ABSTRACT

In beginning the analysis of a large data set for research, it is often of interest to screen large numbers of variables for their statistical association with a dichotomous outcome (or treatment) variable. But who wants to sift through pages and pages of output from the FREQ procedure? Or risk the transcription errors inherent in manually transferring the data to a more concise and readable format? This paper will present a set of useful and flexible SAS macros to test multiple variables for their bivariate or one-way association with a binary variable of interest and to succinctly report the results. The output data set from PROC FREQ along with the TRANSPOSE procedure and DATA step processing are used for the statistical analyses, and a DATA _NULL_ step is used for producing the summary report. The final table includes user-selected measures of association (Relative Risks or Odds Ratios with confidence intervals) as well as the chi-square statistic and its associated p-value.

INTRODUCTION

In epidemiologic or clinical research we frequently have one dichotomous variable in which we are most interested. For a clinical trial it would be treatment (e.g. intervention vs. usual care or new drug vs. placebo); for an observational study, it is usually the occurrence of some outcome of interest (e.g. death or incidence of cancer). In contrast there are often many variables that we want to screen for their association with our key variable, either as potential risk factors for the outcome of interest or as confounding variables that would need to be accounted for in later, multivariable analyses. Often this boils down to the analysis of many two by two tables, each set up as in Figure 1.

From this point forward, our key variable of interest will be referred to as the outcome, and the candidate variables we wish to screen will be called risk factors or exposures. The outcome and risk factors are also sometimes referred to as the column and row variables respectively, corresponding to their position in the table above as well as in the output produced by this program. The following measures of association and/or statistical tests are generally of interest from each of these tables:

1) The number and proportion with and without the risk factor who develop the outcome;
2) The Pearson chi-square (1 degree of freedom) and its associated p-value;
3) The Relative Risk (RR) and its (1-α)% confidence interval;
4) The Odds Ratio (OR) and its (1-α)% confidence interval.

Now, you can readily get all of these measures out of the FREQ procedure in SAS by using the CHISQ and MEASURES options; however, it usually involves two to three pages of output per variable. If you have scores of variables, this is quite unwieldy. Of course, you can turn it over to a typist (if you are lucky enough to have one) to produce a concise table, but you risk transcription errors; and every time the data changes, the whole table has to be retyped and rechecked. I had to do this enough times that I decided to put some time into getting SAS to write my tables for me.

The rest of this paper is devoted to a description of a set of SAS macros that will produce a table such as in Figure 2. This one-page table would have required 49 pages of output if the tables produced by PROC FREQ were used to obtain the same measures. And at the rate I type, it would have taken several hours to produce a presentable table.

Basic Strategy

The program consists of three basic parts -- initialization, processing and output -- and each of these corresponds to a SAS macro. In order to produce our summary table, we want to build a SAS data set containing the information needed to output a report such as in Figure 2. Each observation in this data set will contain the statistics and measures for the association of one risk factor with our outcome of interest. This will then correspond to one row of the output table. The basic steps, and the names of the corresponding macros are as follows:

I. Initialization – The %SETUP macro sets the values of the global parameters, checks for required parameters, and obtains labels for all the analysis variables.
## Figure 2. Sample output

