PharmaSUG2011 – PO14

Automatization of Patient Characteristics Report

Mirjana Stojanovic
Cancer Center Biostatistics, Duke University Medical Center, Durham, NC

ABSTRACT

This paper describes a standardized and presentable patient characteristics report. In an effort to standardize the appearance of the patient characteristics' report our team decided to create one macro that will produce patient characteristics tables for all studies. This macro provides an easy method for producing standardized and nicely formatted tables. It uses the same logic and structure across all studies. In addition the macro automatically computes all stratification parameters based on number of strata and number of levels within each strata. This paper will outline the features of the macro including examples of code and output. This paper is intended for programmers with a sound foundation in SAS macro programming.

INTRODUCTION

Our goal of developing application was to automate computing of stratification parameters. This was the main contribution in developing patient characteristics macro. In implementation of this macro, we have developed and used a specification file in which the user indicates the stratums and levels per each stratum. Driver program includes macro, specification file and requires changes in a few lines only. So this macro is flexible, user friendly and is producing output in RTF format. SAS® Macro Facility enabled macro to produce flexible and dynamic SAS code expanding vertically and horizontally per user's demand. Macro is working for all studies with no intervention into program code.

What did we do to automate the process of reporting patient characteristics for all studies? We already developed code for computing and presenting results for Age, Sex, Race, Ethnicity, and Patient Performance. It allows us to concentrate on development of stratum parts in the macro. How we accomplished this task would be the main emphasis in the rest of this paper. The assumption for design of the macro was (based on specific needs of our department) to work up to **three** statums and up to **four** levels per each stratum.

Development of PAT_CHAR macro

Development was done in several steps.

Step1: Determines the total number of combinations for stratum/levels. Macro variables STRATUM1, STRATUM2, and STRATUM3 get their values in specification file as well as macro variables S1, S2, and S3 (Each value of S1, S2, S3 means number of levels per each of three stratums).

```
data _null_ ;
      * Total is max number of stratum combinations. ;
      total=&s1.*&s2.*&s3.;
      length tot $ 3 ;
      tot=put(total, 3.) ;
      call symput('total',left(tot)) ;
run ;
```

Step 2: Determines the size and dimension of array for further processing;
* Generating array with all stratum combinations;
* Variables STRxx_ are auxiliary variables necessary for generating stratum strings;

```
data _null_ ;
length str11_ str12_ str13_ str14_ str21_ str22_ str23_ str24_ str31_ str32_ str33_
str34_  $ 97 ;
array aux(12) str11_ str12_ str13_ str14_ str21_ str22_ str23_ str24_ str31_ str32_
str33_ str34_ ;
      do ii = 1 to dim(aux) ;
         aux(ii) = ' ' ;
      end ;
      array str(&total.) $ 3;
      seq = 0 ;
      do stratum1 = 1 to &s1. ;
    %IF "&STRATUM2." = "1" %THEN %DO; do stratum2 = 1 to &s2. ; %END ;
     %IF "&STRATUM3." = "1" %THEN %DO; do stratum3 = 1 to &s3. ; %END ;
       seq + 1 ;
       str(seq) = put(stratum1,Z1.) %IF "&STRATUM2." = "1" %THEN %DO; ||put(stratum2,Z1.) %END;
                                    %IF "&STRATUM3." = "1" %THEN %DO; ||put(stratum3,Z1.) %END ; ;
       put str(seq)= ;
     %IF "&STRATUM3." = "1" %THEN %DO; end ; %END ;
    %IF "&STRATUM2." = "1" %THEN %DO; end ; %END ;
   end ;

do j = 1 to dim(str) ;
   if substr(str(j), 1, 1) = '1' then str11_=compress(str11_) ||
trim(left(put(j,2.))) ||',';
   if substr(str(j), 1, 1) = '2' then str12_=compress(str12_) ||
trim(left(put(j,2.))) ||',';
   if substr(str(j), 1, 1) = '3' then str13_=compress(str13_) ||
trim(left(put(j,2.))) ||',';
   if substr(str(j), 1, 1) = '4' then str14_=compress(str14_) ||
trim(left(put(j,2.))) ||',';
   %IF "&STRATUM2." = "1" %THEN %DO;
      if substr(str(j), 2, 1) = '1' then str21_=compress(str21_) ||
trim(left(put(j,2.))) ||',';
      if substr(str(j), 2, 1) = '2' then str22_=compress(str22_) ||
trim(left(put(j,2.))) ||',';
      %IF "&S2." > "1" %THEN %DO;
          if substr(str(j), 2, 1) = '3' then str23_=compress(str23_) ||
trim(left(put(j,2.))) ||',';
          if substr(str(j), 2, 1) = '4' then str24_=compress(str24_) ||
trim(left(put(j,2.))) ||',';
      %END;
   %END;
   %IF "&STRATUM3." = "1" %THEN %DO;
       if substr(str(j), 3, 1) = '1' then str31_=compress(str31_) ||
trim(left(put(j,2.))) ||',';
       if substr(str(j), 3, 1) = '2' then str32_=compress(str32_) ||
trim(left(put(j,2.))) ||',';
       %IF "&S3." > "1" %THEN %DO;
           if substr(str(j), 3, 1) = '3' then str33_=compress(str33_) ||
trim(left(put(j,2.))) ||',';
           if substr(str(j), 3, 1) = '4' then str34_=compress(str34_) ||
trim(left(put(j,2.))) ||',';
       %END;
    %END;
end ;
```

