**The Auto-Calculator of the Graphic y-axis Tickvaluelist, Tickvalueformat with Expected Values for SAS Data Visualizations Part II**

Kaiqing Fan

Mastech Digital Inc.

**Abstract**

When we plot graphics using SAS PROC TEMPLATE or PROC GPLOT or other procedures, we need to decide the x-axis and y-axis tick-value list. However when we plot hundreds or thousands line, bar graphics using SAS, how could we automatically calculate the y-axis tick-values is a huge challenge. My sub-engine can automatically calculate and provide nice y-axis tick-values for sets of minimum and maximum values. Once you decide the minimum and maximum values among a set of numbers, you can use this sub-engine to automatically calculate your x-axis and y-axis tick-values including yaxisopts tickvaluelist, yaxisopts tickvalueformat, barchart statement barlabelformat, referenceline curvelabel, xaxisopts tickdisplaylist and xaxisopts tickvaluelist, and more when you use **PROC TEMPLATE, PROC GPLOT** or other procedures.

**COMMENTS**: I finished three SAS Papers about data visualizations.

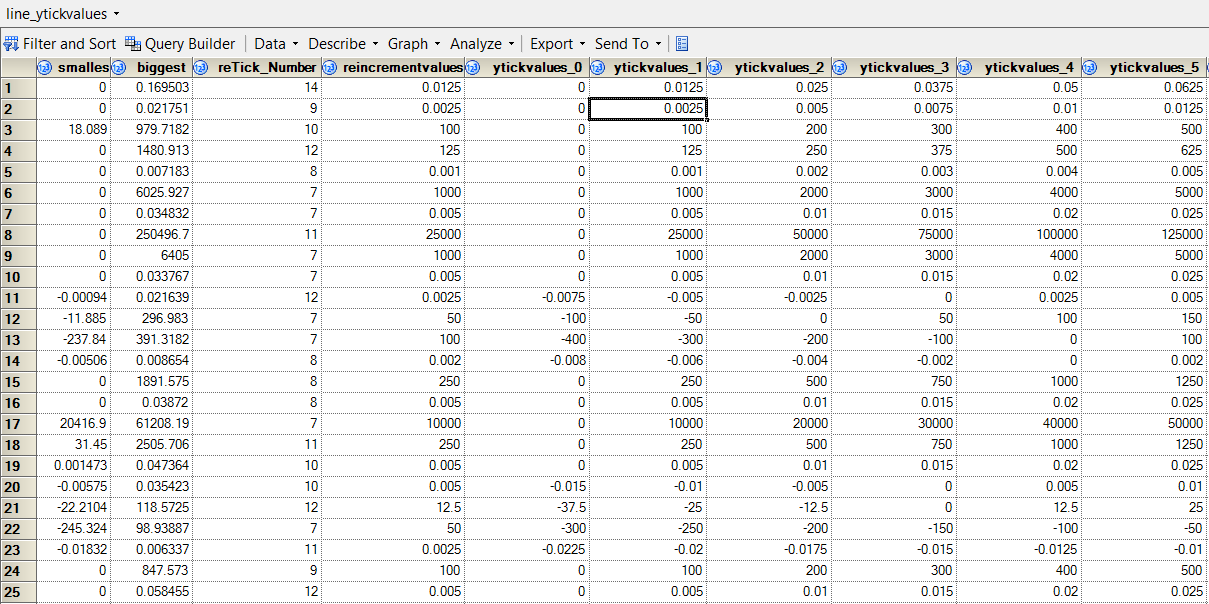
But only the 3rd one was accepted by SAS Global Forum 2018 as e-poster. Acutally, the 3rd one was based on the 1st and 2nd papers. This is the 2nd paper.

Here is the one accepted by SASGF2018. Its paper number is 2545-2018

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

**Introduction**

When we use SAS PROC TEMPLATE or other procedures to draw bars, lines and other graphics, we need to manually decide their y-axis tick-values. But when we draw hundreds of thousands graphics, it is impossible to manually calculate these y-axis tick-values, we need to develop an automatic calculator of the graphic y-axis tick-values. How nice would these y-axis tick-values look like? The following is the list of examples:



**The Features of the Calculator**

Here are the features:

1, the increment values are 1, 1.25, 2, 2.5, 5, 10 … and their … 0.0001, 0.001, 0.01, 1, 10, 100, 1000 … times; it is easy to count with mind only.

