Paper 5042-2020

An efficient way to create descriptive tables with pairwise comparison

Ang Gao, University of Texas Southwestern Medical Center; Chul Ahn, University of Texas Southwestern Medical Center

ABSTRACT

This paper illustrates a SAS macro for descriptive tables, which provides Chi-square and Fisher Exact tests for categorical variables, and parametric and nonparametric statistical tests for continuous variables. A formatted output table includes Mean ± Standard deviation, Median (25th percentile, 75th percentile), non-missing N, count (%), and statistical test for each p-value. It also provides pairwise p-values if the comparison is among more than two groups. A permutation method has been used in the macro to allow huge amount of variables to be processed automatically. This approach is designed to create and update descriptive tables in an efficient way.

INTRODUCTION

Almost all data analyses start with a descriptive table for baseline characteristic and outcome variables. The table summarizes categorical and continuous variables into two or more subgroups to describe statistical significance between variables and the group variable. The table shows a basic picture of the data regarding to the initial study question, where the outliers can be detected. In addition, the descriptive table provides an idea to select variables for the further analyses.

The statistical tests used for descriptive tables could be two-sample t-test, Wilcoxon rank-sum test, ANOVA, Kruskal-Wallis, Fisher's Exact, or Chi-Square test depend on the variable type. The macro introduced in this paper can automatically choose proper statistical test for each variable. Additionally, the macro provides appropriate pairwise p-values if there are more than two groups. There is no multiple comparison adjustment are made when pairwise tests are conducted simultaneously. Readers may use the Bonferroni correction after the table is generated.

Most of the time, the descriptive table includes almost all variables in raw data, and the number of variables could be huge. For example, an investigator needs to select the biomarkers statistically associated with an outcome variable from about hundreds of biomarker variables. Obviously, it is a significant amount of work and very time-consuming as well if you run hundreds of times PROC FREQ manually. To solve this problem, the proposed macro uses a loop to automatically count and produce all variables in the list one by one, and save all results to a SAS dataset. To explain the macro in order, the Cars dataset in SASHelp library is used as an example throughout this work.

USER DEFINED MACRO VARIABLES

The first step is to create a variable list. All variable names in the descriptive table are saved into a dataset, which can be done with PROC CONTENTS, VColumn dataset in SASHelp library, or INPUT statement in DATA STEP. Thus, you can automatically create a series of macro variables to store the number of variables and variable names using PROC SQL. Since the statistical methods are different for categorical and continuous variables, you

need to save the names of categorical variables and continuous variables separately. In this step, the number of categories and category values of the group variable have been saved automatically into a series of macro variables.

Below is the sample program to define the macro variables before the analyses:

```
options minoperator mlogic;
%let group=DriveTrain;
%let groupcat=yes;
data catlist;
length name  $64;
input name  @@;
cards;
Type_  Origin
;
run;
%let fisherlist=DUMMY MAKE model type;
proc sql noprint;
select count(distinct name) into: nCVN from catlist;
select name into: CVN1 -: CVN%left(&nCVN.) from catlist;
quit;
data conlist;
length name   $64;
input name   @@;
cards;
MPG_City   Cylinders
;
run;
proc sql noprint;
select count(distinct name) into: nVN from conlist;
select name into: VN1 -: VN%left(&nVN.) from conlist;
quit;
proc freq data=work;
          table &group./out=nq;
          ods output OneWayFreqs=gname;
run;
data gname;length &group._L $64;set gname;
          &group._L=strip("&group.=|"||&group.);
run;
proc sql noprint;
select count(*) into: ng from gname where not missing(&group.);
select &group. into: g1-:g%left(%trim(&ng)) from gname where not
missing(&group.);
select &group._L into: L1-:L%left(%trim(&ng)) from gname where not
missing(&group.);
select "N="||put(count,6.0) into: nq1-:nq%left(%trim(&ng.)) from nq where
not missing(&group.);
select "N="||put(sum(count),6.0) into: nq99 from nq where not
missing(&group.);
quit;
```