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Present</th>
<th>Absent</th>
<th>Relative Risk (95%CI)</th>
<th>Chi-Square (P-value)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AGE &gt; 80 yrs</strong></td>
<td>94/496 (19.0)</td>
<td>39/607 (6.4)</td>
<td>2.9 (2.1, 4.2)</td>
<td>40.39 (0.000)</td>
</tr>
<tr>
<td><strong>FEMALE GENDER</strong></td>
<td>100/804 (12.4)</td>
<td>33/299 (11.0)</td>
<td>1.1 (0.8, 1.6)</td>
<td>0.40 (0.525)</td>
</tr>
<tr>
<td><strong>NON-WHITE RACE</strong></td>
<td>14/175 (8.0)</td>
<td>119/926 (12.9)</td>
<td>0.6 (0.4, 1.1)</td>
<td>3.26 (0.071)</td>
</tr>
<tr>
<td><strong>&lt;12 YEARS OF SCHOOL</strong></td>
<td>114/914 (12.5)</td>
<td>17/181 (9.4)</td>
<td>1.3 (0.8, 2.2)</td>
<td>1.36 (0.243)</td>
</tr>
<tr>
<td><strong>IN SENIOR HOUSING</strong></td>
<td>106/744 (14.2)</td>
<td>27/359 (7.5)</td>
<td>1.9 (1.3, 2.8)</td>
<td>10.33 (0.001)</td>
</tr>
<tr>
<td><strong>CURRENTLY MARRIED</strong></td>
<td>23/256 (9.0)</td>
<td>110/845 (13.0)</td>
<td>0.7 (0.5, 1.1)</td>
<td>3.01 (0.083)</td>
</tr>
<tr>
<td><strong>LIVES ALONE</strong></td>
<td>102/759 (13.4)</td>
<td>31/344 (9.0)</td>
<td>1.5 (1.0, 2.2)</td>
<td>4.38 (0.036)</td>
</tr>
<tr>
<td><strong>&gt;1 CHRONIC CONDITION</strong></td>
<td>55/436 (12.6)</td>
<td>78/655 (11.9)</td>
<td>1.1 (0.8, 1.5)</td>
<td>0.12 (0.727)</td>
</tr>
<tr>
<td><strong>&gt;3 MEDICATIONS</strong></td>
<td>74/525 (14.1)</td>
<td>59/578 (10.2)</td>
<td>1.4 (1.0, 1.9)</td>
<td>3.92 (0.048)</td>
</tr>
<tr>
<td><strong>INCONTINENT &gt;1X/WEEK</strong></td>
<td>31/155 (20.0)</td>
<td>100/926 (10.8)</td>
<td>1.9 (1.3, 2.7)</td>
<td>10.55 (0.001)</td>
</tr>
<tr>
<td><strong>FOLSTEIN MMSE &lt;24</strong></td>
<td>70/348 (20.1)</td>
<td>59/726 (8.1)</td>
<td>2.5 (1.8, 3.4)</td>
<td>31.99 (0.000)</td>
</tr>
<tr>
<td><strong>SPIELBERGER STAI &gt;31</strong></td>
<td>65/478 (13.6)</td>
<td>43/496 (8.7)</td>
<td>1.6 (1.1, 2.3)</td>
<td>6.00 (0.014)</td>
</tr>
<tr>
<td><strong>NO INSTRUMENTAL SUPPORT</strong></td>
<td>14/78 (17.9)</td>
<td>111/977 (11.4)</td>
<td>1.6 (1.0, 2.6)</td>
<td>3.00 (0.083)</td>
</tr>
<tr>
<td><strong>NO EMOTIONAL SUPPORT</strong></td>
<td>18/96 (18.8)</td>
<td>106/950 (11.2)</td>
<td>1.7 (1.1, 2.6)</td>
<td>4.81 (0.028)</td>
</tr>
<tr>
<td><strong>LOW BMI</strong></td>
<td>55/347 (15.9)</td>
<td>64/688 (9.3)</td>
<td>1.7 (1.2, 2.4)</td>
<td>9.72 (0.002)</td>
</tr>
<tr>
<td><strong>IMPAIRED VISION</strong></td>
<td>79/439 (18.0)</td>
<td>48/636 (7.5)</td>
<td>2.4 (1.7, 3.3)</td>
<td>27.21 (0.000)</td>
</tr>
<tr>
<td><strong>IMPAIRED HEARING</strong></td>
<td>47/253 (18.6)</td>
<td>84/829 (10.1)</td>
<td>1.8 (1.3, 2.5)</td>
<td>12.99 (0.000)</td>
</tr>
<tr>
<td><strong>POOR GAIT/Balance</strong></td>
<td>98/537 (18.2)</td>
<td>32/537 (6.0)</td>
<td>3.1 (2.1, 4.5)</td>
<td>38.12 (0.000)</td>
</tr>
<tr>
<td><strong>ADL IMPAIRED</strong></td>
<td>38/141 (27.0)</td>
<td>95/945 (10.1)</td>
<td>2.7 (1.9, 3.7)</td>
<td>32.60 (0.000)</td>
</tr>
<tr>
<td><strong>IADL IMPAIRED</strong></td>
<td>99/657 (15.1)</td>
<td>33/434 (7.6)</td>
<td>2.0 (1.4, 2.9)</td>
<td>13.69 (0.000)</td>
</tr>
<tr>
<td><strong>INJURIOUS FALL</strong></td>
<td>20/33 (60.6)</td>
<td>79/973 (8.1)</td>
<td>7.5 (5.3, 10.6)</td>
<td>99.10 (0.000)</td>
</tr>
</tbody>
</table>

