“Table 1” in Scientific Manuscripts; Using PROC REPORT and the ODS System
Carter Sevick, DoD Center for Deployment Health Research, San Diego, CA

ABSTRACT
The ability to combine the power and flexibility of PROC REPORT and the Output Delivery System (ODS) has revolutionized summary reporting and table creation. However, the majority of resources available approach the subject predominantly from a business perspective. Researchers in public health tend to emphasize categorical data over continuous and have a different aesthetic when it comes to overall look and design creating the need for a subtly different skill set and approach. This paper will take the reader through the process of creating a summary table from raw data and presenting the results in RTF format. Summarized data will include a binary exposure stratified by demographic information, and in addition to frequencies, univariate analyses will also be presented. It should be emphasized that this paper is not a discussion of a study, but rather a discussion of data presentation. It will be assumed that the reader has beginning or intermediate experience with SAS®.

INTRODUCTION
For many data analysts the production of summary tables is a time consuming process that is often accomplished by exporting SAS procedure results to some third party word processing or spreadsheet software where the actual building of the table takes place. This is a manual process that will need to be repeated in its entirety if the audience (such as a client or supervisor) desires a change in the content. Part of the beauty of programming is that code may be reused, while mouse clicks may not, and if the creation of a summary table is done via programming then small tweaking of code maybe all that is necessary to satisfy demands for changes. This paper will take the reader through the process of creating a summary table from raw data and presenting the results in RTF format.

THE DATA
For this tutorial a data set was generated to simulate what an analyst in the field of public health might encounter. The code for this data set is included in the appendix. Below is a partial listing of these data.

<table>
<thead>
<tr>
<th>Obs</th>
<th>gender</th>
<th>region</th>
<th>exposure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Each row describes an individual birth. The three variables composing the data set include: the child’s gender, the region of birth, and a binary exposure variable. The formats that describe the levels of these variables are included in the appendix.

THE GOAL
From these data we would like to produce a type of table typically seen in the beginning of scientific manuscripts. “Table 1”, as we like to call it, provides the audience with summary descriptions of variables to be studied in an analysis. “Table 1’s” vary from study to study mostly depending on variable type, analyses used and personal preferences. But, the overall message of the table is consistently the same. The goal of this paper will be to produce a table that summarizes the counts of total births and exposed births stratified by gender and region. Additionally we would like to display the results of a Chi-square test for association between exposure status and our
two characteristics. Table 1 (below) shows the end result that we will be striving for. This table shall be produced, as an RTF document, completely through programming without cutting and pasting, or other such manual word-processing activities.

Table 1: Characteristics of Subjects

<table>
<thead>
<tr>
<th>Variable</th>
<th>All births (n = 2,000)</th>
<th>Number exposed n (%)</th>
<th>p-value1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender of infant</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Female</td>
<td>1,002</td>
<td>204 (20.4)</td>
<td>0.4809</td>
</tr>
<tr>
<td>Male</td>
<td>998</td>
<td>216 (21.6)</td>
<td></td>
</tr>
<tr>
<td>Region of birth</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mid-Lands</td>
<td>1,090</td>
<td>214 (19.6)</td>
<td>0.0036</td>
</tr>
<tr>
<td>High-Lands</td>
<td>453</td>
<td>81 (17.9)</td>
<td></td>
</tr>
<tr>
<td>Low-Lands</td>
<td>215</td>
<td>58 (27.0)</td>
<td></td>
</tr>
<tr>
<td>Other-lands</td>
<td>143</td>
<td>42 (29.4)</td>
<td></td>
</tr>
<tr>
<td>Unknown</td>
<td>99</td>
<td>25 (25.3)</td>
<td></td>
</tr>
</tbody>
</table>

1 Chi-square test for association

THE PATH

The process that we will follow will come in two steps. In step one, a summary data set will be created that contains all of the information that needs to be displayed. Step two consists of employing PROC REPORT and the Output Delivery System (ODS) to create the structure of the table and export it into an RTF document. The subject matter of step one will be treated in brief with only the more unusual aspects of the program fully explained.

SUMMARIZING THE DATA

How you go about creating a summary data set is your business, but however you do it aim to end up with variables that mirror the columns that you want to display in your table. Upon inspection, Table 1 (above) can be broken down into the following five columns of information:

- Column 1: This column states which variable is being summarized in the current row.
- Column 2: Indicates which level of the variable is being summarized in the current row.
- Column 3: Counts of births which fall into each level of each variable.
- Column 4: A count of exposed births in each level followed by the percentage of births within that level.
- Column 5: The results of a Chi-square test for association between exposure and the variable summarized in that row.

The following program will create a data set containing five variables describing these five columns.

For the first step PROC MEANS will be called upon to create the basic summary to build on.

```
proc means data = DemoData noprint n sum mean;
 class gender region;
 var exposure;
 ways 1;
 output out = PreTable n =
         sum =
         mean = / autoname;
run;
```

The CLASS statement in PROC MEANS is where you place your class variables over which you want to stratify your analysis variable. When more than one variable is specified in the CLASS statement the stratification of the analysis variable will be made over all possible crossings of the class variables. This behavior can be modified using the WAYS statement. This statement tells PROC MEANS how many class variables to combine to stratify your analysis variable. Since we only want one class variable at a time a “1” is specified.
Specifying "n" as a statistic yields the number of non-missing values of "exposure" in each category (this data set has no missing values) thus the variable "exposure_N" tells us how many births occurred in each category. Specifying the "sum" statistic will sum the values of exposure within each category. Since "exposure" is binary (0,1) the result "exposure_sum" variable represents the number of exposed births within each category. As a further result of the binary nature of the exposure variable the mean calculated for each category (exposure_mean) will represent the proportion of infants born exposed.

