A MACRO SYSTEM FOR CALLING PROC GPLOT FOR GENERAL USAGE BUT ALSO HAVING A MAJOR APPLICATION FOR PROCS LIFETEST AND PHREG PLOTS
Hugh Geary, Trilogy Consulting Corporation, Waukegan, IL

Abstract
This system enables the user to specify titles, labels, footnotes, symbols, input and output data set names, annotations and various control variables in the arguments of a calling statement without having to write the title, footnote and axis statements, etc., for PROC GPLOT each time a graph needs to be produced. The system consists of a general macro for setting up the call to PROC GPLOT and a special purpose macro, called by the general macro, to compute the coordinates and annotations (which in our case are for the Lifetest/Phreg graphs but could be for any application.)

The application to PROCs LIFETEST and PHREG plots the survival data for one, two, three or more groups while depicting censoring. Extra annotations have been computed to display the number-at-risk at each tic mark along the horizontal axis. Also in the general macro a moveable window has been created in which pertinent statistics can be prominently displayed.

The chief advantages of this macro are the ease of creating a large number of similar graphs and the ease of building a new graph by drawing upon a number of refined techniques already embodied in the code.

System Description
This system of two macros, gplot3 and gpoint3, has grown through usage, being continually generalized to include more features. Figure 1, on the next page, shows a Kaplan-Meier Survival graph. It was produced by using an output data set from Proc LIFETEST & PHREG. Gpoint3 was written specifically for processing LIFETEST output. Gpoint3 could be replaced by other macros to produce other types of graphs.

The main program, gplot3, creates titles, footnotes, notes, axis labels, a panel to display certain statistics, such as, hazard ratio, 95% C-I, p-values, etc., and a two or three line display immediately beneath the horizontal axis to show the number-at-risk (patients) at various points in time.

It provides a means to put small moveable labels by each graph and a means to raise and lower the horizontal axis.

It also has the call to PROCs GPLOT and GREPLAY, a call to the system to send the completed graph to a printer and another to delete an unneeded data set.

The other macro, gpoint3, produces two data sets, FINAL, for the plotting of the graphs, and AN, for the number-at-risk display annotations.

The program had the "3" added to its name when it was expanded to plot 3 or more graphs. It can plot 2 or just 1 graph.

Kaplan-Meier - Important Special Case
For the Kaplan-Meier graphs, four are actually being plotted besides the dashed horizontal line across the middle. Two of the four are for censored patients one for each drug and two are for the uncensored patients, one for each drug.

The Number of Patients at risk comes from Proc LIFETEST and is calculated by an algorithm in Gpoint3.mac.

Example
Figure 1, below, is a graph, an example of output from this system, and the macro call that produced it.
Figure 1
Time to Weight Loss of 5% from Baseline
Drug A vs. Drug B Patients

The Macro Call Producing the Above Plot

The following parameters are for display in the statistics "box":
- Character string
- Hazard Ratio value
- Confidence interval bounds
- Logrank & Wilcoxon p-values
- Footnote: drug name & number of patients

Zg plots ( 
GL_vtl, 
disklib=[gcoxy.epi],GL_vtl, 
day2, 
col, 
tc, 
Figure 1, 
Title1, 
Title2, 
Title3, 
Title4,Title5 & Title6

Drug A, Drug B, 
0.689, 
0.407, 1.166, 
= 0.160, = 0.253,

Drug A,74, 
Drug B,76, 
Proportion of Patients with , 
Proportion of Patients with, 
Time to 5% Weight Loss, 
Time to 5% Weight Loss (Days), 
400, 
50, 
print=50, 
box=BP,

up=4, 
n_A_Razz);
Key Features

Titles:

Up to 6 titles may be specified. They are automatically positioned by PROC GPLOT.
A typical title line would be:

```
title f=mapf h=1 "title1";
```

where title1 is a macro parameter passed by the macro call beneath Figure 1;

Superscript or subscript capability can be added to this program by installing code to raise/lower and change the size of a character string. Such code is used in a footnote.

Statistics Display Panel:

Six parameters are for character strings and statistics that need to accompany the graph. The macro automatically positions it under the last title line which means it will be higher if fewer titles are used. If a graph happens to pass through it, it can be moved up or down with a parameter named 'up'. It can also be split into a top half and a bottom half. A parameter named 'box' can be given values, RP, BP or NO to cause the panel to, respectively, appear in Regular size Print (h=.7), Big Print (h=1) or NOT to be printed at all.

In Figure 1 this has been accomplished by using the remaining title lines: title7-title10, but it could also be done using note statements.

Spikes:

GEPlot automatically positions the axes after it has allocated space for titles and footnotes. It does not leave space for notes. But if you need space for notes, say between the horizontal axis and the footnote below it, extra space there can be forced by using 'move's to create a spike in the footnote statement with:

```
m=(+0,+u) " " m=(+0,-u)
```

where u is a number. Determine u by trial and error. Similarly, space between the graph and the titles can be increased with a downward spike in the last title.

In this macro a spike is used in footnote 1 to allow room for the At-Risk annotations.

The spike was especially necessary because when the tick mark labels were suppressed, GELOT allocated that space for use by the plot.

Footnotes:

This macro uses optionally two or three footnotes. Parameters provide character strings to be used in the footnotes to identify symbols that are being used in the various graphs. These symbols have to be created in the footnote, a feat which sometimes requires the use of the draw, move and other commands. For example,

```
draw=(2.8,.95,9.6,.95,2.8,.90,9.6,.90)
```

creates a heavy line (a very narrow rectangle).

