PharmaSUG 2016 - Paper BB12

Preparing the Data to Ensure Correct Calculation of Relative Risk

Ravi Kankipati, Seattle Genetics, Inc., Bothell, WA

Abhilash Chimbirithy, Merck & Co., Inc., Rahway, NJ

ABSTRACT

Relative Risk (RR) used frequently in statistical analysis can be easily obtained by using SAS® procedure. However, how the data are organized before calling the SAS procedure will affect the outcome and could result in an incorrect RR. From a programmer's perspective when asked to provide RR for a group, or subgroups within that group one should be very careful in having the data organized correctly. In this paper we will walk through one such approach to derive the desired classification variables based on the data, focusing on organizing the data before finally calling SAS procedures to obtain a correct RR calculation.

INTRODUCTION

Statistics is the study of the collection, analysis, interpretation, presentation, and organization of data. Statistical analysis helps to explore and present large amounts of data to discover underlying patterns and trends. SAS BASE, SAS/STAT supports several statistical procedures and several analyses can be easily generated. But to make sure that the outputs from these SAS procedures are correct depends on several factors such as input data, SAS programming codes etc. Incorrect SAS programming codes can be easily identified as SAS software would not execute correctly and will generate error or warning messages. But it is not very easy to identify if there is any issue with input dataset unless the output results are carefully reviewed or validated. There could be nothing incorrect with an input data itself but how the data are organized before calling the SAS procedure will affect the outcome and could result in an incorrect analysis. In this paper we will present such an example for RR. The example explained in the paper requires RR generated for a group.

The RR is the ratio of event probabilities at two levels of a variable where the "event" is the response level of interest. RR is used frequently in the statistical analysis of binary outcomes where the outcome of interest has relatively low probability and hence it is very popular in clinical trials. RR can be generated using the Cochran-Mantel-Haenszel (CMH) option in PROC FREQ. This paper will focus on generating the correct p-value, estimate and 95% confidence intervals for a group and also showing the difference in values when data is not arranged correctly. We will not go into details of interpreting statistical analysis results. The main focus will remain to show how data needs to be arranged to obtain the correct results and have to use efficient SAS programming steps to achieve the same.

In the process of arranging the data in the correct format we have highlighted uses of different SAS procedures and functions like PROC TABULATE, ARRAY and PROC FREQ. The SAS program presented in this paper is easy to follow and its use can be expanded to other types of analysis. In the first section we provide the table requirements. In the second section we provide the program, RELRISK.sas, with comments and explanation for each step. We will clearly show the results from CMH method when the data is not sorted and the difference when it is sorted correctly. Also, we have used SAS array which is one of the most convenient way of temporarily identifying a group of variables for processing within a data step which increases programming efficiency.

TABLE REQUIREMENT

The requirement is to generate the RR for endpoints defined based on a selected criteria for overall group and the categories that lie within it i.e. for the category group we have male, female. The table is displayed for TRT1 and TRT2 with RR calculated against TRT2 for each category.

Table Example

	TRT1		TRT2		Relative Risk vs. TRT2	
	n	(%)	n	(%)	Estimate (95% CI)	p-Value
Group1 <Total>						
Flag01	x	(x.x)	x	(x.x)	x.xx (x.xx, x.xx)	x.xxx
Flag02	x	(x.x)	x	(x.x)	x.xx (x.xx, x.xx)	x.xxx
Flag03	x	(x.x)	x	(x.x)	x.xx (x.xx, x.xx)	x.xxx
Flag04	x	(x.x)	x	(x.x)	x.xx (x.xx, x.xx)	x.xxx
Group2 <Male>						
Flag01	x	(x.x)	x	(x.x)	x.xx (x.xx, x.xx)	x.xxx
Flag02	x	(x.x)	x	(x.x)	x.xx (x.xx, x.xx)	x.xxx
Flag03	x	(x.x)	x	(x.x)	x.xx (x.xx, x.xx)	x.xxx
Flag04	x	(x.x)	x	(x.x)	x.xx (x.xx, x.xx)	x.xxx
Group3 <Female>						
Flag01	x	(x.x)	x	(x.x)	x.xx (x.xx, x.xx)	x.xxx
Flag02	x	(x.x)	x	(x.x)	x.xx (x.xx, x.xx)	x.xxx
Flag03	x	(x.x)	x	(x.x)	x.xx (x.xx, x.xx)	x.xxx
Flag04	x	(x.x)	x	(x.x)	x.xx (x.xx, x.xx)	x.xxx

