

SAS[®] GLOBAL FORUM 2018

USERS PROGRAM

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

Kaiqing Fan, Mastech Digital Inc., Brecksville, OH

April 8 - 11 | Denver, CO
#SASGF

Some Tricks When Plotting Graphic Images Using PROC TEMPLATE SAS® Enterprise Guide Part III

Kaiqing Fan, Mastech Digital Inc.

Abstract

Without affecting the wholeness of the **Part I** and **Part II**, I split the part of tricks as the Part III here to present and explain these tricks here that may confuse you when you use them in your SAS codes as example. These tricks include the options in **PROC TEMPLATE** layout overlay and layout lattice statements, y-tick values calculator. Without clear presenting them, developers may repeat same mistakes when encounter same issues.

Key words

Technical tricks,
x and y-tick value list calculator,
x and y-tick value display list calculator,
PROC TEMPLATE

Some Some Tricks When Plotting Graphic Images Using PROC TEMPLATE SAS® Enterprise Guide Part III

Kaiqing Fan, Mastech Digital Inc.

One, the value of curvelabel, y-tickvaluelist and y-tickdisplaylist

Here is the rule:

1) when the minimum or maximum of the y values is 0, we should set

referenceline y=0 / curvelabel="0.0%" or "0.000%" with values;

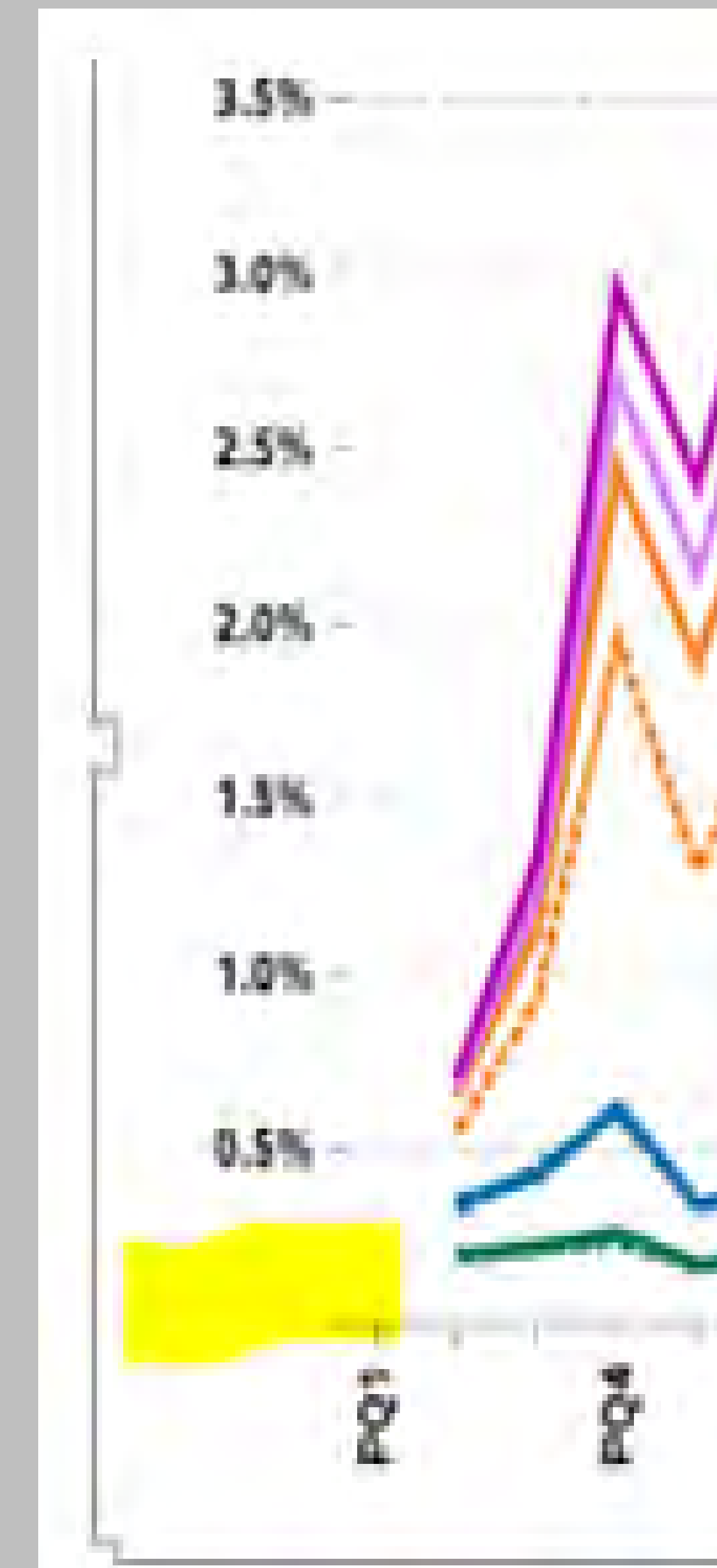
2) when the minimum or maximum of the y values is not 0, we should set

curvelabel=" " blank ;

If only set as curvelabel=" " blank, then it may cause the 0 y-tickvalue is missing;

If only set as curvelabel="with value" blank, then it may cause the 0 y-tickvalue is overlaid.

Troubles



Some Tricks When Plotting Graphic Images Using PROC TEMPLATE SAS® Enterprise Guide Part III

Kaiqing Fan, Mastech Digital Inc.

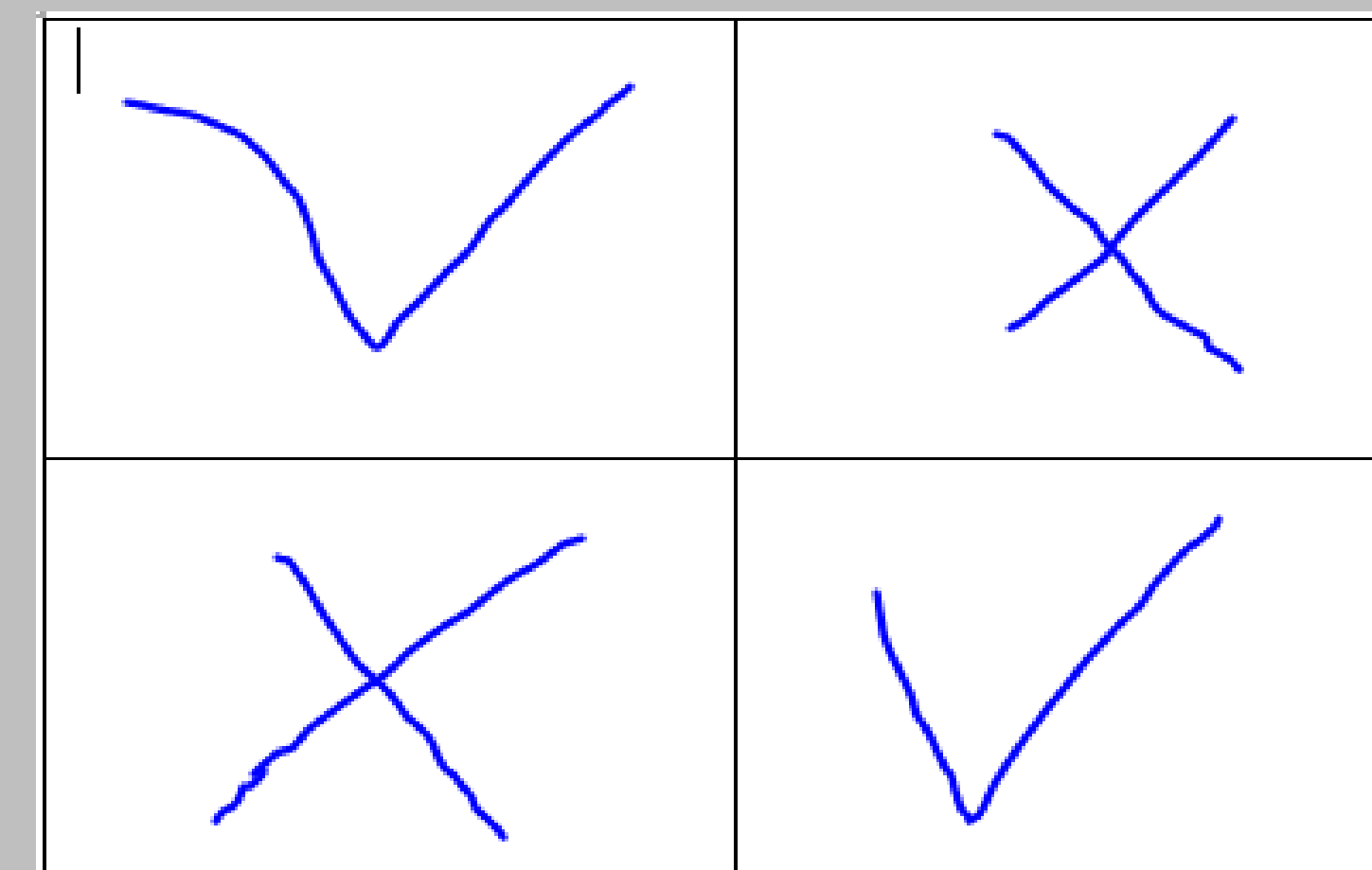
Two, the fixed position sequence position of each cell

