates the ODS REGIONS options:

Figure 3. OVERLAP DEMOGRAPHIC TABLE WITH LINE PLOT



Figure1 Cohort1 (0%, 0%) -> Region begin from their top left corner

Demographic Table (10%, 4%) -> Region begin from 10% of the left corner and 4% of the top corner

Table1 Cohort1 (0%, 50%) -> Region begin from the left corner and 50% of the top corner

Absolute layout works well to create the following types of output:

- cover page
- static data
- preprinted form
- single page output

With ABSOLUTE LAYOUT, each location needs to be explicitly placed and it would be challenging to maintain program code when one container's position is altered, you might need to manually alter all the other containers to maintain your report's integrity.

Both STATRPAGE =NO and ODS LAYOUT can combine the multiple graph. You can just sandwich your original code with ODS start and end block. While STARTPAGE and ODS LAYOUT are difficult to support the RTF destination. If you need to combine the multiple graph on RFT format, you can consider these two ways:

- PROC GREPLAY.
- GTL

METHOD 4: PROC GREPLAY

PROC GREPLAY can lay out multiple graphs on one page by calling the graphics stored in SAS catalogs. While if you use the PROC SGPLOT/SGPANEL/SGSCATTER/SGRENDER/GTILE/GKPI, PROC GREPLAY cannot combine the multiple graphs directly, because these procedures are not able to write the graphic into a SAS/GRAPH Catalog. In such case, PROC GREPLAY should be used with PROC GSLIDE and IBACK

I'll use the same example (Graph1) to demonstrate how it works. Please see the following as below.

EXAMPLE 4: PROC GREPLAY WITH PROC GSLIDE AND IBACK

Step 1: Use PROC SGPLOT/PROC REPORT to produce PNG files and store them by ODS.

For example, I've created four individual items (Figure1, Figure2, Table1, Table2) and stored them under the path (&mypath) on PNG format.

```
/*Create an libname to store your file*/
libname MISC "&mypath";
options device='PNG';

/*Delete all the old item in the catalog*/
proc greplay igout=misc.sgplot nofs;
  delete _all_;
run;

/*Create the 4 individual items and store them into &mypath*/
ods listing gpath="&mypath" image_dpi=300;
ods graphics on /reset=all width=4in height=4in imagename="Figure1 "
  imagefmt=png noborder;
  [Figure1 Cohort1: proc sgplot code.. ]
ods graphics off;

ods graphics on /reset=all width=4in height=4in imagename="Figure2"
  imagefmt=png noborder;
  [Figure2 Cohort2: proc sgplot code.. ]
ods graphics off;

ods printer file="&mypath/Table.png" ;
  [Table1 Cohort1: proc report code....]
ods printer close;

ods printer file="&mypath/Table.png" ;
```

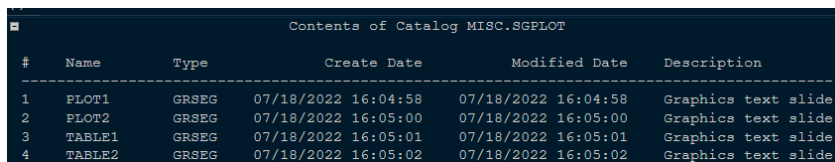
```
[Table2 Cohort2: proc report code...]  
ods printer close;
```

Step 2: Use PROC GSLIDE with IBACK to read the PNG files which created in Step1 and bring them into SAS catalog

```
/*Read the PNG file from &mypath and put them into catalog*/  
goptions iback="&mypath/Figure1.png" imagestyle=fit;  
proc gslide GOUT=misc.sgplot name='Plot1';  
run;
```

Repeat the above code to read other PNG files. Use proc catalog to verify if the individual items which you need to display are successfully stored under your catalog

```
/*Print all the items in the sascatalog*/  
proc catalog catalog=misc.sgplot;  
contents;  
run;
```



