**ABSTRACT**

The ability to integrate information, perform accurate and unbiased analysis, and generate flexible reports is in great demand in healthcare research. This paper uses a real-world example to illustrate an automated and easily managed process to perform a comprehensive analysis. It will demonstrate macros that help programmers determine what types of statistical tests will be appropriate to examine the probability of mean differences based on sample size and type and distribution of variables. For example, the macros automatically examine if a distribution is parametric or nonparametric and then decide which test should be carried out. Furthermore, the macros automatically perform various tests such as paired or unpaired t-tests, McNemar, Fisher Exact, or Wilcoxon test upon the previous determination.

**INTRODUCTION**

For most studies in healthcare research, the purpose of studies varies. Therefore, the analysis plans differ as well. The purpose of this paper is to use a real-world example to illustrate an automated and easily managed process by which intermediate and innovative SAS programmers can perform a comprehensive analysis based on the purpose of studies.

In details, the paper will demonstrate principles and macros that help programmers select appropriate statistical tests to examine the probability of a mean difference based on the purpose of a study, sample size, type of variables and distribution of data. For example, the procedure will automatically examine distributions and decide if a parametric or non-parametric test should be performed. In addition, based on type of a study such as pre- or post-experiment comparison, or type of variables such as categorical or continuous variables, macros will automatically select appropriate tests to compare if there is a statistical significant difference between or among the groups. Finally, the paper will show some tips for automatically renaming variables, formulating an equation from a pre-defined model and arranging results in a user-specified order to be outputted.

**BACKGROUND**

The example used for this paper is from a large urban health care agency. In order to define a better way to provide higher quality for patients with functional disabilities, the agency developed a new program and applied it to a group of patients who needed physical therapy. However, to determine if the program achieved the goals of the project, the agency would like to have a report to compare functional improvement, service utilization, patients satisfaction, and other outcome indicators between a group of patients who were in the program, named the intervention group, and a group of patients who were not enrolled into the program but had similar characteristics at start of care.

**PRINCIPLES OF SELECTING APPROPRIATE STATISTICAL TESTS**

Selecting appropriate statistical tests is very important. There are different types of tests, and it can be quite complicated and confusing to determine which test is appropriate, especially for programmers without a strong statistical background. However, in order to go further, the followings will briefly highlight some principles and guidelines for choosing appropriate statistical tests.

Overall, selecting appropriate statistical significance tests is based on the number of samples, sample size, nature of dependent and independent variables, and distribution of data.

In terms of samples, there are independent and dependent samples. For independent samples, the numbers of samples don’t have to be the same, and they are independent and are randomly selected from different groups. For dependent samples, they are usually paired samples, and a typical example is samples used to test effect before and after a treatment or paired samples from both intervention and control groups.

Basically, there are two types of data, and each one of them could be defined further:

- **Categorical data:**
A. Nominal data: numbers are often used to represent categories, and the order of values is irrelevant, and the statistical analysis should not depend on that ordering. Nominal data that take on one of two distinct values, such as male and female are said to be dichotomous or binary.

B. Ordinal data: categorical variables having ordered scales are called ordinal variables, such as patients' satisfaction scale (very satisfied, satisfied, fair, unsatisfied, very unsatisfied).

**Numerical data:**

A. Discrete data (count data): the numbers represent actual measurable quantities rather than mere labels. They are restricted to taking on only specified values – often integers or counts. Fractional values are not possible. The example of discrete data is number of accidents in a given day, number of hospitalization, and etc.

B. Continuous data: data that represent measurable quantities but are not restricted to taking on certain specified values. In all instances, fractional values are possible. The examples of continuous data could be temperature, weights, and etc.

In summary, according to the nature of dependent and independent variables, underlining assumptions of data and samples, the following table lists some common often used statistics significance tests as a guideline of this paper.

