A Macro to Output the Statistics from the TTEST Procedure to a SAS® Data Set

Erik S. Larsen, Price Waterhouse LLP, Washington, DC
Daniel M. DiPrimeo, Wyeth-Ayerst Research, Radnor, PA

Abstract

The SAS System provides the statistician with a useful tool in determining if there is a difference between two population means. The procedure, PROC TTEST, can perform the comparison with or without the assumption of equal variances or equal sample sizes. A drawback of PROC TTEST is that you can output the statistics only to hard copy and not to a SAS data set. Often, it may be necessary for the statistician to output these values in a form other than that which SAS provides. The macro TTEST will output the statistics computed by PROC TTEST to a SAS data set. There are other statistics used in computations which are output as well, such as tests for equal variances with different sample sizes and p-values. The macro also has the capability to process the t-test by classification variables. Rather than using PROC PRINTTO and reading data off the output, the macro will perform all the calculations necessary for computing the statistics, along with writing the variables to a SAS data set which can be used to make customized reports with DATA _NULL_ or PROC REPORT.

The macro TTEST has the following arguments: SORTV, DATAIN, DATAOUT, CLASSV, and ANALYV. &SORTV is (are) the variable(s) by which the macro will execute the t-test (e.g. the BY variables). It is not necessary to have the data sorted by these variables, because PROC SORT is called in the macro. If later merging is desired however, it may be convenient to have the data already sorted. &DATAIN is the data set which contains the data to be analyzed and &DATAOUT is the data set which will contain the statistics from the t-test. &CLASSV is the qualitative variable which defines the two groups, or classes, in the data set. Finally, &ANALYV is the analysis variable on which the t-test is performed (e.g. the VAR statement in PROC TTEST).

The data are sorted and then summarized using PROC SUMMARY. The SUMMARY procedure is executed by &CLASSV and &SORTV, and the statistics are computed on the variable &ANALYV. Statistics requested from the output statement are n, mean, standard deviation, standard error, minimum, and maximum. These statistics are written to the data set &DATAOUT, the output data set specified by the user.

PROC FREQ is used to output data set CLASSOUT, which contains 2 records that correspond to the two populations that you are testing for equality. Since there are two means that you are testing, there are two observations in the data set. These are needed for identification purposes later in the macro. Each record is written to a separate data set, CLASS1 or CLASS2, in the macro MAKEZ. Inside this same macro, the data set Z1 is created by merging CLASS1 with &DATAOUT, and Z2 is created by merging CLASS2 with &DATAOUT.

Data set &DATAOUT is redefined by the merging by &SORTV of the two data sets, Z1 and Z2, created above. All statistics needed for the test are calculated, retained and labeled in this data step. *w1* and *w2* are the sample variances of each population divided by the respective sample sizes. These are needed for both Satterthwaite's T and Cochran-Cox T.

sdbar is defined as the square root of the sum of the weighted variances *w1* and *w2*; this is needed for the calculation of the t-statistic given that the variances of the two populations are unequal. *t1* is the probability of obtaining a larger t-statistic for population 1 (p-value); *t2* is the probability of obtaining a larger t-statistic for population 2. These are used for calculating Cochran-Cox T.

probt is the probability of a greater absolute value of t under the null hypothesis that the means of the two groups are equal, and under the assumption that the variances of the two groups are equal. The t-statistic is defined by *tequal*. Two (2) measures for testing the hypothesis that the means are equal given that the variances are not the same are also provided. The first is Satterthwaite's T, *tunequal*, where the statistic is defined as the difference in the means divided by *sdbar*, as defined above. Associated with this t-statistic are the statistics *sattdf*, Satterthwaite's degrees of freedom, and *probtsw*, its p-value. The other statistic for testing the hypothesis given that the variances are not equal is the Cochran-Cox T. In the case of equal sample sizes, the degrees of freedom of Cochran-Cox, *probdf*, is n1-1, and the p-value of the t-statistic, *probcoch*, is equal to

2*(1-probt(abs(tunequal),n1-1)).

In the case of unequal sample sizes, the degrees of freedom are assigned a missing value, and the p-value *probcoch* is a weighted average of *t1* and *t2*, discussed above.

Finally, the F-statistic for testing the hypothesis that $\sigma^2_1=\sigma^2_2$ (variances are equal) is defined as the max(*s1*,*s2*)/min(*s1*,*s2*), where *s1* and *s2* are the sample variances of each group. *probf* is the probability of a greater F value, and it is a two-tailed significance probability.