Figure 2 contains almost all of the information that is needed for our Table 1. All that remains is to apply some minor cosmetic changes and add a column for results of a Chi-square analysis. But first, the cosmetic changes. Observe the following code:

data table (keep = variable levels exposure_N exposure_sum exposure_mean pct ExpPct indexvar);
  set pretable;
  length variable $ 20; *** These four variables ***;
  length levels $ 20; *** will describe the first ***;
  length pct $ 8; *** four columns of the table ***;
  length ExpPct $ 15; ***
  if gender ne . then do; ***Building "variable" and "Levels" columns for "Gender" ***;
    variable = 'gender';
    levels = put(gender, gender.);
    IndexVar = 1;
  end;
  if region ne . then do; ***Building "variable" and "Levels" columns for "region" ***;
    variable = 'region';
    levels = put(region, region.);
    IndexVar = 2;
  end;
  pct = put(exposure_mean*100,4.1); ***Calculate % exposed ***;
  ExpPct = compress(put({exposure_sum, comma4.}, ',' || '|' || '|' || compress(pct, ' ') || ' '|')'); ***creating data in the form of "count (%)" ***;
run;
Figure 3 displays the result of the above code:

![Figure 3. output of data set 'table'](image)

<table>
<thead>
<tr>
<th>Obs</th>
<th>exposure_N</th>
<th>exposure_Sum</th>
<th>exposure_Mean</th>
<th>variable</th>
<th>levels</th>
<th>pct</th>
<th>ExpPct</th>
<th>IndexVar</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1090</td>
<td>214</td>
<td>0.19633</td>
<td>region</td>
<td>Mid-Lands</td>
<td>19.6</td>
<td>214 (19.6)</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>453</td>
<td>81</td>
<td>0.17881</td>
<td>region</td>
<td>High-Lands</td>
<td>17.9</td>
<td>81 (17.9)</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>215</td>
<td>58</td>
<td>0.26977</td>
<td>region</td>
<td>Low-Lands</td>
<td>27.0</td>
<td>58 (27.0)</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>143</td>
<td>42</td>
<td>0.29371</td>
<td>region</td>
<td>Other-Lands</td>
<td>29.4</td>
<td>42 (29.4)</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>99</td>
<td>25</td>
<td>0.25253</td>
<td>region</td>
<td>Unknown</td>
<td>25.3</td>
<td>25 (25.3)</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>1002</td>
<td>204</td>
<td>0.20359</td>
<td>gender</td>
<td>Female</td>
<td>20.4</td>
<td>204 (20.4)</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>998</td>
<td>216</td>
<td>0.21643</td>
<td>gender</td>
<td>Male</td>
<td>21.6</td>
<td>216 (21.6)</td>
<td>1</td>
</tr>
</tbody>
</table>

To get Chi-square results we need to use PROC FREQ and then use the ODS system to obtain a data set that contains the Chi-square results.

```sas
ods trace on;
ods output chisq = ChiData;
proc freq data = DemoData;
   table exposure*gender / chisq;
   table exposure*region / chisq;
run;
ods trace off;
```

The first statement "ods trace on;" will signal SAS to display a list of all available output data sets from following procedures in the log file. The data set containing the Chi-square results is called 'chisq' and the next line down "ods output chisq = ChiData;" tells SAS to output the Chi-square results into a file in the work directory called "ChiData".

Figure 4 displays the Chi-square results.

![Figure 4. Results from PROC FREQ](image)

<table>
<thead>
<tr>
<th>Obs</th>
<th>Table</th>
<th>Statistic</th>
<th>DF</th>
<th>Value</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Table exposure * gender</td>
<td>Chi-Square</td>
<td>1</td>
<td>0.4969</td>
<td>0.4809</td>
</tr>
<tr>
<td>2</td>
<td>Table exposure * gender</td>
<td>Likelihood Ratio Chi-Square</td>
<td>1</td>
<td>0.4969</td>
<td>0.4809</td>
</tr>
<tr>
<td>3</td>
<td>Table exposure * gender</td>
<td>Continuity Adj. Chi-Square</td>
<td>1</td>
<td>0.4225</td>
<td>0.5157</td>
</tr>
<tr>
<td>4</td>
<td>Table exposure * gender</td>
<td>Mantel-Haenszel Chi-Square</td>
<td>1</td>
<td>0.4966</td>
<td>0.4810</td>
</tr>
<tr>
<td>5</td>
<td>Table exposure * gender</td>
<td>Phi Coefficient</td>
<td>_</td>
<td>0.0158</td>
<td>_</td>
</tr>
<tr>
<td>6</td>
<td>Table exposure * gender</td>
<td>Contingency Coefficient</td>
<td>_</td>
<td>0.0158</td>
<td>_</td>
</tr>
<tr>
<td>7</td>
<td>Table exposure * gender</td>
<td>Cramer's V</td>
<td>_</td>
<td>0.0158</td>
<td>_</td>
</tr>
<tr>
<td>8</td>
<td>Table exposure * region</td>
<td>Chi-Square</td>
<td>4</td>
<td>15.6325</td>
<td>0.0036</td>
</tr>
<tr>
<td>9</td>
<td>Table exposure * region</td>
<td>Likelihood Ratio Chi-Square</td>
<td>4</td>
<td>14.9464</td>
<td>0.0048</td>
</tr>
<tr>
<td>10</td>
<td>Table exposure * region</td>
<td>Mantel-Haenszel Chi-Square</td>
<td>1</td>
<td>9.1454</td>
<td>0.0025</td>
</tr>
<tr>
<td>11</td>
<td>Table exposure * region</td>
<td>Phi Coefficient</td>
<td>_</td>
<td>0.0884</td>
<td>_</td>
</tr>
<tr>
<td>12</td>
<td>Table exposure * region</td>
<td>Contingency Coefficient</td>
<td>_</td>
<td>0.0881</td>
<td>_</td>
</tr>
<tr>
<td>13</td>
<td>Table exposure * region</td>
<td>Cramer's V</td>
<td>_</td>
<td>0.0884</td>
<td>_</td>
</tr>
</tbody>
</table>

Now, the data sets in Figures 3 and 4 need to be merged, but a quick inspection will show that there is nothing to merge by. The following code will rectify the situation.
Data ChiData2 (keep = variable prob);
  set ChiData (where = (statistic = 'Chi-Square'));
  length variable $ 20;
  variable = scan(table,-1,' '); /* Returns the last word in a character value from the "table" variable*/
run;

Figure 5. Output of Data Set 'ChiData2'

<table>
<thead>
<tr>
<th>Obs</th>
<th>Prob</th>
<th>variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.4809</td>
<td>gender</td>
</tr>
<tr>
<td>2</td>
<td>0.0036</td>
<td>region</td>
</tr>
</tbody>
</table>

We now have a data set which describes the Chi-square results for the gender vs. exposure and region vs. exposure, as well as a common variable to merge by. The result of a simple match-merge of "table" and "ChiData2", called "TableData" is displayed in Figure 6.

