

PharmaSUG 2014 – Paper CC32
1 of N Methods to Automate the Y- Axis
Sanjiv Ramalingam, Biogen Idec

ABSTRACT

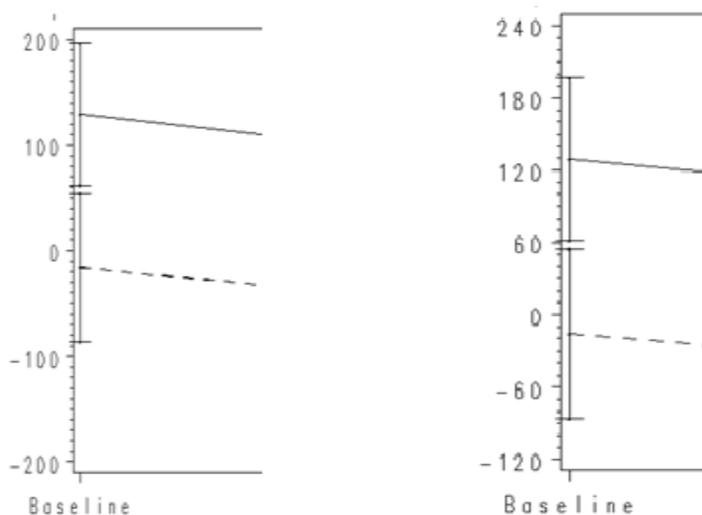
Methodologies for automating the Y-axis has been discussed previously ^[1,2,3]. The motivation to share this code is to not only present readers an another algorithm but because the methodology is simple, robust in that it can accommodate not only integer data but also floating point data and has comparatively very few lines of code. The algorithm allows the user to specify how many divisions they would like their data to be visualized. Even though SAS 9.3 handles Y-axis automation, reasons as to why automation by the user is warranted in certain situations have been elucidated.

INTRODUCTION

Even though SAS by itself does populate and create the Y-axis when the order statement is not specified it will not be suitable if one or a combination of the following conditions are met.

1. If tick marks are annotated in the X-axis, a minimum value must be specified ^[4].
2. Expectation to have the displayed range very close to the actual data range as there is a tendency of SAS to not have ranges close to the actual data range.

This is especially true when plotting error bars and when using interpolation=HILOCTJ. For example, For an actual data range of -112 to 200, SAS plotted with a data range of -200 to 200. This limitation is true only for versions of SAS that do not support SGPLOT. When using the algorithm described, the range chosen was -120 to 240.



a.

b.

Figure 1.1 a. Depicting the Y axis order when the order is not specified and automatically chosen by SAS.
b. Depicting the Y axis order when using the algorithm described.

3. When Treatment un-blinding is involved and there is an un-blinded partner and the only expectation from the un-blinded partner is to re-run programs without changing programs i.e. modifying the Y-axis range.
4. Using the algorithm allows the y-axis label to be dynamically changed to suit multiple page outputs.

1 of N Methods to Automate the Y- Axis, continued

METHODOLOGY

The core methodology involves determining the factor(YBY) that divides the actual range of data into equal divisions. The formula used to determine that factor(YBY) in this algorithm was:

$$YBY = (MAX - MIN) / YINTR$$

Where MAX and MIN are the maximum and minimum values obtained from PROC MEANS. YINTR is the number of divisions the data can be divided into. It is user specified and has been set to 5 as default.

Once YBY has been determined, it is used as a reference to compute the minimum and maximum using DO loops. For determining the minimum(YMIN), YBY is incremented in multiples of YBY until it is greater than MIN-YBY. This is used when MIN is greater than or equal to 0.

```
i=yby;
do until (i gt (min-yby)) ;
    i=i+yby;
end;
```

When MIN is less than 0, YBY is decremented in multiples of YBY until it is less than MIN.

```
i=yby;
do until (i lt min) ;
    i=i-yby;
end;
```

Once YMIN has been determined, it is used as the reference value and incremented in multiples of YBY until that value is greater than MAX to get YMAX.

```
j=ymin;
do until (j gt (max)) ;
    j=j+yby;
end;
```

CONCLUSION

A methodology to automate the Y –axis has been explained and a macro has been included in the appendix section. Since the need for calculating the parameters of the ORDER statement in the AXIS statement has been justified it would be very handy for the user if SAS can generate macro variables for the required parameters.