Step 3: Removing the comma at the very end of each **str11_ to str34_** string (code is not presented here).

Step 4: Producing corresponding STRxx_ macro variables.
We assume that there are at least two stratums in each study. With conditional **%IF %THEN %DO** SAS Macro Facility produces just the right number of statements after resolving macro code.

```
      call symput('str11_', str11_) ;
      call symput('str12_', str12_) ;
      %IF "&S1." = "3" %THEN %DO;
         call symput('str13_', str13_) ;
      %END;
      %IF "&S1." = "4" %THEN %DO;
         call symput('str13_', str13_) ;
         call symput('str14_', str14_) ;
      %END;
      %IF "&STRATUM2." = "1" %THEN %DO;
```

```
         call symput('str21_', str21_) ;
         call symput('str22_', str22_) ;
         %IF "&S2." = "3" %THEN %DO;
            call symput('str23_', str23_) ;
         %END;
         %IF "&S2." = "4" %THEN %DO;
            call symput('str23_', str23_) ;
            call symput('str24_', str24_) ;
         %END;
      %END;
```

Note: The third stratum macro code is similar and thus not presented here.

The results for example with 3 stratums and 2 levels after generating and resolving macro variables are outlined below.

Total=8 (number of combinations of stratums and levels)

Seqno	StrComb	STRxx_	
1	str1=111	str11_=1,2,3,4	(first stratum first level)
2	str2=112	str12_=5,6,7,8	(first stratum second level)
3	str3=121	str21_=1,2,5,6	(second stratum first level)
4	str4=122	str22_=3,4,7,8	(second stratum second level)
5	str5=211	str31_=1,3,5,7	(third stratum first level)
6	str6=212	str32_=2,4,6,8	(third stratum second level)
7	str7=221		
8	str8=222		

