PharmaSUG China 2018 PO-13

How to Create Graphic Images in Assembly Line Way Using PROC TEMPLATE in SAS Enterprise Guide Part I

Kaiqing Fan, Mastech Digital Inc.

Jing Wang, China Minsheng Bank

ABSTRACT

In banking industry, all the variables, their values and requirements are keeping changing. As a SAS developer, we may be asked to create thousands or hundreds composite or single graphic images such as scatterplot, seriesplot, stepplot, vectorplot, barchart, linechart, piechart, waterfallchart, boxplot, densityplot, histogram, loessplot, Pbsplineplot, regressionplot using pipeline operation method through SAS graphic engines. To reach this purpose, it is impossible to manually modify many parameters or engines codes for each graphic image. Any manual interventions may cause horrible disasters or mass with the requirement of creating hundreds or thousands graphic images. Then question is coming: how to assembly line create composite or single graphic images using **PROC TEMPLATE** is the topic now. To reach this target, we need to automatically generate all or most parameters and cover expected changes if possible.

COMMENTS: I finished three SAS Papers about data visualizations. This is the Part I.

The Auto-Calculator of the Graphic y-axis Tickvaluelist, Tickvalueformat with Expected Values for SAS Data Visualizations Part II was published by Ohio SAS Users Conference 2018.
Some Tricks and Explanations When Plotting Graphic Images Using PROC TEMPLATE SAS® Enterprise Guide Part III was published by SAS Global Forum 2018 as e-poster. Its paper number is 2545-2018

INTRODUCTION

In banking industry, we may be asked to create thousands of hundreds graphic images including scatterplot, seriesplot, stepplot, vectorplot, barchart, linechart, piechart, waterfallchart, boxplot, densityplot, histogram, loessplot, Pbsplineplot, regressionplot by Assembly Line way through SAS graphic engines.

How to automatically create graphic images using **PROC TEMPLATE** is our target. To reach this target, we need to conquer couple challenges.

1) How to define those hundreds of parameters into the SAS engine?
2) How to develop automatic SAS engine to avoid manual intervention or hard coding when we execute the engine to create these thousands of hundreds graphic images?

