PharmaSUG China 2019 - Paper DV-021

Automation of Axes Scale Setting in Figures using SAS Graph Template Language (GTL)

Xiang Liu, Pfizer China Research and Development Co., Ltd

ABSTRACT

Currently, there're basically two methods of setting axis scales in the generation of figures using SAS Graph Template Language (GTL); One is to set every feature (minimum and maximum tick value, interval of major ticks as well as minor tick numbers) manually, causing extra workload of deciding and QCing every plot, and not applicable in multiple-page figures. The other method is to not set any feature and let the GTL assign axis scales automatically. With this method the workload is lower, but the plot may not fit some specific requirements, like force display of tick 0, or unified scales in different pages.

A macro is developed to automatically assign the axis scales of figures using SAS GTL. It can be used in the generation of different plot types with continuous axes (linear or log-scale) to assign proper scale for every axis based on actual data range. For plot with multiple pages, parameter can be easily set to either make axis scales identical for all pages, or properly assign for every single page. User's preference of axis, like the tick density, major tick interval, force display of tick 0 and axis offset preference can be adjusted by optional parameters, and the rest will be done automatically by the code. It is easily to use by just introducing the macro prior to PROC SGRENDER, and only two parameters are necessary at minimum.

INTRODUCTION

During plots generation in clinical trials, there are two common methods to scale axes: one is manual setting of all parameters; the other depends on GTL itself to judge axis scale. The first part of this paper discusses both method with their advantages and disadvantages.

To automatically set axis an algorithm is developed to perform a two-direction searching for appropriate major tick interval of axis based on raw data range, then accordingly calculate and set the rest scale parameters of axis.

Based on this algorithm, a macro %autorange is introduced covering the syntax, usage as well as its two mandatory and several optional parameters.

ABBREVIATION

GTL – Graph Template Language

NOTATION

V_{min} – Minimum value of the variable in source data

V_{max} – Minimum value of the variable in source data

Step – Scanned step length of major tick interval

$Step_p$ – Preferred major tick interval

Start – Calculated start tick value in axis

Stop – Calculated end tick value in axis

CURRENT METHOD TO SCALING AXES

One of the major challenges of clinical data visualization is to set appropriate scales for plot axes including axis range, major tick interval, as well as minor tick numbers between two major ticks. Generally, we have two methods to assign these parameter for a certain axis in Graphic Template Language (GTL) plots.

MANUAL SCALING

Programmers can manual scale the axis by setting every parameter in statement LINEAROPTS= of PROC TEMPLATE.

```
proc template;
    …
    yaxisopts= (… linearopts= (viewmin=0 viewmax=1000
                   tickvaluesequence= (start=0 increment=100 end=1000)
                   minorticks=true minortickcount=1) …)
    …
```

This statement defines a linear Y-axis from 0 to 1000, with major tick interval as 100, and one minor tick between two major ticks.

This method scales the axis precisely, so programmers can set the appropriate axis scale one by one manually. However, the workload is heavy especially when large number of plots are produced, and the data changes from time to time which makes back and forth reviews and revisions of axis scale. Moreover, if the axis scale for every page is different, this method is not applicable since the assigned axis scale is fixed among pages.

AUTOMATIC SCALING BY GTL

This method requires no parameter in LINEAROPTS= statement, and let GTL automatically decides the axis scales.

```
proc template;
    yaxisopts= (… minorticks=true …)
    …
```

This statement does not assign any parameter about Y-axis scale and GTL would automatically set them based on data range. Apparently, the workload decreases, but the lack of parameter assignment leads to little control of axis, and programmers can do little when the axis by GTL is inappropriate (e.g. the major tick number is too few or too many), or there's special request for the axis. (e.g. The axis scale must keep identical across pages, or tick "0" must be included in PK concentration plots.)

This chart gives a summary of advantages and disadvantages of these two methods.

