Paper 068-2012

PROC REPORT: Dynamic Column Headings

Jason Levy, PPD, Hamilton, NJ

ABSTRACT

This paper shows a technique to dynamically make a column header that is independent of the data in the column itself. It also uses one macro variable to subset the data and to create the column header.

INTRODUCTION

You have just been asked to create a table which you need to to have quality of life data, visits, and question results. In order to save space the left most column heading needs to the question category but the body of the table should be each visit. You start programming but realize it is not straight forward. How will you force a category in a title heading when a visit is in the data? One method is to subset the data one category at a time using a macro.

DATA EXAMPLES

Below is the code used to produce the data set for this paper.:

```
data test;
  length cyctext $20;
  do ord = 0 to 4;
    do cycle = 1 to 11;
      cyctext = "Cycle " || strip(put(cycle,best.)) || ", Day 1";
       do trtcd = 1 to 3;
         do score = 1 to 5;
           seq = score;
           output;
         end;
       end;
    end;
  end;
run;
```

Per ord value we have many cycles (or visits) and within each visit are possible treatment groups. Also there are multiple scores per treatment group within a visit.

GET THE LIST AND ORDER OF VALUES FROM THE FORMAT

Using PROC FORMAT and cntlout the values needed can be obtained:

```
proc format library=work cntlout=fmtout(keep=fmtname start label);
  value testcatf
  0 = 'Overall Score'
  1 = 'Physical Well Being'
  2 = 'Social/Family Well Being'
  3 = 'Emotional Well Being'
  4 = 'Functional Well Being'
  ;

  value tmptrtf
  1 = 'Active'
  2 = 'Placebo'
  3 = 'Total'
  ;
run;
```

Now that we have all the possible values to subset and the format we can subset the report. Format testcaf will be used as the column heading for variable cyctext. The start values of format testcatf will be used to subset the dataset per category. A format isn't mandatory but it does provide a central place for the subsetting values to exist. Having a central location allows for easy changing of values as needed such as if the requirement change of CRF changes.

USING PROC REPORT WITHIN A MACRO

The report will be called for each subsetting value so no page values will be used.

```
options pageno=1;
%macro printm(inx=);
  proc report missing data=test(where=(ord=&inx)) headline nowd spacing=2;
    column cycle seq cyctext trtcd,score foolrpt;
    define cycle / group order=internal noprint;
    define seq / group order=internal noprint;
    define cyctext / group left width=20 "%sysfunc(putn(&inx, testcatf.))" flow;
    define trtcd / across order=internal f=tmptrtf. ' ' width = 7;
    define score / display width=7 "Score";
    define foolrpt / noprint;

    break after cycle / skip;
  run;
%mend;
```

Using where= on the dataset will keep only the values we need for this table. This also allows the column heading to be consistent with the subsetted data. Using putn with %sysfunc allows the subsetted value to be formatted at run time. Only one macro variable is needed since we made the subsetting values the same as the format start values. Thus it is less to maintain. Now we can call the macro using call execute:

```
data fmtout;
  set fmtout(where=(strip(upcase(fmtname)) = ‘TESTCATF’));
  ord = input(strip(start),best8.);
run;

proc sort data=fmtout out=fmtout nodupkey;
  by ord start;
run;

data _null_;
  set fmtout(keep=ord start);
  by ord;
  if first.ord then call execute('%printm(inx=' || strip(start) || ');');
run;
```

Using call execute we can take advantage of the dataset created from the cntlout of the PROC FORMAT. Calling the macro without call execute will work as well but you will need to add/change calls as needed based on changes.

CONCLUSION

This is a simple way to force a dynamic column heading using into a table without supplying a separate macro parameter for each call to the table. Other efficiencies for maintaining the program were gained by using call execute in conjuction with cntlout of PROC FORMAT.

This method is not limited to PROC REPORT (e.g., PROC PRINT can use this method as well).

CONTACT INFORMATION

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

Name: Jason Levy
Enterprise: PPD, Inc.
Address: 3575 Quakerbridge Rd.
City, State ZIP: Hamilton, NJ 08619
Work Phone: 1-609-528-8121
E-mail: jason.levy@ppdi.com