N=Number, CI=Confidence Interval

Input ds: tempds -- 08:59 Sunday, 20JUL97

PAGE 1
II. Processing – This part of the program is iterative. For each risk factor, the %SUMM macro
1) Determines the values of the cell counts from the two by two table (i.e. a, b, c, and d)
2) Calculates the desired measures (proportions, chi-square, RR, etc.)
3) Adds this variable and its associated measures to the accumulating summary data set.

III. Output -- Once all variables have been processed in this way, the %OUT macro utilizes user-supplied information regarding the measures selected, as well as formatting options, to set up the format for the final table and output it. This includes determining the space needed for the output of each column.

Each of these steps will be discussed in some detail below, but first I briefly describe the information the user specifies to the program to produce the desired table, and show a short program which calls each of the three macros.

User-Supplied Options (macro parameters)

Table 1 shows the options specified in the call to the %SETUP macro. Each corresponds to a global macro variable. Also shown are the macro variable names, and the default values, if any.

Table 1. Parameters for the %SETUP macro

<table>
<thead>
<tr>
<th>Macro variable</th>
<th>Meaning</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;DATA</td>
<td>Input SAS data set</td>
<td>Last data set created (&amp;syslast)</td>
</tr>
<tr>
<td>&amp;OUTCOME</td>
<td>Dependent or outcome variable</td>
<td>REQUIRED, numeric</td>
</tr>
<tr>
<td>&amp;OUTYES</td>
<td>Value of outcome variable indicating event</td>
<td>1</td>
</tr>
<tr>
<td>&amp;OUTNO</td>
<td>Value of outcome variable indicating no event</td>
<td>0</td>
</tr>
<tr>
<td>&amp;OUTLABL</td>
<td>Label for outcome variable</td>
<td>Label for outcome variable on the input data set</td>
</tr>
<tr>
<td>&amp;RR*</td>
<td>YES/NO indicating if Relative Risk desired in output table</td>
<td>YES</td>
</tr>
<tr>
<td>&amp;OR*</td>
<td>YES/NO indicating if Odds Ratio desired in output table</td>
<td>NO</td>
</tr>
<tr>
<td>&amp;ALPHA</td>
<td>If RR and/or OR specified, % confidence intervals desired</td>
<td>95</td>
</tr>
<tr>
<td>&amp;CHISQ*</td>
<td>YES/NO indicating if Chi-Square statistic desired in output table</td>
<td>YES</td>
</tr>
</tbody>
</table>

* a maximum of 2 of these measures can be specified for one table.

For each risk factor (or row variable), corresponding to each call of the %SUMM macro, the parameters shown in Table 2 are specified.

Table 2. Parameters for the %SUMM macro

<table>
<thead>
<tr>
<th>Macro variable</th>
<th>Meaning</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;EXPVAR</td>
<td>Exposure or risk factor variable</td>
<td>REQUIRED, numeric</td>
</tr>
<tr>
<td>&amp;EXPYES</td>
<td>Value of risk factor variable indicating presence of characteristic</td>
<td>1</td>
</tr>
<tr>
<td>&amp;EXPNO</td>
<td>Value of risk factor variable indicating absence of characteristic</td>
<td>0</td>
</tr>
<tr>
<td>&amp;EXPXABL</td>
<td>Label for exposure variable</td>
<td>Label for exposure variable on the input data set</td>
</tr>
</tbody>
</table>

After the %SUMM macro has been called for all the desired rows in the table, the %OUT macro is called once to write the output table, and the following parameters are specified:

Table 3. Parameters for the %OUT macro

<table>
<thead>
<tr>
<th>Macro variable</th>
<th>Meaning</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;TITL1, &amp;TITL2, &amp;TITL3</td>
<td>Up to three titles for the final table</td>
<td>No titles</td>
</tr>
<tr>
<td>&amp;COLS</td>
<td>Width of the output table in number of columns</td>
<td>80</td>
</tr>
<tr>
<td>&amp;LINES</td>
<td>Number of lines per page for the output table</td>
<td>65</td>
</tr>
</tbody>
</table>

Sample Program

A simple table with one title and three rows of data, each showing the chi-square and relative risk with 99% confidence intervals, could be produced by the program below. It assumes that the file containing the macro code (RELRSK.INC) resides in the directory ‘c:\maclib’ and that the input data set ANALSYS1 is in a directory that has been given the LIBNAME ‘in’. The outcome variable DEATH is coded 0=NO, 1=YES. Each of the three risk factor variables (HIGHBP, LOWBMI, and HXCHF) are coded 1=YES, 2=NO. The output table will be 95 columns wide, with 60 lines on the page.

```%inc 'c:\maclib\relrsk.inc';
%setup(death,data=in.analysys1,rr=yes,
     chisq=yes, alpha=99);
%summ(highbp,exno=2);
%summ(lowbmi,exno=2);
%summ(hxchf,exno=2);
%out(titl1=Risk Factors for Death after MI, cols=95,lines=60);```
Part I: Initialization

This part of the program does four things.

1) The values of the global parameters are set.
   If one or more of the required parameters is not specified by the user, a message is
   printed to the log, and the program stops executing.

2) Use the DATASETS procedure to delete any existing output data set named FINAL.
   Because the APPEND procedure is used to add observations to the summary data set, it
   is important that at the start of the program no such data set exists.

3) Use the CONTENTS procedure and then the FORMAT procedure with the CNTLIN=
   option to create a SAS format matching the variable names in the input data set to their
   labels. If there is no label, the SAS variable name is used. The following code
   accomplishes this task:

   ```sas
   proc contents data=&ds noprint
      out=tempcont;
   run;
   
   data tempcont
      (rename=(name=start));
   set tempcont (keep=name label);
   
   fmtname = '$varf';
   if label= '' then label=name ;
   run;
   
   proc format cntlin=tempcont;
   run;
   
   %macro ckvar(varname,ds);
   data _null_;
   
   v=put(upcase('&varname'),$ckvarf.);
   call symput('ckvar',left(v));
   run;
   
   %if &ckvar ne Ok %then %do;
   %put;
   %put ***********************;
   %put ** ERROR!! **;
   %put ** VARIABLE &varname **;
   %put ** not on input &ds! **;
   %put ** PROCESSING WILL STOP**;
   %put ***********************;
   %put;
   %let continue=NO;
   %end;
   %mend ckvar ;

   %ckvar(&depvar,&ds);
   %if &continue=NO %then %goto endinit;
   ```

Part II: Processing

This section does the real work of the program. The %SUMM macro, which is called once for each risk factor variable, performs the analysis to obtain the desired measures of association and adds this information to our summary data set. Below is an example with the SAS code and a sample PRINT of the results for each step. In the example, the macro variables &DS, &DEPVAR and &EXPVAR refer to the input SAS data set and the outcome (or dependent) variable and the exposure variables respectively. In the example the outcome variable is LTCFWP (long-term care placement during study follow-up), and the exposure variable is AGE_GE80 (age greater than or equal to 80). The macro variable &EXPN is simply a counter of this exposure variable among all the variables analyzed. Figure 3 is a two by two table for this example, set up as in Figure 1.

The macro call for this example is, simply:

```sas
%SUMM(age_ge80,expno=2) ;
```

**STEP A:** Check that the specified exposure variable is on the input data set. The %CKVAR macro (shown above) is called by the %SUMM macro to ensure that the specified exposure variable exists. If not, a message is written to the log and the program stops executing.

**STEP B:** Get the cell frequencies. A simple PROC FREQ with the NOPRINT option will suffice. The SPARSE option

```
   proc freq data=age_ge80
      noprint;
   run;
   ```

<table>
<thead>
<tr>
<th>AGE_GE80</th>
<th>Yes</th>
<th>No</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>94</td>
<td>402</td>
<td>496</td>
</tr>
<tr>
<td>No</td>
<td>39</td>
<td>568</td>
<td>607</td>
</tr>
<tr>
<td>Total</td>
<td>133</td>
<td>970</td>
<td>1,103</td>
</tr>
</tbody>
</table>

Figure 3. Sample data for one two by two table
is critical in case there is ever a cell with a count of 0. The macro variables &DEPYES and &DEPNO and &EXPYES and &EXPNO refer to the yes and no values of the outcome and exposure variable respectively. The WHERE statement ensures that only the selected values are analyzed. This also allows one to use the same exposure variable more than one time in separate calls to the %SUMM macro but compare different levels in each call.

