PharmaSUG 2017 - Paper PO21

Maximum Use of TFL Mock Shells Metadata of Titles, Footnotes, Formatting, and Reporting Details

Julius Kirui, Sarah Cannon Development Innovations, Nashville, TN

ABSTRACT

Biostatisticians devote numerous hours of research time to the development of tables, figures, and listings (TFL) mock shells. The information provided by the mock shells include titles, footnotes, and formatting details, such as column labels, alignments, and widths. Statistical SAS programmers are tasked with transferring this information word by word into the final TFLs. Manual transfer of this metadata can be challenging, labor-intensive, and prone to errors. Several SAS users have written and shared their thoughts on how titles and footnotes can be adopted for TFL production with minimal manual involvement. This paper will discuss and provide a SAS 9.4 macro program to generate ready to use metadata of titles, footnotes, and TFL formatting details from a standard set of TFL mock shells. Also, TFL reporting details such as output labels and program names will be discussed. Automatic transfer of this metadata will eliminate major quality issues and formatting challenges, which in turn will free up more time for statistical programmers and biostatisticians to focus on data analysis.

INTRODUCTION

Often statisticians spend numerous hours developing and reviewing tables, figures, and listings (TFL) mock shells. The information provided by the mock shells include titles, footnotes, and formatting details, such as column labels, widths, and alignments. Statistical programmers are tasked with transferring this information to the final TFLs. Manual transfer of this metadata can be time consuming, tedious, and error prone. Major or minor changes to the mock shells require a programming team member to manually edit effected TFL-generating SAS programs; these edits may result in significant quality issues, such as typos, text omissions, erroneous deletions, changes not saved, changes applied to the wrong program or text, etc. Most TFL errors or reviewer comments are attributed to formatting or erroneous transfer of TFL mock shell metadata. This paper will discuss and provide a SAS 9.4 macro to generate ready to use metadata of titles, footnotes, and formatting details, including column labels, widths, and alignments, from a standard set of TFL mock shells. Also, TFL reporting details such as output labels and program names based on the parent mock shell will be discussed.

MACRO CALL

The initial set up of this macro requires very minimal SAS experience.

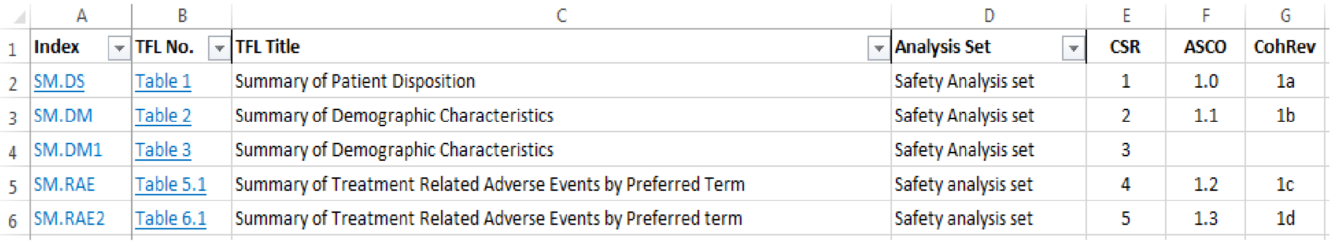
```
%tfdh(subset =, span =, type =, tflno =, filename =, path =);
```

Parameters	Notes
subset	Used for extracting required columns from the mock shell e.g subset in('F1','F2','F3','F4','F7'), skip ‘F5’ and ‘F6’ if not needed for a particular deliverable.
span	Number of columns to span over. This parameter should not be blank if subset macro parameter above is not blank.
type	Used for extracting metadata for desired deliverable e.g CSR, ASCO, Cohort review, etc
tflno	Used for altering TFL numbering criteria. Set the parameter to NEW if new TFL numbers are needed, else set to blank.
filename	Name of TFL mock shell file name e.g TFL_Shells.xlsx
path	The path to import the TFL mock shells and to store resulting metadata.

Macro parameters ***subset***, ***span***, ***type*** and ***tflno*** may require frequent updates depending on the desired deliverables and changes in the TFL numbering criteria.

REQUIREMENTS

TFL mock shells are created in Excel and stored as an Excel workbook (.xlsx) file. The first Excel sheet (labeled as the index page) is reserved for the table of contents (see below). Each of the Excel sheets following the index page will be populated with TFL mock shells (one mock shell per sheet).



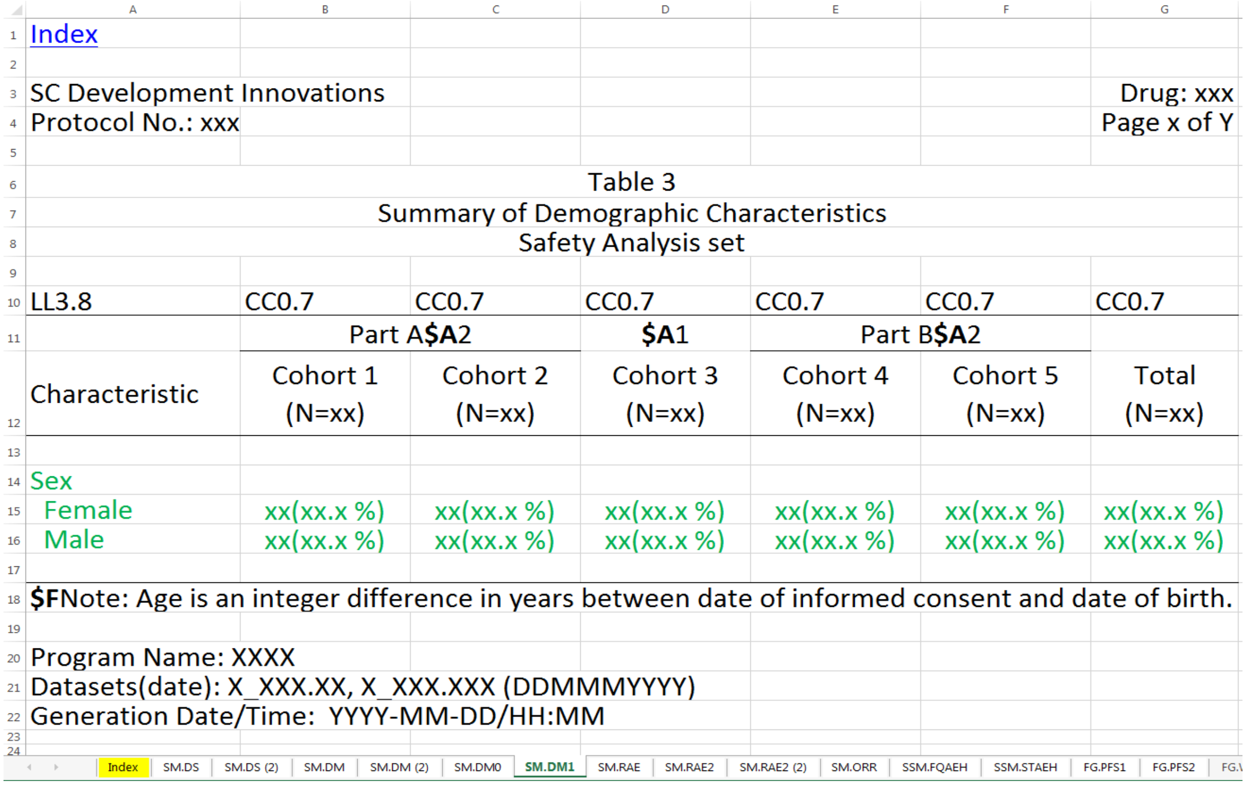
Index	TFL No.	TFL Title	Analysis Set	CSR	ASCO	CohRev
SM.DS	Table 1	Summary of Patient Disposition	Safety Analysis set	1	1.0	1a
SM.DM	Table 2	Summary of Demographic Characteristics	Safety Analysis set	2	1.1	1b
SM.DM1	Table 3	Summary of Demographic Characteristics	Safety Analysis set	3		
SM.RAE	Table 5.1	Summary of Treatment Related Adverse Events by Preferred Term	Safety analysis set	4	1.2	1c
SM.RAE2	Table 6.1	Summary of Treatment Related Adverse Events by Preferred term	Safety analysis set	5	1.3	1d