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

If the sequences of positions were required to be fixed, if any cell was missing, it would be blank on its position. We should use the statement without any option parameters to present the blank image to keep its position. If any one of their statements was missing, the one behind it would move forward to fill its position.

Expected 2x2 image

Let's see the 2x2 composite images with blanks, In the following image, **X is blank images**, **V is real images**.



UL	1	UR	2
LL	3	LR	4

The sequence of the four positions would be
UL, UR,
LL, LR ;

Some Tricks When Plotting Graphic Images Using PROC TEMPLATE SAS® Enterprise Guide Part III

Kaiqing Fan, Mastech Digital Inc.

How to fill the specified position with missing image?

*/*fill the missing image with the following codes for position UR, LL*/*

```
cell; /*Up Right or Low Left cell, UR or LL, Empty*/
cellheader;
entry " " /border=false BORDERATTRS=(color=white);
endcellheader;
layout overlay;
endlayout;
endcell;
```

The specified position with expected image

```
cell; /*Up Left cell, UL*/
cellheader;
entry "Title UL" /border=false BORDERATTRS=(color=white) textattrs=(size=30pt weight=bold);
endcellheader;
layout overlay / border=false WALLCOLOR=white WALLDISPLAY=(FILL)
xaxisopts=(display=all linearopts=(origin=0 THRESHOLDMIN=1 THRESHOLDMAX=1)
label=' ' labelFitPolicy=Split labelattrs=(weight=bold)
tickvalueattrs=(size=22pt weight=bold)
griddisplay=off discreteopts=(tickvaluefitpolicy=splitalways
tickvaluesplitchar=" " tickvaluesplitjustify=CENTER))
yaxisopts=(display=(label ticks tickvalues) label=' '
labelattrs=(weight=bold SIZE=22pt) griddisplay=on
gridattrs=(pattern=solid THICKNESS=1 color=darkgrey) |
tickvalueattrs=(size=22pt weight=bold)
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;
endlayout;
endcell;
```

Some Tricks When Plotting Graphic Images Using PROC TEMPLATE SAS® Enterprise Guide Part III

Kaiqing Fan, Mastech Digital Inc.

Three, tricks in x & y-tickvalues calculator

1), why I divide the min and max values with 0.001, or 0.0001, or 0.00001, ... if all the absolute values of numbers are falling between 1 and -1?

If we want to get the integer number from decimal number, we need to use **round**, **ceil**, **floor** functions, they cannot reach our expected values between 1 and -1.

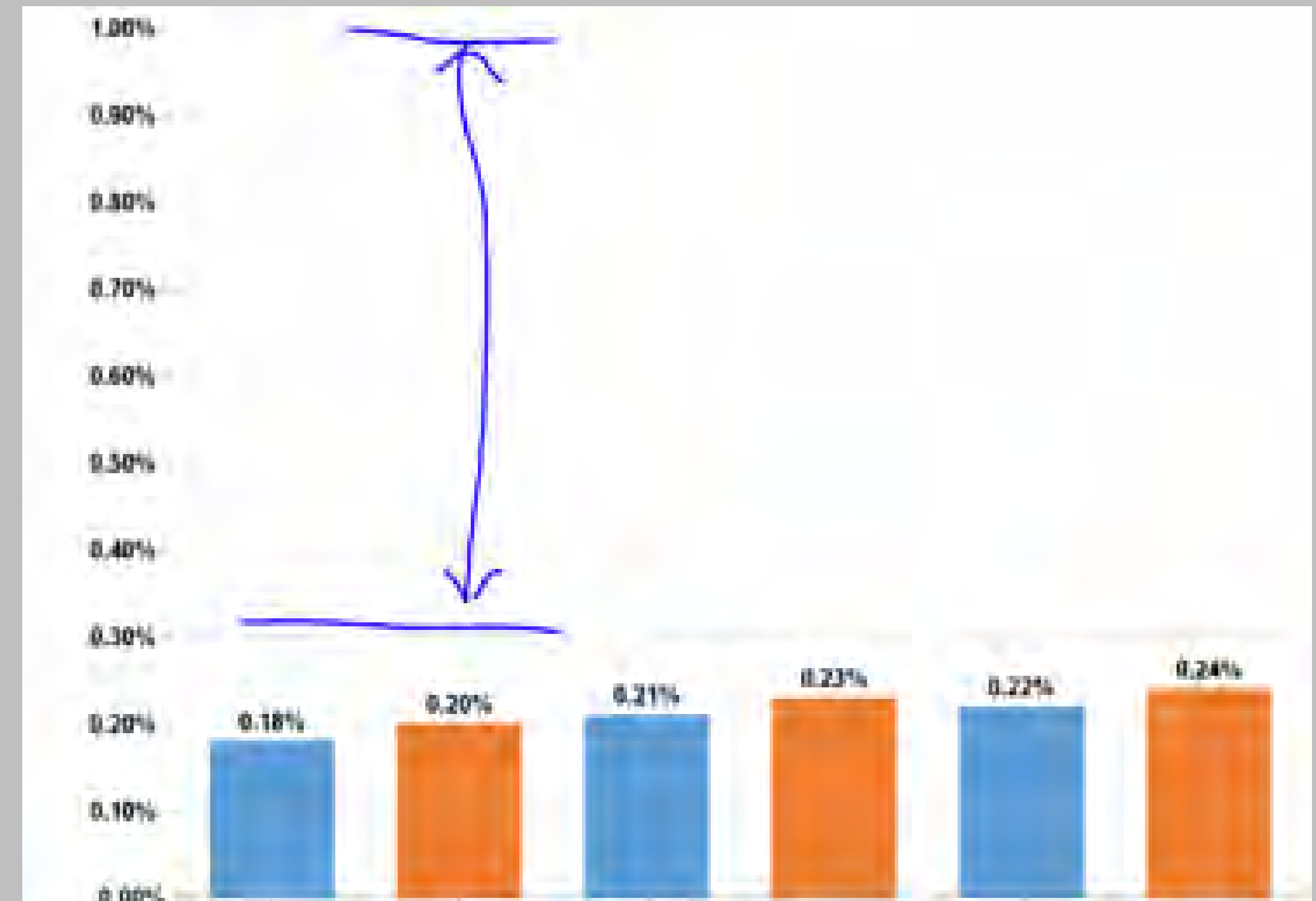
2), we need always to keep 0 in the y-tick values.

