Paper p906

Computation of pvalues for Multiple Comparisons with a Control in the SAS System

Jean G. Orelien, Analytical Sciences, Inc., Durham, North Carolina

ABSTRACT

In statistics, multiple comparison procedures (MCP) to compare several means of a treatment are widely used. However, the computations of p-values for MCPs can be complex. In this paper, we discuss and generalize the algorithm used for computations of pvalues for multiple comparisons with a control (MCC) in the SAS system. The limitations of the algorithm will also be discussed. Currently, besides the function PROBMC, p-values for MCC can only be obtained in PROC GLM or PROC MIXED. However, there are other cases when a researcher may want to perform MCC for data that can only be analyzed in PROC LOGISTIC or PROC GENMOD. For example, in toxicology, it is often of interest to compare several levels of a toxic substance against a zero dose (the control). Many of the dependent variables that are of interest to researchers in this field are binary such as presence or absence of malformation in which case logistic regression methods would be used rather than analysis of variance (ANOVA) or linear mixed models. A SAS macro implemented in IML will be provided to compute pvalues for MCC using as input the parameter estimates (beta vector) and the variance-covariance matrix of the betas.

I INTRODUCTION

Besides the question of whether a treatment effect is statistically significant, in many cases, another equally important question to researchers is to determine which level of a treatment in an experiment might be different from the control. If there are *k* levels of this treatment in addition to the control, this corresponds to performing *k* statistical tests. In performing these tests, we want the probability of incorrectly rejecting one or more of the hypothesis to be less or equal to a specified family wise error rate (FWE), also called an experiment wise error rate. In contrast, for a given hypothesis, for example that treatment *i* is equal to the control, the probability of incorrectly rejecting that single hypothesis is called a comparison wise error rate. The comparison wise error (CWE) rate is less or equal to the FWE. We use the term multiple comparison (MCPs) to denote the fact that we are performing these *k* statistical comparisons while controlling for the FWE rate.

In performing MCPs, a simple approach to limit the FWE to a desired α level is to use the Bonferroni method which consists of choosing the CWE rate for each of the *k* tests to be α/k . This method of controlling the FWE rate has been shown to be conservative and is not recommended for large values of *k*. In practice, the test test proposed by Dunnett (1955) is often used to perform MCPs with a control. When the researcher is willing to assume that there is a monotonic treatment response, multiple step procedures can be used to find the lowest level (or highest) where a significant statistical difference occurs. Multiple step procedures can be step-down or step-up procedures. In a step-down procedure, the *k* statistics are tested in order starting with the most significant until a rejection occurs whereas with a step-up procedure, the order of testing of the *k* test statistics start with the least significant until a difference is detected. Step-down procedures for multiple comparisons with a control were proposed by Williams (1971, 1972) and Dunnett and Tamhane (1995). Methods for step-up procedures for multiple comparisons with a control were proposed by Dunnett and Tamhane (1992).

In this paper, we will restrict our attention to the single step MCC procedure proposed by Dunnett (1955). P-values for this test (Dunnett's test) can be obtained in PROC GLM, PROC Mixed or using the function PROBMC. The algorithm used to compute these p-values in the SAS system will be discussed. In section II, we present the general form of the test statistic for the Dunnett's test. Methods for obtaining p-values are discussed in section III. In section IV, we discuss limitations of the algorithms. An example is given where one of the algorithms used in SAS returns a missing value for the p-value. Section V discusses how p-values for MCC can be obtained for data that are analyzed in PROC GENMOD or PROC LOGISTICS. (As of version 8.0, p-values for MCC are not provided in these procedures). For these cases where p-values cannot be obtained directly in the SAS system, we provide a SAS macro in the appendix.

