Paper AD10

Generating Customized Analytical Reports from SAS Procedure Output

Brinda Bhaskar, RTI International, Rockville, MD
Kennan Murray, RTI International, Rockville, MD

ABSTRACT

SAS® has many powerful features, including ODS, macro facilities, the REPORT and TRANSPOSE procedures, and variable information functions that can be used to enhance reports displaying results from statistical analyses. It is often necessary to create tables providing summary statistics, including frequencies, percents, means, standard deviations, and p-values, for populations or comparison groups. We have developed macro programs that calculate these statistics and output results directly into customized tables in MS Word. The macros use data step processing and the FREQ, TTEST, and TRANSPOSE procedures to obtain summary statistics. Variable labels and categories are obtained directly from the data using variable information functions. Tables are generated using ODS that concisely summarize key information directly from SAS output, which saves time and avoids errors associated with manual table creation. These macros are particularly useful when statistics must be produced for a large number of variables and when analyses must be repeated.

INTRODUCTION

In practice, statisticians are often required to present large amounts of complex data to investigators in a concise, simple, and user-friendly format. There are several ways of accomplishing this, but one of the most commonly used methods of data presentation is tables. Tables, if well designed, can communicate results of statistical findings simply and effectively. Creating tables manually can be a laborious, time-consuming, and error-prone process. However, there are several functionalities and features of SAS® that can be utilized to automatically generate highly customized tables in Microsoft Word from statistical output. We have created two macros that output the results of bivariate analyses for both categorical and continuous data from the SAS FREQ and TTEST procedures directly into customized Microsoft Word tables. These macros utilize various SAS features, including simple procedures such as TRANSPOSE to obtain required statistics in the appropriate table-specific format; SAS variable information functions to automatically obtain variable labels and categories; SAS ODS tables to extract the required descriptive statistics and information into output data sets; and the REPORT procedure to finally generate the tables in Microsoft Word. They are simple, easy to use, and can be manipulated with ease to handle output from other SAS procedures and to create tables of differing specifications. These macros are particularly useful when a large number of variables are included in the analysis and therefore many statistical comparisons must be made. In addition, they may be helpful for SAS users who are just beginning to write macro code.

DATA SET

We use the data set called WORK.HEART to illustrate our macro code in this paper. It has 200 observations and contains the following variables:

VARIABLE NAME	VARIABLE DESCRIPTION	VARIABLE TYPE
Age	Age in years	Continuous
CHD	Coronary Heart Disease, Yes / No	Categorical
Gender	Gender (Male / Female)	Categorical
Age Group	Age in years (<40 years, 40+ years)	Categorical
Smoking Status	Smoking status (Current smoker, former smoker, never smoked)	Categorical
Weight	Weight in pounds	Continuous
LDL	LDL cholesterol	Continuous
HDL	HDL cholesterol	Continuous

There are no missing values for any variable. Each categorical variable is numeric and has a format permanently associated with it. The outcome variable is coronary heart disease (CHD), a dichotomous variable.

SAS CODE TO CREATE TABLES FOR CATEGORICAL PREDICTOR VARIABLES

In practice, we often use PROC FREQ to examine bivariate associations between an outcome variable and a set of predictor variables, all of which are categorical in nature. In this example, we examine the associations between CHD and the categorical predictor variables gender, smoking status, and age group. We wish to create a table which presents the total number of individuals in each level of the predictor variable, the prevalence of coronary heart disease in each level of the predictor variable, and the associated p-value. Thus, we must create a table that looks like this:

Demographic Characteristic	Category	Total	Coronary Heart Disease % (n)	p-value (2-sided)
Variable	1			
	2			

All of this information is readily available from our data; we will obtain the description of the demographic characteristic variable and categories from our data using variable information functions. The remainder of the information will be taken from PROC FREQ output. To begin, we create a macro which we call CATEG. It uses two macro variables: PRED, which is the categorical predictor variable; and I, which is a counter. These macro variables will be named when the macro is invoked.

```
%macro categ(pred,i);
```

STEP 1: PROC FREQ

The first step in the macro is to run PROC FREQ and generate the data sets that we will need to create our table. Recall that our outcome variable is CHD.

```
 proc freq data = heart;
   tables &pred * chd / chisq sparse outpct out = outfreq&i ;
   output out = stats&i chisq;
 run;

 proc sort data = outfreq&i;
  by &pred;
 run;
```

The TABLES statement generates a cross tabulation between our predictor variable and CHD. The OUT = OUTFREQ&I and OUTPCT options in the TABLES statement asks SAS to generate a data set with the frequency statistics from this cross tabulation, including the counts and percentages associated with each cell. The SPARSE option ensures that cells with zero counts will be included in this output data set. The OUTPUT statement asks SAS to generate a data set containing the chi-square statistics resulting from the cross tabulation. We could easily request other statistics; by adding the EXACT option, for example, we would also be able to obtain statistics from Fishers' exact tests. These two data sets, OUTFREQ&I and STATS&I, contain all of the information that we need to create our table. OUTFREQ&I contains one record for each cell in the frequency table, while the data set STATS&I contains one record containing all of the chi-square statistics.

