

SOME STRATEGIES FOR EFFICIENTLY PRODUCING GRAPHS OF CLINICAL TRIALS DATA

Michael A. Riddle, Quintiles, Inc., Research Triangle Park NC

INTRODUCTION

Programmers working with clinical trials data face various challenges when required to produce graphical displays. Sometimes work begins on a project which is then put on hold for any number of reasons. The original programmer might change positions or leave the company. When work on the project resumes, a new programmer confronts the sometimes daunting choice of whether it will be more efficient to try to reproduce results using unfamiliar or unwieldy code, or to start from scratch.

This paper illustrates the flexibility of SAS/GRAPH® by producing a clinical trials graphic with two different methods. A real world situation is summarized, starting with code inherited from another programmer. An unexpected problem using this technique is related. Finally, an alternative, more straightforward method of producing the same graph is given.

METHOD 1 - 2 PLOT, 3 AXIS AND 8 SYMBOL STATEMENTS

Our company produced a number of graphs for a series of pharmacokinetics studies. Work began on preliminary data for the first studies two years ago, including the generation of several graphs of treatment group means. The production of additional figures stopped while further clinical trials were conducted, and the original programmer left the company. Two years later, we were then asked to resume work, using the final data for the studies.

Figure 1 shows a simple graph of four treatment group means with 95% confidence intervals, produced using the original programming code. PROC MEANS was used to generate the mean values and standard errors. The study population contains one treatment group of elderly patients ($n_{trt}=1$) and three groups of young subjects (n_{trt} in (3,4,5)).

```
proc means data=both;
class n_trt;
var nauc;
output out=nauc mean=mean stderr=se n=n;
run;
```

The confidence interval was obtained by manipulating the output data set from the PROC MEANS, using the TINV function. A dummy treatment variable was also created. This will be used as the 'z' in the plot statement to obtain the legend that SAS/GRAPH® automatically generates for $x*y=z$ plots, even though this example has treatment group as the only by-variable. The original code outputs three observations for each treatment, with CI set to the upper limit, the mean, and the lower limit. The CIU (upper) and CIL (lower) variables are used to facilitate the QC of the graph.

```
data nauc;
set nauc(where=(type=1));
dummy=n_trt; /* use dummy as 'Z' */
alpha=0.05;
DF=n-1;
t_crit=tinv((1-alpha/2),DF);
ciu=mean + (t_crit*se); ci=mean + (t_crit*se); output;
ciu=.; ci=mean; output;
cil=mean - (t_crit*se); ci=mean - (t_crit*se); output;
run;
```

Since it was desired to separate the one treatment group with elderly patients from the three young subject groups, the labels that SAS/Graph would automatically generate are suppressed (LABEL=NONE) in the axis2 statement. This will allow for the underlining of treatment groups, as well as eliminate the tick mark for $N_{TRT}=2$, which enables the space between the elderly and young subjects but does not represent a data value. Instead, labels are annotated.

```
data anno;
length function $8 text $18;
function="label"; size=1; xsys="2"; ysys="3";
style="triplex";

/* create first label line */
```

FIGURE 1: ORIGINAL CODE AS PRODUCED IN-HOUSE
Your Favorite Compound/ By Your Favorite Client
Protocol 123-4567

