

Systematically Reordering Axis Major Tick Values in SAS Graph

Brian Shen, PPDi, ShangHai

ABSTRACT

Once generating SAS graphs, it is a headache to programmers to reorder the axis tick values parameter by parameter, especially for generating big amount of graphs for different parameters or for data changing on daily basis, for example, marketing data. SAS default ordering is sometimes not desirable for procedures like GPLOT, GCHART, etc. Although some great enhancements on new 9.3 STAT graph package and the Graph Template Language [GTL], the old GCHART, GPLOT procedures are still widely applied by SAS programmers. This paper presents a method on how to systematically replace the default SAS graph ordering for better graph output to simplify programming activities.

INTRODUCTION

Default axis ordering mechanism for SAS procedures like GPLOT, GCHART often generate undesirable axis tick value list. For example, the major tick numbers are too many [Figure 1] or the tick values looks really strange [Figure 2]. It is usually a tedious work for programmer to revise the axis tick ordering, especially when you generate many graphs through macro or generate graph on changing data. Even for fixed data, since the axis ordering is decided by both graph size and tick value font size, changing one will sometime result in change on both graph ordering and potentially the plot itself. It might take a lot of effort to recover to original format for such undesirable changes. This paper presents a new axis ordering mechanism through a SAS macro. Generally, a minimum of four to eight major ticks will be generated based on data value range by using step of 1's (... , 0.01, 0.1, 1, 10, 100,...), 2's or 5's. Enhancements on the axis ordering are illustrated through examples from SAS procedures like GPLOT and SAS 9.3 Graph Template Language [GTL].

COMMON COMPLAINTS FROM SAS USER ON GRAPH GENERATIONS

The Axis ordering issues are common complaints to SAS programmer during graph generation using old procedures like GCHART, GPLOT, etc. The most common issues for default axis ordering are listed as following:

- Too many major tick numbers for axis ordering and make the axis looks crowded (Figure 1).
- Major tick value list looks strange (Figure 2).
- If data has only one data point and option "`label = none`" is applied, graph shrinks into a single line (Figure 3).
- If data is all missing, warning is given and no graph is generated.
- If font size or graph size change, change on axis ordering or plot itself might also be changed.
- If the scale for negative data and positive data are not the same, for example, negative part is too small, and positive part is too big, only non-negative or non-positive order is generated. This might be misleading to reviewers that all data are non-negative or non-positive (Figure 4).

This paper proposes mathematical logics to redefine more desirable axis ordering to simplify programming activities. Enhancements are illustrated with examples in Figure 1 to Figure 4. Procedure [GCHART or GTL] used for graph generation. Methods applied [GTL or GCHART], together with the Min and Max for order calculation are annotated on graph. Except for Figure 3, which is generated using SAS default for single data point with axis option `label = none`, all other plots have two overlay plots with left vertical axis on SAS default ordering and right vertical axis on user defined ordering by using %ORDER macro.

