Create an Informative Summary from PROC T-TEST Output

Jennifer K. Warner, Marsh Affinity Group Services, Ft. Washington, PA
Email: Jennifer.K.Warner@marshpm.com / phone: (215)-653-8000

ABSTRACT

When comparing two populations, or modeling binary response, PROC TTEST is a valuable tool for evaluating numeric variables and an effective technique for identifying potential model variable candidates. Once identified, the analyst can more quickly prioritize further investigation of these variables. The disadvantage to PROC TTEST is that it does not summarize all the information you need into one easy to read report. To get the whole story about a particular variable, you need to flip back and forth between three different pieces of output. When there are literally hundreds or thousands of variables to evaluate, poring through the numerous parts and pages of PROC TTEST output can be a daunting, arduous task.

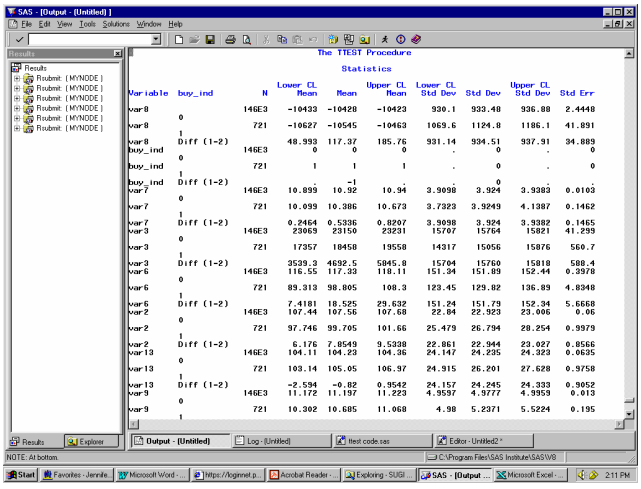
This paper outlines code that takes the various parts of the PROC TTEST output and summarizes all of the vital information for each variable into a SAS® data set. The data set can then either be printed for use as a handy reference or exported to Excel for further cosmetic formatting. The example presented in this paper will use Base SAS ® and SAS/STAT® and is appropriate for the beginning to intermediate statistical programmer or analyst.

INTRODUCTION

PROC TTEST is a useful tool for comparing two populations or for determining potential variables to use for modeling binary response. However, the output from the following code is a bit cumbersome. This code is examining the data set "file1" which is stored in the "mainlib" library. The _numeric_ specification indicates that all numeric variables are to be tested.

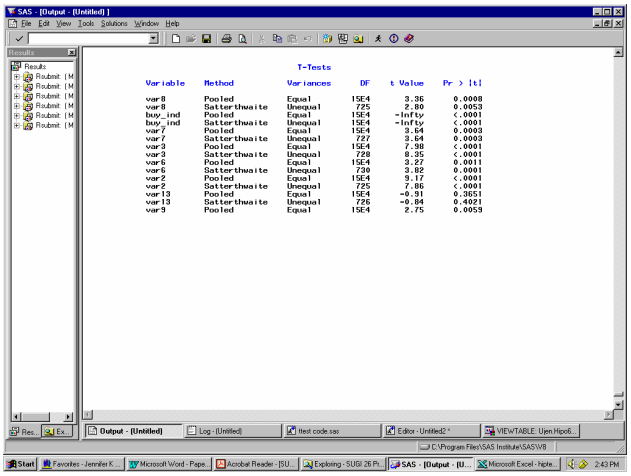
```
proc ttest data = mainlib.file1;
  var _numeric_;
  class buy_ind;
run;
```

The first section of output lists each variable as well as the number of observations falling into each by-group. In this case, the class variable, buy_ind, has a value of 0 or 1 to indicate if a person is a non-buyer or a buyer, respectively. The output also reports the mean and standard deviation of the buyers and non-buyers for each variable.

