PharmaSUG 2023 - Paper QT-121

Smart Use of SAS® Output System and SAS® Macro for Statistic Test Selection

Mengxi Wang, University of Southern California, Los Angeles, CA

ABSTRACT

Choosing the optimum statistical test is crucial to generate accurate results in a quantitative research study. However, digging through piles of diagnostic test reports for useful results may, at times, be a tedious task for beginners in the field of biostatistics, especially when there are many variables to examine. The purpose of this paper is to share an efficient way to automate the process of selection between non-parametric and parametric tests with the SAS® output system and SAS® macro. The statistical tests that are used as examples are some of the most widely used tests in Table 1, the summary table of population characteristics most medical publications: Chi-square test, Fisher exact test, Independent two-sample T-test, Wilcoxon-Mann-Whitney test, ANOVA test, and Kruskal-Wallis test. The test selection is based on variable type, variable category, sample size, and sample distribution. The results of the test selection and associated p-values, as well as basic descriptive statistics, will then be compiled into one dataset, which can either be printed for use as a handy reference or exported to Excel for further formatting.

INTRODUCTION

When conducting quantitative research studies, descriptive tables to demonstrate differences among study populations and provide the audience an overview of the data characteristics. However, these tables can be very time-consuming to create. Variables often need to be individually specified, as different SAS® procedures may be required for the calculation of statistics tests. Some variables, due to smaller sample size or skewed sample distribution, require non-parametric tests, such as Fisher exact test, Wilcoxon-Mann-Whitney Test or Kruskal Wallis test; while other variables with larger sample size or normal distribution, need parametric tests for more accurate results, such as T-test, Chi-square test and ANOVA test. The inspection process can be very tedious when handling a large number of variables, as the diagnosis tests often produce pages of reports.

Since SAS® stores all calculated statistical data from each procedure in temporary datasets, it is possible to suppress the cumbersome SAS® reports. With the powerful Output Delivery System (ODS) and “OUT=” option, you can convert selective diagnostic tests report into datasets. The key values of the tests can then be stored in macro variables and used to guide to guide your code to the relevant tests.

In the following sections, I will outline a macro which automate the process of test selection and output the results with p values into a readable dataset. The dataset generated can be

effectively used for Table 1 in many research papers. The example presented in this paper will use Base SAS® and is appropriate for the beginning to intermediate statistical programmer or analyst.

MACRO PARAMETERS

```
%macro table1(dataset=, row_=, col_=, n_=, median=, dec=,
output=);
```

- dataset: Identifies the user input dataset.
- col_: Name of the comparison groups, delimited by space.
- row_: List of variables to be summarized in the table, delimited by space. All continuous variables need to be numeric. All categorical variables need to be character.
- n_: List of continuous variables that required to be analyzed using parametric tests, irrespective of the results of normality tests provided by this macro. Delimited by space.
- median: Options to output median and interquartile range or mean and standard deviation for all continuous variables.

  1: output median(q1-q3);

  0: output mean(SD).
- dec: Number of decimal places that you want to keep in the table.
- output: Name of the output dataset.

DETERMINE VARIABLE TYPE

First, the macro uses the function “vtype” to determine variable type and stores the result into the macro variable “type”. If the variable is numeric, the macro will proceed to execute the test procedures for continuous variables; otherwise follow the test procedures for categorical variables instead. This allows the macro to differentiate between the two variable types and appropriately apply the corresponding statistical tests.

```
data _null_;
  set &dataset(obs=1);
  call symput('type',vtype(&row));
run;
%if &type=N %then %do;
```

CONTINUOUS VARIABLE

When choosing a statistical test for continuous variables, two key points to consider are the number of categories to compare and whether the data follows a normal distribution. By taking these factors into account, you can select an appropriate test that will help you draw meaningful conclusions from your data. To examine the difference between two independent groups of continuous variables, the Independent Two-Sample T-Test or Wilcoxon-Mann-Whitney Test are the most commonly used tests. The Independent Two-Sample T-Test is suitable for normally distributed samples, while the Wilcoxon-Mann-Whitney Test is better for skewed or non-normal data and small sample sizes. If the comparison groups have more than two categories, an ANOVA test could be used when the samples are normally distributed, and a Kruskal–Wallis test is used when the distribution is not normal.

