A Robust Generalized Axis-scaling SAS® Macro

Dorothy E. Pugh, Independent Consultant

ABSTRACT

Generating high-quality plots quickly is an important task for the clinical trials programmer. Creating reasonable-looking axes for these plots presents a special challenge because of the mutually conflicting needs to 1) make the number of tick marks moderate and consistent, 2) reduce the number of significant figures in the tick mark labels, and 3) control the amount of margin space in the resulting plot. This paper describes an axis-scaling algorithm, which enables programmers with different needs to achieve a tradeoff among these criteria that suits these needs. Users can control settings with two types of input variables in the form of keyword parameters with carefully set defaults: 1) explicit constraints, and 2) those that control the calculation of this tradeoff.

INTRODUCTION

In the clinical trials field, there is a continuing demand for the rapid production of presentable graphs. Typically, a single program produces many of these graphs.

Axis scaling is the determination of the values of the tick marks along a graph axis. A typical graph of experimental data plots the values of “independent variable” values against those of a “dependent variable.” The independent variable contains values that are controlled by the experimenter, such as the data collection time-points, or the dose of an administered drug; determining the axis scaling for these involves simply entering these known quantities. The dependent variable, on the other hand, contains the responses of the experimental subjects, which are highly variable. For example, a single clinical trials laboratory data-plotting program may produce approximately 30 graphs, each summarizing the data for one lab parameter over all the patients in the trial. Automation of axis-scaling for all of these graphs is a practical necessity as a result.

Sometimes, however, there is a need to plot two variables that both have initially unknown values. This macro has been modified from its previous version (Pugh, 1998) to allow you to scale axes for multiple variables with one call of the %AXISCALE macro.

The most basic consideration in determining tick marks is the determination of the maximum and minimum of the values to be plotted. For aesthetic reasons, there should also be some space between the top of the graph and the maximum point plotted, and an analogous consideration holds for the minimum value. On the other hand, if this space is too large, the data may appear cluttered and possibly be unreadable. There are an optimum number of tick marks: too many make the graph too cluttered, while too few impart too little information. The tick mark label mantissas should have as few non-zero digits as possible for the sake of both appearance and clarity.

Achieving the best tradeoff among all these considerations is further complicated by the fact that the number of tick marks that minimizes the number of digits (ideally, those that are integral multiples of powers of ten) varies greatly. For example, if the maximum value on an axis is 200 and the minimum 0, one additional tick mark, at 100, would minimize the number of non-zero digits, but the number of tick marks would be only three. In the original version of %AXISCALE, various parameters controlling these things were hard-coded, but in this version, they are provided as input parameters, with the defaults set to the previously hard-coded values.

THE ALGORITHM

%AXISCALE first calculates the range of the data, expresses it in exponential notation, and extracts the mantissa (MANTISSA) and exponent (REXP). The value of the mantissa determines the formula to be used to calculate the tick-mark interval. Which one of three different formulas is applied to calculate the interval depends on where the mantissa falls relative to the (default) breakpoints 3.162 (√10), 1.25 (chosen through experiment), and 1. Each formula sets the interval to a power of 10 raised to a different exponent, and then calculates the new axis maximum/minimum by rounding up/down the data maximum/minimum using that interval value. For the biggest mantissa values, over 3.162, it is 10^{REXP}. For those in the middle, 1.25 to 3.162, it is $10^{.5*REXP}$. For the rest, it is $10^{(REXP-1)}$.

You may want to impose controls on the margin space between the data and the borders of the graph, or on the maximum and minimum number of major (labeled) tick marks. These two goals are in conflict and the best tradeoff must be calculated. Wasted space is a special problem when range mantissas are close to &CUTPT2. You can specify maximum and minimum constraints for tick mark counts with the input parameters &MINTICK and &MAXTICK. &MAXWASTE, controlling the proportion of graph space “wasted,” is a weaker constraining variable: if, together with &MINTICK and &MAXTICK, you specify an impossible set of conditions (unfortunately, more likely as &MAXWASTE decreases), the

&MINTICK and &MAXTICK values take precedence.