2, the number of grids are between 6 and 15, we can make this number smaller if need;

3, the gaps between maximum value and the upper-band are no more than 2 or 3 times of increment values;

4, the gaps between minimum value and the low-band are no more than 2 or 3 times of increment values except that the low\_band is 0;

5, the y-axis tick-values are always including 0 value;

6, the biggest and smallest values are always covered by the up and low bands;

7, This calculator will automatically calculate and define the parameters of yaxisopts origin, yaxisopts viewmin, yaxisopts tickvaluelist, yaxisopts tickvalueformat, barchart statement barlabelformat, referenceline curvelabel, xaxisopts tickdisplaylist and xaxisopts tickvaluelist.

In the calculator engine, we call them as

%GLOBAL ytickvaluelist\_&page.\_&postn

ytickdisplaylist\_&page.\_&postn (you can add it into the engine by yourself)

xtickvaluelist\_&page.\_&postn

xtickdisplaylist\_&page.\_&postn

curvelabel\_&page.\_&postn

tickvalueformat\_&page.\_&postn

origin\_&page.\_&postn

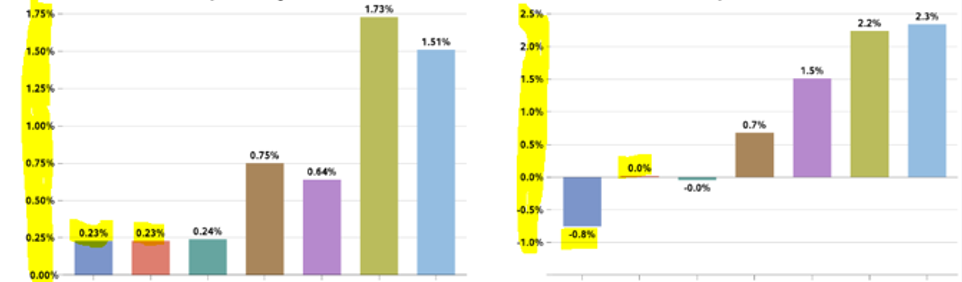
xtickvaluelist\_&page.\_&postn

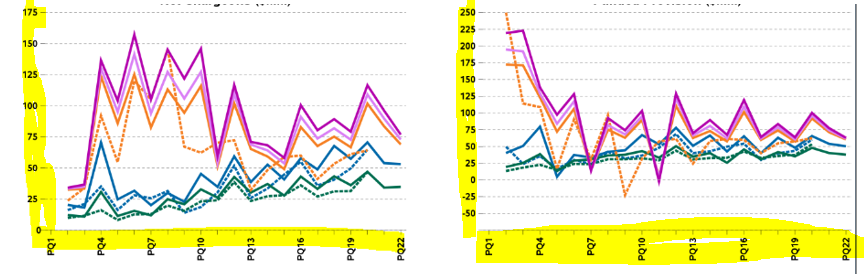
xtickdisplaylist\_&page.\_&postn ;

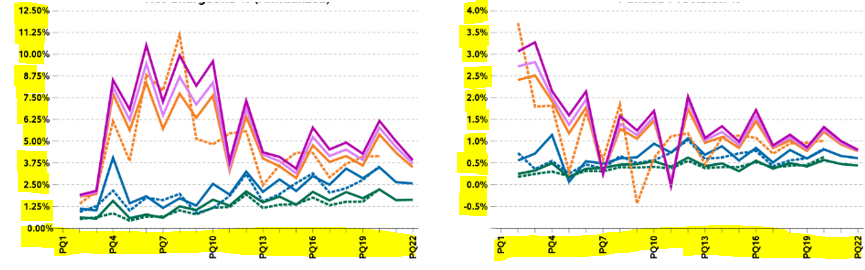
8, the tickvalueformat\_&page.\_&postn will make all values in the image with same format, please see the first graphics.

Attached EXCEL file **6\_line\_ytickvalues.xlsx** contains more than 210 examples of minimum and maximum values as biggest and smallest in the left two columns, and their y-axis tick-values. If interested, please feel free to review them by yourself.

These features can guarantee the y-axis tick-values look very nice like the following two graphic images: one is bar chart, another is line chart.





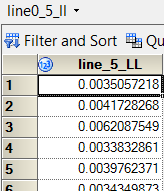


Why would we like to calculate y-axis tick-values like the above? I think we should make the above y-tick-values as easy and friendly as possible to be read by users, leaders, decision-makers.

