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

Richann Watson and Lynn Mullins
DataRich Consulting and PPD

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.

INPUT AND OUTPUT

Input:
12 macro parameters
Only 2 are required
– Name of data set
– Sort variables

Output:
SAS Data Sets:
– CTF: cross tabulation with counts and percentages
– T_PVALS: test statistic and associated p-values
– TREND: Cochran-Armitage Trend test if requested
– BICI: Binomial Proportion confidence interval if requested

MACRO PROCESS

1. Certain macro parameters are needed to execute. If these are not specified, then they will be determined based on the value of sortby.
2. Retrieve the data
3. PROC FREQ is executed. By default it will produce a cross tabulation frequency (CTF), Chi-Square statistic and Fisher Exact statistic. Upon request it will produce the Cochran Armitage trend test and a frequency plot.
4. The CTF is used to determine if Chi-Square or Fisher Exact test statistic should be used. This is based on if more than the desired threshold (fshchi) of cells have expected counts that is less than the desired expected counts (expcnt) (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).
6. Depending on which test statistic should be used based on the finding in step 4, the test statistic and p-value are captured in a data set (t_pvals).
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 will do it all.

Contact Information

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

Lynn Mullins
PPD
(910) 558-4343
Lynn.mullins@ppdi.com
SAS® GLOBAL FORUM 2018

April 8 - 11 | Denver, CO
Colorado Convention Center

#SASGF
**Macro Process**

**PROC FREQ**

### PROC FREQ from Step 3 in the Macro Process

- ODS output crosstabfreqs=ctf (where=(\_TYPE\_ ne '00') drop = Table _TABLE_ Missing Percent);
- ODS output chisq = chi_oall;
- ODS output fisherexact = 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;
- /* this semicolon ends the tables statement - do NOT delete */

### PROC FREQ from Step 5 in the Macro Process

Part of the processing of step 1 is to determine the number of variables used in PROC FREQ.

If there are only two variables, a pairwise comparison can be done.

However, in order to do a Binomial Proportions CI, then the program will determine the number of levels for each variable being compared. In order to produce a BiCI, the data must be a 2x2 table. In other words, two variables with at most 2 values each.

A data set is produced for the pairwise comparison and/or BiCI if it was possible to run the tests.

<table>
<thead>
<tr>
<th>Values for <code>plttwo</code></th>
<th>Corresponding Plot</th>
</tr>
</thead>
<tbody>
<tr>
<td>GROUPHORIZONTAL</td>
<td></td>
</tr>
<tr>
<td>GROUPVERTICAL</td>
<td></td>
</tr>
<tr>
<td>STACKED</td>
<td></td>
</tr>
</tbody>
</table>

If `catrnd` is Y then the Cochran-Armitage trend test is run and the output 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`, and `STACKED`.
## Macro Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>indsn</td>
<td>input data set - include libname (i.e., adam.adeff)</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>sortby</td>
<td>sort data - if need to sort descending specify</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>whrcls</td>
<td>where clause used to subset input data set</td>
<td>-</td>
<td>No</td>
</tr>
<tr>
<td>kpvars</td>
<td>variables to keep</td>
<td>sortby*</td>
<td>No</td>
</tr>
<tr>
<td>tbvars</td>
<td>table statement</td>
<td>sortby*</td>
<td>No</td>
</tr>
<tr>
<td>grpvar</td>
<td>group variable (i.e., variable that p-value will be for)</td>
<td>sortby*</td>
<td>No</td>
</tr>
<tr>
<td>expcnt</td>
<td>minimum expected counts for each cell used for determining if Chi-Sqr or Fisher Exact is used</td>
<td>5</td>
<td>No</td>
</tr>
<tr>
<td>fshchi</td>
<td>threshold used to determine whether Chi-Sqr or Fisher Exact p-value will be used</td>
<td>0.25</td>
<td>No</td>
</tr>
<tr>
<td>plttwo</td>
<td>option for a twoway plot, if want 2-way plot specify GROUPVERTICAL, GROUPHORIZONTAL or STACKED</td>
<td>-</td>
<td>No</td>
</tr>
<tr>
<td>catrnd</td>
<td>if Cochran-Armitage trend test is needed</td>
<td>Y</td>
<td>No</td>
</tr>
<tr>
<td>bimain</td>
<td>if two group comparison and/or proportional Binomial CI is needed then specify main group that others will be compared against; embed in quotes</td>
<td>-</td>
<td>No</td>
</tr>
<tr>
<td>bicomp</td>
<td>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 (!)</td>
<td>-</td>
<td>Conditional on bimain</td>
</tr>
</tbody>
</table>

