ABSTRACT
Axes are important components of graphs. Axes are usually the simple range frames with increment units, but can be more informative and in special formats. The SAS graphic programming facilities provide options to control the expression of axes of graphs. Sometimes, building an axis that meets special requirements is challenging. This article provides some how-to tips with practical examples, in which specific graphical axes are required. SAS codes for the technical tricks are also included.

INTRODUCTION
Defining axes for SAS graphs is the common task for SAS users. We may know the general ways of defining graphical axes, however, graphical axes can be more informative and cater to more specific needs. For example, the information displayed with axes may be beyond the only line of axis scale; the tick-mark values of the axis may be expressed with special formats; the axis may need to be broken to fit the large value jump; the description of the tick mark values can be displayed with specific formats; there could be more vertical axes on a two dimensional graph that contains variables measured in different scales.

Pretty often SAS programmers, especially the new users, may feel that it is rather complicated to code a graphic job. It is true because SAS uses program code to control the displays of the graphical elements rather than using interactive editing tool. Programmers need an abstractive mind to link the codes with their actual expressions on the picture. In this article, five cases derived from practical works are summarized with code examples and corresponding outputs. The key technical issues are discussed. The author tries to bridge the drab codes with their graphical expressions to demonstrate the function of the code.

CASE 1. MORE INFORMATIVE AXIS (Multiple Line Tick Mark Values)
Usually there is just one line for axis major tick mark values (for example, range with increment or labels of axis variable). However, sometimes there is information that is associated with each major tick mark so the texts showing their values may be in more than one line. The information of patient number at risk at each follow-up time point for a graphical survival analysis may be a perfect example. At the beginning of each follow-up time point, there are different numbers of patients by treatment group. These numbers are directly related with the time points. Many flexible graphical tools do this sort of job with manual annotation. The patient numbers are manually entered after the major survival curve graph is created. The positions of the numbers are to be adjusted to align with the time points. In the National Cancer Institute of Canada, Clinical Trials Group, we established a standard generic macro to automatically create the survival graphs with patients at risk annotation. The macro can detect the number of treatment groups and determine the scales of the time axis. A frequency data set is then created based on how many time points are determined. The patient numbers by treatment group at each time point are converted to a two-dimension series of macro variables and applied to the value option of the axis statement by macro looping.

To clearly illustrate the process, here we simplify the program to keep only the basic code. The code is for annotating the extra information related with axis major tick marks using the multiple lines tick mark value definition. Let us focus on a specific situation: suppose we have six increments of time points and four treatment groups, and we want to make the patient at risk numbers by treatment group shown at each follow-up time points. The code example for defining axis statement is as follows:
It may be noticed that in this section of axis statement, there are no label and order options. Both are buried in the value option. In this statement, each t (of t1 to t7) is followed by multiple line tick mark values. The number/text in the quotation marks are the information to be displayed. The empty quotation marks are nothing else but just space holders. Although the labels are in horizontal order for each tick, they actually displayed vertically in the graph. For example at the first column, “0” comes at first row, and then comes an empty line (for saving space for axis label “Time-Months” at the fourth column), then consequently ‘179’, ‘180’, ‘119’ and ‘117’ at rows 3, 4, 5, and 6, are shown respectively.

The output of a real data example looks like:

The labels for treatment groups at the left-bottom corner are annotated with an annotation data set.
CASE 2. FORMATTED DISPLAY OF TICK MARK VALUES

Although commonly numeric variable is used to define the range of an axis or the abbreviated terms are directly used as the tick mark values of an axis, the tick mark values can actually be displayed in specific formats. The examples include, but are not limited to, currency format for money, full term for short abbreviation, and text date format for numeric date. An example from clinical trial practice is used here to demonstrate the idea. Accrual graph is a useful way to show how patients are accumulated along with time and how the real accrual departs from the planned expectation. The horizontal axis represents the SAS numeric date. However, it does not make sense to show the date information as numbers of days from SAS date cut-off. To show the numeric date in text format, just apply the desired format in the plot procedure:

```plaintext
data _anno1;
   length function $ 8  position $ 1 text $15;
   xsys = '3'; ysys = '4'; position = '6';
   function='label'; style = "hwcgm001";
   x=2; y=7.25; size = 1.2;  text=UTF at Risk';
output;
   x=2; y=y-1.5; size = 1;  text="ARM 1"; output;
   x=2; y=y-1.5; size = 1;  text="ARM 2"; output;
   x=2; y=y-1.5; size = 1;  text="ARM 3"; output;
   x=2; y=y-1.5; size = 1;  text="ARM 4"; output;
run;
```

The format statement in the procedure enables the program to display the x-axis tick mark values in formatted text date (see figure 2.). If we don’t like the format 'monyy5.', it is easy to change to other text date formats (such as 'monyy7.' or 'date7.'). In the same way we can apply format to replace the original brief signs/values with full descriptions whenever necessary.
CASE 3. CHANGE ANGLE IF TICK MARK TEXT IS LONG

Usually the tick mark values are in brief terms so that they can be displayed on the axis without overlapping or being forced into vertical direction. This does not mean that we are unable to display the long value terms if necessary. The long value terms can be displayed in an angled text format.