When there is too much “wasted” space, i.e., &MAXWASTE is less than the value of 1 minus the ratio of an axis range to the range of the data plotted along it, %AXISCALE reduces the previously calculated axis maximum and increases its corresponding minimum. This forces the number of tick marks to increase because the interval needs to be halved for this new range to have equal intervals. The program does not allow this to happen, however, if it would generate too many tick marks or the reduced range would exclude any data points.

The previous paragraph implies value judgments, but never fear: you get to make these judgments! You can reset these breakpoints, determine what limits will be set to moderate the number of tick marks, and much margin space you will allow. But within the constraints that you specify, %AXISCALE will calculate the “best” tradeoff.

THE INPUT PARAMETERS

&VARLIST is a list of numerical variables the values of which are to be plotted on the axes being scaled. &DS is the data set containing the data to be plotted. Both of these are required. &MARGIN increases the data-free space at the extremes of the axes. &MAXWASTE, on the other hand, lets you set the maximum amount of wasted space allowed in the plot. &CUTPT1, &CUTPT2, and &CUTPT3 set the mantissa values at which the scaling algorithm changes to make the tick mark intervals smaller. These values are set to minimize the number of nonzero digits in the tick mark label mantissas. &MAXTICK and &MINTICK set the maximum limits of the range of the possible number of tick marks, imposing a moderating influence on values calculated according to other parameter values.

This version of %AXISCALE contains three new input parameters. &OVERALL is the name of an internally used SAS data set used to calculate the overall maximum for multiple variables contained and separated by commas in &VARLIST. &FIRST is the sequence number of the first variable of &VARLIST within a series of %AXISCALE calls, which corresponds to the n in the first AXISn statement that that %AXISCALE call generates. &LABEL contains the axis label for those AXIS statement

THE OUTPUT PARAMETERS

&MIN1, &MIN2, …, &MINn are the minimum axis values assigned to their respective input variables (where n is the number of variables in &VARLIST). &MAX1, &MAX2, …, &MAXn are the maximum values for the same. &INTRVL1, &INTRVL2, …, INTRVLn are the intervals between tick mark labels for the same. &MINOR1, MINOR2, …, MINORn are the numbers of minor tick marks. The %AXISCALE macro generates AXIS statements containing these values.