II General Form of the Test Statistic

Consider an experiment with *k* treatments and a control. The experimenter wishes to test the differences between the control and each of the treatment with a FWE rate of α :

$$H_o : u_i = u_0 ,\ H_a : u_i \neq u_0 \text{ for } i = 1, 2, ..., k. \qquad (1)$$

Let σ^2_i be the variance for $\overline{x_i} - \overline{x_0}$ and u_i , u_0 are respectively the i^{th} treatment mean and the mean for the control. Define t_i as follows:

$$t_i = \frac{|\overline{x_i} - \overline{x_0}|}{\hat{\sigma}_i} \qquad (2)$$

In the GLM model, $\hat{\sigma}_i = \sqrt{s^2 * (\frac{1}{N_i} + \frac{1}{N_0})}$ where s^2 is the pooled variance and N_i , N_0 are respectively the sample size for the i^{th} treatment and the sample size for the control. Under the null hypothesis, t_i has a t distribution. If we assume that σ is known, the test statistic is then given by

$$d_i = \frac{|\overline{x_i} - \overline{x_0}|}{\sigma_i} \qquad (3)$$

Now assume a model of the form:

$$f(y) = X\beta + \varepsilon \qquad (4)$$

Where *f()* is the link function . In the GLM case, *f()* is the identity function. In the case of binary data, *f()* is the logit function. We denote by β , the vector of the parameter estimates and by Σ the variance-covariance matrix of the betas. Let L be the matrix corresponding to the set of contrast for comparing each treatment *i* against the control. The test statistic in (3) can be expressed as a function of $L, \beta, and\ \Sigma$. Namely, the test statistic is given by:

$$d_i = \frac{|(\hat{u}_i - \hat{u}_0)|}{\sqrt{var(u_i - u_0)}} = \frac{|l_{i'} * \hat{\beta}|}{\sqrt{l_{i'} \Sigma l_i}} \qquad (5)$$

If we assume that the betas have a normal distribution and the true variance of the betas, Σ is known then the $d_i s$ have a multivariate normal probability with mean **0** and variance-covariance matrix the correlation matrix of $L' \Sigma L$. To see this, notice that $L' \Sigma L$ can be written as the variance-covariance matrix for a vector z' where z' can be written as:

$$Z' = B * \begin{pmatrix} \overline{x}_1 - \overline{x}_0 \\ \overline{x}_2 - \overline{x}_0 \\ \ldots \\ \overline{x}_k - \overline{x}_0 \end{pmatrix} \text{and } B = \begin{bmatrix} \frac{1}{\sigma_1} & 0 & 0 & \ldots & 0 \\ 0 & \frac{1}{\sigma_2} & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 0 & \frac{1}{\sigma_k} \end{bmatrix}$$

$\overline{x}_i - \overline{x}_0$ has mean 0 and variance, the i^{th} element on the diagonal of $L'\Sigma L$.

The computations of p-values for MCC is then equivalent to computing a probability from a multivariate normal or a multivariate T.

Under the null hypothesis, the critical value for this test, *q* is such that: $Prob(z_1 < q, z_2 < q, \ldots, z_k < q) = 1 - \alpha$ .The p-values for comparing treatment *i* and the control are given by the expression:

$$p\text{-}value = 1 - prob(Z < d_i, Z < d_i, \ldots, Z < d_i) \qquad (6)$$

III Methods for the Computation of the p-values

Because the p-values involve multivariate normal or multivariate T, their computations can be computer intensive. This is due to the fact that if the number of treatment levels is k, to compute the p-values one needs to compute the values of an integral with k dimensions.

To compute p-values for MCC, SAS uses an algorithm by Hsu (1992) that reduces the dimension of the integrals to two for multivariate T and to one for multivariate normal. This algorithm is used to provide p-values for MCC when the option ADJUST= DUNNETT is specified by the user in the LSMEANS statement of Proc GLM and Proc MIXED. It is also used in the function PROBMC. P-values can also be obtained by using the option ADJUST=SIMULATE in the LSMEANS statement. When this option is used p-values are computed by simulation of values from a multivariate T distribution.

The Hsu algorithm

We now discuss the Hsu algorithm used in SAS to compute p-values for MCC.

The vector for the test statistic $\begin{pmatrix} d_1 \\ d_2 \\ \ldots \\ d_k \end{pmatrix}$ (see equation 3)has

a multivariate normal distribution with mean **0** and variance-covariance matrix R which is the correlation of $L'\Sigma L$.

