PROC TABULATE is the most widely used reporting tool in SAS®, along with PROC REPORT. Any kind of report with the desired statistics can be produced by PROC TABULATE or PROC REPORT. In clinical studies summary table commonly known as Table-1 is produced for summary statistics for various variables of interest. Comparison between different treatment groups can also be reported using P-values from different statistical tests. This is usually done by running different Procedures depending on type of test required and type of variable used and then grouping the output manually in a word processor software which can be a tedious task especially if your dataset changes over time or if it is a periodical report. In here we introduce a macro that is capable of performing all major statistical tests and then combining their output in a single dataset that can be output as a formatted summary report using PROC TABULATE or PROC REPORT.

A summary often called as Table-1 is the very first table presented in published articles and is mainly used to describe a cohort or to examine crude relation or associations between different variables and the variable of interest, mainly the outcome variable. During the process of building such type of tables, continuous changes or refinements are often encountered which result in running the summary statistics again and then populating the formatted output report.

In here we present a macro that is capable of performing all these tasks automatically, it primarily focuses on a two level outcome variable such as comparison between Treatment and non treatment group by exploring different variables. This work is based on two-way macro presented by Lan Hu and Xiaoqiang Wang.

A typical summary table in publication’s looks likes

<table>
<thead>
<tr>
<th></th>
<th>Active (N=31)</th>
<th>Placebo (N=31)</th>
<th>Overall (N=60)</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age (years)</td>
<td>51.4</td>
<td>50.1</td>
<td>50.8</td>
<td>0.2681</td>
</tr>
<tr>
<td>Male</td>
<td>22(71.0%)</td>
<td>16(51.6%)</td>
<td>38(63.3%)</td>
<td></td>
</tr>
<tr>
<td>Female</td>
<td>9(29.0%)</td>
<td>15(48.4%)</td>
<td>22(36.7%)</td>
<td></td>
</tr>
<tr>
<td>Race</td>
<td></td>
<td></td>
<td></td>
<td>0.9270</td>
</tr>
<tr>
<td>White</td>
<td>18(58.1%)</td>
<td>18(58.1%)</td>
<td>36(60.0%)</td>
<td></td>
</tr>
<tr>
<td>Black</td>
<td>10(32.3%)</td>
<td>8(25.8%)</td>
<td>18(30.0%)</td>
<td></td>
</tr>
<tr>
<td>Other*</td>
<td>3(9.7%)</td>
<td>3(10.3%)</td>
<td>6(10.0%)</td>
<td></td>
</tr>
</tbody>
</table>

Table1: Desired Output
In general to produce a similar summary table we need to run different procedures to get P-values and for specific type of test requested and required.
SUMMARY TABLE MACRO

Macro call –
@summary_table(
 dsn=sgf2014_test,
xvars=age.wilcoxon ethnicity.fisher smoke_ever.fisher weight.ttest,
yvar = class,
comp=1,
out_type=1,
path=)

Where –
 Dsn - Dataset name
 Xvars- Variables to summarize followed by a keyword for test after period if P-values are desired.
 Yvar - variable to base summary on its different categories
 comp - 1 if want p-values in output, 0 – just summary
 Out_type – 1 – Output created using PROC Tabulate, 2 – Output created using PROC Report
 Path – Path for output RTF file

WORKING

1. To start a configuration dataset is created that stores the keywords used for various tests called in macro.

```sas
DATA config;
  INFILE DATALINES4 DSD DLM=',';
  INPUT freq_output_option:$8.
               freq_table_option:$10.
               freq_exact_statement:$20. test:$50.;
  DATALINES4;
    AJCHI,CHISQ,,Continuity-adjusted chi-square for 2x2 tables
    CMHCOR,CMH2,,Cochran-Mantel-Haenszel correlation statistic
    CMHGA,CMH,,Cochran-Mantel-Haenszel general association statistic
    CMHRMS,CMH2,,Cochran-Mantel-Haenszel row mean scores (ANOVA) statistic
    CONTGY,CHISQ,,Contigency coefficient
    CRAMV,CHISQ,,Cramer%’s V
    FISHER,FISHER,,Fisher’s exact test
    GAMMA,MEASURES,,gamma
    JT,JT,EXACT JT;,Jonckheer-Terpstra test
```

Table2: Configuration file

2. Input variables are scanned to separate Statistical test keywords from input variables.

```sas
%let i=1;
%let xvar = %scan(&xvars, &i, %str( ));
%do %while(&xvar ne);
  %put PROCESSING variable &xvar;
  %let j=1;
  %let part = %scan(&xvar, &j, .);
  %do %while(&part ne %str());
    %let part&j = &part;
    %let j = %eval(&j+1);
    %let part = %scan(&xvar, &j, .);
  %end;
  %let j = %eval(&j-1);
```
%let itemize=0;
%let keyword=
%let exact=0;
%let transpose_freq_test=0;

%if &j=2 %then %do;
  %let xvar = &part1;
  %let keyword = &part2;
%end;

3. For categorical variables counts and percentages are generated using PROC FREQ and if any statistical test are requested then the keyword is used to declare it.

