

Paper RIV122

The Last Axis Macro You'll Ever Need

Shane Rosanbalm, Rho, Inc.

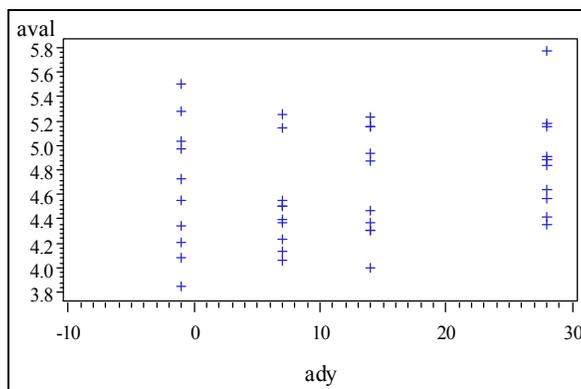
ABSTRACT

There are several good papers out there about automating the creation of SAS® axes (Dorothy E. Pugh 2000, Don Li 2003, Rick Edwards 2012). These papers are written with classic SAS/GRAPH in mind. But, with the rise of ODS Graphics and the corresponding much improved axes, one might reasonably ask the question, "Do we even need axis macros anymore?" While I wholeheartedly agree that the ODS Graphics axis defaults are much nicer than what we were used to getting out of classic SAS/GRAPH, there are still numerous situations in which we will not want to leave control of axis ranges entirely up to SAS.

In this paper I take what I see as the best ideas from the above papers and combine them into a bigger, faster, stronger axis macro. The default behavior of the new macro is to mimic what ODS Graphics would give you. The new macro also includes several optional parameters that allow the behavior to be customized to fit a wide variety of specialty situations (multiple variables, reference values, preferred number of tick marks, etc.). By the end of this paper I hope you'll agree that this is indeed the last axis macro you'll ever need!

HOW AXIS MACROS CAME TO BE

The continuous axis ranges selected by GPLOT leave much to be desired. From excessive white space to wildly inconsistent numbers of tick marks, many are the complaints against the GPLOT axes.



Being displeased with these y-axis ranges, programmers set about writing axis macros to calculate better axis ranges. These axis macros would typically use observed data values to generate the components of the AXISn ORDER option. For example,

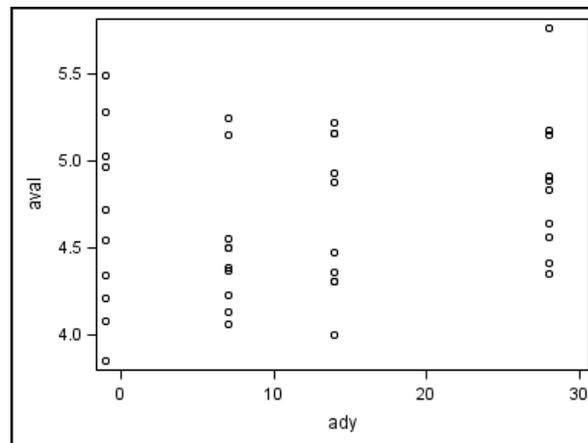
```
axis1 order=(3.5 to 6 by .5);
```

These axis macros provide the results in a variety of ways, but for the sake of concreteness let's assume that an axis macro generates macro variables &start, &end, and &by which contain the values 3.5, 6, and 0.5. The programmer then uses these macro variables to build the ORDER option. For example,

```
axis1 order=(&start to &end by &by);
```

DO WE STILL NEED AXIS MACROS?

And then along comes SGPLOT. Suddenly the axis ranges are so much better. White space problems are virtually non-existent, and the number of tick marks is remarkable consistent from one plot to the next.



So, maybe all of these axis macros can be swept into the dustbin of programming obscurity.

WHY YOU STILL NEED AXIS MACROS

As good as SGPLOT is at producing nice axis ranges, there are still several reasons why you might not want to leave axis ranges subject to the SGPLOT defaults.

You Know Something That SGPLOT Does Not

Perhaps you want to make sure that a certain value is always part of the axis range (e.g., zero). Perhaps you need to allow extra space for some annotations at the top of the data area. Perhaps you know that your principal investigator likes to use a certain set of increments for various data types (e.g., increments of 5 for blood pressure plots). Whatever the something is that you know, it often requires the use of the VALUES option to implement.

Managing START, END, and BY Values Is Tedious

You've just combined the current data values with the "something that SGPLOT doesn't know" to form the perfect suite of START, END, and BY values for your suite of lab plots. It took a few hours, but everything now looks great, so you don't mind the manual effort that was required.