Now assume that the correlation between two test statistics is: $\rho_{ij} = \lambda_i * \lambda_j$. The correlation matrix for Z is then given by:

$$R = \begin{bmatrix} 1 & \lambda_1 * \lambda_2 & \cdots & \cdots & \lambda_1 * \lambda_k \\ \lambda_2 * \lambda_1 & 1 & \cdots & \cdots & \lambda_2 * \lambda_k \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \lambda_k * \lambda_1 & \cdots & \cdots & \cdots & 1 \end{bmatrix} = \qquad (7)$$

$$\begin{bmatrix} 1-\lambda^{2}_{1} & 0 & 0 & \cdots & 0 \\ 0 & 1-\lambda^{2}_{2} & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 & 1-\lambda^{2}_{k} \end{bmatrix} + \begin{bmatrix} \lambda^{2}_{1} & \lambda_1 * \lambda_2 & \cdots & \cdots & \lambda_1 * \lambda_k \\ \lambda_2 * \lambda_1 & \lambda^{2}_{2} & \cdots & \cdots & \lambda_2 * \lambda_k \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \lambda_k * \lambda_1 & \cdots & \cdots & \cdots & \lambda^{2}_{k} \end{bmatrix} \qquad (8)$$

So, the test statistic has the same distribution as:

$$\left(\sqrt{1-\lambda^{2}_{1}}\, Z_1 + \lambda_1 * Z_0, \ldots, \sqrt{1-\lambda^{2}_{k}}\, Z_k + \lambda_k * Z_0\right)$$

where $Z_0, Z_1, \ldots, Z_k$ are *iid* standard normal random variables.

According Hsu and Nelson (1998), by conditioning on the pooled standard deviation $\hat{\sigma}$ and Z_0, the critical value *q* can be written as a solution to:

$$\int_0^{\infty}\int_{-\infty}^{\infty} \prod_{i=1}^{k} \left[\Phi\left(\frac{\lambda_i y + du}{\sqrt{1-\lambda^{2}_{i}}} \right) - \Phi\left(\frac{\lambda_i y - du}{\sqrt{1-\lambda^{2}_{i}}} \right) \right] d\Phi(y) \Upsilon(u) du = 1-\alpha \quad (9)$$

where Φ is the standard normal distribution and Υ is the density of $\frac{\hat{\sigma}}{\sigma}$. With infinite degrees of freedom (9), becomes:

$$\int_{-\infty}^{\infty} \prod_{i=1}^{k} \left[\Phi\left(\frac{\lambda_i y + d}{\sqrt{1-\lambda^{2}_{i}}} \right) - \Phi\left(\frac{\lambda_i y - d}{\sqrt{1-\lambda^{2}_{i}}} \right) \right] d\Phi(y) = 1-\alpha \qquad (10)$$

Thus, to obtain p-values for testing whether any of the level of a treatment is different from the control, we can simply use (7) or (8). For a given level *i*, p-values are obtained by computing one minus the left side of the expression in (7) or (8) using the values for the $\lambda_{i'} s$ and substituting $d_i = (|(\bar{x}_i - \bar{x}_0)|)/\hat{\sigma}_i$ for the value of *d*. That is for a given level *i*, p-values would be computed as:

$$1 - \int_0^{\infty} \int_{-\infty}^{\infty} \prod_{i=1}^{k} \left[\Phi\left(\frac{\lambda_i y + (d_i)u}{\sqrt{1 - \lambda^2_i}} \right) - \Phi\left(\frac{\lambda_i y - (d_i)u}{\sqrt{1 - \lambda^2_i}} \right) \right] d\Phi(y) \Upsilon(u) du \qquad (11)$$

or

$$1 - \int_{-\infty}^{\infty} \prod_{i=1}^{k} \left[\Phi\left(\frac{\lambda_i y + d_i}{\sqrt{1 - \lambda^2_i}} \right) - \Phi\left(\frac{\lambda_i y - d_i}{\sqrt{1 - \lambda^2_i}} \right) \right] d\Phi(y) \qquad (12)$$

where (11) is for finite degrees of freedom and (12) is for the case when we are assuming infinite degrees of freedom (i.e., known variance).

