Paper BB-09

Automating SAS/Graph® Axis Ranges

Rick Edwards, PPD Inc., Wilmington, NC

ABSTRACT

This paper describes a macro that determines the min, max and increment for the axis order definition based on the axis' data which creates 8 to 12 increments based on multiples of 1, 2, 2.5 or 5 for easy interpolation. Flexibility is provided by parameters to insure a default minimum, maximum or both are contained in the range and to insure that zero is the minimum or maximum as desired.

INTRODUCTION

The motivation for creating this macro was provided by an assignment of creating 80+ graphs every two weeks with a maximum of three days to run the graphs (and accompanying tables and listings) against the new data, distribute for review, make any needed updates and rerun the deliverables before sending them to the client. Additionally, the client wanted each graph's vertical axis to have 10 increments. In order to consistently produce the deliverables within the allotted time, the axes' ranges had to be determined programmatically.

There are at least two easy solutions to this problem; allow SAS/GRAPH® to set the range and increments, or find the min and max and divide the difference by 10 for the increment. SAS/GRAPH produces rounded, easily interpolated numbers for the tick marks. However, the number of increments SAS/GRAPH selects can range in number from 3 to 20. Using the min, max and one-tenth interval produces the desired 10 increments, but the values of the tick marks are usually values which make interpolating between them nearly impossible.

The solution lay in a compromise that allows an acceptable range of increments, from 8 to 12, whose values facilitate interpolation. The axis ranges are calculate by a fairly simple macro and prevents the need to manually adjust axis ranges due to new minimum and maximum values appearing in the bi-weekly updated data. The macro is designed to give the minimum amount of empty space at the ends of the axis for maximum readability, while, simultaneously, insuring that the axis range included all of the data points.

Two sets of comparisons follow:

Figures: A B C

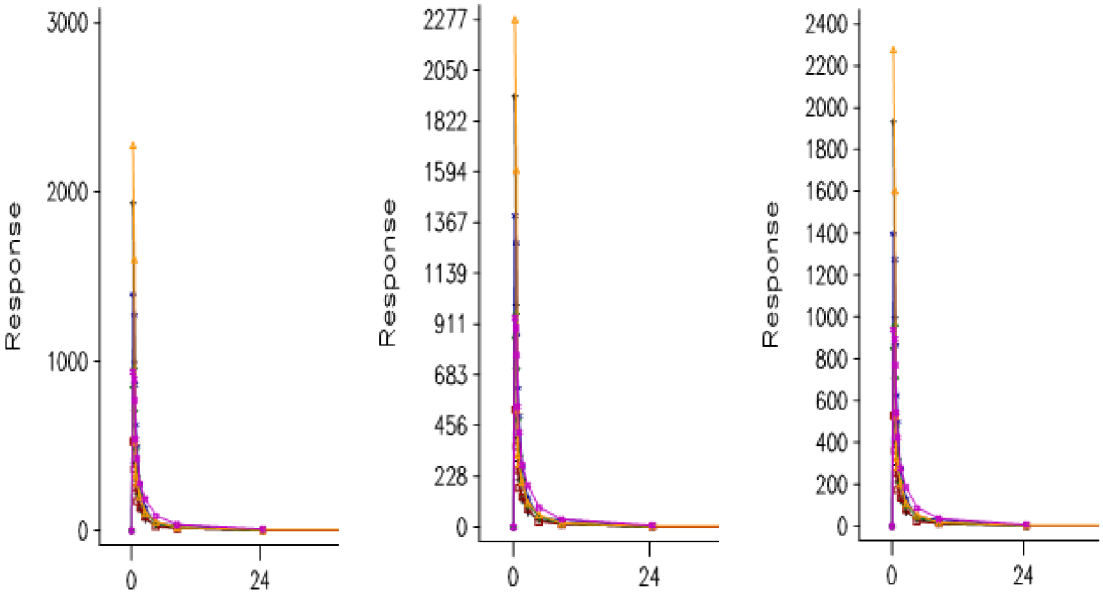

Figure 1 shows the differences in the Response axis min, max and increment using the SAS/GRAPH default (A), the min/max and one-tenth increment method (B) and the macro calculated values (C) for a large data range.

