Let's Get FREQy with our Statistics: Data-Driven Approach to Determining Appropriate Test Statistic

Richann Watson, DataRich Consulting, Batavia, OH
Lynn Mullins, PPD, Cincinnati, OH

ABSTRACT

As programmers, we are often asked to program statistical analysis procedures to run against the data. Sometimes the specifications we are given by the statisticians outline which statistical procedures to run. But other times, the statistical procedures to use need to be data dependent. To run these procedures based on the results of previous procedures' output requires a little more preplanning and programming. We present a macro that dynamically determines which statistical procedure to run based on previous procedure output. The user can specify parameters (for example, fshchi, plttwo, catrnd, bimain, and bicomp), and the macro returns counts, percents, and the appropriate p-value for Chi-Square versus Fisher Exact, and the p-value for Trend and Binomial CI, if applicable.

INTRODUCTION

Running statistical significance tests and creating p-values is a common practice for generating statistical analyses. But what if the statistic that you want to run is based on the data? You could run the program first and look at the output and then change the program so that it produces the statistic that you want and that applies to the data. But what if the data changes? You would have to repeat this process again. Wouldn't it be nice if you did not need to run code multiple times? We have created a macro to do just that. It runs the statistical test requested based on the data. This paper will discuss what the macro does and explain the macro parameters. In addition, sample output will be shown.

MACRO PARAMETERS

There are 12 macro parameters used in the program. The data set name (*indsn*) and the sort variables (*sortby*) macro parameters are required to be passed to the macro. In addition to the two required parameters that need to be passed to the macro, there are three macro parameters that are needed for the macro to execute. These are *kpvars, tbvars,* and *grpvar.* If these parameter values are not passed to the macro, then the values will be based on the *sortby* parameter value.

List of all parameters and description and any default values are provided in Table 1.

Parameter	Description	Default Value	Required
indsn	**input data set - include libname (i.e., adam.adeff)**		**Yes**
sortby	**sort data - if need to sort descending specify**		**Yes**
whrcls	where clause used to subset input data set		No
kpvars	variables to keep	*sortby**	No
tbvars	table statement	*sortby**	No
grpvar	group variable (i.e., variable that p-value will be for)	*sortby**	No
expcnt	minimum expected counts for each cell used for determining if Chi-Sqr or Fisher Exact is used	5	No
fshchi	threshold used to determine whether Chi-Sqr or Fisher Exact p-value will be used	0.25	No

Parameter	Description	Default Value	Required
plttwo	option for a twoway plot, if want 2-way plot specify GROUPVERTICAL, GROUPHORIZONTAL or STACKED		No
catrnd	if Cochran-Armitage trend test is needed	Y	No
bimain	if two group comparison and/or proportional Binomial CI is needed then specify main group that others will be compared against; embed in quotes		No
bicomp	if two group comparison and/or proportional Binomial CI are needed then specify all comparator groups; embed each group in quotes separated by an exclamation mark (!)		Conditional on *bimain*

* If macro parameter is not specified, then the value will be determined by what is provided in the *sortby* parameter

Table 1. Macro Parameters for RUNFREQ

MACRO PROCESS

The macro is fairly simple and has six main steps as described below.

1. Determine the value of the three necessary macro parameters using the *sortby* parameter if they are not specified. Refer to Table 1 for the three parameters. Also, generate macro parameters that are necessary for the rest of the macro to execute.

2. Retrieve the data using *indsn* parameter.

3. Execute the Frequency procedure with the options to produce a cross tabulation, Chi-Square statistic, Fisher Exact statistic, and with any specific options specified in the macro call to produce Cochran-Armitage trend test and/or frequency plot. More detail is provided in the PROC FREQ from the Step 3 section below.

4. The cross-tabulation frequency can be used to determine if Chi-Square or Fisher Exact should be used. This is based on if more than the desired threshold of cells has expected counts is less than the desired expected counts (i.e., if more than 25% of cells have expected counts less than 5).

5. If group comparison and/or Binomial Proportions CI (BICI) are needed, then the main group is specified separately from the comparison groups. The macro will determine if a pairwise comparison and/or a BICI can be done. If a pairwise comparison can be done, the macro will loop through each comparison group and produce a Chi-Square and Fisher Exact test statistic for each pairwise comparison. If a BICI can be generated, the macro will generate one for each pair and put the BICI in the format of (x.x, x.x). Note BICI will only calculate for 2x2, so you may get a warning indicating the data being compared is not 2x2. See Optional Output – Pairwise Comparison with Binomial Proportion CI section more details. For pairwise comparison if there are more than 2 variables it skips this section entirely.

6. Depending on which test statistic should be used based on finding in step 4, the test statistic and p-value are captured in a data set. If there are more than 2 variables the 'Table' variable is also kept to distinguish between the different OALL records.

