Paper 105

Presenting Multi-Level Information in One Plot by Dynamically Generating the PLOT Statement

Li Yuan and Bill Zhang Trilogy Consulting, Info Tech Services, PGA, Waukegan, IL

ABSTRACT

In clinical and pre-clinical trial studies, data sets usually contain information on treatment groups, subjects, event time points, and efficacy parameters. Very often we are interested in presenting treatment group level, subject level, and even subgroup (i.e. gender, age group, tooth surface etc.) level information in the same plot. For example, we might be interested in drawing lines for each subject over the visit time points and at the same time using different line styles and colors to distinguish each treatment group and subgroup in the same plot. Using the current version of SAS/Graph® one can only present one level of information on one plot, i.e., parameter * time point = subject, parameter * time point = treatment group, or parameter * time point = subgroup. One way to improve this is to use SAS® macro facilities and to dynamically generate the PLOT statement. In this paper, we will show you how to use the SAS® macro facility and proc GREPLAY to present multi-level information in one plot.

INTRODUCTION

Suppose we have a clinical study where the average oral gingivitis score (avggi) is measured for the buccal areas of the mouth and measured again for the lingual areas of the mouth for each subject. Subjects received either active or placebo treatment and are seen at three visits. It is trivial to generate a plot to show the comparison between the treatment groups. The code is:

```
proc gplot nofs;
plot avggi * visit = trt;
```

which works perfectly. But sometimes, it is very useful to present multi-level information in the same plot. More complex techniques are needed for presenting all the information we need in the same plot. One of the most significant features of the SAS® Macro facility is that they can be used to make dynamic updates, and one of the most powerful features of proc GREPLAY is that we can overlay as many plots as we need. In the following, we will illustrate how to use the SAS® macro facility to dynamically generate the PLOT statement and to use proc GREPLAY to place the correct legend.

METHORD

Quite often it is desired to create a plot which presents efficacy parameters verses event time points for all subjects in all treatment groups. Different line types are used to distinguish the different treatment groups and different colors are used to distinguish the different subgroups. Basically, we need two steps to accomplish this task, drawing the proper lines and generating the correct legend to associate with the plot.

A. PLOT STATEMENT

Lines are drawn by plot statement. One plot statement is needed for each subgroup. The plot statement that we need should have following structure:

```
plot pat11 * visit =1 … pat1n_1 * visit = 1
     pat21 * visit =2 … pat2n_2 * visit = 2
                    .
                    .
                    .
     patm1 * visit =m … patmn_m * visit = m;
```

Here we have m treatment groups and n_i subjects in the i^{th} treatment group for the current subgroup. This plot statement will draw a line for each subject in that subgroup and the subject in i^{th} treatment group will use the i^{th} symbol defined by statement symbol. All lines in this subgroup will have the same color.

The following code will dynamically generate the plot statement which has above structure.

```
%global r;
%let r = %str(plot );
%do I = 1 %to &trtcnt;
    %do j =1 %to &ni;
        %let p = %str(d&i&j * visit = &I);
        %let r = &r&p;
    %end;
%end;
```

where trtcnt = number of treatment groups.

B. LEGEND

The previous plot statement would not generate a correct legend for the plot, so we need to generate a proper legend separately. The idea is, first we choose a subset of the original data set to contain only one subject for each treatment group. The following code

```
proc gplot nofs;
     plot avggi * visit = trt;
```

will generate correct legend for treatment groups. Then, we choose a subset of the original data set that contains only one subject for each subgroup from the first treatment group. Using the following code

```
proc gplot nofs;
     plot avggi * visit = subgroup;
```

we will get the correct legend for subgroups.

Finally, we use proc GREPLAY to overlay all plots together as follows:

```
proc greplay igout = work.gseg
    tc = tempcat nofs;
    tdef test
    1/ llx = 0 lly = 0 ulx = 0 uly =100
        urx =100 ury =100 lrx =100 lry = 0
    2/ llx = 0 lly = 0 ulx = 0 uly =100
        urx =100 ury =100 lrx =100 lry = 0
                     .
                     .
                     .
    k/ llx = 0 lly = 0 ulx = 0 uly =100
        urx =100 ury =100 lrx =100 lry =0;
    template test;
    treplay 1:plot1 2:plot2 … k:plotk;
run;
```

Here we have k-2 subgroups.

For the purpose of matching all plots, we need to control all major properties of the axis, such as, label, tick mark, length, font, offset, and origin. This is needed so that the lines from all plots will lay together on the same plot and no extra lines will appear.

EXAMPLE

Appendix A is the code for generating the plot in Figure 1 which contains treatment group level , subject level, and tooth surface location information for the oral gingivitis study. Simulated data are used in this example.

CONCLUSION

The programming techniques presented in this paper can be applied to a wide variety of applications using SAS®.