* If macro parameter is not specified, then the value will be determined with what is provided in the sortby parameter
**Default Output Data Sets**

### Cross Tabulation Frequencies (CTF)

<table>
<thead>
<tr>
<th>TRTA</th>
<th>CRIT1FL</th>
<th>TYPE</th>
<th>Frequency</th>
<th>Expected</th>
<th>RowPercent</th>
<th>ColPercent</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM D</td>
<td>Y</td>
<td>11</td>
<td>24</td>
<td>24.2832</td>
<td>85.7143</td>
<td>24.4898</td>
</tr>
<tr>
<td>ARM D</td>
<td>N</td>
<td>11</td>
<td>4</td>
<td>3.7168</td>
<td>14.2857</td>
<td>26.6667</td>
</tr>
<tr>
<td>ARM D</td>
<td></td>
<td>10</td>
<td>28</td>
<td></td>
<td>3.0168</td>
<td></td>
</tr>
<tr>
<td>ARM C</td>
<td>Y</td>
<td>11</td>
<td>24</td>
<td>25.1504</td>
<td>82.7586</td>
<td>24.4898</td>
</tr>
<tr>
<td>ARM C</td>
<td>N</td>
<td>11</td>
<td>5</td>
<td>3.8496</td>
<td>17.2414</td>
<td>33.3333</td>
</tr>
<tr>
<td>ARM C</td>
<td></td>
<td>10</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARM B</td>
<td>Y</td>
<td>11</td>
<td>22</td>
<td>21.6814</td>
<td>88</td>
<td>22.449</td>
</tr>
<tr>
<td>ARM B</td>
<td>N</td>
<td>11</td>
<td>3</td>
<td>3.3186</td>
<td>12</td>
<td>20</td>
</tr>
<tr>
<td>ARM B</td>
<td></td>
<td>10</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARM A</td>
<td>Y</td>
<td>11</td>
<td>28</td>
<td>26.885</td>
<td>90.3226</td>
<td>28.5714</td>
</tr>
<tr>
<td>ARM A</td>
<td>N</td>
<td>11</td>
<td>3</td>
<td>4.115</td>
<td>9.6774</td>
<td>20</td>
</tr>
<tr>
<td>ARM A</td>
<td></td>
<td>10</td>
<td>31</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td></td>
<td>01</td>
<td>98</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
<td>01</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Test Statistic and P-Value (T_PVALS)

<table>
<thead>
<tr>
<th>TRTAn *</th>
<th>pvalue</th>
<th>Tvalue</th>
<th>Test †</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>FIS</td>
</tr>
<tr>
<td>2</td>
<td>0.4653</td>
<td>0.47</td>
<td>FIS</td>
</tr>
<tr>
<td>3</td>
<td>0.6978</td>
<td>0.7</td>
<td>FIS</td>
</tr>
<tr>
<td>OALL</td>
<td>0.8546</td>
<td>0.85</td>
<td>FIS</td>
</tr>
</tbody>
</table>

* VAR1n is set to OALL for the overall comparison and is the one record that will always be produced. VAR1n will be set to some numeric value which is programmatically determined based on the value of bicomp. The order in which the values in bicomp appear correspond to the numeric order.