```
proc freq data = &ds ;
  where &expvar in (&expyes,&expno) and &depar in (&depyes,&depno) ;
tables &expvar*&depar / nprint sparse out=freq&expn ;
run;
```

A test print of the output data set FREQ1 is shown in the top panel of Figure 4. After dropping the PERCENT variable (these are not the proportions we are most interested in), FREQ1 is transposed:

```
proc transpose data=freq&expn
  out=freq&expn_T ;
run;
```

A test print of FREQ1_T is shown in the bottom panel of Figure 4. We now have all the information we need in one observation, but we need to determine which count corresponds to which cell in our two by two table.

**STEP C: Determine which observation corresponds to which cell in the two by two table.** The data set FREQ1 is sorted by the exposure variable and the dependent variable; therefore, in the transposed data set the variables COL1-COL4 are also in that order. The macro %GETCELL, called by the %SUMM macro uses the information provided by the user regarding the values corresponding to each level of the exposure and outcome variables to determine which counts refer to each cell in the two by two table (i.e. a, b, c and d)

```
%MACRO GETCELL;
  %** Determine which column of transposed data corresponds to which cell in the 2x2 table;
  %global col1 col2 col3 col4;
  %if &expyes&expno %then %do;
  %if &depyes&depno %then %do;
    %* CASE 1 ;
    %let col1=d; %let col2=c;
    %let col3=b; %let col4=a;
  %end;
  %else %if &depyes<&depno %then %do;
    %* CASE 2 ;
    %let col1=c;
    %let col2=d;
    %let col3=a;
    %let col4=b;
  %end;
  %end;
  %else %if &expyes<&expno %then %do;
    %if &depyes>&depno %then %do;
      %* CASE 3 ;
      %let col1=b;
      %let col2=a;
      %let col3=d;
      %let col4=c;
    %end;
    %else %if &depye<&depno %then %do;
      %* CASE 4 ;
      %let col1=a;
      %let col2=b;
      %let col3=c;
      %let col4=d;
    %end;
  %end;
%MEND GETCELL ;
```

In our example, in which both variables are coded 1=YES and 2=NO, case 4 applies; in the default case, in which both variables are coded 1=YES and 0=NO, case 1

<table>
<thead>
<tr>
<th>OBS</th>
<th>AGE_GE80</th>
<th>LTCFWP</th>
<th>COUNT</th>
<th>PERCENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>94</td>
<td>8.52</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>402</td>
<td>36.45</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>39</td>
<td>3.54</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>568</td>
<td>51.50</td>
</tr>
</tbody>
</table>

Test Print of output data set from PROC FREQ: (FREQ1)

<table>
<thead>
<tr>
<th>OBS</th>
<th>NAME</th>
<th>LABEL</th>
<th>COL1</th>
<th>COL2</th>
<th>COL3</th>
<th>COL4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AGE_GE80</td>
<td>Age 80+</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>LTCWP</td>
<td>LONG TERM CARE</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>COUNT</td>
<td>Frequency Count</td>
<td>94</td>
<td>402</td>
<td>39</td>
<td>568</td>
</tr>
</tbody>
</table>

Test Print of transposed FREQ1 data set (FREQ1_T)

**FIGURE 4: Sample Output of Test Prints of intermediate steps in the %SUMM macro**
applies. In this way, the coding can vary from one exposure variable to the next. A DATA STEP follows in which the COL1-COL4 variables created by PROC TRANSPOSE are appropriately renamed, and the calculations of the measures of association are performed.

**STEP D: Calculate measures of association.** References for the formulas for the relative risk, odds ratio and their associated standard errors (used to compute the confidence intervals) as well as the chi-square statistic for a two by two table are provided in the appendix. These calculations are very straightforward now that all our counts are in a single observation. Note that we use the variable name _OR for the odds ratio to avoid possible confusion with the SAS reserved variable OR operator. We also save the variable names and their labels for identification of this observation in our summary data set and for later use in the output phase of the program.