*/*if min is positive, then reset min as 0, else set as itself, if max is negative, then reset max as 0, else set as itself*/*

```
%if %index("&min_value",-)=0 %then %do; %let startvalue=0; %end;
%if %index("&min_value",-)>0 %then %do; %let startvalue=&min_value; %end;
%if %index("&max_value",-)>0 %then %do; %let endvalue=0; %end;
%if %index("&max_value",-)=0 %then %do; %let endvalue=&max_value; %end;
```

Three, tricks in y-tickvalues calculator (cont'd)

3), why do we need rescaling the y-tick values?



Some Tricks When Plotting Graphic Images Using PROC TEMPLATE SAS® Enterprise Guide Part III

Kaiqing Fan, Mastech Digital Inc.

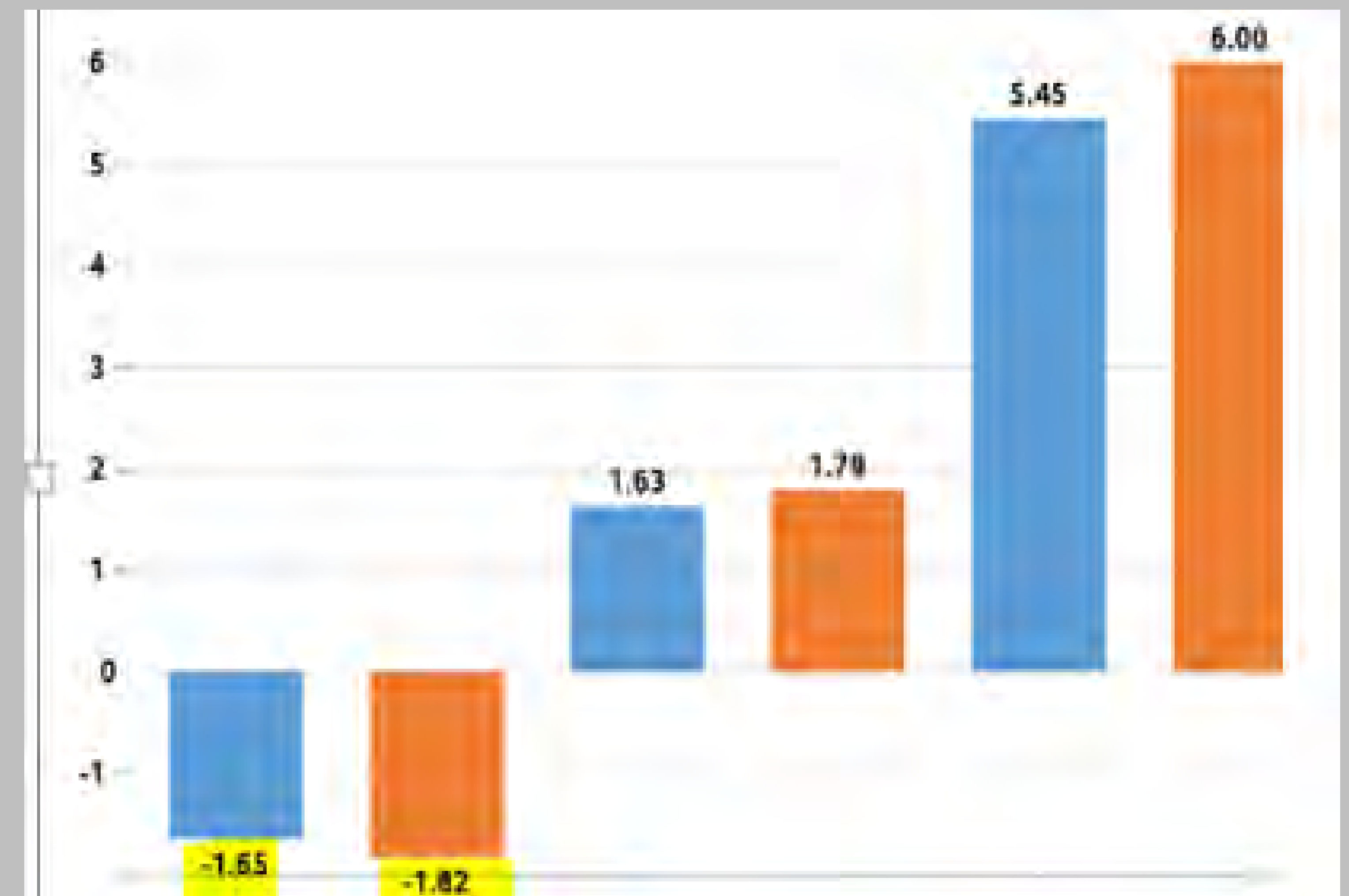
Three, tricks in y-tickvalue calculator (cont'd)

4), we can control the numbers of y-tick values through changing the following numbers in the highlighted part.

```
if 12 < (upper_band-lower_band)/incremental_value then redivider=.5;  
if 6 < (upper_band-lower_band)/incremental_value < 13 then redivider=1;  
if 3 < (upper_band-lower_band)/incremental_value < 7 then redivider=2;  
if 1 < (upper_band-lower_band)/incremental_value < 4 then redivider=4;  
if 0 =< (upper_band-lower_band)/incremental_value < 2 then redivider=10;  
reincremental_value=incremental_value/redivider;  
reTickcount=(upper_band-lower_band)/reincremental_value;  
reTickdigitNumber=ceil(reTickcount);  
If reTickdigitNumber < 15 then multiplier=1;  
if 14 < reTickdigitNumber < 28 then multiplier=2;  
if 27 < reTickdigitNumber then multiplier=3;
```

Three, tricks in y-tickvalue calculator (cont'd)

5), for the bar chart image, why do we need to minus one increment value?



Some Tricks When Plotting Graphic Images Using PROC TEMPLATE SAS® Enterprise Guide Part III

Kaiqing Fan, Mastech Digital Inc.

Three, tricks in y-tickvalues calculator (cont'd)

6), The SAS system 0 may not be 0, it may be like 0.000000000000030.

Under this situation we always need to set it as 0, that is why SAS system 0 values may not be 0.

*/*remove system round values because sometimes, 0 is not 0*/*

```
data grp_comp.line6_check0_&page._&postn;  
set grp_comp.line6_check0_&page._&postn;  
if 0.000000000001>COL1> -0.000000000001 then COL1=0;  
run;
```

Three, tricks in y-tickvalues calculator (cont'd)

7), 0 may not be in the list of COL1 or ytickvalues.

We can use referenceline statement curvelabel to insert it, but the inserted referenceline 0 and y-tick tickvaluelist may cause unequal distances because of the inserted 0. So we added 0 into the ytickvaluelist, and create equal distances between any two values.

N		
N		
-0		
T		
N		
N		

Some Tricks When Plotting Graphic Images Using PROC TEMPLATE SAS® Enterprise Guide Part III

Kaiqing Fan, Mastech Digital Inc.

Three, tricks in y-tickvalues calculator (cont'd)

8), the trick of y-tick tickvalueformat – better be generalized.

For our case, we use two kind of numbers: **regular number** like 18.3386 and **percentage** like 12.3456%, so we use two different format values as **form=compress("percentn9." || len1);** and **form=compress("9." || len1);**

We better use **percentn9.** , not **percentn8.**; if we use **percentn8.**, because some numbers are very huge, we may not cover them. Sometimes this situation is very difficult to notice.