Method	Advantages	Disadvantages
Assign parameters manually	• Axes are appropriate after choosing best parameters based on data manually. • Axes can precisely fit any special request in one-page plot.	• Workload is heavy especially for large number of figures with updating data. • Does not applicable for multiple page with axis varying among pages.
Let GTL manage axes scale	• Axes scale updates automatically when data is change. • Axes scale fits every page in plots with multiple pages. • Work load is relatively low.	• Can't fit special requests of axes. • Revision is difficult if automatically generated axes are inappropriate.

Table 1. Comparison of Two Methods of Axes Management in GTL

To solve this challenge once and for all, an algorithm is introduced to keep the advantages of both methods that automatically setting axes based on data status when giving users control of axes for specialized requests.

ALGORITHM OF AUTOMATIC AXIS ASSIGNMENT

To give an appropriate axis scale solution, the major tick interval is the key value. With certain data range, different major tick interval leads to different major tick numbers on the axis. So, this algorithm scans a

series of major tick intervals, calculates the corresponding tick numbers, then chooses the appropriate tick numbers hence major tick interval.

Since the minimum and maximum value (V_{min} and V_{max} respecrtively), hence ranges of data values (*DataRange*) in clinical trials varies a lot, we chose a two-direct searching from major tick interval (*Step*)= 1. One direction is 1, 2, 5, 10, 20, 50, … and the other is 1, 0.5, 0.2, 0.1, 0.05, 0.02, …

The tick number in the axis is calculated as *round(DataRange, step)* (plus 1 if rounded value is less than *DataRange*) for every search step.

The criterion of "preferred" major tick interval is that, the calculated total tick number in the axis is between 5-12. After numerous practice, the density of ticks is appropriate for a common plot if we choose this range of preferred tick numbers. Moreover, since the maximum ratio for search step length are 2.5 (50 to 20), this 5-12 range (i.e. 4-11 intervals that the ticks dividing the axis length) makes sure that at least one searching result locates in it.

The criterions of ending search are also based on the total tick number that 1) the calculated tick number is >12 and greater than previous calculated tick number, or <5 and lesser than or equal to previous one. These conditions indicate that the searching result has been out, and getting away from the preferred range, so the searching in this direction can stop.

When the searching or two directions are both completed, there must be 1 or 2 of the preferred major tick intervals kept. A parameter is kept in the macro to let user keep the solution with greater or lesser tick number in the axis, i.e. the final preferred major tick interval is decided.

With the preferred major tick interval ($step_p$), the start and stop tick value (*start* and *stop*) can be easily defined as $Start = round(V_{min}, step_p)$ (minus $step_d$ if rounded result is greater than V_{min}), $Stop = round(V_{max}, step_p)$ (plus $step_p$ if rounded result is lesser than V_{max}).

The minor tick number (if user choose to display) is decided also based on the major tick interval. If major tick interval is a multiple of 5 (e.g. 0.05, 0.5, 500), the minor tick number is 4; If it is a multiple of 2 (e.g. 0.02, 2, 2000), the minor tick number is set to 1; If it is a multiple of 10 (e.g. 0.001, 1, 100), the minor tick number is set to 3 (if total major tick number is < 8) or 1 (if total major tick number is >= 8).

Please refer to this schema for a practical example of the algorithm.