In this macro, the first step of diagnostic test is to check the data distribution. SAS® has offered four commonly used normality tests: Shapiro-Wilk test, Kolmogorov-Smirnov (K-S) test, Anderson-Darling test, and Cramer-von Mises test. These tests are generally powerful to detect abnormality in the data.

In this macro, I use Shapiro-Wilk / Kolmogorov-Smirnov test in PROC UNIVARIATE procedure for normality test.

```
proc univariate data=&dataset normal noprint;
  var  &row;
  by &col;
  output out=test_
    normal=normal
    PROBN=PROBN;
run;

data test;
  set test_;
  var="&row";
  varcat="&row";
  if min(probn)< 0.05 then normal_=0; else normal_=1;
  call symput('normal',normal_);
  drop normal PROBN ;
run;
```

“Normal” is the variable name of the normality test statistic, and “PROBN” represents the p value of the normality test. SAS® outputs the Shapiro test statistic for sample sizes up to 2,000, and the Kolmogorov-Smirnov test result for sample size greater than 2,000. The test result is used to generate a macro “normal” which indicates the normality.

Relying solely on test statistics and p-values can lead to incorrect conclusions in certain situations. SAS® also provides powerful Graphical methods, such as normal quantile-quantile plot (Q-Q plot) and kernel density plot, to display the overall distribution along with the statistical tests. It is recommended to use graphical methods in conjunction with statistical

tests to ensure accurate and reliable results. The following codes produce a Q-Q plot to further investigate the normality.

```
proc univariate data=&dataset normal noprint;
     var &row;
     by &col;
     qqplot/normal;
     title "Q-Q Plot for &row by &col (Normal=&normal)";
run;
```

If the Shapiro/Kolmogorov-Smirnov test rejects the normality assumption, but upon reviewing Q-Q plots, you determine that a parametric test would better optimize the statistical power, you can add the variable name to the "n_=" parameter of the macro. This will cause the macro to ignore the normality test results and directly conduct the parametric test:

```
%if "&n_" ne "" %then %do;
   %let ncount= %sysfunc(countw(&n_.));
   %do k= 1 %to &ncount;
      %let n = %scan(&n_, &k, ' ');
     %if %sysfunc(index(&row, &n))>0 %then %do;
        %let normal=1;
     %end;
    %end;
%end;
```

The second step of diagnostic test is to count the number of categories. The following process will count the column categories and store it into a macro variable “n_cat”.

```
proc sql noprint;
   select count(unique(&col)) into:
   n_cat
   from &dataset;
quit;
```

Now, with the results of the normality test and category count stored in macro variables "normal" and "n_cat", we can perform the appropriate test and output the p-values. The following code will output the test results of the Wilcoxon-Mann-Whitney Test or Kruskal-Wallis Test when the normality assumption does not hold:

```
%if &normal=0 %then %do;
    proc npar1way data=&dataset wilcoxon noprint;
         class &col;
         var &row;
         output out=wil;
    run;

  data teststat;
```

```
    set wil;
    var=_VAR_;
    if &n_cat=2 then do;
      testtype="Wilcoxon";
      pvalue=PT2_WIL;
    end;
    else if &n_cat=3 then do;
       testtype="Kruskal-Wallis";
       pvalue=P_KW;
    end;
  keep var pvalue testtype;
    run;
%end;
```

“PT2_WIL” is the default variable name for p value of Wilcoxon-Mann-Whitney Test, whereas “P_KW” is the name for the p value of Kruskal–Wallis test.

If the normality assumption is not rejected, PROC TTEST can be an effective tool to accurately estimate the difference between the two population means. Notice that there’s no “out=” option in PROC TTEST procedure, this is when ODS OUTPUT statement comes into play. Using ODS OUTPUT, this section of code puts the PROC TTEST results into two different SAS® data sets, one containing the statistics portion of the PROC TTEST, whereas the other containing variance equality section. “ODS exclude all” and “ODS exclude none” is used to suppress output of the test results.

```
ods exclude all;
    proc ttest data=&dataset;
     class &col;
     var &row;
     ods output Equality=Equality ttests=ttest;
    run;
ods exclude none;
```

Now we need to take one step further, to determine which probability of mean equality is the one to use. “Probt” refers to the p value for the Satterthwaite or Pooled T-test, and “ProbF” is the p value for the of Equality of Variances test. Merging the output together, we can then decide whether to use Satterthwaite or Pooled T-test based on a selected cutoff value for the probability of the variances being equal.

```
proc sql;
     create table ttestp as
     select a.Variable, a.Method, a.Probt, b.ProbF
     from  ttest as a, Equality as b
     where a.variable=b.variable;
quit;
```

```
 data teststat;
   set ttestp;
   testtype="TTest";
   if ProbF<0.05 then do;
      if Method='Satterthwaite';
   end;
   else do;
      if Method='Pooled';
   end;
   rename Probt=pvalue
   variable=var;
   drop Method ProbF ;
run;
```

The ANOVA test results can be output into a dataset using similar way. the specific code is not provided here due to page limitation.