Step 5: Using macro variables **&strxx_** that were generated in the prior data step.
Again after resolving macro code, the correct number of SAS statements were generated so it is possible to have SAS code when there are 2, 3, or 4 levels. By using **SELECT** block statement with **WHEN** each observation from input data set PharmaSUG2011 (example dataset) gets correct value for each **STRATUM**. Argument in **WHEN** statement is **&strxx_** macro variable with value assigned in prior data step.

```
data master ;
  set db.PharmaSUG2011(drop=performance_id);
  where schedule_id=&schedule_id.;

  * Other code unimportant for STRATUMs ;

  %IF "&STRATUM1." = "1" %THEN %DO;
     select (stratum_grp_id);
       when (&str11_.) stra1=1;
       when (&str12_.) stra1=2;
            %IF "&S1." = "3" %THEN %DO;
              when (&str13_.) stra1=3;
            %END;
            %IF "&S1." = "4" %THEN %DO;
              when (&str13_.) stra1=3;
              when (&str14_.) stra1=4;
            %END;
       OTHERWISE stra1 = . ;
     end;
  %END;
run;
```

Note: Since the other two stratum codes are similar as the above, they were not presented.

Step 6: Obtaining counts and percents for the first Arm column in report.
For each stratum defined in specification file frequency and percentage is computed after resolving of macro code in the lines below. In that way we have great flexibility in building SAS program with no intervention in real code.

```
proc freq data=master;
  %IF "&STRATUM1." = "1" %THEN %DO;
     table stra1  /  missing list out=Stratum1_arm1(rename=(count=cnt1
percent=pct1)) ;
  %END;
  %IF "&STRATUM2." = "1" %THEN %DO;
     table stra2  /  missing list out=Stratum2_arm1(rename=(count=cnt1
percent=pct1)) ;
  %END;
  %IF "&STRATUM3." = "1" %THEN %DO;
     table stra3  /  missing list out=Stratum3_arm1(rename=(count=cnt1
percent=pct1)) ;
  %END;
 where schedule_id=&schedule_id.;
run;
```

Note: For all other ARMs code are similar to the above code.

Step 7: Merging together data for **STRATUMs** per each ARM with other variables**.**

```
%IF "&STRATUM1." = "1" %THEN %DO;
   data Stratum1_arm(keep=stra1 cnt1 pct1 %IF "&ARM." = "2" %THEN %DO; cnt2 pct2
%END;
                                          %IF "&ARM." = "3" %THEN %DO; cnt2 pct2
                                        cnt3 pct3 %END;
                                          %IF "&ARM." = "4" %THEN %DO; cnt2 pct2
cnt3 pct3 cnt4 pct4 %END;) ;
      merge Stratum1_arm1 %IF "&ARM." = "2" %THEN %DO; Stratum1_arm2 %END;
                          %IF "&ARM." = "3" %THEN %DO; Stratum1_arm2 Stratum1_arm3
%END;
                          %IF "&ARM." = "4" %THEN %DO; Stratum1_arm2 Stratum1_arm3
Stratum1_arm4 %END;;
      by stra1 ;
      if cnt1 = . then cnt1 = 0 ;
      if pct1 = . then pct1 = 0 ;
   %IF "&ARM." = "2" %THEN %DO;
      if cnt2 = . then cnt2 = 0 ;
      if pct2 = . then pct2 = 0 ;
   %END;
   %IF "&ARM." = "3" %THEN %DO;
      if cnt2 = . then cnt2 = 0 ;
      if pct2 = . then pct2 = 0 ;
      if cnt3 = . then cnt3 = 0 ;
      if pct3 = . then pct3 = 0 ;
   %END;
   %IF "&ARM." = "4" %THEN %DO;
      if cnt2 = . then cnt2 = 0 ;
      if pct2 = . then pct2 = 0 ;
      if cnt3 = . then cnt3 = 0 ;
      if pct3 = . then pct3 = 0 ;
      if cnt4 = . then cnt4 = 0 ;
      if pct4 = . then pct4 = 0 ;
   %END;
run ;
%END;
```

Step 8: STRATUM1, STRATUM2, and STRATUM3 are data sets for each stratum variable. Each is generated by merging **StratumX_arm StratumX_T** (X = 1, 2, or 3).

```
%IF "&STRATUM1." = "1" %THEN %DO;
data Stratum1 ;
   length sort_id 3 pat_char $ 16 ;
   merge Stratum1_arm Stratum1_T ;
   by stra1 ;
   sort_id = 6 ;
   pct1 = round(pct1) ;
   pct2 = round(pct2) ;
   pct3 = round(pct3) ;
   pct4 = round(pct4) ;
   if cntT = . then cntT = 0 ;
   if pctT = . then pctT = 0 ;
   pctT = round(pctT) ;
   pat_char = left(put(stra1, stra1f.)) ;
   drop stra1 ;
run ;
%END;
```

Note: Code for producing STRATUM2 and STRATUM3 SAS data sets are similar and thus not presented here.