For the case *k*=3, that is when there are three treatment levels and a control, the values for the $\lambda_{i'} s$ can be determined exactly by solving:

$$\rho_{12} = \lambda_1 * \lambda_2$$
$$\rho_{13} = \lambda_1 * \lambda_3$$
$$\rho_{23} = \lambda_2 * \lambda_3$$

The same is true for the case *k*=2. When *k* is greater than three, factor analysis can be used to estimate the values of the $\lambda_i s$ (Hsu, 1992). These values for the $\lambda_i s$ are not exact (except in the one way ANOVA) and so the p-values are only an approximation. In SAS, factor analysis with the iterated principal factor method is used to obtain the values of the $\lambda_i s$.

IV Limitations of the Hsu Algorithm

There are cases when the Hsu algorithm in SAS will yield missing values. This can occur when the decomposition of the matrix R is degenerate. When this happens some of the values of the lambdas for example might be close to 0. In this section, we give an example where the option ADJUST=DUNNETT in the LSMEANS statement returns missing p-values where as the option ADJUST=SIMULATE returns valid p-values.

Table 1. Data example where the option=Dunnett in PROC GLM returns missing p-values.

Dose	Data	
0	AGD	2.8 2.75 2.8 2.7 2.75 2.7
	Body Wt.	102 101 89 87 92 98
0.08	AGD	2.55 2.8 2.65 2.7
	Body Wt.	99 102 97 96
0.8	AGD	2.6 2.55 2.7
	Body Wt.	89 86 86
8	AGD	2.2 2.3 2.3
	Body Wt.	73 76 63

The data was analyzed in PROC GLM with the following statements:

```
PROC GLM DATA=ONE;
CLASS DOSE;
MODEL AGD=DOSE;;
LSMEANS DOSE / ADJUST=DUNNETT PDIFF=CONTROL('0') ;
```

For these statements, SAS returns missing P-values for comparing each of the nonzero dose group versus the zero dose group.

When the data was analyzed in PROC GLM with the statements below this paragraph, the p-values returned were 0.3226, 0.17225, 0.0052 respectively for comparing dose group=0.08 versus the control, dose group=0.8 versus the control and dose group=8 versus the control.

```
PROC GLM DATA=ONE;
CLASS DOSE;
MODEL AGD=DOSE;
LSMEANS DOSE / ADJUST=SIMULATE CONTROL PDIFF=CONTROL('0') ;
```

V Multiple Comparisons Outside of Proc GLM and Proc MIXED

As we stated before p-values for MCC cannot be obtained directly in other procedures besides PROC GLM and PROC MIXED. For these procedures, equation (5) can be used to compute the test statistic (d_i). PROC FACTOR can be used to to obtain the lambdas in equation (7) and (8). The p-values can then be computed using PROBMC as follows:

$$P\text{-}value = 1 - PROBMC(string, d_i, ., k, of\ \lambda_1 - \lambda_k)$$

A SAS macro is provided in the appendix to perform these multiple comparisons for models with one treatment factor that are analyzed in PROC GENMOD. This macro assumes that the model based covariance matrix for the betas are used (COVB). For models in which the empirical variance-covariance matrix of the betas are used, the reader is referred to Orelien et al. (2000).

VI Conclusion

MCC are widely used in Statistics. However, the computation of the p-values can be computer intensive. In this paper, we discuss how p-values are computed in the SAS System. We give a generalization of the Hsu algorithm used in PROC GLM and PROC MIXED. This algorithm can be used to compute p-values for MCC in models that may be more complex than the traditional ANOVA. A SAS macro is provided to compute these p-values for models that are fitted in PROC GENMOD or PROC LOGISTIC and could not be fitted in PROC GLM or PROC GENMOD.

References:

Chung-Kei Chang and Dror Rom (1998). Multiple comparison procedures with SAS. Presented at the SouthEast SAS User Conference in Norforlk, Virginia.

Dunnett, C. W. (1955). “A multiple comparison procedure for comparing Several Treatments with a control.”, JASA, 50, 1096-1121.

