PharmaSUG 2019 - Paper BP-175

Create publication-ready variable summary table using SAS® macro

Geliang Gan, Yale Center for Analytical Sciences, New Haven, Connecticut

ABSTRACT

Variable summary table is an important tool of not only getting to know the data but helping data management by picking up outlier data points. Even though SAS equips us with all procedures needed, it is not a fun task to make one. Creating a variable summary table involves numerous calls on certain SAS procedures.

The purpose of this paper is to share a practical way of automatically generating variable summary table with p-values using a carefully designed and easy-to-use SAS macro. Before the macro starts calling any SAS procedure and conducts calculation, a series of foolproof checking steps against data and parameter assignments are run to assure the successful macro execution. After that, the macro utilizes procedures necessary to produce and collect all information needed according to parameter values specified. By default, frequencies and percentages are tallied for categorical variable, Chi-square or Fisher's Exact test is performed for p values. Mean and standard deviation are computed for continuous variable requesting p values with parametric method, T-test or ANOVA is conducted for p values. Median and interquartile range are provided for continuous variable requesting p values with non-parametric method and p values are calculated using Wilcoxon Rank Sum or Kruskal Wallis test. Final reports, a publication-ready summary table, is presented on html format using SAS internal web browser or a Rich Text Format (RTF) file usually pop-opened with Microsoft Word. Macro call, parameter value assignments and tips for special purpose are discussed in detail with examples.

INTRODUCTION

Creating a variable summary table is usually the first step for any analytical project. Variable summary tables not only provides the profile of our data but can be used as a data validation tool identifying singular data values. SAS procedures including means and freq, are usually utilized in creating descriptive statistics such as means, standard deviation, median, range, frequency and percentage for continuous and categorical variable, respectively. Procedure ttest, glm and npar1way are called on calculating p-values using corresponding statistical test. Even though all these procedures are available and it's not difficult to write SAS code calling them to create summary statistics needed. It's not a pleasant job to prepare a variable summary table. After running all relative SAS procedures, the variable summary table is then finished by copying and pasting collected summary statistics into a text processing software, such as Microsoft word, with a designed table shell. This is very time-consuming, labor-intensive and error-prone. Especially, sometimes, a variable summary table is repeatedly requested against the same data with growing number of observation and used as data monitoring tool.

In fact, when a SAS procedure is employed to compute descriptive statistics. The Output Delivery System (ODS) saves all calculated numbers underneath and uses them to exhibit results. Therefore, theoretically, we can ask ODS to provide a copy of the calculated numbers and store them in a SAS data set. All the outputs collected can be combined and reshaped into a format enabling us to create a presentable table. Based on this idea, a SAS macro named Table1Macro is designed to help ease this tiring process. Table1Macro automates the process of manually creating a variable summary table with minimal requirement for parameter specification. However, options are provided to make the table in variety of styles. With the assistance of this macro, statistician can focus on more important data analyzing issues. This macro is specifically useful for those who work on multiple projects simultaneously.

DATA CHECK FEATURES

A whole array of parameter and data checking steps are carried out to make sure the successful execution of macro.

1. Check all parameter assignments, if not valid, re-assign default value to corresponding parameter or stop macro and ask for new parameter assignment

2. Source data set must exist and not be empty
3. All variables listed in variable list parameters, including calist, coplist and cononplist, must be in the source data set
4. Duplicated variable names in any variable list, including calist, coplist and cononplist, are removed
5. Variables listed in continuous variable lists, coplist and cononplist, must be numeric
6. If specified, group variable must be in the data set and is removed from summary variable list, including calist, coplist and cononplist
7. If calist parameter is not empty, check eratio parameter and make sure its value ranges between 0 and 1, this parameter provides the threshold of whether Chi-square or Fisher's Exact test is used to test association between categorical variables
8. If file path is included in the savefilename parameter and operating system is Microsoft Windows, check if the folder specified exists, if not, result file is saved in SAS current folder
9. Duplicated statistic requested in copmain, copsupplement, cononpmain and cononpsupplement parameters are removed
10. If any variable are all missing, no statistic will be calculated and missing information is reported even though caincludemissing, copincludemissing or cononpincludemissing is set as "no"
11. Check if statistic requested in copmain, copsupplement, cononpmain and cononpsupplement is valid. This marco uses proc means to calculate statistic for continuous variables. Only statistics supported by proc means is allowed
12. If more than 1 statistic is specified in copmain or cononpmain, the first one is picked to be reported in final table
13. If the same statistic is specified in copsupplement or cononpsupplement as copmain or cononpmain, it'll not calculated
14. If more than 1 unique statistic is in the copsupplement or cononpsupplement and the first one is the same as the one in copmain or cononpmain, the second one in copsupplement or cononpsupplement is calculated
15. Duplicated variables in outputorder parameter are removed. The position of duplicated variable in final report is decided by first time it's listed in outputorder parameter