For the purposes of this program, column A through D should be labeled as shown above. Columns following column D can be labelled as needed by the user. These additional columns may be used to store further TFL identifiers for unique study deliverables. The contents of these identifiers can be used to alter the numbering criteria, by concatenating the text part of column B and any preferred column after column D (see macro call parameters ***type*** and ***tflno***).

```
set XLSLIB."&toc$"n(where=(&type ^=''));
 %if &tflno=NEW %then %do; /*&tflno=NEW, use new numbering criteria*/
           yy=compress(TFL_No_,,'ka')||' '||compress(&type);%end;
  %else %if &tflno= %then %do;yy=TFL_No_;%end;
```

The program extract above illustrates how an additional column after D can be used to subset and change the numbering criteria of a desired TFL deliverable.

Below is an example of a standard TFL mock shell in Excel.



	A	B	C	D	E	F	G
1	Index						
2							
3	SC Development Innovations						Drug: xxx
4	Protocol No.: xxx						Page x of Y
5							
6				Table 3			
7				Summary of Demographic Characteristics			
8				Safety Analysis set			
9							
10	LL3.8	CC0.7	CC0.7	CC0.7	CC0.7	CC0.7	CC0.7
11		Part A$A2		$A1	Part B$A2		
12	Characteristic	Cohort 1 (N=xx)	Cohort 2 (N=xx)	Cohort 3 (N=xx)	Cohort 4 (N=xx)	Cohort 5 (N=xx)	Total (N=xx)
13							
14	Sex						
15	Female	xx(xx.x %)	xx(xx.x %)	xx(xx.x %)	xx(xx.x %)	xx(xx.x %)	xx(xx.x %)
16	Male	xx(xx.x %)	xx(xx.x %)	xx(xx.x %)	xx(xx.x %)	xx(xx.x %)	xx(xx.x %)
17							
18	$FNote: Age is an integer difference in years between date of informed consent and date of birth.						
19							
20	Program Name: XXXX						
21	Datasets(date): X_XXX.XX, X_XXX.XXX (DDMMYYYY)						
22	Generation Date/Time: YYYY-MM-DD/HH:MM						
23							
24							

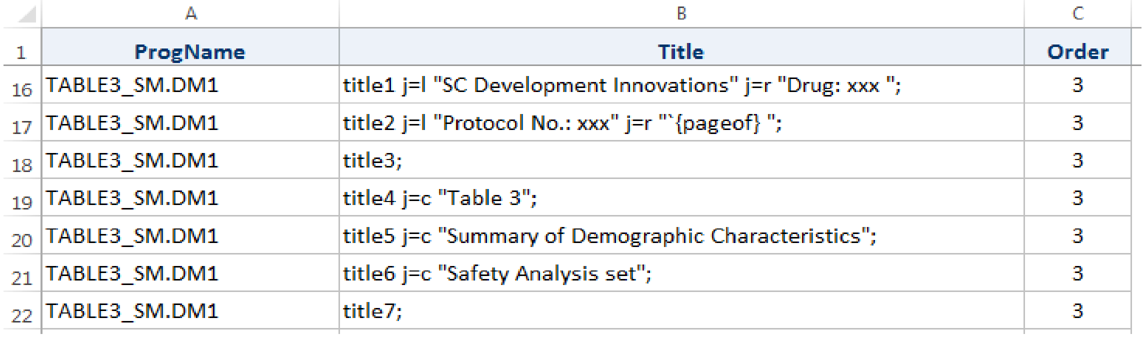
MACRO EXECUTION AND RESULTS

TITLES

Rows 3 to 9 of the standard mock shell are reserved for titles. This program assumes that the titles in rows 3 and 4 are primarily left aligned with some information possibly aligned to the right. The titles in rows 5 to 9 are aligned to the center. The program extract below can be edited if alternate title alignments are desired.

```
if _n_ le 8 and _n_ gt 1  then do;
 if _n_ in(2,3)  and index ^='' then title='title'||strip(put((_n_-1),2.))||
        ' j=l "'||strip(index)||'" j=r "'||catx(' ',of F2-F%trim(&f))||' ";';
 else if _n_ ^in(2,3)  and index ^='' then title =
             'title'||strip(put((_n_-1),2.)) ||' j=c "'||strip(index)||'";';
 else if _n_ ^in(2,3)  and index ='' then title =
             'title'||strip(put((_n_-1),2.))|| ';';end;
```

Note: Row 1 of the mock shell is deleted as soon as the shell is imported to SAS.