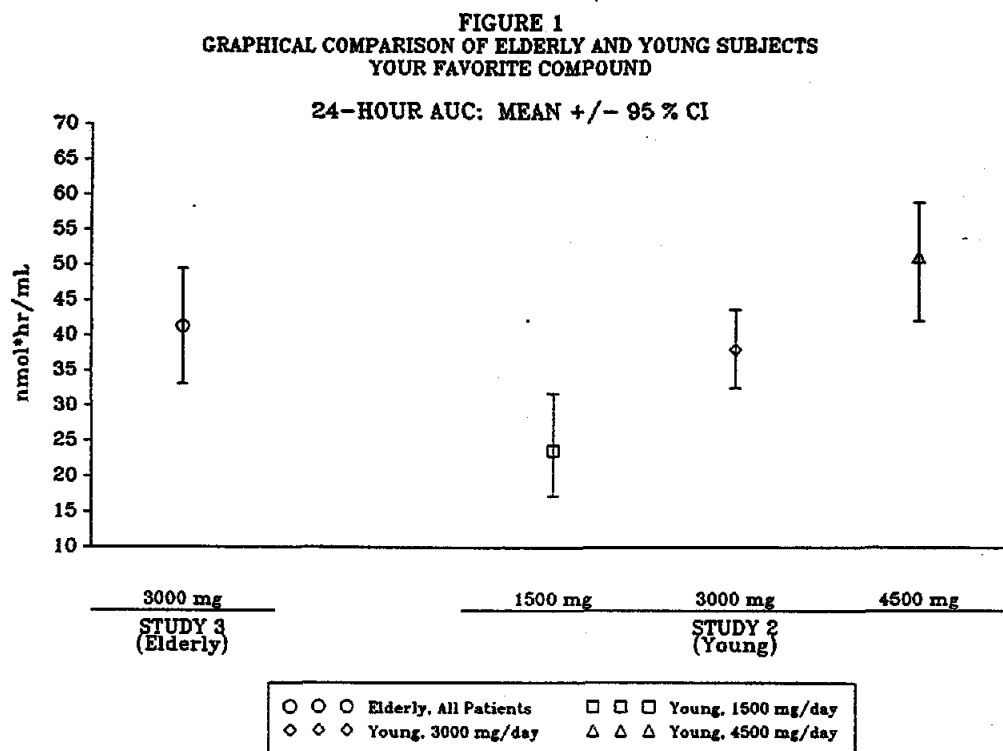
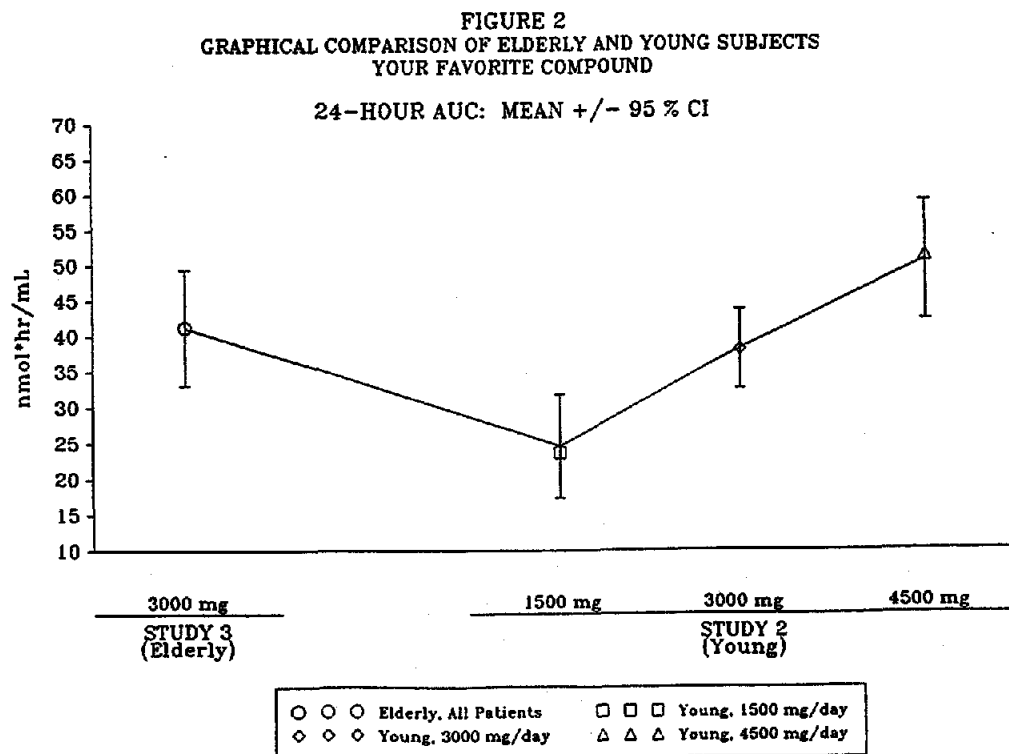


FIGURE 2: ORIGINAL CODE AS PRODUCED BY CLIENT
Your Favorite Compound/ By Your Favorite Client
Protocol 123-4567



```

x=1; y=26.5; text="3000"; output;
x=3; y=26.5; text="1500"; output;
x=4; y=26.5; text="3000"; output;
x=5; y=26.5; text="4500"; output;

/* underline to separate elderly/young patients */

function="move"; x=.5; y=25.2; output;
function="draw"; x=1.5; line=1; size=1; output;
function="move"; x=2.5; y=25.2; output;
function="draw"; x=5.5; line=1; size=1; output;

/* create second label line */

function="label"; size=.85;
x=1; y=24; text="STUDY 3"; output;
x=4; y=24; text="STUDY 2"; output;

/* create third label line */

x=1; y=22.5; text="(Elderly)"; output;
x=4; y=22.5; text="(Young)"; output;
run;

goptions ...