As a final step, all variables are appropriately labeled. Then, PROC DATASETS is executed to delete the temporary data sets in the macro, and the macro is ended.

Conclusion

Macro TTEST provides the statistician with all of the tools that PROC TTEST calculates, but goes a step further to output the various statistics and p-values to a SAS data set. This feature is useful when customized reports are desired, such as ones created by DATA _NULL_ or PROC REPORT. The macro saves the statistician the work of writing the output from PROC TTEST to an ASCII file (using PROC PRINTTO) and having to read the data back in by using a dataset with an INFILE statement. Thus allows you to write a report and not be concerned with programming complex input schemes to locate the values (e.g. statistics, p-values) which you desire.

Acknowledgements

The authors would like to thank Mike Rossi for his statistical and programming advice.

References

Cochran,W.G., and Cox, G. M. (1957), *Experimental Design*. New York: John Wiley & Sons, Inc.

Conover, W. J., Johnson, M. E., and Johnson, M. M. (1981), "A Comparative Study of Tests for Homogeneity of Variances, with Applications to the Outer Continental Shelf Bidding Data," *Technometrics*, 23, 351-361.

SAS Institute, Inc. (1990), *SAS/STAT User's Guide: Fourth Edition, Version 6*. Cary, NC: SAS Institute, Inc.