If we use fix value for the format, we must have round troubles such as 2.5 round up to 3.

We can also use

```
proc format ;  
picture million (round)  
low-<0 ='09.99M' (prefix="-$" mult=.00001)  
0-high ='09.99M' (prefix="$" mult=.00001);  
run;
```

Three, tricks in y-tickvalues calculator (cont'd)

9), how can we pick x-axis tickdisplaylist values to avoid crowded or overlaid.

```
/*calculate xaxis tickdisplaylist and define it as macro parameter*/  
data grp_comp.line12_xtickdisplaylist_&page(keep=xtickdisplay);  
set grp_comp.Slide_&page._Line(keep=xaxis) nobs=nobs end=last;  
ID=_N_;  
xtickdisplay="" || strip(xaxis) || "";  
do i=1 to nobs-1;  
if mod(ID,3) ne 1 then do;  
/*automatically pick one xtickvalue from every 3 to avoid overlay  
each other */  
xtickdisplay="";  
end;  
end;  
run;
```

Some Tricks When Plotting Graphic Images Using PROC TEMPLATE SAS® Enterprise Guide Part III

Kaiqing Fan, Mastech Digital Inc.

Conclusion

When you assembly line plot graphics using **PROC TEMPLATE** and using *How to Assembly Line Create Graphic Images Using PROC TEMPLATE In SAS Enterprise Guide? (Part I)* and *The Calculator of the Graphic y and x-axis Tickvaluelist, Tickvalueformat with Expected Values for SAS PROC TEMPLATE (Part II)* to calculate the y and x **tickvaluelist** or **tickvaluedisplaylist**, it would be very helpful for you to read this paper to understand its logic to avoid many tricks which may cost you too much time to find and fix them.

References

[a] Kaiqing Fan

How to Assembly Line Create Graphic Images Using PROC TEMPLATE In SAS Enterprise Guide? (Part I)

[b] Kaiqing Fan

The Calculator of the Graphic y and x-axis Tickvaluelist, Tickvalueformat with Perfect Values for SAS PROC TEMPLATE (Part II)

[c] **PROC FORMAT** and its examples

<http://documentation.sas.com/?docsetId=proc&docsetVersion=9.4&docsetTarget=p0n990vq8gxca6n1vnsracr6jp2c.htm&locale=en>

<http://documentation.sas.com/?docsetId=proc&docsetTarget=n0kl9qj532rbqln187us4ao371h7.htm&docsetVersion=9.4&locale=en#n0kl9qj532rbqln187us4ao371h7>

Some Tricks When Plotting Graphic Images Using PROC TEMPLATE SAS[®] Enterprise Guide Part III

Kaiqing Fan, Mastech Digital Inc.

CONTACT INFORMATION

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

Kaiqing Fan

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/>

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.



SAS[®] GLOBAL FORUM 2018

April 8 - 11 | Denver, CO
Colorado Convention Center

#SASGF

Some Tricks and Explanations When Plotting Graphic Images

Using PROC TEMPLATE SAS[®] Enterprise Guide

Part III

Kaiqing Fan, Mastech Digital Inc., Brecksville, Ohio

Abstract

Without affecting the wholeness of the Part I and Part II, I split the part of tricks as the Part III here to present and explain these tricks here that may confuse you when you use them in your SAS codes as example. These tricks include the options in **PROC TEMPLATE** [layout overlay](#) and [layout lattice](#) statements, y-tick values calculator. Without clear presenting them, developers may repeat same mistakes when encounter same issues.

Key words: Technical tricks, x and y-tick value list calculator, x and y-tick value display list calculator, PROC TEMPLATE

Introduction

Without affecting the wholeness of the Part I and Part II, I split the tricks as the Part III to explain and present these tricks that may confuse you when you encounter them in your SAS codes as example. Here are the trick list:

One, the Relationship among the Value of [curvelabel](#), [yaxisopts tickvaluelist](#) and [yaxisopts tickdisplaylist](#).

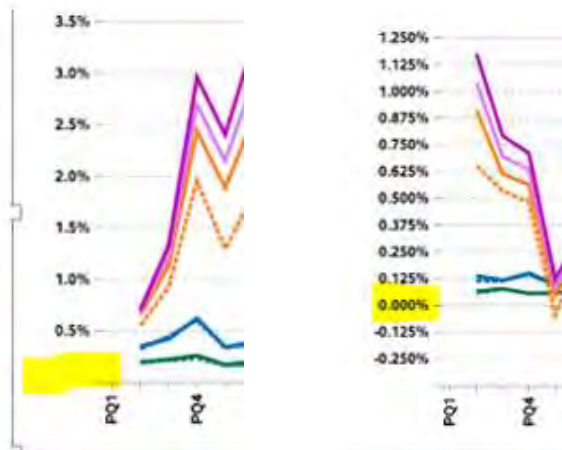
Here are the rules I found:

- 1) When the 0 is the maximum or minimum values of [tickvaluelist](#) of [yaxisopts](#), if we only define them through [yaxisopts tickvaluelist](#) or [yaxisopts tickdisplaylist](#), with defining [referenceline y=0 / curvelabel=""](#), the reference line will be missing without 0 like **situation one** the left image. Under this situation, we should define [referenceline y=0 / curvelabel="0.00%"](#) or ["0"](#).
- 2) When the 0 is not the maximum or minimum values of [yaxisopts tickvaluelist](#), if we define [referenceline y=0 / curvelabel="0"](#) or ["0.00%"](#) or

other values, it will be overlaid with the 0 value in the `yaxisopts tickvaluelist` like **situation two** the right image. Under this situation, we should use `referenceline y=0 / curvelabel=""`.

Let's see the following four situations, then you would understand why we should get the above solutions:

Situation one, if we define `referenceline y=0 / curvelabel=""` -----missing here, it causes missing y tick value in the image.



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

```
cell;
cellheader;
entry "Title1" / border=FALSE BORDERATTRS=(color=white)
                textattrs=(size=30pt weight=bold);
endcellheader;
```

```
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
```

```
/*left image*/
tickvalueformat=percentn9.1
```



```

tickvalueformat=percentn9.1
tickvaluelist=(0 0.005 0.01 0.015 0.02 0.025 0.03 0.035)
tickdisplaylist=("0.0%" "0.5%" "1.0%" "1.5%" "2.0%" "2.5%"
"3.0%" "3.5%")
TICKVALUEPRIORITY=TRUE)
gridattrs=(pattern=dash THICKNESS=2 color=darkgrey));
    
```

(seriesplot statements are ignored here;)

```

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

```

endlayout;
endcell;
    
```

*/*right image*/*

```

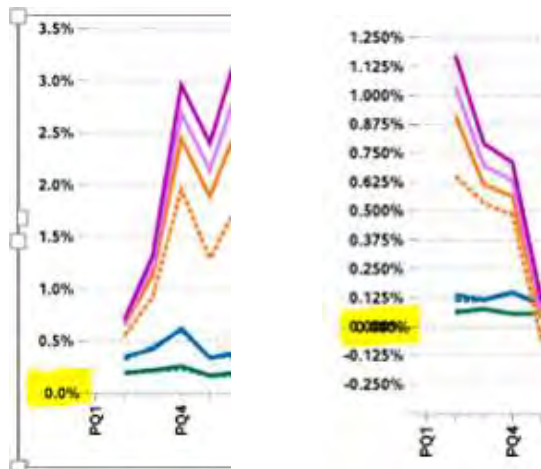
tickvalueformat=percentn9.1
tickvaluelist=(0 0.005 0.01 0.015 0.02 0.025 0.03 0.035)
tickdisplaylist=("0.0%" "0.5%" "1.0%" "1.5%" "2.0%" "2.5%"
"3.0%" "3.5%")
TICKVALUEPRIORITY=TRUE)
gridattrs=(pattern=dash THICKNESS=2 color=darkgrey));
    
```

```

referenceline y=0 / curvelabel="" curvelabelattrs=(size=22pt weight=bold)
    
```

Here both `referenceline y=0 / curvelabel=""` are blank. It is not good for the left case with minimum is 0, the reference line is with missing value; but it is good for the right case that 0 is not the minimum or maximum values.

Situation two, if we define `referenceline y=0 / curvelabel="0.0%"` or `"0.000%"`----- not missing here, it causes overlaid y tick values in the image.