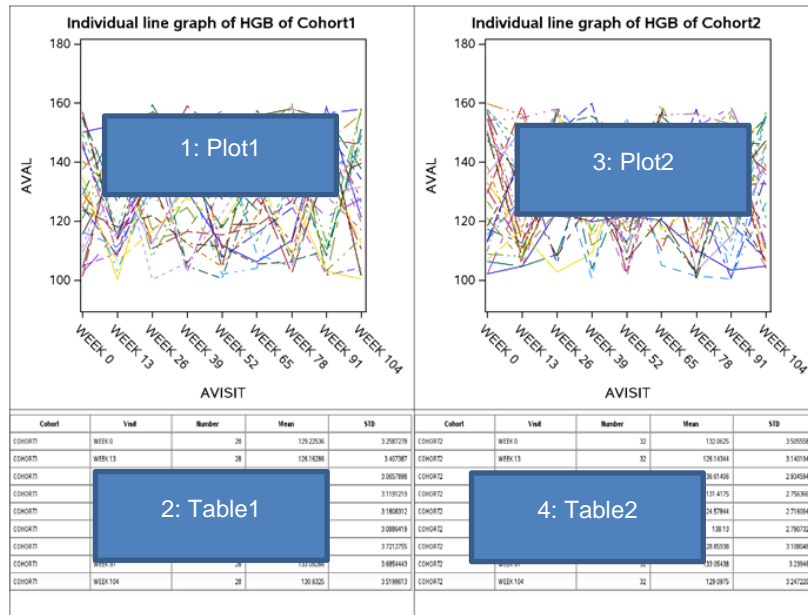
#	Name	Type	Create Date	Modified Date	Description
1	PLOT1	GRSEG	07/18/2022 16:04:58	07/18/2022 16:04:58	Graphics text slide
2	PLOT2	GRSEG	07/18/2022 16:05:00	07/18/2022 16:05:00	Graphics text slide
3	TABLE1	GRSEG	07/18/2022 16:05:01	07/18/2022 16:05:01	Graphics text slide
4	TABLE2	GRSEG	07/18/2022 16:05:02	07/18/2022 16:05:02	Graphics text slide

Step 3: Use PROC GREPLAY to replay graphs from Step 2

```
/**Display the 4 items by using SAS template L2R2*/  
proc greplay igout = misc.sgplot gout=misc.sgplot tc=sashelp.templt  
template=L2R2 nofs;  
treplay 1:plot1 2:table1 3:plot2 4:table2;  
run;
```

- The IGOUT option specifies the input catalogue is misc.sgplot, which contains the 4 items that you want to present.
- The GOUT option specifies the output catalog is misc.sgplot.
- The TC= store templates created with the GREPLAY procedure. This example use the SAS templates which stored in sashelp.templt, named "L2R2"

Figure 4 . PROC GREPLAY WITH TEMPLATE=L2R2



You can also customize the template by using TDEF statement in PROC REPLAY

```
proc greplay tc=work.tempcat igout = misc.sgplot gout=misc.sgplot nofs;
  /**Define the new template*/
  tdef newtemp des='Four panel template'
    1/ulx=5 uly=95 urx=48 ury=95 llx=5 lly=52 lrx=48 lry=52 color=white
    2/ulx=52 uly=95 urx=95 ury=95 llx=52 lly=52 lrx=95 lry=52 color=white
    3/ulx=10 uly=48 urx=50 ury=48 llx=10 lly=5 lrx=50 lry=5 color=white
    4/ulx=60 uly=48 urx=100 ury=48 llx=60 lly=5 lrx=100 lry=5 color=white
;
  /**Use the new template*/
  template=newtemp;
  treplay 1:plot1 2:plot2 3:table1 4:table2;
run;
```

Statement TDEF defines the template (newtemp): The co-ordinate values are all percentages of the total area.

- co-ordinates (llx, lly) define the lower left-hand corner
- co-ordinates (ulx, uly) define the upper left corner
- co-ordinates (lrx, lry) define the lower right corner
- co-ordinates (urx, ury) define the upper right corner

Figure 5 . PROC GREPLAY WITH CUSTOMIZED TEMPLATE



PROC GREPLAY is a powerful method to lay out multiple graphs on one page. Table 4 lists all the SAS template that we can directly use by PROC GREPLAY. If we only need to create the simple layouts, it's more efficient to choose one of them to render the multiple graphs per page.

Table 3. Sample template provided by SAS

TEMPLATE NAME	DESCRIPTION	TEMPLATE NAME	DESCRIPTION
H2	1 BOX LEFT, 1 BOX RIGHT	L2R2S	2 BOXES LEFT, 2 BOXES RIGHT (WITH SPACE)
H2S	1 BOX LEFT, 1 BOX RIGHT (WITH SPACE)	U1D2	1 BOX UP, 2 BOXES DOWN
H3	3 BOXES ACROSS (HORIZONTALLY)	U1D2S	1 BOX UP, 2 BOXES DOWN (WITH SPACE)
H3S	3 BOXES ACROSS (WITH SPACE)	U2D1	2 BOXES UP, 1 BOX DOWN
H4	4 BOXES ACROSS (HORIZONTALLY)	U2D1S	2 BOXES UP, 1 BOX DOWN (WITH SPACE)
H4S	4 BOXES ACROSS (WITH SPACE)	V2	1 BOX UP, 1 BOX DOWN
L1R2	1 BOX LEFT, 2 BOXES RIGHT	V2S	1 BOX UP, 1 BOX DOWN (WITH SPACE)
L1R2S	1 BOX LEFT, 2 BOXES RIGHT (WITH SPACE)	V3	3 BOXES STACKED VERTICALLY
L2R1	2 BOXES LEFT, 1 BOX RIGHT	V3S	3 BOXES STACKED VERTICALLY (WITH SPACE)
L2R1S	2 BOXES LEFT, 1 BOX RIGHT (WITH SPACE)	WHOLE	ENTIRE SCREEN TEMPLATE
L2R2	2 BOXES LEFT, 2 BOXES RIGHT		

