

Semi log plots - Getting the axis tick mark labels in expanded Log₁₀ scale

Neha Mohan, i3 Statprobe (Ingenix Pharmaceutical Services), Mumbai, India

ABSTRACT

In Semi-log plots, one axis is plotted on a logarithmic scale while the other on a linear scale (here the actual data is plotted on a log scale and not the log transformed data) - the advantage being that it can bring out features in the data that would not easily be seen if both the axes had been plotted linearly. When such a semi-log plot is required to be created in SAS® using Log₁₀ scale, we usually use the LOGBASE and LOGSTYLE options in the axis statement. If the data points are either too small or too large, the resultant log axis that SAS generates is sometimes presented in exponential form having tick mark labels such as '1.00E-02', '1.00E-04', etc. However, if the display is required to be in the expanded form instead i.e. tick mark labels such as '0.001', '0.00001', etc., the user has to then define these labels in the AXIS statement by means of ORDER and/or VALUE options. In cases when the data is likely to change or when multiple graphs are created using the same program, defining axis scale for each data re-refresh and/or for all the graphs in case of multiple graphs can be tedious and inefficient. This article describes an approach, used in one such practical situation from a clinical trial, which automatically determines the optimized way of displaying the expanded log₁₀ scale on the graph axis.

INTRODUCTION

Practically speaking, if it is required to present data with very small or very large values graphically, on a log₁₀ transformed y-axis and a linear x-axis, due to the nature of data, SAS may automatically collapse the labels of the y-axis tick marks to present them in the exponential form (Refer Figure 1). However, if the requirement is to have the tick marks in an expanded form (Refer Figure 2) the labels will have to be manually set in the code. This approach would be inefficient if more data is expected to be added for each data re-refresh and/or if several graphs are to be generated from the same code. Hence the need arises to have a solution that would automatically set the expanded log₁₀ scale tick marks based on the actual data range axis scale.

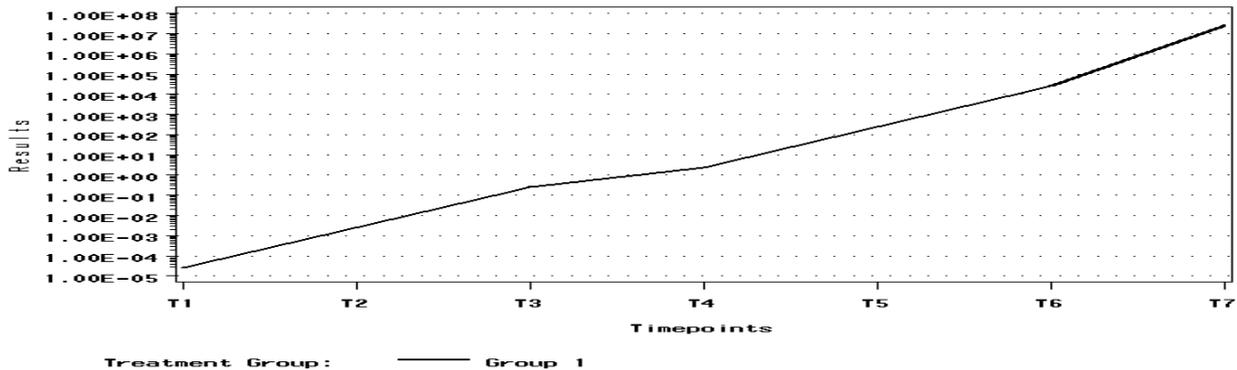


Figure 1: Collapsed Y-axis

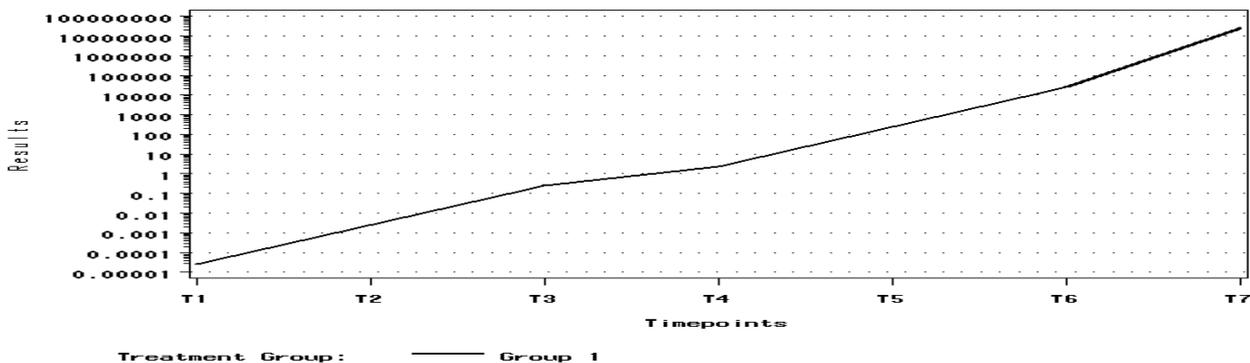


Figure 2: Expanded Y-axis

APPROACH

The following three fold approach can be used (Please refer the code provided on page 3 for derivations as well as usage of the macro variables described hereafter):