PROC FREQ DATA=&tmpdsn6 ORDER=DATA NOPRINT;
  TABLE _1 / OUT=&tmpdsn5(KEEP=count percent) %if &missing %then MISSING;;
  WEIGHT _freq_ %if &zeros %then /ZEROS;;
  WHERE _type_=
  .................More code..............
%if %sysfunc(inputc(&keyword, &freq_table_option_fmt)) eq %str() %then %let keyword = &cat_test;
PROC FREQ DATA=&tmpdsn3 ORDER=DATA NOPRINT;
  TABLE %if &transpose_freq_test %then &yvar*&xvar; %else &xvar*&yvar; /
  %sysfunc(inputc(&keyword, &freq_table_option_fmt));
  %if &exact %then %sysfunc(inputc(&keyword, &freq_exact_statement_fmt));
  WEIGHT _freq_
  OUTPUT OUT=&tmpdsn2 &keyword;

4. If it is a continuous variable then PROC MEANS and PROC NPAR1WAY are used for required statistics.

PROC MEANS DATA=&dsn(KEEP=&xvar &yvar %if %sysevalf(%superq(weight) ne, boolean) %then &weight; RENAME=(&yvar=2 &xvar=1)) &means_statistics
  %if &zeros %then COMPLETETYPES; %if &missing %then MISSING; NOPRINT;
  CLASS _2/%if &zeros and &preloadable_Y %then PRELOADFMT; %if &mlf %then MLF;;
  VAR _1;
  %if %sysevalf(%superq(weight) ne, boolean) %then %str(FREQ &weight;);
  OUTPUT OUT=&tmpdsn1(KEEP=2 _type_ &means_statistics)
  %do j=1 %to &N_MEANS_STATISTICS;
    &&orig_means_statistics&j = &&orig_means_statistics&j
  %end;

%if not %sysfunc(inputn(&keyword, &npar1way_option_fmt)) %then %let keyword=&con_test;
PROC NPAR1WAY DATA=&dsn(KEEP=&xvar &yvar) &keyword NOPRINT %if &missing %then MISSING;;
  CLASS &yvar;
  VAR &xvar;
  %if &exact eq e or &exact eq E %then EXACT;;
  %if %sysevalf(%superq(weight) ne, boolean) %then %str(FREQ &weight;);
  OUTPUT OUT=&tmpdsn2 &keyword;
5. Then output from both is combined into a single raw data file which has variable names, counts, percentages, summary statistics for continuous variables, Test statistics and P values.

6. Raw output table is broken down for categorical and continuous variables for proper formatting required for output.

<table>
<thead>
<tr>
<th>For continuous</th>
<th>For categorical</th>
</tr>
</thead>
<tbody>
<tr>
<td>data var_table; set output; if categorical=0; run; data var_table; set var_table; xcat=<em>name</em>; class1=input(round(_stat_1, 0.01), $12.); class2=input(round(_stat_2, 0.01), $12.); total=input(round(<em>stat</em>, 0.01), $12.); run; data var_table; set var_table; dsid=open('var_table'); if varnum(dsid,'pt2_wil')=0 then pt2_wil=.; if varnum(dsid,'p_f')=0 then p_f=.; if pt2_wil ne . then p_v=input(round(pt2_wil, 0.0000001), $21.); if p_f ne . then p_v=input(round(p_f, 0.0000001), $21.); run;</td>
<td>data cat_table; set output; if categorical=1; run; data cat_table; set cat_table; c1=put(COUNT_1, $9.); pl=put(pct_col_1, $9.); class1= trim(c1)</td>
</tr>
</tbody>
</table>

7. Now as in same format and having similar variable names, both of above tables can be appended.

```sql
data var_table ;
set var_table(keep=varname xcat class1 class2 total p_v categorical) ; run;
data cat_table ;
set cat_table(keep=varname xcat class1 class2 total p_v categorical) ; run;
proc append data=var_table base=cat_table;
run;
```
8. After making some other minor changes resulting dataset is output, which looks like