One Example: Hard Coding Linechart **PROC TEMPLATE** Engine

Many developers or project managers may not believe it, they may question: "do we really need to define hundreds of parameters for the graphics?" To answer it, let us see one example below. Suppose we are asked to create 2x2 Line graphic images using **PROC TEMPLATE**. Here is the Linechart **PROC TEMPLATE** engine with hard coding of all parameters. I highlighted them with yellow colors which there are around 400 places which we may need to modify.

```
ods _all_ close;    options number nodate;
ods graphics on/reset imagename="Slide5" imagefmt=png width=30in height=22in
    scale=off border=off;
ods listing gpath="/sas/development/reports/run2/grouped_images";
ods path reset;
ODS path (PREPEND) a.TEMPLAT(update);
proc template;
define statgraph a.class;
begingraph;
discreteattrvar attrvar=classfill var=fore_sce attrmap='colors';
[1]
legendItem type=Line name="AAA_REPORTB2017"     / lineattrs=(color=CX0069AA
   pattern=solid  THICKNESS=8)  labelattrs=(size=22pt) label="REPORTB 2017 A";
legendItem type=Line name="AAA_REPORTA2017"     / lineattrs=(color=CX0069AA
   pattern=MediumDash  THICKNESS=8)  labelattrs=(size=22pt) label="REPORTA 2017 A";
legendItem type=Line name="BBB_REPORTB2017"     /  lineattrs=(color=CX006E51
   pattern=solid  THICKNESS=8)  labelattrs=(size=22pt) label="REPORTB 2017 B";
legendItem type=Line name="BBB_REPORTA2017"     /  lineattrs=(color=CX006E51
   pattern=MediumDash  THICKNESS=8)  labelattrs=(size=22pt) label="REPORTA 2017 B";
legendItem type=Line name="SSS_AAA_REPORTB2017" / lineattrs=(color=CXF58025
   pattern=solid THICKNESS=8)  labelattrs=(size=22pt) label="REPORTB 2017 SA";
legendItem type=Line name="SSS_AAA_REPORTA2017" / lineattrs=(color=CXF58025
   pattern=MediumDash THICKNESS=8)  labelattrs=(size=22pt) label="REPORTA 2017 SA";
legendItem type=Line name="UU_15_REPORTB2017"   / lineattrs=(color=CXDA81F5
   pattern=solid  THICKNESS=8) labelattrs=(size=22pt) label="REPORTB17 HA15";

layout lattice/border=FALSE BORDERATTRS=(color=white) columns=2 rows=2
              COLUMNGUTTER=2cm ROWGUTTER=2cm;
cell;    /*Up Left cell, simply as UL*/
cellheader;
entry "UL Graph"/border=FALSE BORDERATTRS=(color=white) textattrs=(size=30pt weight=bold);
endcellheader;
[2]
layout overlay /border=false WALLCOLOR=white WALLDISPLAY=(FILL)
xaxisopts=(display=all linearopts=(origin=0 THRESHOLDMIN=1 THRESHOLDMAX=1)
   label=' ' griddisplay=off labelattrs=(weight=bold  size=22pt)
   tickvalueattrs=(weight=bold size=22pt) discreteopts=(tickdisplaylist=("PQ1" "" "" "PQ4"
   "" "" "PQ7" "" "" "PQ10" "" "" "PQ13" "" "" "PQ16" "" "" "PQ19" "" "" "PQ22")
