

Paper 112-25

A Robust Generalized Axis-scaling Macro

Dorothy E. Pugh, University of North Carolina at Chapel Hill, Chapel Hill, NC

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) minimize wasted space in the resulting plot. This paper describes a simple axis-scaling algorithm, which enables programmers with different needs to achieve a tradeoff among these criteria that suits these needs. Five new input keyword parameters make it possible to control settings to achieve the above three goals; carefully chosen defaults exist for all of them.

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 the determination of tick marks 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; a large space can make room for a legend. On the other hand, if this space is too large, the data may appear cluttered and possibly be unreadable. There is 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 previous 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 ($\sqrt{10}$), 1.25 (chosen through experiment), and 1. Each formula sets the interval to a power of 10 raised to a different exponent, 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 $10^{5*\text{REXP}}$. For the rest, it is $10^{(\text{REXP}-1)}$.

Unfortunately, this third class containing the smallest mantissas presents the greatest problem with wasted margin space. Because of this, another part of the algorithm corrects this problem. It reduces the previously calculated maximum and minimum axis range. 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.

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 space at the top and the bottom of the plot which does not contain data; it is useful when you need to use legends. &WASTED, 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.

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 CODE

```
*****;
* NAME: AXISCALE
*
* VERSION: 2
*
* 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 of
* plot with blank space.
* Default is .01.
* WASTED: The minimum ratio of
* of range of data to range of
* relevant axis to allow.
* Default is .8.
* 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.
```

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(resp1 resp2 resp3,
* respdata)
* where resp1, -2, and -3 are
* numerical variables on data set
* respdata.
```

* AUTHOR: Dorothy E. Pugh

```
*****;
options mprint symbolgen ls=78;
```

```
%macro axiscale(varlist=,ds=,margin=.01,
cutpt1=1,cutpt2=1.25,cutpt3=3.162,
maxtick=11,mintick=3,wasted=.8);
```

```
*****;
```

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

```
%let i = 1;

%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;
eformat = put(oldrange,e7.);
rexp = 1*substr(left(eformat),5,3);
mantissa =
round(abs(oldrange/10**rexp),.1);
oldmax = max;
oldmin = min;
```

```
*****;
* 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;
```

```
*****
* This reduces wasted space by
* making axis range smaller, but
* only if the resulting tick
* mark count does not exceed
* &maxtick, the number of tick
* marks is odd, and no points are
* excluded.
*****;
```

```
flag = 0;
range = max-min;
oldnew = oldrange/range;
ticktest = round(1 + (max-min-
interval)/(.5*interval));
if oldnew lt &wasted and
ticktest le &maxtick and
&margin le .01 and
oldmax lt max - .5*interval
and
oldmin gt min + .5*interval
then do;
max = max - .5*interval;
min = min + .5*interval;
interval = .5*interval;
flag = 1;
if minortck eq 9 then minortck =
4;
end;
```

```
*****;
```

```

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

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

do while (test ge &maxtick and flag
eq 0);
  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 and flag
eq 0);
  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;

%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 = 208*ranuni(34) - 104;
    var2 = 22.44*ranuni(34) - 11.22;
    var3 = 2.8*ranuni(87) - 1.4;
    var4 = 34.5*ranuni(34) - 17.25;
    output;
  end;

```

```

run;

%macro prt(v1,v2);

  goptions device=win;

  %do i = 1 %to &nlist;
    axis&i order=&&min&i to &&max&i
      by &&intrvl&i
        minor=(number=&&minor&i);
  %end;

  proc gplot data=a;
    plot &v2*&v1=1 / haxis=axis1
      vaxis=axis2;

  run;

%mend;

%axiscale(varlist=var1 var2,ds=a)
%prt(var1,var2)
%axiscale(varlist=var3 var4,ds=a)
%prt(var3,var4)
%axiscale(varlist=var3 var4,ds=a,
margin=1)
%prt(var3,var4)

```

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 them in mind.

ACKNOWLEDGMENTS

The author would like to thank Jeff Abolafia for his suggestions and for testing the program.

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.

REFERENCE

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*, 6,275-276.

AUTHOR CONTACT INFORMATION

Dorothy E. Pugh
 29 Sandstone Ridge Dr.
 Durham, NC 27713
 (919) 493-1237
 KarlG87710@aol.com

APPENDIX A: Output of program