CATEGORICAL VARIABLE

The Chi-square test is a popular choice for testing the independence of two categorical variables. However, this test assumes that the expected minimum count is at least 5 in at least 80% of the cells in the contingency table. If this assumption is not met, using the Fisher exact test is a more appropriate option.

To determine whether the expected minimum count criterion is met, the macro use the PROC FREQ procedure in SAS®, which outputs the expected cell counts for each cell in the contingency table. If the percentage of cells with an expected minimum count less than 5 is greater than 20%, this suggests that the Chi-square test may not be reliable, and a macro variable named "warning" will be generated. It indicates that using the Fisher exact test may be more appropriate in this scenario.

```
proc freq data=&dataset noprint;
     tables &row*&col / chisq fisher sparse outexpect out=_out1;
     output out=stats n chisq fisher;
run;

data _out1;
  set _out1;
  if percent ne . ;
  warn=(.<expected<=5);*an indicator of cells which expected
minimum count less than 5;
run;
```

```
proc sql;
     create table _out2 as
     select count(count) as freq, count(warn) as warn_n
     from _out1;
quit;

data _out2;
   set _out2;
   pct_lt5=warn_n/freq;
   warning=(pct_lt5>=.2);
   call symput('warning',warning);
run;
```

In the dataset “stats” created by PROC FREQ procedure, the variable “XP2_FISH” refers to the p value of fisher exact test, and the variable “P_PCHI” refers to the p value of Chi-square test. The macro can select which test statistic to use base on the value of “warning”.

The final part of the code brings together the analytic tests and descriptive statistics, as explained in further detail in the appendix. This process iterates through each variable input specified in the "row_=" parameter and consolidates the results into a single table.

FINAL OUTPUT:

Using the default datasets sashelp.cars, sashelp.heart as examples, here are some sample output tables.

Example 1:

The sashelp.heart data set provides results from the Framingham Heart Study. In this example, we are trying to compare some patients character between alive and dead(status). The “median” is set to 0 to output mean and standard deviation, and the decimal place is set to 1.

```
%table1(dataset=Sashelp.heart,  row_=Diastolic  Systolic  Chol_Status
height weight, col_=Status, n_=, median=0, dec=1, output=stat);
```

Descriptive Analysis for Status

varname	varcat	Alive	Dead	total	P_value	testtype
Diastolic	Diastolic	82.8(±11.3)	89.5(±14.3)	85.4(±13.0)	<0.0001	Wilcoxon
Systolic	Systolic	131.2(±18.6)	146.2(±27.9)	136.9(±23.7)	<0.0001	Wilcoxon
Chol_Status	Borderline	1186(37.8%)	675(35.1%)	1861(36.8%)	<0.0001	Chisq
	Desirable	998(31.8%)	407(21.2%)	1405(27.8%)		
	High	951(30.3%)	840(43.7%)	1791(35.4%)		
height	height	64.7(±3.6)	65.0(±3.6)	64.8(±3.6)	<0.001	Wilcoxon
weight	weight	149.9(±28.0)	158.3(±29.7)	153.1(±28.9)	<0.0001	Wilcoxon

After examining the Q-Q plots, we determine that the distribution of both height and weight is approximately normal, so we decide to use parametric tests for these variables to obtain more reliable results:

```
%table1(dataset=Sashelp.heart, row_=Diastolic Systolic Chol_Status
height weight, col_=Status, n_=height weight, median=0, dec=1,
output=stat), output=stat);
```