STEP 2: OBTAIN THE NUMBER OF INDIVIDUALS IN EACH GROUP

We need to generate the total number of individuals in each level of the predictor variable, and we can obtain this information from the data set OUTFREQ&I. To do so, we run PROC MEANS to sum the variable COUNT, which contains the cell counts, across the levels defined by our predictor variable.

```
proc means data = outfreq&i noprint;
  where chd ne . and &pred ne .;
  by &pred;
  var COUNT;
  output out=moutfreq&i(keep=&pred total rename=(&pred=variable)) sum=total;
  run;
```

This procedure generates a data set, MOUTFREQ&I, which contains one record for each level of our predictor and includes the variable category as well as the number of individuals who fall into that category.

STEP 3: OBTAIN COUNTS AND PERCENTAGES

Recall that the data set OUTFREQ&I contains cell counts and percentages from our initial cross tabulation. We now need to manipulate this data set to display the percentage and number of individuals in each group with coronary heart disease. Therefore, we need to take the cell counts and row percentages, and put them in a format that will be easily understandable. In this example, we display the row percent, followed by the cell count.

```
data routfreq&i(rename = (&pred = variable));
   set outfreq&i;
   length varname $20.;
       if chd = 1 and &pred ne .;
       rcount = put(count,8.);
    rcount = "(" || trim(left(rcount)) || ")";
    pctnum = round(pct_row,0.1) || " " || (rcount);
```

```
      index = &i;
    varname = vlabel(&pred);
    keep &pred pctnum index varname;
 run;
```

In this code, we take only records where CHD = 1, because for this example, we are not interested in looking at the group without CHD. However, we could easily include this group in our output as well, if that was desired. The key point of this code is the creation of the variable PCTNUM, which displays the row percentage corresponding to coronary heart disease, followed by the number of individuals in that group with the disease. This is a character string, and we have chosen to display percentages with format 8.1.

An interesting note is the creation of the variable VARNAME. We know that we want to fill in the column "Demographic Characteristic" in our table with an accurate description of our predictor variable. A convenient way to do this is to use the variable label as a descriptor. Here, we use the VLABEL function to pull the variable label from our predictor variable, and to create a new variable, VARNAME, which contains this value. This text string thus becomes the descriptor for the predictor variable in our table.

STEP 4: OBTAIN P-VALUES

The next item necessary for our table is the p-value. We wish to present precise p-values with significant values indicated. The p-values resulting from chi-square tests are contained in the data set STATS&I, and we can use this data set to create p-values that meet these expectations.

```
data rstats&i;
  set stats&i;
  length p_value $8.;
  if P_PCHI <= 0.05 then do;
     p_value = round(P_PCHI,0.0001) || "*";
     if P_PCHI < 0.0001 then p_value = "<0.0001" || "*";
  end;
  else p_value = put(P_PCHI,8.4);
  keep p_value index;
  index = &i;
 run;
```

This code takes the numeric p-value, converts values less than 0.0001 to the text string "<0.0001" and keeps all other p-values with 4-level precision. Furthermore, p-values less than 0.05 are starred so that they are evident.

STEP 5: OBTAIN THE FORMAT ASSIGNED TO THE PREDICTOR VARIABLE

The last piece of information that we need for our table is the formatted categories of our predictor variable. We know that all of our variables have been assigned permanent formats which adequately describe the variable categories. We can use the VFORMAT function to pull the variable format from the predictor variable. This format name will be called into a macro variable and used later to create our variable categories.

```
data _null_;
  set heart;
  call symput("fmt",vformat(&pred));
run;
```

STEP 6: COMBINE INFORMATION

We have obtained all of the information that we need for our table, but it is contained in separate data sets: MOUTFREQ&I, which contains our totals; ROUTFREQ&I, which contains our variable descriptors, percentages, and cell counts; and RSTATS&I, which contains our p-values. By merging these data sets together, we will have our data exactly as we need it for our table.

```
proc sort data = moutfreq&i;
  by variable;
 run;

 proc sort data = routfreq&i;
  by variable;
 run;

 data temp&i;
  merge moutfreq&i routfreq&i;
  by variable;
 run;
```

```
data final&i;
  merge temp&i rstats&i;
    by index;
  length formats $20.;
  formats=put(variable,&fmt);
  if not first.index then do;
             varname = " ";
             p_value = " ";
             end;
  drop variable;
 run;

%mend;
```

Note that the variable FORMATS is created by using the PUT statement and the format of our predictor variable obtained in the previous step. This creates a final variable which contains a description of our groups. For example, with the variable GENDER, our formats indicate that a value of 1 = Male and 2 = Female. We therefore create a variable that takes the values Male and Female. Thus, we have labeled our categories, thereby eliminating any manual work.