† Test will either be CHI for Chi-Square or FIS for Fisher’s Exact and the p-value and t-value captured will be associated with the indicated test.

---

### Optional Output Data Sets

#### Binomial Proportion CI (BICI)

<table>
<thead>
<tr>
<th>l_RDIF1</th>
<th>U_RDIF1</th>
<th>TRTAn *</th>
<th>bici</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.18772</td>
<td>0.14127</td>
<td>1</td>
<td>(-18.8, 14.1)</td>
</tr>
<tr>
<td>-0.24807</td>
<td>0.09679</td>
<td>2</td>
<td>(-24.8, 9.7)</td>
</tr>
<tr>
<td>-0.21231</td>
<td>0.12014</td>
<td>3</td>
<td>(-21.2, 12.0)</td>
</tr>
</tbody>
</table>

* VAR1n is programmatically determined. It is based on the value of bicomp. The order in which the values in bicomp appear correspond to the numeric order.

#### Cochran-Armitage Trend Test (TREND)

<table>
<thead>
<tr>
<th>Table</th>
<th>Name1</th>
<th>Label1</th>
<th>cValue1</th>
<th>nValue1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table TRTA * CRIT1FL</td>
<td><em>TREND</em></td>
<td>Statistic (Z)</td>
<td>0.6903</td>
<td>0.69028</td>
</tr>
<tr>
<td>Table TRTA * CRIT1FL</td>
<td>PR_TREND</td>
<td>One-sided Pr &gt; Z</td>
<td>0.245</td>
<td>0.245009</td>
</tr>
<tr>
<td>Table TRTA * CRIT1FL</td>
<td>P2_TREND</td>
<td>Two-sided Pr &gt;</td>
<td>0.49</td>
<td>0.490018</td>
</tr>
</tbody>
</table>

Order of variables on the PROC FREQ table statement will effect the layout of CTF and values of RowPercent and ColPercent. In addition, the output for BICI is assessing the Risk Difference between rows therefore the order in which the table is specified is important. If you which to assess the difference between VARx and the table statement has VARy*VARx then the BICI will produce Risk Difference for VARy.

to see the data set if the table statement in PROC FREQ is CRIT1FL * TRTA.
If a plot is desired, then the type of the plot should be specified. 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 table statement in PROC FREQ is TRTA * CRIT1FL then the orientation will be as displayed below.

- **GROUPHORIZONTAL**

- **GROUPVERTICAL**

- **STACKED**

To see the layout if the table statement in PROC FREQ is CRIT1FL * TRTA.
## Output Data Sets
### Alternate Table Statement

### Default Output Data Sets

<table>
<thead>
<tr>
<th>CRIT1FL</th>
<th>TRTA</th>
<th>TYPE_</th>
<th>Frequency</th>
<th>Expected</th>
<th>RowPercent</th>
<th>ColPercent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>ARM D</td>
<td>11</td>
<td>24</td>
<td>24.2832</td>
<td>24.4898</td>
<td>85.7143</td>
</tr>
<tr>
<td>Y</td>
<td>ARM C</td>
<td>11</td>
<td>24</td>
<td>25.1504</td>
<td>24.4898</td>
<td>82.7586</td>
</tr>
<tr>
<td>Y</td>
<td>ARM B</td>
<td>11</td>
<td>22</td>
<td>21.6814</td>
<td>22.449</td>
<td>88</td>
</tr>
<tr>
<td>Y</td>
<td>ARM A</td>
<td>11</td>
<td>28</td>
<td>26.885</td>
<td>28.5714</td>
<td>90.3226</td>
</tr>
<tr>
<td>Y</td>
<td></td>
<td></td>
<td>10</td>
<td>98.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>ARM D</td>
<td>11</td>
<td>4</td>
<td>3.7168</td>
<td>26.6667</td>
<td>14.2857</td>
</tr>
<tr>
<td>N</td>
<td>ARM C</td>
<td>11</td>
<td>5</td>
<td>3.8496</td>
<td>33.3333</td>
<td>17.2414</td>
</tr>
<tr>
<td>N</td>
<td>ARM B</td>
<td>11</td>
<td>3</td>
<td>3.3186</td>
<td>20</td>
<td>12</td>
</tr>
<tr>
<td>N</td>
<td>ARM A</td>
<td>11</td>
<td>3</td>
<td>4.115</td>
<td>20</td>
<td>9.6774</td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
<td>10</td>
<td>15.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Optional Output Data Sets