Data TableData;
  merge table (in = a) ChiData2 (in = b);
  by variable;
  if a;
run;

Figure 6. Output of Data Set 'TableData'

<table>
<thead>
<tr>
<th>Obs</th>
<th>exposure_N</th>
<th>exposure_Sum</th>
<th>exposure_Mean</th>
<th>variable</th>
<th>levels</th>
<th>pct</th>
<th>ExpPct</th>
<th>IndexVar</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1002</td>
<td>204</td>
<td>0.20359</td>
<td>gender</td>
<td>Female</td>
<td>20.4</td>
<td>204 (20.4)</td>
<td>1</td>
<td>0.4809</td>
</tr>
<tr>
<td>2</td>
<td>998</td>
<td>216</td>
<td>0.21643</td>
<td>gender</td>
<td>Male</td>
<td>21.6</td>
<td>216 (21.6)</td>
<td>1</td>
<td>0.4809</td>
</tr>
<tr>
<td>3</td>
<td>1090</td>
<td>214</td>
<td>0.19633</td>
<td>region</td>
<td>Mid-Lands</td>
<td>19.6</td>
<td>214 (19.6)</td>
<td>2</td>
<td>0.0036</td>
</tr>
<tr>
<td>4</td>
<td>453</td>
<td>81</td>
<td>0.17881</td>
<td>region</td>
<td>High-Lands</td>
<td>17.9</td>
<td>81 (17.9)</td>
<td>2</td>
<td>0.0036</td>
</tr>
<tr>
<td>5</td>
<td>215</td>
<td>58</td>
<td>0.26977</td>
<td>region</td>
<td>Low-Lands</td>
<td>27.0</td>
<td>58 (27.0)</td>
<td>2</td>
<td>0.0036</td>
</tr>
<tr>
<td>6</td>
<td>143</td>
<td>42</td>
<td>0.29371</td>
<td>region</td>
<td>Other-lands</td>
<td>29.4</td>
<td>42 (29.4)</td>
<td>2</td>
<td>0.0036</td>
</tr>
<tr>
<td>7</td>
<td>99</td>
<td>25</td>
<td>0.25253</td>
<td>region</td>
<td>Unknown</td>
<td>25.3</td>
<td>25 (25.3)</td>
<td>2</td>
<td>0.0036</td>
</tr>
</tbody>
</table>

We now have a data set with the five columns that we originally wanted.

Variable: Identifies which characteristic the observation is describing.
Levels: Identifies the category within each characteristic that the observation is describing.
Exposure_N: Describes the number of births in each category.
ExpPct: Describes the number and proportion of births exposed in each category.
Prob: Contains the Chi-square results.

USING ODS RTF

The Output Delivery System (ODS) is a huge topic. It is a system that allows you to channel procedure output to a variety of formats. The SAS listing window is one of the ODS destinations and is the default. However, it is a simple task to redirect output to other destinations such as PDF, HTML, and RTF among others. Let’s create the most basic RTF output with our summary data set.
ods rtf
   file = 'C:\your complete document path\BasicRTF.rtf'
   bodytitle;

proc print data = TableData;
   title "Figure 7. RTF output of data set 'TableData'";
run;

ods rtf close;

The above code redirects the print procedure output and writes it to an RTF file in the location specified in the FILE option. Ending the program with "ods rtf close;" closes the RTF destination and returns SAS to its previous state. When the destination is closed and the RTF file has been written out, SAS opens a preview window so that the results may be easily seen. If changes are made to the code and you wish to re-run it then the preview window will need to be closed first or the new file will not be written and an error will appear in the log file. The PRINT procedure output will still appear in the listing window. The default action of ODS RTF is to place titles in the header of the document created. The option BODYTITLE is included to prevent this from occurring. When this option is specified the title is placed in the document body, centered and placed just above the procedure output.

### Figure 7. RTF output of data set 'TableData'

<table>
<thead>
<tr>
<th>Obs</th>
<th>exposure_N</th>
<th>exposure_Sum</th>
<th>exposure_Mean</th>
<th>variable</th>
<th>levels</th>
<th>pct</th>
<th>ExpPct</th>
<th>IndexVar</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1002</td>
<td>204</td>
<td>0.20359</td>
<td>gender</td>
<td>Female</td>
<td>20.4</td>
<td>204 (20.4)</td>
<td>1</td>
<td>0.4809</td>
</tr>
<tr>
<td>2</td>
<td>998</td>
<td>216</td>
<td>0.21643</td>
<td>gender</td>
<td>Male</td>
<td>21.6</td>
<td>216 (21.6)</td>
<td>1</td>
<td>0.4809</td>
</tr>
<tr>
<td>3</td>
<td>1090</td>
<td>214</td>
<td>0.19633</td>
<td>region</td>
<td>Mid-Lands</td>
<td>19.6</td>
<td>214 (19.6)</td>
<td>2</td>
<td>0.0036</td>
</tr>
<tr>
<td>4</td>
<td>453</td>
<td>81</td>
<td>0.17881</td>
<td>region</td>
<td>High-Lands</td>
<td>17.9</td>
<td>81 (17.9)</td>
<td>2</td>
<td>0.0036</td>
</tr>
<tr>
<td>5</td>
<td>215</td>
<td>58</td>
<td>0.26977</td>
<td>region</td>
<td>Low-Lands</td>
<td>27.0</td>
<td>58 (27.0)</td>
<td>2</td>
<td>0.0036</td>
</tr>
<tr>
<td>6</td>
<td>143</td>
<td>42</td>
<td>0.29371</td>
<td>region</td>
<td>Other-lands</td>
<td>29.4</td>
<td>42 (29.4)</td>
<td>2</td>
<td>0.0036</td>
</tr>
<tr>
<td>7</td>
<td>99</td>
<td>25</td>
<td>0.25253</td>
<td>region</td>
<td>Unknown</td>
<td>25.3</td>
<td>25 (25.3)</td>
<td>2</td>
<td>0.0036</td>
</tr>
</tbody>
</table>

Figure 7 shows the default formatting for RTF output for PROC PRINT and it looks somewhat the same for all procedures. This default formatting may be altered in several ways, one of which is to write your own formatting definition known as a style definition, using PROC TEMPLATE. Creating your own style definition is not difficult, but it may be more convenient to make the alterations in the procedure code that you wrote to create the output in the first place, especially if your formatting standards change regularly. Consequently, we will not approach PROC TEMPLATE in this paper but review the recommended reading section to find information on this subject. There are three procedures which allow you to affect the resulting format by specifying options directly in the procedure code: PROC PRINT, TABULATE, and REPORT. In the next section we will begin to explore this aspect of the REPORT procedure.