**Brief summary of selecting appropriate statistical significance test:**

<table>
<thead>
<tr>
<th>Samples</th>
<th>Independent Variable</th>
<th>Dependent Variable</th>
<th>Assumptions</th>
<th>Recommended Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>For Two Independent Samples</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Categorical Test</strong></td>
<td>Nominal (any level)</td>
<td>Nominal</td>
<td>Large sample and less than 20% of cells have expected counts ≤ 5</td>
<td>Chi-square test</td>
</tr>
<tr>
<td></td>
<td>Nominal (any level)</td>
<td>Nominal</td>
<td>Small sample</td>
<td>Fisher's exact test</td>
</tr>
<tr>
<td></td>
<td>Ordinal/Nominal (any level)</td>
<td>Ordinal</td>
<td>Large sample</td>
<td>Mantel haenszel Chi-square test</td>
</tr>
<tr>
<td><strong>Numerical Test</strong></td>
<td>Binary</td>
<td>Continuous/Count</td>
<td>Normal distribution</td>
<td>Two sample t-test</td>
</tr>
<tr>
<td></td>
<td>Ordinal/Nominal (&gt; 2)</td>
<td>Continuous/Count</td>
<td>Normal distribution</td>
<td>Wilcoxon-Mann-Whitney test</td>
</tr>
<tr>
<td></td>
<td>Ordinal/Nominal (&gt; 2)</td>
<td>Continuous/Count</td>
<td>Nonparametric</td>
<td>One-way ANOVA</td>
</tr>
<tr>
<td></td>
<td>Ordinal/Nominal (&gt; 2)</td>
<td>Continuous/Count</td>
<td>Nonparametric</td>
<td>Kruskal-Wallis Test</td>
</tr>
</tbody>
</table>

| For Two Dependent Samples |                      |                    |                                                  |                           |
| **Categorical Test**     | Binary               | Binary             | Large sample                                     | McNemar test              |
|                          | Ordinal              | Nominal/Ordinal    | Small sample                                     | Exact McNemar test        |
|                          | Ordinal              | Ordinal            | Stratified samples                               | Cochran-Mantel-Haenszel test |
|                          | Binary               | Ordinal            | More than 2 levels for the independent variable  | Cohen's Kappa             |
| **Numerical Test**       | Binary               | Continuous/Count   | Normal distribution                              | Paired t-test             |
|                          | Ordinal/Nominal (> 2) | Continuous/Count   | Nonparametric                                    | Wilcoxon signed-rank test |
|                          | Ordinal/Nominal (> 2) | Continuous/Count   | Nonparametric                                    | Repeated measures ANOVA   |
|                          | Ordinal/Nominal (> 2) | Continuous/Count   | Nonparametric                                    | Friedman's two way ANOVA  |

**PROCESS ILLUSTRATION**

Section I - Constructing data:

Data are stored in an Oracle database that contains patients' demographic, service utilization, clinical and functional information measured at start of care, every 60 days, and at discharge. After cleaned and organized data was retrieved from the Oracle database, the final data set used for the study was presented as two records per patient: one from the start of care, and the other from the discharge. However, the total numbers of patients who received special treatment and who did not receive special treatment are unequal. In other words, there are two different types of samples involving this study: one is paired sample for comparing pre- and post- status, and the other is two independent samples, received treatment versus did not receive treatment.

Section II – Matching cases and renaming variables:

In order to match patients within a similar range of characteristics at start of care, parameter values from a pre-defined logistic regression model were applied to the data set, which automatically selected and matched patients for
fair comparison.

Section III – Analyzing data:

The objectives of this project are to examine if there are significant difference between the group that received treatment versus the group that did not receive treatment, named control group, as well as the significant changes before and after treatment for the treatment group. There are two types of tests needed for the study. One is testing statistical significance in terms of outcome at discharge between two groups, and the other is to compare patient status before and after treatment for the treatment group.

A macro, named %STATTEST, which generates a subset based on user-defined parameters, then calls other macros according to type of variables and a purpose of analysis such as a pre- and/or post-treatment comparison, or both. It then organizes results from both numerical and categorical variable analyses together into a user-defined order and outputs them. It calls macros %DOTEST_1 if a type of analysis is for two unpaired independent samples, or %DOTEST_2 if a type of analysis is for a paired sample comparison, generates P-values that indicate the probability of getting mean differences between the groups.