There are 3 macro variables need to be input manually by the users. The macro variable "Group" is to save the group variable name, while the macro variable "groupcat" is to indicate that the type of the group variable is whether character(groupcat=yes) or numeric(groupcat=no). A categorical variable may have too many categories, for example, "Make" and "Model" in dataset Cars. There could be a warning in SAS log for situation like this.

```
WARNING: Computing exact p-values for this problem may require much time
and memory. Press the system interrupt key to terminate exact computations.
```

The macro variable "fisherlist" is to exclude those categorical variables from Fisher exact calculation. In order to perform a valid statistical analyses, categories of those variables need to be combined later.

CATEGORICAL VARIABLES

For categorical variables, PROC FREQ is used for calculating count (%), chi-square, or Fisher's exact p-value. The macro catches whether or not there is a Warning for Chi-square test using WARN=OUTPUT option in TABLE statement. Fisher's exact test is used in the macro when more than 20% of the table cells have expected frequencies that are less than five. The same analyses are performed for all combinations of pairwise comparisons. Please see the code:

```
%macro cat;
%do i=1 %to &NCVN.;
%put &&CVN&i..;

data work;set work;
    if missing(&group.) then delete;
    &&CVN&i..=upcase(&&CVN&i..);
run;
proc datasets;delete t;run;quit;
proc freq data=work;
    table &group. * &&CVN&i../chisq(warn=output);
%if not ( %sysfunc(upcase(&&CVN&i..)) in ( %sysfunc(upcase(&fisherlist.)) )
)
%then %do;
    exact fisher;
%end;
    ods output CrossTabFreqs=fq ;
    output out=t chisq;
run;
data fq;set fq;
    m1=put(Frequency,6.0);
    s1=put(RowPercent,5.2);
if missing(&group.) or missing(&&CVN&i..) then
    s1=put(Percent,5.2);
%if %sysfunc(upcase(&groupcat.))=NO %then %do;
    if missing(&group.) then &group.=99/*"Total"*/;
%end;
%else %if %sysfunc(upcase(&groupcat.))=YES %then %do;
    if missing(&group.) then &group.="99"/*"Total"*/;
%end;
if missing(&&CVN&i..) then delete;
    ms=m1|| '(' || s1 || '%'||')' ;
    keep &group. &&CVN&i.. ms;
run;
proc sort data=fq;by &&CVN&i..;run;
proc transpose data=fq out=fqT;
by &&CVN&i..;
ID &group.;
var ms;
run;
data fqT;set fqT;
length Variable Category Label $128;
Variable="&&CVN&i..";Category=left(&&CVN&i..);Label="&&CVN&i..";
drop &&CVN&i.. _name_;
run;
```

```
%if %sysfunc(exist(t)) %then %do;
data t;set t;
length test $10 PValue $20;

if WARN_PCHI=1 then do;
    test="Fisher";pv=XP2_FISH;