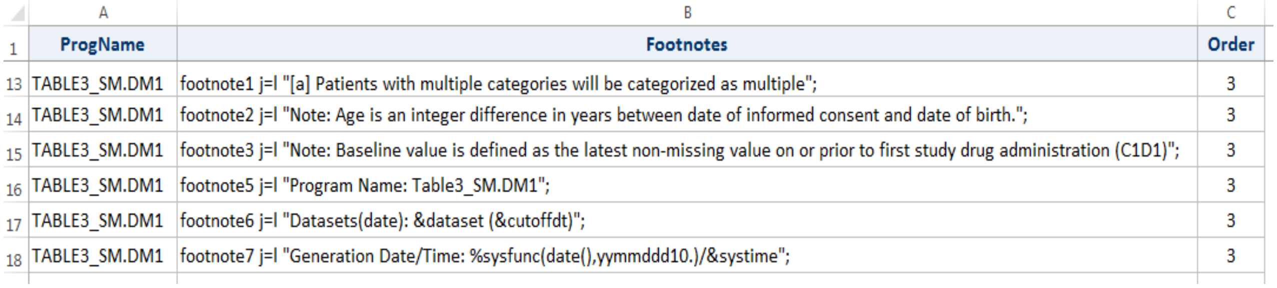
	A	B	C
1	ProgName	Title	Order
16	TABLE3_SM.DM1	title1 j=l "SC Development Innovations" j=r "Drug: xxx ";	3
17	TABLE3_SM.DM1	title2 j=l "Protocol No.: xxx" j=r "`{pageof} ";	3
18	TABLE3_SM.DM1	title3;	3
19	TABLE3_SM.DM1	title4 j=c "Table 3";	3
20	TABLE3_SM.DM1	title5 j=c "Summary of Demographic Characteristics";	3
21	TABLE3_SM.DM1	title6 j=c "Safety Analysis set";	3
22	TABLE3_SM.DM1	title7;	3

FOOTNOTES

The primary row associated with the first footnote depends on the number of rows occupied by the body section of the mock shell. Due to this inconsistency the first footnote will be preceded by **$F**. For the purpose of this macro, the last three footnotes will remain as illustrated by the mock shell extract shown previously. The program extract below can be updated as desired.

```
if substr(F1,1,2)='$F' then do;
  call symput("ffot",put(_n_,3.));/*_n_ row associated with the 1st footnote*/
  call symput("flen",input(length(F1),best.));end;
if scan(upcase(F1),1,'/') ='GENERATION DATE' then do;
  call symput("efot",put(_n_,3.));/*_N_ row associated to last footnote*/
F1=strip(scan(F1,1,':'))||':%sysfunc(date(),yymmddd10.)/&systime';end;
if scan(F1,1,':')= 'Datasets(date)’  then F1=
  strip(scan(F1,1, ':'))|| ':&dataset (&cutoffdt)'; /*to be assigned later*/
if scan(F1,1,':')='Program Name' then F1=strip(scan(F1,1,':'))
    ||": &sid"||"_&&tfl&i";/*&sid=Table No. and _&&tfl&i =excel sheet name*/
```

Note: This macro assumes that all footnotes are aligned to the left (left justified).



	A	B	C
1	ProgName	Footnotes	Order
13	TABLE3_SM.DM1	footnote1 j=l "[a] Patients with multiple categories will be categorized as multiple";	3
14	TABLE3_SM.DM1	footnote2 j=l "Note: Age is an integer difference in years between date of informed consent and date of birth.";	3
15	TABLE3_SM.DM1	footnote3 j=l "Note: Baseline value is defined as the latest non-missing value on or prior to first study drug administration (C1D1)";	3
16	TABLE3_SM.DM1	footnote5 j=l "Program Name: Table3_SM.DM1";	3
17	TABLE3_SM.DM1	footnote6 j=l "Datasets(date): &dataset (&cutoffdt)";	3
18	TABLE3_SM.DM1	footnote7 j=l "Generation Date/Time: %sysfunc(date(),yymmddd10.)/&systime";	3

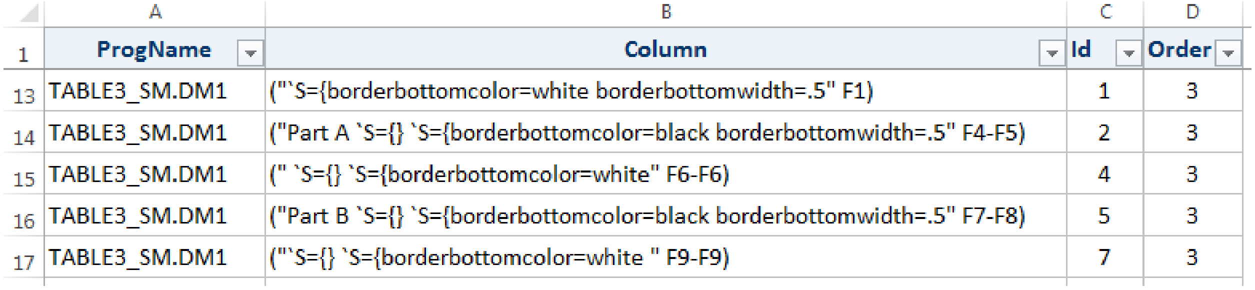
FORMATTING DETAILS: COLUMN LABELS, ALIGNMENTS, AND WIDTHS

Column labels and spanning headers are populated on rows 10 and 11 of the standard mock shell. Column labels for some mock shells may end with (N=xx) to indicate subject population, i.e big N. Row 10 is reserved for spanning headers endding with $A#, where # represents the number of columns to span over. Special character tilde (~) specifies the point at which the column label can be split.

Column alignment and widths are populated on row 9 of the standard mock shell. Alignments are specified by the first two letters, for example: C = center and L = left. The first letter aligns the column label and the second letter aligns the final data columns, and the digits following these letters represents the column widths.

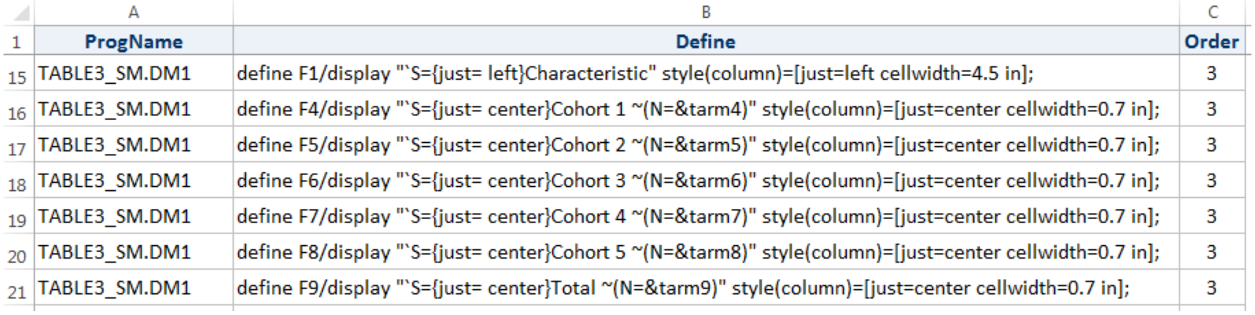
Statisticians assign random names to Excel sheets populated with the TFL mock shells. For the purposes of this program, these names will be preceded by SM., SSM., FG., and LS. for summary tables, shift summary table, figures, and listings respectively. SM. commands this program to skip two columns, typically F2 and F3, before the first column with N=xx. The two skipped columns will act as security for future programming and changes to TFL formats.