```

tickvalueformat=percentn9.1
tickvaluelist=(0 0.005 0.01 0.015 0.02 0.025 0.03 0.035)
    
```

```
tickdisplaylist=("0.0%" "0.5%" "1.0%" "1.5%" "2.0%" "2.5%"
"3.0%" "3.5%")
```

```
referenceline y=0 / curvelabel="0.0%" curvelabelattrs=(size=22pt weight=bold)
```

```
tickvalueformat=percentn9.3
```

```
tickvaluelist=(-0.00250 -0.00125 0 0.00125 0.00250 0.00375
0.00500 0.00625 0.00750 0.00875 0.01000 0.01125 0.01250)
```

```
tickdisplaylist=(-0.250%" "-0.125%" "0.000%" "0.125%"
"0.250%" "0.375%" "0.500%" "0.625%" "0.750%" "0.875%"
"1.000%" "1.125%" "1.250%")
```

```
referenceline y=0 / curvelabel="0.000%" curvelabelattrs=(size=22pt weight=bold)
```

Here `referenceline y=0/ curvelabel="0.0%"` or `"0.000%"`. The left image is good for the left case with minimum is 0; but it is not good for the right case that 0 is not the minimum or maximum values, it causes overlaid at 0 position.

Situation three, if we set 0 value as missing, and define `referenceline y=0/ curvelabel="0.0%"` or `"0.000%"`, it does not work too.

Let's try `tickdisplaylist` with missing value at 0 position.

```
data grp comp.line11 yticklist &image n. &postn. &graph number;
  set grp comp.line10 yticklist &image n. &postn. &graph number;
  format ytickvalues char &&tickvalueformat_&image_n. &postn. &graph_number;
  ytickvalues_char = ytickvalues;
  if COL1=0 then ytickvalues_char = . ;
run;
```

	_NAME	COL1	len	lenh	lenl	form	ytickvalues	ytickvalues_char
1	ytickvalues_0	-0.00375	5	3	3	percentn9.3	0.375%	-0.375%
2	ytickvalues_2	-0.00125	5	3	3	percentn9.3	-0.125%	-0.125%
3	ytickvalues_4	0.00125	5	3	3	percentn9.3	0.125%	0.125%
4	ytickvalues_6	0.00375	5	3	3	percentn9.3	0.375%	0.375%
5	ytickvalues_8	0.00625	5	3	3	percentn9.3	0.625%	0.625%
6	ytickvalues_10	0.00875	5	3	3	percentn9.3	0.875%	0.875%
7	ytickvalues_12	0.01125	5	3	3	percentn9.3	1.125%	1.125%
8	ytickvalues_1	-0.0025	4	2	2	percentn9.2	-0.250%	-0.250%
9	ytickvalues_5	0.0025	4	2	2	percentn9.2	0.250%	0.250%
10	ytickvalues_9	0.0075	4	2	2	percentn9.2	0.750%	0.750%
11	ytickvalues_13	0.0125	4	2	2	percentn9.2	1.250%	1.250%
12	ytickvalues_7	0.005	3	1	1	percentn9.1	0.500%	0.500%
13	ytickvalues_11	0.01	2	0	0	percentn9.0	1.000%	1.000%
14	ytickvalues_3	0	0	0	0	percentn9.0	0.000%	

```
proc sql;
  select distinct ytickvalues
  into :curvelabel &image n. &postn. &graph number
  from grp comp.line10_yticklist_&image_n. &postn. &graph_number
  where COL1 in (0);
quit;
```

```

proc sql;
  select distinct ytickvalues _char
  into :ytickdisplaylist _&image_n._&postn._&graph_number
  separated by " "
  from grp_comp.line11_yticklist _&image_n._&postn._&graph_number;
quit;

tickvalueformat=percentn9.3
tickvaluelist=(-0.00375 -0.0025 -0.00125 0 0.00125 0.0025 0.00375
              0.005 0.00625 0.0075 0.00875 0.01 0.01125 0.0125)
tickdisplaylist=(" " "-0.375%" "-0.250%" "-0.125%" "0.125%"
                "0.250%" "0.375%" "0.500%" "0.625%" "0.750%"
                "0.875%" "1.000%" "1.125%" "1.250%")
referenceline y=0 / curvelabel="0.000%" curvelabelattrs=(size=22pt weight=bold)

```

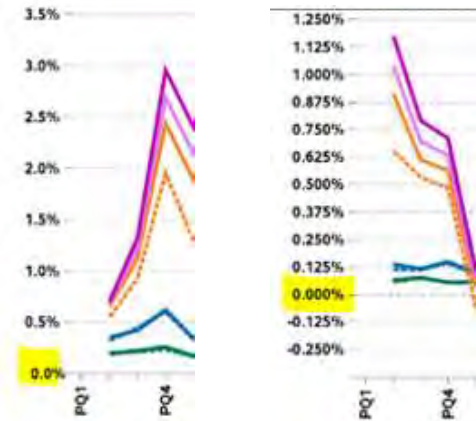


Because missing value was at the bottom, there was no space at the place of should-be-0. So the above value -0.125% would move up to the should-be-0 position, overlaid with the curvelabel value **0.000%**, it looks bad.

If we can put the missing value to the should-be-0 position, it would be OK. But it is very difficult. Looks like the missing value is always at the bottom of the y-axis.

Situation four is the perfect solution: we should set **referenceline y=0 / curvelabel="0.0%"** or **"0.000%"** when the minimum or maximum is 0; set **curvelabel=""** blank when the minimum or maximum is not 0. We will be OK.

In short, if its max or min is 0, then define curvelabel as 0 with its format, or define curvelabel as ' ' blank .



```

/*if its max or min is 0, then define curvelabel as 0 with its format*/
/***** or define curvelabel as ' ' blank *****/
%if %index(%sysfunc(compress("a&maxCOL1.a")),a0a)>0
  or %index(%sysfunc(compress("a&minCOL1.a")),a0a)>0 %then %do;
  proc sql;
  select distinct ytickvalues into :curvelabel &page. &postn
  from grp_comp.line10_yticklist_&page._&postn where COL1 in (0);
  quit;
%end;
%else %do;
%let curvelabel_&page._&postn=;
%end;

tickvalueformat=&&tickvalueformat_&image_n._&pos._&graphn
tickvaluelist=(&&ytickvaluelist_&image_n._&pos._&graphn)
tickdisplaylist=("&&ytickdisplaylist_&image_n._&pos._&graphn")
gridattrs=(pattern=dash THICKNESS=2 color=darkgrey));

&&Line_seriesplot_&image_n._&graphn._&pos;

referenceline y=0/ curvelabel="&&curvelabel_&image_n._&pos._&graphn"
curvelabelattrs=(size=&font_size weight=bold)

```

Two, the Fixed Sequence of Position of Each Cell Image in the layout lattice statement in PROC TEMPLATE.