    if 0<XP2_FISH<0.0001 then PValue='<.0001*'; else PValue=put(XP2_FISH,
6.4)||"*";
    if XP2_FISH=. and 0<P_PCHI<0.0001 then PValue='<.0001';
    else if XP2_FISH=. and P_PCHI>=0.0001 then PValue=put(P_PCHI, 6.4)||"";
end;
else if WARN_PCHI=0 then do;
    test="Chi-Square";pv=P_PCHI;
    if 0<P_PCHI<0.0001 then PValue='<.0001'; else PValue=put(P_PCHI, 6.4);
end;
if missing(XP2_FISH) and missing(P_PCHI) then PValue="N/A";

keep test WARN_PCHI P_PCHI XP2_FISH PValue pv;
run;
%end;
%else %do;
data t;length test $10 PValue $20;
PValue="N/A";
run;
%end;

proc sort data=fqT;by Variable Category;run;
data fqT;set fqT;by variable Category;if not first.variable then variable="
";run;
******************;
proc datasets;delete tpair;run;quit;
%macro cat_pw;
%do k=1 %to %eval(&ng.-1);
%do kk=%eval(&k.+1) %to &ng.;

proc datasets;delete t&&g&k..&&g&kk..;run;quit;
proc freq data=work;
%if %sysfunc(upcase(&groupcat.)) = YES %then %do;
    where &group. in ("&&g&k.." "&&g&kk..");
%end;
%else %do;
    where &group. in (&&g&k.. &&g&kk..);
%end;
    table &group. * &&CVN&i../chisq(warn=output);
%if not ( %sysfunc(upcase(&&CVN&i..)) in ( %sysfunc(upcase(&fisherlist.)) )
)
%then %do;
    exact fisher;
%end;
    output out=t&&g&k..&&g&kk.. chisq;
run;
%if %sysfunc(exist(t&&g&k..&&g&kk..)) %then %do;
data t&&g&k..&&g&kk..;set t&&g&k..&&g&kk..;length test $10
PV&&g&k..&&g&kk.. $20 variable $128;
variable="&&CVN&i..";

if WARN_PCHI=1 then do;
    test="Fisher";
```

```
    if 0<XP2_FISH<0.0001 then PV&&g&k..&&g&kk..='<.0001*'; else
PV&&g&k..&&g&kk..=put(XP2_FISH, 6.4)||"*";
    if XP2_FISH=. and 0<P_PCHI<0.0001 then PV&&g&k..&&g&kk..='<.0001';
    else if XP2_FISH=. and P_PCHI>=0.0001 then
PV&&g&k..&&g&kk..=put(P_PCHI, 6.4);
end;
else if WARN_PCHI=0 then do;
    test="Chi-Square";
    if 0<P_PCHI<0.0001 then PV&&g&k..&&g&kk..='<.0001'; else
PV&&g&k..&&g&kk..=put(P_PCHI, 6.4);
end;
if missing(XP2_FISH) and missing(P_PCHI) then PV&&g&k..&&g&kk..="N/A";

keep test variable WARN_PCHI P_PCHI XP2_FISH PV&&g&k..&&g&kk..;
run;
%end;
%else %do;
data t&&g&k..&&g&kk..;length test $10 PV&&g&k..&&g&kk.. $20 variable $128;
variable="&&CVN&i..";PV&&g&k..&&g&kk..="N/A";
run;
%end;
proc sort data=t&&g&k..&&g&kk..;by variable;run;

data tpair;
%if %sysfunc(exist(tpair)) %then %do;merge tpair t&&g&k..&&g&kk..;%end;
%else %do;set t&&g&k..&&g&kk..;%end;
run;

%end;
%end;
%mend cat_pw;
%cat_pw;

data tab_t;merge fqT t tpair;run;
******************;

data tab_cat;
%if &i=1 %then %do;
set tab_t;
%end;
%else %do;
set tab_cat tab_t;
%end;
run;

%end;
%mend cat;
```