%macro stattest (vardata, /*a list of variables need to be tested*/
data, /*an data set need to be analyzed*/
type, /*type of analysis 1=two independent samples, 2=paired samples*/
depvar, /*name of a dependent variable such as treatment indicator or before and after indicator*/
equal, /*a sign to decide to retrieve a subset*/
value, /*a value of the dependent variable*/
in)          /*if an analysis is for before or after treatment or for comparing both*/
/* create a data set based on if two sample or one sample tests */
data temp;
set &data;
if &depvar %str(&equal) &value; run;
/*perform tests based on the type of variables: &vartype=0 (nominal variable) &vartyp=1(continuous variables) &vartyp=2 (ordinal variable)*/
%do vartyp=0 %to 2;
/*retrieve variables list from pre-defined variable list table */
data var;
set &vardata;
if mean=&vartyp and &in=1; run;
/* total number of variables to be analyzed */
proc sql noprint;
select count(*) into :count from var; quit;
%if &type=2 %then %do; /*paired sample test, need to restructure data and rename variables */
proc sql noprint;
select label, label, compress(label) || '_d', compress(label) || '_d' into :soc1 thru :soc%trim(&count), :varprior separated by ' ', :varafter separated by ' ', :dis1 thru :dis%trim(&count) from var; quit;
%dotest_1; /*run tests*/
%end;
%else %do; /*two independent sample test*/
proc sql noprint;
select label, label into :var separated by ' ', :var1 thru :var%trim(&count) from var; quit;
%dotest_2; /*run tests*/
%end;
%end;
/*organize results into the order that you specified */
data &in;
set &depvar._&in._type&type_0 &depvar._&in._type&type_1 &depvar._&in._type&type_2;
run;
proc sort data=&in by label;
proc sort data=&vardata by label;
data &in;
merge &in &vardata(keep=label obs); by label; run;
proc sort data=&in by obs; run;
%mend;
For example, the following codes illustrate how to run an analysis by using the macros mentioned above. The example #1 runs an analysis to compare patients' characteristics across two independent groups, the treatment versus the control group; and the example #2 returns results that compare patients' status before and after treatment.

Example 1: %statetest (var, data, 1, intervention, >=, 0, pre);
Example 2: %statetest (var, data, 2, intervention, =, 1, both);

MACROS USED FOR THE PROCESS

In addition to the above macro, the following macros are used for the purpose of the process, and detail codes are referred to the appendix of this paper.

%DOTEST_1: A macro performs probability significance tests for two independent samples. It examines the normalization for continuous/count variables first, and then based on the sample size decides if a result from the Shapiro-Wilk or the Kolmogorov test should be used as a criterion for normalization judgment. Furthermore, based on type of variables and distributions, it calls the macros %PAIR_UN, %PAIR_NORM, or %CHISQ_1, and organizes all results together.

%DOTEST_2: A macro performs probability significance tests for a paired sample comparison. Similar to the macro %DOTEST_1, it examines the normalization for continuous/count variables first, and then based on the sample size decides if a result from the Shapiro-Wilk or the Kolmogorov test should be used as a criterion for normalization judgment. In addition, based on types of variables and distributions, it calls the macros %TTEST_UN, %TTEST_N, or %CHISQ_2, and organizes all results together.

%PAIR_UN: A macro that carries the Wilcoxon signed-rank test for nonparametric distributed variables in a paired sample test. It generates difference between pairs and ranks it, and then examines its statistical significance.

%PAIR_NORM: A macro that performs a paired t-test for a normally distributed continuous/count variable and outputs its result through ODS.

%TTEST_UN: A macro that outputs the result from a Wilcoxon Mann-Whitney test through the procedure PROC NPAR1WAY. It examines the probability of there being mean difference for a non-parametric distributed continuous/counts variable in an unpaired independent sample test.

%TTEST_N: A macro that performs two-independent sample t-test for continuous independent variables with normal distributions. It examines if the independent variable has equal variance and then decides which probability test result should be read in.

%CHISQ_1: A macro that first examines cell counts in a contingency table, and then conducts either a Chi-square test or Fisher's exact test based on whether one or more cells has an expected frequency of five or more. In addition, if there is no statistical test to be outputted, the macro creates a dummy result with a p-value of null.

%CHISQ_2: A macro that examines the statistical difference for ordinal variables for two independent samples. It performs a Mantal haenszel Chi-square test and then outputs result.

%CHISQ_3&4: Macros that examine the statistical difference for categorical variables between two paired samples. It performs a McNemar's test and Cochran-Mantel-Haenszel test, respectively, and then outputs result.