Below are extracts of column and define statements respectively.



	A	B	C	D
1	ProgName	Column	Id	Order
13	TABLE3_SM.DM1	("`S={borderbottomcolor=white borderbottomwidth=.5" F1)	1	3
14	TABLE3_SM.DM1	("Part A `S={} `S={borderbottomcolor=black borderbottomwidth=.5" F4-F5)	2	3
15	TABLE3_SM.DM1	(" `S={} `S={borderbottomcolor=white" F6-F6)	4	3
16	TABLE3_SM.DM1	("Part B `S={} `S={borderbottomcolor=black borderbottomwidth=.5" F7-F8)	5	3
17	TABLE3_SM.DM1	("`S={} `S={borderbottomcolor=white " F9-F9)	7	3

Distinct column statement (column B) will be extracted by order (column D) and sorted by id (column C) before being assigned to an independent macro variable.



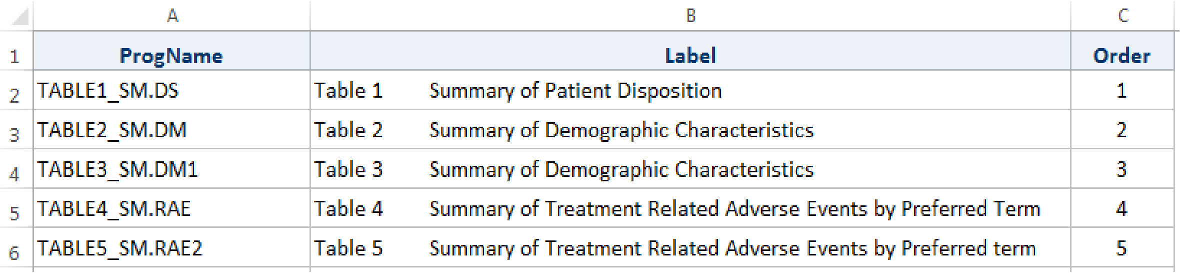
	A	B	C
1	ProgName	Define	Order
15	TABLE3_SM.DM1	define F1/display "`S={just= left}Characteristic" style(column)=[just=left cellwidth=4.5 in];	3
16	TABLE3_SM.DM1	define F4/display "`S={just= center}Cohort 1 ~(N=&tarm4)" style(column)=[just=center cellwidth=0.7 in];	3
17	TABLE3_SM.DM1	define F5/display "`S={just= center}Cohort 2 ~(N=&tarm5)" style(column)=[just=center cellwidth=0.7 in];	3
18	TABLE3_SM.DM1	define F6/display "`S={just= center}Cohort 3 ~(N=&tarm6)" style(column)=[just=center cellwidth=0.7 in];	3
19	TABLE3_SM.DM1	define F7/display "`S={just= center}Cohort 4 ~(N=&tarm7)" style(column)=[just=center cellwidth=0.7 in];	3
20	TABLE3_SM.DM1	define F8/display "`S={just= center}Cohort 5 ~(N=&tarm8)" style(column)=[just=center cellwidth=0.7 in];	3
21	TABLE3_SM.DM1	define F9/display "`S={just= center}Total ~(N=&tarm9)" style(column)=[just=center cellwidth=0.7 in];	3

Subject population counts will be assigned to ***&tarm*** macro variables at the TFL programming level. Summary variable names to be reported will be preceded by F and variables for empty columns (the white space after the spanning header) will be preceded by b.

REPORTING DETAILS: OUTPUT LABELS AND PROGRAM NAMES

Output labels and program names are created from the table of contents shown previously, by concatenating columns B & C and columns C & A, respectively.

Note: Special characters such as (<, >,?, etc.) should be avoided.



	A	B	C
1	ProgName	Label	Order
2	TABLE1_SM.DS	Table 1 Summary of Patient Disposition	1
3	TABLE2_SM.DM	Table 2 Summary of Demographic Characteristics	2
4	TABLE3_SM.DM1	Table 3 Summary of Demographic Characteristics	3
5	TABLE4_SM.RAE	Table 4 Summary of Treatment Related Adverse Events by Preferred Term	4
6	TABLE5_SM.RAE2	Table 5 Summary of Treatment Related Adverse Events by Preferred term	5

TFL output labels (column B) will be extracted by distinct order (column C) and assigned to an independent macro variable (see below).

```
select label into :outlabel separated by ' `
 from out.tfdh where order="3" ;/*program name or text after _ can be used*/
```

Note: Above sql program snapshot can be repeated to extract titles, footnotes, define, and column statements.

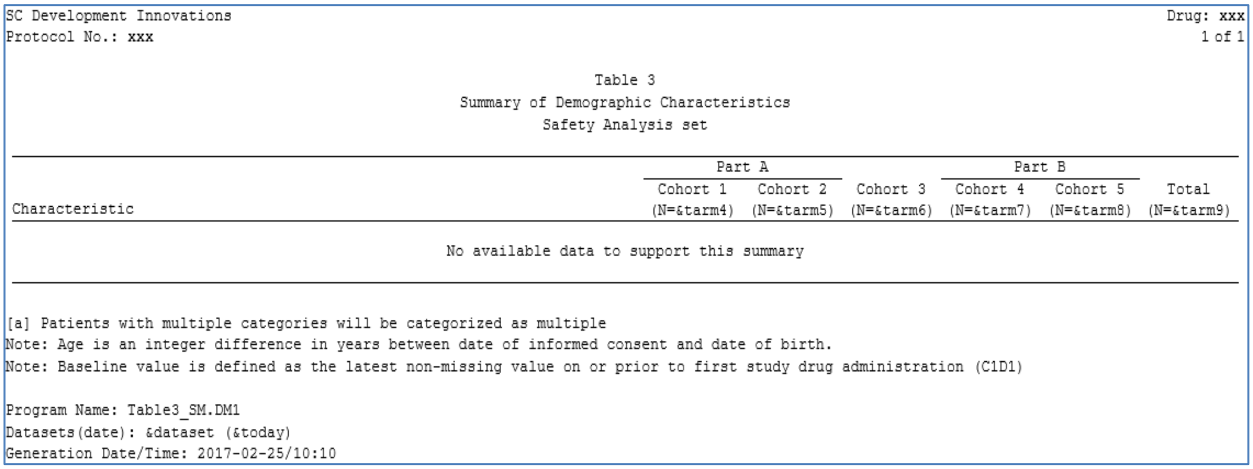
USING/ VALIDATING THE METADATA

The resulting metadata can be easily validated with a dummy dataset whose column names are preceded by F and b to match the column and define statements above. The program below can be used to generate a validation dummy dataset.

```
array av(*) $ 2 F1-F20; /*20 char. variables (F1,...,F20), length = 2*/
array av2(*) $ 2 b1-b20;/*20 char. variables (b1,...,b20), length = 2*/
 do a= 1 to dim(av);
   av(a)='';av2(a)='';end;
```

FINAL TFL OUTPUT

The formatted TFL shown below is based on a dummy dataset and the metadata discussed above.