SAS and SAS/GRAPH are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

REFERENCES

SAS Institute Inc. (1990), *SAS Guide to Macro Processing*, Version 6, Second Edition, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1990), *SAS/Graph Software*, Volume 1 & 2, Reference Version 6, First Edition, Cary, NC: SAS Institute Inc.

ACKNOWLEDGMENTS

We would like to thank Mr. Robert D. Bartizek for his careful review and helpful comments.

CONTACT INFORMATION

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

Li Yuan
Trilogy Consulting, PGA
850 South Greenbay Rd.
Waukegan, IL 60085
(513)622-2464
(513)622-1600
yuan.l@pg.com

Bill Zhang
Trilogy Consulting, PGA
850 South Greenbay Rd.
Waukegan, IL 60085
(513)626-1229
(513)622-1600
zhang.b.3@pg.com

APPENDIX A

```
******    PLOT.SAS      ******;

/* Define colors */
%let color1=blue;
%let color2=lime;

/* Define axis and legend */
axis1 label=(a=90 r=0 h=0.8 c=black f=swissl
                   "Average GI Score")
      minor=none
      length=4 in
      value=(h=0.8 f=swissl c=black )
      order=(0.4 to 2.2 by 0.2);
axis2 label=(h=0.8 f=swissl c=black "Visit")
      value=(h=0.8 f=swissl c=black )
      minor=none
      length=6 in
      origin=(4 cm, 7 cm)
```

```
      offset=(2,2);
legend1 label=(h=0.8 f=swissl c=black
              ‘Treatment')
        value=(h=0.8 )
        position=(bottom center outside)
        origin=(3 cm, 2.5 cm)
        across=1
        frame;
legend2 label=(h=0.8 f=swissl c=black
              'Surface')
        value=(h=0.8 'Lingual' 'Buccal')
        position=(bottom center outside)
        origin=(12 cm, 2.5 cm)
        across=1
        frame;
data test;
     set library.test1;
run;
proc sort; by trt patno visit; run;
data test1;
     set test;
     by trt patno visit;
     if first.trt;
run;

/* Count total treatment groups */
data trtcode;
     set test1 end=eof;
     by trt patno;
     treat = _n_;
     if eof then
        call symput('trtcnt',_N_);
     keep trt treat;
run;
data test;
     merge test trtcode;
     by trt;
run;

%macro subgroup(subgroup);
/*Count subjects numbers for treatment
group */
%do i=1 %to &trtcnt;
    data trt&i;
         set test;
         if surface=&subgroup and
treat=&i;
    run;
    proc sort; by visit patno; run;
    proc means noprint;
         var treat;
         where visit=1;
         output out=patcnt&i n=N;
    run;
    proc transpose prefix=d&i data=trt&I
         out=trt&i;
         var avggi;
         by visit;
    run;
    proc sort; by visit; run;
%end;

/* Merge all treament groups together */
data all&subgroup;
     merge trt1 trt2;
     by visit;
run;
%do i=3 %to &trtcnt;
    data all&subgroup;
         merge all&subgroup trt&i;
         by visit;
    run;
```

```
%end;

/* Generate plot statement for first plot*/
%global r&subgroup;
%let r&subgroup=%str(plot );
%do i= 1 %to &trtcnt;
    data _null_;
         set patcnt&i;
         call symput ('patcnt', N);
    run;
    %do j=1 %to &patcnt;
        %let p=%str(d&i&j*visit=&i );
        %let r&subgroup=&&r&subgroup&p;
    %end;
%end;
%let q=%str( / overlay haxis=axis2
       vaxis=axis1 nolegend
       name="plot&subgroup";);
%let r&subgroup=&&r&subgroup&q;

/* Define symbols for all treatment groups */
%do i=1 %to &trtcnt;
    symbol&i h=0.8 width=3 i=join l=&I
             c=&&color&subgroup;
%end;
/* Draw first plot */
PROC gplot data=all&subgroup;
    &&r&subgroup;
run;
%mend;
%subgroup(1);
%subgroup(2);

/* Generate data set for first legend */
data test2;
     set test;
     if surface=2;
run;
proc transpose data=test2 out=test2 prefix=visit;
     var avggi;
     by trt patno;
run;
data dummy1;
     set test2;
     by trt;
     if first.trt;
     visit=1; avggi=visit1; output;
     visit=2; avggi=visit2; output;
     visit=3; avggi=visit3; output;
     keep trt visit avggi;
run;
proc sort; by trt visit;

/* Draw first legend */
proc gplot data=dummy1;
     plot avggi*visit=trt / haxis=axis2
          vaxis=axis1
               legend=legend1
          name='legend1';
run;

/* Generate data set for second legend */
data test3;
     set test;
     if treat=1;
run;
proc sort; by surface patno visit; run;
proc transpose data=test3 out=test4
     prefix=visit;
     var avggi;
     by surface patno;
run;
data dummy2;
     set test4;
     by surface patno;
     if first.surface;
     visit=1; avggi=visit1; output;
     visit=2; avggi=visit2; output;
     visit=3; avggi=visit3; output;
     keep surface visit avggi;
run;
proc sort; by surface visit;
symbol1 h=0.8 width=3 i=join l=1
c=&color1;
symbol2 h=0.8 width=3 i=join l=1
c=&color2;
/* Draw second legend */
proc gplot data=dummy2;
     plot avggi*visit=surface /
haxis=axis2
     vaxis=axis1
     legend=legend2
     name='legend2';
run;
goptions display;

/* Overlay all plots on same page */
proc greplay igout=work.gseg tc=tempcat
nofs;
    tdef test
    1/ llx=0 lly=0 ulx=0 uly=100
        urx=100 ury=100 lrx=100 lry=0
    2/ llx=0 lly=0 ulx=0 uly=100
        urx=100 ury=100 lrx=100 lry=0
    3/ llx=0 lly=0 ulx=0 uly=100
        urx=100 ury=100 lrx=100 lry=0
    4/ llx=0 lly=0 ulx=0 uly=100
        urx=100 ury=100 lrx=100 lry=0;
  template test;
  treplay 1:plot1 2:plot2
          3:legend1 4:legend2;
run;
```

