Paper CT-13

Programmatic Automation of Categorizing and Listing Specific Clinical Terms

Ravi Kankipati, Pinnacle Technical Resources, Dallas, TX
Abhilash Chimbirithy, Accenture, Florham Park, NJ

ABSTRACT

Typical clinical trial counts tables sometimes require listing specific terms and their respective groupings in the table footnotes. Manually hard coding this list of terms into a SAS® program is time-consuming and prone to typographical error. Thus, it is important to automate this process. The macro presented in this paper uses an Adverse Events (AE) data set to demonstrate an automated process that will increase efficiency and accuracy. This paper will also provide a brief introduction to the use of SAS automatic macro variable SQLOBS. The example presented may be expanded to include other types of clinical data such as Concomitant Medication (CM) or Medical History (MH).

INTRODUCTION

An AE (Adverse Event) usually refers to a side effect, acute or late effect, complication, toxicity, morbidity, etc. – all essentially pointing to a change possibly caused by treatment. An AE can be reported using multiple terminologies by physicians across research centers, hospitals and countries. These AE terms are then grouped or standardized into preferred terms (SAS variable AEDECOD) and body systems or organ classes (SAS variable AEBODSYS). For illustration purposes a common AE data set format is included in the Example of AE Data Set section.

AE summary or counts tables are very important to physicians and pharmaceutical companies to assess the safety profile of the test drug. These tables usually display the number of adverse events, the number of patients in each treatment group in whom the event occurred, and the incidence of occurrence. Typically, AEs are grouped by AEBODSYS, preferred terms and/or other variables of interest. AE counts tables may sometimes require listing the specific AE terms and their respective groupings in table footnotes. This is a time consuming and tedious task if one manually hard codes the list of AE terms into a SAS program. Hence, it is important and efficient to have this process automated.

We developed a macro to automate the above task that increases the efficiency by reducing time and avoiding typographical errors. The SAS macro program presented in this paper is easy to follow and its use can be expanded to include other type of clinical data such as concomitant medications (CM) or medical history (MH). In the first section we provide an AE data set example and the table footnote requirements. In the second section we provide the macro, ae_auto.sas, with comments and explanation. Additionally, we provide a brief introduction about the SAS ® automatic macro variable, SQLOBS, which is used to find the number of AE groupings in the data set which is used as an upper bound for recursive calling in the macro.

DATA AND TABLE REQUIREMENTS

EXAMPLE OF AE DATA SET

Table 1 presents a common AE data set displaying AE records for 3 patients with subject ID (SUBJID) values equal to 1001, 1002 and 1003. AETERM is SAS variable name for the actual verbatim term recorded by the physician. As mentioned earlier in the paper, AEDECOD is the preferred term and AEBODSYS is the body system or organ class. AEDECOD and AEBODSYS are derived after MedDRA dictionary coding. TRT is the SAS variable name for the treatment groups for the patients in the trial. Below is the data set with variables of interest for this paper.

Table 1: AE Data set Example

SUBJID	AETERM	AEDECOD	AEBODSYS	TRT
1001	Abdominal Pain	Abdominal pain	Gastrointestinal disorders	A
1001	Diarrhoea	Diarrhoea	Gastrointestinal disorders	A
1001	Duodenitis	Duodenitis	Gastrointestinal disorders	A
1001	Arterial hypertension	Hypertension	Vascular disorders	A
1001	vomiting	Vomiting	Gastrointestinal disorders	A
1002	hyperthiroidism	Hyperthyroidism	Endocrine disorders	B
1002	Loose Motions	Diarrhoea	Gastrointestinal disorders	B

1002	hypertension	Hypertension	Vascular disorders	B
1002	Constipation	Constipation	Gastrointestinal disorders	B
1002	toothache	Toothache	Gastrointestinal disorders	B
1003	diahrreic syndrome	Diarrhoea	Gastrointestinal disorders	A
1003	nausea	Nausea	Gastrointestinal disorders	A
1003	uremic gastropathy	uremic gastropathy	Gastrointestinal disorders	A
1003	hypogonadal gonadism	Hypogonadism	Endocrine disorders	A
1003	Dry mouth	Dry mouth	Gastrointestinal disorders	A

TABLE FOOTNOTE REQUIREMENT

The requirement could be a counts table displaying the number of AEs for different AEBODSYS or a summary table displaying number of AEs grouped under AEDECOD and AEBODSYS. The focus of this paper is on generating the footnotes for an ad hoc request. Hence, the derivation of the actual final counts is not addressed. For the footnotes the programmer must display different AETERMs grouped under each AEBODSYS as shown below in Table 2. Footnote1 is a standard footnote. Footnote2 to footnoteN depends on the number of unique AEBODSYS present in AE data set. The example data set has 3 unique AEBODSYS. In this example, when including standard footer plus the 3 unique body systems, the table has 4 footnotes.

