

When Asked for Subject Incidences, Go Ahead and FREQ OUT

Rod Norman, Inventive Health Clinical, San Diego, CA

ABSTRACT

A frequent task of the clinical programmer is to provide subject incidences or counts of specific values for relevant study variables. For complex ADaM datasets, table specifications may request counts structured from both the differing values of individual variables as well as values of different variables found with the dataset. To perform such counts, programmers frequently rely on restructuring the data so that subject incidences can be more directly calculated. Layers of SQL SELECT statements or macro looping are commonly employed. In this paper, I show that use of the FREQ procedure with ODS OUTPUT can more directly provide a data set that is relevant to obtaining incidences. Drawing on an example of summarizing adverse events from an ADaM like data set, I show how one run of PROC FREQ with a multiple specification in the TABLE statement can generate a data set that is structured for a direct counting of subject incidences of adverse events organized by terms of the Medical Dictionary for Regulatory Affairs (MedDRA) incorporating customized MedDRA queries (CQs), Common Terminology Criteria (CTCAE) and a seriousness classification.

INTRODUCTION - ADAPTING TO THE ADAM DATA STRUCTURE

As the implementation of Analysis Data Model (ADaM) structures progresses, programmers are looking to produce output results using code that is transparent and easily traceable back to data. Simply modifying code, often complex, that was used to produce outputs from older CRT data is not desired nor warranted. A more direct coding approach is needed to provide clarity and rapid review of the analyses. In addition, an ADaM structure should provide traceability for each observation.

The ADaM structure for Adverse Events (ADAE) is not suited to the ADaM basic data structure (BDS) because it does not make use of the parameter (PARAM) or analysis value (AVAL) variables of other ADaM data sets. It is better structured as a Study Data Tabulation Model (SDTM) with variables added to facilitate the production of adverse event incidences at the event and subject levels. The data are well suited to production of summary tables using standardize terms of the Medical dictionary for Regulatory Affairs (MedDRA). More complex summaries, however, such as those using standardized MedDRA queries (SMQs) or customized queries (CQs or adverse events of special interest) may require a modification of this basic data structure.

In this paper, I propose a use of the FREQ procedure in conjunction with its ODS OUTPUT object List to perform a straightforward restructure of a basic ADAE into a structure more suited to the production of high level summary tables for adverse events of special interest. The approach could be applied to produce another ADaM data set or, more efficiently, used to produce tables directly from ADAE. The code is relatively simple and transparent producing analysis ready data in just one Procedure and two small blocks of code.

THE STRUCTURE OF A REQUESTED SUMMARY TABLE

Consider a request for a summary table of adverse events of special interest. Each category of interest is defined by a flag variable in ADAE derived as a customized query (CQ) involving System Organ Class (SOC) and Preferred Terms (PT). The statistician wishes to count the number of events in all the categories and also, separately, in each of the defined CQs. The counts are to be a subject incidence and additional subsets are desired by the categories of maximum Common Terminology Criteria (CTCAE) grade occurring within the subject and whether the adverse event is considered serious. Counts are requested for the medication (Drug) group and for the untreated group (Placebo) as well as overall Totals for all subjects. Below (Table 1) shows a section of the mock table specifications.

Subjects with	CTC Grade or Serious	DrugX	Placebo	Total
Any special query AE	One or more events	30 (91%)	27 (71%)	57 (80%)
	Severe or Greater	16 (48%)	9 (24%)	25 (35%)
	Mild	2 (6%)	6 (16%)	8 (11%)
	Moderate	12 (36%)	12 (32%)	24 (34%)
	Severe	13 (39%)	8 (21%)	21 (30%)

Subjects with	CTC Grade or Serious	DrugX	Placebo	Total
Query 1 (Heart)	Life Threatening	3 (9%)	1 (3%)	4 (6%)
	Serious	8 (24%)	5 (13%)	13 (18%)
	One or more events	19 (58%)	15 (39%)	34 (48%)
	Severe or greater	4 (12%)	3 (8%)	7 (10%)
	Mild	7 (21%)	9 (24%)	16 (23%)
	Moderate	8 (24%)	3 (8%)	11 (15%)
	Severe	4 (12%)	3 (8%)	7 (10%)
	Life Threatening	3 (9%)	2 (5%)	5 (7%)
Query 2 (Spleen)	Serious	6 (18%)	2 (5%)	8 (11%)
	One or more events
	Severe or greater

(cont.)