Figures: A B C

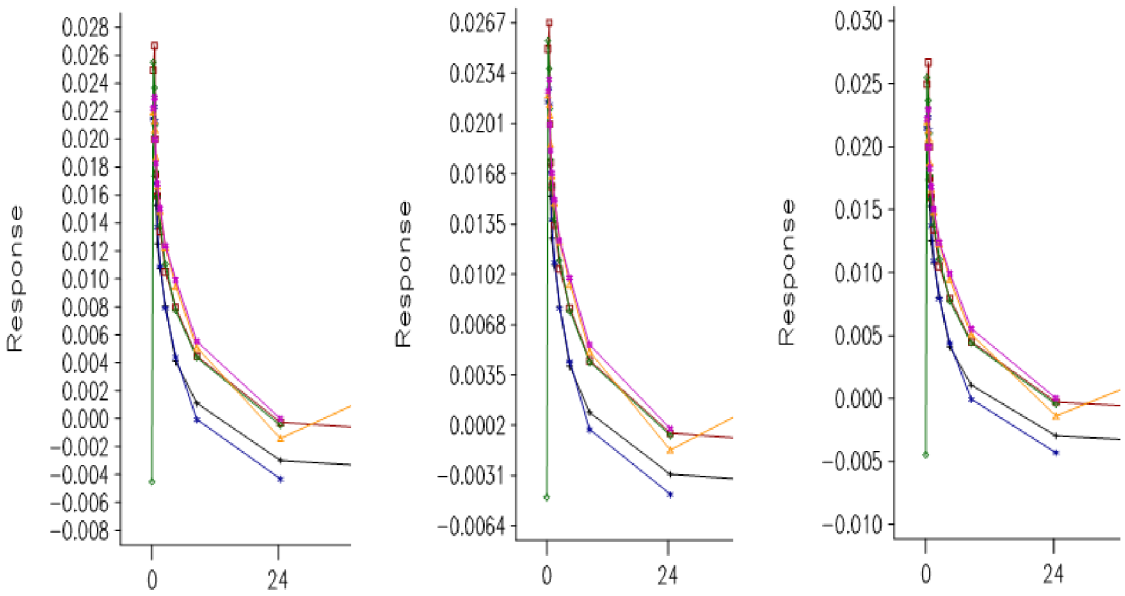

Figure 2 shows the differences in the Response axis min, max and increment using the SAS/GRAPH default (A), the min/max and one-tenth increment method (B) and the macro calculated values (C) for a small data range.

THE BASICS

The macro definition is:

```
%macro set_axis_minmaxincrement(ds                 = ,
                                axisvar            = ,
                                returned_min       = axis_min,
                                returned_max       = axis_max,
                                returned_increment = axis_increment,
                                axis_length        = 51,
                                sa_min             = 999999,
                                sa_max             = -999999,
                                force_zero         = 0
                               );
```

The first two parameters are required and must be set in the macro call. DS is set to the name of the dataset which holds the values that will be used to derive the axis range and increment and AXISVAR is the name of the variable I the dataset which holds those values. Typically, the dataset name named in the DS parameter is the dataset that will be used by the SAS/GRAPH proc using the axis definition and AXISVAR is the name of the variable that will be plotted on the axis whose order values are returned by the macro. However, that does not have to be the case. One example would be when additional data-driven annotation would require the annotations to be drawn above the maximum and/or below the minimum plotted values. In this case, the new minimums and maximums for the axis range required for the annotation would be calculated in a new dataset which is then passed to the macro.

The remaining parameters are pre-defined and therefore optional. The last 4 parameters provide additional flexibility that is occasionally required for certain graphs and they will be detailed in a later section. The three parameters which define the returned values for the ORDER option are used for clarity in the code if the macro definition is not in the same program as the call.

BASIC MACRO CALL

The typical use of the macro follows. The dataset name is assumed to be GRAPH_DATA and the variable, whose minimum and maximum values will be used to determine the range, is RESPONSE.

```
%set_axis_minmaxincrement(ds=graph_data, axisvar=response);
axis1  order(&axis_min to &axis_max by &axis_increment <additional axis options> ;
```

For additional clarity, the returned macro variables may be named explicitly.

```
%set_axis_minmaxincrement(ds=graph_data, axisvar=response, returned_min=the_min,
                          returned_max=the_max, returned_increment=the_increment,);
axis1  order(&the_min to &the_max by &the_increment <additional axis options> ;
```

THE ALGORITHM

The heart of the macro consists of three simple steps. First, the data's range is compared to a standardized range, which lies between 6 and 60, to find the conversion unit between the standard range and the actual range. Next, the standard increment is chosen based on set divisions within the standardized range and converted to the axis increment. Finally, the min and max for the axis range are found by rounding the low and high values using the increment as the rounding unit.