Table 2: Table Example

	Treatment A		Treatment B	
	n	(%)	n	(%)
Patients in trial	X	**(x.x)**	X	**(x.x)**
….*AEBODSYS1* ….	X	(x.x)	X	(x.x)
….*AEBODSYS2* ….	X	(x.x)	X	(x.x)
….*AEBODSYS3* ….	X	(x.x)	X	(x.x)

Every patient is counted a single time for each applicable row and column.
***YYYY1, YYYY2 ….*are grouped into *AEBODSYS1*.**
***ZZZZ1, ZZZZ2……*are grouped into *AEBODSYS2*.**
……Display all AEBODSYS and the corresponding AETERMs. Do not display if there is only one AETERM existing for AEBODSYS and they are identical……

MACRO AE_AUTO.SAS

The approach taken here is to create global macro variables footnote1, footnote2 … footnoteN for each AEBODSYS and use these as footnotes once the table counts are generated.

Step1: Obtain distinct AETERMs from the data set for each AEBODSYS and create the footer

For AEBODSYS="Endocrine disorders" we can obtain AETERMs into which it's grouped as follows

```
proc sql;
      select distinct propcase(strip(aeterm)) into: maeterms
      separated by ", "
      from ae_sample
      where aebodsys = "Endocrine disorders";
quit;
```

by using `%put maeterms=&maeterms.` we see in the log:

```
maeterms=Hyperthiroidism, Hypogonadal Gonadism
```

Now to generate the footnotes we can do

```
%let footer = %str(&maeterms. are grouped into Endocrine disorders);
```

But, based on the table layout which has a standard footer this should be footer2. This can be done by creating a suffix and initializing it to 2. We increment the suffix after the footer1 to create the next one. Furthermore we need to list the footer only if AEBODSYS and AETERM are not identical.

```
%if %upcase("&maeterms.") ne %upcase("Endocrine disorders") %then %do;
%let footer&suffix. = %str(&maeterms. are grouped into Endocrine disorders.);
  %let footnum=&suffix.;
  %let suffix=%eval(&suffix.+1);
%end;
```

The macro variable "footnum" will give us the number of footers in the table including the standard footer.

Step2: Find the different AEBODSYS in the sample data set and pass it into a macro variable.

```
proc sql;
       select distinct strip(aebodsys) into:grps
           separated by "$"
         from ae_sample;
quit;
```

The program uses PROC SQL to identify the distinct AEBODSYS and saves the same in the macro variable named "grps" separated by the special character sysmbol "$".

By using `%put grps=&grps.` we see in the log:

```
Grps=Endocrine disorders$Gastrointestinal disorders$Vascular disorders
```

SQL provides the INTO clause in the SELECT statement for creating SAS macro variables. PROC SQL produces three automatic macro variables. SQLOBS contains the number of rows or observations produced by a SELECT statement. SQLRC contains the return code from an SQL statement. And SQLOOPS contains the number of iterations that the inner loop of PROC SQL processes.

In this paper we make use of the automatic macro variable, SQLOBS, created after executions of step2. Here are few more details about SQLOBS from SAS help. When the NOPRINT option is specified, the value of the SQLOBS macro variable depends on whether an output table, single macro variable, macro variable range, or macro variable list is created:

If no output table, macro variable list, or macro variable range is created, then SQLOBS contains the value 1.

If an output table is created, then SQLOBS contains the number of rows in the output table.

If a single macro variable is created, then SQLOBS contains the value 1.

If a macro variable list or macro variable range is created, then SQLOBS contains the number of rows that are processed to create the macro variable list or range.

The macro variable SQLOBS created from the PROC SQL step in our program contains the number of distinct AEBODSYS produced by the SELECT statement.

Using `%put sqlobs=&sqlobs.` we see in the log

```
sqlobs=3
```

The value from the macro variable SQLOBS is used in the third part of our program. It will indicate the number of times the second SQL statement needs to be executed.

Step 3:

The only missing link is to make Step 1 a macro program (%mfoot) and then you can call the macro for each distinct AEBODSYS as shown below.

```
%mfoot(grpname=%str(Endocrine disorders));
%mfoot(grpname=%str(Gastrointestinal disorders));
%mfoot(grpname=%str(Vascular disorders));
```

We can improve the above code using the macro variable "grps" and automatic macro variable "sqlobs" created in step 2 as the upper bound for the do loop. This will increase efficiency and automatically call the macro %mfoot for all distinct AEBODSYS.

```
%do i=1 %to &sqlobs.;
      %let grpval=%scan(&grps, &i, '$');
      %mfoot(grpname=%str(&grpval.) );
%end;
```

The standard footer is initialized as shown below:

```
%let footer1= Every patient is counted a single time for each applicable row and column;
```

Now we can display all the remaining footnotes for distinct AEBODSYS with the help of "footnum" macro variable as shown below.

```
data _null_;
  %do i=1 %to &footnum.;
    %put "Footer&i. = &&footer&i.";
 %end;
run;
```

Imagine a scenario where one of the AEBODSYS has only one AETERM and they are identical. Creating macro variable footnum in the first step should take care of this and we get footnum = 3.