The following code shows how the macro is called:

```
%include "C:\ClinicalScience\Macros\DescriptiveTable.sas";
%cat;
```

CONTINUES VARIABLES

The macro includes both parametric (%anova) and non-parametric (%kw) method for continuous variables. PROC GLM and PROC NPAR1WAY are used for ANOVA, Kruskal-Wallis, and Wilcoxon rank-sum Test. PROC TTEST computes two-sample T-test for equal or unequal variances. PROC MEANS is used for obtaining Mean ± Standard deviation, Median (25th percentile, 75th percentile) and non-missing N.

In the pairwise comparison, both parametric and non-parametric method are performed. Here is the code:

```
%macro anova;
%do i=1 %to &NVN.;
proc glm data=work;
    class &group.;
    model &&VN&i..=&group.;
    ods output ModelANOVA=acon ;
run;quit;
data acon;set acon;length post $128;
    Post="&&VN&i..";
    if HypothesisType=3;
    keep post ProbF;
run;

/*pairwise*/
proc datasets;delete APall;run;quit;
%macro ANOVA_pw;
%do k=1 %to %eval(&ng.-1);
%do kk=%eval(&k.+1) %to &ng.;

proc ttest data=work;
%if %sysfunc(upcase(&groupcat.)) = YES %then %do;
    where &group. in ("&&g&k.." ,"&&g&kk..");
%end;
%else %do;
    where &group. in (&&g&k.. &&g&kk..);
%end;
    class &group.;
    var &&VN&i..;
    ods output Statistics=ms TTests=tt Equality=ev;
run;
data tt;merge tt ev;by variable;
    if probF>=0.05 and variances="Unequal" then delete;
    if probF<0.05 and variances="Equal" then delete;
run;
data AP&k.&kk.;set tt;length post $128;
    Post="&&VN&i..";
    rename probt=_&k.&kk.;
    keep post probt;
    format probt 6.4;
run;
proc sort data=AP&k.&kk.;by post;run;

data APall;
%if %sysfunc(exist(APall)) %then %do;merge APall AP&k.&kk.;by post;%end;
%else %do;set AP&k.&kk.;%end;
run;

%end;
%end;
%mend ANOVA_pw;
%ANOVA_pw;

proc sort data=acon;by post;run;
data atemp;merge acon APall;by post;run;
data atab;
%if %sysfunc(exist(atab)) %then %do;set atab atemp;%end;
```

```
%else %do;set atemp;%end;
run;
%end;
%mend anova;

%macro kw;
%do i=1 %to &NVN.;

proc means data=work noprint maxdec=1;
    class &group.;
    var &&VN&i..;
    output out=conmn mean=mn std=sd n=n median=md p25=p25 p75=p75 min=_min
max=_max;
run;
data nn;set conmn;keep &group. _freq_;run;
data conmn;set conmn;
    if missing(&group.) then do; &group.=99; end;
    ms=trim(left(put(mn,12.3)))||"±"||trim(left(put(sd,12.3)));
    no="N="||trim(left(put(n,6.0)));
    mq=trim(left(put(md,12.3)))||"
("||trim(left(put(p25,12.3)))||","||trim(left(put(p75,12.3)))||")";
/*  mq=trim(left(put(md,12.3)))||"
("||trim(left(put(_min,12.3)))||","||trim(left(put(_max,12.3)))||")";*/
run;
data Tconmn;set conmn;length Post Category $128;
    Post="&&VN&i..";Category=put(_n_,1.0)||": "||&group.;
    drop _type_ _freq_ ;
run;

proc npar1way data=work;
class &group.;
var &&VN&i..;
ods output KruskalWallisTest=kw;
run;
data kw;
%if %sysfunc(exist(kw)) %then %do;set kw;length post $128;
    Post="&&VN&i..";
    if Name1="P_KW";
    keep post nValue1;
    format nValue1 6.4;
%end;
%else %do;length post $128;
    Post="&&VN&i..";
    nValue1=.;
    keep post nValue1;
    format nValue1 6.4;
%end;
run;

proc datasets;delete kwall;run;quit;
%macro pair;
%do k=1 %to %eval(&ng.-1);
%do kk=%eval(&k.+1) %to &ng.;

proc npar1way data=work;
%if %sysfunc(upcase(&groupcat.)) = YES %then %do;
    where &group. in ("&&g&k.." ,"&&g&kk..");
%end;
```

```
%else %do;
    where &group. in (&&g&k.. &&g&kk..);
%end;
class &group.;
var &&VN&i..;
ods output WilcoxonTest=kw&k.&kk.;
run;
data kw&k.&kk.;
%if %sysfunc(exist(kw&k.&kk.)) %then %do;set kw&k.&kk.;length post $128;
    Post="&&VN&i..";
    if Name1="P2_WIL";
    rename nValue1=_np&k.&kk.;
    keep post nValue1;
    format nValue1 6.4;
%end;
%else %do;length post $128;
    Post="&&VN&i..";
    nValue1=.;
    rename nValue1=_np&k.&kk.;
    keep post nValue1;
    format nValue1 6.4;
%end;
run;
proc sort data=kw&k.&kk.;by post;run;

data kwall;
%if %sysfunc(exist(kwall)) %then %do;merge kwall kw&k.&kk.;by post;%end;
%else %do;set kw&k.&kk.;%end;
run;

%end;
%end;
%mend pair;
%pair;

proc sort data=Tconmn;by post;run;
proc sort data=kw;by post;run;
data temp;merge Tconmn kw kwall;by post;run;
data tab;
%if %sysfunc(exist(tab)) %then %do;set tab temp;%end;
%else %do;set temp;%end;
run;

%end;
%mend kw;

%anova;
%kw;
```