PROC FREQ FROM STEP 3

Below is the PROC FREQ macro code that is executed for step 3 above. The red text indicates the macro language code based on the parameters *catrnd* and **plttwo** which will be described in the next section:

```
ods output crosstabfreqs=ctf (where=(_TYPE_ ne '00') drop = Table _TABLE_
Missing Percent);
ods output chisq = chi_oall;
ods output fishersexact = fis_oall;
%if &catrnd = Y %then ods output trendtest = trend;
proc freq data = outdsn order = data;
    tables &tbvars / OUTPCT chisq cmh fisher expected
                    %if &plttwo ne %then
                       plots=freqplot(twoway=&plttwo);
                    %if &catrnd = Y %then trend;
   ;
run;
```

Macro Parameters Used in the PROC FREQ Step 3

The two parameters used in the PROC FREQ Step 3 of the macro are *catrnd* and *plttwo*. If *catrnd* is Y then the Cochran-Armitage trend test is run and the data set TREND is generated. If *plttwo* is specified, then the PROC FREQ will produce the corresponding plot based on the value of *plttwo*.

The only possible values for *plttwo* are:

- GROUPHORIZONTAL

- GROUPVERTICAL

- STACKED

Details on these plots are provided under the section with the corresponding macro parameter value.

OUTPUT

Two main data sets are created, and additional data sets are created if the Cochran-Armitage trend test is needed and if Binomial Proportion CIs are needed. Frequency plots can also be generated.

CROSS TABULATION FREQUENCIES

A data set named CTF will be created that contains all frequency counts and the percentages. There is a _TYPE_ variable that is included. _TYPE_ is character and it sets flags that represent whether or not the variables are populated. Using the generic PROC FREQ code below we will explain how to interpret _TYPE_:

```
proc freq data = indsn;
    tables var1 * var2;
run;
```

When the code above is executed, _TYPE_ = '11' would indicate that var1 and var2 are both populate; _TYPE_ = '10' would indicate var1 is populated and var2 is null; _TYPE_ = '01' would indicate var1 is null and var2 is populated.

You need to keep _TYPE_ = '11' to get cross tab counts. If the group variable is the first variable (i.e., var1) then you need to use the Row Pct since we want percentages based off the total for that specific group value.   Otherwise the percentage would be based off the specified response. See SAS Output 1. If the group variable is second (i.e., var2) then you need to use the Col Pct value since the group is the column variable. See SAS Output 2.

```
CRIT1FL(Criterion 1 Evaluation Result Flag)

Frequency │
Expected  │
Percent   │
Row Pct   │
Col Pct   │ Y        │ N        │ Total
──────────┼──────────┼──────────┼────────
ARM D     │      24  │       4  │     28
          │  24.283  │  3.7168  │
          │   21.24  │    3.54  │  24.78
          │   85.71  │   14.29  │
          │   24.49  │   26.67  │
──────────┼──────────┼──────────┼────────
ARM C     │      24  │       5  │     29
          │   25.15  │  3.8496  │
          │   21.24  │    4.42  │  25.66
          │   82.76  │   17.24  │
          │   24.49  │   33.33  │
──────────┼──────────┼──────────┼────────
ARM B     │      22  │       3  │     25
          │  21.681  │  3.3186  │
          │   19.47  │    2.65  │  22.12
          │   88.00  │   12.00  │
          │   22.45  │   20.00  │
──────────┼──────────┼──────────┼────────
ARM A     │      28  │       3  │     31
          │  26.885  │   4.115  │
          │   24.78  │    2.65  │  27.43
          │   90.32  │    9.68  │
          │   28.57  │   20.00  │
──────────┼──────────┼──────────┼────────
Total           98         15        113
             86.73      13.27     100.00
```

SAS Output 1. Cross Tabulation with Group as First Variable

```
CRIT1FL(Criterion 1 Evaluation Result Flag)
          TRTA(Actual Treatment)

Frequency │
Expected  │
Percent   │
Row Pct   │
Col Pct   │ ARM D   │ ARM C   │ ARM B   │ ARM A   │
Total
──────────┼─────────┼─────────┼─────────┼─────────┼
Y         │     24  │     24  │     22  │     28  │
98        │         │         │         │         │
          │ 24.283  │  25.15  │ 21.681  │ 26.885  │
          │  21.24  │  21.24  │  19.47  │  24.78  │
86.73     │         │         │         │         │
          │  24.49  │  24.49  │  22.45  │  28.57  │
          │  85.71  │  82.76  │  88.00  │  90.32  │
──────────┼─────────┼─────────┼─────────┼─────────┼
N         │      4  │      5  │      3  │      3  │
15        │         │         │         │         │
          │ 3.7168  │ 3.8496  │ 3.3186  │  4.115  │
          │   3.54  │   4.42  │   2.65  │   2.65  │
13.27     │         │         │         │         │
          │  26.67  │  33.33  │  20.00  │  20.00  │
          │  14.29  │  17.24  │  12.00  │   9.68  │
──────────┼─────────┼─────────┼─────────┼─────────┼
Total          28        29        25        31
113
             24.78     25.66     22.12     27.43
100.00
```

SAS Output 2. Cross Tabulation with Group as Second Variable

The data in SAS Output 1 and SAS Output 2 is captured in the cross-tabulation frequencies (CTF) data set created by PROC FREQ. The CTF data set contains all necessary information to produce a table similar to Display 1, including n and % for each level and total.

