

MWSUG 2003

A Macro Program for Displaying T-test Results

Brent L. Baxter, Ph.D., Alcohol and Drug Abuse Institute, Seattle, WA

Abstract

The Student's t-test is one of the most common statistical tests used to compare the means (or proportions) of two independent groups. However, the standard SAS® output from PROC TTEST proves to be cumbersome for identifying and displaying the most meaningful results. This paper illustrates the use of a macro program as a means to highlight the key statistical output from PROC TTEST. It can also serve as a model for customizing the output of most other SAS procedures. The intermediate SAS user should come away with a better appreciation of (1) the manipulation of ODS output objects to select and assess selected output fields and (2) the use of macros to automate the entire process to facilitate repeated runs.

Standard PROC TTEST Output: Too Much of a Good Thing

Standard PROC TTEST output, while providing vital information (including results with which to judge the appropriateness of the particular t-test), does not make it easy for the user to discern such key results as the groups' means and the statistical significance of the intergroup difference of means.

For example, consider standard SAS output for a PROC TTEST (using a sample data set) comparing the proportions of drivers ticketed for reckless driving in two groups of Washington State drivers: (1) drivers ticketed by an aggressive driving apprehension team (ADAT) and (2) a control sample of drivers ticketed by other law enforcement patrols

in the same counties patrolled by ADAT (COUNTY) (see Output #1).

The output provides more information than the typical user requires. For example, seven consecutive columns display the following statistics: Lower Confidence Limit (CL) Mean, Mean, Upper CL Mean, Lower CL Standard Deviation, Standard Deviation, Upper CL Standard Deviation, and Standard Error. Typically, the user is interested only in the mean and standard deviation results.

The typical user also wants to determine the statistical significance of the intergroup difference of means. For this we need to look at the results of the F-test of the intergroup difference in distribution variances (found in the "Pr > F" column in the "Equality of Variances" section of the output). Based on the F-test results, we must then locate the results for the appropriate t-test method ("Pooled" for equal variances, "Satterthwaite" for unequal variances) to determine the correct statistical significance of the T-statistic (in the "Pr > |t|" column in the "TTests" section).

Interpreting t-test results from standard output is especially cumbersome if the output contains a long series of t-test results, which can occur if one is comparing a series of dependent variables (e.g., various types of traffic violations) or a variety of different groups.

This paper presents an alternative method, using a macro, to generate output that summarizes t-test results (for a single test or series of tests) in an easily readable format.

Two points should be made up front. First, limiting printed output to "key statistics" should not be seen as an excuse to avoid surveying the entire output to judge the appropriateness of a statistical test (e.g., are the assumptions of the test being met?). Second, while the creation of macros such as this might seem onerous, it will prove to be time well spent if you need to run the macro frequently.

Identifying the Key Output Fields

The essential statistics we typically want to draw out of the standard PROC TTEST output (see Output #1) are:

- ① The dependent variable being measured (in our example, proportion of drivers given citations for Reckless Driving);
- ② The two groups being compared (ADAT and County drivers);
- ③ The sample sizes (885 for ADAT drivers and 448 for County drivers);
- ④ The group means or proportions (2.26% for ADAT drivers and 0.89% for County drivers);
- ⑤ The group standard deviations (0.1487 for ADAT drivers and 0.0942 for County drivers);
- ⑥ The probability or significance, level results of the F-test for equality of variances ($\text{Prob}(F) < .0001$, which is sufficiently small to reject the null hypothesis that the variances are equal);
- ⑦ The appropriate method of t-test (Satterthwaite) based on the results of the F-test (which found the variances to be unequal); and
- ⑧ The probability, or significance, level of the appropriate t-test result for the intergroup difference of means/proportions ($\text{Prob}(t) = 0.413$).

Beginning the Program

```
title MWSUG 2003 Conference;
title2 Macro Program for Displaying T-test
      Results - Brent Baxter;
title3; * Blank line for spacing in title;
```

```
libname Traffic "C:\MWSUG2003";
```

Assigning Macro Variables to Represent Our Comparison Groups (Optional)

A first, though optional, step is to use macro variables to define the two groups we will be comparing in the t-test(s). I've chosen to use the macro language "%let" command to assign the names of my two groups of drivers (ADAT and County) to the macro variables "Group1" and "Group2".

```
%let group1 = ADAT;
%let group2 = County;
```

Defining these macro variables will allow me to subsequently reference, for example, my first group (ADAT), by calling the macro variable with the phrase "&group1". This makes it convenient if I later want to change my designation of comparison groups. For example, I can change my second group to a statewide sample of drivers with a single change in my program by redefining the appropriate macro variable (%let group2 = State).