TEST DATA

A modified cars data set shipped with SAS package in SAShelp library is used to demonstrate all examples. For this complete data set, some missing data points are created to illustrate the macro features requiring missing data.

```
data car1;
   set cars;
   call streaminit(12345);
   where drivetrain in ("Front" "Rear");
   if drivetrain = "Front" then do;
         gender = rand("uniform") > 0.5;
         if rand("uniform") > 0.9 then call missing(gender);
   end;
   else gender = rand("uniform") > 0.65;
   temprandom = rand("uniform");
   select;
         when(temprandom > 0.6) carmake = 1;
         when(temprandom > 0.3) carmake = 2;
         otherwise carmake = 0;
   end;
   if rand("uniform") > 0.95 then call missing(mpg_city);
   if rand("uniform") > 0.98 then call missing(mpg_highway);
run;
```

MACRO PARAMETERS INTRODUCTION

Table 1 Macro Parameters

Parameter	**Function**
dsn	Source data set name
group	Group variable name, if provided, report is stratified on this variable, p values are calculated using this variable as group indicator
calist	Categorical variable list, delimited by space
coplist	Continuous variable list requesting p values calculated using parametric test, including T-test and ANOVA, delimited by space
cononplist	Continuous variable list requesting p values calculated using non-parametric test, including Wilcoxon Rank Sum and Kruskal Wallis test, delimited by space
outputorder	Output order of variable in final report
continuousorder	Order of 2 types of continuous variable 1 - variables in coplist show first in final report 2 - variables in cononplist show first in final report
outdsn	Data set name storing information for final report
eratio	Ratio of number of cells with expected frequency less than 5, used for deciding type of test on variable in calist. If calculated ratio is greater than the value specified for this parameter, Fisher's Exact test is performed for p value, otherwise, Chi-square test is conducted for p value. Value of this parameter must range between 0 and 1. By default, it's set at 0.2. Setting it at 0 for only Fisher's Exact test or 1 for only Chi-square test.
percenttype	Type of percentage reported for categorical variables Row - row percentage Column - column percentage Overall column only has column percentage
copmain	Main statistic for continuous variable in coplist, by default, it's mean
copsupplement	Supplemental statistic for continuous variable in coplist, by default, it's standard deviation
cononpmain	Main statistic for continuous variable in cononplist, by default, it's median
cononpsupplement	Supplemental statistic for continuous variable in cononplist, by default, it's inter-quarter range
caexcludelist	List of categorical variables not requesting p value calculation
copexcludelist	List of continuous variables in coplist not requesting p value calculation
cononpexcludelist	List of continuous variables in cononplist not requesting p value calculation
caincludemissing	Include missing information for categorical variables
copincludemissing	Include missing information for continuous variables in coplist
cononpincludemissing	Include missing information for continuous variables in cononplist
missingtop	Yes - missing information is put on top row of report for each variable No - missing information is put on bottom row of report for each variable

	By default, it’s yes
notest	If yes, P values are not calculated for all variables
cadec	Number of decimal on percentages calculated for categorical variable, by default, it’s 2
copdec	Number of decimal on statistics calculated for continuous variable in coplist, by default, it’s 2
cononpdec	Number of decimal on statistics calculated for continuous variable in cononplist, by default, it’s 1
overall	If yes, add overall column to final report, by default, it’s no
paddingchar	Type of padding char used to line up numbers in the report 0 - no padding char 1 – 0, any font can be used for final report 2 – space, only typewriter style font can be used for best visualization By default, its 1
reportinSAS	If yes, a copy of report shown on SAS, by default, it’s no
variableshading	If yes, background color of the cell with variable name/label is grey, by default, it’s yes
labelvariable	If yes, label categorical variable if Fisher’s Exact test is conducted for p value or continuous variable when T-test is conducted for p value and Satterthwaite method is used for calculating degree of freedom. By default, it’s no
suppresswarning	If yes, suppress all warning message from popping up, only shows in log window. By default, it’s yes.
createRTF	If yes, create a RTF file with final report. By default, it’s yes.
compacttable	If yes, final report is in compact style. By default, it’s yes.
tabletitle	Specify report table title
savefilename	Specify name and location of RTF file
showgroup	If yes, show group variable label/name in final report, by default, it’s yes
orderbyformat	If yes, order categories of categorical variable with their formatted value, by default, it’s no
pvaluetop	For continuous variable, put p value on top row of their report, by default, it’s no
orientation	Set up the file orientation of final report, by default, it’s portrait for report without a group variable and landscape for report with a group variable provided
missingpercent	If yes, provide percentage calculation for categorical variable missing information row, by default, it’s no
missinglabel	Row header for categorical variable missing row, by default, it’s Missing
NAlabel	Label for non-computable continuous variable statistic, by default, it’s NA
Showtotalcount	Specify if the total number of observation is shown in the final table