Treatment	Category	n (%)
ARM A (N=XX)	Success	xx (xx.x)
	Failure	xx (xx.x)
	Total	xx
ARM B (N=XX)	Success	xx (xx.x)
	Failure	xx (xx.x)
	Total	xx
ARM C (N=XX)	Success	xx (xx.x)
	Failure	xx (xx.x)
	Total	xx
ARM D (N=XX)	Success	xx (xx.x)
	Failure	xx (xx.x)
	Total	xx

Note: Denominator based on 'n' in Total row

Display 1. Sample Table Shell

As illustrated in SAS Data Set 1, the Frequencies and RowPercent on the rows where _TYPE_ = '11', are used to create the 'xx (xx.x)' for the 'Success' and 'Failure' rows. The Frequencies on rows where _TYPE_ = '10' are used to create 'xx' for the 'Total' rows.

TRTA	CRIT1FL	_TYPE_	Frequency	Expected	RowPercent	ColPercent
ARM D	Y	11	24	24.2832	85.7143	24.4898
ARM D	N	11	4	3.7168	14.2857	26.6667
ARM D		10	28	.	.	.
ARM C	Y	11	24	25.1504	82.7586	24.4898
ARM C	N	11	5	3.8496	17.2414	33.3333
ARM C		10	29	.	.	.
ARM B	Y	11	22	21.6814	88	22.449
ARM B	N	11	3	3.3186	12	20
ARM B		10	25	.	.	.
ARM A	Y	11	28	26.885	90.3226	28.5714
ARM A	N	11	3	4.115	9.6774	20
ARM A		10	31	.	.	.
	Y	01	98	.	.	.
	N	01	15	.	.	.

SAS Data Set 1. Cross Frequency Tabulation - CTF

CHI-SQUARE OR FISHER EXACT TEST STATISTIC

The macro produces a data set called T_PVALS which will contain the appropriate test statistic and p-value. A table like the sample shell in Display 2 can be created with the test and p-values included from the data set created.

```
Treatment                     Category              n (%)      Test*     P-Value*
ARM A (N=XX)                   Success             xx (xx.x)
                               Failure             xx (xx.x)
                               Total                  xx        x.x        0.xxx

ARM B (N=XX)                   Success             xx (xx.x)
                               Failure             xx (xx.x)
                               Total                  xx        x.x        0.xxx

ARM C (N=XX)                   Success             xx (xx.x)
                               Failure             xx (xx.x)
                               Total                  xx        x.x        0.xxx

ARM D (N=XX)                   Success             xx (xx.x)
                               Failure             xx (xx.x)
                               Total                  xx        x.x        0.xxx

Note: Denominator based on 'n' in Total row

* Pearson's Chi-Square or Fisher's Exact tests will be used, as appropriate (e.g., if
25% of cells have expected counts < 5 then Fisher's Exact test will be used)
```

Display 2. Sample Shell Incorporating Test Statistic

The macro will determine if Chi-Square or Fisher Exact will be used based on the data set. SAS *Data* Set 2 shows that Fisher Exact was used. If Chi-Square would have been used the test would have been 'CHI'.

TRTAN is created based on **bicomp** macro parameter. It will be considered the numeric equivalent of **grpvar**. **bicomp** contains a list of all the comparators and the macro will parse it out and assign a numeric value to **grpvar** (i.e., create &grpvar.n) based on the iteration.

```
bicomp = 'ARM B'!'ARM C'!'ARM D'
```
→

ARM B = 1
ARM C = 2
ARM D = 3

Note that if only an overall treatment comparison is needed, then there is no need to specify **bicomp**. The OALL row in T_PVALS will be generated with the basic run as illustrated in SAS Data Set 3.