COMPARE THE RANGE TO THE STANDARD

The standard range for this macro lies between 6 and 60. This range was chosen to make the code as transparent as possible when selecting the standard increment, to allow a large number of choices (due to the number of factors of 10 and 12) and to keep the conversion unit the same for all of the standard increments. The actual range is compared with the standard range by reducing or increasing the range check point by powers of ten until the range falls within the standard range. If the axis range is below the standard minimum, the low range check point is reduced by a power of ten until it is less than the actual range. Similarly, if the axis range is larger than the standard range, then the high range check point is increased by a power of ten until it is greater than the actual range. The conversion unit is saved for converting the standard increment to the actual increment used for the axis.

```
/* Range is less than the standard range */
if axisrange <= 6 then do;
   check = 6;
   conversion_unit = 1;
   do until (axisrange > check);
       check = check / 10;
       if axisrange <= check then conversion_unit = conversion_unit / 10;
   end;
end;
/* Range is within or greater than the standard range */
else do;
   check = 60;
   conversion_unit = 1.0;
   do while (axisrange > check);
       check = check * 10;
       conversion_unit = conversion_unit * 10;
   end;
end;
/* standardize the range to lie in the standard range (between 6 to 60) */
unit_range = axisrange / conversion_unit;
```

FIND THE INCREMENT

The divisions for determining the increment from the standardized range are chosen to provide a maximum of 12 increments. Starting with a standardized increment of 1, each increment division stops at its multiple of 12. The standard increment is then converted to the actual increment using the conversion unit from the first step. If additional refinement is desired, other increment values can be added simply by setting an upper limit of 12 times the increment, such as a standard increment of 3 when unit_range < 36.

```
If      unit_range < 12 then  axisinc = 1   * conversion_unit;
else if unit_range < 24 then  axisinc = 2   * conversion_unit;
else if unit_range < 30 then  axisinc = 2.5 * conversion_unit;
else                          axisinc = 5   * conversion_unit;
```

SET THE ACTUAL RANGE

The final step is to set the min and max for the actual axis range. In order to make the min and max multiples of the increment, the low and high values are rounded to the increment value. To prevent the min from rounding up to a value that is greater than the lowest value in the data, a check is made for this result and, if so, the min is reduced by one increment to insure that the axis minimum value is less than the lowest data value. A similar check is made for the high value rounding down to a value that would make the max value less than the data's highest value.

```
/* Round the min's value to match the increment; if the number is rounded up so that
   it becomes larger than the lowest data value, decrease the min by one increment */
axislow = round(low, axisinc);
if axislow > low then axislow = axislow - axisinc;
/* Round the max; if the number is rounded down, increase the max by one increment */
axishigh = round(high, axisinc);
if axishigh < high then axishigh = axishigh + axisinc;
```

THE MACRO PARAMETERS

This section contains a short description of each of the macro's parameters. Additional details on the use of the optional parameters, AXIS_LENGTH, SA_MIN, SA_MAX and FORCE_ZERO, are found in the following section.

DS

Default value: None; mandatory parameter. The name of the dataset which contains the smallest and largest values that will appear on the graph for a particular axis. Typically, this is the name of the dataset specified in the SAS/GRAPH proc data = option.

AXISVAR

Default value: None; mandatory parameter. The name of the variable which contains the smallest and largest values that will appear on the graph for a particular axis. Typically, this is the name of the variable specified in the SAS/GRAPH proc that will be plotted on that axis.

RETURNED_MIN

Default value: axis_min. The name of the global macro variable set to the starting value of the range that will be used in the order option of the AXIS statement.

RETURNED_MAX

Default value: axis_max. The name of the global macro variable set to the ending value of the range that will be used in the order option of the AXIS statement.

RETURNED_INCREMENT

Default value: axis_increment. The name of the global macro variable set to the increment value for the range that will be used in the order option of the AXIS statement.

AXIS_LENGTH

Default value: 51. A numeric value greater than 50 or less than or equal to 50 that determines the range of increments. If the value is greater than 50, the range has 8 – 12 increments. If the value is less than or equal to 50, the range has 4 – 6 increments.