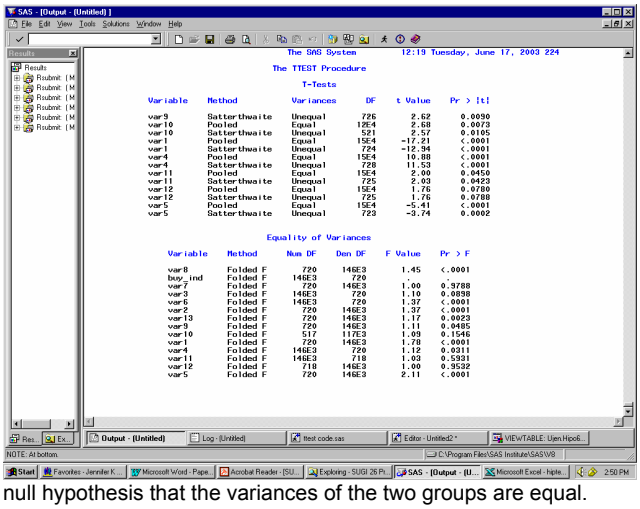


The question is, are the means of the by-groups for particular variables statistically different from each other? For example, if you were looking at "Income" (var3), you would want to know if the mean, or average, income of buyers, $18,458, was statistically significantly different from the mean income of non-buyers, $23,150.

To determine this, you must look at the second part of the output. The T-Tests section lists the results of testing the null hypothesis that the means of the two groups are equal under two different assumptions: 1: The variances of the two groups are equal and 2: The variances are not equal.



As can be seen in the case of var3, the result of statistically different means is the same whether the variances are the same or not. However, the results of the same tests for var8 yield slightly different results for the two variance equality assumptions. Thus, you must look to the third part of the output that tests the



null hypothesis that the variances of the two groups are equal.

Here it can be seen that the probability of the variances of the two groups being the same is very low and thus you should look at the T-test results under the assumption of unequal variances.

When you have hundreds or thousands of variables to look at, this process can be tedious. Manually turning this information into a report to use as a quick reference for your statistical modeling can also be very time consuming. The remainder of this paper will outline steps that can be used to consolidate the output from PROC TTEST into a SAS® data set that can be printed for use as a reference or output for further use as an appendix to a report.

THE RESULT

The data set below is the end result of the code discussed in this paper. It is a SAS data set that includes the variable name, the mean of the buyers, the mean of the non-buyers, the t-value, the probability and finally, a classification of the level of difference found between the two sample means based on an arbitrary cut-off.

Of course, the code can be modified to reflect any binary population and the cutoffs used to determine the level of difference between the two sample means can also be changed depending on the circumstances.

	Variable	mean_buyers	mean_nonbuyers	t Value	Pr > |t|	difference
1	buy_ind	1	0	M	<.0001	highly
2	var1	0.3966712899	0.1604590227	-12.94	<.0001	highly
3	var10	4.1930501931	4.5297555474	2.68	0.0073	weak
4	var11	16.910987483	17.692255943	2.00	0.0450	weak
5	var12	17.467315716	18.268738888	1.76	0.0780	not at all
6	var13	105.05353581	104.23366574	-0.84	0.4021	not at all
7	var2	99.704853391	107.55974434	7.86	<.0001	highly
8	var3	18457.659121	23150.192825	7.98	<.0001	highly
9	var4	0.3300970874	0.5326501495	10.88	<.0001	highly
10	var5	0.0624133148	0.028623755	-3.74	0.0002	somewhat
11	var6	98.804992337	117.32986347	3.82	0.0001	somewhat
12	var7	10.386016555	10.919582249	3.64	0.0003	somewhat
13	var8	-10545.16644	-10427.79182	2.80	0.0053	weak
14	var9	10.685159501	11.197200005	2.75	0.0059	weak

THE PROCESS

STEP ONE

Using ODS, this section of code puts the PROC TTEST output into three different SAS data sets, one for each component of the output. Macro variable references are utilized so that the code can be easily modified for testing additional data sets.