   tickvaluelist=("PQ1" "PQ2" "PQ3" "PQ4" "PQ5" "PQ6" "PQ7" "PQ8" "PQ9" "PQ10" "PQ11"
   "PQ12" "PQ13" "PQ14" "PQ15" "PQ16" "PQ17" "PQ18" "PQ19" "PQ20" "PQ21" "PQ22")
   TICKVALUEFITPOLICY=ROTATEALWAYS   TICKVALUEROTATION=VERTICAL))
yaxisopts=(display=(label ticks tickvalues) label=' ' labelattrs=(weight=bold size=22pt)
   griddisplay=on  tickvalueattrs=(weight=bold  size=22pt)
   linearopts=(integer=FALSE  origin=0  viewmin=0  tickvalueformat=9.1
   tickvaluelist=(0 12.5 25 37.5 50 62.5 75 87.5 100 112.5 125 137.5 150)
   TICKVALUEPRIORITY=TRUE) gridattrs=(pattern=dash THICKNESS=2 color=darkgrey));
 [3]
seriesplot x=xaxis  y=REPORTB2017_AAA_UL / LEGENDLABEL="REPORTB 2017 A"
   PRIMARY=TRUE  lineattrs=(color=CX0069AA  pattern=solid  THICKNESS=8);
seriesplot x=xaxis  y=REPORTA2017_AAA_UL / LEGENDLABEL="REPORTA 2017 A"
   PRIMARY=TRUE  lineattrs=(color=CX0069AA  pattern=MediumDash THICKNESS=8);
seriesplot x=xaxis  y=REPORTB2017_BBB_UL / LEGENDLABEL="REPORTB 2017 B"
   PRIMARY=TRUE  lineattrs=(color=CX006E51  pattern=solid THICKNESS=8);
seriesplot x=xaxis  y=REPORTA2017_BBB_UL / LEGENDLABEL="REPORTA 2017 B"
   PRIMARY=TRUE  lineattrs=(color=CX006E51  pattern=MediumDash THICKNESS=8);
seriesplot x=xaxis  y=REPORTB2017_SSS_AAA_UL / LEGENDLABEL="REPORTB 2017 SA"
   PRIMARY=TRUE lineattrs=(color=CXF58025 pattern=solid THICKNESS=8);
seriesplot x=xaxis  y=REPORTA2017_SSS_AAA_UL / LEGENDLABEL="REPORTA 2017 SA"
   PRIMARY=TRUE  lineattrs=(color=CXF58025  pattern=MediumDash  THICKNESS=8);
seriesplot  x=xaxis  y=REPORTB2017_UU_15_UL / LEGENDLABEL="REPORTB17 HA15"
   PRIMARY=TRUE  lineattrs=(color=CXDA81F5  pattern=solid  THICKNESS=8);
referenceline y=0 / curvelabel="0.0" curvelabelattrs=(size=22pt  weight=bold)
   clip=true curvelabelposition=min lineattrs=(pattern=dash THICKNESS=2 color=darkgrey);
endlayout;
endcell;

cell;  …   endcell;    /*Up Right cell, UR*/
```

```
cell;  …   endcell;    /*Low Left cell, LL*/
cell;  …   endcell;    /*Low Right cell,LR*/
endlayout;

entryfootnote halign=left  TEXTATTRS=(size=22pt weight=bold) "Slide5=XY "
              halign=right TEXTATTRS=(size=22pt weight=bold) "Release 1 ";

layout  globallegend/border=false type=COLUMN weights=uniform;
[4]
discretelegend "AAA_REPORTB2017" "BBB_REPORTB2017"
               "SSS_AAA_REPORTB2017"  "UU_15_REPORTB2017"
               /border=false across=4 down=1 autoalign=(CENTER) ORDER=COLUMNMAJOR;
discretelegend "AAA_REPORTA2017"  "BBB_REPORTA2017"   "SSS_AAA_REPORTA2017"
               /border=false across=4 down=1 autoalign=(CENTER) ORDER=COLUMNMAJOR;
endlayout;
endgraph;
end;
run;
proc sgrender data=grp_comp.Slide_5_Line template=a.class;    Run;
ods graphics off;
ods listing;
```