TRTAN	pvalue	tvalue	test
1	1	1	FIS
2	0.4653	0.47	FIS
3	0.6978	0.7	FIS
OALL	0.8546	0.85	FIS

SAS Data Set 2. Appropriate Test Statistic and P-value with Pairwise Comparison – T_PVALS

TRTAN	pvalue	tvalue	test
OALL	0.8546	0.85	FIS

SAS Data Set 3. Appropriate Test Statistic and P-Value without Pairwise Comparison – T_PVALS

OPTIONAL OUTPUT

By default, a Cochran-Armitage Trend test will be produced since the default value for the macro parameter is 'Y'. However, if a trend test is not needed, then the value can be set to null or some value other than 'Y'. In addition, if the Binomial Proportion CI is needed, then the two macro parameters for BICI would need to be specified.

Cochran-Armitage Trend Test

The macro performs a Cochran-Armitage trend test if the value of the **catrnd** parameter = 'Y', which is the default. A trend test is only allowed on a 2xR or Rx2 table.

It contains all the necessary information to produce a table similar to Display 3.

Treatment	Category	n (%)	Test*	P-Value*	Cochran-Armitage Trend Value	Cochran-Armitage Trend P-value
ARM A (N=XX)	Success	xx (xx.x)				
	Failure	xx (xx.x)				
	Total	xx	x.x	x.xxx		
ARM B (N=XX)	Success	xx (xx.x)				
	Failure	xx (xx.x)				
	Total	xx	x.x	x.xxx		
ARM C (N=XX)	Success	xx (xx.x)				
	Failure	xx (xx.x)				
	Total	xx	x.x	x.xxx		
ARM D (N=XX)	Success	xx (xx.x)				
	Failure	xx (xx.x)				
	Total	xx	x.x	x.xxx	x.xx	0.xxx

Note: Denominator based on 'n' in Total row

* Pearson's Chi-Square or Fisher's Exact tests will be used, as appropriate (e.g., if 25% of cells have expected counts < 5 then Fisher's Exact test will be used)

Display 3. Sample Shell Incorporating Cochran-Armitage Trend Statistic

As illustrated in SAS Data Set 4, the values for the sample shell above are created in the output data set. The one-sided p-value tests for either an increase in response as dose increases or decrease in response as dose increases. SAS will determine whether it is a left-sided trend, or a right-sided trend based on whether the expected value is greater

than the observed value (right-sided trend) or less than or equal the observed value (left-sided). Since SAS is determining the direction of the trend the value of the variable Name1 may be 'PL_TREND' for a left-side p-value or 'PR_TREND' for a right-side p-value. In SAS Data Set 4, it was a right-side p-value and so the value of Name1 = 'PR_TREND'.

Cochran-Armitage Trend Test (TREND)

Table	Name1	Label1	cValue1	nValue1
Table TRTA * CRIT1FL	_TREND_	Statistic (Z)	0.6903	0.69028
Table TRTA * CRIT1FL	PR_TREND	One-sided Pr > Z	0.245	0.245009
Table TRTA * CRIT1FL	P2_TREND	Two-sided Pr > |Z|	0.49	0.490018

SAS Data Set 4. Cochran-Armitage Trend Test output data set

Pairwise Comparison with Binomial Proportion CI

The macro performs a pairwise comparison with binomial proportion CI test if the values of the ***bimain*** and ***bicomp*** parameters are provided. It contains all the necessary information to produce a table similar to Display 4. Note that the values for these two macro parameters must be the values of the first variable on the table statement in PROC FREQ.

```
                                                        Treatment          Binomial
Treatment        Category        n (%)   Test*   P-Value*   Comparison      Proportion
───────────────────────────────────────────────────────────────────────────────────
ARM A (N=XX)     Success      xx (xx.x)

                 Failure      xx (xx.x)

                 Total           xx       x.x     0.xxx

ARM B (N=XX)     Success      xx (xx.x)

                 Failure      xx (xx.x)

                 Total           xx       x.x     0.xxx  ARM B – ARM A    (x.x, x.x)

ARM C (N=XX)     Success      xx (xx.x)

                 Failure      xx (xx.x)

                 Total           xx       x.x     0.xxx  ARM C – ARM A    (x.x, x.x)

ARM D (N=XX)     Success      xx (xx.x)

                 Failure      xx (xx.x)

                 Total           xx       x.x     0.xxx  ARM D – ARM A    (x.x, x.x)
───────────────────────────────────────────────────────────────────────────────────
Note: Denominator based on 'n' in Total
row

* Pearson's Chi-Square or Fisher's Exact tests will be used, as appropriate (e.g., if 25% of
cells have expected counts < 5 then Fisher's Exact test will be used)
```

Display 4. Sample Shell Incorporating Pairwise Comparison with Binomial Proportion CI Statistic

If a pairwise comparison is needed, you still need to specify the ***bimain*** and ***bicomp*** parameters but keep in mind it will produce the pairwise comparison and may produce a program defined warning for the BICI if it is not a 2x2 table:

```
     WARNING: data needs to be 2x2 to do BiCi - at least one variable has more than 2 levels or
       there are more than two variables
```

Note that if there are more than two variables specified on the table statement, then the program will not run the pairwise comparison and BICI portion even if ***bimain*** and ***bicomp*** are specified. It will ignore the code thus BICI data set will not be produced and the t_pvals will only have the OALL record(s).