EXAMPLES

1. CREATE VARIABLES SUMMARY WITHOUT GROUP VARIABLE AND P VALUES

```
%Table1Macro(dsn  =  car1,
          calist  =  type origin gender carmake,
       coplist  =  msrp invoice enginesize cylinders horsepower mpg_city
                   mpg_highway weight wheelbase length,
    cononplist  =  msrp invoice enginesize cylinders horsepower mpg_city
                   mpg_highway weight wheelbase length);
```

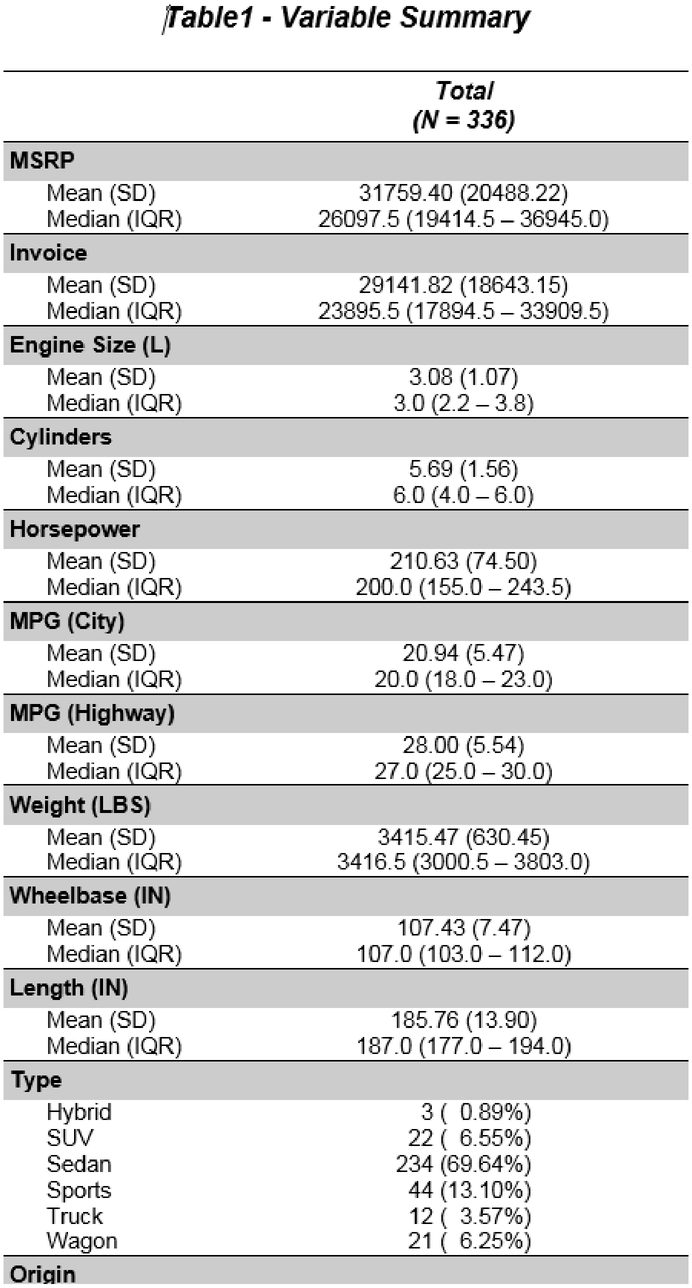

Table1 - Variable Summary

	Total ***(N = 336)***
MSRP	
Mean (SD)	31759.40 (20488.22)
Median (IQR)	26097.5 (19414.5 – 36945.0)
Invoice	
Mean (SD)	29141.82 (18643.15)
Median (IQR)	23895.5 (17894.5 – 33909.5)
Engine Size (L)	
Mean (SD)	3.08 (1.07)
Median (IQR)	3.0 (2.2 – 3.8)
Cylinders	
Mean (SD)	5.69 (1.56)
Median (IQR)	6.0 (4.0 – 6.0)
Horsepower	
Mean (SD)	210.63 (74.50)
Median (IQR)	200.0 (155.0 – 243.5)
MPG (City)	
Mean (SD)	20.94 (5.47)
Median (IQR)	20.0 (18.0 – 23.0)
MPG (Highway)	
Mean (SD)	28.00 (5.54)
Median (IQR)	27.0 (25.0 – 30.0)
Weight (LBS)	
Mean (SD)	3415.47 (630.45)
Median (IQR)	3416.5 (3000.5 – 3803.0)
Wheelbase (IN)	
Mean (SD)	107.43 (7.47)
Median (IQR)	107.0 (103.0 – 112.0)
Length (IN)	
Mean (SD)	185.76 (13.90)
Median (IQR)	187.0 (177.0 – 194.0)
Type	
Hybrid	3 (0.89%)
SUV	22 (6.55%)
Sedan	234 (69.64%)
Sports	44 (13.10%)
Truck	12 (3.57%)
Wagon	21 (6.25%)
Origin	