### Binomial Proportion CI (BICI)

<table>
<thead>
<tr>
<th>L_RDIF1</th>
<th>U_RDIF1</th>
<th>TRTAn *</th>
<th>biel</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.48307</td>
<td>0.36307</td>
<td>1</td>
<td>(-48.3, 36.3)</td>
</tr>
<tr>
<td>-0.52527</td>
<td>0.19834</td>
<td>2</td>
<td>(-52.5, 19.8)</td>
</tr>
<tr>
<td>-0.50073</td>
<td>0.28095</td>
<td>3</td>
<td>(-50.1, 28.1)</td>
</tr>
</tbody>
</table>

With table statement CRIT1FL * TRTA instead of TRTA * CRIT1FL the RowPercent and ColPercent values are reversed, so extra care should be taken when selecting the percentage.

Binomial proportion CI is based on Row Risk difference. If the need is to assess the TRTA Row Risk difference then TRTA should be listed first. In this illustration, CRIT1FL is first on the table statement and therefore the risk assessment is based on CRIT1FL and not TRTA.
Table Statement in PROC FREQ is CRIT1FL * TRTA

GROUPHORIZONTAL

GROUPVERTICAL

STACKED

to see the layout if the table statement in PROC FREQ is TRTA * CRIT1FL
Paper 2138-2018

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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>indsn</td>
<td>input data set - include libname (i.e., adam.adeff)</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>sortby</td>
<td>sort data - if need to sort descending specify</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>whrcls</td>
<td>where clause used to subset input data set</td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>kpvars</td>
<td>variables to keep</td>
<td>sortby*</td>
<td>No</td>
</tr>
<tr>
<td>tbvars</td>
<td>table statement</td>
<td>sortby*</td>
<td>No</td>
</tr>
<tr>
<td>grpvar</td>
<td>group variable (i.e., variable that p-value will be for)</td>
<td>sortby*</td>
<td>No</td>
</tr>
<tr>
<td>expcnt</td>
<td>minimum expected counts for each cell used for determining if Chi-Sqr or Fisher Exact is used</td>
<td>5</td>
<td>No</td>
</tr>
<tr>
<td>fshchi</td>
<td>threshold used to determine whether Chi-Sqr or Fisher Exact p-value will be used</td>
<td>0.25</td>
<td>No</td>
</tr>
</tbody>
</table>
Table 1. Macro Parameters for RUNFREQ

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>plttwo</td>
<td>option for a twoway plot, if want 2-way plot specify GROUPVERTICAL, GROUPOHORZIOTAL or STACKED</td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>catrnd</td>
<td>if Cochran-Armitage trend test is needed</td>
<td>Y</td>
<td>No</td>
</tr>
<tr>
<td>bimain</td>
<td>if two group comparison and/or proportional Binomial CI is needed then specify main group that others will be compared against; embed in quotes</td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>bicomp</td>
<td>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 (!)</td>
<td>Conditional on bimain</td>
<td></td>
</tr>
</tbody>
</table>

