Diplomarbeit im Lehrgebiet Bioinformatik

Sommersemester 2001

Dozent: Prof. Dr. L. Hothorn

Fachbereich Gartenbau

Universität Hannover

Simultaneous tests and confidence intervals for experimental agricultural designs

Cornelia Schratz

10. Semester

Schäferei 3

31319 Sehnde

Abgabetermin: 17. April 2001

Contents

Abbreviations

α	significance level, type I error
df	degree of freedom
H	null hypothesis
k	number of groups
μ	intercept (e.g. grand mean)
MS	means square
N	total sample size
n_j	sample size of group j
p	number of parameters
σ^2	variance
s^2	estimate of variance, usually the mean square error
$stderr(\bar{x}_{i..})$	standard error of the lsmean of level i of the main factor
t	number of hypotheses
$\bar{x}$	arithmetic mean

ANOVA	analysis of variance
ANCOVA	analysis of covariance
CER	comparisonwise error rate
FWE	familywise error rate
MCP	multiple comparisons procedures
lsmean	least square mean

1 Introduction

Almost every agricultural experiment includes more than two groups. Common are comparisons such as several groups against a control or comparisons where all groups are tested against each other. To analyze such multiple comparisons correctly, in view of the false positive rate, Westfall *et al.* (1999) provide two SAS macros, *%SimTests* and *%SimIntervals*. The aim of this thesis is to show their applications for common agricultural designs. Compared to the SAS procedures MIXED, GLM and MULTTEST the two macros have several advantages. Both macros perform contrast tests. They analyze and control the familywise error for all these tests over all factors simultaneously. The above mentioned SAS procedures compute contrasts tests, too. However there is no option to adjust the p-values for multiplicity in PROC GLM and PROC MIXED except for single step procedures. Although we can analyze contrast test and control the familywise error rate with e.g. bootstrap with PROC MULTTEST, this procedure does not allow more than one factor in the `class` statement. However it is possible but difficult to analyze more than one factor simultaneously with PROC MULTTEST, therefore a single factor has to be created where all desired combinations of the levels of the factors appear. Some SAS procedures calculate simultaneous confidence intervals only for all-pairs and many-to-one tests. However the simultaneous confidence intervals are not available for any other contrasts in SAS; the *%SimIntervals* macro calculates these intervals. The *%SimTests* macro performs the closed testing procedure. With PROC MULTTEST this is available for SAS, too. But SAS uses only the less powerful unconstrained step-down method of Holm, whereas *%SimTests* uses the unconstrained step-down method of Holm and the logical constraint method of Shaffer. And again it is difficult to analyze more than one factor simultaneously with PROC MULTTEST.

In section 2 we give a brief explanation of multiple comparisons. We introduce two error rates, the model and the estimates, which are used in the further sections. After that several multiple comparisons procedures are shown. Section 3 provides the introduction and invocation of the macros *%SimTests* and *%SimIntervals*. The analysis via these two

macros for the agricultural designs are given in section 4. Finally four examples are analyzed in section 5. Every invocation and all data sets used in this thesis are provided on disk.

The macros *%SimTests* and *%SimIntervals* are programmed for the SAS System. According to Westfall *et al.* (1999) the macros run not in versions prior to version 7 of the SAS System. In this thesis version 8.0 is used.

2 Explanation of multiple comparisons

As described in the introduction multiple comparisons appear in many experiments. To control the type I error (α) multiple comparisons procedures (MCPs) have to be used. In this section the comparisonwise error rate and the familywise error rate will be introduced. Then a general model and the estimates for multiple comparisons will be shown. Afterwards there will be a brief introduction to the multiple comparisons procedures which are used later in this thesis.