Dunnett, C.W. and Tamhane, A.C. (1991). Step-down multiple tests for comparing treatments with a control in unbalanced one-way layouts. Statistics in Medicine 10:939-947.

Dunnett, C.W. and Tamhane, A.C. (1992). A step-up multiple test procedure. Journal of the American Statistical Association 87:162-170.

Gupta, S. S. (1963). “Probability integrals of multivariate normal and multivariate t” Annals of Mathematical Statistics, 34, 792-828.

Hsu J. C. (1992). “The factor analytic approach to simultaneous Inference in the general linear models”. Journal of Computational and Graphical Statistics, 1, 151-168

Hsu J. C., Nelson B. (1998). “Multiple comparisons in the general linear model”, Journal of Computational and Graphical Statistics, 7, 23-41.

Orelien et al. (2000). Multiple Comparison with a control in GEE models using the SAS System. Presented at the SAS User Group International Conference in Indianapolis.

Royen, T. (1987) “An approximation for multivariate normal probabilities of rectangular regions”, Statistics, 18, 389-400.

Williams, D.A. (1971). A test for differences between treatment means when several dose levels are compared with a zero dose control. Biometrics 27:103-117.

Williams, D.A. (1972). The comparison of several dose levels with a zero dose control. Biometrics 28:519-531.

Zbynek, S. (1967). “On multivariate normal probabilities of rectangles: Their Dependence on Correlations”, The Annals of Mathematical Statistics, 39, 1425-1434.

SAS Institute. *SAS/STAT Software: Changes and Enhancements through Release 6.12*, Cary, NC: SAS Institute, Inc.

Acknowledgments

We thank Dr. Robert Chapin and Martha Harris of NIEHS for allowing us the use of data from NTP experiments.

Contact Information

Your questions and comments are welcome. Please contact:

Jean G. Orelien
Analytical Sciences, Inc.
2605 Meridian Pkwy.
Durham, NC 27713
Work Phone: (919)544-8500 (ext. 112)
Fax: (919)544-7307
Email: jorelien@asciences.com
Web: http://www.asciences.com