SA_MIN

Default value: 999999. A numeric value which specifies the maximum start of the range. If the minimum data value of AXISVAR is greater than SA_MIN, then SA_MIN is used as the low data value for calculating the range; otherwise, the minimum data value of AXIS VAR is used as the low value. The start of the axis range will be at least as low as SA_MIN.

SA_MAX

Default value: -999999. A numeric value which specifies the minimum start of the data range. If the maximum data value of AXISVAR is less than SA_MAX, then SA_MAX is used as the high data value for calculating the range;

otherwise, the maximum data value of AXIS VAR is used as the high value. The end of the axis range will be at least as high as SA_MIN.

FORCE_ZERO

Default value: 0. A numeric value of 0 or 1. If the value is 1, then the macro insures that 0 is included in the range. If the data range's low value is greater than 0, then the low data value, used to calculate the standard range, is set to 0. If the data range's high is less than 0, then the high data value is set to 0.

USE OF THE OPTIONAL PARAMETERS

REDUCE THE NUMBER OF INCREMENTS

The length of an axis may be shortened due to space constraints; for example, displaying two graphs on one page or a large number of titles and footnotes accompanying the graph. If the length of the axis is too short for 8 to 12 increment values to fit properly, you can still use the macro to set the axis range using the AXIS_LENGTH parameter. Set the AXIS_LENGTH parameter to a value of 50 or less to have the macro produce an increment value which will produce 4 to 6 increments instead of 8 - 12.

Originally, the parameter was meant to correspond to the LENGTH option (in percent) in the AXIS statement with multiple sets of code to set different numbers of increments based on the values passed in. Due to time constraints and practical considerations, only the two options, 8 – 12 increments or 4 -6 increments, are currently coded in the macro.

INSURE THAT A VALUE IS CONTAINED IN THE AXIS RANGE

There are some graphs that require the axis range to include a value that is not guaranteed to be in the range of data values. Use the SA_MIN and/or SA_MAX parameters to insure the axis range includes the desired value. If SA_MIN is set to a value other than the default value of 999999, then the macros sets the data low to the SA_MIN if the data low value is greater than the SA_MIN value. SA_MAX is treated similar when compared to the data high value.

One example of the use of SA_MIN is when an axis should always start at 0. If the data values are relatively small, it would not be uncommon for the minimum value to be greater than 0 since the macro attempts to minimize the amount of empty space at the ends of the axis. To insure that the minimum axis value is at least 0, the call should include SA_MIN = 0 as one of the parameters.

Another example is the use of reference lines to indicate normal low and high ranges in the data. Setting SA_MIN to the normal low value and SA_MAX to the normal high value would insure that the axis range includes these values regardless of the actual low and high data ranges.

Finally, annotations that may lay above or below certain data points would require additional space in the axis range. A variable that contains the axis data values with the additional range percentage added or subtracted needed to display the annotations could be used as the AXISVAR instead of the variable that will actually be plotted against the axis.

INSURE THAT THE AXIS RANGE CONTAINS ZERO

A special case of the insuring that the graph contains a certain value is the need to insure that an axis range always contains the value of 0. There are two ways this can be done. First, both SA_MIN and SA_MAX can be set to 0. If the data range contains positive and negative numbers, then the SA_MIN and SA_MAX values will not be used. If the data range contains all positive numbers, then SA_MIN will be used to set the data low to 0. Finally, if the data values are all negative, then SA_MAX will be used to set the data high value to 0.

As an alternative, the FORCE_ZERO parameter can be set to 1 to accomplish the same result. Historically, this parameter was introduced before the SA_MIN and SA_MAX parameters when the need to insure that 0 was included in the axis range was encountered. FORCE_ZERO remains as a option even after the SA_MIN and SA_MAX parameters were added for those programs already using the parameter, and as an easy way to use one parameter to accomplish this goal. Additionally, the parameter is a backup method if either one or both of the SA_MIN or SA_MAX parameters are used for specifying a desired axis minimum or maximum value.

ADDITIONAL EXAMPLES

Three sets of figures follow to show an additional comparison of the macro's range with the default SAS/GRAPH range and the simple one-tenth increment range as well as demonstrate differences in the macro produced range produced by use of the optional parameters.