Table 1. Basic Structure of Desired Output

INITIAL MODIFICATIONS TO ADAE

ORIGINAL STRUCTURE OF ADAE

Below (Table 2) shows a partial plausible structure for an ADAE data set listing the adverse events for two subjects. Shown are the Preferred Terms of each event (PREFERRED TERM), the CTC severity grade (CTCGRD), flag for seriousness (SERIOUS) and four flags for special adverse events of interest (CQ01NM-CQ04NM). Note that these data have elements of both a vertical and horizontal structure; events and scaling in each observation and flags listed across each observation.

SUBJ	TREAT	PREFERRED TERM	CTCGRD	SERIOUS	CQ01NM	CQ02NM	CQ03NM	CQ04NM
6001	Drug	DYSPNOEA	3	NO	no	no	yes	no
		CANDIDIASIS	1	NO	no	yes	no	no
		CARDIAC CONGESTION	2	NO	no	yes	no	no
		CARDIAC FAILURE	2	NO	no	no	yes	no
		HYPOKALAEMIA	2	NO	no	no	no	yes
		DERMATITIS	2	NO	no	yes	no	no
		HYPOMAGNESAEMIA	1	NO	no	no	no	yes
		PHLEBITIS	2	NO	yes	no	no	no
		FATIGUE	3	NO	yes	no	no	no
		OSTEOMYELITIS	3	YES	yes	yes	no	no
6002	None	COUGH	2	NO	yes	no	no	no
		DEHYDRATION	2	NO	no	no	no	yes
		VOMITING	1	NO	no	yes	no	no
		ANAEMIA	2	NO	no	no	yes	no
		ORAL DISORDER	1	NO	yes	no	yes	no
		DIARRHOEA	1	NO	no	no	no	yes

Table 2. Structure of ADAE for 2 Subjects and Select Variables

MODIFICATION TO ADAE TO REDUCE ANALYSIS STEPS

To better take advantage of the properties of PROC FREQ, a simple modification of the data in Table 2 is performed. First, a macro variable is created containing a list of the CQ variables

```
%let sAE1st=CQ01NM CQ02NM CQ03NM CQ04NM;
```

Then, each of these flag variables are recorded as 'Y' when 'yes' but set to missing when 'no'. In addition the flag for seriousness (aesr) is also set with the values used in the CQ flags. Since the counts will include only events

When Asked for Subject Incidences, Go Ahead and FREQ OUT, continued

that are flagged as 'Y' for one or more variables, the data set size can be reduced to include only such observations.

```

data aered (where=(numb>=1));    ***== reduc obs to those needed ===;
  retain subject treat;
set libadam.ADAE;
  array AEsp[*] $ &sAElst  ;
  numb=0;
  do N=1 to dim(AEsp) ;
    if AEsp[N] in ('Yes') then do; numb+1; AEsp[N]='Y'; end;
    else AEsp[N]=' ';
  end;
  ***-- serious flag mod --****;
if aeser='Y' then aeser='Y';
  else aeser=' ';
keep treat subject aept aeser sevcd &sAElst numb ;

```

The resulting data set (Output 1) will have the following structure as shown for the two selected subjects.

SUBJ	TREAT	PREFERRED TERM	CTCGRD	SERIOUS	CQ01NM	CQ02NM	CQ03NM	CQ04NM	numb
6001	Drug	DYSпноEA	3				Y		1
6001	Drug	CANDIDIASIS	1			Y			1
6001	Drug	CARDIAC CONGESTION	2			Y			1
6001	Drug	CARDIAC FAILURE	2				Y		1
6001	Drug	HYPOKALAEMIA	2					Y	1
6001	Drug	DERMATITIS	2			Y			1
6001	Drug	HYPOMAGNESAEMIA	1					Y	1
6001	Drug	PHLEBITIS	2			Y			1
6001	Drug	FATIGUE	3				Y		1
6001	Drug	OSTEOMYELITIS	3	Y	Y	Y			2
6002	None	COUGH	2		Y				1
6002	None	DEHYDRATION	2					Y	1
6002	None	VOMITING	1			Y			1
6002	None	ANAEMIA	2				Y		1
6002	None	ORAL DISORDER	1		Y		Y		2
6002	None	DIARRHOEA	1					Y	1