And then you get an email that a new data cut has been sent. You now have to rerun all of the plots and check that all of the START, END, and BY values that you specified for the previous data cut still work with the new data cut. Don't you wish you'd used a macro to determine the START, END, and BY to use in your suite of plots?

Consistent Axes across Multiple Pages

It is common in Phase I clinical trials to produce PK concentration plots with one subject per page. Reviewers like to see consistent axis ranges across all of the pages. However, if you create a macro %do loop that generates one SGPLOT per subject, you will get different axis ranges for each subjects. In order to get consistent axis ranges for all subjects you would need to specify the common axis range using the VALUES option.

SGPLOT Is Fairly Rigid about the Number of Tick Marks Provided

If you experiment with the HEIGHT option of the ODS GRAPHICS statement you'll find that you really have to go to extremes to get SGPLOT to give you a different number of tick marks. You can make a 2in figure or a 12in figure, and both are likely to have the same number of tick marks. If you want a little more freedom of choice in the number of tick marks, you'll have to use the VALUES option.

SAS Version

SGPLOT was first introduced in v9.2. While the aesthetics were quite nice, there was still a lot of missing functionality relative to what could be produced with GPLOT. But with v9.3, and certainly with v9.4, the SGPLOT feature set has grown rich enough to essentially push GPLOT to the side.

Unfortunately, many organizations are not able to immediately upgrade to the latest SAS releases. Some large companies go years between upgrades, meaning that some are still using v9.2 (or even v9.1.3). For these organizations there is really no choice but to use GPLOT (and therefore axis macros).

SOME DATA TO WORK WITH

Throughout this paper the dataset ALBUMIN is utilized. This straightforward dataset is generated with DO loops and the RANNOR function.

```
data albumin;
  do subject = 1 to 10;
    do ady = -1, 7, 14, 28;
      aval = 4.4 + rannor(1)/2 + ady/50;
      anrlo = 3.4;
      anrhi = 5.4;
      output;
    end;
  end;
run;
```

THE %AXISORDER MACRO

The Basics

The axis macro presented in this paper is named %AxisOrder. The macro has two required parameters.

Data (req)	Input dataset E.g., data = albumin
Var (req)	List of variables whose values are to be included in the axis range E.g., var = aval

Given a dataset and variable name(s), the macro performs some calculations and communicates the derived axis range back to the programmer through a set of output macro variables.

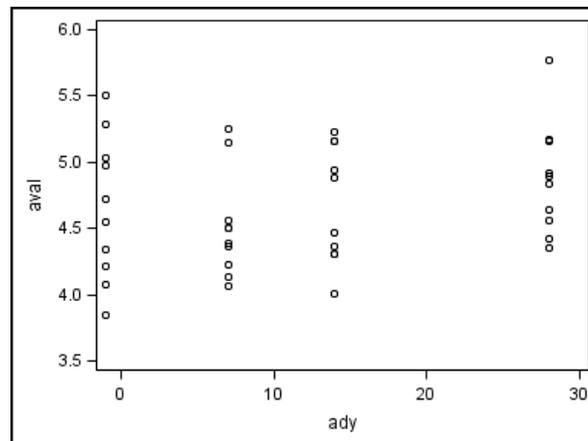
&_AxisStart	Start value for VALUES option E.g., &_AxisStart = 3.5
&_AxisEnd	End value for VALUES option E.g., &_AxisEnd = 6
&_AxisBy	By value for VALUES option E.g., &_AxisBy = 0.5
&_AxisList	A space-separated list of values beginning at &_AxisStart, ending at &_AxisEnd, and incrementing by &_AxisBy E.g., &_AxisList = 3.5 4 4.5 5 5.5 6

A basic call to the macro might appear as follows:

```
%AxisOrder
  (data=albumin
  ,var=aval
  );
```

This call to the macro produces an axis range that is large enough to contain all values of AVAL. For this particular dataset the macro returns &_AxisStart = 3.5, &_AxisEnd = 6, and &_AxisBy = 0.5. The output macro variables are then used in the SGLOT VALUES option as follows:

```
proc sgplot data=albumin;
  scatter x=ady y=aval;
  yaxis values=(&_AxisStart to &_AxisEnd by &_AxisBy);
run;
```



The macro also returns `&_AxisList = 3.5 4 4.5 5 5.5 6`, which will allow for a slightly more succinct coding:

```
proc sgplot data=albumin;
  scatter x=ady y=aval;
  yaxis values=(&_AxisList);
run;
```

The beauty of using axis macros is, if the range of AVAL changes in a subsequent data cut, the values of the macro variables will change with them. You write the code once and then it updates the `&_AxisStart`, `&_AxisEnd`, `&_AxisBy`, and `&_AxisList` values for you with each rerun of the code.