### BRIEF PROC REPORT

PROC REPORT is very powerful and has an incredible amount of functionality built in to it. In fact, much of what was done in the previous section of this tutorial could have been done with PROC REPORT alone. However, PROC REPORT does have something that other procedures do not have and that is a user friendly, robust interface with the Output Delivery System (ODS). The interface that PROC REPORT has with the ODS system allows detailed specification of document formatting within the procedure code itself. It is this capacity which sets PROC REPORT apart from the other procedures that we have used, so instead of re-learning how to create summaries with a new procedure and new syntax, we will focus only on the tools that we need to interact with the ODS system and build a table to present the summary data that we have already created.
BASIC USAGE

In its basic form, PROC REPORT acts and looks very much like PROC PRINT. It is important to realize that while it is a procedure for displaying data, it is also a summarizing procedure. PROC REPORT looks at each data set variable in one of six ways:

- Display – default for character data
- Analysis – default for numeric data
- Across
- Group
- Order
- Computed

Let's start with analysis variables. If an input data set contained only numeric variables then PROC REPORT would look at each of them as analysis variables and collapse all observations into one row. By default the SUM statistic would then be computed for each variable, for example:

```sas
data test;
input a b c d e;
datalines;
4 5 1 2 3
5 1 2 3 4
1 2 3 4 5;
proc report data = test nowd;
run;
```

The results are displayed in Figure 8:

![Figure 8. PROC REPORT and analysis variables](image)

In the PROC REPORT statement line you might have noticed the option NOWD. If this option is not included then SAS will open a report window, separate from the program editor in which a report can be constructed. The inclusion of the option insures that we remain in the program editor and can construct our report with code.

A variable classified as “Display” will cause the report to have one row for every observation. Observe the result when a character variable is added to the above data set.

```sas
data test;
input a1 $ a b c d e;
datalines;
x 4 5 1 2 3
y 5 1 2 3 4
z 1 2 3 4 5;
proc report data = test nowd;
run;
```
Figure 9 displays the output:

![Figure 9. PROC REPORT with analysis and display variables](image)

Grouping variables will cause PROC REPORT to attempt to write one row for every unique value of that variable. The rows of the report will be re-ordered based on the formatted values of the grouping variable (in ascending order) and analysis variables will be consolidated on these rows and a statistic will be calculated to represent the observations within that cell. Several things may prevent the consolidation of the data set, the presence of a display variable for one, but when this occurs the printing of repeated values of the grouping variable will be suppressed. These usage types will be discussed in more detail as we delve further into PROC REPORT. The remaining usage types are beyond the scope of this paper.

Our summary data set “TableData” contains both numeric and character data. Thus, if we were to run PROC REPORT with it as our input data set then we would have RTF output very much like Figure 6 but without the “obs” column.

**BUILDING THE SUMMARY TABLE**

Now we turn our attention back to our summary data set and continue to explore the features of PROC REPORT. There are nine columns of information remaining in the data set, but we only want five of them. The COLUMN statement in PROC REPORT works in a similar fashion to the VAR statement in PROC PRINT although it can accomplish a great deal more.

```plaintext
ods rtf
  file = 'C:\Your complete path here\ColumnStatement.rtf'
  bodytitle;
  proc report data = TableData nowd;
    column variable levels exposure_N ExpPct prob;
    Title 'PROC REPORT with COLUMN statement';
  run;
ods rtf close;
```

Following is the output:

**PROC REPORT with COLUMN statement**

<table>
<thead>
<tr>
<th>variable</th>
<th>levels</th>
<th>exposure_N</th>
<th>ExpPct</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>gender</td>
<td>Female</td>
<td>1002</td>
<td>204 (20.4)</td>
<td>0.4809</td>
</tr>
<tr>
<td>gender</td>
<td>Male</td>
<td>998</td>
<td>216 (21.6)</td>
<td>0.4809</td>
</tr>
<tr>
<td>region</td>
<td>Mid-Lands</td>
<td>1090</td>
<td>214 (19.6)</td>
<td>0.0036</td>
</tr>
<tr>
<td>region</td>
<td>High-Lands</td>
<td>453</td>
<td>81 (17.9)</td>
<td>0.0036</td>
</tr>
<tr>
<td>region</td>
<td>Low-Lands</td>
<td>215</td>
<td>58 (27.0)</td>
<td>0.0036</td>
</tr>
<tr>
<td>region</td>
<td>Other-lands</td>
<td>143</td>
<td>42 (29.4)</td>
<td>0.0036</td>
</tr>
<tr>
<td>region</td>
<td>Unknown</td>
<td>99</td>
<td>25 (25.3)</td>
<td>0.0036</td>
</tr>
</tbody>
</table>
Now we have only the columns that we wanted and in the correct order. Of course, this table is a long way from looking like the table that we want. In order to start making some changes we need to learn how to modify the columns of the report. This is accomplished through the DEFINE statement. The syntax is as follows:

```
define report-item / option(s);
```

Where a report-item is the name of one of the columns in the report.

The options affect the characteristics of the column, and there are a large number of options. In this paper we will discuss: setting formats, specifying a column header, ordering the rows, and STYLE options.