Output 1. SAS Data Set for Two Subjects and Select Variables

The data in Output 1 has a notable feature that makes the deriving Table 1 more complicated. That feature is the possible co-occurrence of 'Y' flags for the different flag variables as evident in 'OSTEOMYELITIS' of subject 6001 and the 'ORAL DISORDER' observation of subject 6002. To arrange the data so that the values desired in Table 1 can be straightforwardly derived would seem to require more restructuring perhaps using the sequential execution of SQL Procedure with the DISTINCT keyword, or by using multiple OUTPUT statements for different variables in a DATA step. There is, however, a feature of PROC FREQ that can be employed to simplify the derivation of a directly countable data set.

CREATION OF DATA SET FOR SUBJECT INCIDENCES

BASIC USES OF PROC FREQ FOR DERIVING INCIDENT DATA SET

Most programmers are aware of using PROC FREQ to create data sets of unique multiway combinations of variable values along with their counts. This method simply uses the LIST keyword along with an OUT= statement to create a data set of these combinations and their counts and percents. Code for creating such a data set follows:

```

proc freq data=aeincid;
  tables  subject*treat*sevcd*CQ1 / nocum list missing;
  out=freqout;

```

The counts and percents are not of interest here, but the unique combinations of variables in the TABLES statement are. For the two example subjects, the output data is shown in Output 2. The approach requires additional coding when combinations with more than one flag variable are required and would entail running multiple PROC FREQs and combining all the resulting tables.

Subject	Treat	sevcd	CQ01NM
6001	Drug	1	
6001	Drug	1	Y
6001	Drug	2	
6001	Drug	2	Y
6001	Drug	3	
6002	None	1	
6002	None	1	Y
6002	None	2	
6002	None	2	Y

Output 2 The Freqout Data Set for the Variables Specified in the TABLES Statement

Providing more output per unit code is the code below, which uses PROC FREQ to create, in one step, a data set structure for counting adverse events flagged by all the CQ variables.

```
ods listing close;
ods output List=subjfreq (drop=F_: Frequency );
proc freq data=aeincid ;
tables subject*treat*sevcd*( &sAElst ) / list nopercnt nocum;
ods output close;
ods listing;
```

This code capture the ODS OUTPUT object List, which produces a data set that is a multiway expansion of variables listed in the TABLES statement. As specified, the output data set (subjfreq) contains four strata, defined in the table variable, for each subject, each composed as a tabulation of treatment, severity and one of the four CQ variables. There are extra variables in subjfreq that are redundant or not needed, including Frequency, which are dropped. A look at the data in subjfreq for the two examples subjects (Output 3) shows the advantage of using PROC FREQ in this manner.

Table	subject	treat	sevcd	CQ01NM	CQ02NM	CQ03NM	CQ04NM
subject*treat*sevcd*CQ01NM	6001	Drug	3	Y			
subject*treat*sevcd*CQ02NM	6001	Drug	1		Y		
subject*treat*sevcd*CQ02NM	6001	Drug	2		Y		
subject*treat*sevcd*CQ02NM	6001	Drug	3		Y		
subject*treat*sevcd*CQ03NM	6001	Drug	2			Y	
subject*treat*sevcd*CQ03NM	6001	Drug	3			Y	
subject*treat*sevcd*CQ04NM	6001	Drug	1				Y
subject*treat*sevcd*CQ04NM	6001	Drug	2				Y
subject*treat*sevcd*CQ01NM	6002	None	1	Y			
subject*treat*sevcd*CQ01NM	6002	None	2	Y			
subject*treat*sevcd*CQ02NM	6002	None	1		Y		
subject*treat*sevcd*CQ03NM	6002	None	1			Y	
subject*treat*sevcd*CQ03NM	6002	None	2			Y	
subject*treat*sevcd*CQ04NM	6002	None	1				Y
subject*treat*sevcd*CQ04NM	6002	None	2				Y