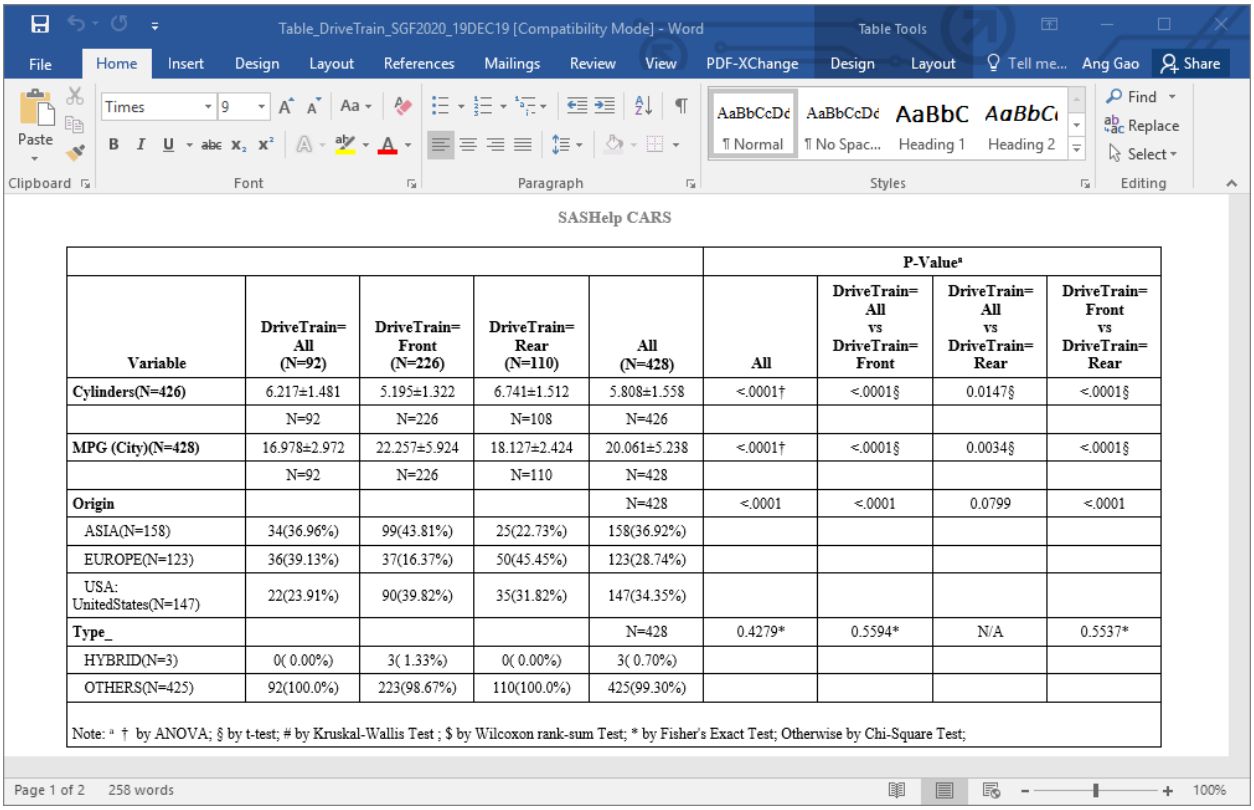
The parametric results are then saved into data “atab”, and non-parametric results are saved into data “tab”. In addition, two results can be further merged together as follows:

```
proc sort data=atab;by post ;run;
proc sort data=tab;by post category;run;
data tab_con;merge atab tab;by post;
run;
```

OUTPUT TABLES

After statistics calculations, the macro re-constructs the result dataset to the desired output format. If the number of categories in the group variable is greater than three, Microsoft Excel file maybe a better choice to carry a wide table with all pairwise comparison p-values. Due to the limited space and the different requirement for the output format, the program for this part is not included.