Figure 1 Result of Example 1

2. CREATE VARIABLES SUMMARY WITH GROUP VARIABLE AND P VALUES

```
%Table1Macro(dsn  =  car1,
          calist  =  type origin gender carmake,
         coplist  =  msrp invoice enginesize cylinders horsepower mpg_city
                     mpg_highway weight wheelbase length,
     cononplist  =  msrp invoice enginesize cylinders horsepower mpg_city
                    mpg_highway weight wheelbase length,
          group  =  Drivetrain);
```

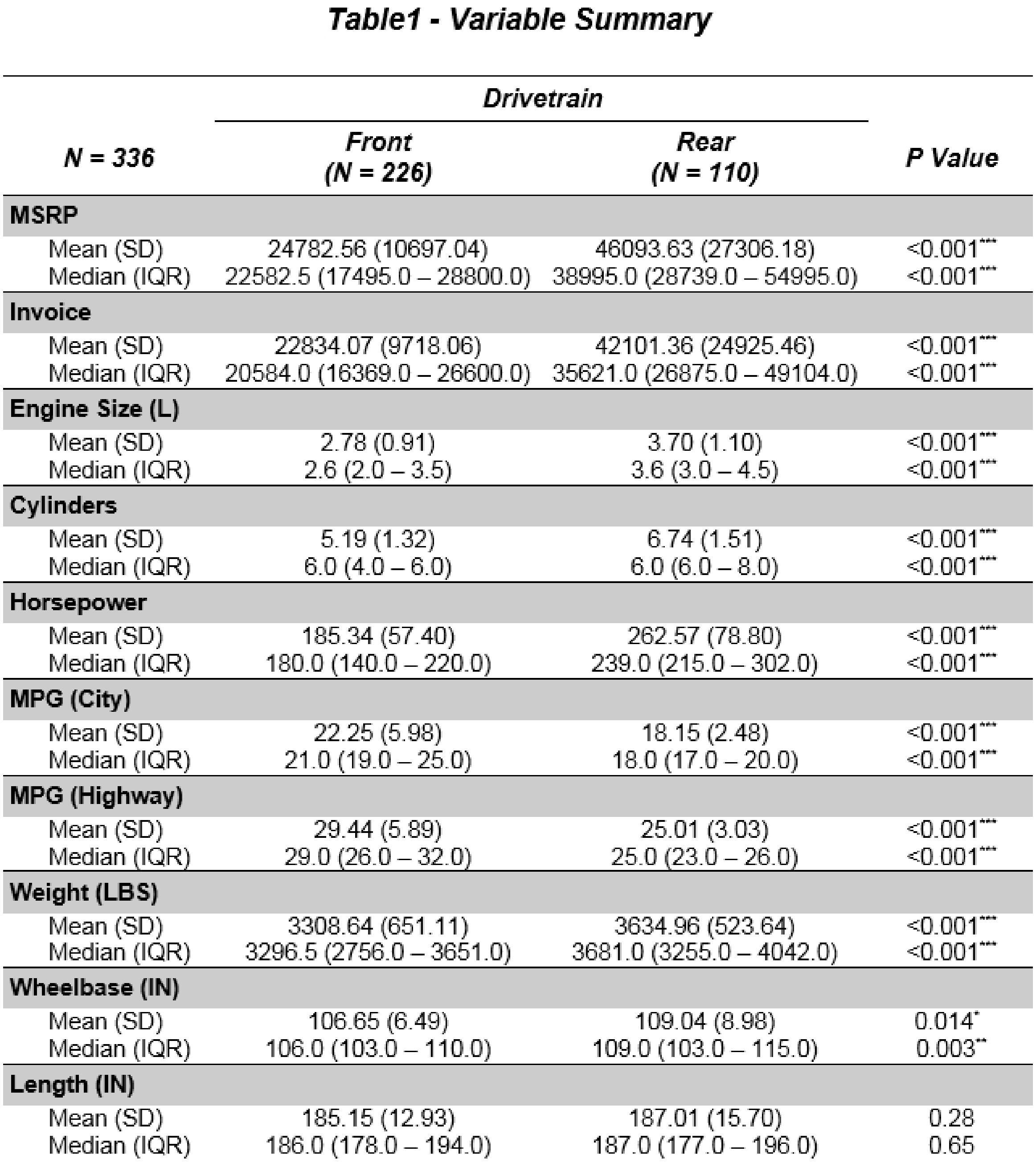

Table1 - Variable Summary

	Drivetrain		
N = 336	***Front (N = 226)***	***Rear (N = 110)***	***P Value***
MSRP			
Mean (SD)	24782.56 (10697.04)	46093.63 (27306.18)	<0.001***
Median (IQR)	22582.5 (17495.0 – 28800.0)	38995.0 (28739.0 – 54995.0)	<0.001***
Invoice			
Mean (SD)	22834.07 (9718.06)	42101.36 (24925.46)	<0.001***
Median (IQR)	20584.0 (16369.0 – 26600.0)	35621.0 (26875.0 – 49104.0)	<0.001***
Engine Size (L)			
Mean (SD)	2.78 (0.91)	3.70 (1.10)	<0.001***
Median (IQR)	2.6 (2.0 – 3.5)	3.6 (3.0 – 4.5)	<0.001***
Cylinders			
Mean (SD)	5.19 (1.32)	6.74 (1.51)	<0.001***
Median (IQR)	6.0 (4.0 – 6.0)	6.0 (6.0 – 8.0)	<0.001***
Horsepower			
Mean (SD)	185.34 (57.40)	262.57 (78.80)	<0.001***
Median (IQR)	180.0 (140.0 – 220.0)	239.0 (215.0 – 302.0)	<0.001***
MPG (City)			
Mean (SD)	22.25 (5.98)	18.15 (2.48)	<0.001***
Median (IQR)	21.0 (19.0 – 25.0)	18.0 (17.0 – 20.0)	<0.001***
MPG (Highway)			
Mean (SD)	29.44 (5.89)	25.01 (3.03)	<0.001***
Median (IQR)	29.0 (26.0 – 32.0)	25.0 (23.0 – 26.0)	<0.001***
Weight (LBS)			
Mean (SD)	3308.64 (651.11)	3634.96 (523.64)	<0.001***
Median (IQR)	3296.5 (2756.0 – 3651.0)	3681.0 (3255.0 – 4042.0)	<0.001***
Wheelbase (IN)			
Mean (SD)	106.65 (6.49)	109.04 (8.98)	0.014*
Median (IQR)	106.0 (103.0 – 110.0)	109.0 (103.0 – 115.0)	0.003**
Length (IN)			
Mean (SD)	185.15 (12.93)	187.01 (15.70)	0.28
Median (IQR)	186.0 (178.0 – 194.0)	187.0 (177.0 – 196.0)	0.65

Figure 2 Result of Example 2

3. SPECIFY OUTPUT ORDER OF VARIABLES

```
%Table1Macro(dsn  =  car1,
          calist  =  type origin gender carmake,
          coplist  =  msrp invoice enginesize cylinders horsepower mpg_city
                      mpg_highway weight wheelbase length,
