

Adding (p-) Value to Tables – a Programmer’s Perspective

Kathryn Wright, Oxford Pharmaceutical Sciences

INTRODUCTION

Many programmers are petrified by the requirement to put a p-value onto a table. In reality, it is not as difficult as many people think. The SAS code required is fairly simple and it is a case of matching the correct procedure with the correct type of p-value. The aim of this paper is to show some methods that can be used.

Examples of SAS code that will produce these p-values will be shown and the output produced by the code will be displayed, indicating the data of interest.

Different methods of storing the output values as SAS datasets will be explored including Output Delivery System (ODS).

ONE SAMPLE T-TEST AND TWO SAMPLE T-TEST

“The one-sample t-test is used to infer whether an unknown population mean differs from a hypothesized value. The test is based on a single sample of ‘n’ measurements from the population” (Walker, 2002)

“The two-sample t-test is used to compare the means of two independent populations.” (Walker, 2002)

The data that will be used to demonstrate a one-sample t-test and a two sample t-test is:

Subject	Sex	SBP	DBP
1	M	120	80
2	F	140	91
3	F	118	79
4	M	150	98
5	M	123	84

PROC TTEST

PROC TTEST can be used to produce p-values for one and two sample t-tests. The variable that is to be analysed is specified in the VAR statement.

H0 is used to specify the value of the analysis variable to be considered normal. The default value is 0. This code specifies 120 as the normal value for systolic blood pressure.

```
proc ttest H0=120 data=vitals;
  var sbp;
run;
```

This is the output that would be created when the code above is run on the data shown above. It is an example of a one sample t-test.

The TTEST Procedure										
Statistics										
Variable	N	Lower CL		Upper CL	Lower CL		Upper CL		Minimum	Maximum
		Mean	Mean	Mean	Std Dev	Std Dev	Std Dev	Std Err		
sbp	5	112.72	130.2	147.68	8.4348	14.078	40.455	6.296	118	150
T-Tests										
Variable	DF	t Value	Pr > t							
sbp	4	1.62	0.1805							

In order to produce a two-sample t-test the class statement needs to be added to the code. The class statement specifies the variable that the results are to be grouped by. In this example the results are split by sex.

```
proc ttest H0=120 data=vitals2;
  class sex;
  var sbp;
run;
```

PhUSE 2006

This code for a two sample t-test would produce the following output.

The TTEST Procedure									
Statistics									
Variable	sex	N	Mean	Lower CL	Upper CL	Lower CL	Upper CL	Lower CL	Upper CL
				Mean	Mean	Std Dev	Std Dev	Std Dev	Std Err
sbp	F	2	-10.77	129	268.77	6.9405	15.556	496.41	11
sbp	M	3	89.955	131	172.04	8.6027	16.523	103.84	9.5394
sbp	Diff	(1-2)	-49.08	-2	45.084	9.1811	16.207	60.429	14.795

T-Tests						
Variable	Method	Variiances	DF	t Value	Pr > t	
sbp	Pooled	Equal	3	-8.25	0.0037	
sbp	Satterthwaite	Unequal	2.39	-8.38	0.0079	

Equality of Variances						
Variable	Method	Num DF	Den DF	F Value	Pr > F	
sbp	Folded F	2	1	1.13	1.0000	

The output from PROC TTEST can be saved into a SAS dataset using the Output Delivery System (ODS). This requires only a small amendment to the code shown above. Here is the example of the one sample t-test.

```
ods output ttests=ttest;

proc ttest H0=120 data=vitals;
  var sbp;
run;
```

On the line that states 'ods output ...' the text entered before the '=' is the name of the table that SAS produces from PROC TTEST. In this case it is 'ttests'. The text entered after the '=' is the name of the dataset to be saved. In this case a dataset called ttest will be saved into the work directory.

The dataset that would be saved from the one sample t-test would look like:

Variable	tvalue	DF	Probt
sbp	1.62	4	0.1805

The dataset that would be saved from the two sample t-test would look like:

Variable	Method	Variiances	tvalue	DF	Probt
sbp	Pooled	Equal	-8.25	3	0.0037
sbp	Satterthwaite	Unequal	-8.38	2.39	0.0079

When looking at the output displayed above for the one sample t-test the p-value is shown under the text 'Pr > |t|'. In the dataset the p-value is stored in the variable called PROBT.

With a two-sample t-test there are results two results produced, one assuming equal variance and the other assuming unequal variance. This example will assume equal variance. The assumption of equal variance can be tested using the F-test. This will not be explored in this paper.