The data set that we create with this code, FINAL&I, contains all of the information that we have gathered to date: the description of the predictor variable, the total number of individuals in each group, the cell counts and row percentages for each group, and the p-value resulting from the comparison of the prevalence of coronary heart disease among the various groups.

STEP 7: CALL IN THE CATEG MACRO FOR EACH PREDICTOR VARIABLE

With our macro complete, we invoke it for the following categorical variables: gender, smoking status, and age group. This creates three data sets: FINAL1, FINAL2, and FINAL3.

```
%categ(gender,1);
%categ(smoke,2);
%categ(age_group,3);
```

STEP 8: COMBINE DATA SETS FOR EACH PREDICTOR VARIABLE

These three data sets can now be combined to create one data set ready to be converted to a Word document. The following macro can be used to combine these data sets:

```
%macro names(j,k,dataname);
  %do i=&j %to &k;
  &dataname&i
  %end;
 %mend names;
```

The NAMES macro is called in below. The user needs to indicate the starting and stopping point of the counter (the I macro variable) and the data set name. This macro is particularly useful when a large number of data sets are combined, as is often the case, since it bypasses the need to type in the names of many data sets.

```
data categ_chd;
  set %names(1,3,final);
  label varname = "Demographic Characteristic"
             total = "Total"
             pctnum = "Coronary Heart Disease * % (n)"
             p_value = "p-value * (2 sided)"
             formats = "Category";
 run;
```

STEP 9 : OUTPUT TABLE TO MICROSOFT WORD

The data set above, CATEG_CHD, will now be converted to a table in Microsoft Word. We do so using ODS and PROC REPORT.

```
ods listing close;
ods rtf file = "c:\nesug\table1a.rtf"  style = forNESUG;
proc report data = categ_chd nowd split = "*";
 column index varname formats total pctnum p_value;
 define index /group noprint;
compute before index;
  line ' ';
```

```
endcomp;
 define varname / order = data style(column) = [just=left] width = 40;
 define formats / order = data style(column) = [just=left];
 define total   / order = data style(column) = [just=center];
 define pctnum  / order = data style(column) = [just=center];
 define p_value / order = data style(column) = [just=center];
  title1 " NESUG PRESENTATION: TABLE 1A (NESUG 2004)";
  title2 " CROSSTABS OF CATEGORICAL VARIABLES WITH CORONARY HEART DISEASE OUTCOME";
run;
ods rtf close;
ods listing;
```

Because we have already manipulated our data to appear exactly as we wish to display it in the table, PROC REPORT allows us to generate a customized table in Microsoft Word. In fact, by using ODS to export the table, no further manipulation in the word processing program is required. Our final table appears below. It is concise, informative, and the numbers are accurate.

Demographic Characteristic	Category	Total	Coronary Heart Disease % (n)	p-value (2-sided)
Gender	Male	102	52.9 (54)	0.7592
	Female	98	55.1 (54)	
Smoking Status	Current Smoker	68	52.9 (36)	0.5064
	Former Smoker	71	59.2 (42)	
	Never Smoked	61	49.2 (30)	
Age Group	< 40 years	46	52.2 (24)	0.7770
	40+ years	154	54.5 (84)	

We note that the macro CATEG can be modified easily to create tables of different specifications. Furthermore, when multiple tables must be created, for example, with several different outcome variables, the macro can be expanded by adding an additional parameter for the outcome variable in the macro invocation. Thus, this macro is extremely useful when a large number of predictor variables or several outcome variables are being examined.