Suppose we want to show the frequencies of treatment responses. We can show the responses in short terms (for example 'CR' = 'Complete Remission', 'PD' = 'Progression Disease', etc.), and then use footnotes to interpret the short terms. However, we can also display the full-terms with the angled tick mark texts using the codes showing at right:

The full response group identifications are shown as texts in 45° angle on the axis. The r= option rotates the letters to an appropriate degree (figure 3.). The line angle, rotate angle, text height, text font, and text color are flexible to change.

CASE 4. USE DOUBLE VERTICAL AXES

In one graph, variables with different value systems may need to be shown simultaneously. For example, in epidemiological studies, the ecological correlation between disease incidences and some environmental risk factor measurements may be displayed on one graph to demonstrate the coherences of their trends. The measurements of disease incidence and environmental risk factor, however, are totally different. We can analogue this situation to other examples: such as displaying price trends in different currency systems, showing values of laboratory examinations derived from different criteria, and so on. If the same scale is used, the tremendous difference between the value systems may split the values with an unreasonable gap and make the graph look strange. It is also hard to define the vertical axis because of the different units representing the systems. In order to offset this problem, we can use another vertical axis representing a different value system. The program for doing this includes an extra axis definition statement and a plot2 statement within the plot procedure:
*** Besides x-axis and y-axis, an extra axis is defined ***;
axis3 order=100 to 500 by 100
label=(h=1.3 a=90 "Tobacco Consumption (Tons)"
color=red w=10 minor=(number=5);

proc gplot data=tobacco;
plot lcincid*year / vaxis=axis1 autovref haxis=axis2;
plot2 tobacco*year / vaxis=axis3;
title 'Tobacco Consumption and Lung Cancer Incidence in XXX';
run; quit;

With hypothetical data, the amount of tobacco consumption is related to the incidence of lung cancer. As seen in the output graph, lung cancer incidence (in solid line) and tobacco consumption (in dotted line) are shown with separate lines and axes. This graph demonstrates a clear coherence between the tobacco consuming trend and the trend of lung cancer incidence in the selected country.

CASE 5. BREAK THE VERTICAL AXIS

Following the last problem, if the case is that the value-unit measured with different systems is the same but with remarkably different sensitivities, it is inappropriate to use different vertical axes but better to use a unique one. In this situation, there may be a big gap on the graph between the values derived from different systems. There is a special technique to solve the problem: breaking the vertical axis. To do this, three steps are followed:

1) Define an uneven order axis: see order option:
axis1 order=(0 to 50 by 10, 200, 400, 2000 to 5000 by 1000) minor=none width=2;
data box;
length function style color $8;
retain xsys '5' ysys '2' when 'a' style 'solid';
function='move'; x=1; y=80; output;
function='bar'; x=10; y=1000; color='white'; output;
color='black'; size=2;
function='move'; xsys='1'; x=0; ysys='2'; y=80; output;
function='draw'; xsys='B'; ysys='B'; x=+4; y=+6; output;
function='draw'; xsys='B'; ysys='B'; x=+-8; y=+5; output;
function='draw'; xsys='1'; x=0; ysys='2'; y=1000;
output;
run;

Figure 4. Double vertical axes
The technical trick lies in creating a correct annotation data set. The coder has to be clear about what do the values in the annotation data mean. Basically, the functions tell the program what to do, and the x and y define the points on the graph where the actions happen. For example, the code makes a white bar from 1, 80 to 10, 1000 on the uneven vertical axis to cover the area, and draws a zigzag line with the values of xsys, ysys, x and y.

**DISCUSSION**
Axes are elements as important for a wagon as for a graph. SAS is actually very powerful in defining graphical axes, but needs programming skills. Besides the practical examples used in this article, there are other way to control the expression of axes, such as log-scale axis and axis with special tick mark values. With some simple options or annotation codes, we can build more informative and specially formatted graphical axes to meet the specific requirements. It is hoped that the examples and the programming hints summarized in this article may help the primary level programmers to increase their interests and skills in creating variant SAS graphs with specially required axes.

The full program that creates all the examples graphs can be provided per request.

**REFERENCES**


**CONTACT INFORMATION**
Don (Dongguang) Li, PhD, MPH, Senior Biostatistician
National Cancer Institute of Canada - Clinical Trials Group
10 Stuart Street, Queen’s University, Kingston, Canada, K7L 3N6
Work Phone: 613 533 6000 Ext. 78337
Email: dli@ctg.queensu.ca