Writing the Macro

The next task is to create our macro. The macro, once called, will perform a t-test on a selected type of traffic violation (or any other numeric variable) and manipulate the output to present the results in an easily readable format.

Macro Parameters

Our macro (I've named it "TT") uses macro parameters (unassigned variables listed within the parentheses in the first line of the macro). This allows us to assign

values to these parameters each time we call the macro.

```
%macro TT (item, itemlabel, class, group1,  
            group2, plevel, datafile);
```

Note that there is no need for a DATA statement prior to the macro, since the macro contains all the DATA steps we need for our task.

Macro Step #1: Perform t-test on selected variable and output the results (as ODS output objects) into separate output data sets.

The first step within the macro is to perform the t-test and output the three resulting "ODS tables" (Statistics, TTests, and Equality), containing what are called "ODS output objects," into separate output data sets (which I've named "TStats", "TTests" and "TEqual", respectively). These results (or "objects") are created and saved as sets of variables which can be referenced and manipulated essentially like any other SAS variable (see Output #2 for listings of the three output data sets). (To discover the ODS tables for whatever procedure you're running, insert the "ODS TRACE ON" command into your program just prior to the procedure.)

```
* Output #1;  
proc ttest data=&datafile;  
    class &class;  
    var &item;  
    title4 "T-test Results for &itemlabel";  
    ods output  
        Statistics = TStats  
        TTests = TTests  
        Equality = TEqual;  
run;  
title4; * To clear 4th line of title heading until re-  
        assigned;  
  
* Output #2;  
proc print data= TStats heading=h;  
    title4 "PROC PRINT of 'TStats' Output Data  
Set"; run;  
proc print data= TTests heading=h;  
    title4 "PROC PRINT of 'TTests' Output Data  
Set"; run;  
proc print data= TEqual heading=h;  
    title4 "PROC PRINT of 'TEqual' Output Data  
Set"; run;
```

```
title4;
```

Macro Step #2: Create merged output data set.

Next we create a data set ("AllStats") containing all the ODS output objects from the t-test by merging the three individual output data sets (TStats, TTests, TEqual). Since there are two different variables (objects) named "Method" (one in the "TTests" dataset, another in the "TEqual" data set), I'll first assign their values to two new variables ("TMethod" and "FMethod," respectively) to distinguish them in the merged data set (see Output #3 for a listing of the merged output data set).

```
data TTests; set TTests; TMethod = Method; run;  
data TEqual; set TEqual; FMethod = Method;  
run;
```

```
* Output #3;  
data AllStats; merge TStats TTests TEqual;  
proc print data=AllStats;  
    title4 "PROC PRINT of 'AllStats' Output  
Data Set";  
run;  
title4;
```

Note that the merged output data set (AllStats) contains three observations. The first observation contains results on Group 1 (ADAT), the Pooled t-test method (for equal variances) and the F-test. The second observation has results on Group 2 (County) and the Satterthwaite t-test method (for unequal variances). The third observation (which we won't be using) contains statistics on the difference of means/proportions.

Macro Step #3: Assign relevant results.

Next we manipulate the saved output (in the AllStats data set) to assign (in appropriate format) specific output values (e.g., mean, standard deviation) to their appropriate comparison group (ADAT, County) and assign the appropriate t-test significance result to a new variable I'll name "TStatSignif". We can also delete

the third observation in AllStats, since we won't be needing the results it contains.

```
data AllStats;
  set AllStats;
format &group1.Yes &group2.Yes 6.0
  &group1.Percent &group2.Percent 5.1
  TStatSignif PVALUE6.4;
if compress(Class) = "&group1" then do;
  &group1.N = N;
  &group1.Yes = round (N * Mean);
  &group1.Percent = Mean * 100;
  &group1.SD = StdDev;
end;
else if compress(Class) = "&group2" then do;
  &group2.N = N;
  &group2.Yes = round (N * Mean);
  &group2.Percent = Mean * 100;
  &group2.SD = StdDev;
end;
else delete;
```

Macro Step #4: Determine appropriate t-test method.

We next ascertain the appropriate t-test method (Pooled or Satterthwaite) on the basis of the F-test (difference in variance) results.

```
if ProbF lt .05 and Variances = "Unequal" then
  TStatSignif = Probt;
if ProbF ge .05 and Variances = "Equal" then
  TStatSignif = Probt;
```

Macro Step #5: Collect and retain key results from first observation.