METHOD1: OBTAINING RELATIVE RISK <DATA NOT SORTED – INCORRECT RR>

Sample data mentioned below is an intermediate dataset obtained after processing from ADLB (ADaM dataset for laboratory test records in Basic Data Structure) and not in the desired sort order, just before passing through PROC FREQ to obtain RR. How this dataset can be obtained from an Analysis dataset is addressed in later part of the paper 'Preparation of Data'. This is only a sample of the entire dataset as variable ENDPOINT contains all flags FLG01-FLG04.

	endpoint	sex	trtan	event	count
1	flg01	F	1	No	28
2	flg01	F	2	No	62
3	flg01	M	1	No	24
4	flg01	M	2	No	58
5	flg01	F	1	Yes	2
6	flg01	F	2	Yes	7
7	flg01	M	1	Yes	6
8	flg01	M	2	Yes	10
9	flg02	F	1	No	29
10	flg02	F	2	No	64
11	flg02	M	1	No	28
12	flg02	M	2	No	59
13	flg02	F	1	Yes	1
14	flg02	F	2	Yes	5

code:

```
proc freq data=test order=data;
        weight count /zeros;
        tables sex*trtan*event /cmh;
        by endpoint;
        output out=res_ cmh bdchi;
  run;
```

Output:

Above procedure provides an output dataset 'RES_' which contains variables for the appropriate statistics. For obtaining RR with Confidence Intervals, we can use the generated variables _LGRRC1_, L_LGRRC1, U_LGRRC1 and after passing through the macro %relrisk mentioned in Appendix 2 we obtained the following output. Also, the variable P_CMHGA contains p-value.

Obs	Name	cnt1	cnt2	RR	pval
1	Patients in population	60	137		
2	with one or more events of flg01	8	17	1.01 (0.91, 1.12)	0.862
3	with one or more events of flg02	3	14	1.05 (0.98, 1.14)	0.229
4	with one or more events of flg03	4	8	0.99 (0.92, 1.05)	0.828
5	with one or more events of flg04	2	3	0.98 (0.93, 1.04)	0.635

METHOD 2: OBTAINING RELATIVE RISK <DATA SORTED – CORRECT RR>

Sort the dataset as mentioned below:

```
proc sort  data = tabrr2_
           out = tabrr3_;
   by endpoint sex trtan descending event;
 run;
```

Above we used the descending option to have 'Yes' appear before 'No' consistently for each endpoint, sex, and trtan combination. The rationale behind that is to have the event value of interest appear first and here it is event = 'Yes'. Sample output of the dataset in the desired order.

	endpoint	sex	trtan	event	count
1	flg01	F	1	Yes	2
2	flg01	F	1	No	28
3	flg01	F	2	Yes	7
4	flg01	F	2	No	62
5	flg01	M	1	Yes	6
6	flg01	M	1	No	24
7	flg01	M	2	Yes	10
8	flg01	M	2	No	58
9	flg02	F	1	Yes	1
10	flg02	F	1	No	29
11	flg02	F	2	Yes	5
12	flg02	F	2	No	64
13	flg02	M	1	Yes	2
14	flg02	M	1	No	28
15	flg02	M	2	Yes	9
16	flg02	M	2	No	59

By passing the above sorted dataset into PROC FREQ code mentioned in method 1 and after formatting the data for display purpose below is the output:

Obs	Name	cnt1	cnt2	RR	pval
1	Patients in population	60	137		
2	with one or more events of flg01	8	17	1.12 (0.51, 2.45)	0.862
3	with one or more events of flg02	3	14	0.49 (0.15, 1.63)	0.229
4	with one or more events of flg03	4	8	1.13 (0.36, 3.62)	0.828
5	with one or more events of flg04	2	3	1.56 (0.25, 9.65)	0.635

COMPARISON OF THE OUTPUTS: WHEN INPUT DATA IS NOT SORTED VS SORTED

From the above outputs it is very clear that data sorted and not sorted generates different results. The CNT1 and CNT2 value remains the same but values in RR column has changed with difference in estimate values. Interestingly p-values are same. There are no warning or error messages in the log files when generating both outputs and the desired output is obtained when data is sorted correctly.

STEP1: PREPARATION OF DATA

Having mentioned the main idea behind the paper above, we will now explain how a programmer can get the input analysis dataset to a format required by PROC FREQ procedure to obtain RR results.