```

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

```

If the sequences of positions were required to be fixed, if any cell was missing, it would be blank on its position. We should use the statement without any option parameters to present the blank image to keep its position. If any one of their statements was missing, the one behind it would move forward to fill its position.

Some options may cause confusions:

1), **AUTOALIGN=(TOPRIGHT BOTTOMLEFT)** --- AUTOALIGN has many options, but any option with/out it does not affect the sequence of positions.

2), **SKIPEMPTYCELLS=TRUE|FALSE** does same thing as **AUTOALIGN** here because these options are ignored if this **layout statement is the outermost layout or if the parent layout is not an overlay-type layout.**

3), **VALIGN=CENTER | TOP | BOTTOM | number** .This option has effect only when this layout is **nested** within a region layout, or when this layout is nested in an overlay-type layout and **AUTOALIGN=NONE**.

An example from SAS official website:

```
dynamic VAR STAT1 STAT2 STAT3;
```

```
layout overlay / height=500px width=600px;
  histogram VAR;
  layout lattice / VALIGN=TOP HALIGN=RIGHT
    height=80px width=70px columns=1;
    entry STAT1;
    entry STAT2;
    entry STAT3;
  endlayout;
endlayout;
```

If we want to plot a 2 by 2 cells images, the **up left** and **up right** are true images, **low left** and **low right** are blanks, then we should develop codes like the following:

The sequence of the four positions would be UL, UR,
LL, LR ;

UL	1	UR	2
LL	3	LR	4

The sequence of the six positions would be UL, UM, UR,

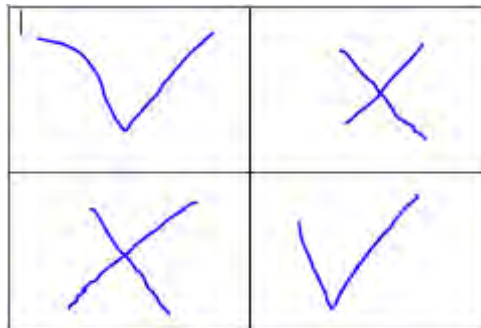
LL, LM, LR ; M is middle.

UL	1	UM	2	UR	3
LL	4	LM	5	LR	6

The sequence of the nine positions would be UL, UM, UR,
ML, MM, MR, M is middle.
LL, LM, LR.

UL	1	UM	2	UR	3
ML	4	MM	5	MR	6
LL	7	LM	8	LR	9

Let's see the 2x2 composite images with blanks:



In the above image, X means blank images, V means real images.

```

layout lattice/border=FALSE BORDERATTRS=(color=white) columns=2 rows=2
    COLUMNGUTTER=2cm ROWGUTTER=2cm;
/* here don't need AUTOALIGN=(TOPRIGHT BOTTOMLEFT) SKIPEMPTYCELLS=TRUE;*/

cell; /*Up Left cell, UL*/
cellheader;
entry "Title UL" /border=false BORDERATTRS=(color=white) textattrs=(size=30pt weight=bold);
endcellheader;
layout overlay / border=false WALLCOLOR=white WALLDISPLAY=(FILL)
    xaxisopts=(display=all linearopts=(origin=0 THRESHOLDMIN=1 THRESHOLDMAX=1)
        label=' ' labelFitPolicy=Split labelattrs=(weight=bold)
        tickvalueattrs=(size=22pt weight=bold)
        griddisplay=off discreteopts=(tickvaluefitpolicy=splitalways
        tickvaluesplitchar=" " tickvaluesplitjustify=CENTER))
    yaxisopts=(display=(label ticks tickvalues) label=' '
        labelattrs=(weight=bold SIZE=22pt) griddisplay=on
        gridattrs=(pattern=solid THICKNESS=1 color=darkgrey)
        tickvalueattrs=(size=22pt weight=bold)
        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;
endlayout;
endcell;

cell; /*Up Right cell, UR, Empty*/
cellheader;
  entry ""/border=false BORDERATTRS=(color=white);
endcellheader;
layout overlay;
endlayout;
endcell;

cell; /*Low Left cell, LL, empty */
cellheader;
  entry ""/border=false BORDERATTRS=(color=white);
endcellheader;
layout overlay;
endlayout;
endcell;
endlayout;

cell; /*Low Right cell, LR*/
cellheader;
entry "Title UR" /border=false BORDERATTRS=(color=white) textattrs=(size=30pt weight=bold);
endcellheader;
layout overlay / border=false WALLCOLOR=white WALLDISPLAY=(FILL)
  xaxisopts=(display=all linearopts=(origin=0 THRESHOLDMIN=1 THRESHOLDMAX=1)
  label=' ' labelFitPolicy=Split labelattrs=(weight=bold)
  tickvalueattrs=(size=22pt weight=bold)
  griddisplay=off discreteopts=(tickvaluefitpolicy=splitalways
  tickvaluesplitchar=" " tickvaluesplitjustify=CENTER))
  yaxisopts=(display=(label ticks tickvalues) label=' '
  labelattrs=(weight=bold SIZE=22pt) griddisplay=on
  gridattrs=(pattern=solid THICKNESS=1 color=darkgrey)
  tickvalueattrs=(size=22pt weight=bold)
  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;
endlayout;
endcell;

```

Three, some Tricks in y-tick values Calculator Engine

1), why I divide the min and max values with 0.001 if all the absolute values of numbers are falling between 1 and -1?

If we want to get the integer number from decimal number, we need to use round, ceil, floor functions, they cannot reach our expected values between 1 and -1.

For my case, we use two kind of numbers: regular number like 108.3386 and percentage like 12.3456%. So 0.001 or 0.01 are enough for us.

We must generalize 0.01 or 0.001 case by case, an exceptional example is that all numbers were between 0.0000001 and -0.0000001.

A good solution may be the following:

```

Data calculate_scale;
%if (%sysfunc(abs(&minPQ))<1 and %sysfunc(abs(&minPQ))=0) and
(%sysfunc(abs(&maxPQ))<1 and %sysfunc(abs(&maxPQ))=0) %then %do;
  %if %sysfunc(abs(&minPQ)) < %sysfunc(abs(&maxPQ)) %then
    scale = 10/%sysfunc(abs(&minPQ));
  %if %sysfunc(abs(&minPQ)) > %sysfunc(abs(&maxPQ)) %then
    scale = 10/%sysfunc(abs(&maxPQ));
call symput('scale', scale);
%end;
Run;

```

In my case, I am using %let scale = 0.001.

```

%if %sysfunc(abs(&minPQ))<1 and %sysfunc(abs(&maxPQ))<1 %then %do;
  maxPQ=%SYSEVALF(&maxPQ/&scale, ceil);
  minPQ=%SYSEVALF(&minPQ/&scale,ceil);
%end;
%else %do;
  maxPQ=%sysfunc(round(&maxPQ,1));
  minPQ=%sysfunc(round(&minPQ,1));
%end;

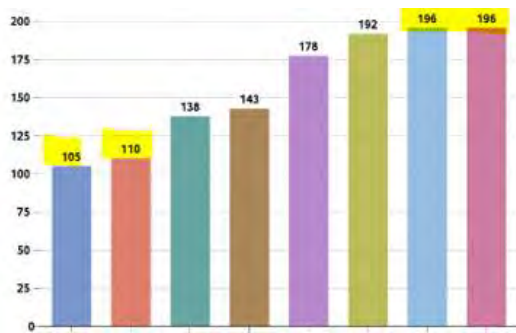
%if %sysfunc(abs(&maxPQ))<1 and %sysfunc(abs(&maxPQ))<1 %then %do;
  max_digit=%SYSEVALF((10**(&lengthmax-1))*&scale);
  min_digit=%SYSEVALF((10**(&lengthmin-1))*&scale);
%end;
%else %do;

```