Figures: A B C

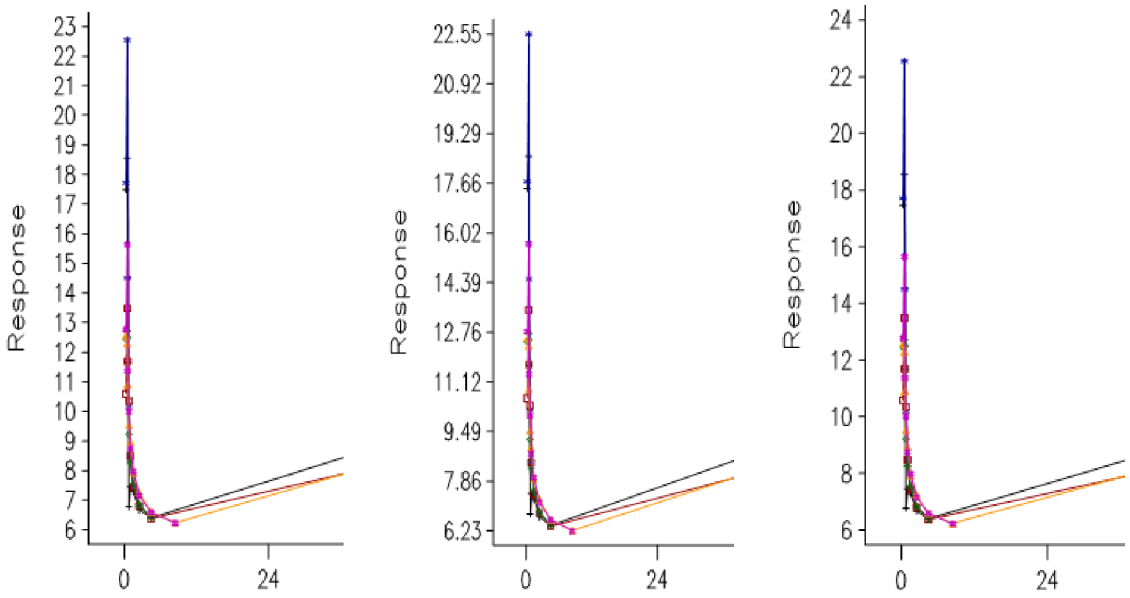

Figure 3 shows the differences in the Response axis min, max and increment using the SAS/GRAPH default (A), the min/max and one-tenth increment method (B) and the macro calculated values (C) for a medium data range.

Figures: A B C

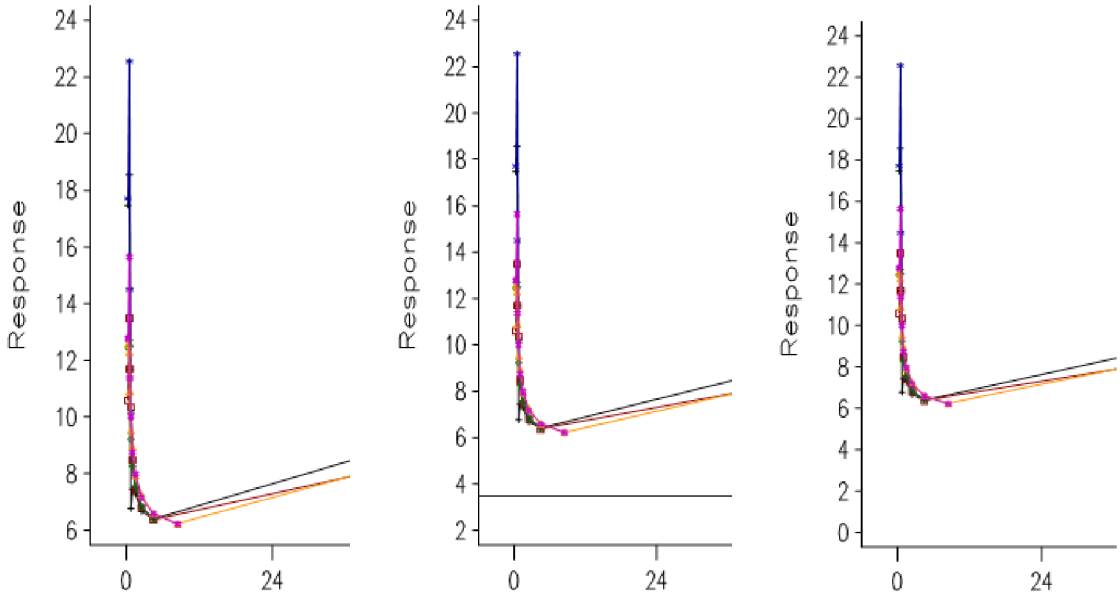

Figure 4 shows the differences in min, max and increment values returned by the macro with no optional parameters specified (A), with sa_min=3.5 specified to insure that the reference line at 3.5 is visible (B) and with force_zero=1 specified (C).