```

Now comes the interesting part. The original programmer decided to produce the graph using two separate plot statements, one to place the symbols showing the mean values, another to draw the confidence intervals. To do this, three axis statements are used, with axis1 as the horizontal axis in the first plot statement and axis3 as the second horizontal axis.

```

axis1
v=none
label=(r=0 a=90 h=.85 "nmol*hr/mL")
order= 10 to 70 by 5
value=(h=.85)
offset=(0,0)
minor=none;

axis 2
v=none
label=none
offset=(0,0)
order=(.5 to 5.5 by .5)
major=none
minor=none
length=5.5 in;

```

```

axis3
v=none
label=none
minor=none
major=none
order=10 to 70 by 5
style=0;

```

Note the use of style=0, which suppresses the axis line.

Next there are 8 symbol statements. The first four are used for the means in the first plot statement below, symbol5 through symbol8 for the confidence intervals in plot2.

```

symbol1 v=circle i=none h=1 l=1;
symbol2 v=square i=none h=1 l=1;
symbol3 v=diamond i=none h=1 l=1;
symbol4 v=triangle i=none h=1 l=1;

symbol5 v=point cv=black ci=white i=hilojt h=1;
symbol6 v=point cv=black ci=white i=hilojt h=1;
symbol7 v=point cv=black ci=white i=hilojt h=1;
symbol8 v=point cv=black ci=white i=hilojt h=1;

proc gplot data=nauc annotate=anno;
plot mean*dummy=n_trt/
      vaxis=axis1
      haxis=axis2
      legend=legend;

plot2 ci*dummy=n_trt/
      vaxis=axis3
      nolegend;

```

Probably the most unusual part of this code is the use of ci=white in combination with the specified interpol options in symbol5-symbol8. By specifying i=hilojt, SAS/GRAPH® produces a solid vertical line connecting the maximum and minimum values of the ci variable for each value of n_trt (i=hilo). Adding 'j' in effect suppresses the tick mark at the mean value, and 't' adds a top and bottom to the confidence interval line. Since it was not desired to connect the four means, using ci=white fakes this result as a white line is not visible on the paper. SAS/GRAPH® does produce a note in the log stating "Foreground color WHITE same as background. Part of your graph may not be visible." Note that you must specify cv=black using this method, as the value color (cv) would default to the line color (ci) if not specified.

The end result is a figure which shows the different treatment groups with a easily differentiated symbol. The symbol marks the mean value, and there is no undesired tick mark confusing the issue at the mean points. The legend automatically generated by the plot1 statement is very suitable for this figure.

```
legend position=(bottom center)
      origin=(30,6)
      label=none
      frame
      across=2
      value=(h=.65 "Elderly, ALC 3000 mg/day"
              "Young, ALC 1500 mg/day"
              "Young, ALC 3000 mg/day"
              "Young, ALC 4500 mg/day");
```

Note that the legend associated with the plot2 statement is suppressed.

Even though the above code is somewhat confusing and more than a little cumbersome, it did produce the desired results. A variety of other figures were produced for the series of studies, including several modeled after the Figure 1. After in-house work on the first study was complete, all figures were rerun with the device driver changed to one of those available to the client. The graphics stream files were all copied to diskette, and sent back to the client to be incorporated into their reports.

Although there were no problems with the majority of graphs produced for the first studies, the client noted that the confidence interval figures shown had unwanted lines connecting the means (see Figure 2). As we were unable to print the version of the figure sent to the client, we were initially confused as to what was causing this unexpected change. We finally discovered that the device driver specified for the client was for an older printer that did not allow for any grey shading. Therefore, SAS/GRAPH® converted the ci option white to black.

Method 2 - Use More Annotate and Fewer Plots and Symbols

As it seemed impossible to get the desired results with the existing code, we decided to overhaul the entire program. One of the reasons the original programmer used the 2 plot

statements was a general dislike of the annotate facility in SAS/GRAPH®. With some practice, however, it seemed that a little additional annotation could produce the desired graphs with a more straightforward and efficient program. The Annotate Macros, explained in pages 570-587 of SAS/GRAPH Software: Reference, Volume 1 for Version 6, made the modified code simpler.