2.1 Error rates

To show the control of the multiplicity effect for pairwise comparisons we consider a family of t null hypotheses (further on H is used as an abbreviation for any null hypothesis). We assume that from these hypothesis $H_1, \ldots, H_m$ are true and $H_{m+1}, \ldots, H_t$ are wrong. For some unknown $m \leq t$ the probability P to reject a true null hypotheses is called type I error α or false positive rate. When a single true null hypothesis H_i $(i = 1, \ldots, m)$ is tested on significance its rejection probability is called *comparisonwise error rate* (CER) irrespective of the other hypothesis and their rejection. Thus, CER is defined as:

$$\text{CER} = P(\text{Reject } H_i | H_i \text{ is true}), \quad i = 1, \ldots, m.$$

Equivalently in terms of confidence intervals, this is

$$\text{CER} = P(\text{Corresponding interval for } H_i \text{ does not contain the parameter}).$$

Regarding the entire family of hypotheses we do not have any control of the error type I rate if only the CER is examined. Here the *familywise error rate* (FWE) is used. It is the probability of rejecting a true null hypothesis when the entire family of inferences is considered. The FWE is always less or equal than the CER because it does regard the multiplicity effect, which the CER does not. The FWE for test of hypotheses is defined as

$$\text{FWE} = P(\text{Reject at least one of } H_1, \ldots, H_m | H_1, \ldots, H_m \text{ all are true}).$$

For simultaneous confidence intervals this is

$$\text{FWE} = P(\text{at least one interval is incorrect}).$$

More error rates than CER and FER exist, for example the false discovery rate (FDR), further details are discussed by Hochberg and Tamhane (1987, p.2-12) or Westfall *et al.* (1999, p.16-19).

2.2 Model and estimates

It is necessary for multiple comparisons procedures to specify a model and parameters of interest. To introduce the model and the estimates an example from Neter *et al.* (1996, p.711) is used who give the following data set: Four rust inhibitors (A, B, C and D) were tested. The experimental units were treated by severe weather conditions. Thus the endpoint is a coded value of effectiveness of the inhibitors on a continuous scale. With increasing effectiveness the coded value becomes larger. To the four brands 40 experimental units were randomly assigned, with sample size 10 per brand. This is the data set:

obs.	A	B	C	D
1	43.9	89.8	68.4	36.2
2	39.0	87.1	69.3	45.2
3	46.7	92.7	68.5	40.7
4	43.8	90.6	66.4	40.5
5	44.2	87.7	70.0	39.3
6	47.7	92.4	68.1	40.3
7	43.6	86.1	70.6	43.2
8	38.9	88.1	65.2	38.7
9	43.6	90.8	63.8	40.9
10	40.0	89.1	69.2	39.7

2.2.1 Model

The usual model for the analysis of variance (ANOVA) for fixed effects is:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where

- $\mathbf{Y}$ is a $N \times 1$ vector that contains all measured values (observations) (N is the total sample size),
- $\mathbf{X}$ is the fixed and known $N \times p$ design matrix, where p is the number of parameters,
- $\boldsymbol{\beta}$ is the fixed and unknown $p \times 1$ parameter vector. Here the grand mean, the levels of the treatments, called factors, and the covariates appear,
- $\boldsymbol{\varepsilon}$ is the $N \times 1$ vector which comprises the levels of the error term. Thus the vector is random and unobservable.

The parametrization of the model for our *rust inhibitor* example might be:

$$\mathbf{Y} = \begin{pmatrix} 43.9 \\ 39.0 \\ 46.7 \\ \vdots \\ 89.8 \\ 87.1 \\ \vdots \\ 68.4 \\ \vdots \\ 36.2 \\ \vdots \\ 39.7 \end{pmatrix}, \mathbf{X} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix}, \boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \end{pmatrix} \text{ and } \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_{40} \end{pmatrix}.$$

To regard the intercept parameter β_1, the first column of $\mathbf{X}$ contains 1's. After the first one the next four columns represent the four levels of the treatment. These are the rust inhibitors, their parameters are $\beta_2, \ldots, \beta_5$.
Each observation y_i follows the linear additive model

$$y_i = x_{i1}\beta_1 + \ldots + x_{ip}\beta_p + \varepsilon_i, \quad i = 1, \ldots, N.$$

For example for the first observation of the second level of the factor this formula might be as followed. Note that for better differentiation the indices are altered.

$$y_{11} = x_{11,1}\beta_1 + \ldots + x_{11,5}\beta_5 + \varepsilon_{11}, \quad i = 1, \ldots, 40.$$

This becomes

$$89.9 = 1\beta_1 + 0\beta_2 + 1\beta_3 + 0\beta_4 + 0\beta_5 + \varepsilon_{11}.$$

To use the analysis of variance certain assumptions have to be given:

- The errors ε_i are randomly distributed with mean zero and common variance σ^2.
- All errors belonging to any pair of observations have to be uncorrelated.