A footnote statement can optionally be put in use by using macro language:

```
if &cond Ithen Ido;
  Footnote statement;
Iend;
```

where &cond is some parameter passed to the macro. Thus the parameter, cond, determines whether the footnote will be displayed or not.

Superscripts:

One means of creating a superscript or exponent in a footnote (or title) is to put a marker, say, **, in the text of the footnote, just ahead of the character string to be displayed as a superscript. Then use macro language and macro string functions to find this marker and the string following it. In the footnote text, use a move up function, reduce the height or h value, print the superscript, then move back down and reinstate the original value of h.

```
let cc=index(&label2,**);
if &cc>0 Ithen Ido;
  let dd=substr(&label2,&cc+3);
  let ee=substr(&label2,&cc+2,1);
  let label2=substr(&label1,1,&cc -1)
  " m=(+0,.5) h=.6 "see" m=(+0,-.5) h=1 "&dd;
Iend;
```

In the above example the exponent is to be a single character. It converts the string **x to an exponent x.
For the horizontal axis, three parameters, 
\( x_0, x_{\text{max}} \) and \( x_{\text{incr}} \) specify the beginning 
value, the endpoint value and the step 
size. Similarly for the vertical axis, 
\( y_{\text{max}} \) and \( x_{\text{incr}} \) specify the maximum value 
and step size (it begins at zero.) The 
use of \( x_{\text{max}}, x_0 \) and \( y_{\text{max}} \) provides you 
with scale control for the axes.

**Axis-Value option:**

The axis value option supplies tick mark 
labels. If you want to put something else 
where these labels are positioned by GPLOT 
you may use the option:

\[
\text{value=none (on the axis statement)}
\]

and build your own text in this space by 
using the annotation data set.

In Figure 1 that space is needed for the 
number-at-risk display and yet the tick 
mark labels still are needed. Consequently, 
both the number-at-risk display and 
the tick mark labels are produced with the 
annotation data set.

**Two Line Axis Label:**

In the plot example the vertical axis label 
consists of two lines. To split the label 
two parts put a 

\[
\text{(Justify=Center) or (J=C)}
\]

between the two parts or else GPLOT will 
attempt to put them both on the same line. 
The full axis statement might then be:

**AXIS1**

\[
\text{VALUE = (F=SWISSL)}
\]

\[
\text{LABEL = (F=SWISSL A=90 R=0 "&LABEL1"\ )}
\]

\[
\text{UNQUOTE(&JUST) UNQUOTE("&LABEL1A")}
\]

\[
\text{ORDER = 0 TO &y_max by &y_incr}
\]

\[
\text{MINOR = (N=2)}
\]

\[
\text{OFFSET= (0)};
\]

where \( &\text{JUST} = \text{UNQUOTE(2str(J=C))} \) 

whenever the label is to be split and 
null when it is to be a single line.

**Optional Display of the Key to the Graphs**

**on Either the Footnote Line Or Small**

**Labels Placed Adjacent to the Actual**

**Plot Lines:**

Figure 1 above shows the key explained 
by the bottom footnote. Small moveable 
labels can be positioned beside the plot 
lines themselves to make them more easily 
identifiable though at greater cost to the 
programmer. One small character string 
can be placed beside each plot line by 
using a note with move functions before 
each label. Note statements must come 
after the PROC GPLOT statement and may 
have the following form:

\[
\text{Note f=zapf h=7 m=(&u1,&v1) "&lab1"}
\]

\[
\text{m=(&u2,&v2) "&lab2";
\]

where \((u1,v1)\) and \((u2,v2)\) specify the \(x\) & 
\(y\) coordinates of each label. The coor­
dinates must be found empirically. Usually 
three runs is sufficient to find good ones.

**Optional Printing of the Input Data:**

The parameter, Print, if given the value, 
yes, such as, 

\[
\text{Print=yes},
\]

in the parameter list will act as a switch 
to cause printing of the input data set to 
PROC GPLOT.

**The G_Point3 Macro:**

This macro computes the data set called, 
Final, which is given to PROC GPLOT and 
has the coordinates of all the points to 
be plotted. The plot statement used with 
PROC GPLOT here is:

\[
\text{plot survival*time=plotline}
\]

(with some options.) Plotline=1 for all of 
the number pairs, (time,survival), that 
go with the first plot. All of the records 
with plotline=2 comprise the 2nd plot, etc. 
The names, time and survival, have come 
from the outputs of Proc Lifetest and 
PHREG, but any data expressed in two coor­
dinates (appropriately scaled) can be 
plotted here.

The macro also produces the annotation 
data set, AN. Each record in this data 
set contains all of the information needed 
to put a character string somewhere on the 
graph – i.e. coordinates, size, font, 
color, function, text, etc.
Conclusions:

The highlights of a useful macro system have been summarized. A listing would be useful but too lengthy. Nevertheless, anyone wanting to build their own macro plotting system would not want to rework some existing system but may benefit from explanations of some essential points in another system.

Once the parameters have been decided upon and the macro logic written, a system can be used for many similar or nearly similar graphs with very little additional work.

For more information/comments contact:

Hugh Geary
Trilogy Consulting Corporation
630 Morning Street
Worthington, OH 43085
(614) 885-3164

*SAS is a trademark or registered trademark of SAS Institute Inc. in the USA and other countries.