Below are the specifications defining events of interest:

For ADLB we would like to calculate RR for subjects having a lab parameter value equal to some number (for e.g. AVAL = 10 then FLG01 = 'Yes'). Now we are provided with 4 such definitions for which RR table needs to be obtained as mentioned in the table shell:

From ADLB based on the condition we create a new PARAMN which satisfies it (i.e. AVAL = 10 then PARAMN = flg01, AVAL =20 then PARAMN = flg02 etc.). Each subject can satisfy multiple conditions but can be displayed only once per meeting condition in the dataset. In the following sample we have 2 subjects 787, 800 that satisfy multiple criteria. This dataset contains only subjects that have satisfied at least one criterion.

	usubjid	paramn	trtan	aval
1	783	flg03	2	30
2	785	flg01	1	10
3	787	flg01	1	10
4	787	flg04	1	40
5	788	flg01	1	10
6	800	flg01	2	10
7	800	flg03	2	30
8	821	flg01	2	10

Above dataset is processed through Appendix 1 program by merging with ADSL and follows its structure. If a subject doesn't fall under any criteria i.e. not present in above dataset then per specifications all the flag values will be 'No' and appears as shown below with rest of the subjects. Refer to USUBJID=784 below. A simple SAS step can be used to obtain the same as shown in the code.

	usubjid	flg01	flg02	flg03	flg04	sex	trtan
1	783	No	No	Yes	No	M	2
2	784	No	No	No	No	F	1
3	785	Yes	No	No	No	F	1
4	787	Yes	No	No	Yes	M	1
5	788	Yes	No	No	No	F	1
6	800	Yes	No	Yes	No	M	2
7	821	Yes	No	No	No	M	2

STEP 2: USING APPENDIX 2 PROGRAM

Once we obtain the above dataset for a total of 197 subjects based on sample data, we will derive the variables ENDPOINT, EVENT and COUNT using Proc Tabulate and Data Step with arrays. Classification variable SEX is mentioned directly below, whereas this is passed into a macro variable in appendix program which helps to obtain for different categories.

```
proc tabulate data = adch
                out = tabrr_ ;
    class flg01 flg02 flg03 flg04 trtan;
     class sex /missing;
     tables (flg01 flg02 flg03 flg04) * sex,
    trtan*(N*F=3.0 COLPCTN) / printmiss;
run;
```

MISSING option can be used in PROC TABULATE statement or in the CLASS statement. If a programmer wants to use MISSING to apply only to selected class variables, but not to others, then we can specify MISSING in a separate CLASS statement with the selected variables as shown in the program. The MISSING option includes observations

with missing values of a class variable in the report. Hence in the below output from PROC TABULATE we get counts for category that did not exist i.e. last record for FLG04 as 'Yes' where TRTAN = 2 and SEX = 'M' is created with N=missing. For RR it is a must that we have all possible values for a category for an event. Once we have the rows for all categories and for the N=missing records we should make the value to be 0 required for RR. Another approach is to create a dummy dataset with all possible combinations and then merge with original counts dataset. When compared to PROC TABULATE it uses more steps and time from the programmer.

TABRR_

	flg03	flg04	sex	trtan	N
17	No		F	1	29
18	No		F	2	68
19	No		M	1	27
20	No		M	2	61
21	Yes		F	1	1
22	Yes		F	2	1
23	Yes		M	1	3
24	Yes		M	2	7
25		No	F	1	29
26		No	F	2	66
27		No	M	1	29
28		No	M	2	68
29		Yes	F	1	1
30		Yes	F	2	3
31		Yes	M	1	1
32		Yes	M	2	

Above dataset is used and transformed as per the following variables.

ENDPOINT = Criterion of interest defined per specifications, for e.g. lab value = 10 then 'flg01'

EVENT = if it satisfies the endpoint as defined above then 'Yes' otherwise 'No'

COUNT = Number of subjects with 'Yes' or 'No' by desired classification variables (SEX, TRTAN)

	endpoint	sex	trtan	event	count
17	flg03	F	1	No	29
18	flg03	F	2	No	68
19	flg03	M	1	No	27
20	flg03	M	2	No	61
21	flg03	F	1	Yes	1
22	flg03	F	2	Yes	1
23	flg03	M	1	Yes	3
24	flg03	M	2	Yes	7
25	flg04	F	1	No	29
26	flg04	F	2	No	66
27	flg04	M	1	No	29
28	flg04	M	2	No	68
29	flg04	F	1	Yes	1
30	flg04	F	2	Yes	3
31	flg04	M	1	Yes	1
32	flg04	M	2	Yes	0