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

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 have 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_:

```sas
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 of the total for that specific group value. Otherwise the percentage would be based off of 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.
### SAS Output 1. Cross Tabulation with Group as First 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.

<table>
<thead>
<tr>
<th>Treatment Category</th>
<th>n (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM A (N=XX)</td>
<td></td>
</tr>
<tr>
<td>Success</td>
<td>xx xx.x</td>
</tr>
<tr>
<td>Failure</td>
<td>xx xx.x</td>
</tr>
<tr>
<td>Total</td>
<td>xx</td>
</tr>
<tr>
<td>ARM B (N=XX)</td>
<td></td>
</tr>
<tr>
<td>Success</td>
<td>xx xx.x</td>
</tr>
<tr>
<td>Failure</td>
<td>xx xx.x</td>
</tr>
<tr>
<td>Total</td>
<td>xx</td>
</tr>
<tr>
<td>ARM C (N=XX)</td>
<td></td>
</tr>
<tr>
<td>Success</td>
<td>xx xx.x</td>
</tr>
<tr>
<td>Failure</td>
<td>xx xx.x</td>
</tr>
<tr>
<td>Total</td>
<td>xx</td>
</tr>
<tr>
<td>ARM D (N=XX)</td>
<td></td>
</tr>
<tr>
<td>Success</td>
<td>xx xx.x</td>
</tr>
<tr>
<td>Failure</td>
<td>xx xx.x</td>
</tr>
<tr>
<td>Total</td>
<td>xx</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>TRTA</th>
<th>CRIT</th>
<th><em>TYPE</em></th>
<th>Frequency</th>
<th>Expected</th>
<th>RowPercent</th>
<th>ColPercent</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM D</td>
<td>Y</td>
<td>11</td>
<td>24</td>
<td>24.2832</td>
<td>85.7143</td>
<td>24.4898</td>
</tr>
<tr>
<td>ARM D</td>
<td>N</td>
<td>11</td>
<td>4</td>
<td>3.7168</td>
<td>14.2857</td>
<td>26.6667</td>
</tr>
<tr>
<td>ARM D</td>
<td></td>
<td>10</td>
<td>28</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARM C</td>
<td>Y</td>
<td>11</td>
<td>24</td>
<td>25.1504</td>
<td>82.7586</td>
<td>24.4898</td>
</tr>
<tr>
<td>ARM C</td>
<td>N</td>
<td>11</td>
<td>5</td>
<td>3.8496</td>
<td>17.2414</td>
<td>33.3333</td>
</tr>
<tr>
<td>ARM C</td>
<td></td>
<td>10</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARM B</td>
<td>Y</td>
<td>11</td>
<td>22</td>
<td>21.6814</td>
<td>88.2667</td>
<td>22.4499</td>
</tr>
<tr>
<td>ARM B</td>
<td>N</td>
<td>11</td>
<td>3</td>
<td>3.3186</td>
<td>12.2414</td>
<td>20.0000</td>
</tr>
<tr>
<td>ARM B</td>
<td></td>
<td>10</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARM A</td>
<td>Y</td>
<td>11</td>
<td>28</td>
<td>26.885</td>
<td>90.3226</td>
<td>28.5714</td>
</tr>
<tr>
<td>ARM A</td>
<td>N</td>
<td>11</td>
<td>3</td>
<td>4.115</td>
<td>9.6774</td>
<td>20.0000</td>
</tr>
<tr>
<td>ARM A</td>
<td></td>
<td>10</td>
<td>31</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>01</td>
<td>98</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>01</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Treatment</th>
<th>Category</th>
<th>n (%)</th>
<th>Test*</th>
<th>p-Value*</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM A [N=XX]</td>
<td>Success</td>
<td>xx (xx.x)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Failure</td>
<td>xx (xx.x)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>xx</td>
<td>x.x</td>
<td>0.xxx</td>
</tr>
<tr>
<td>ARM B [N=XX]</td>
<td>Success</td>
<td>xx (xx.x)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Failure</td>
<td>xx (xx.x)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>xx</td>
<td>x.x</td>
<td>0.xxx</td>
</tr>
<tr>
<td>ARM C [N=XX]</td>
<td>Success</td>
<td>xx (xx.x)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Failure</td>