METHOD 5: GTL

GTL can create multiple cells graph. To combine the graphs and tables within one page, we can use layout lattice and %SGANNO to achieve this goal.

EXAMPLE 5: GTL WITH %SGANNO

Step 1: Define the graph template

- The template definition starts with the proc template statement

- The define statement specifies the type as 'begingraph' and provides the name of template
- The template definition is provided inside the begingraph/endgraph block. All the GTL statements must be inside of this block. The detail code and options are marked as below:

```

proc template;
  define statgraph seriesplot;

  /**Define graph**/
  begingraph /designwidth=1300 designheight=100 border=false;
    layout lattice /columnweights=(.50 .50) rowweights=(.50 .50)
      columns=2 rows=2 order=rowmajor columngutter=1 rowgutter=1;

    /**Define header**/
    column2headers;
    entry textattrs=GraphData1(weight=bold size=9pt Family="Albany
      AMT") "Cohort1";
    entry textattrs=GraphData2(weight=bold size=9pt Family="Albany
      AMT") "Cohort2";
    endcolumn2headers;

  /**Define plot1**/
  layout overlay/yaxisopts=(display=(ticks tickvalues line label)
    label=("Hemoglobin (g/L)")
    Offsetmin=0.02 Offsetmax=0.01
    linearopts=(viewmin=&ymin. viewmax=&ymax.
    tickvaluesequence=(start=&ymin. end=&ymax.
    increment=1))
    griddisplay=auto_on)
    xaxisopts=(label=("Study Week"));
    seriesplot x=avisit y=aval1/group=usubjid
    INCLUDEMISSINGGROUP=FALSE;

    /*Add annotation*/
    annotate/id='anno1';
    annotate/id='anno2';
  endlayout;
  /**Define plot2**/
  layout overlay/
    yaxisopts=(display=(ticks tickvalues line label)
    label=("Hemoglobin (g/L)")
    Offsetmin=0.02 Offsetmax=0.01
    linearopts=(viewmin=&ymin. viewmax=&ymax.
    tickvaluesequence=(start=&ymin. end=&ymax.
    increment=1))
    griddisplay=auto_on)
    xaxisopts=(label=("Study Week"));
    seriesplot x=avisit y=aval2/group=usubjid
    INCLUDEMISSINGGROUP=FALSE;
    endlayout;
  endlayout;
  endgraph;
end;
run;

```

Step2: Create the annotation dataset

HIGHLOWPLOT (Forest plot) or BLOCKPLOT can define the simple table which could be inserted into

begingraph/endgraph block. While it requires more time to prepare the appropriate source data and template to produce sophisticated graphics. SGANNO macros provided by SAS is a very useful tool to help us to add custom text or shapes in the graph. When using SGANNO,

- %sganno : We need to run %sganno firstly, which compiles the other SGANNO macros and makes them available.
- %sganno_help: This macro helps to understand the purpose of these macros, macro variables and the possible values of macro-variables. For example, %sganno_help(sgimage)

Annotation data shown as below is created by %sgimage. With this data, proc template can add table1.png and table2.png into the graph by annotate statement

Annotation Data

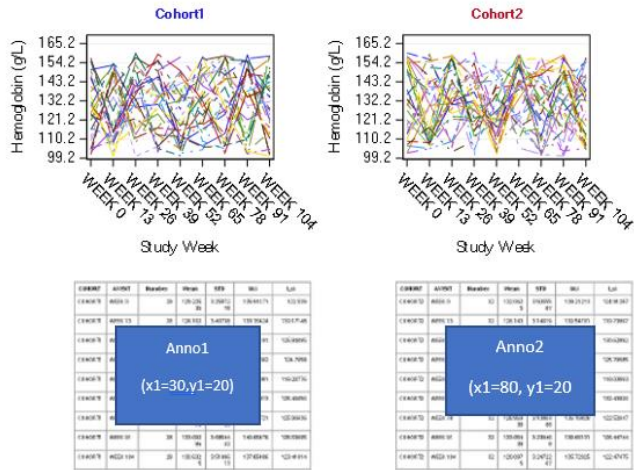
DRAWSPACE	FUNCTION	ID	IMAGE	LAYER	TRANSPARENCY	X1	Y1
datapercnt	IMAGE	anno1	&mypath/Table1.png	front	0	30	20
datapercnt	IMAGE	anno2	&mypath/Table2.png	front	0	80	20