In the following thesis we deviate from the above general model representation for notational convenience thus the general model is adapted for the particular model.

2.2.2 Estimates

In this paragraph it is explained how the estimates are calculated. The first column (intercept respectively grand mean) of the model is the sum of the columns of the brand groups. Therefore the columns of $\mathbf{X}$ are linearly dependent and thus the model is over-parameterized. If this is the case it is not possible to estimate all of the parameters unbiasedly because $\mathbf{X}'\mathbf{X}$ may not be invertible. Nevertheless certain linear combinations of the parameters are estimable. Such linear combinations could be adjusted means or

least square means (or in SAS-Syntax: lsmeans) and they can be expressed as

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{Y},$$

where $(\mathbf{X}'\mathbf{X})^{-}$ denotes a generalized inverse. The mean square error is the estimate of variance:

$$s^2 = (\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}})'(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}})/df,$$

where

$$df = (N - \text{rank}\mathbf{X}),$$

which is the total sample size minus the number of linearly independent columns in $\mathbf{X}$. In the rust inhibitor example the degree of freedom is $df = 40 - 4 = 36$.
In the following sections multiple comparisons are represented by contrasts. As Neter *et al.* (1996, p.720) describe a contrast is defined as a linear combination of the factor level means β_i where the coefficients c_i sum to zero. For example to estimate the difference of brand 2 and brand 3, $\beta_3 - \beta_4$, this would be:

$$\beta_3 - \beta_4 = \mathbf{c}'\boldsymbol{\beta} = \begin{pmatrix} 0 & 0 & 1 & -1 & 0 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \end{pmatrix}.$$

Later on the standard errors of the estimates are needed. For a general estimable function $\mathbf{c}'\boldsymbol{\beta}$, the variance of the estimate is

$$\sigma^2\{\mathbf{c}'\boldsymbol{\beta}\} = \sigma^2\mathbf{c}'(\mathbf{X}'\mathbf{X})^{-}\mathbf{c}.$$

And the standard error is the standard deviation of the estimate:

$$stderr\{\mathbf{c}'\boldsymbol{\beta}\} = s\sqrt{\mathbf{c}'(\mathbf{X}'\mathbf{X})^{-}\mathbf{c}}.$$

2.3 Multiple comparisons procedures

In the following sections several multiple comparisons procedures are used which are shown in this paragraph. In the beginning a two-sample test, an all-pair test and a many-to-one test are defined, then the Bonferroni method is introduced and after that the methods of Holm and Shaffer. Then the closed testing procedure is shown with two special types of it, these are the unconstrained step-down method of Holm and the logical constraint method of Shaffer. Finally the multivariate t distribution, contrasts tests, confidence intervals and critical values are introduced.

2.3.1 Two-sample, many-to-one and all-pairs tests

Later on two-sample, many-to-one and all-pairs tests are used. We give a brief introduction in their definition. When two groups i and j are tested a two-sample test is used. Under the assumptions of normal distribution, continuous and independent data and homogeneous variances the t-test as a two-sample test can be used. The test statistic is:

$$t_{i,j} = \frac{|\bar{x}_i - \bar{x}_j|}{s\sqrt{\frac{1}{n_i} + \frac{1}{n_j}}}.$$