Descriptive Analysis for Status

varname	varcat	Alive	Dead	total	P_value	testtype
Diastolic	Diastolic	82.8(±11.3)	89.5(±14.3)	85.4(±13.0)	<0.0001	Wilcoxon
Systolic	Systolic	131.2(±18.6)	146.2(±27.9)	136.9(±23.7)	<0.0001	Wilcoxon
Chol_Status	Borderline	1186(37.8%)	675(35.1%)	1861(36.8%)	<0.0001	Chisq
	Desirable	998(31.8%)	407(21.2%)	1405(27.8%)		
	High	951(30.3%)	840(43.7%)	1791(35.4%)		
height	height	64.7(±3.6)	65.0(±3.6)	64.8(±3.6)	0.006	TTest
weight	weight	149.9(±28.0)	158.3(±29.7)	153.1(±28.9)	<0.0001	TTest

Example 2:

The sashelp.cars data set provides the 2004 car data. In this example, we are trying to compare characteristics between different drivetrain of the cars. The “median” is set to 1 to output median and interquartile range, and the decimal place is set to 1.

```
%table1(dataset=Sashelp.cars, row_=MSRP Invoice Origin Weight Length,
col_=DriveTrain, n_= , median=1,dec=1, output=stat);
```

Descriptive Analysis for DriveTrain

varname	varcat	All	Front	Rear	total	P_value	testtype
MSRP	MSRP	33,510.0(26,195.0-41,357.5)	22,582.5(17,495.0-28,800.0)	38,995.0(28,739.0-54,995.0)	27,635.0(20,329.5-39,215.0)	<0.0001	Kruskal-Wallis
Invoice	Invoice	30,879.0(23,961.5-36,899.5)	20,584.0(16,369.0-26,600.0)	35,621.0(26,875.0-49,104.0)	25,294.5(18,851.0-35,732.5)	<0.0001	Kruskal-Wallis
Origin	Asia	34(37.0%)	99(43.8%)	25(22.7%)	158(36.9%)	<0.0001	Chisq
	Europe	36(39.1%)	37(16.4%)	50(45.5%)	123(28.7%)		
	USA	22(23.9%)	90(39.8%)	35(31.8%)	147(34.3%)		
Weight	Weight	3,935.5(3,484.0-4,729.0)	3,296.5(2,756.0-3,651.0)	3,681.0(3,255.0-4,042.0)	3,474.5(3,103.0-3,978.5)	<0.0001	Anova
Length	Length	187.0(179.0-193.0)	186.0(178.0-194.0)	187.0(177.0-196.0)	187.0(178.0-194.0)	0.60	Kruskal-Wallis

TIPS:

- The “out=” option is a convenient tool to create an output data set. However, it is not available for many procedures. Instead, ODS OUTPUT can output the result into datasets from almost any procedure. You can use ODS TRACE statement to find the path and name of the temporary dataset that SAS® has created.
- You can use the "NOPRINT" option to suppress reports when using an OUTPUT statement, and use "ODS exclude all" and "ODS exclude none" to suppress report. These options allow you to manipulate the elements of output, giving you more control over the result.

LIMITATION

This macro has some limitations that I would like to make a few improvements in the future:

- The macro does not report missing values**;**
- It provides a limited set of statistical tests for independent two-sample tests and does not include tests for paired samples**;**
- Variable labels are not considered in the macro**;**
- The macro only categorizes variables into broad types (continuous and categorical), which may not be detailed enough for some analysis**.**

CONCLUSION

Generating descriptive statistics and performing comparisons between study groups can be a time-consuming process, particularly when dealing with a large number of variables that require different statistical tests. Despite some limitations, the code presented in this paper offers an automated solution for selecting some of the most widely used tests in Table 1. It

generates reader-friendly descriptive tables by outputting calculated statistical data into temporary datasets and storing key values in macro variables.

Unfortunately, the program is too long to include in this paper -- but I am glad to send it via e-mail upon request. Please see the contact information below.

REFERENCE:

Wicklin, Rick. “ODS Output Any Statistic: A General Tool for Data Analysis.” The DO Loop: Statistical Programming Techniques, 9 Jan. 2017. Available at

https://blogs.sas.com/content/iml/2017/01/09/ods-output-any-statistic.html

Peng, Guangbin. “Testing Normality of Data using SAS®” Available at https://www.lexjansen.com/pharmasug/2004/Posters/PO04.pdf

CONTACT INFORMATION

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

Mengxi Wang
USC Alfred E. Mann School of Pharmacy
1985 Zonal Ave
Los Angeles, CA 90089
(404) 632-4874
mengxiwa@usc.edu/mengxi.wang275@gmail.com