Output 3. SAS Data Set from ODS List Object of PROC FREQ

This data set has the structure desired for obtaining the counts in mock Table 1. There is an observation for each adverse event of a given severity within each subject as flagged by each CQ. Notice that the double flagged adverse

event 'OSTEOMYELITIS' of sevcd=3 in Output 1 is now a separately flagged output observation of the CQ01NM and CQ02NM columns. Thus the co-occurrence flags have been uncoupled into separate observations simply by the processing of PROC FREQ. Also, note that for the three observations of subject 6001 flagged by CQ2NM='Y' are listed in an increasing order of sevcd by the default action of PROC FREQ. This will permit selection of maximum severity grade for each CQ by taking the last observation with subject and CQ variable combination.

EXTENDED USE OF PROC FREQ FOR OTHER VARIABLES AND OVERALL CQ SUMMARY

The approach used to produce Output 3 can be extended to add observations for obtaining the overall counts of CQs, i.e. the top section of Table 1 ('Any special query AE'). The code below shows how to add these observations to subjfreq along with observations for counting the serious events overall and within each CQ.

```
ods listing close;
ods output List=subjfreq (drop=F_: frequency);
proc freq data=aeincid ;
  tables  subject*sevcd*treat
         subject*aeser*treat
         (&sAE1st)*subject*sevcd*treat
         (&sAE1st)*subject*aeser*treat / list nopercnt nocum;
ods output close;
ods listing;
```

The table statement looks rather complex, however, it is simple to understand. The first table construct requests unique observations for combinations of subject and severity. (Treatment and subject combinations are matched in this study and treat is included in the table statement only so it will appear in the table output without additional code.). The second construct captures subjects with a serious flag and any CQ flag. Similarly, the third and fourth constructs output subjects for each CQ flagged for different severity codes or seriousness, respectively. This new version of the subjfreq data set has the structure show in Output 4 for the same example subjects and sorted for display here.

Table	sevcd		SERIOUS	CQ01NM	CQ02NM	CQ03NM	CQ04NM
	subject	treat					
subject*sevcd*treat	6001	1	Drug				
subject*sevcd*treat	6001	2	Drug				
subject*sevcd*treat	6001	3	Drug				
subject*AESER*treat	6001	.	Drug	Y			
subject*sevcd*treat	6002	1	None				
subject*sevcd*treat	6002	2	None				
CQ01NM*subject*sevcd*treat	6001	3	Drug	Y			
CQ02NM*subject*sevcd*treat	6001	1	Drug		Y		
CQ02NM*subject*sevcd*treat	6001	2	Drug		Y		
CQ02NM*subject*sevcd*treat	6001	3	Drug		Y		
CQ03NM*subject*sevcd*treat	6001	2	Drug			Y	
CQ03NM*subject*sevcd*treat	6001	3	Drug			Y	
CQ04NM*subject*sevcd*treat	6001	1	Drug				Y
CQ04NM*subject*sevcd*treat	6001	2	Drug				Y
CQ01NM*subject*AESER*treat	6001	.	Drug	Y	Y		
CQ02NM*subject*AESER*treat	6001	.	Drug	Y	Y		
CQ01NM*subject*sevcd*treat	6002	1	None	Y			
CQ01NM*subject*sevcd*treat	6002	2	None	Y			
CQ02NM*subject*sevcd*treat	6002	1	None		Y		
CQ03NM*subject*sevcd*treat	6002	1	None			Y	
CQ03NM*subject*sevcd*treat	6002	2	None			Y	
CQ04NM*subject*sevcd*treat	6002	1	None				Y
CQ04NM*subject*sevcd*treat	6002	2	None				Y

Output 4 SAS Data from ODS Object List of PROC FREQ with Extended TABLE Statement

Compared to Output 3, this output contains some additional observations and includes the variable serious. The top six rows show the combinations that can be used to count the subjects (here 6001, 6002) in the 'Any special query AE' part of Table 1. The serious variable will permit the counts of overall serious classified CQs and the serious counts within each CQ.