As illustrated in SAS Data Set 5, the values for the Display 4 sample shell above are created in the output data set. The BICI data set will provide the lower and upper confidence limit for the proportion of the first entry. For example, the proportion for ARM D with a response of 'N' is 14.2857 thus the BICI is calculated based on that proportion.

L_RDIF1	U_RDIF1	TRTAn	bici
-0.18772	0.14127	1	(-18.8, 14.1)
-0.24807	0.09679	2	(-24.8, 9.7)
-0.21231	0.12014	3	(-21.2, 12.0)

SAS Data Set 5. Binomial Proportion CI output data set

Frequency Plots

If a plot is desired, then the type of the plot should be specified in the macro parameter *plttwo*. The option specified in the macro call will determine the layout of the plot. In addition, the table variables will determine the layout. For example, if the table statement in PROC FREQ is TRTA * CRIT1FL then the orientation will be as displayed as illustrated in

Figure 1, Figure 3 and Figure 5. If the table statement in PROC FREQ is CRIT1FL * TRTA then the orientation will be as displayed shown in Figure 2, Figure 4 and Figure 6.

GROUPHORIZONTAL
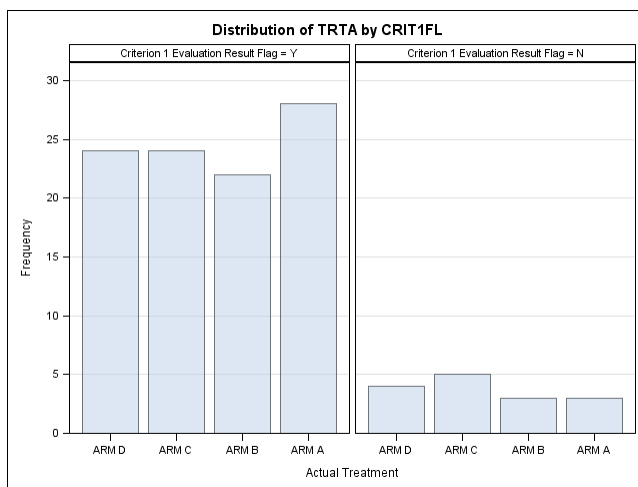
Figure 1. Horizontal: TRTA * CRIT1FL
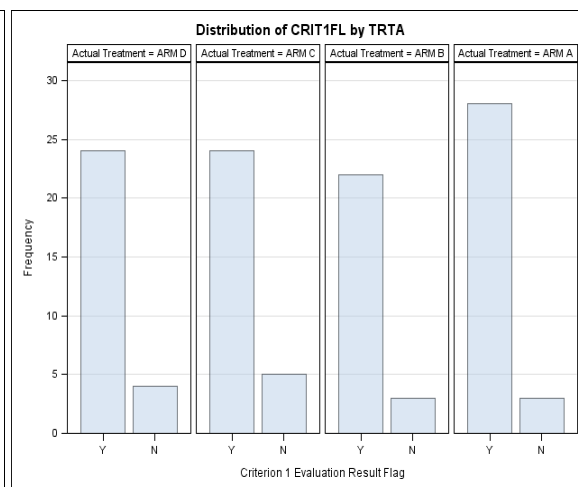
Figure 2. Horizontal: CRIT1FL * TRTA