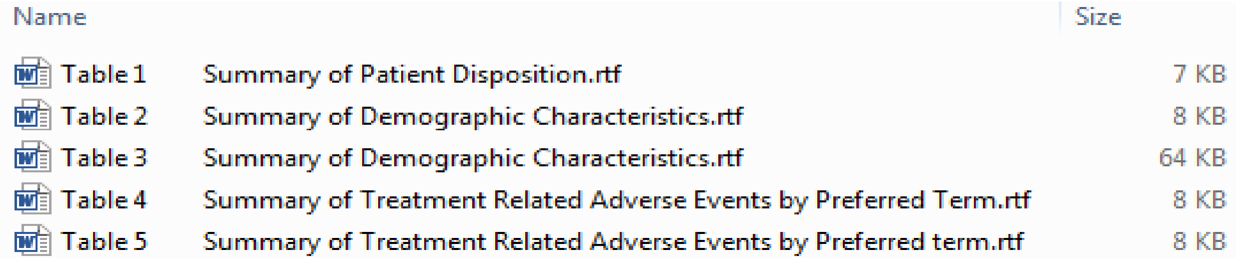


```
SC Development Innovations                                                                                Drug: xxx
Protocol No.: xxx                                                                                            1 of 1

                                                   Table 3
                                     Summary of Demographic Characteristics
                                             Safety Analysis set

 ------------------------------------------------------------------------------------------------------------------
                                                                 Part A                          Part B
                                                          -------------------           -------------------
                                                          Cohort 1   Cohort 2   Cohort 3   Cohort 4   Cohort 5    Total
 Characteristic                                          (N=&tarm4) (N=&tarm5) (N=&tarm6) (N=&tarm7) (N=&tarm8) (N=&tarm9)
 ------------------------------------------------------------------------------------------------------------------

                                      No available data to support this summary

 ------------------------------------------------------------------------------------------------------------------

[a] Patients with multiple categories will be categorized as multiple
Note: Age is an integer difference in years between date of informed consent and date of birth.
Note: Baseline value is defined as the latest non-missing value on or prior to first study drug administration (C1D1)

Program Name: Table3_SM.DM1
Datasets(date): &dataset (&today)
Generation Date/Time: 2017-02-25/10:10
```

Pre-formatted TFLs give the programming team plenty of time to review and provide comments on the mock shells, which in turn will give the statisticians ample time to make the necessary changes.

OUTPUT LABELS

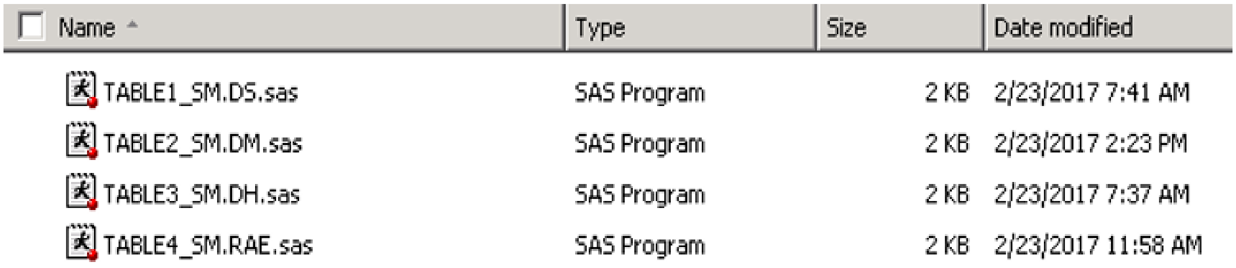
TFLs labeled with distinct numeric and descriptive text are easy to identify. Reviewing TFLs with meaningless labels can be frustrating and time consuming to the statistician, medical writers, and other Data Science study team members.



Name	Size
Table 1 Summary of Patient Disposition.rtf	7 KB
Table 2 Summary of Demographic Characteristics.rtf	8 KB
Table 3 Summary of Demographic Characteristics.rtf	64 KB
Table 4 Summary of Treatment Related Adverse Events by Preferred Term.rtf	8 KB
Table 5 Summary of Treatment Related Adverse Events by Preferred term.rtf	8 KB

PROGRAM NAMES

Well labeled programs are easy to identify and maintain.



□ Name ▲	Type	Size	Date modified
TABLE1_SM.DS.sas	SAS Program	2 KB	2/23/2017 7:41 AM
TABLE2_SM.DM.sas	SAS Program	2 KB	2/23/2017 2:23 PM
TABLE3_SM.DH.sas	SAS Program	2 KB	2/23/2017 7:37 AM
TABLE4_SM.RAE.sas	SAS Program	2 KB	2/23/2017 11:58 AM

CONCLUSION

Once the TFL mock shells have been approved, a study programming team member can easily execute this macro to generate a metadata of titles, footnotes, formatting and reporting details. Implementing a standard transfer of TFL mock shell’s metadata to the final research summaries will eliminate major quality issues, which in turn will free up more time for statistical programmers and biostatisticians to focus on data analysis.

ACKNOWLEDGMENTS

Thanks to the following colleagues for their contribution to the success of this paper.