```plaintext
data freq&expn._t;
  set freq&expn._t (rename=(col1=&col1_
  col2=&col2 col3=&col3 col4=&col4));
if _name_ = 'COUNT';

%** Assign identifiers to observation;
length expvar depvar $8;
dvartext evartext $40;
expvar = '&expvar';
evartext = &expvar;

%** use the format created above to label each variable;
%if %length(&deplabl)=0 %then $do;
dvartext = put(upcase(depvar),$varf.);
%end;
%else $do;
dvartext = '&deplabl';
%end;
%if %length(&explabl)=0 %then $do;
evartext = put(upcase(expvar),$varf.);
%end;
%else $do;
evartext = '&explabl';
%end;

%** Compute statistics for 2x2 table;
  N = a + b + c + d;
  %** Compute Chi-square and P-value;
  chisqr = N*(((a*d - b*c)**2. / ((a+b)*(c+d)*(a+c)*(b+d)));
  P_value = 1 - probchi(chisqr,1);
  pctdis_y = 100*a/(a+b);
  pctdis_n = 100*c/(c+d));
numyes numno = a+b;
numno = c+d;
%** Determine multiplier for SE depending on desired confidence interval;
%** Set to 95% if missing -- if already <1 (e.g. .95) multiply by 100;
length alpha 4;
%if &alpha= 9 then %let alpha=95;
alpha = &alpha;
if 0 < alpha < 1 then alpha = alpha/100;
p = (1 - (alpha/100)) / 2;
multip = -(probit(p));

%** Compute Relative Risk and CI;
  RR = (a/(a+b))/((c/(c+d)));
  var_lnRR = (b/(a*(a+b))) + (d/(c*(c+d)));
  SE_lnRR = sqrt(var_lnRR);
  lnRR_u = log(RR) + multip*(SE_lnRR);
  lnRR_1 = log(RR) - multip*(SE_lnRR);
  RR_u = exp(lnRR_u);
  RR_1 = exp(lnRR_1);

%** Compute Odds Ratio and CI;
  _OR = (a*d) / (b*c);
  var_lnOR = 1/a + 1/b + 1/c + 1/d;
  SE_lnOR = sqrt(var_lnOR);
  lnOR_u = log(_OR) + multip*(SE_lnOR);
  lnOR_1 = log(_OR) - multip*(SE_lnOR);
  OR_u = exp(lnOR_u);
  OR_1 = exp(lnOR_1);

keep expvar depvar evartext dvartext a b c d pctdis_y pctdis_n
numyes numno RR rr_1 rr_u _OR or_1 or_u N chisqr p_value;
run;

**STEP E: Add this variable and the associated measures to the accumulating summary data set.** The APPEND procedure is more efficient than a SET statement would be.

%* Add new obs to final ds for output;
proc append base=final data=summ&expn._t;
run;

Part III: Output

The final portion of the program uses a DATA _NULL_ step to produce the output table. The trickiest part of this was setting up the table — getting the correct width of each column. Column one will hold the risk factor labels. Column two will be the number and percent having the risk factor who have the outcome. Column three is the number and percent NOT having the risk factor who have the outcome. Columns four and five will hold the measures of association requested. Each table will have at least columns one to three. Depending on the measures requested, there may be up to two more columns. The table is laid out so that the number of columns required takes up the full width specified by the user.
STEP A: Determine the space required for the first three columns. A DATA _NULL_ step in the macro %GETMAX determines the maximum space required for each of these columns, and creates macro variables &MAXLEN, &MAXNUM, &MAXDEN for the maximum exposure variable label length and maximum number of numerator and denominator digits respectively.

```
%MACRO GETMAX;
%*determine maximum length of labels for exposure variables and maximum cell counts;
%global maxlen maxnum maxden;
data _null_;
    set final (keep = expvar evartext a c numerys numno) end=lastobs;
    if _N_ = 1 then do;
        maxlen=length(evartext);
        maxnum=max(maxnum,a,c);
        maxden=max(maxden,numno);
    end;
    retain maxlen maxnum maxden;
    maxlen = max(maxlen,length(evartext));
    maxnum = max(maxnum,a,c);
    maxden = max(maxden,numno,numno);
    if lastobs then do;
        %** must be at least as long as word 'Characteristic' in Col 1 header;
    maxlen = max(maxlen,14);
    call symput('maxlen',
        left(put(maxlen,4.4)));
    call symput('maxnum',
        trim(left(length(left
        (put(maxnum,8.))))));
    call symput('maxden',
        trim(left(length(left
        (put(maxden,8.))))));
    end;
    run;
%MEND GETMAX;
```