Step 9: Put all individual data sets into one (Total) for final processing.

```
data total ;
   set age sex rac eth
     %IF "&PS." = "1" %THEN %DO; ps %END;
     %IF "&STRATUM1." = "1" %THEN %DO; Stratum1 %END;
     %IF "&STRATUM2." = "1" %THEN %DO; Stratum2 %END;
     %IF "&STRATUM3." = "1" %THEN %DO; Stratum3 %END;
     ;
run ;
```

Step 10: Proc Report with ODS is used for presenting result in RTF format.
As mentioned previously, the macro is capable of handling 3 stratums with up to 4 levels and up to 4 ARMs.
Request was that column for total (ArmT) should not be present if there is only 1 ARM.

```
ods rtf file="&Report_Location."  ;
proc report data=total nowd split='~' ;
TITLE "2.2    PATIENT CHARACTERISTICS  -  Study &study_num." ;
columns pat_char1 ArmA %IF "&ARM." = "2" %THEN %DO ; ArmB %END;
                       %IF "&ARM." = "3" %THEN %DO ; ArmB ArmC %END;
                       %IF "&ARM." = "4" %THEN %DO ; ArmB ArmC ArmD %END;
                       %IF "&ARM." > "1" %THEN %DO ; ArmT %END; ;
define pat_char1 / display width=20 " " ;
define ArmA      / display width=12 "Arm A~N=&ArmA." right;
%IF "&ARM." = "2" %THEN %DO ;
define ArmB      / display width=12 "Arm B~N=&ArmB." right;
%END;
%IF "&ARM." = "3" %THEN %DO ;
define ArmB      / display width=12 "Arm B~N=&ArmB." right;
define ArmC      / display width=12 "Arm C~N=&ArmC." right;
%END;
%IF "&ARM." = "4" %THEN %DO ;
define ArmB      / display width=12 "Arm B~N=&ArmB." right;
define ArmC      / display width=12 "Arm C~N=&ArmC." right;
define ArmD      / display width=12 "Arm D~N=&ArmD." right;
%END;
%IF "&ARM." > "1" %THEN %DO ;
define ArmT      / display width=12 "Total~N=&ArmT." right;
%END;
run ;
```

```
footnote "All data in report above are completely fabricated!" ;
```

Statisticians have to do two things before using PAT_CHAR macro. They need to fill out spec file and to make small changes in the few lines in the driver program.

Filling out spec file:

```
libname db  'H:\pharmasug\PharmaSUG2011' ;
%let cutoff_date=mdy(07,11,2010);
%let study_num=XXXXX;
%let schedule_id=2;
%let arm=3;
%let ps=1;
%let stra_char1 =Therapy:;
%let stra_char2 =Prior therapy 1:;
%let stra_char3 =Prior therapy 2:;
%let stratum1=1; %let s1=2; * Levels for first stratum;
%let stratum2=1; %let s2=2; * Levels for second stratum;
%let stratum3=1; %let s3=2; * Levels for third stratum;
proc format ;
  value stra1f 1="A"   2="B" ;
  value stra2f 1="No" 2="Yes";
  value stra3f 1="No" 2="Yes";
run ;
```

End user – statistician can modify just the **bolded** lines below.

```
/* Here will be included specification file with %INCLUDE statement.*/
%include   'H:\PharmaSUG\PharmaSUG2011\spec_file.txt' / source2 ;

%include 'H:\PharmaSUG\PharmaSUG2011\pat_char.mac' ;

* Report Location and name of RTF file is choice of statistician;

%pat_char(study_num=study_num,  Report_location=study_num_Pat_Char.rtf)
```

Patient Characteristics report with 3 stratums (2 levels per each stratum) and 3 ARMs can be seen in the appendix (the very last page).