<table>
<thead>
<tr>
<th>varname</th>
<th>categorical</th>
<th>xval</th>
<th>class1</th>
<th>class2</th>
<th>total</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>1</td>
<td>28</td>
<td>328(124)</td>
<td>3(2)</td>
<td>331(124)</td>
<td>8.023E-01</td>
</tr>
<tr>
<td>age</td>
<td>2</td>
<td>45</td>
<td>128(42)</td>
<td>3(2)</td>
<td>131(42)</td>
<td>1.000E+00</td>
</tr>
<tr>
<td>age</td>
<td>3</td>
<td>60</td>
<td>153(52)</td>
<td>3(2)</td>
<td>156(52)</td>
<td>7.760E-02</td>
</tr>
<tr>
<td>study</td>
<td>1</td>
<td>12</td>
<td>12(4)</td>
<td>3(2)</td>
<td>15(5)</td>
<td>0.792E00</td>
</tr>
<tr>
<td>study</td>
<td>2</td>
<td>13</td>
<td>13(4)</td>
<td>3(2)</td>
<td>16(5)</td>
<td>0.692E00</td>
</tr>
<tr>
<td>study</td>
<td>3</td>
<td>14</td>
<td>14(4)</td>
<td>3(2)</td>
<td>17(5)</td>
<td>0.582E00</td>
</tr>
<tr>
<td>study</td>
<td>4</td>
<td>15</td>
<td>15(4)</td>
<td>3(2)</td>
<td>18(5)</td>
<td>0.472E00</td>
</tr>
<tr>
<td>study</td>
<td>5</td>
<td>16</td>
<td>16(4)</td>
<td>3(2)</td>
<td>19(5)</td>
<td>0.362E00</td>
</tr>
</tbody>
</table>

This resulting dataset can be output using PROC TABULATE or PROC REPORT as desired.

**EXAMPLES**

Ex-1: Using combination of continuous and categorical variables just to create a simple summary report without p-values.