THE CODE

```
****************************************;
*  NAME: AXISCALE
*
*  VERSION: 3
*
*  FUNCTION:
*  Determines the placement/value of tick
*  marks along a graph axis
*  according to both practical and
*  aesthetic considerations.
*
*  INPUT PARAMETERS:
*     VARLIST - list of numerical
*       variable to be plotted
*       (required)
*     DS - SAS data set variable is on
*       (required)
*     MARGIN: Pads top and bottom with
*       space (fraction of INTERVAL).
*       Default is .01.
*     MAXWASTE: 1 - the minimum ratio of
*       range of data to range of
*       relevant axis to allow.
*       Default is .05.
*     CUTPT1: Keep at 1.
*     CUTPT2: The first mantissa value at
*       which the scaling algorithm
*       changes to generate a larger
*       scaling interval. Default is
*       1.25.
*     CUTPT3:  The second mantissa value
*       at which scaling algorithm
*       changes to generate a yet larger
*       scaling interval. Default is
*       3.162.
*     MAXTICK: Maximum number of tick
*       marks.  Default is 11.
*     MINTICK: Minimum number of tick
*       marks.  Default is 3.
*     OVERALL:  The output data set including
*       all of the data, created if OVERALL is
*       non-blank.  Default is blank.
*     FIRST:  The sequence number of the first
*       AXIS statement where there are multiple
*       calls to AXISCALE before a PROC GPLOT.
*     LABEL:  Axis label.  Default is blank.
*
*   OUTPUT MACRO VARIABLES:
*     MIN1, MIN2, ..., MINn, where n is
*       the number of variables to scale,
*       are the minimum major axis values
*       assigned to their respective
*       input variables.  n can be a
*       maximum of 99.
*     MAX1, MAX2, ..., MAXn: same as
*       above, but for maximum axis
*       values.
*     INTRVL1, INTRVLn, ..., INTRVLn:
*       same as above, but for intervals
*       between labeled tick marks.
*     MINOR1, MINOR2, ..., MINORn:
*       number of minor tick marks.
*
*  EXAMPLE:
*     %AXISCALE(ds=respdata,
*         varlist=resp1 resp2 resp3)
*         where resp1, -2, and -3 are
*         numerical variables on data set
*         respdata.
*
*  AUTHOR: Dorothy E. Pugh
*
****************************************;
options mprint symbolgen ls=78;

%macro
axiscale(ds=,varlist=,margin=.01,maxwaste=.2,
     cutpt1=1,cutpt2=1.25,cutpt3=3.162,

maxtick=11,mintick=3,label=%str(),overall=,
     first=1);

***********************************************
*  Get an overall min, max, interval for each
*  variable and, if OVERALL is nonblank, for
*  all of them.
***********************************************
;

   data _new;
      set &ds;
      %if %index(&varlist,%str(,)) gt 0 %then
      %do;
         new = max(&varlist);
      %end;
      %else new = &varlist;;
      output;
      %if %index(&varlist,%str(,)) gt 0 %then
      %do;
         new = min(&varlist);
      %end;
      %else new = &varlist;;
      output;
   run;

****************************************;
*  Create separate macro variable for
*  every variable in list
*  and the number of new variables.
****************************************;

   %let i = &first;

   %do %while (%scan(&varlist,&i,' ')
      ne);
      %let var&i = %scan(&varlist,&i,' ');
      %if %scan(&varlist,&i,' ') ne
         %then %let var&i =
             %scan(&varlist,&i,' ');
      %let i = %eval(&i+1);
   %end;

   %let nlist = %eval(&i-1);

      %do i = 1 %to &nlist;
```

```
         %global min&i max&i intrvl&i
                 minor&i;
      %end;
****************************************
*  Loop: One iteration for each variable.
****************************************;

   %do i = 1 %to &nlist;

   proc means data=&ds noprint;
      var &&var&i;
      output out=_temp
             min=min
             max=max;
   run;

   data _null_;
      set _temp;
      oldrange = max - min;
      oldmax = max;
      oldmin = min;
      eformat = put(oldrange,e7.);
      rexp = 1*substr(left(eformat),5,3);
      mantissa =
        round(abs(oldrange/10**rexp),.1);

****************************************;
* Depending on the value of the mantissa,
* determine interval, and by implication,
* the maximum and minimum, for this plot.
****************************************;

      if mantissa ge &cutpt3 then do;
         interval = 10**rexp;
         max = round(max + (.5+&margin)
            *interval,interval);
         min = round(min -(.5+&margin)
            *interval,interval);
         minortck = 9;
      end;

      else if &cutpt2 <= mantissa <
         &cutpt3 then do;
         interval = .5*10**rexp;
         max = round(max + (.5+&margin)
             *interval,interval);
         min = round(min - (.5+&margin)
             *interval,interval);
         minortck = 4;
      end;

      else if &cutpt1 <= mantissa <
         &cutpt2 then do;
         interval = 10**(rexp-1);
         max = round(max + (.5+&margin)
            *interval,interval);
         min = round(min-(.5+&margin)
            *interval,interval);
         minortck = 9;
      end;


***********************************************
*
*  If wasted space is too big and tick mark
*  count is below the specified maximum then
*  decrease maximum, increase minimum, and
*  halve tick mark interval accordingly
*  as long as data is not excluded.
***********************************************
*;
      range = max-min;
      wasted = 1-oldrange/range;
      ticktest = round(1 + (max-min-
         interval)/(.5*interval));

      do while (&maxwaste < wasted and
            ticktest <= &maxtick and
            oldmax lt max - .5*interval
            and oldmin gt min + .5*interval);
         range = max-min;
         max = max - .5*interval;
         min = min + .5*interval;
         interval = .5*interval;
         ticktest = round(1 + (max-min-
           interval)/(.5*interval));
         if minortck eq 4 then minortck = 9;
      end;

****************************************;
* Moderate number of tick marks.
****************************************;

      test = round((max-min)/interval);

      do while (test ge &maxtick);
        if mod(test,2) ne 0 then max =
           max + interval;
        interval = 2*interval;
        test = round((max-min)/interval);
        if minortck eq 4 then minortck = 9;
      end;

      do while (test le &mintick);
         if test le 3 then interval =
            .5*interval;
         test = round((max-min)/interval);
         if minortck eq 9 then minortck = 4;
      end;

      call symput("min&i",trim(left(min)));
      call symput("max&i",trim(left(max)));
      call symput("intrvl&i",
            trim(left(interval)));
      call symput("minor&i",
            trim(left(minortck)));

      run;

      %end;

****************************************
*  Generate AXIS statements.
****************************************;
     %do i = &first %to &nlist;
        axis&i order=&&min&i to &&max&i
           by &&intrvl&i
        minor=(number=&&minor&i)
        label=("&label" h=2);
     %end;

%mend axiscale;
```