CONCLUSION

Goal was to produce one macro for Patient Characteristics report. Now all statisticians in the department would get the patient characteristics report with the same (unified) structure and with variables requested by the study protocol.

REFERENCES

Carpenter, Arthur L. (2004), Carpenter's Complete Guide to the SAS Macro Language, Second Edition, Cary, NC: SAS Institute Inc.

Carpenter, Arthur and Smith, Richard (2002), "Library and File management: Building a Dynamic Application," *Proceedings of the Twenty-seventh Annual SAS Users Group International Conference*, Paper 21-27.

Peng, Fuping, and Perdomo, Carlos (2203) "A Modular Approach to Develop Patient Profile Application" PharmaSug 2003, paper Ad003.

Aster, Rick (2005), Professional SAS Programming shortcuts, 2nd edition, A Breakfast Book.

ACKNOWLEDGMENTS

I would like to thank Ms. Laura Archer MS, Statistician, Duke University Medical Center, NC for her advice and support in review of this paper.

CONTACT INFORMATION

Your comments are greatly appreciated and encouraged.

Contact the author at:

Mirjana Stojanovic
Duke University Medical Center
phone (919) 668-9337
E-mail: mirjana.stojanovic@duke.edu

TRADEMARK INFORMATION

PATIENT CHARACTERISTICS - Study XXXXX

	Arm A N=36	Arm B N=23	Arm C N=33	Total N=92
Age:				
20-29	0 (0%)	1 (4%)	0 (0%)	1 (1%)
30-39	3 (8%)	3 (13%)	4 (12%)	10 (11%)
40-49	4 (11%)	4 (17%)	3 (9%)	11 (12%)
50-59	9 (25%)	8 (35%)	13 (39%)	30 (33%)
60-69	10 (28%)	3 (13%)	7 (21%)	20 (22%)
70-79	8 (22%)	2 (9%)	5 (15%)	15 (16%)
80+	2 (6%)	2 (9%)	1 (3%)	5 (5%)
Sex:				
Male	18 (50%)	16 (70%)	14 (42%)	48 (52%)
Female	18 (50%)	7 (30%)	19 (58%)	44 (48%)
Race:				
White	34 (94%)	16 (70%)	26 (79%)	76 (83%)
Black	2 (6%)	4 (17%)	5 (15%)	11 (12%)
Oriental	0 (0%)	0 (0%)	1 (3%)	1 (1%)
Unknown	0 (0%)	3 (13%)	1 (3%)	4 (4%)
Ethnicity:				
Hispanic	3 (8%)	3 (13%)	5 (15%)	11 (12%)
Non-Hispanic	32 (89%)	17 (74%)	26 (79%)	75 (82%)
Unknown	1 (3%)	3 (13%)	2 (6%)	6 (7%)
PS:				
0	20 (56%)	14 (61%)	21 (64%)	55 (60%)
1	16 (44%)	9 (39%)	12 (36%)	37 (40%)
Therapy:				
A	26 (72%)	17 (74%)	27 (82%)	70 (76%)
B	10 (28%)	6 (26%)	6 (18%)	22 (24%)
Prior therapy 1:				
No	31 (86%)	19 (83%)	27 (82%)	77 (84%)
Yes	5 (14%)	4 (17%)	6 (18%)	15 (16%)
Prior therapy 2:				
No	33 (92%)	19 (83%)	32 (97%)	84 (91%)
Yes	3 (8%)	4 (17%)	1 (3%)	8 (9%)

Footnote: All data presented in this report are fully fabricated for presentation purposes!