CONCLUSION

Selecting appropriate statistical significant tests is a key to help researchers to interpret and evaluate study results. There are many types of different statistical tests and sometimes are very confusing. However, the most helpful and important thing to assist you to select appropriated statistical test is to understand the nature of variables, relationship of samples, underlining data assumptions, and objectives of your analysis. In addition, being familiar with the capabilities of SAS procedures is the path to help you to achieve goals.

REFERENCES
SAS OnlineDoc, Version Eight, Copyright (c) 1999 SAS Institute Inc., Cary, NC, USA. All rights reserved.
http://www.ats.ucla.edu
CONTACT INFORMATION
Your comments and questions are valued and encouraged. The author may be contacted at:

Liping Huang
Center for Home Care Policy and Research
Visiting Nurse Service of New York
11th Floor, 5 Penn Plaza
New York, NY 10001
Telephone: (212)609-5770
Email: liping.huang@vnsny.org

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. Other brand and product names are trademarks of their respective companies.

APPENDIX: SAS Code Illustrations

*--------------------------------------------------------------------------- *
| For Paired Dependent Sample Test                                         |
*--------------------------------------------------------------------------- *
%macro dotest_1; /*For paired samples*/
%do i=1 %to &count; /*start a loop for each individual variable*/
/*create a subset to make sure that variables used for before and after comparison are not missing */
data test; set temp; if &&soc&i ne . and &&dis&i ne .; run;
%if &vartyp=1 %then %do; /*for numerical variables*/
/*check normality*/
ods output TestsForNormality=normal(keep=varname testlab pvalue);
proc univariate data=test normal;
var &&soc&i &&dis&i;
proc sql;
select count(&&soc&i) into :total from test; quit;
data normal(keep=varname pvalue); set normal;
/* if total observations is less than 2000 then use the result from Shapiro-Wilk test; else use the result from Kolmogorov-Smirnov test */
if &total < 2000 then do; if lowcase(testlab)='w'; end; else do; if lowcase(testlab)='d'; end;
proc transpose data=normal out=normal; id varname;
data _null_; set normal; pvalue=max(&&soc&i, &&dis&i); call symput('p', pvalue); run;
%if &p < 0.05 %then %pair_un;
%else %pair_nom;
%end;
%else %if &vartyp=0 %then %do; /*for 2X2 table*/
%if &vartyp=0 %then %do; /*for 2X2 table*/
%else %pair_nom; %end;
%else %do; %end; /* for ordinal variables*/
%end;
data out_type&type._&vartyp(rename=(pvalue=pvalue_&depvar._&value));
format pvalue pvalue6.4;
set %do i=1 %to &count; out&i %end;; run;
%mend;

*--------------------------------------------------------------------------- *
| For Unpaired Independent Sample Test                                     |
*--------------------------------------------------------------------------- *
%macro dotest_2; /*For unpaired samples*/
%do i=1 %to &count; /*start a loop for each individual variable*/
%if &vartyp=1 %then %do; /*if it is a numerical variable*/
/*check normality*/
proc sql; select count(&&var&i) into :total from temp; quit;
ods output TestsForNormality=normal(keep=testlab pvalue);
proc univariate data=temp normal; var &&var&i;
data _null_; set normal; pvalue=max(&&var&i); call symput('p', pvalue); run;
%if &p < 0.05 %then %ttest_un;
%else %ttest_n;
%end;
%end;
%else %if &vartyp=0 %then %do; %chisq_1; %end; /* for nominal variables*/
%else %do; %chisq_2; /* for ordinal variables*/
%end;
%end;
data out_type&type._&vartyp(rename=(pvalue=pvalue_&in));format pvalue pvalue6.4; set %do i=1 %to &count; out&i %end;; run;
%mend;
*-------------------------------------------------------------------------- *
| Wilcoxon-Mann-Whitney test for normally distributed independent samples |
*-------------------------------------------------------------------------- *
%macro ttest_un;
proc npar1way data=temp wilcoxon noprint ;
  class &depvar;
  var &&var&i;
  output out=a(keep=pt2_wil rename=(pt2_wil=pvalue)) wilcoxon; run;
%if %sysfunc(exist(a)) %then %do;
  data out&i(keep=label pvalue); format label $25.; set a; label=lowcase("&&var&i");
  proc datasets library=work; delete a; quit;
%end;
%else %do; /*if there is no statistics computed, then create a dummy data set*/
  data out&i(keep=label pvalue); format label $25.; label=lowcase("&&var&i"); pvalue =.; run;
%end;
%mend;