Often we are interested in experiments containing more than two groups. If we have k groups and we are interested in all possible pairwise comparisons we would have to test $t = k(k-1)/2$ null hypotheses. If the data fulfill the assumptions of normal distribution, continuous and independent data, homogeneous variances and equal sample sizes, then the Tukey-test is the appropriate all-pairs test, see Tukey (1953) for details. It has the test statistic:

$$t_{i,j}^{Tukey} = \frac{|\bar{x}_i - \bar{x}_j|}{s\sqrt{\frac{2}{n}}}.$$

A many-to-one test is needed when we have k groups where one of them is a control group c and the other $k-1$ groups are tested only against this control. Then we have $k-1$ null hypotheses. These can be tested with the Dunnett-test, which is a many-to-one test, see Dunnett (1955) for details. The Dunnett-test requires data which are independent,

continuous, normally distributed and have variance homogeneity. This is the test statistic:

$$t_i^{Dunnett} = \frac{|\bar{x}_i - \bar{x}_c|}{s\sqrt{\frac{2}{n}}}.$$

Note that a test statistic from the t-test is compared with the critical values from the marginal t distribution. The test statistics from the Tukey and the Dunnett test are compared with the critical values from the multivariate t distribution, which will be introduced in 2.3.6.

2.3.2 Bonferroni method

If k groups are compared with two-sample tests, and every one of the $t = k(k-1)/2$ hypotheses has the significance level of α then the CER is controlled. However the FWE will be larger than α. A logical way to correct the CER is CER $= \alpha/t$, which is the Bonferroni method. This method is always valid. But neither does it consider the stochastic dependencies among the test statistics nor the logical dependencies between the hypotheses, thus the Bonferroni method is equal or less powerful than other multiple comparisons procedures.

2.3.3 Holm method

An equal or more powerful MCP as the Bonferroni method is the Holm method, see Holm (1979) for details. It still controls the FWE while the CER is larger. In the following text the method is introduced. Again we are interested in all pairwise comparisons: In the beginning all $t = k(k-1)/2$ hypotheses are tested with the appropriate two-sample test. After that, the p-values are sorted in ascending order, with p_1 being the smallest p-values and p_t being the largest p-value. The smallest one, p_1 is compared with α/t. If $p_1 > \alpha/t$ then the null hypothesis appendant to p_1 (H_1) and all other H_i ($i = 2, \ldots, t$) are accepted. Otherwise p_2 is compared with $\alpha/(t-1)$. If $p_2 > \alpha/(t-1)$ then the null hypothesis belonging to p_2 (H_2) and all other H_i ($i = 3, \ldots, t$), which are not tested yet, are valid. In the case of rejection of H_2 the procedure continues. So the method ends either

with the first rejection of a null hypothesis or with the last comparison, which is p_t versus α.

2.3.4 Shaffer method

In comparison to the Holm method the method according to Shaffer (1986) uses restricted hypotheses. Thus certain combinations of true hypotheses imply truth or falsehood of other hypotheses. In contrast to the ones of Holm's method the p-values are smaller while the FWE is still controlled. As in Holm's method the t hypotheses and their p-values have to be ordered in ascending order. Again H_i will only be rejected if $H_1, \ldots, H_{i-1}$ have been previously rejected. But the multiplicity adjustment considers only those hypotheses that possibly can be true, given that the previous tested null hypotheses are all false. As an example this is shown with three groups. If we are interested in all-pairs comparisons we have three null hypotheses: $H_{1,2}$, $H_{1,3}$ and $H_{2,3}$, where the groups are denoted by the indices. First each of the null hypotheses is tested with the appropriate two-sample test then the p-values are sorted. For example this could be $p_{1,2} < p_{2,3} < p_{1,3}$. Note that the p-values have the indices of the groups, *not* the number from the sorting! The the smallest p-value $p_{1,2}$ is compared with $\alpha/3$. If the null hypothesis is rejected, both other p-values are compared with α because only one of the two remaining null hypotheses can be true. Both of them can not be true at the same time because if $1 \neq 2$ then it is not possible that $1 = 3$ *and* $2 = 3$. The Shaffer method has an equal or larger power than the methods of Bonferroni and Holm. This is caused by the logical dependencies between the hypotheses.