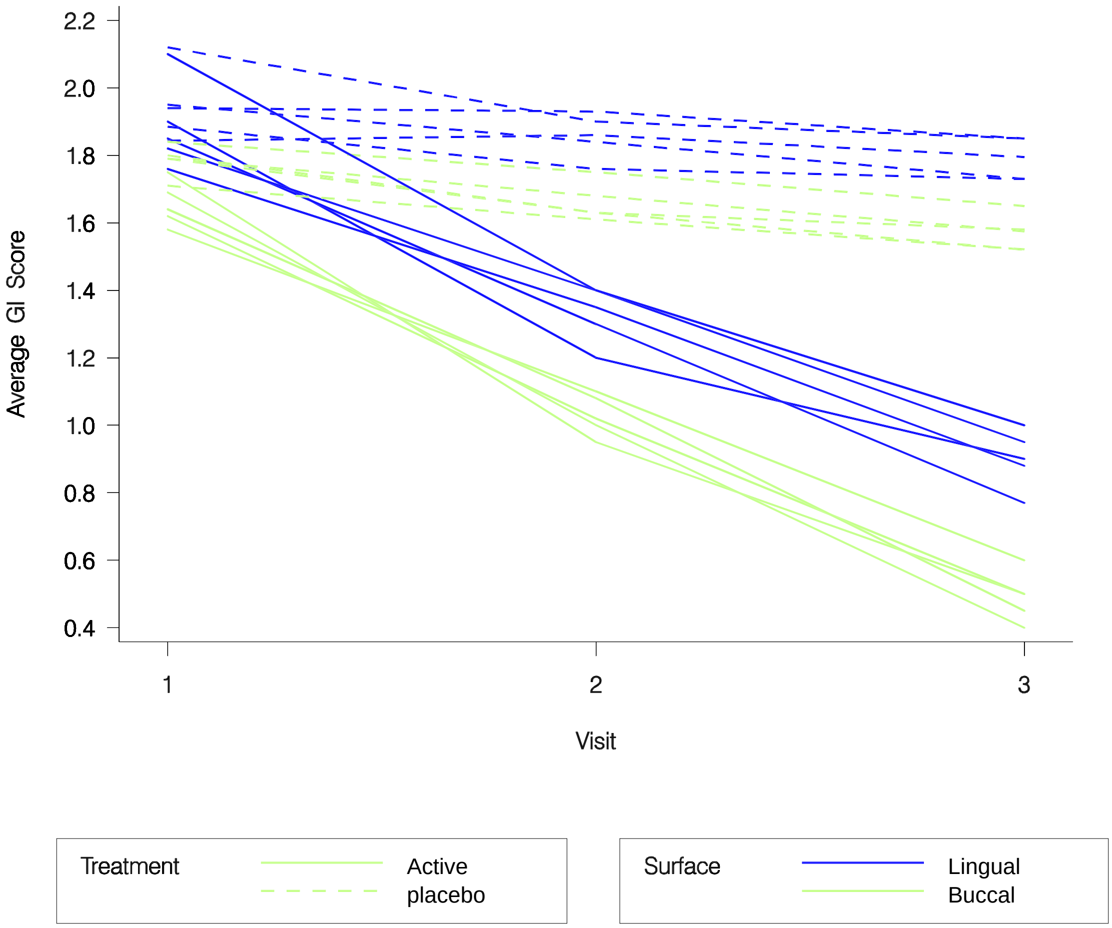

Figure 1. Plot of Average Löe-Silness Gingival Index Score versus visit