*-------------------------------------------------------------------------- *
| T-test for normally distributed independent samples |
*-------------------------------------------------------------------------- *
%macro ttest_n; /*t-test for normally distributed independent samples*/
ods output equality=equal(keep=probf);
ods output ttests=ttest(keep=variances probt);
proc ttest data=temp; class &depvar; var &&var&i;
/*examine if statistics was computed or not*/
%if %sysfunc(exist(ttest)) %then %do;
  data out&i(rename=(probt=pvalue)); format label $25.; set equal;
  if probf < 0.05 then do; set ttest; if lowcase(variances)='unequal'; end;
  else do; set ttest; if lowcase(variances)='equal'; end; label=lowcase("&&var&i"); run;
  proc datasets library=work; delete ttest equal; quit;
%end;
%else %do; /*if no statistics was computed, then create a dummy data set*/
  data out&i(keep=label pvalue); format label $25.; label=lowcase("&&var&i"); pvalue =.; run;
%end;
%mend;

*-------------------------------------------------------------------------- *
| Paired t-test for normally distributed paired samples |
*-------------------------------------------------------------------------- *
%macro pair_nom;
ods output ttests=out&i(keep=probt label);
proc ttest data=; paired &&soc&i*&&dis&i;
data out&i(rename=(probt=pvalue)); format label $25.; set out&i; label=lowcase("&&soc&i"); run;
%mend;

*-------------------------------------------------------------------------- *
| Wilcoxon signed-rank test for not normally distributed paired samples |
*-------------------------------------------------------------------------- *
%macro pair_un;
data diff; set test; diff=&&dis&i - &&soc&i;run;
data out&i(keep=pvalue label where =(lowcase(testlab)='s'));
proc univariate data=diff; var diff;
data out&i(keep=pvalue label); format label $25.; set out&i; label=lowcase("&&soc&i"); run;
%mend;
**Chisq test or Fisher exact test**

```sas
%macro chisq_1;
proc freq data=temp noprint;
tables &depvar*&&var&i / out=a(keep=count) chisq fisher;
%if %sysfunc(exist(b)) %then %do;
   proc sql noprint;
     select min(count) into :min from a;
   quit;
%end;
%if &min <= 5 %then %do;
   data out&i(rename=(xp2_fish=pvalue)); format label $25.;
   set b; label=lowcase("&&var&i");
   keep label xp2_fish;
%end;
%else %do;
   data out&i(rename=(p_pchi=pvalue)); format label $25.;
   set b; label=lowcase("&&var&i");
   keep label p_pchi;
%end;
proc datasets library=work; delete a b; quit;
%end;
%mend;
```

**Mantel-Haenszel test for ordinal variables with independent sample tests**

```sas
%macro chisq_2;
proc freq data=temp; tables &depvar*&&var&i / chisq;
output agree out=a(keep=p_mhchi); run;
%if %sysfunc(exist(a)) %then %do;
   data out&i(rename=(p_mhchi=pvalue)); format label $25.;
   set a; label=lowcase("&&soc&i");
   proc datasets library=work; delete a; quit;
%end;
%mend;
```

**Cochran-Mantel-Haenszel test for multiple row/column paired sample test**

```sas
%macro chisq_3;
proc freq data=test; tables &&soc&i*&&dis&i/ agree;
   output agree out=a(keep=p_cmhcor); run;
%if %sysfunc(exist(a)) %then %do;
   data out&i(rename=(p_cmhcor=pvalue)); format label $25.;
   set a; label=lowcase("&&soc&i");
   proc datasets library=work; delete a; quit;
%end;
%mend;
```

**McNemar test for 2X2 table paired sample tests**

```sas
%macro chisq_4; /*McNemar test for 2X2 tables paired sample tests*/
proc freq data=test; tables &&soc&i*&&dis&i/ agree;
   output agree out=a(keep=p_mcnem); run;
%if %sysfunc(exist(a)) %then %do;
   data out&i(rename=(p_mcnem=pvalue)); format label $25.;
   set a; label=lowcase("&&soc&i");
   proc datasets library=work; delete a; quit;
%end;
%mend;
```