THE CHALLENGES SHOWED IN THE ABOVE EXAMPLE

We want to generalize the above two hard coding engine, and achieve the purpose of assembly line creating graphic images. To fulfil this target, there are some challenges in the above bar and line charts engines we must carefully face to:

1) How can we define these 220 or 400 parameters?
2) How to read these parameters into the engine correctly? If we define all or some of these 400 parameters using CALL SYMPUT function or **PROC SQL** SELECT INTO, it would be very complex and cause tons of headaches as my experience. It is feasible but not strongly recommended.
3) The worse situation is that currently we have 8 situations like for REPORTA: BBB, AAA, SSS; For REPORTB: AAA, BBB, SSS, HA15. We may have more situations like TU15, TU30, TU45 … this number 8 is random changing, can be 6, 20 and any natural numbers.
4) In our examples, we have 4 cell images on the same page, but we may want to have 2 cell images, 6 cell images, 9 cell images, and more cell images on the same page. Some of the cell images may be blanks.
5) Because the graphic images can contain any numbers and they are changing case by case, how to automatically calculate the yaxisopts (origin, viewmin, tickvaluelist, tickvalueformat), barchart statement barlabelformat and referenceline curvelabel would be big both mathematical and coding challenges.

The referenceline curvelabel in the following two images are from bar chart.
The first one has curvelabel="0.00%" because the 0 is the maximum or minimum of the values of tickvaluelist of yaxisopts.
The second one has curvelabel=" " because the 0 is not the maximum or minimum of the values of tickvaluelist of yaxisopts.

```
        linearopts=(integer=FALSE origin=0 viewmin=0
        tickvaluelist=(0 0.0025 0.005 0.0075 0.01 0.0125 0.015 0.0175)
      TICKVALUEPRIORITY=TRUE tickvalueformat=percentn9.2));
barchart category=legend_value y=PQ1_UL/ orient=vertical YAXIS=Y barlabel=true dataskin=none barwidth=.66
        BARLABELFITPOLICY=auto group=classfill BASELINEINTERCEPT=0 barlabelformat=percentn9.2
        baselineattrs=(thickness=1) name="OGROUP" display=(FILL) BARLABELATTRS=(SIZE=22pt weight=bold);
referenceline y=0/ curvelabel=" 0.00% " curvelabelattrs=(size=22pt weight=bold) clip=true curvelabelposition=min ;

        linearopts=(integer=FALSE origin=-0.015 viewmin=-0.015
        tickvaluelist=(-0.015 -0.01 -0.005 0 0.005 0.01 0.015 0.02 0.025)
        TICKVALUEPRIORITY=TRUE tickvalueformat=percentn9.1));
barchart category=legend_value y=PQ1_UR/orient=vertical YAXIS=Y barlabel=true dataskin=none barwidth=.66
        BARLABELFITPOLICY=auto group=classfill BASELINEINTERCEPT=0 barlabelformat=percentn9.1
        baselineattrs=(thickness=1) name="OGROUP" display=(FILL) BARLABELATTRS=(SIZE=22pt weight=bold);
referenceline y=0/ curvelabel=" " curvelabelattrs=(size=22pt weight=bold) clip=true curvelabelposition=min;
```

6) The following last two are from line chart engine, we need to define the parameters of xaxixopts (tickdisplaylist and tickvaluelist), our engine needs to automatically pick one xaxis tickvalue from every 3 or 4 or 5 of them for xaxisopts tickdisplaylist.

```
layout overlay /border=false WALLCOLOR=white WALLDISPLAY=(FILL)
    xaxisopts=(display=all linearopts=(origin=0 THRESHOLDMIN=1 THRESHOLDMAX=1)
        label=' ' griddisplay=off labelattrs=(weight=bold size=22pt) tickvalueattrs=(weight=bold size=22pt)
        discreteopts=(
        tickdisplaylist=("PQ1" "" "" "PQ4" "" "" "PQ7" "" "" "PQ10" "" "" "PQ13" "" "" "PQ16" "" "" "PQ19" "" "" "PQ22")
        tickvaluelist =("PQ1" "PQ2" "PQ3" "PQ4" "PQ5" "PQ6" "PQ7" "PQ8" "PQ9" "PQ10" "PQ11" "PQ12" "PQ13"
                        "PQ14" "PQ15" "PQ16" "PQ17" "PQ18" "PQ19" "PQ20" "PQ21" "PQ22")
```

THE SOLUTIONS OF THE ABOVE 6 CHALLENGES

Step 1: **Solution of Challenge 1** --- create parameter tables. We can easily create tables in which contains required parameters. The tables can be used in the plot engine. Here are three example tables.

Table1 covers any situatKeys; if any new situatKey added, add its values below.

SituatKeys	legendtype	linecolor	pattern	thick	labelsize	txtsize	weight	comparison	Year
AAA	Line	CX0069AA	solid	8	22pt	30pt	bold	REPORTB	2017 A
BBB	Line	CX006E51	solid	8	22pt	30pt	bold	REPORTB	2017 B
SSS_AAA	Line	CXF58025	solid	8	22pt	30pt	bold	REPORTB	2017 SA
UU_15	Line	CXDA81F5	solid	8	22pt	30pt	bold	REPORTB	2017 HA15
AAA	Line	CX0069AA	solid	8	22pt	30pt	bold	REPORTA	2017 A
BBB	Line	CX006E51	solid	8	22pt	30pt	bold	REPORTA	2017 B
SSS_AAA	Line	CXF58025	solid	8	22pt	30pt	bold	REPORTA	2017 SA

Table2 covers common parameters, it is not changing as per each situation.

commonparam	gridcolor	gridpattern	plotpattern	boardcolor	ROWGUTTER	COLUMNGUTTER
REPORTA	darkgrey	dash	solid	white	2cm	2cm
REPORTB	grey	MediumDash	MediumDash	white	2cm	2cm

Table3 is for position, in our example, we use 2x2 cells, UL, UR, LL, and LR.