2.3.5 Closed testing procedure

The closed testing procedure is a powerful MCP. It is named after the Closure Principle. A closed family of hypotheses, by definition, is one for which any intersection of subset hypotheses is also a member of the family. The procedure starts with the creation of the interesting pairwise hypotheses of k groups; these are the elementary pairwise homogeneity hypotheses (short: elementary hypotheses). After that the power set of the elementary

hypotheses is generated. From the elements of the power set all intersections are created, these intersections are intersection hypotheses. After removing redundancies each intersection is tested with an appropriate α-level test. An elementary hypothesis is only rejected if all intersection hypotheses which contain the two elements of the elementary hypothesis are rejected. Using the data set *rust inhibitor* as an example we show the different intersection hypotheses. The data set contains four levels of the group variable. If we are interested in an all-pairs tests procedure we have six elementary hypotheses. Thus the following hypotheses have to be tested:

- Elementary pairwise homogeneity hypotheses: $H_{1,2} : \mu_1 = \mu_2,\ H_{1,3} : \mu_1 = \mu_3,\ H_{1,4} : \mu_1 = \mu_4,\ H_{2,3} : \mu_2 = \mu_3,\ H_{2,4} : \mu_2 = \mu_4,\ H_{3,4} : \mu_3 = \mu_4$.
- Three means homogeneity hypotheses: $H_{1,2,3} : \mu_1 = \mu_2 = \mu_3,\ H_{1,2,4} : \mu_1 = \mu_2 = \mu_4,\ H_{1,3,4} : \mu_1 = \mu_3 = \mu_4,\ H_{2,3,4} : \mu_2 = \mu_3 = \mu_4$.
- Subset intersection (disjoint) hypotheses: $H_{\{1,2\}\cap\{3,4\}} : \mu_1 = \mu_2$ and $\mu_3 = \mu_4$, $H_{\{1,3\}\cap\{2,4\}} : \mu_1 = \mu_3$ and $\mu_2 = \mu_4,\ H_{\{1,4\}\cap\{2,3\}} : \mu_1 = \mu_4$ and $\mu_2 = \mu_3$.
- Four means homogeneity hypotheses (global hypothesis): $H_{1,2,3,4} : \mu_1 = \mu_2 = \mu_3 = \mu_4$.

Special cases of the closed testing procedure

In this paragraph we introduce two special cases of the closed testing procedure. These are the unconstrained step-down method of Holm and the logical constraint method of Shaffer. Both methods were already shown thus there is only a brief explanation of the methods and afterwards the integration in the closed testing procedure is explained.
The procedure of the unconstrained step-down method of Holm is already shown in 2.3.3. The method can be summarized as follows:

- Test all t hypotheses with an appropriate two-sample test.
- Sort all p-values belonging to the hypotheses in ascending order: $p_1, p_2, \ldots, p_{t-1}, p_t$, where $i = 1, \ldots, t$ and p_1 is the smallest and p_t is the largest p-value.

- If $p_1 \geq \alpha/t$ then stop the procedure. All H_i $(i = 1, \ldots, t)$ are valid. Otherwise go to the next step.

- If $p_2 \geq \alpha/(t-1)$ then stop the procedure. All H_i $(i = 2, \ldots, t)$ are valid. Otherwise go to the next step.

- $\vdots$

- End the procedure with p_t versus α. If $p_t \leq \alpha$ then reject the H_t. Otherwise (only) this null hypothesis is valid.

The integration of the Holm method into the closed testing procedure can be shown via a many-to-one comparison with four groups where c is the control. The indices of the null hypotheses belong to the number of the group.

$$\begin{array}{cccc} H_{c,2,3,4} & & & \text{versus } \alpha/3 \\ H_{c,2,3} & H_{c,2,4} & H_{c,3,4} & \text{versus } \alpha/2 \\ H_{c,2} & H_{c,3} & H_{c,4} & \text{versus } \alpha \end{array}$$

The global hypothesis contains three pairwise comparisons. After testing the three p-values are ordered and the smallest one, p_1, is compared with $\alpha/3$. If $p_1 \leq \alpha/3$ then the global hypothesis is rejected and the three means homogeneity hypotheses have to be tested. Each p-value belonging to one of the three means homogeneity hypotheses is compared with $\alpha/2$. The procedure ends after analyzing the elementary hypotheses by comparing the p-values of these hypotheses with α. Again an elementary hypothesis is only rejected if all other hypotheses, which contain its groups, are rejected, too.
The second special case of the closed testing procedure which is shown here is the logical constraint method of Shaffer. In 2.3.4 the method is shown. Here is a brief summary for an example with three groups:

- Test all three hypotheses with an appropriate two-sample test.