The following is the example for output tables.

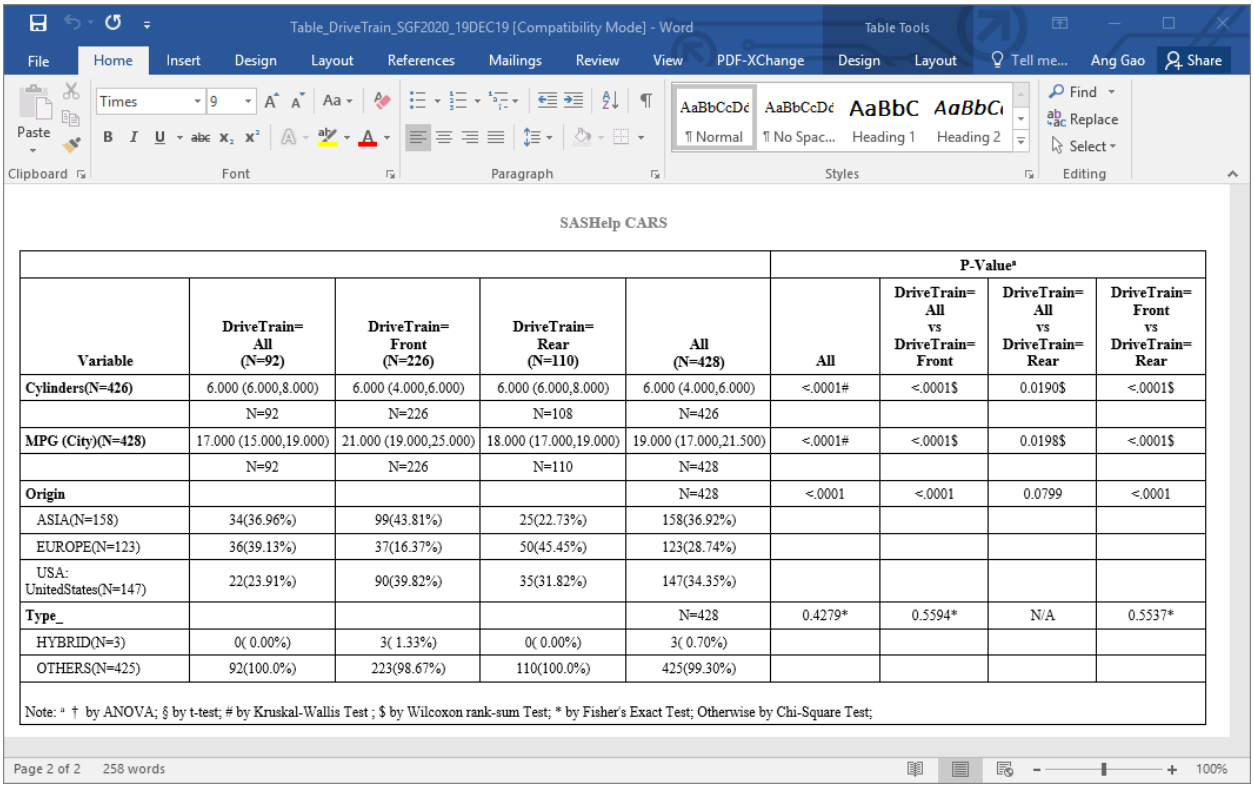


SASHelp CARS

					P-Value[a]			
Variable	**DriveTrain= All (N=92)**	**DriveTrain= Front (N=226)**	**DriveTrain= Rear (N=110)**	**All (N=428)**	**All**	**DriveTrain= All vs DriveTrain= Front**	**DriveTrain= All vs DriveTrain= Rear**	**DriveTrain= Front vs DriveTrain= Rear**
Cylinders(N=426)	6.217±1.481	5.195±1.322	6.741±1.512	5.808±1.558	<.0001†	<.0001§	0.0147§	<.0001§
	N=92	N=226	N=108	N=426				
MPG (City)(N=428)	16.978±2.972	22.257±5.924	18.127±2.424	20.061±5.238	<.0001†	<.0001§	0.0034§	<.0001§
	N=92	N=226	N=110	N=428				
Origin				N=428	<.0001	<.0001	0.0799	<.0001
ASIA(N=158)	34(36.96%)	99(43.81%)	25(22.73%)	158(36.92%)				
EUROPE(N=123)	36(39.13%)	37(16.37%)	50(45.45%)	123(28.74%)				
USA: UnitedStates(N=147)	22(23.91%)	90(39.82%)	35(31.82%)	147(34.35%)				
Type_				N=428	0.4279*	0.5594*	N/A	0.5537*
HYBRID(N=3)	0(0.00%)	3(1.33%)	0(0.00%)	3(0.70%)				
OTHERS(N=425)	92(100.0%)	223(98.67%)	110(100.0%)	425(99.30%)				

Note: [a] † by ANOVA; § by t-test; # by Kruskal-Wallis Test ; $ by Wilcoxon rank-sum Test; * by Fisher's Exact Test; Otherwise by Chi-Square Test;

Display 1. Table output for parametric methods



SASHelp CARS

Variable	DriveTrain= All (N=92)	DriveTrain= Front (N=226)	DriveTrain= Rear (N=110)	All (N=428)	P-Value[a] All	P-Value[a] DriveTrain= All vs DriveTrain= Front	P-Value[a] DriveTrain= All vs DriveTrain= Rear	P-Value[a] DriveTrain= Front vs DriveTrain= Rear