David Moorman, Principal Biostatistician/ IIT Manager
Julia Shi, Biostatistician Fellow
Tamara Tester, Biostatistician Fellow

CONTACT INFORMATION

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

Julius Kirui, MS
Biostatistician / SAS Programmer

Sarah Cannon Development Innovations (SCDI)
1100 Charlotte Avenue, Suite 800
Nashville, TN 37203

615.329.7274 (Phone), 615.524.4453 (Direct), 615.524.4453 (Fax)
Julius.kirui@scresearch.net, Juliuskr3@gmail.com, Sarahcannon.com

APPENDIX 1

Titles, footnotes, define and headers (TFDH) macro program

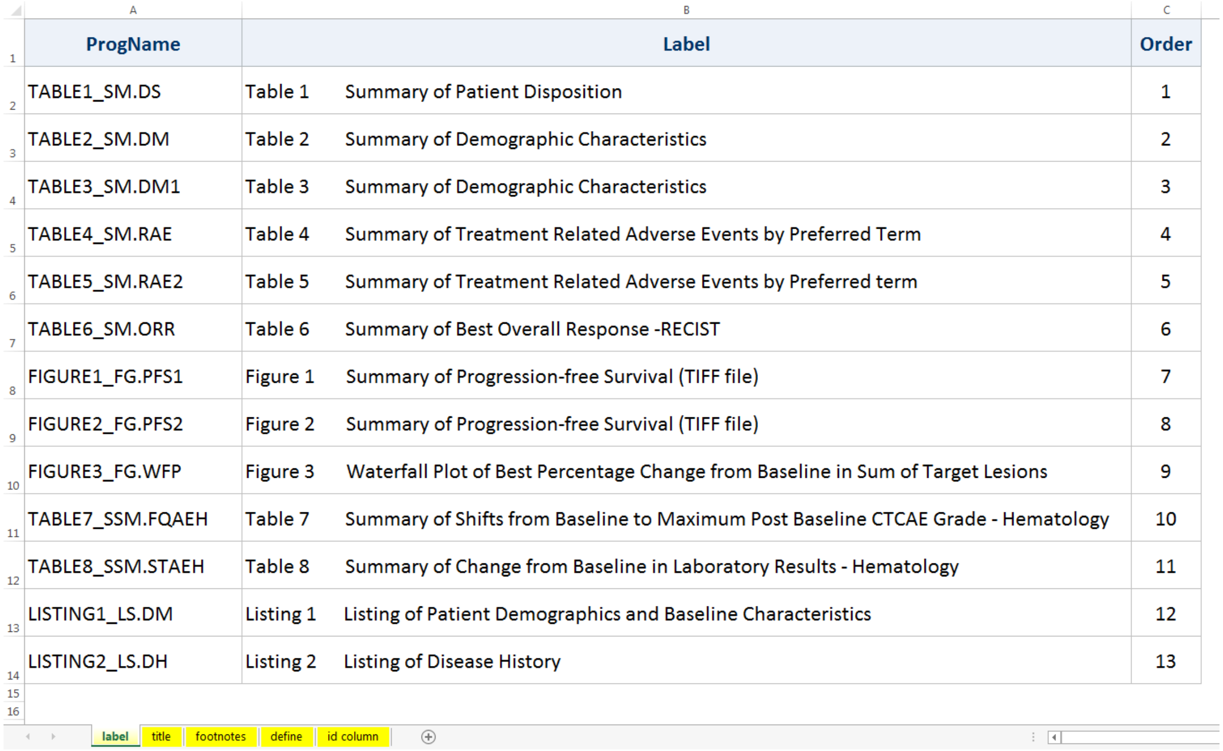
```
%macro tfdh(
type=,subset=,outlabel=,span=,path=,filename=,toc=,tflno=,nshell=,out=);
LIBNAME XLSLIB "&path\&filename"
data toc;
 set XLSLIB."&toc$"n(where=(&type ^=''));
  %if &tflno=NEW %then %do;
    yy=compress(TFL_No_,,'ka')||''||compress(&type);%end;/*new numbering*/
  %else %if &tflno= %then %do;yy=TFL_No_;%end;
   s=length(TFL_No_)+5;
run;
proc sql noprint; /*tot=total No. of mock shells; yy=Table numbers*/
 select count(index),'$'||strip(put(max(s),3.))||'.' into :tot, :tl
  from toc ;
 select index,put(yy,&tl.)||strip(TFL_Title),put(yy,&tl.)
  into :tfl1-:tfl%trim(&tot),:title1-:title%trim(&tot),:tabno1-:tabno1%trim(&tot)
 from toc;
quit;
%if &nshell eq 0 %then %let i=1;
%else %if &nshell gt 0 %then %do; %let i=&nshell;%let tot=&nshell;%end;
%do %while(&i<=&tot);
%if &i gt 0 %then %do;
 proc contents data=XLSLIB."&&tfl&i$"n out=list(keep=name length) noprint; run;
 proc sql noprint; /*grap the last useful column and assign it to macro var f*/
  select max(input(compress(name,'F'),best.)) into :f
  from list where upcase(name) like '%F%' and length gt 1;
quit;
data _st(rename=(index=F1)) ;
 length index title $260;
  set XLSLIB."&&tfl&i$"n;
   index=strip(tranwrd(index,'"',"'")); /*replace " quotation with ' quotation marks*/
    if upcase(F%trim(&f))='PAGE X OF Y' then F%trim(&f)='`{pageof}';
    if _n_=5 then do;if _n_=5 then index="&&tabno&i";
             call symput('tflid2',strip(index));
             call symput('sid',strip(compress(index)));end;
    if _n_=6 then call symput('tfllabel',strip(index));
       <<Titles>>
       if _n_ in(9,10,11) then id=_n_;
run;
%let efot=0;
data _st;
 set _st;
<<assigning footnotes>>
run;
data title(keep=label title);
 length label $ 260 title $260;
  set _st;
   label="&&title&i" ;
     if title ^='';
run;
data _st1;
  informat F1-F%trim(&f) $260.; format F1-F%trim(&f) $260.;length F1-F%trim(&f) $260;
   set _st(where=(id^=.));
    array av(*) F1-F%trim(&f);
     do a=1 to dim(av);
      if id=9 then do; av(a)=tranwrd(av(a),'C','center / ');
                       av(a)=tranwrd(av(a),'L','left / ');end;
      if id=10 then do;if av(a) ^='' then av(a)=tranwrd(av(a),'$A',' $$ ');end;
      if id=11 then do;av(a)=tranwrd(av(a),'(N=','~(N=    ');
                       av(a)=tranwrd(av(a),'(n=','~(n=   ');
      if scan(av(a),2,'=') ^='' and length(compress(scan(av(a),-1,'='))) ge 2
                                then av(a)=strip(scan(av(a),1,'='))||'= XX)';end;end;
run;
proc transpose data=_st1 prefix=col out=_st1;
 id id;
 var F1-F%trim(&f);
run;
%let spin1=0;%let lastspin=0;%let lastshift=0;
data _st2;
 set _st1 %if %length(&subset) gt 0 %then %do;
              %if %scan(&&TFL&i,1,'.')=SM %then %do;(where=(_name_ &subset)); %end;
              %else %if %scan(&&TFL&i,1,'.') ^=SM %then %do; ;%end;%end;
           %else %if %length(&subset) le 0 %then %do; %end;;
  retain _ ;
  id=_n_;
  if col11 ='' then _=0;   _+1;
  if col11 ^='' then _=.;
  if _ ^=. then blank+1;
  if col11 ='' then do col11='b'||strip(put(blank,3.)); col10='b';end;
  if scan(upcase(col11),-1,'~')='(N= XX)' or (scan(upcase(col11),1,'')='GRADE'
                  and scan(col11,2,'') in('0','1','2','3','4','5')) then id_=_n_;
   call symput('nspin', blank);
   if col10 ^in('','b') then call symput('lastspin', _n_);
run;
proc sql noprint;
 select min(id) into :firstarm from _st2 where id_ ^=.;
 select id,compress(scan(col10,2,'$$'),,'kd')  into :spin1-:spin10, :shift1-:shift10
  from _st2 where col10 ^='';
 select sum(input(scan(col9,3,'/'),best.)) into :df1 from _st2 ;
quit;
data _st3;
 set _st2;
  length define $ 500;
  %if %scan(&&TFL&i,1,'.')=SM %then %do;if id ge &firstarm then ids=id+2;
                                        else ids=id;
     order=strip(put((ids-blank),3.));
  if _name_='F1' then do; col9=substr(col9,1,length(col9)-3)||