Sample Dataset –

```plaintext
Proc format;
value $gender
'0'='Female'
'1'='Male';
value $Race
'1'='European';

data sgf2014;
input subjid gender $ trt race $ age @@;
label gender = "Gender"
race = "Race"
age = "Age (years)";
format race $race. gender $gender.;
datalines;
101 0 1 2 3 7 301 0 1 1 1 7 10 1 1 50 701 1 1 1 60
102 1 2 1 6 5 302 0 1 2 5 5 3 1 2 4 6 602 0 2 2 30 702 0 1 1 28

Here gender and race are categorical variables where age is a continuous variable and trt is treatment group on basis of which our summary table will be created.
Macro call-
%summary_table(
    dsn=sgf2014,
    xvars=age Race Gender ,
    yvar = trt,
    comp=0,
    out_type=1,
    path='C:\Documents and Settings\Lovedeep.Gondara\sgf\output1.rtf'
)

Using PROC TABULATE

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Treat1</th>
<th>Treat2</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Female</td>
<td>16(42%)</td>
<td>12(37%)</td>
<td>28(47%)</td>
</tr>
<tr>
<td>Male</td>
<td>22(58%)</td>
<td>9(43%)</td>
<td>31(53%)</td>
</tr>
<tr>
<td>Race</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>European</td>
<td>19(50%)</td>
<td>16(76%)</td>
<td>35(59%)</td>
</tr>
<tr>
<td>Asian</td>
<td>13(34%)</td>
<td>5(24%)</td>
<td>18(31%)</td>
</tr>
<tr>
<td>American</td>
<td>6(16%)</td>
<td>0(0%)</td>
<td>6(10%)</td>
</tr>
<tr>
<td>age</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>38</td>
<td>21</td>
<td>59</td>
</tr>
<tr>
<td>MEAN</td>
<td>49.76</td>
<td>52.62</td>
<td>50.78</td>
</tr>
<tr>
<td>MEDIAN</td>
<td>46</td>
<td>53</td>
<td>49</td>
</tr>
<tr>
<td>STD</td>
<td>12.95</td>
<td>13.88</td>
<td>13.24</td>
</tr>
</tbody>
</table>

Table3: Result

With P-values

%summary_table(
    dsn=sgf2014,
    xvars=age.wilcoxon Race Gender ,
    yvar = trt,
    comp=1,
    out_type=1,
    path='C:\Documents and Settings\Lovedeep.Gondara\sgf\output3.rtf'
)

Using PROC TABULATE

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Treat1</th>
<th>Treat2</th>
<th>Total</th>
<th>P Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Female</td>
<td>16(42%)</td>
<td>12(37%)</td>
<td>28(47%)</td>
<td>0.2880754</td>
</tr>
<tr>
<td>Male</td>
<td>22(58%)</td>
<td>9(43%)</td>
<td>31(53%)</td>
<td></td>
</tr>
<tr>
<td>Race</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>European</td>
<td>19(50%)</td>
<td>16(76%)</td>
<td>35(59%)</td>
<td></td>
</tr>
<tr>
<td>Asian</td>
<td>13(34%)</td>
<td>5(24%)</td>
<td>18(31%)</td>
<td></td>
</tr>
<tr>
<td>American</td>
<td>6(16%)</td>
<td>0(0%)</td>
<td>6(10%)</td>
<td></td>
</tr>
<tr>
<td>age</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>38</td>
<td>21</td>
<td>59</td>
<td>0.3699349</td>
</tr>
<tr>
<td>MEAN</td>
<td>49.76</td>
<td>52.62</td>
<td>50.78</td>
<td></td>
</tr>
<tr>
<td>MEDIAN</td>
<td>46</td>
<td>53</td>
<td>49</td>
<td></td>
</tr>
<tr>
<td>STD</td>
<td>12.95</td>
<td>13.88</td>
<td>13.24</td>
<td></td>
</tr>
</tbody>
</table>

Using PROC REPORT

<table>
<thead>
<tr>
<th>Variable Category</th>
<th>Treat1</th>
<th>Treat2</th>
<th>Total</th>
<th>P-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender Female</td>
<td>16(42%)</td>
<td>12(37%)</td>
<td>28(47%)</td>
<td>0.2680754</td>
</tr>
<tr>
<td>Male</td>
<td>22(58%)</td>
<td>9(43%)</td>
<td>31(53%)</td>
<td></td>
</tr>
<tr>
<td>Race European</td>
<td>19(50%)</td>
<td>16(76%)</td>
<td>35(59%)</td>
<td></td>
</tr>
<tr>
<td>Asian</td>
<td>13(34%)</td>
<td>5(24%)</td>
<td>18(31%)</td>
<td></td>
</tr>
<tr>
<td>American</td>
<td>6(16%)</td>
<td>0(0%)</td>
<td>6(10%)</td>
<td></td>
</tr>
<tr>
<td>age N</td>
<td>38</td>
<td>21</td>
<td>59</td>
<td>0.3699349</td>
</tr>
<tr>
<td>MEAN</td>
<td>49.76</td>
<td>52.62</td>
<td>50.78</td>
<td></td>
</tr>
<tr>
<td>MEDIAN</td>
<td>46</td>
<td>53</td>
<td>49</td>
<td></td>
</tr>
<tr>
<td>STD</td>
<td>12.95</td>
<td>13.88</td>
<td>13.24</td>
<td></td>
</tr>
</tbody>
</table>
Table 4: results

CONCLUSION
Using this macro users can easily generate almost all required summary statistics for a basic summary table and can request p-values to compare groups for y variable. A nicely formatted output dataset is created which contains all the information and can be used if custom formatted report other than available in macro is required. As use of PROC TABULATE generates an extra column containing counts, manual edition will be required. To overcome this and make macro truly autonomous, user can request output via PROC REPORT.

FUTURE WORK
Future work is required on this macro in order for it to be used for y-variable having more than two levels, further work is also required to make this macro less computational intensive as while calculating exact values it can use large amount of system resources.

REFERENCES
- Lan Hu, Xiaociang Wang, *The Ultimate Macro to Generate Summary Tables.*
- Kathryn Martin, *A Researcher’s Guide to Making Descriptive and Analytic Tables that are Ready to Publish, WUSS 2006*
- Zang Xiong, *%YAMGAST: Yet Another Macro to Generate a Summary Table, PharmaSUG 2008*
- Yinmei Zhou, Lijun Zhang, Michael L. Hancock, *%SummaryTable: A SAS Macro to Produce a Summary Table in Clinical Trial, PharmaSUG 2006*
- Jun Zuo, Carl R. Haske, *Creating Clinical Trial Summary Table Containing P-Values: A Practical Approach Using Standard SAS Macros, SUGI 22*

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

Name: Lovedeep Gondara
Organization: BC Cancer Agency
Address: 686 west broadway
City, State ZIP: Vancouver BC
Email: Lovedeep.gondara@bccancer.bc.ca

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.