OBTAINING SUBJECT INCIDENCES FOR THE LIST OBJECT OF PROC FREQ

POST PROC FREQ PROCESSING OF THE SUBJFREQ DATA SET

To prepare the data for counting some modest post processing code is applied. Presented here is suggested code for this processing.

```

data spAEmax;
  length &sAElst $8;
set subjfreq ;
  by subject  notsorted;
  lvl=tranwrd(table, 'AESER', 'sevcd');    *--First ;
  if aeser='Y' then sevcd=6 ;
  rename sevcd=CTC_Ser ;
  array sAE[*] $8 &sAElst ;                *--Second;
  do N=1 to dim(sAE);
    if sAE[N]^='' then sAE[N]=vlabel(sAE[N]);
  end;
  IF last.subject;                          *--Third;
  drop table aeser N frequency;

```

The above code accomplishes the following. *First, it creates a variable lvl from table taking the table values for serious rows and renaming them into the table name used for severity codes. Also, observations of the 'Y' value for aeser are populated with a numeric 6 for the sevcd variable. Then the sevcd variable is renamed to the more descriptive CTC_Ser. These two actions will effectively put the serious observations into the severity column as is requested in Table 1. *Second, the CQ variable values are recreated by using the associated labels. This allows the CQ text descriptions to enter the data set without additional processing.* Third, since the highest severity grade or serious event is counted within each subject, only the last observation of each CQ and subject combination is saved. This is accomplished by the selective IF statement. Although the observations in subjfreq are not sorted only by subject and sevcd, they are arranged within combinations of the CQ variables, serious, subject and ascending sevcd. Thus a NOTSORTED BY statement of last.subject will include these observations. The resulting spAEmax data set looks as in Output 5 for the two example subjects.

lvl	subject	treat	CTC_Ser	CQ01NM	CQ02NM	CQ03NM	CQ04NM
subject*sevcd*treat	6001	Drug	3				
subject*sevcd*treat	6002	None	2				
subject*sevcd*treat	6001	Drug	6				
CQ01NM*subject*sevcd*treat	6001	Drug	3	Heart			
CQ01NM*subject*sevcd*treat	6002	None	2	Heart			
CQ02NM*subject*sevcd*treat	6001	Drug	3		Spleen		
CQ02NM*subject*sevcd*treat	6002	None	1		Spleen		
CQ03NM*subject*sevcd*treat	6001	Drug	3			Kidney	
CQ03NM*subject*sevcd*treat	6002	None	2			Kidney	
CQ04NM*subject*sevcd*treat	6001	Drug	2				Liver
CQ04NM*subject*sevcd*treat	6002	None	2				Liver
CQ01NM*subject*sevcd*treat	6001	Drug	6	Heart			
CQ02NM*subject*sevcd*treat	6001	Drug	6		Spleen		

Output 5 Data Set Spaemax – Modification of Work Data Set Subjfreq

The data is spAEmax is now ready to derive the subject incidences required for Table 1.

SUBJECT INCIDENCES USING MEANS PROCEDURE WITH MULTILABEL FORMATS

To accomplish this task efficiently, PROC MEANS without a VAR statement is employed using associated formats. See Norman, 2012 for further details.

```
%let sAElstwa=%sysfunc(tranwrd(&sAElst,%str( ),%str(*) ));
proc format ;
  value $ trtf (multilabel notsorted) Drug=DrugX None=Placebo Drug, None=Total;
  value allctcf (multilabel notsorted) 1,2,3,4,5=Any 3,4,5=Concerning 1=Mild
    2=Moderate 3=Severe 4=Life Threatening 5=Death 6=Serious;
proc MEANS data=spAEmax noprint;
  format treat $trtf. maxctc allctcf. ;
  class lvl / order=data;
  class &sAElst / missing ;
  class maxctc treat / preloadfmt exclusive mlf order=data ;
  types lvl*treat*ctc_ser*&sAElstwA;
  output out=ctaemax ;
```

In the above code, a new macro variable (&sAElstwa) is created that is simply a list of the CQ variables separate by '**' for use in the TYPES statement.