Author Contact Information

Wyeth Ayerst Research
145 (B-2) King of Prussia Road
Radnor, PA 19087

Erik S. Larsen at: (610) 341-2067
LARSENE@ns4.wp.wyeth.com

Daniel M. DiPrimeo at: (610) 341-2068
DIPRIMED@ns4.wp.wyeth.com

```
%MACRO TTEST(SORTV,     /* SORTV   = The BY variable          */
             DATAIN,    /* DATAIN = SAS dataset to be analyzed*/
             DATAOUT,   /* DATAOUT = Output dataset for stats*/
             CLASSV,    /* CLASSV  = Any class var for test   */
             ANALYV);   /* ANALYV  = Analysis var for test    */


PROC SORT DATA=&DATAIN;    /* Sort data here by class vars.  Will */
     BY &CLASSV &SORTV;    /* use to identify statistics later.   */


PROC SUMMARY NWAY DATA=&DATAIN;   /* Create output dataset of */
     BY &CLASSV &SORTV;           /* mean, number of observations, */
     VAR &ANALYV;                 /* standard deviation, standard  */
     OUTPUT OUT=&DATAOUT          /* error, min and max by */
     MEAN=MEAN N=N STD=STD     /* the classification variables. */
     STDERR=SE MIN=MIN MAX=MAX;


PROC FREQ DATA=&DATAOUT NOPRINT;  /* Creates datasets for   */
     TABLES &CLASSV               /* summary stats for use  */
          / OUT=CLASSOUT;         /* later in the program.  */


%MACRO MAKEZ(ARG);      /* Macro to create a dataset with one ob- */
                        /* servation for each classification      */
                        /* datasets.  ARG is always 1 or 2.       */

DATA CLASS&ARG(KEEP=&CLASSV);
     SET CLASSOUT;
     IF _N_ = &ARG;


DATA Z&ARG;                          /* This dataset contains summary */
     MERGE CLASS&ARG(IN=ONE)  /* statistics based on the 2 class */
           &DATAOUT(IN=TWO); /* vars.  These are referenced by */
     BY &CLASSV;              /* the variable ARG (for 1 and 2) */
     IF ONE & TWO;

PROC SORT DATA=Z&ARG;          /* Sort the dataset by the sorthy variable */
     BY &SORTV;

%MEND MAKEZ;


%MAKEZ(1);          /* Macro call to create first class group   */
%MAXEZ(2);          /* Macro call to create second class group  */


DATA &DATAOUT(DROP=_TYPE_ _FREQ_);          /* Now merge the data */
     MERGE                                   /* with the stats and */
       Z1(RENAME=(N=N1 MEAN=M1 STD=STD1   /* perform all stat- */
                  SE=SE1 MIN=MIN1         /* istical tests by calcu-*/
                  MAX=MAX1 &CLASSV=IND1))      /* lating test*/
       Z2(RENAME=(N=N2 MEAN=M2 STD=STD2    /* statistics for each*/
                  SE=SE2 MIN=MIN2          /* corresponding test.*/
                  MAX=MAX2 &CLASSV=IND2));
     BY &SORTV;

     S1        = STD1 * STD1;                    /* Variance(x)    */
     S2        = STD2 * STD2;                    /* Variance(y)    */
     S2POOL    = ((N1-1)*S1+(N2-1)*S2)/(N1+N2-2);    /* Pooled variance */
     TEQUAL    = (M1-M2)/SQRT(S2POOL*(1/N1+1/N2));   /* t, equal var*/
     W1        = S1/N1;                          /* Weighted Var(x)*/
     W2        = S2/N2;                          /* Weighted Var(y)*/
     SDBAR     = SQRT(W1 + W2);                  /* Sd statistic  */
     TUNEQUAL  = (M1 - M2)/SDBAR;                /* t for unequal vars. */
     SATTDF    = (W1*W1 + 2*W1*W2 + W2*W2) /     /* Satterthwaite's */
                 (W1*W1/(N1-1) + W2*W2/(N2-1));  /*degr of freedom*/
     T1        = 2*(1-PROBT(ABS(TUNEQUAL), N1-1)); /* t1 & t2 for Cox&*/
     T2        = 2*(1-PROBT(ABS(TUNEQUAL), N2-1));  /* Cochran t stats*/
     F         = MAX(S1, S2) / MIN(S1, S2);      /* F for equal vars*/

     IF (S1 > S2) THEN
       PROBF   = 2*(1-PROBF(F,N1-1,N2-1));       /* 2 sided prob for*/
     ELSE                                        /* equal vars     */
       PROBF   = 2*(1-PROBF(F,N2-1,N1-1));
     PROBT     = 2*(1-PROBT(ABS(TEQUAL),N1+N2-2));
                                                 /* equal means,vars*/
     PROBTSW= 2*(1-PROBT(ABS(TUNEQUAL),SATTDF));
                                          /* equal means, unequal vars */
IF N1=N2 THEN DO;                         /*  (Satterthwaite)*/
   COCHDF = N1-1;                         /* Cochran's df   */
   PROBCOCH = 2*(1-PROBT(ABS(TUNEQUAL),N1-1));
END;                        /* equal means, unequal vars, equal sample size */
                            /*  (Cochran)    */
ELSE DO;
   COCHDF=.;          /* Missing df if Cochran has unequal sample size */
   PROBCOCH = (T1*W1 + T2*W2) / (W1 + W2);
END;


LABEL N1       = 'NUMBER OBS IN GRP 1'
      N2       = 'NUMBER OBS IN GRP 2'
      M1       = 'MEAN OF GRP 1'
      M2       = 'MEAN OF GRP 2'
      STD1     = 'STD DEVIATION OF GRP 1'
      STD2     = 'STD DEVIATION OF GRP 2'
      SE1      = 'STD ERROR OF GRP 1'
      SE2      = 'STD ERROR OF GRP 2'
      MIN1     = 'MINIMUM OF GRP 1'
      MAX1     = 'MAXIMUM OF GRP 1'
      MIN2     = 'MINIMUM OF GRP 1'
      MAX2     = 'MAXIMUM OF GRP 2'
      S1       = 'VARIANCE OF GRP 1'
      S2       = 'VARIANCE OF GRP 2'
      S2POOL   = 'POOLED VARIANCE OF TWO GROUPS'
      TEQUAL   = 'T-STAT BASED ON EQUAL VARIANCES'
      W1       = 'S1/N1'
      W2       = 'S2/N2'
      TUNEQUAL= 'T-STAT BASED ON UNEQUAL VARIANCES'
      SATTDF   = 'SATTERTHWAITE DEGREES OF FREEDOM'
      F        = 'F-STAT FOR TESTING UNEQUAL VARIANCES'
      T1       = 'T-STAT FOR SAMPLE 1'
      T2       = 'T-STAT FOR SAMPLE 2'
      IND1     = 'CLASS CORRESPONDING TO M1, N1, . . '
      IND2     = 'CLASS CORRESPONDING TO M2, N2, . . '
      COCHDF   = 'COCHRAN DF (BALANCED SAMPLES)'
      PROBF    = 'PROB > F STAT'
      PROBT    = 'PROB > T STAT (2-SIDED, EQUAL VAR)'
      PROBTSW  = 'PROB > T STAT (2-SIDED, SATTERTHWAITE)'
   PROBCOCH = 'PROB > T STAT (2-SIDED, COCHRAN-COX)';


PROC DATASETS LIBRARY=WORK;      /* Clean up used datasets */
     DELETE Z1 Z2 CLASSOUT
            CLASS1 CLASS2;


%MEND TTEST;
```