

A SAS Macro to Generate Customized Patient Profile Output Simply

Mingyue Zhu, Everest Clinical Research

ABSTRACT

Patient profile displays all (or selected, sometimes) information together for individual subject. It provides an opportunity for clinical reviewers to have a whole perspective of subject's status and to find the issues in database.

Patient profile may just need to simply display all the information collected on the Case Report Forms (CRF). However, in some cases, the study requires a customized patient profile output after integrating different kind of data, which is like what we do in generating CSR tables or listings except the patient profile requires presenting all information together for each subject. In this situation, programmers generally need two steps when programming, generate the datasets according to shell of patient profile first and then start tedious output programming for each dataset.

This paper introduces a macro which can automatically generate customized patient profile output. What you need to do is to generate all needed datasets under the same folder, fill in the excel which will be used in the macro and then call the macro. This macro allows you to add additional titles or footnotes, highlight color, specified width and arrange the layout when there are too many columns in a form. All the output adjusting could be completed by editing the excel and programmers don't need to use lots of procedures for output such as PROC REPORT or PROC PRINT. It will significantly improve the efficiency when generating patient profile.

INTRODUCTION

Generating patient profile is a common task for programmers in clinical trials industry. Programmers usually generate necessary datasets first, and then use many PROC REPORT procedures for each dataset's output, that would be time wasting. The "m_profile" macro could simplify the output generation and adjusting process and is easy to understand and use.

The programmers need to get variable label and display order ready for output when programming final data sets and put them in same folder. This macro is used to generate outputs for all these final data sets in a single Rich Text Format (RTF) file for each subject. And the output format could be adjusted in metadata configure excel file, therefore, the programmer only need to focus on generating final data sets and avoid long text output programming.

CONTROL METADATA CONFIGURE EXCEL

The metadata configure excel file is going to be edited when all corresponding final data sets are ready in a folder. The macro will import the excel file and use it for output format adjusting. An example of the excel file is provided in Figure 1.

The excel contain eight columns: DATASET, TITLE, FOOT, IDNUM, WIDTH, RUN, HIGHLIGHT and INCLUDE as below:

DATASET	Data set name Values in this column should be exactly equal to the final data set name and its order in excel is the order in output.
TITLE	Values in this column determine the title of individual table or figure. This column allows individual table to have multiple titles and "#" is used to split each title.
FOOT	Values in this column determine the footnote of individual table. This column allows individual table to have multiple footnotes and "#" is used to split each footnotes.

Patient Information						
Drug Name	Treatment Date	Treatment Dose	ICF Signing Date	Study End of Date		
A	2020-01-05	50mg	2020-01-02	2020-11-01		

Adverse Event Summary						
Preferred Term Code	Severity/Intensity	Action Taken with Study Treatment	Causality	Outcome of AE	AE Start Date	AE End Date
Ecchymosis	MODERATE	DOSE NOT CHANGED	Y	RECOVERED/RESOLVED	2020-01-11	2020-02-13
Injection site haemorrhage	SEVERE	DOSE NOT CHANGED	Y	RECOVERED/RESOLVED	2020-01-30	2020-02-13
Urticaria	MILD	DOSE NOT CHANGED	N	RECOVERED/RESOLVED	2020-01-07	2020-02-04
Adverse Event Dummy Footnote 1						
Adverse Event Dummy Footnote 2						

Adverse Event Summary						
Preferred Term Code	Severity/Intensity	Cause Birth Defect	Cause Signif Disability/Incapacity	Results in Death	Requires or Prolongs Hospitalization	Is Life Threatening
Ecchymosis	MODERATE	No	No	No	No	No
Injection site haemorrhage	SEVERE	No	No	No	No	No
Urticaria	MILD	No	No	No	No	No
Adverse Event Dummy Footnote 1						
Adverse Event Dummy Footnote 2						