In order to merge the appropriate t-test results into one observation for printing, we need to retain the key results (Group 1 statistics and TStatSignif) contained in the first observation in AllStats.

```
data AllStats;
  set AllStats;
if compress(Class) = "&group1" then do;
  HoldG1N = &group1.N;
  HoldG1Yes = &group1.Yes;
  HoldG1Percent = &group1.Percent;
  HoldG1SD = &group1.SD;
  HoldSST = TStatSignif;
end;
retain HoldG1N HoldG1Yes HoldG1Percent
  HoldG1SD HoldSST;
```

Macro Step #6: Combine key results into single observation.

We can then combine all appropriate t-test results in one convenient observation in a new output data set ("KeyStats"). We can do this by copying the retained results from the first observation in AllStats into the second observation and then keeping only the second observation. We then have all the t-test results we need in a single observation (see Output #4 for a listing of the observation).

```
data KeyStats;
  set AllStats;
length ViolationType $36;
ViolationType = "&Itemlabel";
if &group1.N = . then &group1.N = HoldG1N;
if &group1.Yes = . then &group1.Yes =
  HoldG1Yes;
if &group1.Percent = . then &group1.Percent =
  HoldG1Percent;
if &group1.SD = . then &group1.SD =
  HoldG1SD;
if TStatSignif = . then TStatSignif = HoldSST;
if &group2.N = . then delete;
```

```
* Output #4;
proc print data=KeyStats;
title4 PROC PRINT of Single Observation
  Containing T-test Results;
run;
title4;
```

Macro Step #7 (Optional): Create a variable that describes the level of statistical significance.

To make it easier to discern statistical significance in a series of t-test results, we can choose to construct a variable ("SignifLevel") that categorizes each test's statistical significance into an appropriate level. For example, a statistical significance of 0.0413 could be classified as a level of "p < .05." Notice that we're using the values of a particular ODS output object (TStatSignif) to do the necessary computations for us.

```
data KeyStats;
  set KeyStats;
length SignifLevel $16;
if TStatSignif ne . and TStatSignif < &plevel then
do;
```

```

SignifLevel = "p < .10 ";
if TStatSignif < .05 then SignifLevel = " * p
<.05 ";
if TStatSignif < .01 then SignifLevel = " ** p
<.01 ";
if TStatSignif < .001 then SignifLevel = "***
p <.001";
end;
else if TStatSignif ge &plevel then
SignifLevel = " NS ";
run;

```

Macro Step #8: Print key results from the t-test.

Our macro is now ready to print a customized summary of the t-test results. In this example I've chosen to print, for each group, the number of drivers violating ("Yes"), the proportion violating, and the statistical significance (value and level) of the t-statistic (intergroup difference of proportions). I could have chosen to print other statistics as well, such as each group's total number of drivers (sample size) and standard deviation.

```

* Output #5;
proc print data=KeyStats label noobs;
var ViolationType &group1.Yes
&group1.Percent &group2.Yes
&group2.Percent TStatSignif
SignifLevel;
label Variable = &itemlabel;
title4 "Selected Results from 'KeyStats'
Output Data Set";
run;
title4;

```

Macro Step #9: Collect results from multiple t-tests.

If we want to run (and summarize the results of) multiple t-tests, we can add the results for each t-test to all previous t-test results (from previous calls of the macro) in an "AllViolations" data set.

```

data AllViolations;
set AllViolations KeyStats;
run;

%mend TT; * Ends macro;

```

Calling the Macro

Having written the macro, we can now "call" the macro with a "%TT" command. Our macro call will invoke the execution of all commands within the macro text (between "%macro TT" and "%mend TT"). Executing the "TT" macro program for a t-test of Reckless Driving will produce Outputs #1 through #5.

```

%tt (item=RecklessDriving, itemlabel=Reckless
Driving, class=Group, group1=ADAT,
group2=&Group2, plevel=.10,
datafile=Traffic.Violations);

```

Repeated Macro Calls Enhance Summary Output

In addition, by calling the "TT" macro program repeatedly using different macro parameters, we can produce a data set ("AllViolations") that summarizes the key statistics for an entire series of t-tests (e.g., for different types of traffic violations). We can then print a listing from the AllViolations data set that summarizes the key results from our series of t-tests (see Output #6).

```

%tt (item=RecklessDriving, itemlabel=Reckless
Driving, class=Group, group1=&group1,
group2=&Group2, plevel=.10,
datafile=Traffic.Violations);
%tt (item=ViolHOV, itemlabel=Carpool Lane
Violation, class=Group, group1=&group1,
group2=&Group2, plevel=.10,
datafile=Traffic.Violations);
%tt (item=ViolCrossBarrier, itemlabel=Crossing
Physical Barrier, class=Group,
group1=&group1, group2=&Group2,
plevel=.10, datafile=Traffic.Violations);
%tt (item=ViolDisobeyTCD, itemlabel=Disobey
Traffic Control Device, class=Group,
group1=&group1, group2=&Group2,
plevel=.10, datafile=Traffic.Violations);