PlotID	position	SlideNumber
Plot1	UL	Slide1
Plot2	UR	Slide1
Plot3	LL	Slide1

Plot4	LR	Slide1
Plot5	UL	Slide2
Plot6	UR	Slide2
Plot7	LL	Slide2
Plot8	LR	Slide2

Step2: **Solutions of Challenge 2, 3 & 4** --- automatic code generation method to cover all changing parameters. Automatic code generation method can easily cover most of the changing parameters. Here are one example of automatic code generation method.

```
data legenditem_seriesplot_values;
set chart_type_add_position;
Legend_Value =strip(comparison)|| '  '||strip(Year);
length Line_legendItem $500. Line_seriesplot $500.;
legendItem = 'legendItem'||' type='||strip(legendtype)||' name='||'"'||strip(Situation)||'_'||
       strip(Purpose)||'"  '||'/ lineattrs=(color='||strip(linecolor)||'
pattern='||strip(pattern)||' THICKNESS='||
       strip(thick)||')'||' labelattrs=(size='||strip(labelsize)||')'||'
label='||'"'||strip(Legend_Value)||'"'||';';
/*legendItem  type=Line name="AAA_REPORTB2017" / lineattrs=(color=CX0069AA
   pattern=solid  THICKNESS=8) labelattrs=(size=22pt) label="REPORTB 2017 A"; */

seriesplot='seriesplot'||' x=xaxis y='||strip(Comparison)||'_'||
      strip(SituatKeys)||'_'||strip(Position)
      ||'/LEGENDLABEL='||'"'||strip(Legend_Value)||'"'||' PRIMARY=TRUE lineattrs=(color='||
       strip(linecolor)||' pattern='||strip(pattern)||' THICKNESS='||strip(thick)||');';
/*seriesplot  x=xaxis  y=REPORTB2017_BBB_UL / LEGENDLABEL="REPORTB 2017 B"
   PRIMARY=TRUE  lineattrs=(color=CX006E51  pattern=solid THICKNESS=8); */
run;
```

Then define these variables like legendItem, seriesplot as macro parameters that can cover most options parameters. Use &&legendItem_&pageID._&Position, &&seriesplot_&pageID._&Position as macro parameter in the line chart **PROC TEMPLATE** engine. This way can perfectly cover the changing of parameters.

```
proc sql;
select strip(legendItem) into :legendItem_&pageID._&Position separated by ' '
from legenditem_seriesplot_values where Chart_ID="&&Chart_ID_&Position ";
select strip(seriesplot) into :seriesplot_&pageID._&Position separated by ' '
from legenditem_seriesplot_values where Chart_ID="&&Chart_ID_&position";
quit;
```

Through Table3, we can automatically calculate the numbers of columns and row numbers to solve **Challenge 4** below --- columns=2 rows=2.

```
proc template;
define statgraph a.class;
begingraph;
discreteattrvar attrvar=classfill var=fore_sce attrmap='colors';
[1] &&legendItem_&pageID._&Positn;

layout lattice/border=FALSE BORDERATTRS=(color=white) columns=2 rows=2
    COLUMNGUTTER=2cm ROWGUTTER=2cm;

cell;  /*Up Left cell, simply as UL*/
cellheader;
entry "UL Graph"/border=FALSE BORDERATTRS=(color=white) textattrs=(size=30pt weight=bold);
endcellheader;
layout overlay /border=false WALLCOLOR=white WALLDISPLAY=(FILL)
[2] &&xaxisopts_&pageID._&position;
    &&yaxisopts_&pageID._&position;
[3] &&seriesplot_&pageID._&position;
```

```
    &&reference_&pageID._&position;
endlayout;
endcell;

cell;  …   endcell;    /*Up Right cell, UR*/
cell;  …   endcell;    /*Low Left cell, LL*/
cell;  …   endcell;    /*Low Right cell,LR*/
endlayout;

entryfootnote halign=left TEXTATTRS=(size=22pt weight=bold)  "Slide5=X"
              halign=right TEXTATTRS=(size=22pt weight=bold) "Release 1 ";

layout  globallegend/border=false type=COLUMN weights=uniform;
[4] &&discretelegend1_&pageID;
    &&discretelegend2_&pageID;
endlayout;
endgraph;
end;
run;
proc sgrender data=grp_comp.Slide_5_Line template=a.class;  Run;
```