Example Input and Output

```
****************************************;
*  Example input and output.  Note
*  that GOPTIONS statement is system-
*  specific.
****************************************;
   data a;
      do j = 1 to 200;
         var1 = 1.1*ranuni(47) - 2.123;
         var2 = 1.4*ranuni(34) - 4.17;
         var3 = 8*ranuni(87) - 3.22;
         var4 = 9.1*ranuni(95) - 1.07;
         output;
      end;
   run;

   %macro prt(plotspec);

      goptions device=win;
      legend1 frame;
      symbol1 v=square c=black;
      symbol2 v=star c=black;
      symbol3 v=diamond c=black;
      symbol4 v=triangle c=black;

      proc gplot data=a;
         plot &plotspec;
      run;

   %mend;


   %axiscale(ds=a,varlist=var1,
        maxwaste=.2,label=%quote(domain: var1))
   %axiscale(ds=a,varlist=var3,first=2,
        label=%quote(range: var3))
   %axiscale(ds=a,varlist=var2,first=3,
        maxwaste=.2,label=%quote(domain: var2))
   %axiscale(ds=a,varlist=var4,first=4,
        label=%quote(range: var4))
   %prt(%quote(var3*var1=2 /legend=legend1
        haxis=axis1 vaxis=axis2))
   %prt(%quote(var4*var2=3 /legend=legend1
        haxis=axis3 vaxis=axis4))


   %axiscale(ds=a,varlist=%str(var1,var2),
        overall=over1,
        maxwaste=.2,label=%quote(domain: var1,
        var2))
   %axiscale(ds=a,varlist=%str(var3,var4),
        overall=over2,first=4,
        label=%quote(range: var3, var4))
   %prt(%quote(var3*var1=2 var4*var2=3 /overlay
        legend=legend1 haxis=axis3
vaxis=axis6))
```

Appendix A contains this example program’s output in Figures 1-3, respectively.

CONCLUSION

There are many considerations going into developing an all-purpose algorithm for scaling axes with evenly spaced tick marks, and experienced programmers (and their clients) may disagree on ideal tradeoffs among them. The purpose of the program described above is to enable users to satisfy different needs with a minimum of effort. For those who wish not to explore these subtleties, the defaults were chosen with you in mind.

ACKNOWLEDGMENTS

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are registered trademarks of their respective companies.

REFERENCES

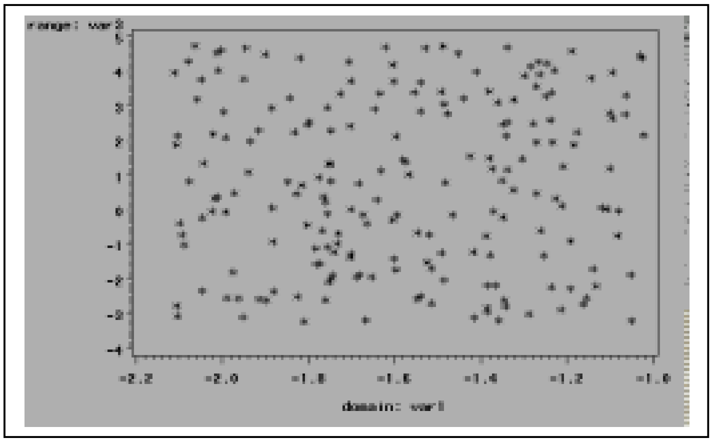
Pugh, D.E. (1998), "Applying a Robust Generalized Axis-Scaling Algorithm to the Use of SAS/GRAPH®," *Proceedings of the Sixth Annual Southeast SAS® Users Group Conference*,275-276.