```

-
- (Additional calls for various types of violations)
-

```

%tt (item=ViolSpeed, itemlabel=Speeding,
class=Group, group1=&group1,
group2=&Group2, plevel=.10,
datafile=Traffic.Violations);

```

```
* Output #6;
proc print data=AllViolations double;
var ViolationType &group1.Yes &group1.Percent
    &Group2.Yes &Group2.Percent TStatSignif
    SignifLevel;
title4 "Summary of Results for Multiple T-tests";
run;
title4;
```

Conclusion

By comparing standard PROC TTEST output (such as in Output #1 in this paper) versus summarized output of key t-test statistics (Output #5), the SAS user will hopefully appreciate the advantages of capturing and manipulating ODS output objects to customize SAS results. Furthermore, comparing multiple instances of standard t-test output versus a customized summary of multiple t-test results (Output #6) should reveal the advantages of creating a macro to automate the process of summarizing results. The user is encouraged to try this approach with the output of other SAS procedures as well.

Acknowledgements

All examples of SAS programs and output in this paper were generated using SAS version 8.02 in a Windows® XP® environment. I would like to acknowledge my use of Art Carpenter's excellent book on macros, *Carpenter's Complete Guide to the SAS® Macro Language* (SAS Institute, Inc., 1998), which I found helpful in preparing this paper. Thanks also to the Washington State Department of Licensing for providing the raw data (from a study funded by the Washington State Traffic Safety Commission) which generated the examples used in this paper.

Brent Baxter
Alcohol and Drug Abuse Institute
University of Washington
1107 NE 45th Street, Suite 120
Seattle, WA 98105-4631 USA
Phone: 206.685.3714
Fax: 206.543.5473
Email: bax@u.washington.edu
Web: <http://depts.washington.edu/adai/>

Contact Information

Comments, suggestions and questions are welcome. Please contact the author:

OUTPUT #1: Standard PROC TTEST Output (with Highlights)

MWSUG 2003 Conference
Macro Program for Displaying T-test Results - Brent Baxter

T-test Results for Reckless Driving

The TTEST Procedure

Statistics

Variable ①	Group ②	N ③	Lower CL Mean	Mean ④	Upper CL Mean	Lower CL Std Dev	Std Dev ⑤	Upper CL Std Dev	Std Err	Minimum	Maximum
RecklessDriving	ADAT	885	0.0128	0.0226	0.0324	0.1421	0.1487	0.156	0.005	0	1
RecklessDriving	County	448	0.0002	0.0089	0.0177	0.0884	0.0942	0.1008	0.0044	0	1
RecklessDriving	Diff (1-2)		-0.001	0.0137	0.0288	0.128	0.1329	0.1382	0.0077		

T-Tests

Variable	Method ⑦	Variances	DF	t Value	Pr > t ⑧
RecklessDriving	Pooled	Equal	1331	1.77	0.0763
RecklessDriving	Satterthwaite	Unequal	1267	2.04	0.0413

Equality of Variances

Variable	Method	Num DF	Den DF	F Value	Pr > F ⑥
RecklessDriving	Folded F	884	447	2.49	<.0001

OUTPUT #2: Contents of Separate Output Data Sets Containing ODS Output Objects

MWSUG 2003 Conference
Macro Program for Displaying T-test Results - Brent Baxter

PROC PRINT of 'TStats' Output Data Set

Obs	Variable	Class	N	Lower CLMean	Mean	Upper CLMean	Lower CLStdDev
1	RecklessDriving	ADAT	885	0.0128	0.0226	0.0324	0.1421
2	RecklessDriving	County	448	0.0002	0.0089	0.0177	0.0884
3	RecklessDriving	Diff (1-2)	—	-0.001	0.0137	0.0288	0.128

Obs	UMPULower CLStdDev	StdDev	UMPUUpper CLStdDev	Upper CLStdDev	StdErr	Minimum	Maximum
1	0.142	0.1487	0.1559	0.156	0.005	0	1
2	0.0883	0.0942	0.1007	0.1008	0.0044	0	1
3	0.128	0.1329	0.1381	0.1382	0.0077	—	—

PROC PRINT of 'TTests' Output Data Set

Obs	Variable	Method	Variances	tvalue	DF	Probt
1	RecklessDriving	Pooled	Equal	1.77	1331	0.0763
2	RecklessDriving	Satterthwaite	Unequal	2.04	1267	0.0413