This construction of PROC MEANS outputs a data set ctaemax containing counts of the multiway combinations of values for variables in the TYPES statement. The order of variables in the CLASS statements dictates the order of values in ctaemax. The keyword phrase ORDER=DATA in combination with MLF and PRELOADFMT preserves the ordering of values in the VALUE statements of PROC FORMAT. The MULTILABEL option of PROC FORMAT permits specification of values to be in multiple categories and thus here creates a total column and multiple counts for CTCAE grades.

Output 5 displays the resulting data for CQ02NM (Spleen). Note that the other CQ variables are blank. For these other CQs the output observations will be populated by counts and labels for each CQ in the appropriate CQ column (data not shown). The name of the CQ variable is also contained within the lvl column. For the 'Any special query AE' section of the table the lvl value is 'subject*sevcd*treat' and all of the CQ variables are blank (data not shown). These data can be simply transposed by the treat variable to obtain the structure required of Table 1.

lvl	CQ01NM	CQ03NM	CQ04NM			
	CQ02NM			CTC_Ser	treat	
					count	
CQ02NM*subject*sevcd*treat	Spleen			Any	DrugX	22
CQ02NM*subject*sevcd*treat	Spleen			Any	Placebo	9
CQ02NM*subject*sevcd*treat	Spleen			Any	Total	31
CQ02NM*subject*sevcd*treat	Spleen			Concerning	DrugX	3
CQ02NM*subject*sevcd*treat	Spleen			Concerning	Placebo	2
CQ02NM*subject*sevcd*treat	Spleen			Concerning	Total	5
CQ02NM*subject*sevcd*treat	Spleen			Mild	DrugX	9
CQ02NM*subject*sevcd*treat	Spleen			Mild	Placebo	6
CQ02NM*subject*sevcd*treat	Spleen			Mild	Total	15
CQ02NM*subject*sevcd*treat	Spleen			Moderate	DrugX	10
CQ02NM*subject*sevcd*treat	Spleen			Moderate	Placebo	1
CQ02NM*subject*sevcd*treat	Spleen			Moderate	Total	11
CQ02NM*subject*sevcd*treat	Spleen			Severe	DrugX	2
CQ02NM*subject*sevcd*treat	Spleen			Severe	Total	2
CQ02NM*subject*sevcd*treat	Spleen			Life Risk	DrugX	1
CQ02NM*subject*sevcd*treat	Spleen			Life Risk	Placebo	2
CQ02NM*subject*sevcd*treat	Spleen			Life Risk	Total	3
CQ02NM*subject*sevcd*treat	Spleen			Serious	DrugX	2
CQ02NM*subject*sevcd*treat	Spleen			Serious	Placebo	2
CQ02NM*subject*sevcd*treat	Spleen			Serious	Total	4

Output 6 Observations for Counts of CQ02NM in Work SAS Data Set Ctsae

CONCLUSION

As the CDISC standards for submission of clinical data advances, the role of SAS programmer evolves. ADaM data sets, created to support specific types of analyses, are design so that reviewers can evaluate clinical studies in an accurate and timely manner. Programs design to produce outputs from such data sets should be straightforward and easily reviewed. In the past, SAS programs submitted in support of clinical studies have frequently been overly complicated, perhaps even long macros with complex logic branching and excessive macro variables. This was natural since many macros were designed to handle many types of data and produce a variety of outputs. In contrast, the purpose of this paper is to show how construction of standard SAS procedures, here FREQ and MEANS, with minimal supplementary code can be employed to simply restructure and analyze ADaM data preserving data traceability and transparency.

REFERENCE

Norman, Rod, 2012, CC30. A Mean Way to Count, Enumerating the Values of Multiple Variables Using Formats with the Means Procedure, PharmaSUG 2012 Conference Proceedings, San Francisco, CA, The Pharmaceutical Industry SAS® Users Group.

CONTACT INFORMATION

Contact the author at:

Name: Rod Norman
Enterprise: InVentiv Health Clinical
City, State ZIP: San Diego, CA 92064
E-mail: rodney.norman@inventivhealth.com
<http://www.inVentivHealthclinical.com>

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.