         cononplist  =  msrp invoice enginesize cylinders horsepower mpg_city
                        mpg_highway weight wheelbase length,
              group  =  Drivetrain,
        outputorder  =  gender carmake mpg_city mpg_highway type);
```

4. REPORT VARIABLE MISSING INFORMATION

```
    %Table1Macro(dsn  =  car1,
              calist  =  type origin gender carmake,
              coplist  =  msrp invoice enginesize cylinders horsepower mpg_city
                          mpg_highway weight wheelbase length,
           cononplist  =  msrp invoice enginesize cylinders horsepower mpg_city
                          mpg_highway weight wheelbase length,
                group  =  Drivetrain,
          outputorder  =  gender carmake mpg_city mpg_highway type,
     caincludemissing  =  yes,
    copincludemissing  =  yes,
  cononpincludemissing  =  yes);
```

5. REQUEST STATISTIC FOR CONTINUOUS VARIABLES OTHER THAN DEFAULT ONE

```
    %Table1Macro(dsn  =  car1,
              calist  =  type origin gender carmake,
              coplist  =  msrp invoice enginesize cylinders horsepower mpg_city
                          mpg_highway weight wheelbase length,
           cononplist  =  msrp invoice enginesize cylinders horsepower mpg_city
                          mpg_highway weight wheelbase length,
                group  =  Drivetrain,
          outputorder  =  gender carmake mpg_city mpg_highway type,
              copmain  =  mean,
        copsupplement  =  clm,
          cononpmain  =  median,
   cononpsupplement  =  range);
```

6. REQUEST ROW PERCENTAGE FOR CATEGORICAL VARIABLE INSTEAD OF COLUMN PERCENTAGE

```
    %Table1Macro(dsn  =  car1,
              calist  =  type origin gender carmake,
              coplist  =  msrp invoice enginesize cylinders horsepower mpg_city
                          mpg_highway weight wheelbase length,
           cononplist  =  msrp invoice enginesize cylinders horsepower mpg_city
                          mpg_highway weight wheelbase length,
                group  =  Drivetrain,
          outputorder  =  gender carmake mpg_city mpg_highway type,
          percenttype  =  row);
```