The same PROC MEANS shown above was used to generate the means and standard errors. In the output data set, however, there is no longer any need to produce three observations for each mean value. Instead, an upper and lower confidence interval is calculated in the same observation.

```
data nauc;
set nauc(where=( _type_ =1));
dummy=n_trt; /* use dummy as 'Z' */
alpha=0.05;
DF=n-1;
t_crit=tinv((1-alpha/2),DF);
ciu=mean + (t_crit*se); cil=mean - (t_crit*se); output;
run;
```

Next, an annotate data set was created for the graph labels. One change requested by the statistician working on the study was to split the last header line, '24-HOUR AUC: MEAN +/- 95% CI', into two lines. With the original code, we were unable to do this, as we had used all 10 available title statements. (The top margin was created by a null title1 statement. Note that, although used in this program, the use of blank title and footnote statements can be a less than ideal method for generating graphics margins. See Kenny, Susan J. (1996), "Useful SAS/GRAPH Tools for a Production Environment", Proceedings of the Twenty-First Annual SAS Users Group International Conference, 21, 836-843.)

With the revised code, we annotated the former final header onto the body of the graph using the %LABEL Annotate Macro.

```
%annomac;
```

```
data anno_lab;
length function $8 text $18;
%system(2,3,4);
```

```
/* create annotated labels */
```

```
%label(2.75,71,"24-HOUR AUC", black,0,0,1,
triplex,5);
%label(2.75,69,"MEAN ( 95% CI",black,0,0,1,
triplex,5);
%label(2.75,69,"G",black,0,0,1,math,5);
```

Note the conversion of '+/-' to a mathematical symbol, which could also have been done in a title statement.

Next the treatment group labels are generated.

```
%label(1,27.5,"3000 mg",black,0,0,.85,triplex,5);
%label(3,27.5,"1500 mg",black,0,0,.85,triplex,5);
%label(4,27.5,"3000 mg",black,0,0,.85,triplex,5);
%label(5,27.5,"4500 mg",black,0,0,.85,triplex,5);
```

The %LINE macro is used to underline the treatment group labels.

```
%line(0.5,25.7,1.5,25.7,black,1,1);
%line(2.5,25.7,5.5,25.7,black,1,1);
```

Finally, the remaining labels are generated.

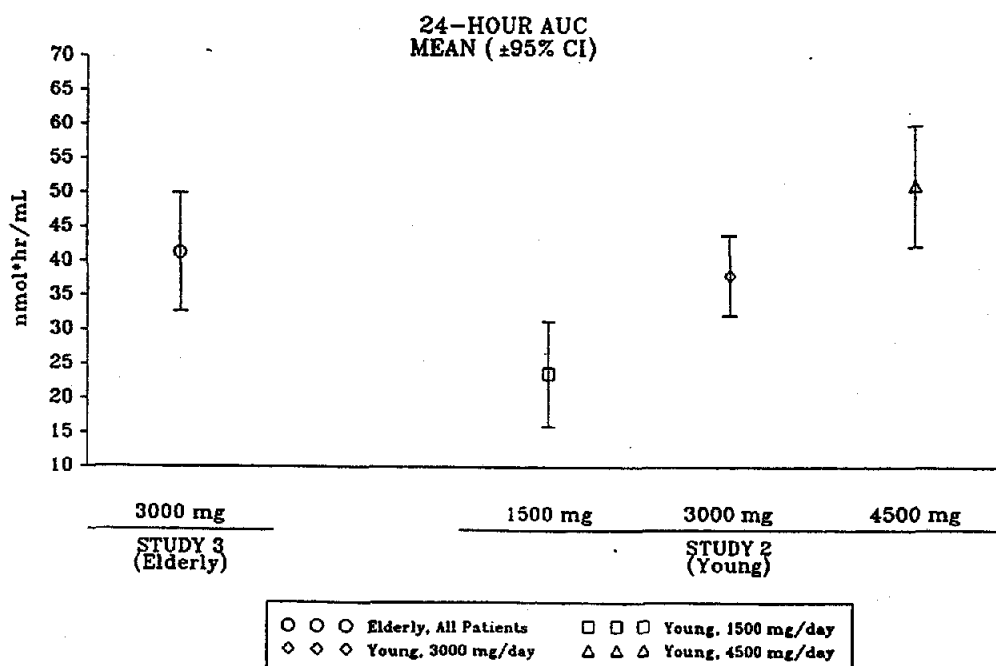
```
%label(1,24.5,"STUDY 3",black,0,0,.85,triplex,5);
%label(4,24.5,"STUDY 2",black,0,0,.85,triplex,5);
%label(1,23, "(Elderly)", black,0,0,.85,triplex,5);
%label(4,23, "(Young)", black,0,0,.85,triplex,5);
run;
```