The DEFINE statement is where changes to specific columns are made. In this next example the text in the headers of the columns will be changed and the usage type of the columns “variable” and “prob” will be changed to “group”. After the slash in the define statement enclose the desired header text in quotes and write the name of the usage type desired for that column. Formats are set by specifying FORMAT = format-name. These options may be specified in any order. “$variable.” is a user-defined format and it is included in the appendix. From this point on, the ODS RTF statements will be omitted in order to save some digital trees.

```
proc report data = TableData nowd;
    column variable levels exposure_N ExpPct prob;
    define variable   / "Variable" group format = $variable.;
    define levels     / " " ;
    define exposure_N / "All births/n/(n = 2,000)" format = comma5.;
    define ExpPct     / "Number exposed/n (%)/(n = 420)";
    define prob       / "p-value" group;
    Title "Table 1: Characteristics of Subjects";
run;
```

Following are the results:

```
Table 1: Characteristics of Subjects

<table>
<thead>
<tr>
<th>Variable</th>
<th>All births n (n = 2,000)</th>
<th>Number exposed n (%) (n = 420)</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender of infant</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Female</td>
<td>1,002</td>
<td>204 (20.4)</td>
<td>0.4809</td>
</tr>
<tr>
<td>Male</td>
<td>998</td>
<td>216 (21.6)</td>
<td></td>
</tr>
<tr>
<td>Region of birth</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mid-Lands</td>
<td>1,090</td>
<td>214 (19.6)</td>
<td>0.0036</td>
</tr>
<tr>
<td>High-Lands</td>
<td>453</td>
<td>81 (17.9)</td>
<td></td>
</tr>
<tr>
<td>Low-Lands</td>
<td>215</td>
<td>58 (27.0)</td>
<td></td>
</tr>
<tr>
<td>Other-lands</td>
<td>143</td>
<td>42 (29.4)</td>
<td></td>
</tr>
<tr>
<td>Unknown</td>
<td>99</td>
<td>25 (25.3)</td>
<td></td>
</tr>
</tbody>
</table>
```

If you want to change the ordering of the rows you may specify the ORDER = option in the DEFINE statement (group, order, or across variables). There are four specifications for this option: DATA, FORMATTED, FREQ, and INTERNAL. Specifying ORDER = DATA will preserve the order that exists in the input data set. To see how the other options affect the order of the rows please see the SAS documentation. If you look closely at the text that was defined for the columns “exposure_N” and “ExpPct” you will notice that the code contains some slashes that do not appear in the output. In PROC REPORT a slash in header text is a signal that you want to start a new line and is known as a “split-character”. This behavior can be easily modified if you would like to change to a new character. In the REPORT statement line use the SPLIT = ‘symbol’ option. The symbol that you choose can be anything, just be sure that you do not intend to use it for anything else in your header text.
Example:

```sas
proc report data = DemoData nowd split = '*';
   ...
```

The above code would re-declare the asterisk (*) as the new split character for the current PROC step.

THE STYLE OPTION

In order to accomplish further changes to the document it will be necessary to employ the STYLE option in the DEFINE and PROC statements. Note that the STYLE options specified will only affect the appearance of the non-listing ODS destination(s) that your report is being sent to. The STYLE option has the following general form:

```
Style<((location)> = <style-element-name> <{style-attribute-specification(s)}>
```

Where location refers to the part of the report that you want to affect; style-element-name is the name of a style element of the style definition used in the report; and style-attribute-specification(s) are the style attributes that are to be changed.

Figuring out within which statement to place a STYLE option can be a bit tricky at first. The SAS documentation has an excellent and thorough discussion of this topic but for now, we will only concern ourselves with the REPORT statement and the DEFINE statement.

STYLE ON THE REPORT STATEMENT LINE

First, we will use the STYLE option on the REPORT statement line to make changes to the overall look of the table. If we compare the most recent Table 1 to the example Table 1 in the beginning we will see a number of differences. The default RTF output contains many borders, rules, shading, and the fonts are mainly Times Roman and Courier. The example Table 1 had minimal borders, no shading, the font is Arial with different sizes, and decreased row height. With the exception of the title and footnote fonts, all of these disparities can be resolved in the REPORT statement. Consider the following code:

```sas
proc report data = TableData nowd
   style(report) = {cellpadding = 1.25pt
      cellspacing = 0pt
      frame = hsides
      rules = groups}
   style(header) = {font = ("arial",11pt)
      background = white}
   style(column) = {font = ("arial",11pt)};
   * Other Statements omitted for space considerations ;
run;
```

The first STYLE option is set to target the “report”. From this location we can target a wide variety of general structural aspects of the table. The CELLPADDING style-attribute controls the distance between the border of a cell and its contents. CELLSPACING refers to the distance set between cells. FRAME controls the border which surrounds the table. Setting the FRAME to HSIDES leaves only the two horizontal sides intact. Some, but not all, of the other possible values include: VSIDES leaves only the two vertical sides in place and VOID eliminates the outer border completely. RULES controls the internal borders of the table. Setting RULES to GROUPS places a horizontal border between the data area of the table and the header, and if the table has a footer the value GROUPS will also create a border between the table and the footer. Other possible values for RULES are: ALL, COLS, NONE, and ROWS.

The second STYLE option is targeting the table header. The attribute FONT is used to modify the style of fonts in the specified location, and has the general form:

```
font = {“font name”, font-size, font-style}
```
The BACKGROUND attribute sets the background color of the table. Following is the resulting RTF output:

Table 1: Characteristics of Subjects

<table>
<thead>
<tr>
<th>Variable</th>
<th>All births</th>
<th>Number exposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender of infant</td>
<td>n (n = 2,000)</td>
<td>n (%)</td>
</tr>
<tr>
<td>Female</td>
<td>1,002</td>
<td>204 (20.4)</td>
</tr>
<tr>
<td>Male</td>
<td>998</td>
<td>216 (21.6)</td>
</tr>
<tr>
<td>Region of birth</td>
<td>n (n = 420)</td>
<td>n (%)</td>
</tr>
<tr>
<td>Mid-Lands</td>
<td>1,090</td>
<td>214 (19.6)</td>
</tr>
<tr>
<td>High-Lands</td>
<td>453</td>
<td>81 (17.9)</td>
</tr>
<tr>
<td>Low-Lands</td>
<td>215</td>
<td>58 (27.0)</td>
</tr>
<tr>
<td>Other-lands</td>
<td>143</td>
<td>42 (29.4)</td>
</tr>
<tr>
<td>Unknown</td>
<td>99</td>
<td>25 (25.3)</td>
</tr>
</tbody>
</table>

STYLE IN THE DEFINE STATEMENT

Something noticeable about the latest table is that it looks a bit cramped and a few items are not justified in the correct direction. The column widths can be adjusted by the CELLWIDTH attribute. Cell justification can be modified by using the JUST attribute. CELLWIDTH is specified by a numeric length, it can be specified in inches although points can be used as well.