The data set that is being tested is named "file1" and is stored in the "mainlib" library. The resulting data sets to be created will be stored in the "mylib" library and the data set names will have "test1" as a prefix to easily identify them. You may also consider making lib1 equal to the work directory to cut down on the number of permanent data sets being stored.

```
%let lib1 = mylib;
%let lib2 = mainlib;
%let fileorg = file1;
%let pre = test1;

ods output "Statistics" = &lib1..&pre.stats
           "T-Tests" = &lib1..&pre.ttests
           "Equality of Variances" = &lib1..&pre.vars;
  proc ttest data=&lib2..&fileorg;
    class buy_ind;
    var _numeric_;
  run;
ods output close;
```

STEP TWO

The following data set, **mylib.test1stats**, is the result of outputting the Statistics portion of the PROC TTEST through ODS. This is the section of output containing the means and standard deviations of each class level of each variable.

	Variable	buy_ind	N	Lower Limit of Mean	Mean	Upper Limit of Mean	Lower Limit of Std Dev	UMPU Lower Limit of Std Dev	Std Dev	UMPU Upper Limit of Std Dev	Upper Limit of Std Dev	Std Error	Minimum	M
1	var8	0	146E3	-10433	-10428	-10423	930.1	930.1	933.48	936.87	936.88	2.4448	-16863	
2	var8	1	721	-10627	-10545	-10463	1069.6	1069.1	1124.8	1185.5	1186.1	41.891	-16619	
3	var8	Diff (1-2)	.	48.993	117.37	185.76	931.14	931.14	934.51	937.9	937.91	34.889	.	
4	buy_ind	0	146E3	0	0	0	.	.	0	.	.	0	0	
5	buy_ind	1	721	1	1	1	.	.	0	.	.	0	1	
6	buy_ind	Diff (1-2)	.	.	-1	.	.	.	0	.	.	.	.	
7	var7	0	146E3	10.899	10.92	10.94	3.9098	3.9098	3.924	3.9382	3.9383	0.0103	0	
8	var7	1	721	10.099	10.386	10.673	3.7323	3.7306	3.9249	4.1368	4.1387	0.1462	1	
9	var7	Diff (1-2)	.	0.2464	0.5336	0.8207	3.9098	3.9098	3.924	3.9382	3.9382	0.1465	.	
10	var3	0	146E3	23069	23150	23231	15707	15707	15764	15821	15821	41.299	0	
11	var3	1	721	17357	18458	19558	14317	14310	15056	15868	15876	560.7	0	
12	var3	Diff (1-2)	.	3539.3	4632.5	5845.8	15704	15704	15760	15818	15818	588.4	.	

This section of code deals with that part of the PROC TTEST output. The code makes it possible to have one row for each variable. In the PROC TTEST output, there are three rows for each variable - one for buyers, one for non-buyers and one for the differences. Each row in the resulting data set will have 2 new variables - mean_buyers and mean_nonbuyers. These variables are equal to the variable that was previously called "mean".

```
%let lib1 = mylib;
%let pre = test1;
data &lib1..&pre.stats1 (keep = variable class
mean_nonbuyers mean_buyers);
  set &lib1..&pre.stats (rename=(mean=avg));
  if class = '          0'
   then mean_nonbuyers = avg;
  if class = '          1'
   then mean_buyers = avg;
run;
```

	Variable	buy_ind	mean_nonbuyers	mean_buyers
1	var8	0	-10427.79182	.
2	var8	1	.	-10545.16644
3	var8	Diff (1-2)	.	.
4	buy_ind	0	0	.
5	buy_ind	1	.	1
6	buy_ind	Diff (1-2)	.	.
7	var7	0	10.919582249	.
8	var7	1	.	10.386016555
9	var7	Diff (1-2)	.	.
10	var3	0	23150.192825	.
11	var3	1	.	18457.659121
12	var3	Diff (1-2)	.	.