So far, even though using the Annotate Macros might be a bit easier than the original method, the efficiency gained is pretty minimal. It is with the generation of the confidence intervals that a significantly simpler program is realized.

Another annotate data set is created to draw the confidence intervals. We first set the output data set from the PROC MEANS listed above.

FIGURE 3: MODIFIED, SIMPLIFIED CODE
Your Favorite Compound/ By Your Favorite Client
Protocol 123-4567

FIGURE 3
GRAPHICAL COMPARISON OF ELDERLY AND YOUNG SUBJECTS
YOUR FAVORITE COMPOUND



DRAFT: Program TEST1_96PAP 12AUG96 16:40

The coordinate system for both the x and y axes are set to '2'. This instructs SAS/GRAPH® that both axes will be based on data values in an absolute system.

```
data anno_ci;
  set nauc(keep=n_trt ciu cil);
  %system(2,2,4);
```

The confidence intervals only required three %LINE macro calls. The first one draws a vertical line from the upper limit to the lower limit for each value of n_trt.

```
%line(n_trt,ciu,n_trt,cil,black,1,1);
```

Two additional macro calls add the tops and bottoms to the confidence intervals. A horizontal line is drawn from .04 of the width of a treatment group to the left of the top of each confidence interval to .04 of the width of a treatment group to the right. The same technique is used to add the bottoms. The two annotate data sets are then set together to be used in PROC GLOT below. Although the Annotate Macros are easy to use, a misplaced parameter can yield unexpected results. Therefore, a PROC PRINT of the data set is always a useful for debugging.

```
%line(n_trt-.04,ciu,n_trt + .04,ciu,black,1,1);
%line(n_trt-.04,cil,n_trt + .04,cil,black,1,1);
run;
```

```
data anno;
  set anno_lab anno_ci;
run;
```

```
proc print data=anno;
  title4 'annotate data set';
run
```

With this method of producing the figure, only two plot and four symbol statements are used.

```
axis1 v=none
  label=(r=0 a=90 h=.85 "nmol*hr/mL")
  order=(10 to 70 by 5)
  origin=(, 31.25 pct)
  value=(h=.85) offset=(0,0)
  minor=none;

axis2 v=none
  label=none
  offset=(0,0)
  order=(.5 to 5.5 by .5)
  major=none
```

```
minor=none
length=5.5 in;
```

```
symbol1 v=circle i=none h=1 l=1;
symbol2 v=square i=none h=1 l=1;
symbol3 v=diamond i=none h=1 l=1;
symbol4 v=triangle i=none h=1 l=1;
```

Note that there was no interpolation required in this method. This eliminated the unwanted lines connecting the treatment means. Finally, only one plot statement is required. We still used the same legend as in the previous method.

```
proc gplot data=nauc annotate=anno;
  plot mean*dummy=n_trt/
    vaxis=axis1
    haxis=axis2
    legend=legend;
run;
```

Conclusion

Most clinical trials graphics can be produced in a number of ways. Sometimes a particular method is used due to existing standards set for your workplace, sometimes due to client specifications, or frequently due to the personal preference of the programmer. Regardless of the reason for choosing a specific technique, or the unexpected problems that particular method causes, SAS/GRAPH® software allows for a variety of creative solutions to most any graphics needs.

REFERENCES

Kenny, Susan J. (1996), "Useful SAS/GRAPH Tools for a Production Environment", Proceedings of the Twenty-First Annual SAS Users Group International Conference, 21, 836-843.

SAS Institute Inc., *SAS/GRAPH® Software: Reference, Version 6, First Edition, Volume 1*, Cary NC: SAS Institute Inc., 1990.

AUTHOR CONTACT

Michael Riddle
Statistical Programming Systems
Quintiles, Inc.
PO Box 13979
Research Triangle Park NC 27709
(919)941-7278
mriddle@quintiles.com