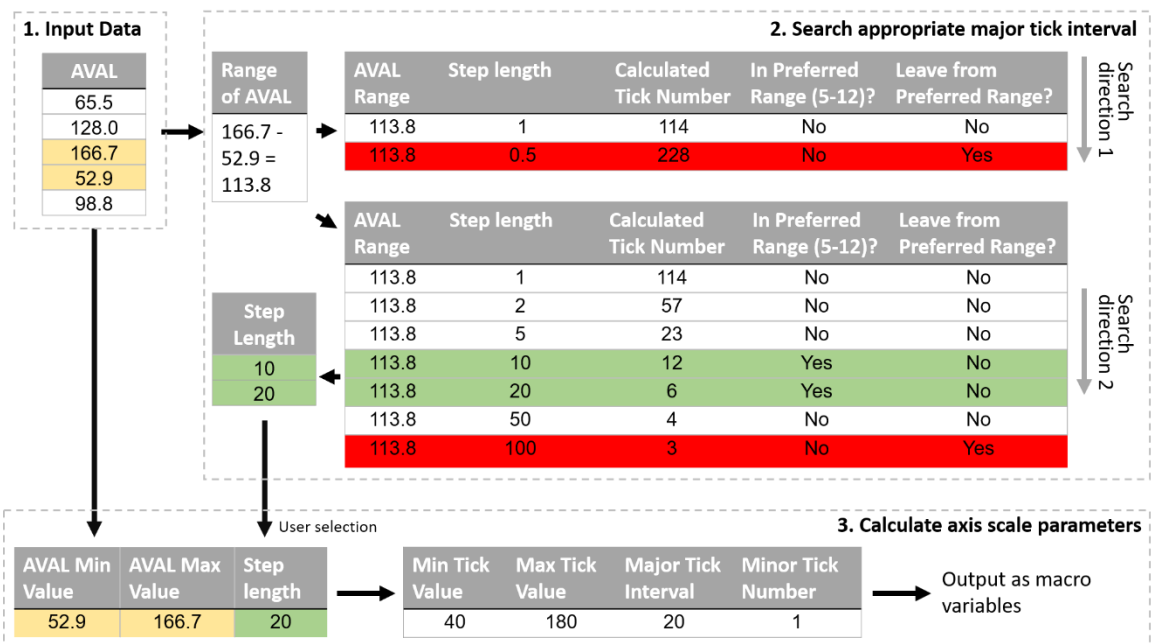

Figure 1. Schema of the Two-direction Search Algorithm

INTRODUCTION OF THE MACRO BASED ON THE ALGORITHM

Based on the algorithm described above, a macro is developed for a general way of automatically scaling axes in clinical studies.

The basic macro usage and syntax are like below:

```
…
data <dataset-name>;
/*the final reporting data set name for PROC SGRENDER*/
    ……
run;
…
proc template;
    define statgraph temp01;
    dynamic … ystart ystop ystep yminor ydisp …;
    …
    yaxisopts=(… linearopts=(viewmin=ystart viewmax=ystop
                 tickvaluesequence=(start=ystart increment=ystep end=ystop)
                 minorticks=ydisp minortickcount=yminor …);
/*Five dynamic variables are used for transit the output macro variables,
the name can be revised by user*/
    …
run;
…
%autorange(dataset_in=<dataset-name>, vars=<variable-names>,
/*These two parameters are mandatory*/
         page_subset=,  step_search_method=,
         tick_density=, out_macrovar_suffix=,
         axis_offset=,  minor_tick_yn=,
         tick_integer=
/*These parameters are optional*/
    );

proc sgrender data=<dataset-name> template=temp01;
    dynamic … ystart=&start. ystop=&stop. ystep=&step.
              yminor=&minor.  ydisp=&minordisplay.;
/*The macro variable names are fixed, or adding suffix by user definition*/
…
```

THE USAGE OF %AUTORANGE

Macro %autorange can be invoked everywhere after the final reporting data set was ready. It is recommended to place it just prior to PROC SGRENDER– due to the reason in the following.

There are two mandatory parameters of %autorange, dataset_in and vars, which represent the final reporting data set name and variable names used for setting axis respectively. Other parameters are optional.

Macro %autorange outputs five macro variables, &start, &stop, &step, &minor and &minordisplay, they represent minimum and maximum of tick values, major tick interval, number of minor tick between two major ticks and yes/no option to display minor tick respectively. These macro variables can be evoked directly in AXISOPTS= statement in PROC TEMPLATE or assigned into five dynamic variables in PROC SGRENDER (recommended).

Please note that there's no assumption or pre/post-process requirement of data set structure, data values or other aspect before using %autorange. Except for five dynamic variable introductions, there's little modification needed for user's original plotting code. Moreover, %autorange just outputs macro variables rather than change anything of user's data set. If user doesn't want to use %autorange any more, he doesn't need to modify any code, and just follows the traditional ways to assign axis parameters to undo

the change. These properties make it a good choice as a “plug-in” of current plotting code to raise the efficiency and reduce the workload.