<td>xx (xx.x)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>xx</td>
<td>x.x</td>
<td>0.xxx</td>
</tr>
<tr>
<td>ARM D [N=XX]</td>
<td>Success</td>
<td>xx (xx.x)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Failure</td>
<td>xx (xx.x)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>xx</td>
<td>x.x</td>
<td>0.xxx</td>
</tr>
</tbody>
</table>

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.

\[
\text{bicomp} = \text{'ARM B'!'ARM C'!'ARM D'} \rightarrow \begin{align*}
\text{ARM B} &= 1 \\
\text{ARM C} &= 2 \\
\text{ARM D} &= 3
\end{align*}
\]

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.

---

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

<table>
<thead>
<tr>
<th>TRTAN</th>
<th>pvalue</th>
<th>tvalue</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>FIS</td>
</tr>
<tr>
<td>2</td>
<td>0.4653</td>
<td>0.47</td>
<td>FIS</td>
</tr>
<tr>
<td>3</td>
<td>0.6978</td>
<td>0.7</td>
<td>FIS</td>
</tr>
<tr>
<td>OALL</td>
<td>0.8546</td>
<td>0.85</td>
<td>FIS</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>TRTAN</th>
<th>pvalue</th>
<th>tvalue</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td>OALL</td>
<td>0.8546</td>
<td>0.85</td>
<td>FIS</td>
</tr>
</tbody>
</table>

---

### 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.

### Cochran-Armitage Trend Test

<table>
<thead>
<tr>
<th>Treatment</th>
<th>Category</th>
<th>n (%)</th>
<th>Test</th>
<th>P-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM A (N=XX)</td>
<td>Success</td>
<td>xx (xx.x)</td>
<td>xx (xx.x)</td>
<td>x.x</td>
</tr>
<tr>
<td>ARM B (N=XX)</td>
<td>Success</td>
<td>xx (xx.x)</td>
<td>xx (xx.x)</td>
<td>x.x</td>
</tr>
<tr>
<td>ARM C (N=XX)</td>
<td>Success</td>
<td>xx (xx.x)</td>
<td>xx (xx.x)</td>
<td>x.x</td>
</tr>
<tr>
<td>ARM D (N=XX)</td>
<td>Success</td>
<td>xx (xx.x)</td>
<td>xx (xx.x)</td>
<td>x.x</td>
</tr>
</tbody>
</table>

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 to 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’.

### Cochrans Armitage Trend Test (TREND)

<table>
<thead>
<tr>
<th>Table TRTA * CRIT1FL</th>
<th>Name1</th>
<th>Label1</th>
<th>cValue1</th>
<th>nValue1</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>TREND</em></td>
<td>Statistic (Z)</td>
<td>0.6903</td>
<td>0.69028</td>
<td></td>
</tr>
<tr>
<td>PR_TREND</td>
<td>One-sided Pr &gt; Z</td>
<td>0.245</td>
<td>0.24509</td>
<td></td>
</tr>
<tr>
<td>P2_TREND</td>
<td>Two-sided Pr &gt;</td>
<td>0.49</td>
<td>0.490018</td>
<td></td>
</tr>
</tbody>
</table>

### SAS Data Set 4. Cochrans 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.

<table>
<thead>
<tr>
<th>Treatment Category</th>
<th>n (%)</th>
<th>Test*</th>
<th>P-Value*</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM A (N=XX)</td>
<td>Success xx (xx.x)</td>
<td>xx</td>
<td>0.xxx</td>
</tr>
<tr>
<td>Failure xx (xx.x)</td>
<td>Total   xx x.x</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ARM B (N=XX)</td>
<td>Success xx (xx.x)</td>
<td>xx</td>
</tr>
<tr>
<td>Failure xx (xx.x)</td>
<td>Total     xx x.x</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ARM C (N=XX)</td>
<td>Success xx (xx.x)</td>
<td>xx</td>
</tr>
<tr>
<td>Failure xx (xx.x)</td>
<td>Total     xx x.x</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ARM D (N=XX)</td>
<td>Success xx (xx.x)</td>
<td>xx</td>
</tr>
<tr>
<td>Failure xx (xx.x)</td>
<td>Total     xx x.x</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 need s 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 BiCl 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 BiCl is calculated based on that proportion.