GROUPVERTICAL
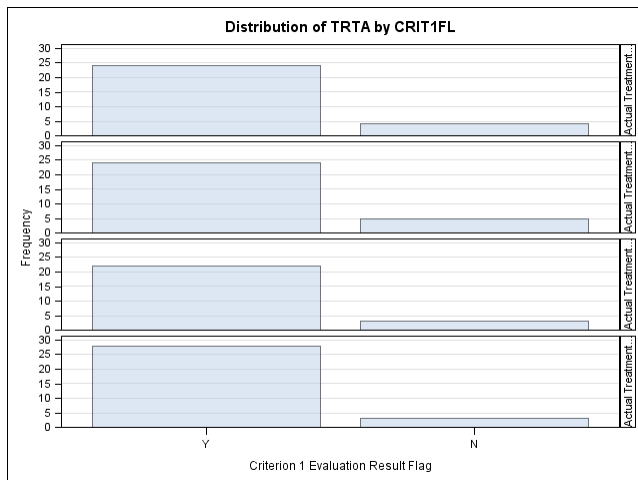

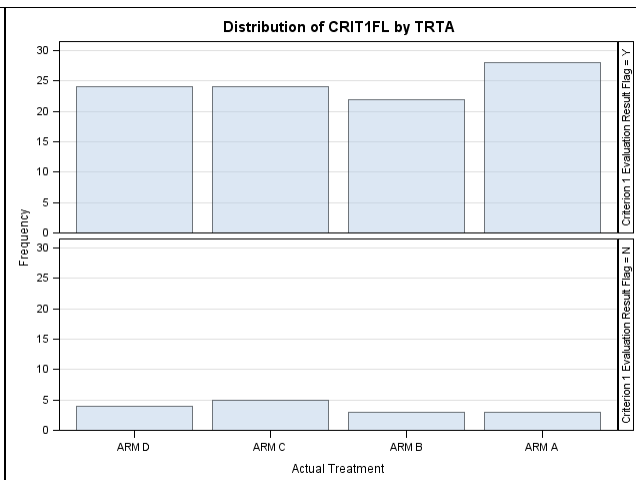**Figure 3. Vertical: TRTA * CRIT1FL**

Figure 4. Vertical: CRIT1FL * TRTA
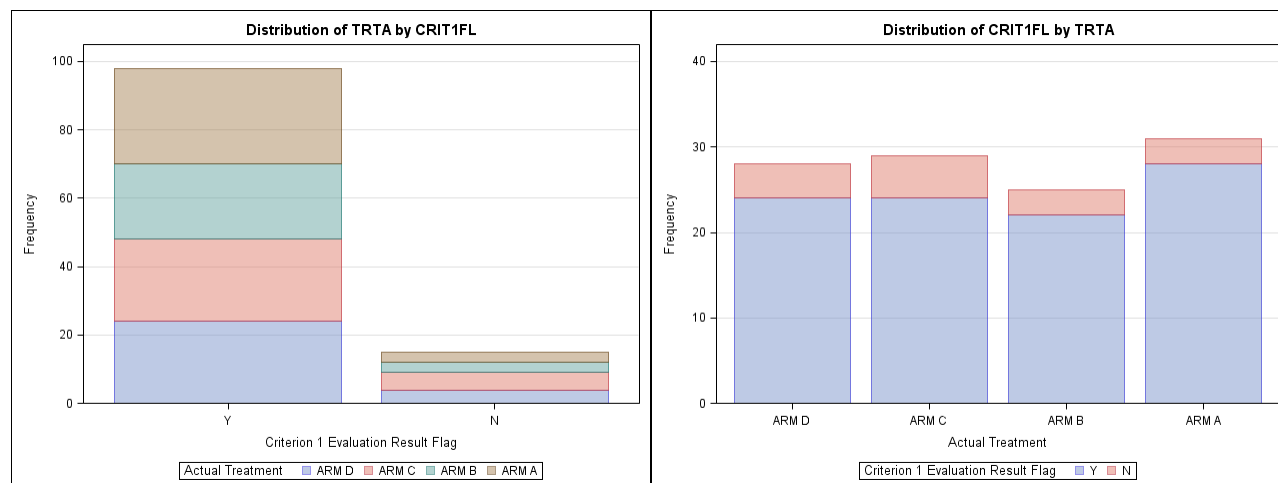
STACKED

Figure 5. Stacked: TRTA * CRIT1FL   Figure 6. Stacked: CRIT1FL * TRTA

ASSUMPTIONS

Although the macro only requires that two macro parameters be provided, it needs several others to execute. Therefore, certain assumptions will be made in lieu of the macro parameters being specified.

1. The where clause should filter the data and structure the data to one record per subject.

2. The following table size assumptions are made based on the desired statistic:

    a. YxRxC – for basic CTF and overall test statistic and p-value

    b. Rx2 or 2xR – for trend test

    c. Only two variables for pairwise comparison and BICI.

    d. Analyzing 2x2 table – for BICI

3. If the macro parameters *kpvars*, *tbvars* and *grpvar* are not specified then the following assumptions are made:

    a. Variables in *sortby* are only ones to be kept

    b. Variables in *sortby* will be used to build table statement

    c. First variable in *sortby* is the main group variable that will be used for pairwise comparison

4. The data set sort must be specified correctly. The descending option needs to be included in the *sortby* parameter if the data needs to be sorted in descending order. This is so the trend test is in the correct direction or so the pairwise comparison or Binomial Proportion CI uses Comparator – Main instead of Main – Comparator.

CONCLUSION

This macro provides an effective solution for running statistics based on the data. It is very robust by providing output data sets and multiple plots. You no longer have to run your program each time the data changes to see which statistic to run. With the passing of just a couple macro parameters, the macro with do it all.

REFERENCES

support.sas.com – Frequency Procedure, 2016-11-22, Available at http://support.sas.com/documentation/cdl/en/procstat/65544/HTML/default/viewer.htm#procstat_freq_syntax01.htmAcknowledgments

ACKNOWLEDGMENTS

Thanks to PPD Management for their reviews and comments. Thanks to our families for their support.