Figure 2. Sample output for repeat column and background high light

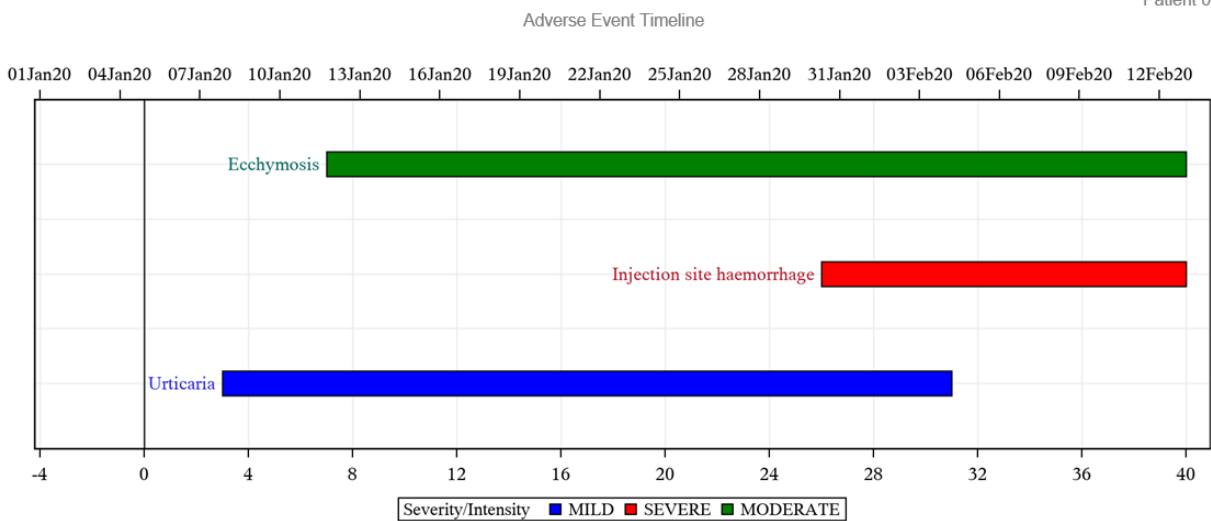


Figure 3. Sample output for figure

CONCLUSION

The introduced macro %m_profile can efficiently generate customized patient profile output base on the final data sets programmed per patient profile mock shell provided by clinical team. This macro is mainly base on SAS Base and the logic is clear to understand, therefore, it's easy to use. The configure excel file make it easy to control the output format and accomplish customized display. Meanwhile, it can include figure programs flexibly to generate tables and figures in one RTF output.

REFERENCES

Sanjay Matange, 2013. "Patient Profile Graphs Using SAS®" Proceedings of the SAS Global Forum 2013 Conference, Cary, NC: SAS Institute Inc. Available at <https://support.sas.com/resources/papers/proceedings13/160-2013.pdf>

CONTACT INFORMATION

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

Mingyue Zhu
Everest Clinical Research
Mingyue.zhu@ecrscorp.com

APPENDIX

```
%macro m_profile(  
datapath=, /*To specify the path of final data set*/  
filepath=, /*To specify the path of metadata configure excel file*/  
extrapath=, /*To specify the path of programs which will be include to  
generated special output*/  
outpath=/*To specify the path where the output generated should be stored*/  
);  
  
proc template;  
  define style styles.test;  
  parent=styles.rtf;  
  style body from document /  
    leftmargin=0.98 in  
    rightmargin=0.98 in  
    topmargin=.29 in  
    bottommargin=.29 in;  
  
  style systemtitle /  
    foreground=black  
    font_face="Arial"  
    font_size=10pt  
    font_style=ROMAN  
    protectspecialchars=off;  
  
  style systemfooter /  
    foreground=black  
    font_face="Arial"  
    font_size=10pt  
    font_style=ROMAN;  
  
  style data /  
    foreground=black  
    font_face="Arial"  
    font_weight=medium  
    font_size=10pt  
    font_style=ROMAN  
    protectspecialchars=off;  
  
  style header /  
    foreground=black  
    background=light grey  
    font_face="Arial"
```

```

        font_weight=medium
        font_size=10pt
        font_style=ROMAN
        borderbottomwidth=2
        borderbottomcolor=black
        bordertopwidth=2
        bordertopcolor=black
        textalign=left
        protectspecialchars=off;
style table /
    font_face      = "Arial"
    font_size      = 9pt
    foreground     = black
    background     = white
    cellspacing    = 0
    cellpadding    = 2
    borderwidth    = 1
    bordercolor    = black
    frame          = box
    rules          = all;
style GraphFonts from graphfonts/
    'GraphDataFont' = ("Times Roman",10pt)
    'GraphValueFont' = ("Times Roman",10pt)
    'GraphLabel2Font' = ("Times Roman",10pt)
    'GraphLabelFont' = ("Times Roman",10pt);
end;
run;

/** Generate macro variable for each subject **/
proc sql noprint;
    select count(*) into :num_obs
    from dm;
quit;
%let tot=&num_obs;
proc sql noprint;
    select subjid into :sub1-:sub&tot
    from dm;
quit;

/**Improt the metedata configure excel file**/
proc import out=dataset
    datafile="%filepath..xlsx"
    dbms=xlsx replace;
    getnames=yes;
    sheet= "report";
run;

data dataset;
    set dataset(where=(run='Y'));
    width=strip(width);
    if width='' then
width='10#10#10#10#10#10#10#10#10#10#10#10#10#10#10#10#10#10#10#10#10#10#10#10#';
    titnum=count(title,'#')+1;
    fotnum=count(foot,'#')+1;
    colorn=count(highlight,'|')+1;
    if idnum=. then idnum=1;
run;