<table>
<thead>
<tr>
<th>L_RDIF1</th>
<th>U_RDIF1</th>
<th>TRTAn</th>
<th>bici</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.18772</td>
<td>0.14127</td>
<td>1</td>
<td>(-18.8, 14.1)</td>
</tr>
<tr>
<td>0.24807</td>
<td>0.09679</td>
<td>2</td>
<td>(-24.8, 9.7)</td>
</tr>
<tr>
<td>0.21231</td>
<td>0.12014</td>
<td>3</td>
<td>(-21.2, 12.0)</td>
</tr>
</tbody>
</table>

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](image1)

![Figure 2. Horizontal: CRIT1FL * TRTA](image2)
**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 in order 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 will do it all.

REFERENCES

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

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

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.
%macro runfreq(indsn = , /* input data set - include libname (i.e., adam.adeff) */
sortby = , /* sort data - if need to sort descending specify */
whrcls = , /* (opt'1) 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'1) variables to keep if not specified will be */
/* determined based on the sortby */
tbvars = , /* (opt'1) table statement if not specified will be */
/* determined based on the sortby */
grpvar = , /* (opt'1) treatment / group variable (i.e., variable */
/* that p-value will be for) - if not specified will */
/* be determined by the sortby */
expcnt = 5, /* (opt'1) minimum expected counts for each cell used */
/* for determining if Chi-Sqr or Fisher Exact is used */
fshchi = 0.25, /* (opt'1) threshold used to determine whether Chi */
/* Sqr or Fisher Exact p-value will be used */
plttwo = , /* (opt'1) option for a twoway plot, if want 2-way plot */
/* specify GROUPVERTICAL, GROUPHORIZONTAL or STACKED */
catrnd = Y, /* (opt'1) if Cochran-Armitage trend test is needed */
bilmain = , /* (opt'1) 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'1) 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 */
/* need to determine keep and table variables if they are not specified */
data _null_;%if "kpvars" = ""
%then call symputx('kpvars', keepvar);%
%if "tbvars" = ""
%then call symputx('tbvars', tablvar);%
%if "grpvar" = ""
%then call symputx('grpvar', scan(tablvar, 1));%
/* delete all temp data sets to avoid using incorrect data sets */
/* 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;
/* create a variable that will be used to determine individual group variable */
/* number of group variables on the table statement */
call symputx('numvars', x + 1);
run;
/* 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 cochranchi-rmitage 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
  ods output trendtest = trend; ;
%end;
proc freq data = outdsn order = data;
  tables &tbvars / OUTPCT chisq cmh fisher expected
%if &plttwo ne
  plots=freqplot(twoway=&plttwo);
%if &catrnd = Y
  trend;
;%end;
/* 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 &bmain ne and &numvars = 2
%then %do;
  %let nobici = 0;
  %let x = 1;
  %let bigrp = %scan(&bicomp, &x, '!
  %end;
%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 &&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 $&x length bici $20;
  /* only want number or 'OALL' portion */
  &grpvar.n = scan(scan(inputdsn, 2), 2, '_');
  /* need to convert values to percentages */
  bici = cats(':(', cats(' ', 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 $&x length 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);
  %end;
pvalue = put(prob, 8.3);
end;
else %do;
tvalue = put(nValue1, 8.2);
pvalue = put(cValue1, 8.3);
end;
run;
%mend runfreq;