- Sort all p-values belonging to the hypotheses in ascending order: p_1, p_2, p_3, where p_1 is the smallest and p_3 is the largest p-value.
- If $p_1 \geq \alpha/3$ then stop the procedure. H_1, H_2 and H_3 are valid. Otherwise go to the next step.
- Both p_2 and p_3 are tested against α.

As an example to show the integration of the logical constraint method of Shaffer into the closed testing procedure an all-pairs design with four groups will be analyzed. Again the indices of the null hypotheses belong to the number of the group.

$$\begin{array}{cc} H_{1,2,3,4} & \text{versus } \alpha/6 \\ H_{1,2,3} \quad H_{1,2,4} \quad H_{1,3,4} \quad H_{2,3,4} & \text{versus } \alpha/3 \\ H_{\{1,2\}\cap\{3,4\}} \quad H_{\{1,3\}\cap\{2,4\}} \quad H_{\{1,4\}\cap\{2,3\}} & \text{versus } \alpha/2 \\ H_{1,2} \quad H_{1,3} \quad H_{1,4} \quad H_{2,3} \quad H_{2,4} \quad H_{3,4} & \text{versus } \alpha \end{array}$$

According to Westfall *et al.* (1999, p.70) these general adjustments are valid for all all-pair comparisons with four groups. This method can be used when the p-values are not sorted. However the procedure can be improved: rather than selecting the *maximum* of the individual adjustment one can calculate the *actual* sequence of the adjustments for a particular problem, which is more powerful.

In the following example the sorting of the p-values could be $p_{3,4} \leq p_{1,2} \leq p_{2,4} \leq p_{1,4} \leq p_{2,3} \leq p_{1,3}$. In the example the null hypotheses are marked when they are rejected, so when a hypothesis is rejected after the first comparison a p-value with an α it gets a "1". The first comparison is the smallest p-value, which is $p_{3,4}$ against $\alpha/6$. If the null hypothesis is rejected then the hypotheses $H^1_{1,2,3,4}$, $H^1_{1,3,4}$, $H^1_{2,3,4}$, $H^1_{\{1,2\}\cap\{3,4\}}$ and $H^1_{3,4}$ will be rejected because all of these hypotheses contain the comparison of group 3 versus group 4. After that the next smallest p-value is $p_{1,2}$. The global hypothesis is already rejected thus it does not have to be analyzed again. There is still at least one of three means homogeneity hypotheses (in this example there are two remaining) thus $p_{1,2}$ has to be compared with $\alpha/3$.

With the rejection the hypotheses $H^2_{1,2,3}$, $H^2_{1,2,4}$ and $H^2_{1,2}$ drop out. Thus there are none of the three means homogeneity hypotheses left, only two disjoint and four elementary hypotheses. The next comparison is $p_{2,4}$ versus $\alpha/2$ because of the disjoint hypotheses. After the rejection there is still one disjoint hypothesis left, so $p_{1,4}$ is compared with $\alpha/2$. Then the last disjoint hypothesis drops out. Thus there are only elementary hypotheses left and the remaining p-values, $p_{2,3}$ and $p_{1,3}$ are compared with α:

$$H^1_{1,2,3,4}$$

$$H^2_{1,2,3} \quad H^2_{1,2,4} \quad H^1_{1,3,4} \quad H^1_{2,3,4}$$

$$H^1_{\{1,2\}\cap\{3,4\}} \quad H^3_{\{1,3\}\cap\{2,4\}} \quad H^4_{\{1,4\}\cap\{2,3\}}$$

$$H^2_{1,2} \quad H^6_{1,3} \quad H^4_{1,4} \quad H^5_{2,3} \quad H^3_{2,4} \quad H^1_{3,4}$$