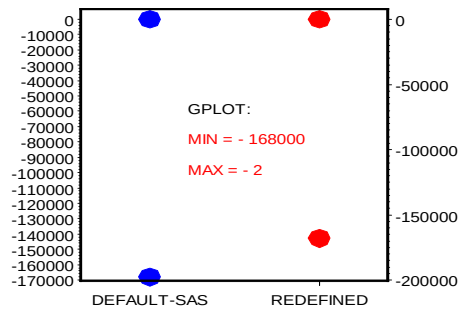


Figure 1. Too Many Major Ticks

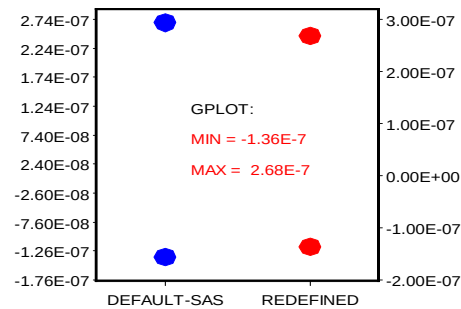


Figure 2. Tick Values are Strange

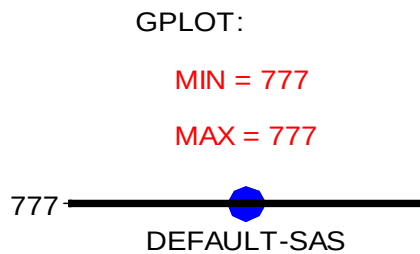


Figure 3. Plot Shrinks to a Line

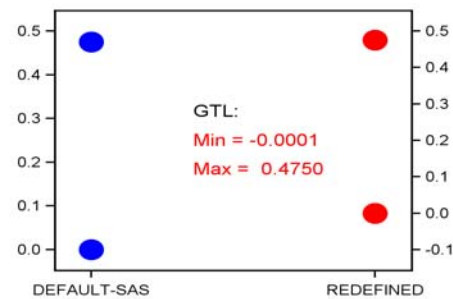


Figure 4. Only non-negative order produced for data with negative values

MATHEMATICAL LOGICS TO RECALCULATE AXIS ORDERING

There are 7 major logics to redefine axis ordering, as illustrated below. For more details, please refer to macro code attached. A minimum of 4 major ticks and maximum of 8 major ticks will be obtained by applying %ORDER. User can easily revise the macro to make it suitable for your own needs, for example, if you want Major Tick 5-8, you can add another statement like this after step 7: %if %sysevalf((&high - &low)/&step = 3) %then %let step = %sysevalf(&step/2); etc.

- 1) Calculate the data *RANGE* either from the dataset variable or by user defined MIN or MAX.

Min = - 301, Max = 401, Range = 702;

Min = -0.001, Max = 0.0001, Range = 0.0011;

Special case for range equal to 0:

 - a) If Min = Max = 0, reset Min = - 0.5, Max = 0.5;
 - b) If Min = Max and not equal to 0, extend Min/Max to half of their absolute value on both direction. For example, if Min=Max=50, Reset Min = 25, Max =75.
- 2) Divide range by 10 to get interim result, called *RESULT1*, if *RESULT1* > 1, round it to single digit.

702/10 = 70.2, *RESULT1* = 70 after round;

0.0011/10 = 0.00011, *RESULT1* = 0.00011;
- 3) Divide *RESULT1* by a *DIVIDER* to make the new result, called *RESULT2*, in a range between 0 to 10, and *DIVIDER* must be exponents with base 10, like 0.01, 0.1, 1, 10, 100, etc.

70 / 10 = 7 → *RESULT2* = 7, *DIVIDER* = 10;

$0.00011 / 0.0001 = 1.1 \rightarrow RESULT2 = 1.1, DIVIDER = 0.0001;$

- 4) Get the initial step, called *STEP1*, based on the following logics:

If $. < RESULT2 \leq 2.5$, $STEP1 = 2.5$; $\rightarrow (RESULT2 = 1.1, STEP1 = 2.5)$

If $2.5 < RESULT2 < 6.2$, $STEP1 = 5$;

If $6.2 < RESULT2 < 10$, $STEP1 = 10$; $\rightarrow (RESULT2 = 7, STEP1 = 10)$

- 5) Get the *STEP* for axis order by multiplying *STEP1* by *DIVIDER* by 2, i.e., $STEP = STEP1 * DIVIDER * 2$.

$STEP = 10 * 10 * 2 = 200$;

$STEP = 2.5 * 0.0001 * 2 = 0.0005$

- 6) Divide Min by STEP and Floor to integer, then multiply STEP calculated, which will be the LOW value for Axis Ordering; Divide Max by STEP and Ceil to integer, then multiply STEP calculated, which will be the HIGH value for Axis Ordering.