REFERENCES

- [1] Don (Dongguang) Li ,Tired of Defining Axis Scale for SAS Graphs? A Solution with an Automatic Optimizing Approach, Pharmasug-2009
- [2] Perry Watts, Generate a Customized Axis Scale with Uneven Intervals in SAS® Automatically, SAS Global Forum-2009
- [3] Rick Edwards, Automating SAS/GRAPH Axis Ranges, SESUG 2012
- [4] Sanjiv Ramalingam, Automating the labeling of X-axis, Pharmasug 2013

1 of N Methods to Automate the Y- Axis, continued

ACKNOWLEDGEMENT

The author would like to thank the management at Biogen Idec for their encouragement and support.

CONTACT INFORMATION

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

Contact Name :Sanjiv Ramalingam

Email : sanjiv.ramalingam@biogenidec.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.

1 of N Methods to Automate the Y- Axis, continued

APPENDIX

Program ID: YAXIS.SAS
Description: Compute the minimum and maximum and the step size to be used in defining the AXIS statement. Returns three macro variables namely YMAX, YMIN and YBY that can be used in the AXIS statement, ORDER option (axis1 order=0 to 10 by 2).

Input parameters:

INDS: Input data set

YVAR: variable in the data set to be scaled

YINTR: Number of divisions data needs to be visualized in. Cannot be 0 or negative. Should atleast be 1. Default is 5

ROUNDTO : Precision if required

Sample call: %YAXIS (INDS=ad1b,YVAR=aval,YINTR=3,ROUNDTO=0.01)

```
%macro yaxis(
inds=,
yvar=,
yintr=5,
roundto=
);

%global ymax ymin yby roundto_;

/* determine precision, if precision determined is greater than 6 then it is set to
7*/
%if &roundto= %then %do;

%global roundto_;
data _NULL_;
length m $50;
set &inds end=eof;
retain dec0 0;
if _n_ eq 1 then dec0=0;
%if &yvar^= %then %do;
m=strip(put(&yvar,best.));
%end;
%else %do;m=""; %end;
if index(m, '.') and not missing(m) then maxaft=length(scan(m,-1, '.'));
if missing(maxaft) then maxaft=0;
dec0=max(dec0,maxaft);
if eof then do;
select (dec0);
when (0) n=1;
when (1) n=0.1;
when (2) n=0.01;
when (3) n=0.001;
when (4) n=0.0001;
when (5) n=0.00001;
when (6) n=0.000001;
otherwise n=0.0000001;
end;
end;
```

```

call symputx('roundto_',n);
    output;
end;
run;
%end;

/* determine maximum and minimum from the range of data */
Proc means data=&inds;
    Var &yvar;
    Output out=ymin_ymax min=min max=max ;
run;

/* determine YBY value and then YMIN and YMAX */
data _NULL_;
    set ymin_ymax;
    yintr=&yintr;
    yby=round(((max-min)/yintr),%if &roundto^= %then %do;&roundto %end; %else
%do;&roundto_ %end;);
    if yby ne 0 then do;
        i=yby;
        if min ge 0 then do;
            do until (i gt (min-yby)) ;
                i=i+yby;
            end;
/* For those instances when after the loop above, 'i' maybe greater than the minimum
value of data, it is decremented until it is less than the minimum value */
            if i gt min then do;
                do until (i le min);
                    i=i-yby;
                end;
            end;
            if min lt 0 then do;
                do until (i lt min) ;
                    i=i-yby;
                end;
            end;
            call symputx('ymin',i);
            j=i;
            do until (j gt max) ;
                j=j+yby;
            end;
            call symputx('ymax',j);
            call symputx('yby',yby);
        end;
/* If all data points are equal a warning message is output */
        if yby eq 0 then do;
            call symputx('ymin',min);
            call symputx('ymax',max);
            call symputx('yby',0);
            put 'WARNING: YBY has been assigned 0 as MIN and MAX are EQUAL';
        end;
run;

/* cleanup */
proc datasets library=work;
    delete ymin_max;
quit;
%mend;

```