SAS CODE TO CREATE TABLES FOR CONTINUOUS PREDICTOR VARIABLES

In addition to examining relationships between categorical variables, in conducting exploratory analysis we often need to examine bivariate associations between a two level categorical outcome variable and continuous predictor variables. This is most commonly done using the TTEST procedure in SAS. In our example, we compare the means of the predictor variables weight, age, LDL cholesterol and HDL cholesterol across levels of the outcome variable CHD. We wish to present the results of this comparison in a compact and comprehensible table that includes the total number of observations used in the analysis, the mean and standard deviation of the predictor variables for each level of CHD, and the associated p-value. Thus, our goal is to create a table with rows as shown below:

Variable	Total	CHD Mean (Std. Dev)	No CHD Mean (Std. Dev)	p-value (2-sided)
Variable 1				

To do this, we have developed a macro called CONTINUOUS which obtains all of this information from the TTEST procedure and generates a summary table. This macro is very similar to the macro that was used for the categorical predictor variables. It has two parameters that need to be specified at the time of invocation: CPRED, the continuous predictor variable, and I, which is a counter.

```
%macro continuous(cpred,i);
```

STEP1: PROC TTEST

The first step in the macro is to run PROC TTEST and to obtain the output data sets that will be required to generate our table.

```
proc ttest data=heart;
  class chd;
  var &cpred;
  ods output equality = equalvar&i(keep = probf)
```

```
              statistics = stat&i(keep = N mean StdDev)
              ttests = ttest&i(keep = method variances probt);
 run;
```

We use the ODS tables available in SAS to obtain the data sets containing the information required for the table. The STAT&I data set comprises the mean, standard deviation, and number of observations of the predictor variable for each level of CHD. The data set EQUALVAR&I contains the p-value for the equality of variances test, and the data set TTEST&I contains the p-values corresponding to the t-test using the pooled variances (equal variances) and Satterthwaite (unequal variances) methods.

As in the categorical macro, we use PROC MEANS to obtain the total number of observations used in the analysis.

```
proc means sum data = stat&i noprint;
  var N;
  where N>.;
  output out = total_n&i sum = tot;
 run;
```

STEP 2: OBTAIN APPROPRIATE P-VALUE FOR T-TEST

We need to select the appropriate p-value to be displayed in the table, based on the equality of variances test done in the TTEST procedure. If the equality of variances test is satisfied, we select the p-value corresponding to the pooled variances method. If not, we select the p-value corresponding to the Satterthwaite method. We accomplish this using the following code:

```
 data _null_;
  set ttest&i;
  if method='Pooled' then call symput('probt1',probt);
  else if method='Satterthwaite' then call symput('probt2',probt);
 run;

data pvalue;
  set equalvar&i;
  if Probf< = 0.05 then do;
     pvalue = "&probt2";
  end;
  else if Probf>0.05 then pvalue = "&probt1";
  if pvalue< = 0.05 then do;
     p_value = (round(pvalue,0.0001)||'*');
  end;
  else p_value = round(pvalue,0.0001);
  if pvalue<0.0001 then p_value = ('<0.0001'||'*');
  keep p_value;
 run;
```

The p-values corresponding to the pooled variances and the Satterthwaite methods are called into the macro variables ‘probt1’ and ‘probt2’ respectively. We then evaluate the results of the equality of variances test and select the p-value to be displayed in the table. Significant p-values are indicated with a “*”.

STEP 3: OBTAIN THE PREDICTOR DATA SET

With some data manipulation, we will have everything that we need for creating the table. First, we put the means and standard deviations of our predictor variables in a consistent format for presentation.

```
data descript&i;
  merge stat&i pvalue;
  cmean = put(mean,8.2);
  std_dev = put(stddev,8.2);
  meanstd = (trim(left(cmean))) || ' (' || trim(left(Std_Dev)) || ')';
  drop Mean StdDev N cmean;
 run;

 proc transpose data=descript&i out=tdescript&i(drop=_NAME_ col3);
 var meanstd;
 copy p_value;
 run;
```

The data set DESCRIPT&I combines the descriptive statistics for the predictor variable and the p-value for the t-test. The variable MEANSTD, which is a character string concatenating the mean and standard deviation of the predictor variable for

each level of CHD, is also constructed. This data set is transposed to obtain the MEANSTD variable in 2 columns, one for each level of CHD.

Next, we extract the descriptor for the predictor variable from its label using the SYMPUT routine and the variable label information function available in SAS.

```
 data _null_;
  set heart;
  call symput('name',vlabel(&cpred));
 run;
```

We then create a final data set, MEANS&I that contains all the information in the required format to generate the table.

```
 data means&i;
  merge total_n&i tdescript&i;
  length name1 $100.;
  name1 = "&name";
  num = &i;
  drop _TYPE_ _FREQ_;
  label p_value = 'P-value* (2 sided)'
        COL1 = 'CHD* Mean (Std. Dev)'
        COL2 = 'No CHD* Mean (Std. Dev)'
         tot = 'Total'
         name1 = 'Variable';
 run;

%mend continuous;
```