STEP B: Count number of measures of association requested and define column headings for selected measures. There is nothing tricky about this code. It is simply a series of macro %IF-%THEN-%ELSE statements that look something like the following. &MA1, representing the first measure of association, and the associated headings (&MATXT1A and &MATXT1B) will go in column 4, and &MA2, along with &MATXT2A and &MATXT2B will go in column 5.

```
%global nm ma1 ma2 matxt1a matxt1b matxt2a matxt2b;
%* initialize macro variables;
%let nm=0; %let ma1=; %let ma2=;
%let matxt1a=; %let matxt2a=;
%let matxt1b=; %let matxt2b=;
%if %upcase(&_RR)=YES %then %do;
    %let ma1=RR;
%end;
%let matxt1a=Relative Risk;
%let matxt1b=& alpha%nrstr(%CI);
%if %upcase(&_OR)=YES %then %do;
    %let nm=2;
    %let ma2=&_OR;
    %let matxt2a=Odds Ratio;
    %let matxt2b=& alpha%nrstr(%CI);
%end;
%else if %upcase(%_CHI2S)=YES %then %do;
    %let nm=2;
    %let ma2=CHI2S;
    %let matxt2a=Chi-Square;
    %let matxt2b=P-value;
%end;
%else %let nm=1;
%end;

... more lines in the same vein ...
```

STEP C: Determine overall column set-up. Now that we know how many columns we will write and what is going in each, we can allot the space across the page accordingly. I simply add up the space required for each column, allowing (as a minimum) two spaces between each column. I then compare the total required to the width specified by the user. The following is within another DATA _NULL_ step. The variables WCOL1-WCOL5 are the widths required for each column (WCOL4 and/or WCOL5 may be 0). If the total required exceeds the line size specified by the user, a message is put to the log stating this and how many more spaces are needed, and the labels in column 1 are truncated as much as necessary. If, on the other hand the total required is less than specified, the extra space is allotted evenly as extra spaces between the columns. Finally, macro variables are created to pass the width information to the final DATA _NULL_ step, which writes the output table.

```
%** start with space between columns of 2;
colspc = 2;
totw = sum(of wcol1-wcol5) +
colspc*(totcols - 1);
if totw > &cols then do;
    over = totw - &cols;
    put /
        **** .............................. ***
        *** !! Too much data for !! ***
        *** !! specified line size !! ***
        *** !! Will truncate labels !! ***
        *** Over &cols by over 2. ***
        *** You may want to run again ***
        *** with larger COLS parameter ***
        *** or shorten some labels ***
        **** .............................. *** /;
wcol1 = wcol1 - (over + 1);
totw = sum(of wcol1-wcol5) +
colspc*(totcols - 1);
end;
else if &cols > totw then do;
    extra = &cols - totw;
```
STEP D: Write the table! I include most of the code here, but break it down into smaller steps:

1) Start the DATA step, specifying FILE PRINT to write the output to the print file. Because we are now processing our summary data set FINAL, the following processing is carried out for each observation in this data set. The variables ENDPAGE are indicators of end-of-page and the page number within this table respectively. UND is simply a text variable that will write an underline across the page.

```sas
data _null_;  
set final end=lastobs;  
retain endpage 1 pagenum 1;  
file print notitles n=ps p= &lines l= &cols  
line= linenum;  
totcols=&totcols;

%* Underlines;  
und = repeat('-', &cols-1);  
u11 = repeat('_', &wcol1);  
u12 = repeat('_', &wcol2);  
u13 = repeat('_', &wcol3);  
u14 = repeat('_', &wcol4);  
u15 = repeat('_', &wcol5);  
```