Style(column)={cellwidth = 2in} would set a column width of two inches in a document.

The JUST attribute can take three values corresponding to the direction that you want your content justified to: LEFT, CENTER or RIGHT.

Style(column)={just = center} would center the column that this style option is targeting.

proc report data = TableData nowd
* Other Statements omitted for space considerations ;

define variable / "Variable" group format = $variable.
style(header) = {just = left}
style(column) = {cellwidth = 1.5in};
define levels / " ".
Style(column) = {cellwidth = 1.5in};
define exposure_N / "All births/n/(n = 2,000)" format = comma5.
style(column) = {just = right cellwidth = 1.25in};
define ExpPct / "Number exposed/n (%)/(n = 420)"
style(column) = {cellwidth = 1.25in just = right} ;
define prob / "p-value" group
style(column) = {just = center};
* Other Statements omitted for space considerations ;
run;

The resulting output is as follows:

Table 1: Characteristics of Subjects

<table>
<thead>
<tr>
<th>Variable</th>
<th>All births</th>
<th>Number exposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender of infant</td>
<td>n (n = 2,000)</td>
<td>n (%)</td>
</tr>
<tr>
<td>Female</td>
<td>1,002</td>
<td>204 (20.4)</td>
</tr>
<tr>
<td>Male</td>
<td>998</td>
<td>216 (21.6)</td>
</tr>
<tr>
<td>Region of birth</td>
<td>n (n = 420)</td>
<td>n (%)</td>
</tr>
<tr>
<td>Mid-Lands</td>
<td>1,090</td>
<td>214 (19.6)</td>
</tr>
<tr>
<td>High-Lands</td>
<td>453</td>
<td>81 (17.9)</td>
</tr>
<tr>
<td>Low-Lands</td>
<td>215</td>
<td>58 (27.0)</td>
</tr>
<tr>
<td>Other-lands</td>
<td>143</td>
<td>42 (29.4)</td>
</tr>
<tr>
<td>Unknown</td>
<td>99</td>
<td>25 (25.3)</td>
</tr>
</tbody>
</table>
Looking at the last table, the “All births” and “Number exposed” columns are nicely right justified but they seem disconnected from their respective headers. The columns could be centered or the headers right justified, but the visual effect of both might not be pleasing. Wouldn’t it be nice if we could shift the data in the columns a little to the left without disturbing the right justification? Well, it turns out that margins can be set within each column using the options LEFTMARGIN and RIGHTMARGIN. In the following code, margins will be set right in these two columns, exactly how big the margins are may require a bit of playing around. These two options also require a distance to be specified from the margin to the corresponding side of the column.

\begin{verbatim}
proc report data = TableData nowd
   * Other Statements omitted for space considerations ;

   define exposure_N / "All births/n/(n = 2,000)" format = comma5.
       style(column) = {just = right cellwidth = 1.25in rightmargin = .4in} ;

   define ExpPct     / "Number exposed/n (%)/(n = 420)"
       style(column) = {cellwidth = 1.25in just = right rightmargin = .25in} ;
   * Other Statements omitted for space considerations ;
run;
\end{verbatim}

The results are as follows:

\begin{table}[h]
\centering
\begin{tabular}{lccc}
\hline
Variable & All births & Number exposed & \multicolumn{1}{c}{p-value} \\
 & (n = 2,000) & (n = 420) & \\
\hline
Gender of infant & & & \\
Female & 1,002 & 204 (20.4) & 0.4809 \\
Male & 998 & 216 (21.6) & \\
Region of birth & & & \\
Mid-Lands & 1,090 & 214 (19.6) & 0.0036 \\
High-Lands & 453 & 81 (17.9) & \\
Low-Lands & 215 & 58 (27.0) & \\
Other-lands & 143 & 42 (29.4) & \\
Unknown & 99 & 25 (25.3) & \\
\hline
\end{tabular}
\caption{Characteristics of Subjects}
\end{table}

Now the data is centered under the header and is still nicely right justified.

\textbf{INLINE FORMATTING}

Not every part of the document can be reached through STYLE options. Titles and footnotes, for example, cannot be affected using the techniques we have discussed thus far. However, there is a way to implant formatting codes wherever text is displayed through inline formatting. The first step is to select a symbol that you will not be using elsewhere in your report. Some popular choices seem to be “~” and “^”. Declare this symbol to be the “ODS ESCAPECHAR” by placing the following code before your ODS RTF code in the program:

\begin{verbatim}
ods escapechar = '~';
\end{verbatim}

If you do not like the tilde, then place your symbol of choice between the quotes. What the ESCAPECHAR does is signal SAS that you will be passing RTF formatting in to the RTF document that you are creating. Note that inline formatting imbedded in a text string will appear in the SAS listing output and will not affect the listing format.