After running the main macro program `%ae_auto(sffx=2)` we obtain all the required footnotes as global macro variables footnote1 – footnote4. In the log file we see 4 footnotes have been created as shown below.

```
"Footer1 = Every patient is counted a single time for each applicable row and
column"
"Footer2 = Hyperthiroidism, Hypogonadal Gonadism are grouped into Endocrine
disorders"
"Footer3 = Abdominal Pain, Constipation, Diahrreic Syndrome, Diarrhoea, Dry
Mouth, Duodenitis,Loose Motions, Nausea, Toothache, Uremic Gastropathy,
Vomiting are grouped into Gastrointestinal disorders"
"Footer4 = Arterial Hypertension, Hypertension are grouped into Vascular
disorders"
```

Footer1 to Footer4 are all macro variables that can be easily used in the table program to generate the table with footnotes as required in the table example (Table2). We have provided below the macro ae_auto and the sample data set.

FULL CODE

```
/*
Input Sample Data set
*/
data ae_sample;
  input subjid aeterm  & $23.  aedecod & $19.  aebodsys & $26.  trt & $1. ;
  datalines;
1001 Abdominal Pain          Abdominal pain      Gastrointestinal disorders  A
1001 Diarrhoea               Diarrhoea           Gastrointestinal disorders  A
```

```
1001 Duodenitis              Duodenitis           Gastrointestinal disorders  A
1001 Arterial hypertension   Hypertension         Vascular disorders          A
1001 vomiting                Vomiting             Gastrointestinal disorders  A
1002 hyperthiroidism         Hyperthyroidism      Endocrine disorders         B
1002 Loose Motions           Diarrhoea            Gastrointestinal disorders  B
1002 hypertension            Hypertension         Vascular disorders          B
1002 Constipation            Constipation         Gastrointestinal disorders  B
1002 toothache               Toothache            Gastrointestinal disorders  B
1003 diahrreic syndrome      Diarrhoea            Gastrointestinal disorders  A
1003 nausea                  Nausea               Gastrointestinal disorders  A
1003 uremic gastropathy      Uremic Gastropathy   Gastrointestinal disorders  A
1003 hypogonadal gonadism    Hypogonadism         Endocrine disorders         A
1003 Dry mouth               Dry mouth            Gastrointestinal disorders  A
   ;
run;


%macro ae_auto(sffx=);

/*
Step1: Derive distinct aeterms from the data set for each AEBODSYS and assign
counts for the footnotes
*/
%macro mfoot(grpname= );
%global footer&sffx. footnum;

proc sql;
      select distinct propcase(strip(aeterm)) into: maeterm
      separated by ", "
      from ae_sample
      where aebodsys = "&grpname.";
quit;

   %if %upcase("&maeterm.") ne %upcase("&grpname.") %then %do;
      %let footer&sffx. = %str(&maeterm. are grouped into &grpname.);
      %put &&footer&sffx.;
      %let footnum=&sffx.;
      %let sffx=%eval(&sffx.+1);
  %end;

%mend mfoot;

/*
Step2: Identify distinct AEBODSYS count using automatic SAS macro variable
SQLOBS.
*/

proc sql;
      select distinct strip(aebodsys) into:grps
        separated by "$"
      from ae_sample;
quit;

%put sqlobs=&sqlobs.;
```

```
/*
Step3: Create footnotes for each distinct AEBODSYS
*/
      %do i=1 %to &sqlobs.;
            %let grpval=%scan(&grps, &i, '$');
            %mfoot(grpname=%str(&grpval.) );
      %end;

/* declare standard footer */
      %global footer1;

      %let footer1=%str(Every patient is counted a single time for each
                        applicable row and column);

      %do i=1 %to &footnum.;
            %put "Footer&i. = &&footer&i.";
      %end;

%mend ae_auto;

/* Call program for the macro */

%ae_auto(sffx=2);
```

CONCLUSION

From the above example we see that it is efficient when a programmer can avoid manual hard coding and automate the SAS program to list different AETERMs and respective AEBODSYS. By using the above macro we can definitely avoid any typographical errors and also ensure that no AETERMs/AEBODSYS will be skipped in the final table. This macro can be perceived as a tool to display a horizontal link between any two variables in a data set by replacing AETERM and AEBODSYS with the variables of interest.

REFERENCES

Common Terminology Criteria for Adverse Events - Instructions and Guidelines
https://webapps.ctep.nci.nih.gov/webobjs/ctc/webhelp/Adverse_Event.htm

SAS Version 9.1 Help System

ACKNOWLEDGMENTS

The authors would like to thank their management team for their advice on this paper/presentation.

CONTACT INFORMATION

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

Ravi Kankipati
Pinnacle Technical Resources
Phone No: 732-360-6079
E-mail: ravi.kankipati@gmail.com

Abhilash Chimbirithy
Accenture
Phone No: 732-594-6962
E-mail: a.v.chimbirithy@accenture.com