At this point, **mylib.test1stats1**, there are still three rows for each variable. This next piece of code separates the rows for buyers into one file and the rows for non-buyers into another file. So, each of these two files will have only one row for each variable. Rows for differences are just dropped.

```
data &lib1..&pre.buyers(drop = class mean_nonbuyers)
     &lib1..&pre.nonbuyers(drop = class mean_buyers);
  set &lib1..&pre.stats1;
  if class = '          0' then output &lib1..&pre.nonbuyers;
  if class = '          1' then output &lib1..&pre.buyers;
run;
```

Now the two separate files are sorted and merged back together on "Variable" and the only variables that remain are the means of the two groups and the variable name - one row for each variable.

```
 proc sort data = &lib1..&pre.buyers;
   by variable;
run;
proc sort data = &lib1..&pre.nonbuyers;
  by variable;
run;
data &lib1..&pre.statsfinal;
  merge &lib1..&pre.buyers
    &lib1..&pre.nonbuyers;
  by variable;
run;
```

The final data set from this step, **mylib.test1statsfinal**, should look like this.

	Variable	mean_buyers	mean_nonbuyers
1	buy_ind	1	0
2	var1	0.3966712899	0.1604590227
3	var10	4.1930501931	4.5297555474
4	var11	16.910987483	17.692255943
5	var12	17.467315716	18.268738888
6	var13	105.05353581	104.23366574

STEP THREE

This section deals with the second part of the PROC TTEST output - the part that tells you whether or not the means of the by-groups for each variable are statistically equal or not. In this section there are two lines for each variable. One showing the probability of the means being the same if the variances are the same, and one showing the probability of the means being the same if the variances are different.

The following code creates a new variable, "difference", which classifies, into one of four categories, the strength of each probability of the means of the by-groups being different. The code uses this data set, **mylib.test1ttests**, which was created above using ODS.

	Variable	Method	Variances	t Value	DF	Pr > |t|
1	var8	Pooled	Equal	3.36	***	0.0008
2	var8	Satterthwaite	Unequal	2.80	725	0.0053
3	buy_ind	Pooled	Equal	M	***	<.0001
4	buy_ind	Satterthwaite	Unequal	M	***	<.0001
5	var7	Pooled	Equal	3.64	***	0.0003
6	var7	Satterthwaite	Unequal	3.64	727	0.0003
7	var3	Pooled	Equal	7.98	***	<.0001
8	var3	Satterthwaite	Unequal	8.35	728	<.0001
9	var6	Pooled	Equal	3.27	***	0.0011
10	var6	Satterthwaite	Unequal	3.82	730	0.0001

```
%let lib1 = mylib;
%let pre = test1;
data &lib1..&pre.ttests1;
  set &lib1..&pre.ttests;
  length difference $10.;
  if probt le .0001 then difference = 'highly';
  else if probt le .005 then difference = 'somewhat';
  else if probt le .05 then difference = 'weak';
  else difference = 'not at all';
run;
```

The resulting data set, **mylib.test1ttests1** should look like this.

	Variable	Method	Variances	t Value	DF	Pr > |t|	difference
1	buy_ind	Pooled	Equal	M	***	<.0001	highly
2	buy_ind	Satterthwaite	Unequal	M	***	<.0001	highly
3	var1	Pooled	Equal	-17.21	***	<.0001	highly
4	var1	Satterthwaite	Unequal	-12.94	724	<.0001	highly
5	var10	Pooled	Equal	2.68	***	0.0073	weak
6	var10	Satterthwaite	Unequal	2.57	521	0.0105	weak
7	var11	Pooled	Equal	2.00	***	0.0450	weak
8	var11	Satterthwaite	Unequal	2.03	725	0.0423	weak
9	var12	Pooled	Equal	1.76	***	0.0780	not at all
10	var12	Satterthwaite	Unequal	1.76	725	0.0788	not at all