PARAMETERS OF %AUTORANGE

Dataset_in

This parameter defines the input data set name.

e.g. `dataset_in = mylib.report`

Vars

This parameter defines the variable names contributing to the axis scaling. If multiple variables should be considered in the axis scaling (e.g. error bar defined by YERRORLOWER= and YERRORUPPER=), all the variable names need to be written here separated by a comma, hence an %STR() macro function should be used.

Especially, constant can be also put into VARS parameter and it also enters the calculation of data range. It is useful when a “0” tick should be added into the axis while the data range does not reach 0.

e.g.
```
vars = AVAL
vars = %STR(yvalue, yupper, ylower)
vars = %STR(pkconc, 0)
```

Page_subset

This parameter is used when the figure has multiple page, and the axis should fit for every single page rather than remain identical among all pages. In this scenario, a %DO loop needs to be used to subset reporting data set, and page_subset must have an identical value with the subset statement.

Here’s an example code where a multi-page plot need to be generated by PARAMN.

```
proc sort data=report; by PARAMN; run;
/*REPORT is the reporting data set*/

proc sql noprint;
/*collect all distinct PARAMN together and save them in a macro variable
ALLPARAMN*/
      Select distinct PARAMN into: allparamn separate by ‘$’
      From report;
quit; /*ALLPARAMN is like 1$2$3$...*/

proc template;
dynamic … ystart ystop ystep yminor ydisp;
…
run;

%let i=1;
%do %while(%length(%scan(&allparamn., &i., $)));
/* a %do loop is executed until all the PARAMN are read*/

%let currparamn = %scan(&allparamn., &i.,$);
/*CURRPARAMN = 1, 2, 3, … for every loop*/

%autorange( dataset_in=report, vars=AVAL,
            page_subset=%STR(paramn = &currparamn.), …);
/*Execute the subset of data set by different PARAMN and calculate
appropriate axis scale respectively*/
```

```
proc sgrender data=report;
      where paramn = &currparamn.;
/*please note that the WHERE clause is identical with the value of
page_subset in %autorange*/
      dynamic … ystart=&start. ystop=&stop. ystep=&step.
                yminor=&minor.  ydisp=&minordisplay.;
/*dynamic variable is used to transit different output values from %autorange
in every loop*/
…
      run;
%let i = %eval(&i.+1);
%end;
```

With this code, the data set REPORT is subset by different PARAMN in every loop, and %autorange calculates appropriate axis scale for every subset part of the data in the loop.

Please note that to bring the function into practice, the %autorange evocation and PROC SGRENDER must be in the same %DO loop; and dynamic variables must be used to transit different values in every loop.

Correspondingly, if user need a unified axis scale among pages, just leave page_subset as missing, and %autorange will calculate the axis scale without any subset of raw data set.

This parameter has missing value as default.

Step_search_method

This parameter defines the step search method. Currently there're two methods (step length sequences that %autorange will use) presented as below.

Step_search_method	Step Length Sequences
1	…, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, …
2	…, 0.005, 0.01, 0.025, 0.05, 0.1, 0.25, 0.5, 1, 2.5, 5, 10, 25, 50, …

Table 2. Values of step_search_method

With step_search_method set to 1 or 2, the possible tick interval is different in plot. User can set either 1 or 2 to this parameter and the default value is 1.

Tick_density

This parameter defines user's preference of tick number in the axis when more than one appropriate major tick intervals are calculated.

Min and Max Value of Raw Data	Calculated Solution	Axis Scale for Each Solution	Tick Numbers in This Solution
475 - 1680	1	400 to 1800 by 200	8
	2	0 to 2000 by 500	5

Table 3. An Example of Tick_density

In this example, data range is 475 – 1680, so %autorange can calculate two solutions for it, one is (400, 600, 800, …, 1800), the other is (0, 500, 1000, 1500, 2000), and there would be 8 and 5 major ticks on axis respectively. When user set tick_density to H, solution (1) would be applied; otherwise solution (2) would be applied in the final plot.