7. CHANGE OUTPUT DECIMALS FOR CATEGORICAL PERCENTAGES AND STATISTICS CALCULATED FOR CONTINUOUS VARIABLES

```
    %Table1Macro(dsn  =  car1,
              calist  =  type origin gender carmake,
              coplist  =  msrp invoice enginesize cylinders horsepower mpg_city
                          mpg_highway weight wheelbase length,
      cononplist  =  msrp invoice enginesize cylinders horsepower mpg_city
                     mpg_highway weight wheelbase length,
           group  =  Drivetrain,
     outputorder  =  gender carmake mpg_city mpg_highway type,
           cadec  =  0,
          copdec  =  1,
       cononpdec  =  2);
```

8. CHANGE ERATIO CUTOFF OF CHOOSING FISHER'S EXACT TEST TO TO CALCULATE P VALUES FOR CATEGORICAL VARIABLES

```
%Table1Macro(dsn  =  car1,
          calist  =  type origin gender carmake,
          coplist =  msrp invoice enginesize cylinders horsepower mpg_city
                     mpg_highway weight wheelbase length,
      cononplist  =  msrp invoice enginesize cylinders horsepower mpg_city
                     mpg_highway weight wheelbase length,
           group  =  Drivetrain,
     outputorder  =  gender carmake mpg_city mpg_highway type,
          eratio  =  0.5);
```

9. SPECIFY VARIABLES FOR WHICH P VALUE IS NOT CALCULATED

```
%Table1Macro(dsn  =  car1,
          calist  =  type origin gender carmake,
          coplist =  msrp invoice enginesize cylinders horsepower mpg_city
                     mpg_highway weight wheelbase length,
      cononplist  =  msrp invoice enginesize cylinders horsepower mpg_city
                     mpg_highway weight wheelbase length,
           group  =  Drivetrain,
     outputorder  =  gender carmake mpg_city mpg_highway type,
    caexcludelist =  gender carmake,
   copexcludelist =  mpg_city,
cononpexcludelist =  mpg_highway);
```

10. REQUEST NO P VALUE CALCULATION FOR ALL VARIABLES

```
%Table1Macro(dsn  =  car1,
          calist  =  type origin gender carmake,
          coplist =  msrp invoice enginesize cylinders horsepower mpg_city
                     mpg_highway weight wheelbase length,
      cononplist  =  msrp invoice enginesize cylinders horsepower mpg_city
                     mpg_highway weight wheelbase length,
           group  =  Drivetrain,
     outputorder  =  gender carmake mpg_city mpg_highway type,
          notest  =  yes);
```

11. INCLUDE THE OVERALL COLUMN IN THE FINAL REPORT

```
%Table1Macro(dsn  =  car1,
          calist  =  type origin gender carmake,
          coplist =  msrp invoice enginesize cylinders horsepower mpg_city
                     mpg_highway weight wheelbase length,
      cononplist  =  msrp invoice enginesize cylinders horsepower mpg_city
                     mpg_highway weight wheelbase length,
           group  =  Drivetrain,
     outputorder  =  gender carmake mpg_city mpg_highway type,
         overall  =  yes);
```

12. REMOVE SHADING ON VARIABLE NAME LINE

```
%Table1Macro(dsn  =  car1,
          calist  =  type origin gender carmake,
          coplist =  msrp invoice enginesize cylinders horsepower mpg_city
                     mpg_highway weight wheelbase length,
      cononplist  =  msrp invoice enginesize cylinders horsepower mpg_city
                     mpg_highway weight wheelbase length,
           group  =  Drivetrain,
     outputorder  =  gender carmake mpg_city mpg_highway type,
 variableshading  =  no);
```

13. REQUEST SPACE AS THE PADDING CHAR TO LINE UP FILE RESULTS

```
%Table1Macro(dsn  =  car1,
          calist  =  type origin gender carmake,
          coplist =  msrp invoice enginesize cylinders horsepower mpg_city
                     mpg_highway weight wheelbase length,
      cononplist  =  msrp invoice enginesize cylinders horsepower mpg_city
                     mpg_highway weight wheelbase length,
           group  =  Drivetrain,
     outputorder  =  gender carmake mpg_city mpg_highway type,
     paddingchar  =  2);
```

14. REQUEST A COPY OF OUTPUT SHOWING ON SAS INTERNAL WEB BROWSER

```
%Table1Macro(dsn  =  car1,
          calist  =  type origin gender carmake,
          coplist =  msrp invoice enginesize cylinders horsepower mpg_city
                     mpg_highway weight wheelbase length,
      cononplist  =  msrp invoice enginesize cylinders horsepower mpg_city
                     mpg_highway weight wheelbase length,
           group  =  Drivetrain,
     outputorder  =  gender carmake mpg_city mpg_highway type,
     reportinSAS  =  yes);
```