$LOW = \text{Floor}(-301/200) * 200 = -400$, $High = \text{ceil}(401/200) * 200 = 600$; $ORDER = -400$ to 600 by 200 ;

$LOW = \text{Floor}(-0.001/0.0005) * 0.0005 = -0.001$, $High = \text{ceil}(0.0001/0.0005) * 0.0005 = 0.0005$; $ORDER = -0.0010$ to 0.0005 by 0.0005 .

- 7) If both Min and Max are non-negative or non-positive, the LOW and HIGH must both be non-negative or non-positive. If Min and Max are in different signs, LOW and HIGH must also be in different signs. For example, if Min = -0.002, Max= 20000, the LOW is set to -5000, instead of 0 as GPLOT/GTL does.

HOW TO APPLY THE MACRO WITH YOUR GRAPH PROGRAMMING

There are six parameters, DSN (Dataset Name) VAR (Numerical Variable Name), MIN (Minimum value), MAX (Maximum Value), ORDER_NAME (Global Macro Variable Name for ordering), VALUelist_NAME (Global Macro Variable Name for ordering in value list format). The macro is allowed to calculate axis order either from data values in a datasets, or user specified Min or Max or both. If either Min or Max, or none is specified, the unspecified one(s) will be calculated from Data values, if both Dataset Name and Variable Name are also specified. For more details, please refer to macro.

Sample call:

```
%order(  
    Dsn          = sashelp.cars,  
    var          = mpg_city,  
    max          = 50.7 ,  
    order_name   = y_order,  
    valuelist_name = y_tickvaluelist)
```

This call will use Min of MPG_CITY from dataset SASHELP.CARS and max = 50.7 to calculate axis order. Global macro name Y_ORDER is generated, which can be called for procedures like GPLOT, GCHART, GMAP, SGPanel, SGPlot, etc. If VALUelist_NAME is specified, Global macro variable specified in VALUelist_NAME will be generated plus global variables named VIEWMIN, VIEWMAX, which are specially designed for GTL language.

- Y_ORDER: =10 to 60 by 10
- Y_TICKVALUELIST: 10 20 30 40 50 60
- VIEWMIN: 10
- VIEWMAX: 60

You can call the Macro variable(s) generated into your graph programming to replace SAS default.

For example for GCHART:

Axis1 Label = none order = (&y_order.) offset=(0, 3)pct;

For Graph Template Language:

yaxisopts=(linearopts =(viewmin = &viewmin viewmax=&viewmax tickvaluelist= (&y_tickvaluelist.))));

COMPARE AXIS ORDER AMONG %ORDER, GCHART/ GTL DEFAULT.

The following table lists axis ordering from default SAS GCHAT/GTL and from %ORDER. In most cases, %ORDER is consistent with GTL default ordering. Except for ALL-MISSING data or data with just one single data point, %ORDER will generate more desirable axis ordering results than GPLOT and GTL. GTL axis ordering mechanisms have significant improvement compared with old SAS graphic packages. Major differences between the three mechanisms are for the following case:

- Plot with Single data point [All are acceptable, if label = none specified, GHART will generate strange results]
- Data with all missing data [GTL is the best choice. Revision needed for %ORDER to get same results as GTL]
- Min and Max are in different sign and the scale are significantly different [%ORDER is better than other two]
- For all other cases, %ORDER and default GTL mechanisms are better than GCHART, GPLOT, etc.