- Identify the range of the actual data for each individual graph
- Map the range to a \log_{10} scale
- Create individual tick mark labels in expanded form and assign the values to macro variables. Then use these macro variables in ORDER and VALUE options while defining the axis

IDENTIFYING THE RANGE OF DATA POINTS:

To begin with, the maximum and the minimum values of the variable being plotted need to be identified and assigned to macro variables, say MAX and MIN respectively.

MAPPING THE RANGE TO \log_{10} SCALE:

Since the actual data is to be plotted on a \log_{10} scale (and not the logarithmic values of the actual data), a range on \log_{10} scale corresponding to the actual data range needs to be identified. This can be achieved using the following logic:

For any number that is less than 1, its corresponding value on a \log_{10} scale depends on the number of places the decimal needs to be shifted in order to convert it to a value greater than or equal to 1. Whereas, for a number that is greater than 1, it is the number of digits before the decimal.

For example, consider the following dummy data of results for one treatment on 7 timepoints –

```
DATA test;
  INPUT TRT TPT RES;
  DATALINES;
  1 1 0.0000247
  1 2 0.00247
  1 3 0.247
  1 4 2.47
  1 5 247
  1 6 24789
  1 7 24789111
  ;
RUN;
```

In this example, the data has a range of 0.0000247 to 24789111 and hence the corresponding range on the \log_{10} scale will be 10^{-5} to 10^8 (i.e. 0.01 to 100000000).

CREATING LABELS FOR EACH TICK MARK

Now that the range on the \log_{10} scale is identified, the next step is to get the intermittent tick mark labels in expanded form. For a range on a \log_{10} scale, the number of intermittent tick marks required is usually the sum of the number of zeros in both the minimum and maximum values of the range. For a number less than 1, consider the number of zeros after the decimal and before 1, whereas, for a number greater than 1, consider the number of zeros after 1. So, using the dummy data above, for a range of 0.00001 to 100000000, the number of intermittent tick marks required will be 12 (4 + 8).

Next, using this count, each individual tick mark (including the lower and upper range values, i.e. the bounds) needs to be created and assigned to macro variables, say TCK_i (i = 1 to 14 in case of dummy data, where 14 comprises of 12 intermittent tick marks and 2 bounds). In addition to these macro variables, we also need another macro variable (say ORD2) that will hold the order in which the tick marks are to be presented on the graph. These macro variables can then be used in the VALUE and ORDER options while defining the y-axis to get the output in the required format.

Following will be the values for the aforementioned macro variables based on the dummy data:

```
&TCK1 = 0.00001          /** Minimum **/
&TCK2 = 0.0001
&TCK3 = 0.001
&TCK4 = 0.01
&TCK5 = 0.1
&TCK6 = 1
&TCK7 = 10
&TCK8 = 100
&TCK9 = 1000
&TCK10 = 10000
&TCK11 = 100000
&TCK12 = 1000000
&TCK13 = 10000000
&TCK14 = 100000000     /** Maximum **/
&ORD2 = 0.000001 0.00001 0.0001 0.001 0.01 0.1 1 10 100 1000 10000 100000 1000000
          10000000 100000000     /** Order of tick marks **/
```

SAS CODE:

The below code is specific to the dummy data used in the paper and may need to be adapted as per the requirement while re-using the code.

```

%MACRO plot;

/*****/
* Identifying the data range and assigning the values to macro  *;
* variables MIN and MAX                                     *;
/*****/

%LET min=.;
%LET max=.;
PROC SQL NOPRINT;
  SELECT MIN(res)INTO: min FROM test WHERE res^=.;
  SELECT MAX(res)INTO: max FROM test WHERE res^=.;
QUIT;

%PUT &min, &max;

/*****/
* Identifying the number of tick marks required             *;
/*****/

DATA inter1;
  min1=SCAN(&min,1, '.');
  min2=SCAN(&min,2, '.');
  max1=SCAN(&max,1, '.');
  max2=SCAN(&max,2, '.');
  l1=LENGTH(COMPRESS(min2))- LENGTH(COMPRESS(PUT(INPUT(min2,best.),best.)))+1;
  CALL SYMPUT ('l1',COMPRESS('Z' || PUT(l1,best.) || ' '));
RUN;

DATA inter2;
  SET inter1;
  IF INPUT(min1,best.)>0 THEN DO;
    CALL SYMPUT ('v1',INPUT(10*LENGTH(min1),best.));
  END;
  ELSE DO;
    CALL SYMPUT ('v1',INPUT('0.' || COMPRESS(PUT(1,&l1)),best.));
  END;
  IF INPUT(max1,best.)>0 THEN DO;
    CALL SYMPUT ('v2',INPUT(10**LENGTH(COMPRESS(max1)),best.));
  END;
RUN;

DATA inter3;
  LENGTH a $500;
  SET inter2;
  a=COMPRESS(PUT(&v1,best.) || '@');
  p=l1+INPUT(LENGTH(COMPRESS(max1)),best.);
  CALL SYMPUT ('p',COMPRESS(p));
RUN;

/*****/
* 1] Creating the labels for the tick marks in expanded form and assigning the values  *;
* to macro variables TCKi                                                             *;
* 2] Creating macro variable ORD2 to assign the order in which tick marks are to be   *;
* presented on the graph                                                               *;
/*****/

DATA inter4;
  SET inter3;
  %DO i = 1 %TO &p;
    pwr=INPUT(&v1,best.)*10**&i.;

```

```

        x= COMPRESS(PUT(pwr,best.));
        a=COMPRESS(a)||COMPRESS(x)||'@';
%END;
CALL SYMPUT ('ORD1',COMPRESS(a));
CALL SYMPUT ('ORD2',trim(left(translate(a,' ','@'))));
%DO h = 1 % TO %EVAL(&p+1);
        CALL SYMPUT ("tck&h",COMPRESS(SCAN(a,&h,'@')));
%END;
RUN;

%DO h = 1 % TO % EVAL (&p+1);
        %PUT &&tck&h;
%END;
% PUT &v1, &v2, &ord1, &ord2, &p;

/*****
* Assigning symbols, defining axes and legends for the graphs *
*****/

SYMBOL1 INTERPOL=j WIDTH=2 C=black VALUE=none LINE=1;

AXIS1 LABEL=(r=0 a=90 h=1.0 "Results")
        ORIGIN=(1.3in,2.25in)
        LOGBASE=10 LOGSTYLE=expand
        WIDTH=1 LENGTH = 3in
        ORDER=(&ord2.)          /** Used macro variable ORD2 here to define the order. **/
        VALUE=( %DO h = 1 %TO %EVAL(&p+1);      /** Used macro variables TCKi here to assign **/
                %STR(TICK=&h "&&tck&h" ) /** labels to the tick marks. **/
        %END;
        )
;

AXIS2 LABEL=(a=0 h=1.0 "Timepoints") MINOR=none
        LENGTH = 6in ORIGIN =(1.3in,2.25in)
        VALUE =( tick=1 'T1'
                tick=2 'T2'
                tick=3 'T3'
                tick=4 'T4'
                tick=5 'T5'
                tick=6 'T6'
                tick=7 'T7'
                tick=8 ' ')
;

LEGEND1 NOFRAME LABEL=("Treatment Group: ")
        POSITION = (center outside)
        ORIGIN = (0.9in,1.3in)
        VALUE = ('Group 1' );

PROC GPLOT DATA=test ;
        PLOT res*tpt =trt / HAXIS = AXIS2
                            VAXIS = AXIS1
                            AUTOVREF LVREF=34
                            FRAME LEGEND=LEGEND1;

RUN;
QUIT;

GOPTIONS RESET = ALL;

%MEND plot;

```

CONCLUSION

The logic described in the paper proved very useful in arriving at the required log axis for multiple graphs and with minimal re-work for each data re-fresh. I hope this information will be helpful to other programmers as well who work with similar data and requirements. The described logic is relevant to semi-log plots (base 10) and may be further generalized and extended to log-log plots or any other plots that require actual data values to be plotted on \log_{10} scale.

ACKNOWLEDGMENTS

I thank all my colleagues for guiding this work and carefully reviewing the paper with comments and suggestions.

CONTACT INFORMATION

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

Neha Mohan
i3 Statprobe (Ingenix Pharmaceutical Services),
7th Floor, Corporate Center,
Opp. To VITS Hotel,
Andheri-Kurla Road,
Andheri (E)- 400059
Mumbai, India
Work Phone: +91-22-30554000
E- mail: neha.mohan@i3global.com ; neha.nm@gmail.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.