With the following invocations of the CONTINUOUS macro, we create one data set for each predictor variable which has a single observation and comprises the total number of observations used, the mean, standard deviation and the p-value from the t-test.

```
%continuous(weight,1)
%continuous(age,2)
%continuous(LDL,3)
%continuous(HDL,4);
```

STEP 4: MERGE THE DATA SETS AND CREATE THE TABLE

To finish, we simply need to combine the data sets for each of our predictor variables and output the table using PROC REPORT and the ODS feature in SAS, as was done for the categorical table.

```
data continuous_chd;
  set %names(1,4,means);
run;

ods listing close;
ods rtf file = "c:\table1b.rtf"  style = forNESUG ;
proc report data = continuous_chd nowd split = '*';
 column num name1 tot COL1 COL2 p_value;

 define num /group noprint;
  compute before num;
  line ' ';
  endcomp;

 define name1/order = data style(column) = [just = left] width = 40;
 define tot/ order = data style(column) = [just = left];
 define COL1/order = data style(column) = [just = center];
 define COL2/order = data style(column) = [just = center];
 define p_value/order = data style(column) = [just = center];
  title1 color = black " NESUG ANALYSES: TABLE 1B (NESUG 2004)";
  title2 color = black " TTESTS FOR CONTINUOUS VARIABLES: CORONARY HEART DISEASE OUTCOME";
run;
ods rtf close;
ods listing;
```

As described previously, we use the macro NAMES to merge the different data sets corresponding to each predictor variable. We use the REPORT procedure to create the final table, which is presented below: We note that the TEMPLATE procedure can be used to modify fonts, text size, justification, and other table characteristics. The PROC TEMPLATE code used to generate the tables has been provided as an appendix, at the end of the paper.

Variable	Total	CHD Mean (Std. Dev)	No CHD Mean (Std. Dev)	p-value (2-sided)
Weight	200	206.77 (60.11)	200.08 (60.69)	0.4356
Maternal Age	200	45.16 (8.40)	44.87 (8.79)	0.8133
LDL Cholesterol	200	159.38 (34.86)	153.43 (33.52)	0.2226
HDL Cholesterol	200	40.98 (11.53)	39.20 (12.37)	0.2924

CONCLUSION

By using readily available features of SAS such as output and ODS data sets, data step processing, and variable information functions, it is possible to automatically generate customized tables in Microsoft Word from SAS statistical procedures. This paper presented simple macros that can be used to generate tables which summarize basic statistical information from bivariate analyses. All of the information required to create these tables can be extracted and manipulated from SAS procedure output, eliminating the need for manual table creation and saving time as well as eliminating errors. Furthermore, in practice, we often find it necessary to create tables containing large amounts of data, for example, when bivariate analyses are conducted for numerous potential predictor variables. These macros are extremely useful in this situation, where any number of variables can be included in the table. Though processing time is increased when more variables are examined, little additional labor is required.

The possibilities of modifying these macros to present different pieces of statistical information or to create tables with different specifications are endless. For example, one could easily present multiple levels of the outcome variable; instead of presenting row percentages, column percentages could be used; and total columns could be produced. Additionally, output from any number of SAS procedures could be presented. For simplicity, we chose to present results from the FREQ and TTEST procedures; however, we routinely use macros such as these to generate tables using output from other procedures such as LOGISTIC and NPAR1WAY. These macros are flexible enough that they can be used by multiple individuals with minimal changes, yet can be made as complicated and sophisticated as desired.

ACKNOWLEDGEMENTS

CONTACT INFORMATION

Brinda Bhaskar
RTI International
6110 Executive Boulevard, Suite 902
Rockville, MD 20852
bbhaskar@rti.org

Kennan Murray
RTI International
kbeckett@rti.org

APPENDIX

PROC TEMPLATE code used in the program to customize the tables and enhance the output.

```
proc template;
define style forNESUG;
  parent = styles.printer;
  replace fonts /
  'titlefont2' = ("Arial", 9pt, Bold)
  'titlefont' = ("Arial", 10pt, Bold)
  'strongfont' = ("Arial", 8pt, Bold)
  'emphasisfont' = ("Arial", 8pt, Bold)
  'fixedemphasisfont' = ("Arial", 8pt, Bold)
  'fixedstrongfont' = ("Arial", 8pt, Bold)
  'fixedheadingfont' = ("Arial", 8pt, Bold)
  'batchfixedfont' = ("Arial", 8pt, Bold)
  'fixedfont' = ("Arial", 8pt, Bold)
  'headingemphasisfont' = ("Arial", 8pt, Bold)
  'headingfont' = ("Arial", 8pt, Bold)
  'docfont' = ("Arial", 8pt);
end;
run;
```