DYNAMICALLY ASSIGNING TREPLAY STATEMENTS FOR PRODUCING MULTIPLE PLOTS PER PAGE

Alice Lail and Karen Wade, Rho Inc., Chapel Hill, NC

ABSTRACT

Often in clinical trials, it is of great interest to create plots of each individual subjects' data. When there numerous subjects in the study, placing more than one plot per page saves space and paper. But what do you do for those subjects who have all missing data for the variable of interest? This poster presents ideas on how to handle situations in which one or more subjects may be missing all data for the given variable of interest. This poster will compare a brute force method for producing multiple plots per page and a more refined solution. The refined solution includes the use of macros that cycle through each subject and dynamically writes TREPLAY statements for the PROC GREPLAY needed to place multiple plots per page.

INTRODUCTION

During clinical studies, a figure for each patient is often desired. For example, in an HIV study, a clinician may wish to see how each patient's HIV viral load changes over time. A plot of the HIV viral load over time provides an informative to find general trends across patients. When there are hundreds of patients in the study, flipping through hundreds of pages of figures is time consuming. However, we can save time and paper by placing multiple figures on each page. Placing 4 figures on each page with a single title at the top of the page will drastically reduce the time and effort to review the figures. In addition, it is also useful information if the patient does not sufficient data to be plotted. Instead of skipping a patient who does not have sufficient data, it would be useful to place text indicating the reason a figure does not appear for that specific patient.

METHODS

In general, PROC GPLOT will produce the figures and PROC GSLIDE will produce the text for patients without sufficient data to be plotted. After creating the GPLOT or GSLIDE as appropriate for each patient, GREPLAY is used to place 4 patients on each page.

The first step is to create an indicator variable for patients without sufficient data to be plotted. The method for creating this indicator will differ from program to program depending on the structure of the input data. Once this indicator variable is created, the patient identifiers are stored in the macro variables *ID1, ID2, ID3*, etc. The count of the total number of patients who fall into the category of insufficient data is also stored in a macro variable. In the example below, the variable *noplot* has a value of 1 for patients who will require a GSLIDE instead of GPLOT to be generated.

```
if noplot=1 then do;
  i+1;
  i2=put(i, 8.);
  call symput("id"||trim(left(i2)), trim(left(id)));
  call symput("noplottot", trim(left(i2)));
 end;
```

For the remaining patients, a figure is created using a BY-statement with PROC GPLOT. The following PROC GSLIDE produces the title that will be placed on each page.

```
proc gplot data=ready gout=work.figs;
 where noplot=0;
 by id;
 plot hivrna*cycle / vaxis=axis1 haxis=axis2 vref=500 anno=anno;
run; quit;

proc gslide gout=work.figs;
title1 h=1 f=simplex "Plot of HIV RNA by Cycle for Participants With at
Least 1 Value Greater Than 5000 copies/mL";
run; quit;
```

Next, using the list of patients without sufficient data to be plotted that was created earlier, a GSLIDE for each patient is created containing an informative text message. The macro variable *&&id&i* resolves to the patient identifier.

```
%macro noplot();
%do i=1 %to &noplottot;
 proc gslide gout=work.figs;
  title1 " ";
  title2 "Participant &&id&i does not have sufficient data to plot";
  footnote " ";
 run;
 quit;
%end;
%mend;
%noplot;
```

In order to place 4 patients on each page, PROC GREPLAY is used to create a 5-panel template and to place each GPLOT and each GSLIDE into its proper location. For each page that is generated, a TREPLAY statement is required. Since the number of pages required can change from program to program, the TREPLAY statement will be created dynamically. In general, the TREPLAY statement appears similar to

treplay 1:Gplot 2:Gplot1 3:Gplot2 4:Gplot3 5:Gslide.