PROC UNIVARIATE

The SAS code required to produce the one sample t-test using PROC UNIVARIATE is also very simple. The variable that is to be analysed is specified in the var statement as was shown in the PROC TTEST. PROC UNIVARIATE uses Mu0 in place of H0 in PROC TTEST.

```
proc univariate mu0=120 data=vitals;
```

PhUSE 2006

```
var sbp;
run;
```

The UNIVARIATE Procedure			
Variable: sbp			
Moments			
N	5	Sum Weights	5
Mean	130.2	Sum Observations	651
Std Deviation	14.0783522	Variance	198.2
Skewness	0.81431246	Kurtosis	-1.6810171
Uncorrected SS	85553	Corrected SS	792.8
Coeff Variation	10.8128665	Std Error Mean	6.2960305
Basic Statistical Measures			
Location		Variability	
Mean	130.2000	Std Deviation	14.07835
Median	123.0000	Variance	198.20000
Mode	.	Range	32.00000
		Interquartile Range	20.00000
Tests for Location: Mu0=120			
Test	-Statistic-	-----p Value-----	
Student's t	t 1.620068	Pr > t	0.1805
Sign	M 1	Pr >= M	0.6250
Signed Rank	S 4	Pr >= S	0.2500
Quantiles (Definition 5)			
Quantile	Estimate		
100% Max	150		
99%	150		
95%	150		
90%	150		
75% Q3	140		
50% Median	123		
25% Q1	120		
10%	118		
5%	118		
1%	118		
0% Min	118		
Extreme Observations			
----Lowest----		----Highest---	
Value	Obs	Value	Obs
118	3	118	3
120	1	120	1
123	5	123	5
140	2	140	2
150	4	150	4

In order to use PROC UNIVARIATE to perform a two sample t-test it is necessary to perform some pre-processing. This can require a greater statistical understanding of the problem. Therefore PROC TTEST is the recommended method to produce a two sample t-test.

PhUSE 2006

There are two methods that can be used to output the results from proc univariate into a SAS dataset. The first method uses the output statement. To use the output statement the names of each of the tests to be stored are required. In this case, only the one sample t-test (probt) and the number of observations (n) have been chosen to be saved to a dataset called ttests3.

```
proc univariate mu0=120 data=vitals noprint;
  var sbp;
  output out = ttest3 probt=probt n=n;
run;
```

An example of the dataset that would be produced by the code above is:

n	probt
5	0.1805324052

The second method that can be used to output the result from PROC UNIVARIATE is ODS. To use ODS only an extra row above the procedure is required. ODS produces outputs called tables. In order to output the one sample t-test results the table called 'testsforslocation' are saved to a dataset called ttests4. How to decide which tables to output will be explored later.

```
ods output testsforlocation=ttest4;

proc univariate mu0=120 data=vitals;
  var sbp;
run;
```

The TESTFORLOCATIONS table produced by ODS outputs more results than those required for this test. The table shown below indicates what the dataset produced by the code above would produce.

Varname	Test	Testlab	Stat	pType	pValue	Mu0
sbp	Student's t	t	0.620068	Pr > t	0.1805	120
sbp	Sign	M	1	Pr>= M	0.6250	120
sbp	Signed Rank	S	4	Pr>= S	0.2500	120

The one sample t-test result is found in the pValue column when pType = 'Pr >|t|'.

PROC MEANS

PROC MEANS cannot be used to produce the same outputs produced by PROC TTEST and PROC UNIVARIATE above. PROC MEANS does not have the functionality to set a value of normal other than 0. Therefore PROC MEANS is not suitable to use in the previous example but it can be used where the normal value is expected to be 0. For example the change from baseline of systolic blood pressure.

The data that will be used in these one-sample t-test examples is:

Subject	SBPBase	SBPFinal	Change
1	120	121	1
2	140	135	-5
3	118	118	0
4	150	145	5
5	123	120	-3

As with the previous examples the variable to be analysed is specified in the var statement. The tests required to be performed need to be specified as options on the PROC MEANS line. In this example probt specifies that the t-test has to be performed and n indicates that the number of observations should be counted.

```
proc means data=vitals3 n probt;
  var change;
run;
```

The MEANS Procedure	
Analysis Variable : change	
N	Pr > t
_____	_____

PhUSE 2006

5	0.8276
—————	

As with PROC UNIVARIATE a greater statistical understanding of the problem is required to ensure the correct value is calculated. Therefore PROC TTEST is the recommended method to produce a two sample t-test.

There are two methods that can be used to output the results from PROC MEANS into a SAS dataset. The first method uses the output statement as described for PROC UNIVARIATE above.

```
proc means data=vitals3 n probt;
  var change;
  output out =ttest5 n=n probt=probt;
run;
```

The output dataset produced using this method is:

TYPE	_FREQ_	n	probt
0	5	5	0.8275647196

The second method that can be used to output the result from PROC MEANS is ODS. Again, this is used in the same way as described above for PROC UNIVARIATE.

```
ods output summary=ttest6;

proc means data=vitals3 n probt;
  var change;
run;
```

The output dataset produced using this method is:

n	probt
5	0.8276

The result of interest from the output is seen next to the label 'P >|t|'. Again, the variable called PROBT contains the p-value.