**The SAS y-axis Tick Values Auto-Calculator Engine**

To make the engine as clear as possible, I add some needed comments as the explanation of its logic here. It is an engine calculator of 1 page with 4 images.

The data example is line0\_5\_ll table with the following data structure:



%letpage = 5;

**%macro** yaxis\_tickvalue\_calculator(postn=);

%GLOBAL ytickvaluelist\_&page.\_&postn ytickdisplaylist\_&page.\_&postn

curvelabel\_&page.\_&postn tickvalueformat\_&page.\_&postn origin\_&page.\_&postn ;

/\*pick the min and max values from data line0\_&page.\_&postn\*/

proc sql;

select distinct MIN(line\_&page.\_&postn), MAX(line\_&page.\_&postn)

into :minPQ, :maxPQ from line0\_&page.\_&postn;

quit;

/\*\*\*\*\*\*\*calculate the lengths of min and max before decimal point\*\*\*\*/

/\*if both their absolute values are less than 1, times with 1000 or 100\*/

data line1\_&page.\_&postn;

length maxPQ $50. minPQ $50.;

%if %sysfunc(abs(&minPQ))<**1** and %sysfunc(abs(&maxPQ))<**1** %then %do;

%let scale = 0.0000001;

maxPQ=%SYSEVALF(&maxPQ/&scale,ceil); minPQ=%SYSEVALF(&minPQ/&scale,ceil);

%end;

%else %do;

maxPQ=%sysfunc(round(&maxPQ,**1**)); minPQ=%sysfunc(round(&minPQ,**1**));

%end;

length\_max = length(strip(tranwrd(maxPQ,'-',''))); length\_min = length(strip(tranwrd(minPQ,'-','')));

call symput('lengthmax',length\_max); call symput('lengthmin',length\_min);

run;

/\*decide the incremental values, tick count number\*/

data line2\_&page.\_&postn; /\*decide max and min value’s digit level as 1, 10, 100, 1000, 10000 …\*/

%if %sysfunc(abs(&minPQ))<**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;

if max\_digit >= min\_digit then digit\_level = max\_digit;

if max\_digit < min\_digit then digit\_level = min\_digit;

/\*decide the min as upper band\*/ minPQ=&minPQ/digit\_level;

if minPQ <**0** then round\_min=floor(minPQ); if minPQ>=**0** then round\_min=ceil(minPQ);

minmmPQ=round\_min\*digit\_level; if minmmPQ<**0** then minimumPQ=minmmPQ;

if minmmPQ>=**0** then minimumPQ=**0**; call symput('min\_value',minimumPQ);

/\*decide the max as upper band\*/ maxPQ=&maxPQ/digit\_level;

if maxPQ <**0** then round\_max=floor(maxPQ); if maxPQ>=**0** then round\_max=ceil(maxPQ);

maxmmPQ=round\_max\*digit\_level; if maxmmPQ<**0** then maximumPQ=**0**;

if maxmmPQ>=**0** then maximumPQ=maxmmPQ; call symput('max\_value',maximumPQ);

/\*decide the incremental values\*/

%if %sysfunc(abs(&minPQ))<**1** and %sysfunc(abs(&maxPQ))<**1** %then %do;

if &lengthmax>=&lengthmin then incremental=%SYSEVALF((**10**\*\*(&lengthmax-**1**))\***0.01**);

if &lengthmax<&lengthmin then incremental=%SYSEVALF((**10**\*\*(&lengthmin-**1**))\***0.01**); %end;

%else %do; if &lengthmax>=&lengthmin then incremental=%eval(**10**\*\*(&lengthmax-**1**));

if &lengthmax<&lengthmin then incremental=%eval(**10**\*\*(&lengthmin-**1**)); %end;

/\*control the tick count numbers, if it is too large, divided by dividers\*/

if **12**<(maximumPQ-minimumPQ)/incremental then divider=**.5**;

if **6**<(maximumPQ-minimumPQ)/incremental<**13** then divider=**1**;

if **3**<(maximumPQ-minimumPQ)/incremental<**7** then divider=**2**;

if **1** < (maximumPQ-minimumPQ)/incremental <**4** then divider=**4**;

if **0**=< (maximumPQ-minimumPQ)/incremental <**2** then divider=**10**;