15. LABEL CATEGORICAL VARIABLES IF FISHER’S EXACT TEST IS USED FOR THEIR P VALUE CALCULATION, OR CONTINUOUS VARIABLES IF SATTERTHWAITE METHOD IS CONDUCTED TO COMPUTE DEGREE OF FREEDOM WHEN USING T-TEST FOR THEIR P VALUE CALCULATION

```
%Table1Macro(dsn  =  car1,
          calist  =  type origin gender carmake,
          coplist =  msrp invoice enginesize cylinders horsepower mpg_city
                     mpg_highway weight wheelbase length,
      cononplist  =  msrp invoice enginesize cylinders horsepower mpg_city
                     mpg_highway weight wheelbase length,
              group  =  Drivetrain,
        outputorder  =  gender carmake mpg_city mpg_highway type,
      labelvariable  =  yes);
```

-

TIPS AND SPECIAL APPLICATION

In final report, variables with same label/name are shown under same section. Therefore, we can provide continuous variable a format, and summarized it in both continuous and categorical styles

```
proc format;
      value invoicef   low - <25000 = "Less than 25,000"
                       25000 - high  = "At least 25,000";
run;
proc datasets lib = work nolist;
      modify car1;
      format invoice invoicef.;
quit;

   %Table1Macro(dsn  =  car1,
             calist  =  invoice,
             coplist  =  invoice,
               group  =  Drivetrain);
```

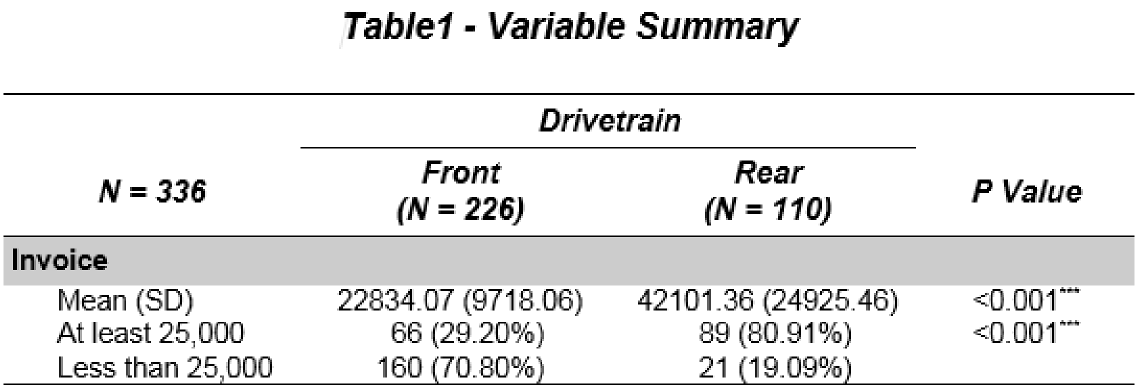

Table1 - Variable Summary

N = 336	Drivetrain		P Value
	Front (N = 226)	Rear (N = 110)	
Invoice			
Mean (SD)	22834.07 (9718.06)	42101.36 (24925.46)	<0.001***
At least 25,000	66 (29.20%)	89 (80.91%)	<0.001***
Less than 25,000	160 (70.80%)	21 (19.09%)	

Figure 3 Result of Tip 1

Report the third statistic for continuous variable

```
     %Table1Macro(dsn  =  car1,
               coplist  =  invoice,
          cononplist  =  invoice,
        cononpmain  =  range,
 cononpsupplement  =  ,
  cononpexcludelist  =  invoice,
                 group  =  Drivetrain);
```

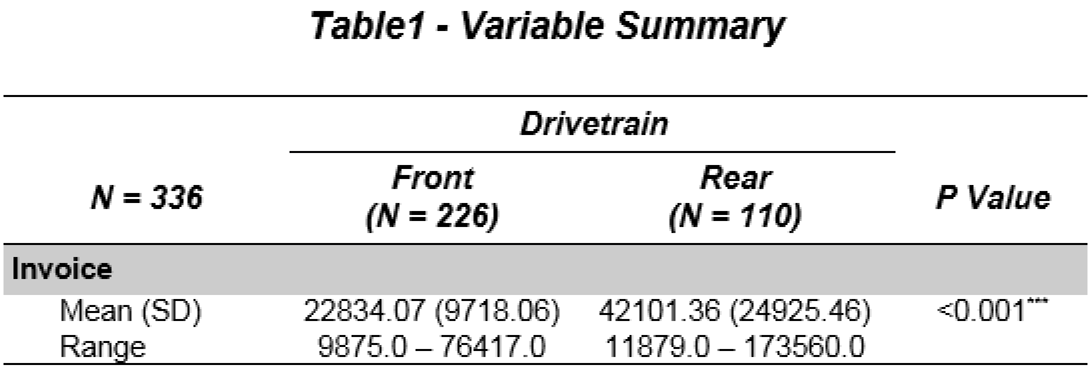

Table1 - Variable Summary

N = 336	Drivetrain		P Value
	Front (N = 226)	Rear (N = 110)	
Invoice			
Mean (SD)	22834.07 (9718.06)	42101.36 (24925.46)	<0.001***
Range	9875.0 – 76417.0	11879.0 – 173560.0	