CONTACT INFORMATION

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

Richann Watson
DataRich Consulting
(513) 843-4081
richann.watson@datarichconsulting.com

Lynn Mullins
PPD
(910) 558-4343
Lynn.mullins@ppdi.com

§.sas. Certified Base
       Programmer for SAS'9

The contents of this paper are the work of the authors and do not necessarily represent the opinions, recommendations, or practices of PPD.

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.

APPENDIX

```sas
%macro runfreq(indsn  = ,     /* input data set - include libname (i.e., adam.adeff) */
               sortby = ,     /* sort data - if need to sort descending specify      */
               whrcls = ,     /* (opt'l) where clause used to subset input data set  */
                              /* note that data should already be one record per subj*/
                              /* or where clause will get it to one record per subj  */
               kpvars = ,     /* (opt'l) variables to keep if not specified will be  */
                              /* determined based on the sortby                      */
               tbvars = ,     /* (opt'l) table statement if not specified will be    */
                              /* determined based on the sortby                      */
               grpvar = ,     /* (opt'l) treatment / group variable (i.e., variable  */
                              /* that p-value will be for) - if not specified will   */
                              /* be determined by the sortby                         */
               expcnt = 5,    /* (opt'l) minimum expected counts for each cell used  */
                              /* for determining if Chi-Sqr or Fisher Exact is used  */
               fshchi = 0.25, /* (opt'l) threshold used to determine whether Chi-Sqr */
                              /* or Fisher Exact p-value will be used                */
               plttwo = ,     /* (opt'l) option for a twoway plot, if want 2-way plot*/
                              /* specify GROUPVERTICAL, GROUPHORIZONTAL or STACKED   */
               catrnd = Y,    /* (opt'l) if Cochran-Armitage trend test is needed    */
               bimain = ,     /* (opt'l) if two group comparison and/or proportional */
                              /* Binomial CI is needed then specify main trtm/grp or */
                              /* that others will be compared against-embed in quotes*/
               bicomp =       /* (opt'l) if two group comparison and/or proportional */
                              /* Binomial CI is needed then specify all comparator   */
                              /* treatments / groups - embed each group in quotes    */
                              /* separated by an exclamation mark (!)                */
               );
  /* delete all temp data sets to avoid using incorrect data sets */
  proc datasets library = work nolist memtype = data kill;
  quit;

  /* need to determine keep and table variables if they are not specified */
  data _null_;
     keepvar = tranwrd(upcase("&sortby"), 'DESCENDING', '');
     tablvar = tranwrd(compbl(keepvar), ' ', '*');

     if first(compress(tablvar)) = '*' then tablvar = substr(tablvar, 2);
     if first(reverse(compress(tablvar))) = '*' then
        tablvar = reverse(substr(reverse(compress(tablvar)), 2));

     %if "&kpvars" = "" %then call symputx('kpvars', keepvar);;
     %if "&tbvars" = "" %then call symputx('tbvars', tablvar);;
     %if "&grpvar" = "" %then call symputx('grpvar', scan(tablvar, 1));;

     x = count(resolve('&tbvars'), '*');

     /* create a variable that will be used to determine individual group variable */
     do i = 1 to x;
       y = i + 1;
       call symputx(cats('grpvar', put(y, 3.)), scan(resolve('&tbvars'), y, '*'));
     end;

     /* count number of variables so the correct value for _TYPE_ can be specified */
     /* since macro variable tbvars is being assigned w/n the data step we need to */
     /* use resolve function so that while being constructed it does not resolve it*/
     /* but will resolve during the data step execution                           */
     call symputx('type', repeat('1', x));

     /* number of group variables on the table statement */
     call symputx('numvars', x + 1);
  run;
```

```sas
   /* so only need to sort once - sort by variables needed for binomial proportion */
   /* i.e., if need comparator group (B) - main group (A) and response (Y) - no    */
   /* response (N) - sort descending group descending response                     */
   proc sort data=&indsn
             out=outdsn (keep = &kpvars);
     by &sortby;
     where &whrcls;
   run;

   /* note that for cochran-armitage trend test we are looking for dose response */
   /* (i.e., did response increase / decrease based on the dose)                 */
   /* so need to look at the one sided p-value                                   */
   /* obtain counts for each treatment / group by response (i.e., _TYPE_='11')   */
   /* if treatment / group is first variable in tables statement & overall counts*/
   /* by treatment / group is needed then use records where _TYPE_='10' and use  */
   /* RowPercent for percentages for each response within a treatment / group    */
   /* if treatment / group is 2nd variable in tables statement & overall counts  */
   /* by treatment / group is needed then use records where _TYPE_='01' and use  */
   /* ColPercent for percentages for each response within a treatment / group    */
   ods output crosstabfreqs = ctf (where = (_TYPE_ ne '00')
                                   drop = Table _TABLE_ Missing Percent);
   ods output chisq = chi_oall;
   ods output fishersexact = fis_oall;
   /* need to semicolon one to end %if and one to end ods statement */
   %if &catrnd = Y %then ods output trendtest = trend; ;
   proc freq data = outdsn order = data;
      tables &tbvars / OUTPCT chisq cmh fisher expected
                       %if &plttwo ne  %then plots=freqplot(twoway=&plttwo);
                       %if &catrnd = Y %then trend;
           ; /* this semicolon ends the tables statement - do NOT delete */
   run;

   /* count the number of expected cells < number specific and divide total */
   /* number of cells to determine if need to use Chi-Square or Fisher test */
   data _null_;
      set ctf (where=(_TYPE_ = "&type")) end=eof;

      retain numcell numcelllt5;
      if _n_ = 1 then do;
         numcell = 0;
         numcelllt5 = 0;
      end;
      numcell + 1;
      if expected < &expcnt then numcelllt5 = numcelllt5 + 1;

      /* determine if a Chi-Square or Fisher test should be performed */
      /* Note keep one record per treatment in odd case they want to  */
      /* do tests by treatment instead of overall                     */
      if eof then do;
         if numcelllt5 / numcell > &fshchi then test = 'FIS';
         else test = 'CHI';
         call symputx ('test', test);
      end;
   run;

   /* if binomial proportions are needed then loop through each comparator group */
   /* will also retrieve Chi-Sqr / Fisher Exact p-value for each comparison      */
   /* can only do pairwise and bici if there are only two variables              */
   %if &bimain ne  and &numvars = 2 %then %do;
      %let nobici = 0;
      %let x = 1;
      %let bigrp = %scan(&bicomp, &x, '!');
```