Cylinders(N=426)	6.000 (6.000,8.000)	6.000 (4.000,6.000)	6.000 (6.000,8.000)	6.000 (4.000,6.000)	<.0001#	<.0001$	0.0190$	<.0001$
	N=92	N=226	N=108	N=426				
MPG (City)(N=428)	17.000 (15.000,19.000)	21.000 (19.000,25.000)	18.000 (17.000,19.000)	19.000 (17.000,21.500)	<.0001#	<.0001$	0.0198$	<.0001$
	N=92	N=226	N=110	N=428				
Origin				N=428	<.0001	<.0001	0.0799	<.0001
ASIA(N=158)	34(36.96%)	99(43.81%)	25(22.73%)	158(36.92%)				
EUROPE(N=123)	36(39.13%)	37(16.37%)	50(45.45%)	123(28.74%)				
USA: UnitedStates(N=147)	22(23.91%)	90(39.82%)	35(31.82%)	147(34.35%)				
Type_				N=428	0.4279*	0.5594*	N/A	0.5537*
HYBRID(N=3)	0(0.00%)	3(1.33%)	0(0.00%)	3(0.70%)				
OTHERS(N=425)	92(100.0%)	223(98.67%)	110(100.0%)	425(99.30%)				

Note: ᵃ † by ANOVA; § by t-test; # by Kruskal-Wallis Test ; $ by Wilcoxon rank-sum Test; * by Fisher's Exact Test; Otherwise by Chi-Square Test;

Display 2. Table output for nonparametric methods

CONCLUSION

By automating the process of defining macro variables and calculating statistics, this macro can generate and update descriptive tables efficiently, especially, when there are huge amount of variables in the descriptive table.

REFERENCES

SAS Institute Inc. 2019. "TABLES Statement." Accessed December 31, 2019.

http://support.sas.com/documentation/cdl/en/procstat/68142/HTML/default/viewer.htm#procstat_freq_syntax08.htm.

CONTACT INFORMATION

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

Ang Gao
Ang.Gao@UTSouthwestern.edu