Above dataset is obtained from TABRR_ by using the following code:

```
data tabrr2_;
        attrib  endpoint length=$12
                    event  length=$5;
        set tabrr_ (drop=_: ) ;
        array aflg(*) flg: ;
        do i=1 to dim(aflg) ;
                if not missing(aflg[i]) then
                        do;
                                event=aflg[i];
                                call vname(aflg[i], endpoint);
                        end;
        end;
        if missing(n) then n=0;
        rename n=count;
        drop i flg:;
run;
```

Please note above that use of array function have simplified the programming processing. It has helped to read and analyze repetitive data with minimal programming steps whereas if this was done in another method of only using data steps and merge options it would have taken several additional steps to achieve the same result.

This output is in the similar format as mentioned in Method 1 and correct Relative Risk can be obtain by sorting the data before passing through proc Freq.

CONCLUSION

This paper has displayed the importance of arranging the input dataset before calling the SAS® statistical RR procedure. It has shown how SAS® generates different results but only one is the desired correct output. All outputs for RR calculation were generated without any warning or error messages in the log files which makes it difficult for the programmers to debug unless the programmer is already aware of such scenarios and carefully reviewed the code and datasets. Also we have shown the uses of different SAS® procedures and functions like PROC TABULATE and ARRAY. These methods avoid additional SAS® programming steps and make it a more efficient way of programming.

REFERENCES

https://en.wikipedia.org/wiki/Statistics

http://www.sas.com/en_us/insights/analytics/statistical-analysis.html

http://support.sas.com/kb/23/003.html

ACKNOWLEDGMENTS

The authors would like to thank their respective management team for their time in reviewing the paper and providing with valuable comments.

CONTACT INFORMATION

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

Ravi Kankipati
Seattle Genetics, Inc.
21823 – 30th Drive S.E.
Bothell, WA 98102
425-527-4890
rkankipati@seagen.com

Abhilash Chimbirithy
Merck & Co, Inc.
126 E Lincoln Avenue,
Rahway, NJ 07065
732-594-6962
chimbirithy_abhilash@merck.com

APPENDIX 1: PREPARE ADRR DATASET

Please note that in the below section we have provided only the key programming steps. With this program user will not be able to generate the final table as displayed under table requirements section. Formatting steps have not been included in this paper.

Please feel free to contact author for the complete program and test data.

```
* ADLB here contains data which meets at least one of 4 defined criterions FLG01-FLG04;
* Output of this macro is ADRR dataset to location mylib and a global macro allflgs

* Location of test data ADSL, ADLB;
libname mylib 'H:\Studies\Pharmasug\Test';


*-------------------------------------------------------------------------------------------------------*;
*- Step : Transpose to have one subjid and each paramn as a separate variable  -*;
*-------------------------------------------------------------------------------------------------------*;

*Count subject once per occurrence so eliminate the duplicates of subjid paramn;
*the dataset contains subjects that have atleast one hit of the mock table conditions
 create a dummy variable val that helps for transpose;

proc sql;
        create table adtemp as
                select distinct usubjid,
                        paramn,
                        trtan,
                        'Yes' as val
                from mylib.adlb
   order by usubjid, paramn;
quit;

*Transpose the adtemp dataset to get one subjid creating new variables based on paramn;
* the following dataset contains USUBJID FLG01 FLG02 FLG03 FLG04;

proc transpose data=adtemp
        out=adflags (drop=_:);
        by usubjid;
        id paramn;
        var val;
run;

proc sort data=adflags;
        by usubjid;
run;
```

```
*-------------------------------------------------------------------------------*;
*- Step : Create the ADCH dataset by merging with ADSL                          -*;
*-------------------------------------------------------------------------------*;
*this dataset has one subjid and the rows of table as a separate flag with values yes or no ;
data mylib.adRR;
        merge mylib.adsl (in=a)
                adflags (in=b keep=usubjid flg:)
        ;
        array afl(*) flg:;
        by usubjid;

        if a;

        do i=1 to dim(afl);
                if missing(afl[i]) then
                        afl[i]='No';
        end;

        drop i;
run;

* get all the flags into a macro variable based on the dataset ;
proc sql;
        select name into: allflgs
        separated by ' '
        from dictionary.columns
        where libname = 'MYLIB' and memname = 'ADRR'
        and upcase(name) like 'FLG%';
quit;

%put &allflgs ;
```