Thus the adjustments for this example are:

Comparison	3 vs. 4	1 vs. 2	2 vs. 4	1 vs. 4	2 vs. 3	1 vs. 3
CER	$\alpha/6$	$\alpha/3$	$\alpha/2$	$\alpha/2$	α	α

2.3.6 Multivariate t distribution, contrast tests, confidence intervals and critical values

Up to now only the logical dependencies between the hypotheses have been considered. However, there is another way to improve power: the consideration of the stochastic dependencies among the test statistics. In this paragraph the multivariate t distribution is introduced. It regards these stochastic dependencies. First this distribution is defined, then contrast tests, confidence intervals and critical values are introduced.

The multivariate t distribution is defined as follows:

$$t = \frac{\mathbf{Z}}{\sqrt{\chi^2_\nu/\nu}},$$

where

Z	is $(Z_1, \ldots, Z_u)$ which is distributed as multivariate normal, with expectation vector 0 and correlation matrix $\mathbf{R}$.
$\sqrt{\chi^2_\nu/\nu}$	is a random variable distributed as χ^2_ν with ν degrees of freedom independent from the numerator.

For testing more than one comparison $\{\mathbf{c_1}'\boldsymbol{\beta}\}$ we can use for simultaneous inferences $\{\mathbf{c_1}'\boldsymbol{\beta}\}$, $\{\mathbf{c_2}'\boldsymbol{\beta}\}, \ldots, \{\mathbf{c_k}'\boldsymbol{\beta}\}$ contrast tests. Their test statistic is:

$$T_i = \frac{\mathbf{c_i}'\hat{\boldsymbol{\beta}} - \mathbf{c_i}'\boldsymbol{\beta}}{s\sqrt{\mathbf{c_i}'(\mathbf{X}'\mathbf{X})^-\mathbf{c_i}}}.$$

And the joint distribution of $\{T_1, \ldots, T_k\}$ is distributed multivariate t. The correlation matrix of the contrast test is $\mathbf{R} = \mathbf{D}^{-1/2}\mathbf{C}'(\mathbf{X}'\mathbf{X})^-\mathbf{C}\mathbf{D}^{-1/2}$ with $\mathbf{C} = (\mathbf{c}_1, \ldots, \mathbf{c}_k)$ and $\mathbf{D}$ is a diagonal matrix having ith element equal to $\mathbf{c_i}'(\mathbf{X}'\mathbf{X})^-\mathbf{c_i}$.

Another way to analyze a comparison is to use confidence intervals. The estimated interval covers the difference of the two means with the probability of $1 - \alpha$. If we are interested in more than one comparison we use $(1 - \alpha)$-level simultaneous confidence intervals. The confidence intervals have the form

$$\mathbf{c}_i'\hat{\boldsymbol{\beta}} \pm c_\alpha stderr(\mathbf{c}_i'\hat{\boldsymbol{\beta}}),$$

where the critical value c_α has to be selected that FWE $= \alpha$ for the multiple comparisons of means, thus c_α satisfies

$$P(\mathbf{c}_i'\hat{\boldsymbol{\beta}} - c_\alpha stderr\{\mathbf{c}_i'\hat{\boldsymbol{\beta}}\} < \mathbf{c}_i'\boldsymbol{\beta} < \mathbf{c}_i'\hat{\boldsymbol{\beta}} + c_\alpha stderr\{\mathbf{c}_i'\hat{\boldsymbol{\beta}}\}, \forall\ i) = 1 - \alpha$$

or

$$P\left(\left|\frac{\mathbf{c}_i'\hat{\boldsymbol{\beta}} - \mathbf{c}_i'\boldsymbol{\beta}}{s\sqrt{\mathbf{c_i}'(\mathbf{X}'\mathbf{X})^-\mathbf{c_i}}}\right| \leq c_\alpha, \forall\ i\right) = 1 - \alpha$$

respectively.