/\*recalculate the incremental values\*/

incremental\_value=incremental/divider; call symput('incremental',incremental\_value);

/\*decide the tick count number\*/ Tickcount=(maximumPQ-minimumPQ)/incremental\_value;

TickdigitNumber=ceil(Tickcount); call symput('Tickcount',TickdigitNumber);

run;

/\*if min is positive, then reset it as 0, or as itself, if max is negative, then reset it as 0, or 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;

/\*rescaling y-tickvalues because there may be lots of blank grids\*/

/\*\*\*\* above max value to top or below min value to bottom \*\*\*\*/

data line3\_ytickvalue\_&page.\_&postn;

retain smallest biggest upper lower upper\_blank\_grids

lower\_blank\_grids rescaling\_upper\_number rescaling\_lower\_number

incremental\_value tickvalues\_:;

set line2\_&page.\_&postn;

smallest=&minPQ; biggest=&maxPQ; tickvalues\_0=&startvalue;

%do z=**1** %to &Tickcount; tickvalues\_&z=tickvalues\_0+%SYSEVALF(&incremental\*&z); %end;

upper = tickvalues\_%sysfunc(left(&Tickcount));

upper\_blank=(upper-biggest)/incremental\_value;

upper\_blank\_grids=int(upper\_blank);

lower = tickvalues\_0;

lower\_blank=(smallest-lower)/incremental\_value;

lower\_blank\_grids=int(lower\_blank);

rescaling\_upper\_number=TickdigitNumber-upper\_blank\_grids;

rescaling\_lower\_number=lower\_blank\_grids;

call symput('rescaling\_upper\_number',strip(rescaling\_upper\_number));

call symput('rescaling\_lower\_number',strip(rescaling\_lower\_number));

run;

/\*remove those blank grids, then recalculate, recalculate the incremental values, tick count number\*/

data 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 line3\_ytickvalue\_&page.\_&postn;

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;

/\*use the reincrementvalues and reTick\_Number to calculate each y-tick value\*/

data line5\_ytickval0\_&page.\_&postn;

retain smallest biggest lower\_band upper\_band reTickdigitNumber reTick\_Number reTickcount reTickcount\_number reincremental\_value redivider multiplier incremental\_value reincrementvalues ytickvalues\_:;

set line4\_UpLw\_Ytickval\_&page.\_&postn;

%if %index("&min\_value",-)>**0** %then %do; /\*\*\* 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;

run;

proc transpose data=line5\_ytickval0\_&page.\_&postn out =line6\_check0\_&page.\_&postn;

var ytickvalues\_:;

run;

/\* there may be some very small values, but show as 0 \*/

data line6\_check0\_&page.\_&postn;

set line6\_check0\_&page.\_&postn;

if **0.00000000001**>COL1> -**0.00000000001** then COL1=**0**;

run;

proc sort data=line6\_check0\_&page.\_&postn; by COL1; run;

proc sql; select distinct COL1 into :check\_zero separated by 'a a' from 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 line6\_ytickval1\_&page.\_&postn;

set line5\_ytickval0\_&page.\_&postn;

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 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 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;

%else %do;

data 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 line5\_ytickval0\_&page.\_&postn;

call symput("origin\_&page.\_&postn",ytickvalues\_0);

run;

%end;

/\*proc append create the 6\_line\_ytickvalues table \*/

/\*proc append base=line\_ytickvalues\_append data=line7\_ytickval\_&page.\_&postn force; run;\*/

/\*create tickvaluelist, their format, automatically pick tickvalueformat based on each y-tick value list\*/

proc transpose data=line7\_ytickval\_&page.\_&postn out =line8\_ytickval\_&page.\_&postn;

var ytickvalues\_:;

run;

data line9\_fmt\_&page.\_&postn;

set line8\_ytickval\_&page.\_&postn;

if **0.00000000001**>COL1>-**0.00000000001** 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(col1,**2**,"."));

len1=put(len,**1.**);

form=compress("9."||len1);

%end;

run;

proc sort data=line9\_fmt\_&page.\_&postn; by descending len; run;

data \_null\_;

set line9\_fmt\_&page.\_&postn;

if \_n\_=**1**;

call symput("tickvalueformat\_&page.\_&postn",form);

run;