\textbf{SUPERSCRIPETS AND SUBSCRIPTS}

It is obvious to us that the column “p-value” contains the results of a Chi-square test, after all… we calculated them! It may not be obvious to the target audience so you might want to mark that column with a superscript. This is a job for inline formatting. Assume that you wanted a superscripted “a” in your table. All that is necessary to do is to insert “~{super a}” into the line of text wherever you want the superscript to go. This is taking for granted that “~” is what you have declared as your ODS ESCAPECHAR. If you would rather have a subscripted “a”
then use "~(sub a)" instead. For example, the following code inserts a superscripted "1" into the header for the "p-value" column. Additionally, a footnote will be set to explain the superscript:

```
ods escapechar = '~';
* Other Statements omitted for space considerations ;
proc report data = TableData nowd
  * Other Statements omitted for space considerations ;
    define prob / 'p-value~{super 1}' group
      style(column) = {just = center};
  * Other Statements omitted for space considerations ;
    footnote '~{super 1}Chi-square test for association';
run;
```

The results are as follows:

```
Table 1: Characteristics of Subjects

<table>
<thead>
<tr>
<th>Variable</th>
<th>All births n (n = 2,000)</th>
<th>Number exposed n (%) (n = 420)</th>
<th>p-value(^1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender of infant</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Female</td>
<td>1,002</td>
<td>204 (20.4)</td>
<td>0.4809</td>
</tr>
<tr>
<td>Male</td>
<td>998</td>
<td>216 (21.6)</td>
<td></td>
</tr>
<tr>
<td>Region of birth</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mid-Lands</td>
<td>1,090</td>
<td>214 (19.6)</td>
<td>0.0036</td>
</tr>
<tr>
<td>High-Lands</td>
<td>453</td>
<td>81 (17.9)</td>
<td></td>
</tr>
<tr>
<td>Low-Lands</td>
<td>215</td>
<td>58 (27.0)</td>
<td></td>
</tr>
<tr>
<td>Other-lands</td>
<td>143</td>
<td>42 (29.4)</td>
<td></td>
</tr>
<tr>
<td>Unknown</td>
<td>99</td>
<td>25 (25.3)</td>
<td></td>
</tr>
</tbody>
</table>

\(^1\) Chi-square test for association
```

TITLES AND FOOTNOTES

The next task to accomplish is to format the title and footnote of the table. We will use the inline formatting method to insert a STYLE code into each line of text. The syntax is almost the same as the regular STYLE option that we have studied previously with a few exceptions. Assuming that "~" is your ODS ESCAPECHAR then:

```
~S={style-attribute-specification(s)}
```

The "S" takes the place of "style", and is capitalized. In the following code we will set LEFTMARGINS, left justify and modify the fonts for both the title and the footnote.

```
ods escapechar = '"';
  * Other Statements omitted for space considerations ;
  Proc report data = TableData nowd
    * Other Statements omitted for space considerations ;
      /* Inline formatting and title/footnote text on the same Statement line may be written in separate quote groups */
      Title   ~S={leftmargin = 2.25in font = ("arial",11pt) just = left} 'Table 1: Characteristics of Subjects';
      footnote ~S={leftmargin = 2.25in font = ("arial",9pt) just = left}~{super 1}' 'Chi-square test for association';
run;
```
The results are as follows:

Table 1: Characteristics of Subjects

<table>
<thead>
<tr>
<th>Variable</th>
<th>All births (n = 2,000)</th>
<th>Number exposed (n = 420)</th>
<th>p-value(^1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender of infant</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Female</td>
<td>1,002</td>
<td>204 (20.4)</td>
<td>0.4809</td>
</tr>
<tr>
<td>Male</td>
<td>998</td>
<td>216 (21.6)</td>
<td></td>
</tr>
<tr>
<td>Region of birth</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mid-Lands</td>
<td>1,090</td>
<td>214 (19.6)</td>
<td>0.0036</td>
</tr>
<tr>
<td>High-Lands</td>
<td>453</td>
<td>81 (17.9)</td>
<td></td>
</tr>
<tr>
<td>Low-Lands</td>
<td>215</td>
<td>58 (27.0)</td>
<td></td>
</tr>
<tr>
<td>Other-lands</td>
<td>143</td>
<td>42 (29.4)</td>
<td></td>
</tr>
<tr>
<td>Unknown</td>
<td>99</td>
<td>25 (25.3)</td>
<td></td>
</tr>
</tbody>
</table>

\(^1\) Chi-square test for association

SPACING BETWEEN GROUPS

Having a break space between the groupings of variables would enhance readability of the table. There is a specific command to do this if the output destination is listing, but for RTF it will not work. One way to place a break space is to make use of COMPUTE blocks and LINE statements.

The COMPUTE block is a powerful part of PROC REPORT's arsenal, and one of its functions is to allow the targeting of specific parts of the report table for special formatting tasks. The general form of a COMPUTE block is:

```
COMPUTE location <target>;
/// program statements ///</
ENDCOMP;
```

The “location” is a specification of where the COMPUTE block's program statements will be executed relative to the target: AFTER or BEFORE. If “target” is omitted the COMPUTE block will execute just under the header of the table but above the data space when the “location” is BEFORE, and at the end of the table if AFTER is the specified “location”. One possible value that “target” can take is a variable included in the report, but only if the usage of that variable is “group” or “order”.

The LINE statement in PROC REPORT has many of the same features as the PUT statement in the DATA step. Its general form is:

```
LINE specification(s);
```

The LINE statement is only valid when declared inside of a COMPUTE block. The particular specification of interest for this program is the placement of text strings. Text may be placed in the location targeted by the COMPUTE block with the following LINE statement:

```
LINE 'character-string';
```

Examine the next addition to the program and the output that follows:

```sql
proc report data = TableData nowd
   * Other Statements omitted for space considerations ;
   compute before variable;
   line ' ';
   endcomp;
   * Other Statements omitted for space considerations ;
run;
```
Table 1: Characteristics of Subjects

<table>
<thead>
<tr>
<th>Variable</th>
<th>All births (n = 2,000)</th>
<th>Number exposed (n = 420)</th>
<th>p-value¹</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender of infant</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Female</td>
<td>1,002</td>
<td>204 (20.4)</td>
<td>0.4809</td>
</tr>
<tr>
<td>Male</td>
<td>998</td>
<td>216 (21.6)</td>
<td></td>
</tr>
<tr>
<td>Region of birth</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mid-Lands</td>
<td>1,090</td>
<td>214 (19.6)</td>
<td>0.0036</td>
</tr>
<tr>
<td>High-Lands</td>
<td>453</td>
<td>81 (17.9)</td>
<td></td>
</tr>
<tr>
<td>Low-Lands</td>
<td>215</td>
<td>58 (27.0)</td>
<td></td>
</tr>
<tr>
<td>Other-lands</td>
<td>143</td>
<td>42 (29.4)</td>
<td></td>
</tr>
<tr>
<td>Unknown</td>
<td>99</td>
<td>25 (25.3)</td>
<td></td>
</tr>
</tbody>
</table>

¹ Chi-square test for association

In the above example the COMPUTE block is set to execute before every unique value of the “variable” column, and the LINE statement places a blank line of text in that position. The table is now much easier to read.