Behind the Scenes

The macro works by first selecting the best BY value for the data at hand. The set of candidate BY values that the macro considers starts with 1, 2, 2.5 and 5. The macro also considers variations on these four candidate BY values by shifting the decimal left and right (e.g., 0.1, 0.2, 0.25, and 0.5; e.g., 10, 20, 25, and 50; etc.).

Two factors are taken into account when evaluating candidate BY values: how many tick marks will be produced and how much white space will appear at the top and bottom of the plot. The best BY value is the one that strikes the right balance between providing a reasonable number of tick marks without leaving too much white space.

Once the best BY value has been selected, it is then straightforward for the macro to use the min and max values in the data to calculate corresponding START and END values. These START, END, and BY values are then placed into macro variables `&_AxisStart`, `&_AxisEnd`, `&_AxisBy`, and `&_AxisList` for use by the programmer.

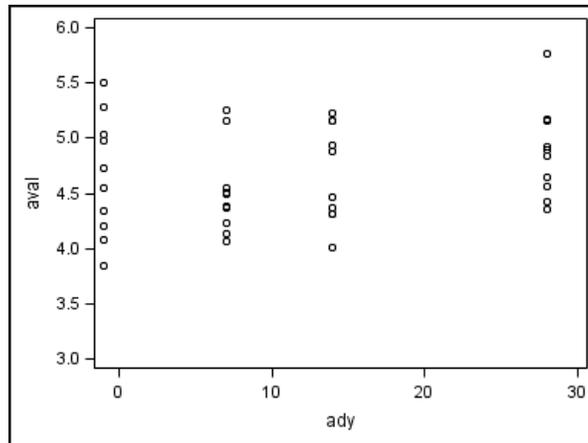
Multiple Variable Support

Suppose that we want to be sure that our albumin axis range contains not just the values of the variable AVAL, but also the values of variables ANRLO and ANRHI. To achieve this goal we would add more variables to the VAR list:

```
%AxisOrder
  (data=albumin
   ,var=aval anrlo anrhi
  );

proc sgplot data=albumin;
  scatter x=ady y=aval;
  yaxis values=(&_AxisList);
run;
```

The addition of ANRLO and ANRHI to the VAR parameter ensures that the values of these variables (3.4 and 5.4) are included in the axis range.



Important Optional Parameters

The macro has several optional parameters. The most important of the optional parameters are:

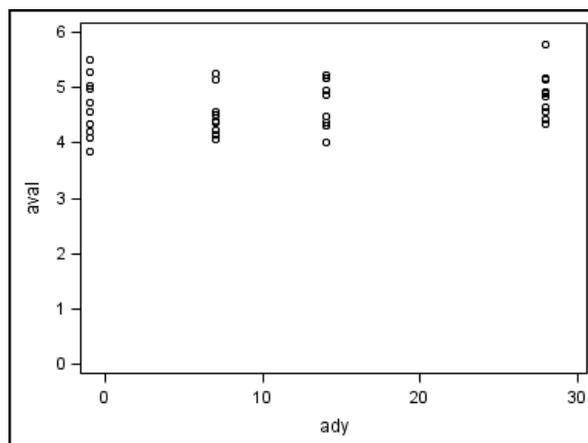
Val (opt)	List of specific values to include in axis range E.g., val = 0
MajTarg (opt)	Target number of major tick marks E.g., majtarg = 4

Suppose that we want to be sure that our albumin axis range contains the value zero. To achieve this goal we would add VAL to the macro call:

```
%AxisOrder
  (data=albumin
   ,var=aval
   ,val=0
  );

proc sgplot data=albumin;
  scatter x=ady y=aval;
  yaxis values=(&_AxisList);
run;
```

The use of VAL=0 ensures that the value zero is included in the axis range produced.



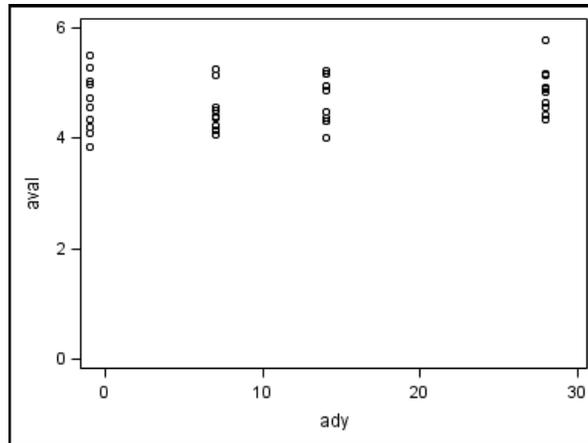
Suppose that we want to have fewer tick marks on our y-axis (in particular, suppose we want 4 tick marks). To achieve this goal we would add MAJTARG to the macro call:

```

%AxisOrder
  (data=albumin
   ,var=aval
   ,val=0
   ,majtarg=4
  );

proc sgplot data=albumin;
  scatter x=ady y=aval;
  yaxis values=(&_AxisList);
run;