                             strip(put(%sysevalf(8.7-&df1)+scan(col9,3,'/'),best.));
                         if col11 ='' then col11='';end;
    if scan(upcase(col11),2,'=')=' XX)' then col11=scan(col11,1,'=')||
                      strip(put('=&tarm',6.))||strip(put(order,3.))||')';
    %if %length(&span) gt 0 %then %do;
     if _n_ = &lastspin then col10=scan(col10,1,'$$')||'$$'||
                                    strip(put("&span",3.));%end;
    %else %if %length(&span) le 0 %then %do; ; %end;%end;
 %else %if %scan(&&TFL&i,1,'.')^=SM %then %do;
                                       ids=id;order=strip(put((id-blank),3.));; %end;

 if _ =. then define='define F'||strip(order)||'/display "`S={just= '||
          strip(scan(col9,1,'/'))||'}'||strip(col11)||'"style(column)=[just='||
            strip(scan(col9,2,'/'))||' cellwidth='||strip(scan(col9,3,'/'))||' in];';
 else if _ ^=. then define='define '||strip(col11)||'/display""style(column)=[just='||
          strip(scan(col9,2,'/'))||' cellwidth='||strip(scan(col9,3,'/'))||' in] ;';
  if _ ^=. then col= col11;
  else col='F'||strip(order);
     call symput('order',compress(put(order,3.)));
if col10 ^in('','b') then call symput('lastshift', compress(scan(col10,2,'$'),,'kd'));
run;
```

```
data _st4;
 length label $ 260 column $ 500;
  set _st3;
   by blank;
    if col10 ^='' then do;sp=input(scan(col10,2,'$$'),best.);
                           sptext=strip(scan(col10,1,'$$'));end;
   %if %scan(%upcase(&&tfl&i),1,'.') =SM %then %do;
      %if &spin1 ge 1 %then %do;
      if _n_ lt &spin1 then column=
                            '("`S={borderbottomcolor=white borderbottomwidth=.5" F1'
       %if %eval(&spin1-1) gt 1 %then %do;||'-F'||strip(put((&spin1-1),2.))||')';%end;
       %else %if %eval(&spin1-1) le 1 %then %do;||')';%end;
      else if _n_ ge &spin1 and col10 ^in('','b') then do;
        if length(scan(col10,1,'$$')) gt 1 then column='("'||scan(col10,1,'$$')||
             '`S={} `S={borderbottomcolor=black   borderbottomwidth=.5" F'||
               strip(put(order,2.))||'-F'||strip(put((order-1+sp),3.))||')';
        else  if length(scan(col10,1,'$$')) le 1 then column='("'||scan(col10,1,'$$')
           ||'`S={} `S={borderbottomcolor=white" F'||strip(put(order,2.))||'-F'||
            strip(put((order-1+sp),3.))||')';end;
      if col10='b' and column ='' then column='("`S={}`S={borderbottomcolor=white "'||
                                                strip(col11)||')';%end;
    %else %if &spin1 lt 1 %then %do;column=col;%end;
      if _n_ gt %eval(&lastshift+&lastspin-1) and column in('') then column=
            '("`S={} `S={borderbottomcolor=white " F'||strip(put(order,3.))||'-F'||
                  strip(put("&order",3.))||')';
      else if _n_ gt %eval(&lastshift+&lastspin) and column in('') then column=
            '("`S={} `S={borderbottomcolor=white " F'||strip(put(order,3.))||'-F'||
                   strip(put("&order",3.))||')';%end;
 %else %if %scan(%upcase(&&tfl&i),1,'.') ^=SM %then %do;<Repeate as shown above>%end;
                                     label="&&title&i";
run;
proc sort data=_st4 out=column(keep=label id column)nodupkey;
        by label column;where column ^='';run;
proc sort data=_st4 out=define(keep=label define); by label id;run;
%if &efot=0 %then %let efot=%eval(&ffot+10);
data ft;
 set  _st;
  if _n_=&ffot and length(F1) gt 2 then do F1= tranwrd(F1,'"',"'");
                                             F1=substr(F1,3,%eval(&flen-2));end;
  else if _n_=&ffot and  length(F1) le 2 then F1='';
    if _n_ ge &ffot and _n_ le &efot;
run;
data ft(keep=footnotes label);
  length footnotes $ 500 label $ 260;
   set ft;
    call symput("nf",(_n_));
       label="&&title&i";
        call symput("nf",(_n_));
          F1=tranwrd(F1,'YYYY-MM-DD','&CUTOFFDT');
 if F1 ^='' then footnotes='footnote'||strip(put(_n_,2.))||' j=l "' ||strip(F1)||'";';
run;

data xx;
 length ProgName $50 label $260;
    merge title ft  define column;
     by label;
      order=&i;
      ProgName=upcase("&sid"||"_&&tfl&i");
run;
%if &nshell eq 0 %then %let dmy=1;
%else %if &nshell gt 0 %then %let dmy=&nshell;
    %if &i le &dmy %then %do; data cdft;set _NULL_;run;%end;
```

```
data cdft;    set xx cdft ;run;%end;
%let i=%eval(&i+1); %end;
proc sort data=cdft; by order;run;
ods listing close;
ODS TAGSETS.EXCELXP file = "&path\TFDH.xls" style=htmlblue  options(autofit_height =
'yes');;
<<Ods to Multiple Excel spreadsheets>>
%mend tfdh;
```