The first 4 panels contain patient data and the last panel contains the overall title that is placed on the top of each page. The trick is knowing where the GSLIDEs for patients with insufficient data are interspersed throughout the GPLOTs. In order to create the TREPLAY statement dynamically, a variety of macro variables is needed.

```
data temp;
length plotid $8;
set ready;
by id;
if first.id then do;
  i+1;
  round=ceil(i/4);
  cround=put(round, 8.);
  call symput("RTotal", trim(left(cround)));

  if noplot=0 then do;
    count+1;
    j=count-1;
    cj=put(j, 8.);
  end;

  if noplot=1 then do;
    p+1;
     cp=put(p, 8.);
  end;

  k+1;
  if k=5 then k=1;
  ck=put(k, 8.);
  call symput("fn", trim(left(ck)));
  if noplot=0 then plotid="Gplot"||trim(left(cj));
  else if noplot=1 then plotid="Gslide"||trim(left(cp));
  if j ne 0 then call symput("Fig"||trim(left(cround))||trim(left(ck)),
      plotid);
  else if j=0 then call symput
      ("Fig"||trim(left(cround))||trim(left(ck)), "Gplot");
  end;
  run;
```

Variable	Function
i	Counter of number of patients
cround	Counter for number of pages
&Rtotal	Total number of pages needed
j, cj	Counter of number of patients with sufficient data
P, cp	Counter of number of pages with insufficient data
K, ck	Counter 1-4 of the position of plots on each the page
&Fn	Counter of the number of plots appearing on the last page
Plotid	Name of catalog member for each plot
&Fig&cround&ck	Names each figure based on the page number and position on the page. For example, a figure on the 3rd page in the 2nd position would be Fig32.

The PROC GREPLAY that appears below defines a 5-panel template and then replays all the GPLOTs or GSLIDEs into their appropriate locations.

```
proc greplay nofs igout=work.figs tc=tempcat;
  tdef spec5 des='five panel template'
  1/ llx=1  lly=50
     ulx=1  uly=90
     urx=50 ury=90
     lrx=50 lry=50
  2/ llx=50 lly=50
     ulx=50 uly=90
     urx=99 ury=90
     lrx=99 lry=50
  3/ llx=1  lly=10
     ulx=1  uly=50
     urx=50 ury=50
     lrx=50 lry=10
  4/ llx=50 lly=10
     ulx=50 uly=50
     urx=99 ury=50
     lrx=99 lry=10
  5/ llx=0  lly=0
     ulx=0  uly=100
     urx=100 ury=100
     lrx=100 lry=0
   color=white;
template spec5;

%do A=1 %to &Rtotal;
  %if &A=&RTotal %then %do;
      treplay %do grph = 1 %to &fn;
        &grph.: &&fig&A.&grph
      %end;
      5: gslide;
      run; quit;
  %end;

%else %do;
      treplay %do grph = 1 %to 4;
        &grph.: &&fig&A.&grph
      %end;
      5: gslide;
%end;
%end;

%mend;

options orientation=landscape;
ods pdf file="&PGMPATH\&PROG..pdf";
%count_pt;
ods pdf close;
```

For each page of output, specified by the do-loop from 1 to &Rtotal, a TREPLAY statement is generated. The macro variable *&grph* will cycle from 1 to 4 to place each plot in its appropriate panel on the page. The 5th panel is always constant – the GSLIDE containing the overall title. The name of each figure is a combination of page number and the panel number on that page. For the last page of output, it is possible that there are only 2 or 3 plots left. Instead of looping from 1 to 4, the loop cycles from 1 to the macro variable *&Fn*. The RUN and QUIT statements are included in the do-loop for the last page and ends the GREPLAY procedure.

CONCLUSION

PROC GREPLAY provides a way to place multiple figures on a page. By combining GPLOTs and GSLIDEs, patients with insufficient data for plotting can be identified. The TREPLAY statements that generate each page of output can be dynamically created using a combination of macro variables and do-loops.

Alice Lail (alail@rhoworld.com) and Karen Wade (kwade@rhworld.com)
Rho, Inc.