ANOVA

"One-way ANOVA (analysis of variance) is used to simultaneously compare two or more group means based on independent samples from each group." (Walker, 2002)

In clinical trials ANOVA is often used to compare the mean responses between two parallel groups. In the examples shown here the heart rate will be compared between subjects in the placebo group and subjects in the drug group.

The data that will be used to produce the ANOVA examples is:

Subject	HR	Drug
1	60	Placebo
2	90	Drug
3	75	Placebo
4	110	Drug
5	93	Placebo

PROC GLM

PROC GLM can be used to produce ANOVA. The key to using PROC GLM correctly is specifying the MODEL statement. The MODEL statement has the format *dependants=independents*. In this case HR is the dependant and DRUG is the independent. The CLASS statement must also be completed. The CLASS statement names the classification variables to be used in the model. Typical class variables are TREATMENT, SEX, RACE, GROUP, and REPLICATION. If you specify the CLASS statement, it must appear before the MODEL statement.

```
Proc glm data=anova1;
  class drug;
  model hr=drug;
run;
```

PhUSE 2006

This code would produce the following output:

The GLM Procedure						
Class Level Information						
Class	Levels	Values				
drug	2	Drug Placebo				
Number of observations						5
The GLM Procedure						
Dependent Variable: hr						
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F	
Model	1	691.200000	691.200000	2.78	0.1941	
Error	3	746.000000	248.666667			
Corrected Total	4	1437.200000				
		R-Square	Coeff Var	Root MSE	hr Mean	
		0.480935	18.42193	15.76917	85.60000	
Source	DF	Type I SS	Mean Square	F Value	Pr > F	
drug	1	691.2000000	691.2000000	2.78	0.1941	
Source	DF	Type III SS	Mean Square	F Value	Pr > F	
drug	1	691.2000000	691.2000000	2.78	0.1941	

There are two methods that can be used to output the results from PROC GLM into a SAS dataset. The first method uses the OUTSTAT statement. The OUTSTAT statement is similar to the OUTPUT statement described previously for PROC MEANS and PROC UNIVARIATE.

```
Proc glm data=anova1 outstat = anovaout1;
  class drug;
  model hr=drug;
run;
quit;
```

This code would produce a dataset called ANOVAOUT1 that would look like:

<u>NAME</u>	<u>SOURCE</u>	<u>TYPE</u>	<u>DF</u>	<u>SS</u>	<u>F</u>	<u>PROB</u>
hr	ERROR	ERROR	3	746	.	.
hr	drug	SS1	1	691.2	2.7796246649	0.19410615915
hr	drug	SS3	1	691.2	2.7796246649	0.19410615915

The second method that can be used to save the output from PROC GLM to a dataset is ODS. The ODS OUTPUT statement is required as described above. In this case the table required to be output is MODELANOVA.

```
ods output modelanova=anovout2;

Proc glm data=anova1 ;
  class drug;
  model hr=drug;
run;
quit;
```

This ODS step would create the following dataset called ANOVAOUT2:

<u>Dependant</u>	<u>HypothesisType</u>	<u>Source</u>	<u>DF</u>	<u>SS</u>	<u>MS</u>	<u>FValue</u>	<u>probf</u>
hr	1	drug	1	691.2000000	691.2000000	2.78	0.1941

PhUSE 2006

hr	3	drug	1	691.2000000	691.2000000	2.78	0.1941
----	---	------	---	-------------	-------------	------	--------

The value of interest from the output is 'Pr > F'. In the dataset produced by the OUTSTAT statement this is found in the variable called PROB. The dataset produced using ODS stores this information in the variable called PROBF.

PROC MIXED

PROC MIXED is another procedure that can be used to calculate ANOVA. The code required requires the MODEL statement and the CLASS statement in the same way as PROC GLM.

```
proc mixed data=anova1;
  class drug;
  model hr = drug;
run;
```

The output produced by this code:

The Mixed Procedure		
Model Information		
Data Set		WORK.ANOVA1
Dependent Variable		hr
Covariance Structure		Diagonal
Estimation Method		REML
Residual Variance Method		Profile
Fixed Effects SE Method		Model-Based
Degrees of Freedom Method		Residual
Class Level Information		
Class	Levels	Values
drug	2	Drug Placebo
Dimensions		
Covariance Parameters		1
Columns in X		3
Columns in Z		0
Subjects		1
Max Obs Per Subject		5
Observations Used		5
Observations Not Used		0
Total Observations		5
Covariance Parameter Estimates		
Cov Parm		Estimate
Residual		248.67
Fit Statistics		
-2 Res Log Likelihood		26.9
AIC (smaller is better)		28.9
AICC (smaller is better)		32.9
BIC (smaller is better)		28.0
The Mixed Procedure		

PhUSE 2006

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
drug	1	3	2.78	0.1941

The easiest method to output results from PROC MIXED into a SAS dataset is to use ODS.

```
ods output tests3=anovout3;

proc mixed data=anova1;
  class drug;
  model hr = drug;
run;
```

The code would produce a dataset called ANOVAOUT3:

Effect	NumDF	DenDF	FValue	probf
hr	1	3	2.78	0.1941

The value of interest from the output is 'Pr > F'. The dataset produced using ODS stores this information in the variable called PROBF.