PROC PRINT of 'TEqual' Output Data Set

Obs	Variable	Method	NumDF	DenDF	FValue	ProbF
1	RecklessDriving	Folded F	884	447	2.49	<.0001

OUTPUT #3: Contents of Merged Output Data Set

MWSUG 2003 Conference
Macro Program for Displaying T-test Results - Brent Baxter

PROC PRINT of 'AllStats' Output Data Set

Obs	Variable	Class	Lower N	Lower CLMean	Upper Mean	Upper CLMean	Lower CLStdDev	UMPULower CLStdDev	StdDev	UMPUUpper CLStdDev	Upper CLStdDev	StdErr	Minimum
1	RecklessDriving	ADAT	885	0.0128	0.0226	0.0324	0.1421	0.142	0.1487	0.1559	0.156	0.005	0
2	RecklessDriving	County	448	0.0002	0.0089	0.0177	0.0884	0.0883	0.0942	0.1007	0.1008	0.0044	0
3	RecklessDriving	Diff (1-2)	-	-0.001	0.0137	0.0288	0.128	0.128	0.1329	0.1381	0.1382	0.0077	-

Obs	Maximum	Method	Variances	tValue	DF	Probt	TMethod	NumDF	DenDF	FValue	ProbF	FMethod
1	1	Folded F	Equal	1.77	1331	0.0763	Pooled	884	447	2.49	<.0001	Folded F
2	1	Satterthwaite	Unequal	2.04	1267	0.0413	Satterthwaite	.	.	.	.	.
3	-			.	.	.		.	.	.	.	

OUTPUT #4: Contents of Key Statistics Merged Into a Single Observation

MWSUG 2003 Conference
Macro Program for Displaying T-test Results - Brent Baxter

PROC PRINT of Single Observation Containing T-test Results

Obs	Variable	Class	N	Lower CLMean	Mean	Upper CLMean	Lower CLStdDev	UMPULower CLStdDev	StdDev	UMPUUpper CLStdDev
1	RecklessDriving	County	448	0.0002	0.0089	0.0177	0.0884	0.0883	0.0942	0.1007

Obs	Upper CLStdDev	StdErr	Minimum	Maximum	Method	Variances	tValue	DF	Probt	TMethod
1	0.1008	0.0044	0	1	Satterthwaite	Unequal	2.04	1267	0.0413	Satterthwaite

Obs	NumDF	DenDF	FValue	ProbF	FMethod	ADATYes	County Yes	ADATPercent	County Percent	TStat Signif	ADATN	ADATSD
1	.	.	.	.		20	4	2.3	0.9	0.0413	885	0.14870

Obs	County N	CountySD	Hold G1N	Hold G1Yes	Hold G1Percent	Hold G1SD	Hold SST	violationType
1	448	0.094174	885	20	2.25989	0.14870	.	Reckless Driving

OUTPUT #5: Output of Selected T-test Results Generated from Macro Call

MWSUG 2003 Conference
Macro Program for Displaying T-test Results - Brent Baxter

Selected Results from 'KeyStats' Output Data Set

ViolationType	ADATYes	ADATPercent	County Yes	County Percent	Stat SignifT	Signif Level
Reckless Driving	20	2.3	4	0.9	0.0413	* p <.05

OUTPUT #6: Output of Selected T-test Results Generated from Repeated Macro Calls

MWSUG 2003 Conference
Macro Program for Displaying T-test Results - Brent Baxter

Summary of Results for Multiple T-tests

Obs	ViolationType	ADATYes	ADATPercent	County Yes	County Percent	Stat SignifT	Signif Level
1	Reckless Driving	20	2.3	4	0.9	0.0413	* p <.05
2	Carpool Lane Violation	142	16.0	3	0.7	<.0001	*** p <.001
3	Crossing Physical Barrier	10	1.1	1	0.2	0.0309	* p <.05
4	Disobey Traffic Control Device	7	0.8	21	4.7	0.0002	*** p <.001
5	Failure to Stop	5	0.6	20	4.5	0.0001	*** p <.001
6	Failure to Use Turn Signal	89	10.1	7	1.6	<.0001	*** p <.001
7	Following Too Closely	129	14.6	8	1.8	<.0001	*** p <.001
8	Illegal Entry Limited Access Hwy	10	1.1	0	0.0	0.0015	** p <.01
9	Lane Violation	97	11.0	5	1.1	<.0001	*** p <.001
10	Negligent Driving	113	12.8	9	2.0	<.0001	*** p <.001
11	Passing Violation	4	0.5	5	1.1	0.2241	NS
12	Speeding	564	63.7	191	42.6	<.0001	*** p <.001