Figure 4 Result of Tip 2 (1)

```
%Table1Macro(dsn  =  car1,
          coplist  =  invoice,
       cononplist  =  invoice,
          copmain  =  range,
    copsupplement  =  ,
   copexcludelist  =  invoice,
  continuousorder  =  2,
            group  =  Drivetrain);
```

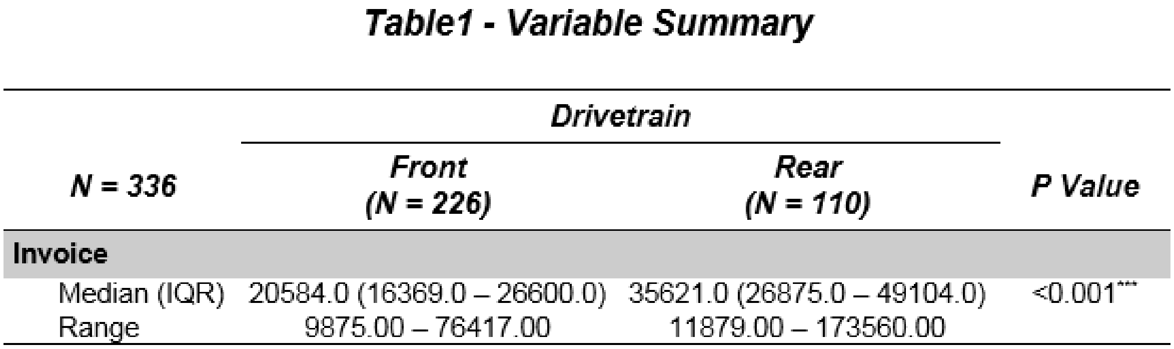

Table1 - Variable Summary

N = 336	Drivetrain		P Value
	Front (N = 226)	Rear (N = 110)	
Invoice			
Median (IQR)	20584.0 (16369.0 – 26600.0)	35621.0 (26875.0 – 49104.0)	<0.001***
Range	9875.00 – 76417.00	11879.00 – 173560.00	

Figure 5 Result of Tip 2 (2)

CONCLUSION

1. Creating variable summary table is a time-consuming and error-prone process
2. This macro provides an easy way to automatically generate publication-ready variable summary table

LIMITATION

1. Program is developed using SAS 9.4.3 and only fully tested in Windows operating system using Microsoft Word 2013
2. It might take very long time if Fisher's Exact Test is performed to calculate p value for certain categorical variable. The p value calculation can be skipped by putting it in caexcludelist
3. Only up to 4 statistics can be reported for continuous variable

4. Total number of character in the name of all variable in source table should fulfill the formula: sum(length of all variable name) + number of all variable - 1 <= 65,534
5. Final result table background color can't be freely changed if value of parameter paddingchar is set as 1

LEARN HOW TO USE THE MACRO

There are 2 SAS files uploaded to PharmaSUG 2019 along with the paper. Anyone interested in the macro can download them. One file is the macro itself and another one is a tutorial file. If you're using windows operating system, you can put 2 files in the same folder and run through the tutorial file which has all the examples listed in this paper.

Hopefully, this macro can help on making your own table 1 easier.

REFERENCES

Art Carpenter. 2007. Carpenter's Complete Guide to the SAS Macro Language, Second Edition. Cary, NC: SAS Institute Inc.

Art Carpenter. 2007. Carpenter's Complete Guide to the SAS REPORT Procedure. Cary, NC: SAS Institute Inc.

Burlew, M. Michele. 2006. SAS Macro Programming Made Easy, Second Edition. Cary, NC: SAS Institute Inc.

ACKNOWLEDGMENTS

Many thanks to all my colleagues who tested and commented about the marco during its development.

Author is motivated, inspired and learned a lot from sample codes posted online by other generous SAS users but completely lost track of when, where and what ideas behind those codes were borrowed during development of this macro. Author is grateful and indebted to all those SAS programmers.

CONTACT INFORMATION

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

Geliang Gan
Yale Center for Analytical Sciences
300 George Street Suite 511
New Haven, CT, 06510
+1-203-737-1044
geliang.gan@yale.edu