/\*\*create curvelabel value with its format\*\*/

data line10\_yticklist\_&page.\_&postn;

set line9\_fmt\_&page.\_&postn;

if **0.00000000001**>COL1> -**0.00000000001** then COL1=**0**;

format ytickvalues &&tickvalueformat\_&page.\_&postn;

ytickvalues=COL1;

run;

proc sql; select distinct max(COL1), min(COL1) into :maxCOL1, :minCOL1

from line10\_yticklist\_&page.\_&postn;

quit;

/\*if its max or min is 0, then define curvelabel as 0 with 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 line10\_yticklist\_&page.\_&postn where COL1 in (**0**);

quit;

%end;

%else %do; %let curvelabel\_&page.\_&postn=; %end;

/\*define yaxis tickvaluelist as macro parameter; if need, define ytickdisplaylist as macro parameter\*/

proc sql; select distinct COL1 into :ytickvaluelist\_&page.\_&postn separated by ' '

from line10\_yticklist\_&page.\_&postn;

quit;

%PUT &&curvelabel\_&page.\_&postn; %PUT &&ytickvaluelist\_&page.\_&postn;

%PUT &&tickvalueformat\_&page.\_&postn; %PUT &lower\_band\_&page.\_&postn;

**%mend** yaxis\_tickvalue\_calculator;

%if %index("&Slide\_Positn",LL)>**0** %then %do; %***yaxis\_tickvalue\_calculator***(postn=LL); %end;

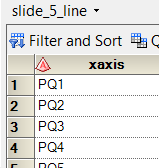
%if %index("&Slide\_Positn",LR)>**0** %then %do; %***yaxis\_tickvalue\_calculator***(postn=LR); %end;

%if %index("&Slide\_Positn",UL)>**0** %then %do; %***yaxis\_tickvalue\_calculator***(postn=UL); %end;

%if %index("&Slide\_Positn",UR)>**0** %then %do; %***yaxis\_tickvalue\_calculator***(postn=UR); %end;

**The SAS x-axis Tick Values Calculator Engine**

The table example of Slide\_5\_Line(keep=xaxis), in the following table, we can have any number of PQs, not only 22, the **macro xaxis\_tickvalue\_calculator** will automatically pick them from the list.



**%macro** xaxis\_tickvalue\_calculator;

%GLOBAL xtickvaluelist\_&page.\_&postn xtickdisplaylist\_&page.\_&postn ;

/\*\*define xaxis tickvaluelist as macro parameter\*\*/

data line11\_xticklist\_&page;

set Slide\_&page.\_Line(keep=xaxis);

xticklist='"'||strip(xaxis)||'"';

run;

proc sql; select strip(xticklist) into :xtickvaluelist\_&page.\_&postn separated by ' '

from line11\_xticklist\_&page;

quit;

/\*\*define xaxis tickdisplaylist as macro parameter\*\*/

data line12\_xtickdisplaylist\_&page(keep=xtickdisplay);

set 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='""';

end;

end;

if last then xtickdisplay='"'||strip(xaxis)||'"';

run;

proc sql; select strip(xtickdisplay) into :xtickdisplaylist\_&page.\_&postn separated by ' '

from line12\_xtickdisplaylist\_&page;

quit;

%PUT &&xtickdisplaylist\_&page.\_&postn; %PUT &&xtickvaluelist\_&page.\_&postn;

**%mend** xaxis\_tickvalue\_calculator;

%if %index("&Slide\_Positn",LL)>**0** %then %do; %***xaxis\_tickvalue\_calculator***(postn=LL); %end;

%if %index("&Slide\_Positn",LR)>**0** %then %do; %***xaxis\_tickvalue\_calculator***(postn=LR); %end;

%if %index("&Slide\_Positn",UL)>**0** %then %do; %***xaxis\_tickvalue\_calculator***(postn=UL); %end;

%if %index("&Slide\_Positn",UR)>**0** %then %do; %***xaxis\_tickvalue\_calculator***(postn=UR); %end;

**Contact Information:**

The SAS engine is kind of complex, if anybody is interested in the logic, please feel free to contact with me, I would like to present its logic to you. I will be happy to provide necessary assistantship to you.

**Kaiqing Fan**

Sr. Data Scientist

Sr. SAS Tech Lead

Sr. SAS Developer

Sr. Quantitative Risk Researcher

Mastech Digital Inc.

Address: Brecksville, OH-44141

(Mobile) 504-344-7267

Personal email: [fankaiqinguw@gmail.com](mailto:fankaiqinguw@gmail.com)