APPENDIX 2

Resulting Excel output after macro execution

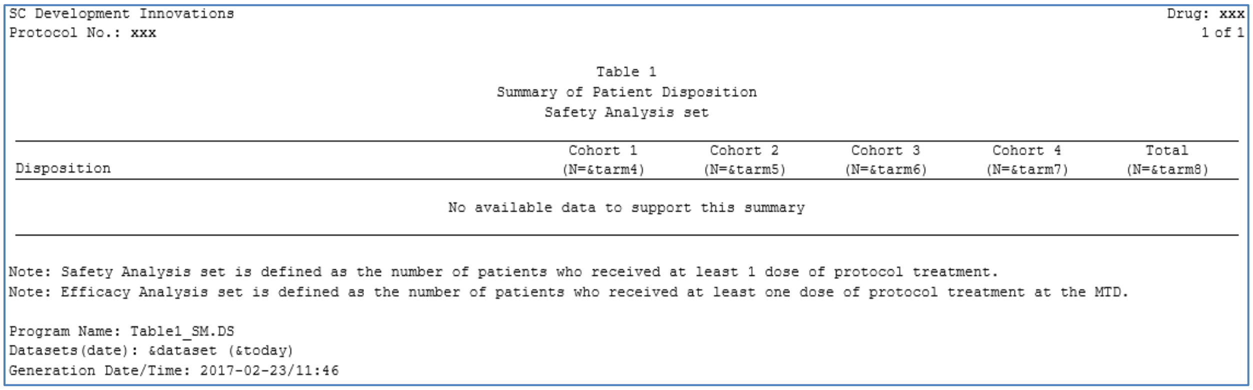


ProgName	Label	Order
TABLE1_SM.DS	Table 1 Summary of Patient Disposition	1
TABLE2_SM.DM	Table 2 Summary of Demographic Characteristics	2
TABLE3_SM.DM1	Table 3 Summary of Demographic Characteristics	3
TABLE4_SM.RAE	Table 4 Summary of Treatment Related Adverse Events by Preferred Term	4
TABLE5_SM.RAE2	Table 5 Summary of Treatment Related Adverse Events by Preferred term	5
TABLE6_SM.ORR	Table 6 Summary of Best Overall Response -RECIST	6
FIGURE1_FG.PFS1	Figure 1 Summary of Progression-free Survival (TIFF file)	7
FIGURE2_FG.PFS2	Figure 2 Summary of Progression-free Survival (TIFF file)	8
FIGURE3_FG.WFP	Figure 3 Waterfall Plot of Best Percentage Change from Baseline in Sum of Target Lesions	9
TABLE7_SSM.FQAEH	Table 7 Summary of Shifts from Baseline to Maximum Post Baseline CTCAE Grade - Hematology	10
TABLE8_SSM.STAEH	Table 8 Summary of Change from Baseline in Laboratory Results - Hematology	11
LISTING1_LS.DM	Listing 1 Listing of Patient Demographics and Baseline Characteristics	12
LISTING2_LS.DH	Listing 2 Listing of Disease History	13

APPENDIX 3

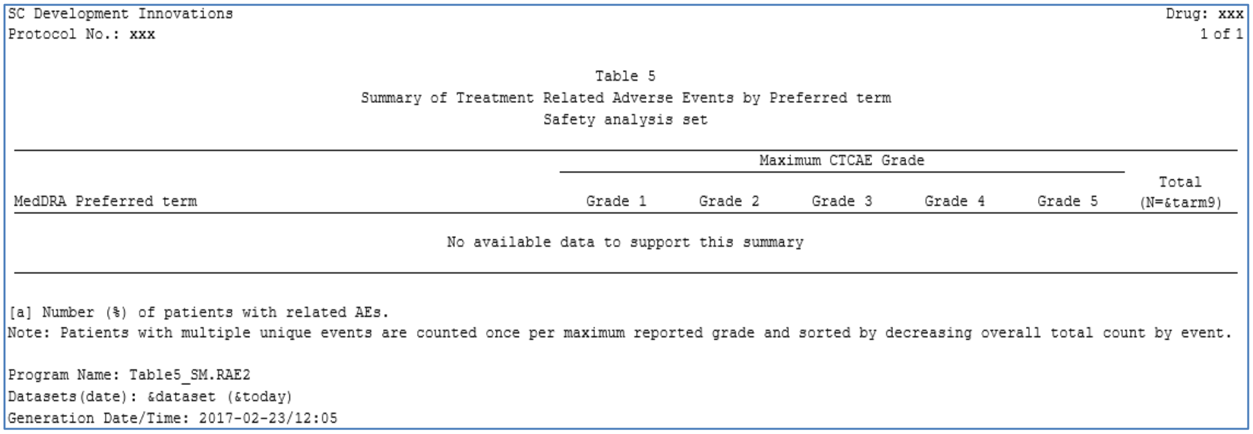
Using/ validating the Metadata

Table 1



```
SC Development Innovations                                                                              Drug: xxx
Protocol No.: xxx                                                                                          1 of 1

                                                      Table 1
                                          Summary of Patient Disposition
                                               Safety Analysis set
 ____________________________________________________________________________________________________________
                                                     Cohort 1        Cohort 2        Cohort 3        Cohort 4        Total
 Disposition                                        (N=&tarm4)      (N=&tarm5)      (N=&tarm6)      (N=&tarm7)      (N=&tarm8)
 ____________________________________________________________________________________________________________

                                          No available data to support this summary
 ____________________________________________________________________________________________________________

Note: Safety Analysis set is defined as the number of patients who received at least 1 dose of protocol treatment.
Note: Efficacy Analysis set is defined as the number of patients who received at least one dose of protocol treatment at the MTD.

Program Name: Table1_SM.DS
Datasets(date): &dataset (&today)
Generation Date/Time: 2017-02-23/11:46
```

Table 5



```
SC Development Innovations                                                                                          Drug: xxx
Protocol No.: xxx                                                                                                      1 of 1

                                                          Table 5
                                     Summary of Treatment Related Adverse Events by Preferred term
                                                    Safety analysis set
 ______________________________________________________________________________________________________________________
                                                                             Maximum CTCAE Grade
                                                          _____________________________________________________
                                                                                                                Total
 MedDRA Preferred term                                      Grade 1    Grade 2    Grade 3    Grade 4    Grade 5   (N=&tarm9)
 ______________________________________________________________________________________________________________________

                                                No available data to support this summary
 ______________________________________________________________________________________________________________________


[a] Number (%) of patients with related AEs.
Note: Patients with multiple unique events are counted once per maximum reported grade and sorted by decreasing overall total count by event.

Program Name: Table5_SM.RAE2
Datasets(date): &dataset (&today)
Generation Date/Time: 2017-02-23/12:05
```