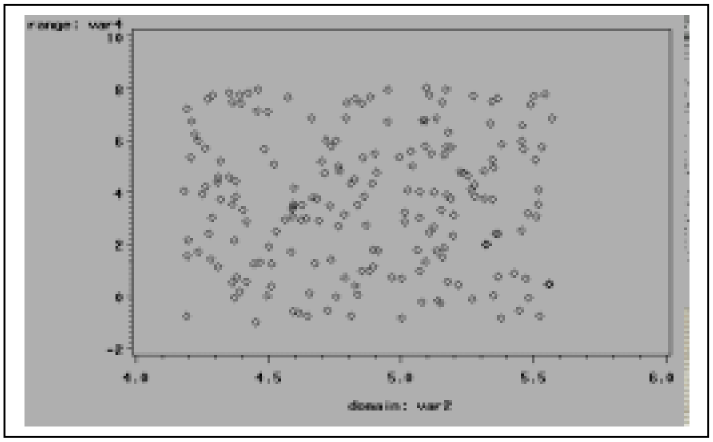
Pugh, D.E. (2000), "A Robust Generalized Axis-Scaling Macro," *Proceedings of the Twenty-Fifth SAS® Users Group International Conference, 580-584.*

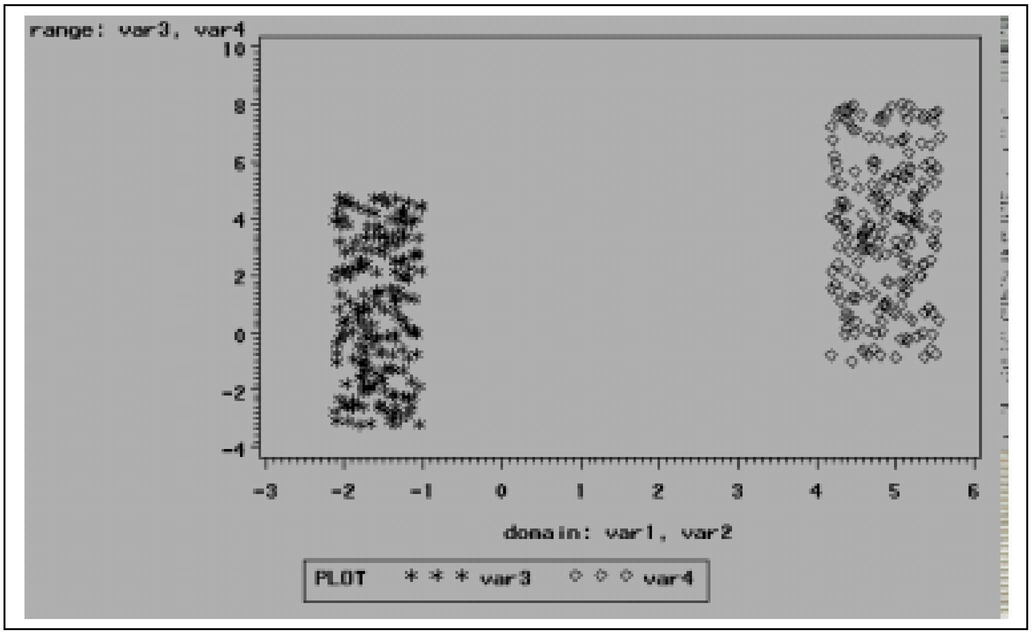
AUTHOR CONTACT INFORMATION

Dorothy E. Pugh
29 Sandstone Ridge Dr.
Durham, NC 27713
(919) 493-1237
dorothypugh@aol.com

APPENDIX A: Output of program






range: var3, var4
10
8
6
4
2
0
-2
-4
-3
-2
-1
0
1
2
3
4
5
6
domain: var1, var2
PLOT * * * var3 ◇ ◇ ◇ var4