Table 1 Comparison of Axis ordering by Default GPLOT, Default GTL and %ORDER

Min Max	GPLOT DEFAULT (Tick Numbers)	GTL DEFAULT (Tick Numbers)	%ORDER (Tick Numbers)
Missing Missing	WARNING: ALL VALUES MISSING or out of range. No plot will be Generated	WARNING: Y=Y is invalid. The option expects at least one non-missing value in the column. Graph is Generated with empty AXIS	WARNING: ALL VALUES MISSING or out of range. Graph will be Generated with AXIS = 999 to 9999 by 9000. Can make revision to be consistent with GTL
0 0	0 to 0 by 0 (1) <i>plot shrinks into a line (Figure 4) if axis option Label = none is specified]</i>	0 to 0 by 0 (1)	-1 to 1 by 0.5 (5)
1 4	1 to 4 by 1 (4)	1 to 4 by 0.5 (7)	1 to 4 by 1 (4)
1.1 6	1 to 6 by 1 (6)	1 to 6 by 1 (6)	1 to 6 by 1 (6)
100 635	100 to 700 by 100 (7)	100 to 700 by 100 (7)	100 to 700 by 100 (7)
777 777	777 to 777 by 0 (1),	777 to 777 by 0 (1)	200 to 1200 by 200 (6)
-0.002 20000	0 to 30000 by 10000 (4)	0 to 20000 by 5000 (5)	-5000 to 20000 by 5000 (6)
-1.36E-7 2.68E-7	-1.76E-7 to 2.74E-7 by 0.5E-7 (10)	-2E-7 to 3E-7 by 1E-7 (6)	-2E-7 to 3E-7 by 1E-7 (6)
-168000 -2	-170000 to 0 by 10000 (18)	-200000 to 0 by 50000 (5)	-200000 to 0 by 50000 (5)

CONCLUSION

This macro has been significantly enhanced SAS axis ordering, especially for procedures like GPLOT, GCHART. It will significantly save programming efforts on graph generation, especially when you generating a huge number of plots or generating plots on changing data. It provided a care –free ordering mechanism for better plot output. Users can easily revise the macro to satisfy their own needs. Further enhancement on missing data, and for plot with only one data point can be made by revising the macro from user end.

REFERENCES

SAS Institute Inc.(2012), SAS®9.3 Online Document, Cary, NC, SAS Institute Inc.

ACKNOWLEDGMENTS

CONTACT INFORMATION

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

Name: Brian Shen
Enterprise: PPD
Address:
City, State ZIP:
Work Phone: (86)21-53834000

E-mail: brian.shen@ppdi.com

Web: www.ppdi.com

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.

APPENDIX: MACRO %ORDER FOR AXIS ORDER GENERATION

```
%macro order(  
    DSN          = , /* Dataset Name */  
    VAR          = , /* Numerical Variable Name */  
    MIN          = , /* User Defined Minimum value */  
    MAX          = , /* User Defined Maximum value */  
    ORDER_NAME   = , /* Macro Variable Name for Ordering */  
    VALUelist_NAME = ); /* Macro Variable Name for Ordering for GTL */
```

/*Sample Calls:

/** -This call will Generate Four macro variables: X_ORDER, X_VALUelist, VIEWMIN, VIEWMAX, on Min value of MPG_CITY and Max Value of 50.7;

```
%order(dsn          = sashelp.cars,  
    var          = mpg_city,  
    max          = 50.777 ,  
    order_name   = y_order,  
    valuelist_name = y_tickvaluelist);
```

/** -This call will Generate one macro variable MY_ORDER, on Min value of -0.1 and Max Value of 50.7;

```
%order(min          = -0.1,  
    max          = 50.7,  
    order_name   = y_order); /*
```

```
%local temp_min temp_max temp_max1 max1 step result1 result2 divider digitposition low high vnum type message;
```

/** -Assign Macro Variable Name for Ordering to Global;

```
    %if %length(&order_name) %then %global &order_name;  
    %if %length(&valuelist_name) %then %global &valuelist_name viewmin viewmax;
```

```
    %let message = 0;
```

/** - CHECK if Dataset Name is Right;

```
    %if %length(&dsn) > 0 and %sysfunc(exist(&dsn.)) = 0 %then %do;  
        %put %sysfunc(compress(ER ROR:)) Dataset &DSN does NOT exist;  
        %let message = %sysfunc(max(&message., 1));  
    %end;
```

/** - CHECK if VAR is specified when DSN is specified;

```
    %if %length(&dsn) > 0 and %length(&var) = 0 %then %do;  
        %put %sysfunc(compress(ER ROR:)) VAR must be specified once DSN is specified;  
        %let message = %sysfunc(max(&message., 1));  
    %end;
```