APPENDIX 2: RELRISK MACRO

```
* The macro relrisk expects the following:
%* * MYLIB: Library location of dataset ADRR
        ADRR: Analysis Dataset Relative Risk with the data format as defined in paper
    ALLFLGS:  Global Macro variable containing all flags from ADRR after running appendix 1 program
              which is %let allflgs = FLG01 FLG02 FLG03 FLG04
   subgroup: macro variable which can be passed the group variable for ex: SEX
              if you have AGE categories defined then you can pass that ex: AGECAT
       subcat: macro variable which can be missing for overall subgroup category
              for a subset of Male subjects declare subcat=M
              for a subset of age < 60 subjects declare subcat = AGELT60 (should be in dataset)

OUTPUT:  After running this macro you can see datasets in work based on call starting with RR_
        ;

%macro relrisk (subgroup=,
                subcat=,
               );

* to suppress all output from tabulate use the following and turn the listing destination on later ;
 *ods _all_ close ;

*to get overall counts value of subcat is missing otherwise we subset for relevant  category;
 proc tabulate data = mylib.adrr
              out = tabrr_&subcat.(drop=_page_ _table_);

   %if &subcat. ne %then
        %do;
            where &subgroup.="&subcat.";
         %end;
        class &allflgs. trtan;
        class &subgroup. /missing;
        tables (&allflgs.)*&subgroup., trtan*(N*F=3.0 COLPCTN) /printmiss;
run;

*-------------------------------------------------------------------------------------------------------*;
*- Step : Prepare the data in the format of specifications to calculate RR and p-values   -*;
*-------------------------------------------------------------------------------------------------------*;
*after this step the variable endpoint will contain the flags;
data tabrr2_&subcat.;
        attrib  endpoint length=$12
                event length=$5;
        set tabrr_&subcat. (drop=_type_ );
        array aflg(*) flg:;
```

```
        do i=1 to dim(aflg);
                if not missing(aflg[i]) then
                        do;
                                event=aflg[i];  * assigns value of flag i.e. Yes or No;
                                call vname(aflg[i], endpoint); *assigns name of flag i.e flg30;
                        end;
        end;

        if missing(n) then
                n=0;

        rename n=count;
        drop i flg:;
run;

*-------------------------------------------------------------------------------*;
*- Step : Sort the dataset before getting Relative Risk statistics   -*;
*-------------------------------------------------------------------------------*;

proc sort data=tabrr2_&subcat. (drop=pctn:)
        out=tabrr3_&subcat.;
        by endpoint &subgroup. trtan descending event;
run;

%if &subcat. EQ %then
 %do;
proc freq data=tabrr3_&subcat. order=data;
        weight count /zeros;
        tables &subgroup.*trtan*event /cmh;
        by endpoint;
        output out=rout cmh bdchi;
run;
%end;

%else %do;
 proc freq data=tabrr3_&subcat. order=data;
        weight count /zeros;
        tables  trtan*event /cmh;
        by endpoint;
        output out=rout cmh bdchi;
 run;
 %end;

*-------------------------------------------------------------------------------*;
*- Step : get the required values of RR, p values into a dataset    -*;
*-------------------------------------------------------------------------------*;

 data rr_&subcat.(keep=endpoint rr pval);
```

```
        set rout;
         RR=trim(left(put(_LGRRC1_, 5.2)))||' ('||trim(left(put(L_LGRRC1, 5.2)))||
         ',  '||trim(left(put(U_LGRRC1, 5.2)))||')';
        pval=trim(left(put(P_CMHGA, 5.3)));
run;

  *ods listing;

proc datasets lib =  work memtype=data nolist;
        save rr:;
quit;

proc print data = rr_&subcat.;
        title "Output for &subgroup. and &subcat ";
        var endpoint rr pval;
run;

%* After obtaining Relative Risk values in the dataset res_&subcat. you can obtain
   the counts and percentage for each treatment group separately and can merge with this
   dataset for final table display
   Please feel free to reach out to the author for that code snippet if it's helpful;

%mend;


%* ----------------------------------------------------;
%* -- Sample calls for this macro   -----;
%* ----------------------------------------------------;

%let msubgp=SEX;

*for overall subgroup category here for all Male, Female subjects;
%relrisk(subgroup=&msubgp., subcat=);
* output dataset: RR_;

*within subgroup = Sex for Male subjects;
%relrisk(subgroup=&msubgp., subcat=M);
*output dataset: RR_M;

*within subgroup = Sex, for Female Subjects;
%relrisk(subgroup=&msubgp., subcat=F);
*output dataset: RR_F;
```