This parameter can be set as H or L representing to high / low tick density on axis respectively, and the default value is H.

Out_macrovar_suffix

This parameter is applied for multiple usages of %autorange when there're more than one axis to be automatically scaled. The parameter defines a suffix of output macro variables so that they can be transit to different dynamic variables to apply in different axis definition statements. e.g.

```
data report;
      /*XVAR and YVAR represent to var used in X- and Y- axis respectively*/
run;
…
proc template;
      define statgraph temp01;
      dynamic … ystart ystop ystep yminor ydisp
                xstart xstop xstep xminor xdiso …;
      /*Two sets of dynamic variables are for X and Y axis*/
      …
      xaxisopts=(… linearopts=(viewmin=xstart viewmax=xstop
                   tickvaluesequence=(start=xstart increment=xstep end=xstop)
                   minorticks=xdisp minortickcount=xminor …);
      yaxisopts=(… linearopts=(viewmin=ystart viewmax=ystop
                   tickvaluesequence=(start=ystart increment=ystep end=ystop)
                   minorticks=ydisp minortickcount=yminor …);
…
run;
…
%autorange(dataset_in=report, vars=xvar, out_macrovar_suffix=1);
%autorange(dataset_in=report, vars=yvar, out_macrovar_suffix=2);

proc sgrender data=<dataset-name> template=temp01;
      dynamic … xstart=&start1. xstop=&stop1. xstep=&step1.
                xminor=&minor1. xdisp=&minordisplay1.
                ystart=&start2. ystop=&stop2. ystep=&step2.
                yminor=&minor2. ydisp=&minordisplay2.;
      …
      /*Different out_macrovar_suffix gives different suffix of output macro
vars (1 and 2) and transit to different dynamic variables*/
```

With this code X and Y axis can be both automatically scaled with their own variable.

This parameter can be valued as any combination of alphabet character and number, the value is applied as the suffix of the five output macro variables.

Default value of this parameter is empty, in this scenario the output macro variable names are START, STOP, STEP, MINOR and MINORDISPLAY rescpectively.

Axis_offset

This parameter enables user to set an offset to axis when calculate the data range to prevent the scenario that symbols are too near to the edge of the plot area.

If set to Y, the data range will be increased by 5% before the major tick searching;

Min and Max Value of Raw Data	Value of *axis_offset*	Data Range for Calculation	Calculated Solution of Axis Scale
500 - 2500	N	500 - 2500	500 to 2500 by 500
	Y	400 - 2600	0 to 3000 by 500

Table 4. An Example for the Function of Axis_offset

If axis_offset is set as Y, %autorange will largen the data range by 100 (= 5% * (2500-500)) in both min and max value of raw data range, so the data range in the step length search is 400 - 2600 not 500 - 2500 as raw data. Only one exception in this process is: if the raw data values are all positive (negative) and largen data range is < (>)0; in this scenario the data range is set to 0 to prevent misleading.

Min and Max Value of Raw Data	Value of *axis_offset*	Data Range for Calculation	Calculated Solution of Axis Scale
0 - 2500	N	0 - 2500	0 to 2500 by 500
	Y	0 (not -125) - 2625	0 to 3000 by 500

Table 5. Exception of Axis_offset

Minor_tick_yn

This parameter controls the display of minor tick. If set to Y, minor tick number is automatically assigned by %autorange; if set to N, minor tick is not displayed in the axis. Set as Y by default.

Tick_integer

This parameter is set to prevent such a scenario that all values in reporting data set are integer (e.g. individual plots of scores of questionnaire) and the ticks referring decimal numbers in the axis will be inappropriate and misleading.

Setting tick_integer=Y will force the macro removing the minor tick when calculated major tick interval is 1 or set only one minor tick between major ticks when major tick interval is 2. If calculated interval is less than 1, then the interval will be reset to 1, causing lesser major ticks than usual.