/** - CHECK if either DSN+VAR or MIN+MAX is specified;

```
    %if %length(&dsn)=0 and (%length(&min.) = 0 and %length(&max.) = 0) %then %do;  
        %put %sysfunc(compress(ER ROR:)) Either MIN or MAX must be specified once DSN is NOT;  
        %let message = %sysfunc(max(&message., 1));  
    %end;
```

/** - CHECK if global ordering macro variable is specified;

```
    %if %length(&order_name)=0 and %length(&valuelist_name) = 0 %then %do;  
        %put %sysfunc(compress(ER ROR:)) ORDER_NAME, or VALUelist_NAME or BOTH must be specified.;  
        %let message = %sysfunc(max(&message., 1));  
    %end;
```

/** - CHECK if Numeric Variable specified by VAR exist;

```
    %if %length(&dsn)>0 and %sysfunc(exist(&dsn.)) > 0 %then %do;  
        %let dsid      = %sysfunc(open(&dsn, i));  
        %let vnum      = %sysfunc(varnum(&dsid, &var));  
        %let rc        = %sysfunc(close(&dsid));
```

```

%if &vnum = 0 %then %do;
%put %sysfunc(compress(ER ROR:)) Variable %upcase(&var.) does NOT exist;
%let message = %sysfunc(max(&message., 1));
%end;

%if &vnum > 0 %then %do;
%let dsid = %sysfunc(open(&dsn, i));
%let type = %sysfunc(vartype(&dsid, &vnum));
%let rc = %sysfunc(close(&dsid));
%if &type = C %then %do;
%put %sysfunc(compress(ER ROR:)) Variable %upcase(&var.) is NOT numeric;
%let message = %sysfunc(max(&message., 1));
%end;
%end;

%end;

%if &message = 0 %then %do;
%* - Get the MIN and MAX value from DATASET;
%if %length(&dsn)>0 %then %do;
proc sql noprint;
select left(put(min(&var),30.15)), left(put(max(&var), 30.15))
into :temp_min, :temp_max
from &DSN.;
quit;
%end;

%* - Assign DSN-Variable MIN to macro variable MIN, if it is NOT specified;
%if %length(&min) = 0 %then %let min = &temp_min;

%* - Assign DSN-Variable MAX to macro variable MAX, if it is NOT specified;
%if %length(&max) = 0 %then %let max = &temp_max;

%* - Exchange MIN and MAX if MIN > MAX;
%if %sysevalf(&min > &max and &max > .) %then %do;
%let max1 = &max;
%let max = &min;
%let min = &max1;
%end;

%* - Assign non-missing value of MIN or MIX to the missing one;
%if %sysevalf(&min > . and &max = . or %length(&max)=0) %then %let max = &min;
%if %sysevalf(&max > . and &min = . or %length(&min)=0) %then %let min = &max;

%* -Reset the Min and Max Values if they are the same;
%if %sysevalf(&max = &min and &max = 0) %then %do;
%let min = -0.5;
%let max = 0.5;
%end;
%if %sysevalf(&max = &min and &max ne 0 and &max>.) %then %do;
%let min = %sysfunc(min(%sysevalf(&max*0.5), %sysevalf(&max*1.5)));
%let max = %sysfunc(max(%sysevalf(&max*0.5), %sysevalf(&max*1.5)));
%end;

%* -Get axis RANGE and divide RANGE by 10 to get RESULT1;
%if %sysevalf(&max>.) %then %do;
data _null_;
range=%sysevalf(&max-&min);
if range<10 then call symput('result1', strip(reverse(substrn(reverse(strip(put(range/10, 22.20))),
findc(reverse(strip(put(range/10, 22.20))), '123456789'), 30))));
if range>=10 then call symput('result1', reverse(reverse(strip(put(int(range/10), 22.0))));
run;