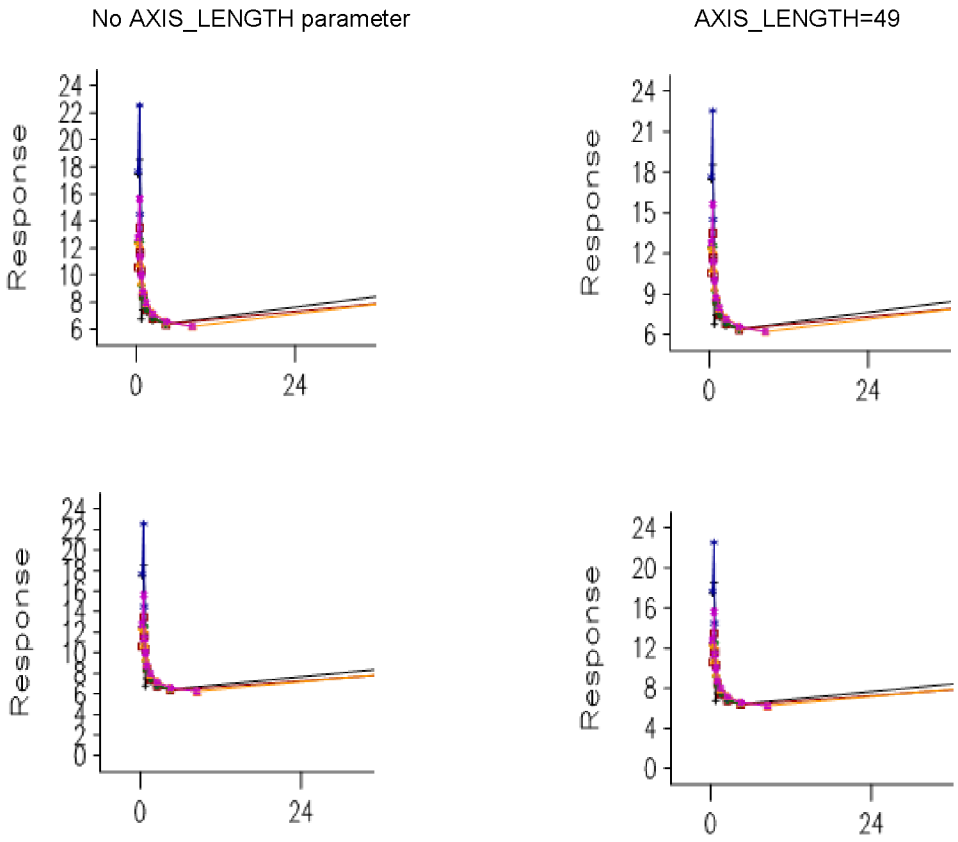

Figure 5 gives two examples of the benefit of using the AXIS_LENGTH parameter. In all four examples, the LENGTH option of the AXIS statement was set to 25 pct and the macro was used to set the range. In the right column, the AXIS_LENGTH macro parameter was set to 49 to trigger the production of fewer increments than the default in the graphs.

CONCLUSION

This macro was developed to solve a particular problem involving creating a large set of graphs within a very short turn-around period. The situation required setting the axis ranges automatically and the client was not satisfied with using SAS/GRAPH's default aixs range and increment.

In most cases, SAS/GRAPH's default range and increment calculations are sufficient and, often, are very similar to the range and increment produced by this paper's macro. However, if you encounter a situation where 'most of the time' is considered unacceptable or have a delivery time frame that does not allow for manual setting of the AXIS ORDER option, then I recommend giving this macro a try.

The time savings in the original study easily justified the development time and I have continued to use the macro for most graphs that I program. In almost all cases, the results have been very satisfactory. I added the SA_MIN and SA_MAX parameters for a study which required reference lines which were not always encompassed by the data. However, as always, there is room for improvement. I have included the code in the next section as a starting point. Adding and/or renaming parameters and tweaking the calculation of the standard increments are the main areas that could be updated to make the macro best suit your needs.