2) Create SAS variables for where each column starts:

```sas
%* Set-up where each column starts;  
col1 = 1;  
col2 = col1 + &wcol1 + &colspce;  
col3 = col2 + &wcol2 + &colspce;  
if totcols ge 4 then col4 = col3 + &wcol3 + &colspce;  
if totcols ge 5 then col5 = col4 + &wcol4 + &colspce;  
```

3) Create SAS character variables (TEXT4 and TEXT5) for the data to be written in columns 4 and 5.

```sas
length text4 text5 $16;  
format rr_or rr_l rr_u or_l or_u 4.1  
chisqr 6.2 p_value 5.3 a &maxnum..  
umyes numno &maxden..;  
%if &ma=RR %then %do;  
  text4 = put(rr,4.1)||  
    '(||put(rr_1,4.1)||,'||  
    put(rr_u,4.1)||);  
%end;  
%else %if &ma=CHISQR %then %do;  
  text4 = put(chisqr,6.2)||  
    '(||put(p_value,5.3)||)';  
%end;  
%else if &ma= OR %then %do;  
  text4 = put(_OR,4.1)||  
    '(||put(_OR,4.1)||,'||  
    put(_OR_u,4.1)||)'  
%end;
%if &ma2=CHISQR %then %do;  
  text5 = put(chisqr,6.2)||  
    '(||put(p_value,5.3)||)';  
%end;  
%else if &ma2= OR %then %do;  
  text5 = put(_OR,4.1)||  
    '(||put(_OR,4.1)||,'||  
    put(_OR_u,4.1)||)'  
%end;  
```

4) Write the header if the ENDPAGE condition is true. This will be the case for the first observation, and each time the risk factor rows have filled the page (depending on &LINES parameter). This portion includes determining the number of titles and where to place them.

```sas
if endpage then link head;  
HEAD:  
  endpage = 0;  
%* count titles and get length;  
t1 = '&tit1';  
t2 = '&tit2';  
t3 = '&tit3';  
array tt{3} $ t1-t3;  
array tl{3} tl1-tlen3;  
array tc{3} center1-center3;  
ntitle=0;  
do i = 1 to 3;  
  tl{i} = length(tt{i});  
  if tl{i} > 0 then ntitle = ntitle + 1;  
  %* truncate title if > page width;  
  if tl{i} > &cols then  
    tt{i} = substr(tt{i},1,&cols);  
    tc{i} = int((&cols - tl{i})/2);  
%* write the titles;  
  put #1 / @;  
  if ntitle ge 0 then do;  
    do i = 1 to ntitle;  
```
CONCLUSION

I have found these macros to be extremely useful as have other people in our programming group at Yale. I have tried to make them flexible enough to suit the needs that we usually have for this type of table, and fairly easy to use even for a programmer with little macro experience. My main purpose in writing them was to save myself time in making tables for presentation to the researchers I work with and to avoid having to print the lengthy SAS output that would be required to provide the same information. They do not save processing time, but for the size data sets that we usually work with (typically in the range of hundreds to a few thousand observations), this is not a limiting factor. I hope that others outside our small group might find these macros useful as well, although I recognize they might require some modification for other, related purposes. Even for those who have no need to produce these types of tables, perhaps the basic strategy and some programming techniques will be of use. For example, the strategy of building a data set containing the measures you wish to print out (whether they are regression coefficients with standard errors, sample means, percentiles, or relative risks and confidence intervals) is widely applicable since so many SAS procedures allow the creation of output data sets containing many of the quantities of interest.

I would be happy to send this program to anyone who would find it useful. Contact information is below.

APPENDIX:
References for Formulas used in the %SUMM macro

Odds Ratio (and confidence interval):

Relative Risk (and confidence interval) and Chi-Square for a 2x2 table:

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries. @indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

Christianna Williams
Yale University Program on Aging
129 York Street, Suite 1N
New Haven, CT 06511
Voice: (203) 764-9827
e-mail: Christianna.Williams@yale.edu