PharmaSUG 2016 - Paper QT09

A SAS Macro Tool to Automate Generation of Customized Patient Profile in PDF Documents

Haining Li, Massachusetts General Hospital, Boston, MA
Merit E. Cudkowicz, Massachusetts General Hospital, Boston, MA
Hong Yu, Massachusetts General Hospital, Boston, MA

ABSTRACT

Clinical trials data are collected from many different sources and the data are often stored in many SAS tables while each SAS table includes multiple patients. Once the trial begins, all of the data needs to be reviewed and cleaned in real time. To effectively monitor study progress and subject safety, patient profiles are commonly used in this review process. These profiles will contain all current data for each patient that will allow reviewers to rapidly assess the patient's overall status as well as any level of detail information. To reduce the effort in organizing the huge volumes of tables and variables within each clinical trial, this paper will provide a SAS Macro to automatically drop, keep or rename variables from all or some of the tables, and generate the ultimate patient profile by subject in PDF documents. This macro also provides options to automatically drop any column when all the data in the column are missing or equal to certain values. This method will significantly improve the efficiency in clinical trial data review process.

INTRODUCTION

In clinical trials, it is a common practice for SAS programmers to work with a big amount of tables and variables in a study, while each table could contain some aspects of information for multiple subjects. It would be time consuming and error prone for reviewer to look across multiple tables for patient information. To effectively monitor study progress and subject safety and reduce the review workload, the SAS programmers might be asked to present the profiles by patient, which needs to extract the data from multiple tables and then present by patient, therefore each subject will has a single file which includes all aspects or any level of detail information. Furthermore, to make the review process more efficient, some of the variables might need to be dropped or renamed to reduce the clutter information.

OVERVIEW OF THE MACRO

To automate the patient profile listing, a SAS macro is created, which accepts multiple SAS tables as input, and generates PDF documents by patient. The output documents can also be in other formats, such as RTF, EXCEL. The details of the macro are given below.

The macro has eight parameters:

1) Lib: indicate the library name of the original data set of a study. This data set will be copied to WORK library to prevent the original library data from being modified.
2) drop_var: indicate if variables need to be dropped. Value 0 can be passed as this parameter if no variable needs to be dropped.
3) drop_stat: indicate the variables or what kind of the variables need to be dropped. Certain variables can be dropped from all or some of the tables in this study. Variables can be dropped by variable names or variable formats.
4) renm_var: indicate if variables need to be renamed. Value 0 can be passed as this parameter if no variable needs to be renamed.
5) renm_stat: indicate the variables or what kind of the variables need to be renamed.
6) re_nm: indicate the new name based on defined pattern.
7) spec_num: indicate a numeric value. This can prevent printing any variable that always equals to a certain numeric value. For example, if spec_num is set to missing, then any numeric variable that only has missing values will not be printed out.

8) spec_char: indicate a character value. This can prevent printing any variable that always equals to a certain character value. For example, if spec_char is set to missing, then any character variable that only has missing values will not be printed out.

The fundamental operations of the macro include: put all the subject IDs in the study into a macro variable separated by ‘ ‘; analyze dataset and variable in the library; drop variables; rename variables; delete any variable that always equal to a certain value so it won’t be printed.