```



Notes that the parameter MAJTARG is exactly what it says it is: a *target* number of major tick marks. Given the candidate list of BY values (i.e., 1, 2, 2.5, and 5), it might not be possible to produce an axis range with exactly the target number of tick marks. The macro does its best to provide the target number of tick marks, but may instead provide slightly fewer or greater tick marks if the target number of tick marks is simply not possible given the data you are working with. The default value is MAJTARG=6. This value seems to roughly approximate the default behavior of SGPLOT.

More Optional Parameters

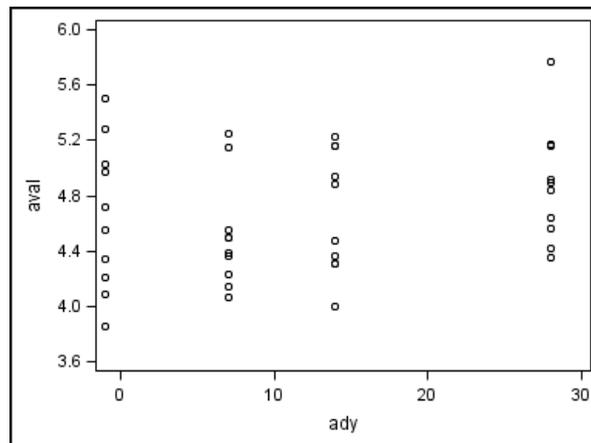
A few of the macro's other optional parameters are:

By (opt)	By value override; used to force the value of &_AxisBy. E.g., by = 5 will result in &_AxisBy=5.
Prefix (opt)	Text to add to the beginning of the output macro variable names; used when both the y-axis and x-axis need to use the axis macro. E.g., prefix = y will result in &_yAxisStart, &_yAxisEnd, etc.
MajDev (opt)	Acceptable deviation from MajTarg value; default value is 1. E.g., majtarg = 6 and majdev = 2 tells the macro that anywhere from 4 to 8 tick marks is acceptable to you.
RoundBys (opt)	Round BY values only (i.e., 1, 2, 2.5, and 5); set to N if BY values of 1.25, 1.5, 3, 4, 6, 7.5, and 8 are also acceptable. E.g., roundbys = N widens the set of candidate BY values.

As an illustration of just one of these optional parameters, suppose that we want to be sure that our resulting axis range increments by 0.4. To achieve this goal we would modify the original call as follows:

```
%AxisOrder
  (data=albumin
   ,var=aval
   ,by=0.4
  );

proc sgplot data=albumin;
  scatter x=ady y=aval;
  yaxis values=(&_AxisList);
run;
```



One Last Optional Parameters

The final optional parameter is an attempt to duplicate the non-bounding tick marks that SGPLOT will sometimes produce:

Threshold (opt)	Specifies a threshold for displaying one more tick at the low and high end of the axis. E.g., threshold=0.7 Default is 0 (i.e., do not add bounding tick marks). Valid values are 0 to 1. Interaction: this option only effects &_AxisListUnb, &_AxisMin, and &_AxisMax.
-----------------	--

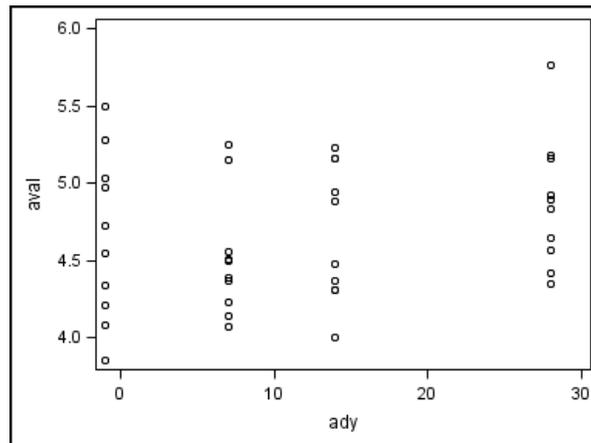
The associated output macro variables for THRESHOLD are:

&_AxisListUnb	An alternate version of &_AxisList that allows for unbounded data values in the resulting plot. E.g., &_AxisListUnb = 4 4.5 5 5.5
&_AxisMin	Value needed for YAXIS MIN= option in order to get SGPLOT-like non-bounding axes with &_AxisListUnb. E.g., &_AxisMin = 3.85
&_AxisMax	Value needed for YAXIS MAX= option in order to get SGPLOT-like non-bounding axes with &_AxisListUnb. E.g., &_AxisMax = 5.76

The following example uses THRESHOLD=0.5, meaning that the low/high values of AVAL must extend at least half way to the next tick mark in order for bounding tick marks to appear:

```
%AxisOrder
  (data=albumin
   ,var=aval
   ,threshold=0.5
  );

proc sgplot data=albumin;
  scatter x=ady y=aval;
  yaxis values=(&_AxisListUnb)
         min=&_AxisMin max=&_AxisMax valueshint;
run;
```



In this particular dataset the highest AVAL of 5.76 is more than half the distance from 5.5 to 6, so the upper tick mark of 6 is retained. The lowest AVAL of 3.85 is less than half the distance from 4 to 3.5, so the lower tick mark of 3.5 is excluded.

Note that in order for &_AxisListUnb to achieve the desired effect, we must additionally modify the YAXIS statement with the addition of options MIN=&_AxisMin, MAX=&_AxisMax, and VALUESHINT.

CONCLUSION

There are other axis macros out there, but they were written with GPLOT in mind. And even though we now have SGPLOT, we will still want (and sometimes need) axis macros from time to time. The design philosophy of %AxisOrder is to approximate the behavior of SGPLOT by producing a reasonable number of tick marks without introducing too much white space. Putting %AxisOrder in your code from the beginning makes the VALUES option self-maintaining, reducing the overall coding effort. And the optional parameters allow for many variations in the default behavior of the macro.

MACRO SOURCE CODE

The axis macro source code is available for download at graphics.rhoworld.com/tools/axismacro.

RECOMMENDED READING

- [Graphically Speaking](#)

CONTACT INFORMATION

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

Shane Rosanbalm
Rho, Inc
6330 Quadrangle Dr.

Chapel Hill, NC 27517
 919.595.6273
 E-mail: shane_rosanbalm@rhoworld.com
 Web: www.rhoworld.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 1: %AXISMACRO PARAMETERS

Parameter	Description
Data (req)	Input dataset E.g., data = albumin
Var (req)	List of variables whose values are to be included in the axis range E.g., var = aval anrlo anrhi
Val (opt)	List of specific values to include in axis range E.g., val = 0
MajTarg (opt)	Target number of major tick marks; default value is 6. E.g., majtarg = 4
By (opt)	By value override; used to force the value of &_AxisBy. E.g., by = 0.75
Prefix (opt)	Text to add to the beginning of the output macro variable names; used when both the y-axis and x-axis need to use the axis macro. E.g., prefix = y
MajDev (opt)	Acceptable deviation from MajTarg value; default value is 1. E.g., majdev = 2
Threshold (opt)	Specifies a threshold for displaying one more tick at the low and high end of the axis; default value is 0. E.g., threshold=0.7 Valid values are from 0 to 1. Interaction: this option only effects &_AxisListUnb, &_AxisMin, and &_AxisMax.
RoundBys (opt)	Round BY values only (i.e., 1, 2, 2.5, and 5); set to N if BY values of 1.25, 1.5, 3, 4, 6, 7.5, and 8 are also acceptable. E.g., roundbys = N Valid values are Y, N; default is Y.

APPENDIX 2: %AXISMACRO OUTPUT MACRO VARIABLES

Output	Description
&_AxisStart	Start value for VALUES option E.g., &_AxisStart = 3.5
&_AxisEnd	End value for VALUES option E.g., &_AxisEnd = 6
&_AxisBy	By value for VALUES option E.g., &_AxisBy = 0.5
&_AxisList	A space-separated list of values beginning at &_AxisStart, ending at &_AxisEnd, and incrementing by &_AxisBy E.g., &_AxisList = 3.5 4 4.5 5 5.5 6
&_AxisListUnb	An alternate version of &_AxisList that allows for unbounded data values in the resulting plot. Use the optional parameter Threshold to control how aggressively to add bounding tick marks (similar to Thresholdmin/Thresholdmax). E.g., &_AxisListUnb = 4 4.5 5 5.5
&_AxisMin	Value needed for YAXIS MIN= option in order to get SGPLOT-like non-bounding axes with &_AxisListUnb. E.g., &_AxisMin = 3.85
&_AxisMax	Value needed for YAXIS MAX= option in order to get SGPLOT-like non-bounding axes with &_AxisListUnb. E.g., &_AxisMax = 5.76