OUTPUT DELIVERY SYSTEM (ODS)

The OUTPUT DELIVERY SYSTEM (ODS) has been used in many of the examples shown above. ODS is a method of delivering output in a variety of formats and of making the formatted output easy to access. ODS is a very powerful tool with many uses. This paper only looks at the OUTPUT statements.

Using the code following code specifies that the output created by the procedure that is run after this statement will be output using ODS.

```
ods output data-set-definition(s);
```

In the examples shown above the *DATA-SET-DEFINITION(S)* have been in the format TABLE = OUTPUT DATASET. When ODS is used all the outputs are stored in tables. In order to determine which tables are created the ODS TRACE statement can be used.

```
ODS trace on;
```

If this is statement is run before this code:

```
proc ttest H0=120 data=vitals;
  var sbp;
run;
```

The following will be displayed in the log:

```
NOTE: Copyright (c) 1999-2001 by SAS Institute Inc., Cary, NC, USA.
NOTE: SAS (r) Proprietary Software Release 8.2 (TS2M0)
      Licensed to OXFORD PHARMACEUTICAL SCIENCES LIMITED, Site 0090882001.
NOTE: This session is executing on the WIN_PRO platform.

NOTE: SAS initialization used:
      real time          3.45 seconds
      cpu time           0.65 seconds

1  data vitals;
2  input subject sbp dbp;
3  datalines;

NOTE: The data set WORK.VITALS has 5 observations and 3 variables.
NOTE: DATA statement used:
      real time          0.15 seconds
      cpu time           0.00 seconds
```

PhUSE 2006

```
9      ;
10     run;
11
12     ods trace on;
13     proc ttest H0=120 data=vitals;
14         var sbp;
15     run;
```

Output Added:

```
-----
Name:      Statistics
Label:     Statistics
Template:  Stat.TTest.Statistics
Path:     Ttest.Statistics
-----
```

Output Added:

```
-----
Name:      TTests
Label:     T-Tests
Template:  Stat.TTest.TTests
Path:     Ttest.TTests
-----
```

NOTE: There were 5 observations read from the data set WORK.VITALS.

NOTE: PROCEDURE TTEST used:

```
real time      0.28 seconds
cpu time       0.01 seconds
```

The example of this code used previously specified the TTESTS table was the output to be saved.

The ODS TRACE will continue to produce this information in the log until the function is switched off. This can be done using the following code:

```
ODS trace off;
```

CONCLUSION

SAS Programmers are often required to produce p-values for clinical report outputs. There are a variety of methods that can be utilised to produce one sample t-test, two sample t-tests and ANOVA.

One sample t-tests can be produced using PROC TTEST, PROC UNIVARIATE or PROC MEANS. None of these procedures require complicated options or a particular understanding of the statistic being produced. However, PROC MEANS has the limitation that it can only be used where the normal value is expected to be 0.

Two sample t-tests can be produced using all the procedures suggested for the one sample t-test however, PROC TTEST would be the procedure to be recommended to any SAS programmer who is not particularly confident about their statistics knowledge. Both PROC MEANS and PROC UNIVARIATE require the data to be pre-processed in a manner which may confuse some programmers therefore the safer option is PROC TTEST.

ANOVA can be produced using PROC GLM or PROC MIXED. Both procedures only require basic code to produce a one-way ANOVA as shown above.

The output of the required statistic into a SAS dataset always used to be the hardest part of producing the value. However, with the introduction of ODS this has become much simpler. ODS can be recommended as the method of choice for saving any of the statistics required into SAS datasets.

REFERENCES

Walker, Glenn A. (2002) "Common Statistical Methods for Clinical Research with SAS Examples"

SAS Online Help <http://v8doc.sas.com/sashtml/>

CONTACT INFORMATION

PhUSE 2006

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

Author Name	Kathryn Wright
Company	Oxford Pharmaceutical Sciences
Address	Lancaster House, Kingston Business Park, Kingston Bagpuize
City / Postcode	Oxon, OX13 5FE
Work Phone:	+44 (0)1865 823815
Fax:	+44 (0)8704 580729
Email:	Kathryn.wright@ops-web.com
Web:	www.ops-web.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.