Appendix. A Macro to Compute p-values for Multiple Comparison With a Control in GEE Models

```
    /************************************
    THIS PROGRAM COMPUTES P-valueS FOR THE DUNNETT_HSU TEST AND 95% CI FOR GEE MODELS.
    THE INPUTS FOR THIS PROGRAM ARE THE BETA VECTOR, THE VARIANCE-COVARIANCE MATRIX FOR THE BETAS
    AND THE NUMBER OF TREATMENTS EXCLUDING THE CONTROL
    THIS PROGRAM ASSUMES THAT THE FIRST BETA PARAMETER CORRESPONDS TO THE CONTROL

    ************************************/
    /*
    THE PARAMETERS THAT HAVE TO BE INPUT INTO THE MACRO ARE THE FOLLOWING:
    BVECTOR: THE VECTOR OF THE BETAS
    COVAR: THE EMPIRICAL VARIANCE-COVARIANCE OF THE BETAS
    K: THE NUMBER OF TREATMENT LEVELS EXCLUDING THE CONTROL
    */
    %macro dunnett (bvector, covar, k);

    /* KEEP ONLY THE VARIABLES THAT HAVE THE COEFFICIENTS FOR THE VARIANCE-COVARIANCE MATRIX */

    %let numlev =%eval(&k+1);

    Data covbeta;
    Set &covar;
    Drop rowname scale;
    IF parameter=’Scale’ then  delete;

    Data parmname;
    Set &bvector;
    If level1=’0’  then delete;
    If parameter=’intercept’ then delete;
    If parameter=’scale’ then  delete;
    Numlevel=%eval(&k);
    K=numlevel;
    Treatmentname=parameter||’ ’||level1;
    Keep treatmentname k ;

    Data parmname;
    Set parmname;
    Order=_n_;
    if upcase(parameter)=’SCALE’ then delete;
    if df=0 then delete;

    /* THE MAIN GOAL OF THIS IML CODE IS TO COMPUTE THE CORRELATION MATRIX FOR THE LSMEANS */

    PROC iml;
    Use covbeta;
    Read all into covb;
    Use &bvector;
    Read all var {estimate} into b;

    L2={0 1 -1, 0 1 0};
    L3={0 1 -1 0, 0 1 0 -1 , 0 1 0 0 };
    L4={0 1 -1 0 0 ,  0 1 0 -1 0 , 0 1 0 0 -1 , 0 1 0 0 0 };
    L5={0 1 -1 0 0 0 , 0 1 0 -1 0 0 , 0 1 0 0 -1 0 , 0 1 0 0 0 -1 , 0 1 0 0 0 0};
    L6={0 1 -1 0 0 0 0 , 0 1 0 -1 0 0 0 , 0 1 0 0 -1 0 0 , 0 1 0 0 0 -1 0 ,
    0 1 0 0 0 0 -1, 0 1 0 0 0 0 0 };

    /* COMPUTE LS MEANS */

    beta=b[1:&numlev,];
    Est=l&k*b[1:&numlev,];

    /* FIX THE VAR-COVARIANCE MATRIX TO ACCOUNT FOR THE PRESENCE OF THE INTERCEPT AND THE SCALE PARAMETERS
IN THE BETA
    VECTOR */

    Cov=covb[1: &numlev,1: &numlev];
    Cov_diff=l&k*cov*t(l&k);
    Cov_diff=l&k*cov*t(l&k);
    Corr_diff=(diag((1/sqrt(cov_diff))))*cov_diff*(diag((1/sqrt(cov_diff))));
    stderror=sqrt(vecdiag(cov_diff));
    q=(est/stderror)||est||stderror;

    print cov est beta;

    Create corr_lsmns from corr_diff;
    Append from corr_diff;

    Create lsmeans from q;
    Append from q;

    Create covdiff from cov_diff;
    Append from cov_diff;

    Quit;

    PROC print data=corr_lsmns;
    Title5 "correlation matrix for the ls-means";
```

```
Data lsmeans;
Set lsmeans;
Order=_n_;
Test_stat=abs(col1);
Lsmeans=col2;
Stderror=col3;
Drop col1-col3;

/* SET UP CORRELATION MATRIX TO BE USED BY PROC CORR */

Data corr_lsmns(type=corr);
Set corr_lsmns;
_type_=’corr’;
_name_=’col’||left(put(_n_, z1.));

/* USE PROC FACTOR TO DECOMPOSE MATRIX */

Ods select factorpattern;
Ods output factorpattern=lambda;

PROC factor data=corr_lsmns method=prinit;

PROC transpose data=lambda out=lambda(keep=l1-l&k) prefix=l;
Var factor1;

Data lambda;
Set lambda;
Do i=1 to &k;
Order=i;
Output;
End;
drop i;


PROC sort data=lambda; by order;

Data probmc;
Merge lambda(in=a) lsmeans(in=b) parmname(in=c);
By order;
If a and b and c;
String=’dunnett2’;
If k=2 then do;
P-value=1-probmc(string,test_stat,.,.,2,l1,l2);
Q95=probmc(string,.,0.95,.,2,l1,l2);
End;
If k=3 then do;
P-value=1-probmc(string,test_stat,.,.,3,l1,l2,l3);
Q95=probmc(string,.,0.95,.,3,l1,l2,l3);
End;
If k=4 then do;
P-value=1-probmc(string,test_stat,.,.,4,l1,l2,l3,l4);
Q95=probmc(string,.,0.95,.,4,l1,l2,l3,l4);
End;
If k=5 then do;
P-value=1-probmc(string,test_stat,.,.,5,l1,l2,l3,l4,l5);
Q95=probmc(string,.,0.95,.,5,l1,l2,l3,l4,l5);
End;
If k=6 then do;
P-value=1-probmc(string,test_stat,.,.,6,l1,l2,l3,l4,l5,l6);
Q95=probmc(string,.,0.95,.,6,l1,l2,l3,l4,l5,l6);
End;

Lci=lsmeans-q95*stderror;
Uci=lsmeans+q95*stderror;
Drop l&numlev-l9;

PROC print data=probmc;
Title5 "Dunnett-Hsu p-values and 95% CI";

%mend dunnett;
```