This Automatic SAS Code Generation Method was clearly explained and presented in my paper ---- *How to automatically cover arbitrary changes using automatic code generation method in SAS Enterprise Guide*. [a]

Step3, Solutions of Challenge 5 & 6 ---- How to automatically calculate the parameter values of yaxisopts (origin, viewmin, tickvaluelist, tickvalueformat), xaxixopts (tickdisplaylist and xaxixopts tickvaluelist), barchart statement (barlabelformat), referenceline statement (curvelabel) is a big and very complex topic which is related to mathematical and coding logic, fortunately I developed a SAS engine to automatically calculate them, For detailed information, please find detailed information from *The Calculator of the Graphic y-axis Tickvaluelist, Tickvalueformat With Expected Values for SAS PROC TEMPLATE*. [b]

```
layout overlay /border=false WALLCOLOR=white WALLDISPLAY=(FILL)
xaxisopts=(display=all linearopts=(origin=0 THRESHOLDMIN=1 THRESHOLDMAX=1)
  label=' ' griddisplay=off labelattrs=(weight=bold size=22pt) tickvalueattrs=(weight=bold
  size=22pt) discreteopts=(tickdisplaylist=(&&xtickdisplaylist_&pageID._&position)
  tickvaluelist=(&&xtickvaluelist_&pageID._&position)
  TICKVALUEFITPOLICY=ROTATEALWAYS   TICKVALUEROTATION=VERTICAL))
yaxisopts=(display=(label ticks tickvalues) label=' ' labelattrs=(weight=bold size=22pt)
  griddisplay=on tickvalueattrs=(weight=bold size=22pt) linearopts=(integer=FALSE
  origin=&&origin_&pageID._&position  viewmin=&&origin_&pageID._&position
  tickvalueformat=&&tickvalueformat_&pageID._&position
  tickvaluelist=(&&ytickvaluelist_&pageID._&position) TICKVALUEPRIORITY=TRUE)
  gridattrs=(pattern=dash THICKNESS=2 color=darkgrey));

referenceline y=0/curvelabel="&&curvelabel_&pageID._&position" curvelabelattrs=(
  size=22pt weight=bold) clip=true curvelabelposition=min
  lineattrs=(pattern=dash THICKNESS=2 color=darkgrey);
```

Step4, Some tricks: during the engine development, there are some tricks during plotting composite or single images for linechart, barchart, and other plots. I list them in the paper of **[c]** Some Tricks and Explanations When Plotting Graphic Images Using PROC TEMPLATE SAS Enterprise Guide Part III. If you can learn from the Part III I believe you can save your lots of time.

CONTACT INFORMATION

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

Jing Wang
Sr. SAS Porgrammer
Sr. Data Scientist
China Minsheng Bank

Kaiqing Fan
Sr. Data Scientist
Sr. SAS Developer Lead
Sr. Risk Predictive Modeler
Mastech Digital Inc.
Address: 6750 Miller Road, Brecksville, OH-44141
Mobile: 504.344.7267
Email: fankaiqinguw@gmail.com
Linkedin: https://www.linkedin.com/in/fan-kaiqing-81776940/

REFERENCES

[a] Kaiqing Fan
How to automatically cover arbitrary changes using automatic code generation method in SAS Enterprise Guide

[b] Kaiqing Fan
The Calculator of the Graphic y-axis Tickvaluelist, Tickvalueformat With Expected Values for SAS PROC TEMPLATE Part II

[c] Kaiqing Fan
Some Tricks and Explanations When Plotting Graphic Images Using PROC TEMPLATE SAS Enterprise Guide Part III