12

```
      %do %while ("&bigrp" ne "");

         /* determine the number of levels for each var to see if bici can be done */
         proc sql noprint;
            select count(distinct &grpvar2) into :nvar2_&x
            from ctf
            where &grpvar in (&bimain "&bigrp");
         quit;

         ods output chisq = chi_&x;
         ods output fishersexact = fis_&x;
         proc freq data = ctf order = data;
            where _TYPE_ = '11';
            weight FREQUENCY;
            tables &tbvars / %if &&nvar2_&x = 2 %then binomial;
                              alpha=0.05 chisq fisher;
            /* only execute binomial proportion CI if data is 2x2 */
            %if &&nvar2_&x = 2 and &numvars = 2 %then %do;
               exact riskdiff;
               output out=bci_&x (keep = L_RDIF1 U_RDIF1) riskdiff;
            %end;
            where also &grpvar in (&bimain "&bigrp");
         run;

         /* if any one of the outputs can't be produced don't produce any of bici */
         %if &&nvar2_&x > 2 %then %let nobici = 1;;

         %let x = %eval(&x + 1);
         %let bigrp = %scan(&bicomp, &x, '!');
      %end;

      /* only produce bici if each group variable has at most 2 levels  */
      /* only produce bici if there are only two variables on table stmt*/
      /* otherwise produce a message to the log                         */
      %if &nobici ^= 1 and &numvars = 2 %then %do;
         data bici;
            set bci_: indsname= inputdsn;
            length &grpvar.n $8 bici $20;
            /* only want number or 'OALL' portion */
            &grpvar.n = scan(scan(inputdsn, 2), 2, '_');
            /* need to convert values to percentages */
            bici = cats('(', catx(', ', put(L_RDIF1 * 100, 8.1),
                                        put(U_RDIF1 * 100, 8.1)), ')');
         run;
      %end;
      %else %do;
         %put %sysfunc(COMPRESS(W ARNING:)) data needs to be 2x2 to do BiCi - at least
one variable has more than 2 levels or there are more than two variables;
      %end;
   %end;

   /* combine the test statistics into one to determine what is needed later */
   data t_pvals (keep=test &grpvar.n tvalue pvalue %if &numvars > 2 %then table;);
      set &test._: (where = (%if &test = CHI %then statistic = 'Chi-Square';
                              %else name1 = 'XP2_FISH';))
                  indsname = inputdsn;
      length &grpvar.n pvalue tvalue $8;
      test = "&test";
         /* only want number or 'OALL' portion */
         &grpvar.n = scan(scan(inputdsn, 2), 2, '_');

      %if &test = CHI %then %do;
         tvalue = put(value, 8.2);
```

13

```
            pvalue = put(prob, 8.3);
        %end;
        %else %do;
            tvalue = put(nValue1, 8.2);
            pvalue = put(cValue1, 8.3);
        %end;
    run;

%mend runfreq;
```