```
max_digit=%eval(10**(&lengthmax-1));
min_digit=%eval(10**(&lengthmin-1));
%end;
```

2), we need to keep 0 in the y-tick values.

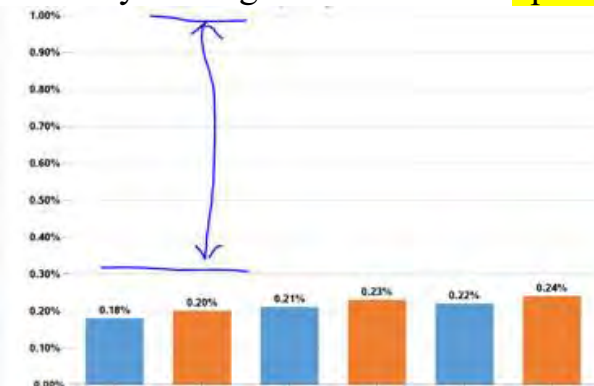
If min is positive, then reset the min as 0, if max is negative, then reset max as 0.



```
/*if min is positive, then reset min as 0, else set as itself*/
/*if max is negative, then reset max as 0, else set as itself*/
%if %index("&min value",-)=0 %then %do; %let startvalue=0; %end;
%if %index("&min value",-)>0 %then %do; %let startvalue=&min value; %end;
%if %index("&max value",-)>0 %then %do; %let endvalue=0; %end;
%if %index("&max value",-)=0 %then %do; %let endvalue=&max_value; %end;
```

3), why do we need rescaling the y-tick values?

Please take a look at the following image, then you will understand why. There are too many blank grids between the up bound 1.00% and the max values 0.24%.



4), we can control the numbers of y-tick values through changing the red numbers in the highlighted part.

Better try to change the red numbers below, and see what happens, you may understand better.

```
***remove those blank grids, then recalculate***
```



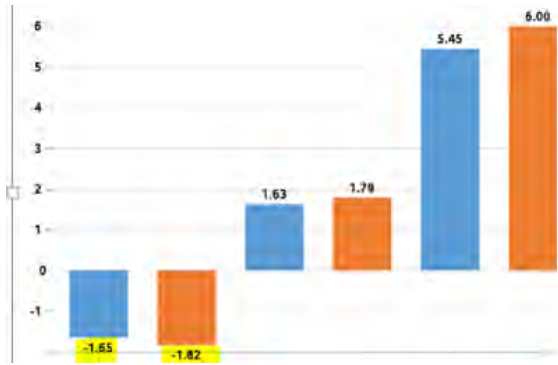
```

**** the incremental values, tick count number**/
data grp_comp.line4_UpLw_Ytickval_&page_&postn
(keep=smallest biggest lower_band upper_band reTickdigitNumber
reTick_Number reTickcount reTickcount_number reincremental_value
redivider multiplier incremental_value reincrementvalues);
retain smallest biggest lower_band upper_band reTickdigitNumber
reTick_Number reTickcount reTickcount_number reincremental_value
redivider multiplier incremental_value reincrementvalues;
set grp_comp.line3_ytickvalue_&page_&postn_&graph_number;
if biggest<0 then do;
lower_band=tickvalues_&rescaling_lower_number;
upper_band=0;
end;
else if smallest>0 then do;
upper_band=tickvalues_&rescaling_upper_number;
lower_band=0;
end;
else do;
upper_band=tickvalues_&rescaling_upper_number;
lower_band=tickvalues_&rescaling_lower_number;
end;
if 12<(upper_band-lower_band)/incremental_value then redivider=.5;
if 6<(upper_band-lower_band)/incremental_value<13 then redivider=1;
if 3<(upper_band-lower_band)/incremental_value<7 then redivider=2;
if 1<(upper_band-lower_band)/incremental_value<4 then redivider=4;
if 0=<(upper_band-lower_band)/incremental_value<2 then redivider=10;
reincremental_value=incremental_value/redivider;
reTickcount=(upper_band-lower_band)/reincremental_value;
reTickdigitNumber=ceil(reTickcount);
if reTickdigitNumber<15 then multiplier=1;
if 14<reTickdigitNumber<28 then multiplier=2;
if 27<reTickdigitNumber then multiplier=3;
reincrementvalues=reincremental_value*multiplier;
reTickcount_number=(upper_band-lower_band)/reincrementvalues;
reTick_Number=ceil(reTickcount_number);
call symput('reincremental',strip(reincrementvalues));
call symput('reTickcount',strip(reTick_Number));
run;

```

5), for the bar chart image, why do we need to minus one increment value?

Because the negative bar chart would have its data label at the bottom of the bar as highlighted below, its values would overlay the x-axis and its tick values under x-axis. Please see the graphic image below.



```

%if %index("&min_value",-)>0 %then %do;          /*For bar chart, if min value is negative*/
  ytickvalues_0=lower_band-&reincremental;      /***we always add one grid blank**/
  %do z=0 %to &reTickcount+1;                  /*it is minus one increment value*/
  ytickvalues_&z=ytickvalues_0+%SYSEVALF(&reincremental*&z);
  %end;
%end;

%else %do;
  ytickvalues_0=lower_band;
  %do z=0 %to &reTickcount;
  ytickvalues_&z=ytickvalues_0+%SYSEVALF(&reincremental*&z);
  %end;
%end;

```

6), The SAS system 0 may not be 0, it may be like 0.00000000000030.

Under this situation we always need to set it as 0, that is why SAS system 0 values may not be 0.

/ remove system round values because sometimes, 0 is not 0*/*

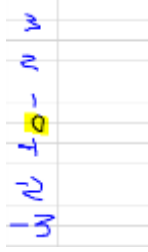
```

data grp_comp.line6_check0_&page._&postn;
set grp_comp.line6_check0_&page._&postn;
if 0.00000000001>COL1>-0.00000000001 then COL1=0;
run;

```

7), 0 may not be in the list of COL1 or ytickvalues.

We can use [referenceline statement curvelabel](#) to insert it, but the inserted [referenceline](#) 0 and y-tick [tickvaluelist](#) may cause unequal distances because of the inserted 0. Please see the highlighted yellow 0, it was inserted into the list.



To solve it, the best way is that if 0 is not in the list of y-tickvaluelist, then we use 0 as center of these values. If above 0, we use 0 plus reincrementvalues; if below 0, we use 0 minus reincrementvalues. We use ytickvalues **_low** and ytickvalues **_up** to distinguish them.

How to check whether we have 0 or not, we need to **separated by 'a a'**, then check `%index("a&check_zero.a",a0a)=0`.