```

```

proc sql noprint;
  select count(*) into:allds
  from dataset;
quit;
%let allds=&allds;

/**Generate macro variable for each data set and adjusting output format**/
proc sql noprint; select
  dataset,
  title,
  titnum,
  foot,
  fotnum,
  width,
  idnum,
  highlight,
  colorn,
  include

  into

  :dsn1-:dsn&allds ,
  :tit1-:tit&allds,
  :titn1-:titn&allds,
  :foot1-:foot&allds,
  :footn1-:footn&allds,
  :wth1-:wth&allds,
  :idn1-:idn&allds,
  :light1-:light&allds,
  :lightn1-:lightn&allds,
  :incl-:inc&allds

  from dataset;
quit;

%let run=%sysfunc(putn(%sysfunc(date()),date9.));
%let title1=Patient Profile Test Macro;

/**Start subject level loop and generate RTF for each subject**/
%do i=1 %to &tot;
ods rtf file="&outpath./&title1._Patient_Profile_&&sub&i...rtf"
style=styles.test startpage=no nogtitle nogfoot;

/**Start data set level loop**/
  %do k=1 %to &allds;

/**Include final data sets**/
  libname pp "&datapath";

  data &&dsn&k;
    set pp.&&dsn&k;
  run;

  proc contents data=&&dsn&k out=dss( keep=name varnum) noprint;
  run;

```

```

proc sort data=dss (where=(upcase(name) ^= 'SUBJID'));
  by varnum;
run;

proc sql noprint;
  select count(*),name into:allv,:varlist separated by ' '
  from dss;
quit;
%let allv=&allv;
%let varlist=&varlist;

/**Check if the subject in the form for 'no records'*/
proc sql noprint;
  select count(*) into :num_obs
  from &&dsn&k(where=(subjid="&&sub&i"));
quit;
%if &num_obs>0 %then %do;
  data report;
    set &&dsn&k(where=(subjid="&&sub&i"));
    pagn=1;
  run;
%end;
%else %do;
  data report;
    if 0 then set &&dsn&k;
    pagn=1;
  run;
%end;

%if %length(&&inc&k) %then %do;
  title; footnote;
  title1 j=right "&title1";
  title2 j=right "Patient &&sub&i.";
  title3 "&&tit&k";
  footnotel j=left "Updated: &run." j=right "Page ~{thispage} of
~{lastpage}";
  %include "&extrapath./&&inc&k...sas";
%end;
%else %do;
  title; footnote;
  title1 j=right "&title1";
  title2 j=right "Patient &&sub&i.";
  title3 " ";
  footnotel j=left "Updated: &run." j=right "Page ~{thispage} of
~{lastpage}";

  proc report data= report split='#' spacing=0 nowindows norkeys
style(report)=[cellspacing=0 width=100pct] style(header)=[just=left];
    column ((pagn &varlist));
    define pagn / order noprint;

    /**Start variable level loop*/
    %do j=1 %to &&idn&k;
      define %scan(&varlist,&j,%str( ))/left
style(column)=[width=%scan(&&wth&k...,1,#)%str(%%) just=left] id;
    %end;

```

```

%do j=%eval(&&idn&k+1) %to &allv;
  %if %index(%scan(&&wth&k.,&j.,#),page) %then %do;
    %let withpage=%scan(&&wth&k.,&j.,#);
    define %scan(&varlist,&j,%str( ))/ left
style(column)=[width=%scan(&withpage.,1,page)%str(%) just=left] page;
    %end;
  %else %if %index(%scan(&&wth&k.,&j.,#),page)=0 %then %do;
    define %scan(&varlist,&j,%str( ))/ left
style(column)=[width=%scan(&&wth&k.,&j.,#)%str(%) just=left];
    %end;
  %end;

%if &num_obs eq 0 %then %do;
  compute after pagn;
  line "No Records.";
  line " ";
endcomp;
%end;

/**For high light part**/
%if %length(&&light&k) %then %do;

  %if %index(&&light&k,|)=0 %then %do;

    %let _var_&k.   =%scan(&&light&k.,1,#);
    %let _con_&k.   =%scan(&&light&k.,2,#);
    %let _color_&k. =%scan(&&light&k.,3,#);

    compute &&_var_&k.;
    if &&_con_&k. then do;
      call define("&&_var_&k.", "style",
"style=[background=&&_color_&k.]");
    end;
    endcomp;
  %end;
  %else %do;
    %do h=1 %to &&lightn&k;
      %let _light_   =%scan(&&light&k.,&h,|);

      %let _var_     =%scan(&_light_.,1,#);
      %let _con_     =%scan(&_light_.,2,#);
      %let _color_   =%scan(&_light_.,3,#);

      compute &_var_;
      if &_con_ then do;
        call define("&_var_", "style",
"style=[background=&_color_.]");
      end;
      endcomp;
    %end;
  %end;

%end;

/**For title**/
%if %length(&&tit&k) %then %do;
  compute before _page_/style={just=center};

```



```

        %if %index(&&tit&k,#)=0 %then %do;
            line "&&tit&k";;
        %end;
        %else %do;
            %do t=1 %to &&titn&k;
                %let tit=%scan(&&tit&k..,&t.,#);
                line "&tit";;
            %end;
        %end;
    endcomp;
%end;

/**For footnote**/
%if %length(&&foot&k) %then %do;
    compute after _page_/style={just=left};
    %if %index(&&foot&k,#)=0 %then %do;
        line "&&foot&k";;
    %end;
    %else %do;
        %do f=1 %to &&footn&k;
            %let fot=%scan(&&foot&k..,&f.,#);
            line "&fot";;
        %end;
    %end;
endcomp;
%end;

run;
%end;

%end;
ods rtf close;

%end;
%mend;

```