Here is the content of the Macro:

```
%macro all (lib=dfile, drop_var=, drop_stat=, renm_var=, renm_stat=, re_nm=, 
spec_num=, spec_char=);

 *Select all the subject IDS in the study;
 proc sql noprint ;
  select ID into :  IDs separated by ' '
   from  dfile.subj
   order by ID;
 quit;

 proc datasets;
  copy in=&lib out=work;
 quit;

 %macro lib_data;
  proc sql;
   drop table work._astdtab_&lib.;
   drop table work._astdcol_&lib.;

   create table work._astdtab_&lib. as
    select memname label='SAS Table Name', nobs label='Num of Obs', nvar 
label='Num of Var'
     from dictionary.tables
     where libname in ("%UPCASE(&lib.)")
     order by memname;

   create table work._astdcol_&lib. as
    select upcase(memname) as memname label='Table Name', upcase(name) as name 
label='Variable Name' , label label='Description/Label', format label='Format'
, informat label='Informat', type label='Type', length  label='Length', varnum
     from dictionary.columns
     where libname in ("%UPCASE(&lib.)")
     order by memname, varnum;
  quit;
 %mend lib_data;
 %lib_data;

 %if &drop_var %then %do;
   proc sql;
    create table drop_vars as
      select memname, name
      from work._astdcol_work
      where &drop_stat
      order by memname, name;
    quit;

   data _null_;
    set drop_vars;
    by memname name;
    retain j k (0,0);j=j+1;
    if first.memname then do;
```

```
           j=1; k=k+1;
        call symput('numtxds',k);
        call symput(compress('nmxdataset'||k),memname);
      end;
      if last.memname then do;
         call symput(compress('numdtxv'||k),j);
      end;
     call symput(compress('nmtxv'||k||'_'||j),name);
   run;

     %do k=1 %to &numtxds;             /* k-dataset count;*/
       %do i=1 %to &&numdtxv&k;
          data  &&nmxdataset&k;
              set  &&nmxdataset&k;
              drop &&nmtxv&k._&i;
          run;
       %end;
     %end;
%end;

%lib_data;
%if &renm_var %then %do;
  proc sql;
    create table renm_vars as
      select memname, name
        from work._astdcol_work
        where &renm_stat
        order by memname, name;
  quit;

  data _null_;
    set renm_vars;
    by memname name;
    retain j k (0,0);j=j+1;
    if first.memname then do;
      j=1; k=k+1;
      call symput('numtxds',k);
      call symput(compress('nmxdataset'||k),memname);
    end;
    if last.memname then do;
      call symput(compress('numdtxv'||k),j);
    end;
    call symput(compress('nmtxv'||k||'_'||j),name);
  run;

      %do k=1 %to &numtxds;
         %do i=1 %to &&numdtxv&k;
            data  &&nmxdataset&k;
              set  &&nmxdataset&k;
              rename &&nmtxv&k._&i =&re_nm._&&nmtxv&k._&i  ;
            run;
         %end;
      %end;
%end;


proc sql;
  select distinct memname into :  datanms_all separated by ' '
    from dictionary.columns
    where libname ='DFILE' and MemType eq "DATA"
    order by memname;
quit;
```

```
  %do ii = 1 %to %eval(%sysfunc(count(&datanms_all,%str( ))) + 1);
    %let datanm = %scan(&datanms_all,&ii,%str( ));
      data _null_;
        call sympt("t_num", trim(left(put(nobs, best.))));
        set &datanm(obs=0) nobs=nobs;
      run;

      proc sql noprint;
        select count (*),   upcase(name) into : &datanm._numv_n, :&datanm._numv
separated by ' '
          from dictionary.columns
           where libname ="WORK" and memname in ("%UPCASE(&datanm)") and
lowcase(type)='num';

        select count (*),  upcase(name) into : &datanm._charv_n, :&datanm._charv
separated by ' '
         from dictionary.columns
         where libname ="WORK" and memname in ("%UPCASE(&datanm)") and
lowcase(type)='char';
      quit;

    %if &t_num %then %do;
      data _null_;
        length num_list $32767;
        set &datanm end=lastrcd;
           %if &&&datanm._numv_n %then %do;
            %do jj = 1 %to %eval(%sysfunc(count(%sysfunc(compbl(
%sysfunc(trim(%sysfunc(left(&&&datanm._numv )))))),%str( ))) + 1);
              %let var  =%sysfunc(compbl(
%sysfunc(trim(%sysfunc(left(%qscan(&&&datanm._numv ,&jj,%str( ))))))));
            retain n_&var (0) ;
            if &var^=&spec_num then  do; n_&var = n_&var+1;  end;
            if lastrcd then do;
              if n_&var=0 then num_list=trim(num_list)||' '||trim(vname(&var));
              call symput('num_list',num_list);
            end;
          %end;
        run;
      %end;


      data _null_;
        length char_list $32767;
        set &datanm end=lastrcd;
        %if &&&datanm._charv_n %then %do;
          %do kk = 1 %to %eval(%sysfunc(count(%sysfunc(compbl(
%sysfunc(trim(%sysfunc(left(&&&datanm._charv )))))),%str( ))) + 1);
          %let var  =%sysfunc(compbl(
%sysfunc(trim(%sysfunc(left(%qscan(&&&datanm._charv ,&kk,%str( ))))))));
          retain n_&var (0) ;
          if &var^=&spec_char then  do;  n_&var = n_&var+1;  end;
          if lastrcd then do;
          if n_&var=0 then char_list=trim(char_list)||' '||trim(vname(&var));
          call symput('char_list',char_list);
          end;
        %end;
       run;
     %end;

     %if &num_list ^=%str( ) or &char_list ^=%str( ) %then %do;
      data  &datanm;
        set &datanm;
        drop &char_list &num_list;
      run;
    %end;
    %else %do;
      data &datanm;
        set &datanm;
      run;
    %end;
  %end;

  %else %if &t_num=0 %then %do;
    data &datanm;
      set &datanm;
    run;
  %end;
  %end;

  %do m = 1 %to %eval(%sysfunc(count(&IDs,%str( ))) + 1);
    %let id = %scan(&IDs,&m,%str( ));
    ODS PDF file="&Output_Dir.\DATA_&ID..pdf";
    title "Subject &ID";
    %do n = 1 %to %eval(%sysfunc(count(&datanms_all,%str( ))) + 1);
      %let datanm = %scan(&datanms_all,&n,%str( ));
       proc print data=  &datanm;
          where ID="&ID";
       run;
       %end;
   ODS PDF close;

  %end;
%mend all;


%all (lib=work,
      drop_var= 1,
      drop_stat = %str(format='APPROXIM.' or (format='MMDDYY10.' and index(name,'N_'
))),
      renm_var= 1,
      renm_stat= %str(type='num')  ,
      re_nm= %str(n),
      spec_num = %str(.),
      spec_char=%str('') );
```

To use the above listed macro, the programmer needs to put all SAS tables into a single folder and specify an output folder to save the generated PDF files.

CONCLUSION

In this paper, we presented a useful method to automate generation of customized patient profile in PDF documents. The method is implemented in SAS macro format and verified by real dataset. This method can be applied in a variety of situations and improve the review process.

REFERENCES

< SAS(R) 9.2 Language Reference: Concepts, Sec
ond Edition>. "<SAS Variable Lists>."   Available at
<http://support.sas.com/documentation/cdl/en/lrcon/62955/HTML/default/viewer.htm#a000695105.htm>.

RECOMMENDED READING

- Advanced Programming for SAS®

CONTACT INFORMATION

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

Name: Haining Li
Enterprise: Mass General Hospital
Address: 50 Staniford St
City, State ZIP: Boston, MA
Work Phone: 617-643-5123
Fax:
E-mail: Hli15@partners.org
Web:
Twitter:

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.