```
proc sql;
  select distinct COL1
  into :check_zero separated by 'a a'
  from grp_comp.line6 check0 &page._&postn;
quit;
  %put &check_zero;

/*check whether ytickvalues include 0*/
/*if no 0, then add 0 into y-tick values*/
%if %index("a&check_zero.a",a0a) = 0 %then %do;
  data grp_comp.line6_ytickval1_&page._&postn;
  set grp_comp.line5_ytickval0_&page._&postn. &graph_number;
  low_ticknum = (0-ytickvalues_0)/reincrementvalues;
  low_ticknumber = ceil(low_ticknum);
  up_ticknum = (0+ytickvalues_%eval(&reTickcount+1))/reincrementvalues;
  up_ticknumber = ceil(up_ticknum);
  call symput('low_ticknumber',low_ticknumber);
  call symput('up_ticknumber',up_ticknumber);
  drop ytickvalues_;;
run;

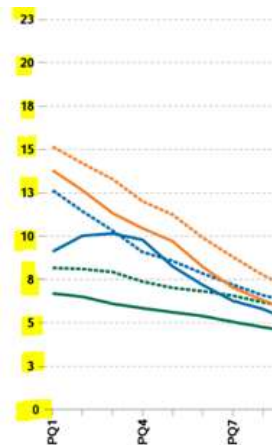
data grp_comp.line7_ytickval_&page._&postn;
retain smallest biggest lower_band upper_band reTickdigitNumber
reTick Number reTickcount reTickcount number reincremental value
redivider multiplier incremental_value reincrementvalues low_ticknumber
up_ticknumber ytickvalues_;;
set grp_comp.line6_ytickval1_&page._&postn;
  %do f = 0 %to &low_ticknumber;
  ytickvalues_ low &f = -%SYSEVALF(&reincremental*&f);
  %end;
  %do k = 0 %to &up_ticknumber;
  ytickvalues_ up &k = %SYSEVALF(&reincremental*&k);
  %end;
call symput("origin_&page._&postn",ytickvalues_low_%left(&low_ticknumber.));
```

```
run;
%end;
```

8), the trick of y-tick tickvalueformat.

For our case, we use two kind of numbers: **regular number** like 18.3386 and **percentage** like 12.3456%, that is why we use two different format values as **form=compress("percentn9."||len1);** and **form=compress("9."||len1);**

We better use **percentn9.**, not **percentn8.**; if we use **percentn8.**, because some numbers are very huge, we may not cover them. Sometimes this situation is very difficult to notice.



The y-tick values on the above image have format 9.0, but clearly, the format should be 9.1. The format 9.0 caused round up to digit number, 2.5 round up to 3, 7.5 round up to 8, it may confuse the readers here.

```
/******create LINE_Y_tickvaluelist and their format and *****/
/*automatically pick tickvalueformat based on each y-tick value list */
proc transpose data=grp_comp.line7_ytickval_&page. &postn
    out =grp_comp.line8_ytickval_&page. &postn;
var ytickvalues_ ;
run;

data grp_comp.line9_fmt_&page. &postn;
set grp_comp.line8_ytickval_&page. &postn;
if 0.0000000001>COL1>-0.0000000001 then COL1=0;
%if %sysfunc(abs(&minPQ))<1 and %sysfunc(abs(&maxPQ))<1 %then %do;
    len=lengthn(scan(COL1,2,"."));
    if len ge 2 then lenth=len-2;
    else lenth=len;
    len1=put(lenth,1.);
    form=compress("percentn9."||len1);
%end;
%else %do;
    len=lengthn(scan(coll,2,"."));
```



```

len1=put(len,1.);
form=compress("9."||len1);
%end;
run;

proc sort data=grp_comp.line9_fmt_&page._&postn;
  by descending len;
run;

data _null ;
set grp_comp.line9_fmt_&page._&postn;
  if _n_=1;
  call symput("tickvalueformat_&page._&postn",form);
run;

```

After this calculation, the `tickvalueformat` will be updated correspondingly with the y-tick values case by case. We would use the `tickvalueformat` value for the `barlabelformat`, then the y-tick values and barlabel values can be under same format. Other situations like the following, we may use the following examples [c]:

```

proc format ;
  picture million (round)
  low-<0 = '09.99M' (prefix="- $" mult=.00001)
  0-high = '09.99M' (prefix="$" mult=.00001);
run;
%let tickvalueformat = million.;

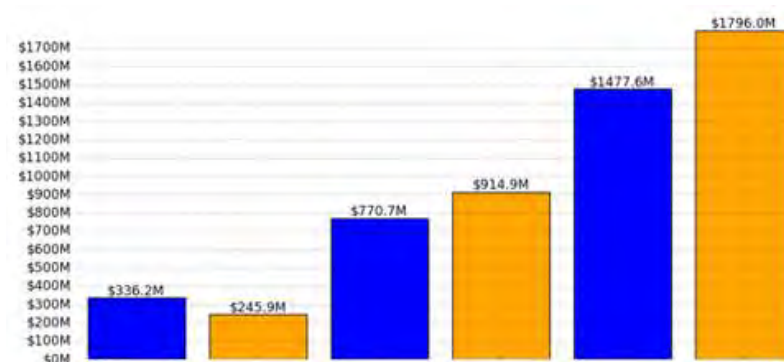
```

OR

```

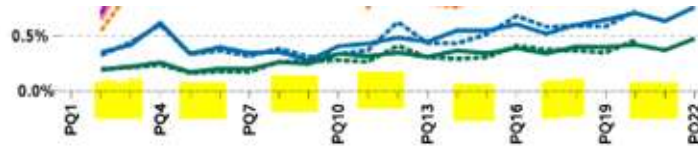
proc format;
  picture bigmoney (fuzz=0)
  1E06-<1000000000='0000.99 M' (prefix='$' mult=.000001)
  1E09-<1000000000000='0000.99 B' (prefix='$' mult=1E-09)
  1E12-<1000000000000000='0000.99 T' (prefix='$' mult=1E-012);
run;
%let tickvalueformat = bigmoney.;

```



9), how can we pick x-axis tickdisplaylist values to avoid crowded or overlaid.

If we list all x-axis tick values under x-axis bar, it would look very crowd, and may cause overlay each other because there are too many tick values. Sometimes, we need only to pick one values for every 3 or 4 of them ... under this situation, `mod` function is a good choice. Same story for y-axis tick values too.



```

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")

```

```

/**calculate xaxis tickdisplaylist and define it as macro parameter**/
data grp_comp.line12_xtickdisplaylist_&page(keep=xtickdisplay);
set grp_comp.Slide_&page._Line(keep=xaxis) nobs=nobs end=last;
  ID= N ;
  xtickdisplay="||strip(xaxis)||";
  do i=1 to nobs-1;
    if mod(ID,3) ne 1 then do; /*automatically pick one xtickvalue from every 3*/
      xtickdisplay="||"; /*to avoid overlay each other */
    end;
  end;
run;

proc sql;
  select strip(xtickdisplay)
  into :xtickdisplaylist_&page. _&postn separated by ' '
  from grp_comp.line12_xtickdisplaylist_&page;
quit;

```

Conclusion

When you assembly line plot graphics using **PROC TEMPLATE** and using *How to Assembly Line Create Graphic Images Using PROC TEMPLATE In SAS Enterprise Guide? (Part I)* and *The Calculator of the Graphic y and x-axis Tickvaluelist, Tickvalueformat with Expected Values for SAS PROC TEMPLATE (Part II)* to calculate the y and x tickvaluelist or tickvaluedisplaylist, it would be very helpful for you to read this paper to understand its logic to avoid many tricks which may cost you too much time to find and fix them.

REFERENCES

[a] Kaiqing Fan

How to Assembly Line Create Graphic Images Using PROC TEMPLATE In SAS Enterprise Guide? (Part I)

[b] Kaiqing Fan

The Calculator of the Graphic y and x-axis Tickvaluelist, Tickvalueformat with Perfect Values for SAS PROC TEMPLATE (Part II)

[c] **PROC FORMAT** and its examples

<http://documentation.sas.com/?docsetId=proc&docsetVersion=9.4&docsetTarget=p0n990vq8gxc a6n1vnsracr6jp2c.htm&locale=en>

<http://documentation.sas.com/?docsetId=proc&docsetTarget=n0k19qj532rbqln187us4ao371h7.htm&docsetVersion=9.4&locale=en#n0k19qj532rbqln187us4ao371h7>

CONTACT INFORMATION

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

Kaiqing Fan

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/>

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.