MACRO CODE

```
%macro set_axis_minmaxincrement(ds=,
                                axisvar=,
                                axis_length        = 51,
                                sa_min             = 999999,
                                sa_max             = -999999,
                                returned_min       = axis_min,
                                returned_max       = axis_max,
                                returned_increment = axis_increment,
                                force_zero         = 0
                               );

    %global &returned_min &returned_max &returned_increment;

    <Parameter checks go here. >

    /* Find the high and low values. Note: a data step was used versus a proc  */
    /* to allow the application of the option parameters, if specified.        */
    proc sort data=&ds out=sortlb(keep=&axisvar);
        by &axisvar;
        where &axisvar ne .;
    run;
    data axisdata(keep=low high);
        retain low 0;
        set sortlb end=eof;
        by &axisvar;
        if _n_ = 1 then low = &axisvar;
        if eof then do;
            high = &axisvar;
            if &sa_min ^=  999999 and &sa_min < low  then low  = &sa_min;
            if &sa_max ^= -999999 and &sa_max > high then high = &sa_max;
            %if &force_zero = 1 %then %do;
                if low > 0 then low = 0;
                else if high < 0 then high = 0;
            %end;
            output;
        end;
    run;


    data axisdata;
        set axisdata;
        /* insure that high is greater than low */
        if high <= low then do;
            if abs(low) <= 1 then high = low + 1;
            else                  high = low + 10;
        end;

        /* Calculate the conversion unit to transform the standard range to      */
        /* include the actual range. This value is used to convert the standard */
        /* to the actual increment for the actual range.                         */
        axisrange = high - low;
        /* Ranges of less than 1 */
        if axisrange <= 6 then do;
            check = 6;
            conversion_unit = 0.1;
            do until (axisrange > check);
                check = check / 10;
                if axisrange <= check then conversion_unit = conversion_unit / 10;
            end;
        end;
```

```
        /* Ranges of 1 or greater */
        else do;
            check = 60;
            conversion_unit = 1.0;
            do while (axisrange > check);
                check = check * 10;
                conversion_unit = conversion_unit * 10;
            end;
        end;
        /* standardize the range to lie between 6 to 60 */
        unit_range = axisrange / conversion_unit;

        /* Set the increment based on the unitized range */
        /* 'Long' axis, 8 - 12 increments                */
        %if &axis_length > 50 %then %do;
            if      unit_range < 12 then axisinc = 1   * conversion_unit;
            else if unit_range < 24 then axisinc = 2   * conversion_unit;
            else if unit_range < 30 then axisinc = 2.5 * conversion_unit;
            else                         axisinc = 5   * conversion_unit;
        %end;
        /* Otherwise, 'short' axis, 4 - 6 increments */
        %else %do;
            if      unit_range < 12 then axisinc = 2  * conversion_unit;
            else if unit_range < 18 then axisinc = 3  * conversion_unit;
            else if unit_range < 24 then axisinc = 4  * conversion_unit;
            else if unit_range < 30 then axisinc = 5  * conversion_unit;
            else                         axisinc = 10 * conversion_unit;
        %end;

        /* Round the min's value to match the increment; if the number is   */
        /* rounded up so that it becomes larger than the lowest data value, */
        /* decrease the min by one increment.                               */
        axislow = round(low, axisinc);
        if axislow > low then axislow = axislow - axisinc;
        /* Round the max; if the number is rounded down,   */
        /* increase the max by one increment.              */
        axishigh = round(high, axisinc);
        if axishigh < high then axishigh = axishigh + axisinc;

        /* put the values into the global macro variables */
        call symput("&returned_min",compress(put(axislow, best.)));
        call symput("&returned_max",compress(put(axishigh, best.)));
        call symput("&returned_increment",compress(put(axisinc, best.)));
    run;
%mend set_axis_minmaxincrement;
```

CONTACT INFORMATION

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

Name:	Rick Edwards
Enterprise:	PPD, Inc.
Address:	929 North Front Street
City, State ZIP:	Wilmington, NC 28401
Work Phone:	910-558-7249
Fax:	919-654-0711
E-mail:	richard.edwards@ppdi.com