The COMPUTE block also provides access to many of the tools that are available in the DATA step: for example, new variables can be created for the report if needed. While further discussion of the capabilities of the COMPUTE block are beyond the scope of this paper the reference section at the end provides several excellent resources for additional information.

CONCLUSION

In the course of this paper we have only scratched the surface of PROC REPORT and ODS. After working through the presented material, you should have the skills necessary to write simple, reusable code to create attractive and functional tables to display your data. Many more techniques exist for creating more complex tables and reporting systems. If you have found this information useful and wish to learn more then refer to the recommended reading section at the end of this paper.

ACKNOWLEDGMENTS

The author would like to thank his wife Myrna for her support and patience throughout this process. Also, CDR Margaret A. K. Ryan MC, USN and the members of the Data Team at the DoD Center for Deployment Health Research have been an invaluable source of inspiration, support, and assistance. The members include: Tyler Smith, Besa Smith, Isabel Jacobson, Cynthia Leard, Laura Chu, William Honner, Christian Hansen, Robert Reed, and Linda Wang.

Thank you to the Leadership of WUSS 2006 for making this venue possible.
RECOMMENDED READING


CONTACT INFORMATION
Your comments and questions are valued and encouraged. Contact the author at:
Carter Sevick
NAVAL HEALTH RESEARCH CENTER
DoD Center for Deployment Health Research
PO Box 85122
San Diego CA 92186-5122
Work Phone: (619) 767-4762
Fax: (619) 553-7601
Email: sevick@nhrc.navy.mil

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.
Appendix:

The Complete Program Code:

```sas
/* create simulated data with which to create a report */
data DemoData (drop = i);
    length gender region exposure 8;
    do i = 1 to 2000;
        exposure = ranbin(12345,1,0.20);
        select;
            when(exposure = 1) do;
                gender = ranbin(12345,1,0.53);
                region = rantbl(12345,0.52,0.22,0.12,0.09,0.05);
            end;
            when(exposure = 0) do;
                gender = ranbin(12345,1,0.5);
                region = rantbl(12345,0.55,0.25,0.10,0.06,0.04);
            end;
            otherwise;
        end;
    output;
end;
run;

/* create user-defined formats for this report */
proc format;
    value gender /* format for infant gender variable */
        1 = 'Male'
        0 = 'Female';
    value region /* format for region of birth variable */
        1 = 'Mid-Lands'
        2 = 'High-Lands'
        3 = 'Low-Lands'
        4 = 'Other-lands'
        5 = 'Unknown';
    value $variable /* format for variable names */
        'gender' = 'Gender of infant'
        'region' = 'Region of birth';
run;

***Calculate counts***;
proc means data = DemoData noprint n sum mean;
    class gender region;
    var exposure;
    ways 1;
    output out = PreTable n =
        sum =
        mean = / autoname;
run;
```
***Acting on the dataset generated by PROC MEANS, create
the column information required for the report***:

```sql
data table (keep = variable levels exposure_N exposure_sum
  exposure_mean pct ExpPct indexvar);
set PreTable;
length variable $ 20;
length levels $ 20;
length pct $ 8;
length ExpPct $ 15;
if gender ne . then do;
  variable = 'gender';
  levels = put(gender, gender.);
  IndexVar = 1;
end;
if region ne . then do;
  variable = 'region';
  levels = put(region, region.);
  IndexVar = 2;
end;
pct = put(exposure_mean*100,4.1);        ***Calculate % exposed ***;
ExpPct = compress(put(exposure_sum, comma4.),' ') || ' ' || '(' || compress(pct,' ') || ')'; ***creating data in the form of
"count (%)" ***;
run;
/* Generating chi-square results for exposure crossed with characteristics */
ods trace on;
ods output chisq = ChiData;
proc freq data = DemoData;
  table exposure*gender / chisq;
  table exposure*region / chisq;
run;
ods trace off;
/* creating a variable so that chi-square results may be merged with our frequency
data */
data ChiData2 (keep = variable prob);
set ChiData (where = (statistic = 'Chi-Square'));
length variable $ 20;
variable = scan(table,-1,' ');} /* Returns the last word in a character value
from the "table" variable */
run;
proc sort data = table;
  by variable;
run;
```

18
/* Creating final data set, we are ready to use PROC REPORT */
data TableData;
merge
table (in = a)
ChiData2 (in = b);
by variable;
if a;
run;

proc sort data = TableData;***this step is only necessary if the index variable is
changed***;
by IndexVar;
run;

ods escapechar = '-';
option nodate nonumber orientation = landscape;
ods rtf file = "OutPutPath" bodytitle;

Proc report data = TableData nowd
style(report) = {cellpadding = 1.25pt
cellspacing = 0pt
frame = hsides
rules = groups}
style(header) = {font = ("arial",11pt)
background = white}
style(column) = {font = ("arial",11pt)};
column variable levels exposure_N ExpPct prob;
define variable / "Variable" group format = $variable.
style(header) = {just = left}
style(column) = {cellwidth = 1.5in};
define levels / " 
Style(column) = {cellwidth = 1.5in};
define exposure_N / "All births/n/(n = 2,000)" format = comma5.
style(column) = {just = right
cellwidth = 1.25in
rightmargin = .4in};
define ExpPct / "Number exposed/n (%)/(n = 420)"
style(column) = {cellwidth = 1.25in
just = right
rightmargin = .25in} ;
define prob / "p-value~{super 1}" group
style(column) = {just = center};
compute before variable;
line ' ';
endcomp;

/* Inline formatting and title/footnote text on the same statement
line may be written in separate quote groups */
Title   '"S={leftmargin = 2.25in font = ("arial",11pt) just = left}'
'Table 1: Characteristics of Subjects';
footnote '"S={leftmargin = 2.25in font = ("arial",9pt) just = left}~{super 1}'
'Chi-square test for association';
run;
ods rtf close;
title;
footnote;