If calculated interval is greater than 2, the minor ticks will remain the same whether tick_integer is Y or N.

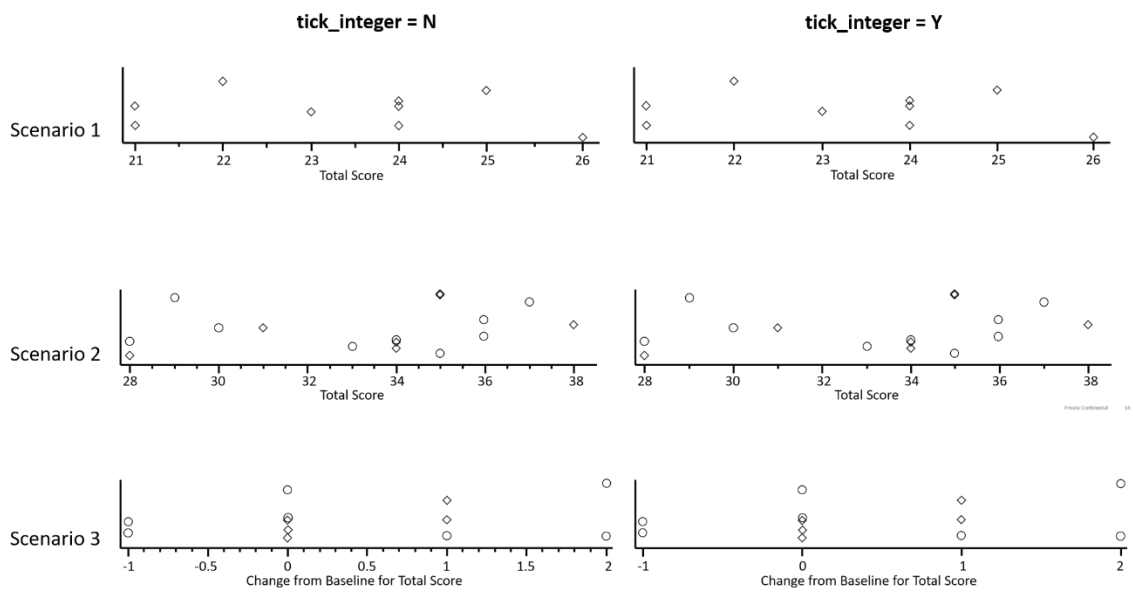

Figure 2. Function of Parameter Tick_integer

CONCLUSION

The algorithm as well as the corresponding macro %autorange can automatically assign axis scale with little modification of main code, input data structure and values, and multiple optional parameters of the macro may help users control the axis scales to fit the specialized requests in different studies and plot types.

This macro also has some points to upgrade, for example the decimal numbers of tick value is not yet controlled except for integer / non-integer. Moreover, this macro is only for linear axis scale not for log scale, or categorical scale. These points can be fixed in future updates of this macro, making it a general tool in plot generation of clinical studies.

REFERENCES

[1] KuenHung Lin (2018), "Efficient Graphing of Basic Data Structure – Let Data Decide Scale of Y-Axis for Themselves" PharmaSUG China 2018 – Paper 46. Available at http://www.pharmasug.org/proceedings/china2018/DV/Pharmasug-China-2018-DV46.pdf

[2] Watts, Perry (2009), "Generate a Customized Axis Scale with Uneven Intervals in SAS® Automatically" Proceedings of the SAS Global Forum 2009, paper 192-2009. Available at http://support.sas.com/resources/papers/proceedings09/192-2009.pdf

RECOMMENDED READING

- *Graph Template Language (GTL) in SAS® Help Document*

CONTACT INFORMATION

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

Name: Xiang Liu
Enterprise: Pfizer China Research and Development Co., Ltd
Address: 9F, Tower B14, Biolake, 666 Gaoxin Avenue,
Eastlake Hi-Tech Development Zone,
Wuhan 430205, P.R China
E-mail: Xiang.Liu@pfizer.com
Mobile: 1387-157-5518