```

```

%* -Find DIVIDER [Exponent with base 10] for RESULT1 to make RESULT2 in range of 0 - 10;
    %let digitposition = %sysevalf(%sysfunc(indexc(&result1., '123456789'))-1);
    %if %sysevalf(&digitposition > 0) %then %let exponent = %sysevalf(0 -
    %length(%sysfunc(compress(%substr(&result1, 1, &digitposition), ' ')));
    %if %sysevalf(&digitposition = 0) %then %let exponent = %sysevalf(%length(%scan(&result1, 1, .))-1);
    %let divider = %sysevalf(10**&exponent);

%* -Divide RESULT1 by DIVIDER to get RESULT2 in a Range of 0 -10;
    %let result2 = %sysevalf(&result1/&divider);

%* -Get Step in 1s, 2s or 5s (i.e: ..., 0.5, 5, 50, ...);
    %if %sysevalf(&result2 <= 2.5) %then %let step = %sysevalf(2.5*&divider*2);
    %if %sysevalf(&result2>2.5 & &result2<=6.2) %then %let step = %sysevalf(5*&divider*2);
    %if %sysevalf(&result2>6.2 & &result2<=10) %then %let step = %sysevalf(10*&divider*2);

%* -Get the LOW and HIGH for order;
    %let low = %sysevalf(%sysfunc(floor(%sysevalf(&min/&step)))*&step);
    %let high = %sysevalf(%sysfunc(ceil( %sysevalf(&max/&step)))*&step);

%* -Make at most 8 Major Ticks for the Axis;
    %if %sysevalf((&high-&low)/&step > 7) %then %do;
        %let step = %sysevalf(&step*2);
        %let low = %sysevalf(%sysfunc(floor(%sysevalf(&min/&step)))*&step);
        %let high = %sysevalf(%sysfunc(ceil( %sysevalf(&max/&step)))*&step);
    %end;

%* -Reset Low and High if Both Min & Max are in the same Sign;
    %if %sysevalf(&min >= 0 and &max >= 0 and &low < 0) %then %let low = 0;
    %if %sysevalf(&min <= 0 and &max <= 0 and &high > 0) %then %let high = 0;

%* -Make at least 4 Major Ticks and at most 8 Major ticks for the Axis;
    %if %sysevalf((&high-&low)/&step < 3) %then %let step = %sysevalf(&step/2);

%* -Remove Extra Ticks;
    %if %sysevalf(&min -&low > &step) %then %let low = %sysevalf(&low + &step);
    %if %sysevalf(&high-&max > &step) %then %let high = %sysevalf(&high - &step);

%* -Reset Low and High due to ROUND Statement for Extremely Small Numbers;
    %if %sysevalf(&low > &min) %then %let low = %sysevalf(&low - &step);
    %if %sysevalf(&high < &max) %then %let high = %sysevalf(&high + &step);
%end;

%* -Reset Low and High if both MIN | MAX are missing;
%if %sysevalf(&max = .) %then %do;
    %let low = 999;
    %let high = 9999;
    %let step = 9000;
%end;

%* -Define ordering macro Variables;
    %* -Define ORDER_NAME;
    %if %length(&order_name) > 0 %then %do;
        %let &order_name = &low to &high by &step;
        %put;
        %put *****;
        %put ***** %upcase(&order_name) = &&&order_name ;
        %put *****;
        %put;
    %end;

```

```

%* -Define valuelist_name;
%if %length(&valuelist_name) > 0 %then %do;
data temp_xxxxx;
do vlist = &low to &high by &step;
if abs(vlist) < 1.0E-15 then vlist =0;
output;
end;
run;

proc sql noprint;
select vlist
into :&valuelist_name separated by ' '
from temp_xxxxx
order by vlist;
drop table temp_xxxxx;
quit;
%let viewmin = &low;
%let viewmax = &high;

%put;
%put *****.
%put ***** %upcase(&VALUELIST_NAME) = &&&VALUELIST_NAME ;
%put ***** VIEWMIN = &VIEWMIN VIEWMAX=&VIEWMAX ;
%put *****.
%put;
%end;
%mend;

```