- Drawspace=Graphpercent : specifies the entire region of the graph will be used for the annotation
- Function=IMAGE : specifies the image will be used for the annotation and it's identified by ID (ID=anno1, anno2, which is stored under &mypath)
- X and Y coordinates for an annotation, the coordinate values refer to a drawing space. The coordinate origin (0, 0) for each of these regions is the lower left corner of the corresponding rectangular region, with positive x values going to the right and positive y values going up. Specifying X=50 and Y=50, refers to a point in the center of the graph area.

```
data anno;
  %sgimage (image=&mypath/Table1.png", drawspace='graphpercent',
            x1=30, y1=20, id='anno1', layer='front', transparency=0);
  %sgimage (image=&mypath/Table2.png", drawspace='graphpercent',
            x1=80, y1=20, id='anno1', layer='front', transparency=0);
run;
```

Step2: Create the final output by PROC SGRENDER

```
proc sgrender data=ads02 template=seriesplot sganno=anno;
run;
```



CONCLUSION

This paper provides 5 different methods to combine multiple graph or table per page. For PDF destination, STARTPAGE=NO or ODS LAYOUT are recommended to combine the multiple graphs per page. For PDF destination, the PROC GREPLAY or GTL can be used. Each method has its own advantage and limitation. Below table summarizes the advantages and limitations of each method. I hope this can be a guidance to help user to choose the appropriate way for the graph combination under different circumstances

Methods	ADVANTAGES	LIMITATIONS
STARTPAGE=NO in ODS PDF	The STARTPAGE=NO option can easily mix SAS graphics, images, text, tables, and arrange them on a page.	If graphic is not generated by a traditional SAS/GRAPH procedure (such as GPLOT), it's hard to control the position when using HORIGIN and VORIGIN
ODS Gridded LAYOUT	<p>Create custom reports that easily mix SAS graphics, images, text, tables, and arrange them on a page.</p> <p>Dynamically compute the size of a grid cell and continue the layout onto the next page when necessary.</p> <p>Easier to maintain the integrity of the report.</p>	<p>Gridded layout is only supported for HTML and PRINTER destinations (PDF, PS, PCL)</p> <p>Each individual graph cannot overlap with each other.</p>

ODS LAYOUT ABSOLUTE	Create custom reports that easily mix SAS graphics, images, text, tables, and arrange them on a page. Enables you to specify an exact page location using x and y coordinates	Only ODS PRINTER (PDF, PS, PCL) destinations support absolute layout. Each location (X,Y) needs to be explicitly placed and it would be challenging to maintain program code
PROC GREPLAY	GREPLAY is more straightforward to render the multiple graphs per page by calling sas default template which is stored in SASHELP.TEMPLT	PROC GREPLAY cannot combine directly the graph produced by PROC SGPLOT/SGPANEL/SGSCATTER/SGRENDER/GTITLE/GKPI
GTL	Create sophisticated analytical graphics that are not available from traditional SAS/GRAPH procedure statements	Cannot to keep original code of each item to combine. This requires more time to prepare the appropriate source data

REFERENCES

Michael Allie, Oncology Biometrics, Early Oncology Programming, R&D, AstraZeneca, Cambridge, United Kingdom “*Using ODS LAYOUT to Create ABSOLUTE-ly Stunning Patient Profiles*” Paper DV03.

Brian T. Schellenberger, SAS Institute Inc., Cary, NC, “ODS LAYOUT: Arranging ODS Output as You See Fit” (3rd revision)

Kevin P. Delaney MPH, Northrop-Grumman Mission Systems, Atlanta, GA, “*Multiple Graphs on One page, the easy way (PDF) and the hard way (RTF)*”, Paper 94-28

Yogesh Pande, Schering-Plough Corporation, Summit NJ “*Multiple Graphs on One Page (Step-by-step approach)*” Paper CC01

KarolAnne Fitzpatrick, ICON Clinical Research, Dublin, Irela “Automating GREPLAY to display multiple Graphs per page” Paper IS03

SAS 9.4 *Output Delivery System: User’s Guide*, Fifth Edition

SAS 9.4 *System Options: Reference*, Fifth Edition

SAS 9.4 *Graph Template Language: User’s Guide*, Fifth Edition

CONTACT INFORMATION

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

Minming CHEN

E-mail: alex199196@hotmail.com