STEP FOUR

The question now is, if any of the "difference" classifications for the same variable don't match, how to determine which probability of mean equality is the one to use without going through the last piece of output and manually checking the probability of the variances being the same for every single variable? The next piece of code merges the output that shows the probability of variances being equal to the output that shows the probabilities of means of the two by-groups being equal. To avoid confusion with the "probf" variable in the means equality section, "probf" in the variance equality section is renamed as "probv".

```
data &lib1..&pre.vars;
  set &lib1..&pre.vars (rename=(probf=probv));
run;
proc sort data = &lib1..&pre.vars;
  by variable;
run;
proc sort data = &lib1..&pre.ttests1;
  by variable difference;
run;
 data &lib1..&pre.merged;
  merge &lib1..&pre.ttests1 (in=a)
        &lib1..&pre.vars    (in=b keep = variable probv);
  by variable;
  if a;
run;
```

The resulting data set, **mylib.test1merged**, will look like this. Note that the same value of "probv" from the variance equality output has been appended to both the "Equal" and "Unequal" rows of the mean equality output.

	Variable	Method	Variances	t Value	DF	Pr > |t|	difference	Pr > |F|
1	buy_ind	Pooled	Equal	M	***	<.0001	highly	.
2	buy_ind	Satterthwaite	Unequal	M	***	<.0001	highly	.
3	var1	Pooled	Equal	-17.21	***	<.0001	highly	<.0001
4	var1	Satterthwaite	Unequal	-12.94	724	<.0001	highly	<.0001
5	var10	Pooled	Equal	2.68	***	0.0073	weak	0.1546
6	var10	Satterthwaite	Unequal	2.57	521	0.0105	weak	0.1546
7	var11	Pooled	Equal	2.00	***	0.0450	weak	0.5931
8	var11	Satterthwaite	Unequal	2.03	725	0.0423	weak	0.5931
9	var12	Pooled	Equal	1.76	***	0.0780	not at all	0.9532
10	var12	Satterthwaite	Unequal	1.76	725	0.0788	not at all	0.9532

Now, based on an arbitrary cutoff chosen for the value of the probability of the variances being equal, "probv", this code outputs only those rows where the value of "probv" matches the 'Equal' or 'Unequal' indicator. Thus, for example, if the probability of variance equality is <.0001, then the row where variances are 'Unequal' will be output.

```
data &lib1..&pre.diffsame;
  set &lib1..&pre.merged;
  if probv le .0050 and variances = 'Unequal'
    then output;
  if probv gt .0050 and variances = 'Equal'
    then output;
run;
```

The resulting data set, **mylib.test1diffsame**, will look like this.

	Variable	Method	Variances	t Value	DF	Pr > |t|	difference	Pr > |F|
1	buy_ind	Satterthwaite	Unequal	M	***	<.0001	highly	.
2	var1	Satterthwaite	Unequal	-12.94	724	<.0001	highly	<.0001
3	var10	Pooled	Equal	2.68	***	0.0073	weak	0.1546
4	var11	Pooled	Equal	2.00	***	0.0450	weak	0.5931
5	var12	Pooled	Equal	1.76	***	0.0780	not at all	0.9532
6	var13	Satterthwaite	Unequal	-0.84	726	0.4021	not at all	0.0023
7	var2	Satterthwaite	Unequal	7.86	725	<.0001	highly	<.0001

STEP FIVE

The final step merges the above file to the file with the means of the two groups. The two files going in and final output file should all have just one row for each variable. This final file will list the variable name, the mean of each by-group and the significance of the difference between those two means.

```
%let lib1 = mylib;
%let pre = test1;
proc sort data = &lib1..&pre.diffsame;
  by variable;
run;
data &lib1..&pre.ttest;
  merge &lib1..&pre.statsfinal
  &lib1..&pre.diffsame (drop =  method variances df probv);
  by variable;
run;
```

CONCLUSION

This may look like a lot of code to do a simple task, especially when compared to the four lines of code in the introduction to this paper. However, once you familiarize yourself with it, this code runs easily and will save hours of work in addition to producing an easy to use reference.

REFERENCES

SAS/STAT® User's Guide, Version 6, Fourth Edition, Volume 2 Cary, NC:SAS Institute Inc., 1989. 846 pp.