The critical value c_α is the 1-α quantile of the distribution of $\max_i |T_i|$. The advantage in contrast to the Bonferroni method is the consideration of stochastic dependencies of the

test statistics thus the multivariate t distribution notes the correlation structures between them.

Common examples for the multivariate t distribution are the Tukey test and the Dunnett test which use special multivariate t distributions with a certain correlation matrix. Both tests have already been introduced in 2.3.1.

3 Explanation of the macros

As described in the introduction multiple comparisons in factorial designs are very common in agricultural experiments. Westfall et al. (1999) introduced two macros for the statistical analysis software SAS, *%SimIntervals* and *%SimTests* (short: *%Sim**), which can be used for such multiple comparison procedures. In the beginning the advantages and the invocation of these two macros are shown. The *%Sim* macros themself use the three macros *%MakeGLMStats*, *%Contrasts* and *%Estimates* for the delivery of certain parameters, these macros are explained thereafter. In the end the invocation is introduced with the example *rust inhibitor*.

3.1 The *%SimIntervals* macro

This macro calculates simultaneous adjusted confidence intervals and adjusted p-values for contrasts. The critical values from the multivariate t distribution are simulated via Monte Carlo simulation. Comparing with the SAS procedures GLM, MIXED and MULTTEST *%SimIntervals* has several advantages:

- PROC GLM and PROC MIXED can compute any contrast tests. However the p-values are not adjusted for multiplicity.
- PROC MULTTEST adjusts p-values by e.g. bootstrap, see Westfall and Young (1993) for details. But it is difficult to calculate a set of contrasts where more than one factor is involved because the class-statement allows just one factor thus it is restricted to simple one-way layout. And no multivariate t distribution based analysis is available with PROC MULTTEST.
- No procedures in SAS are available to compute directly confidence limits for any contrasts, except MIXED and GLM for all-pairs and many-to-one designs.
- Both %Sim* adjust all pairwise comparisons simultaneous, even if the comparisons come from different factors. This is a great advantage in comparison to SAS, which

adjusts the comparisons only per factor. However Biesheuvel (2001) shows a way to control the error for more than one factor in stratified designs for SAS.

Invocation

%SimIntervals may be invoked as follows:

```
%SimIntervals  (nsamp=      ,
                seed =      ,
                conf =      ,
                side =      );
```

where

`nsamp` is the size of the simulation. An input is not necessary because it has the default of 20000.

`seed` is the random number seed. Again there is a default. If there is no entry made the computer clock time is used.

`conf` is the confidence level. The default is 0.95.

`side` determines whether upper-tailed (`side=U`), lower-tailed (`side=L`) or two-tailed (`side=B`) are needed. The default is `side=B`.

Note: it is important to write the letters U, L and B as capitals.

With only these information it is not possible to analyze a data set, for example among other things the data set is not delivered to *%SimIntervals*. The complete invocation will be shown after the introduction of the *%SimTests* macro and the other three macros which both *%Sim** use.

3.2 The *%SimTests* macro

%SimTests performs the closed testing procedure which takes logical dependencies among hypotheses into account. Further on the macro allows any collection of linear combinations, not just pairwise contrasts. The critical values are generated via Monte Carlo simulation or via the Bonferroni method (the Bonferroni method is not available for *%SimIntervals*). Compared to SAS *%SimTests* has the following advantages:

- The procedure MULTTEST from SAS performs a closed testing procedure. However this procedure allows only one factor in the class-statement.
- SAS uses only the unconstrained step-down method of Holm instead of *%SimTests*, which uses the unconstrained step-down method of Holm and the logical constraint method of Shaffer.

Invocation

The *%SimTests* macro is invoked by:

```
%SimTests  (nsamp=      ,
            seed =      ,
            side =      ,
            type =     );
```

where

`nsamp` is the simulation size, with 20000 as the default. In the case of `nsamp` = 0 the macro calculates all adjustments using the Bonferroni method instead of the simulated critical values from the multivariate t distribution. However it still uses the methods of Shaffer (with `type = LOGICAL`) or Holm (with `type = FREE`).

`seed` is the random number seed. Again there is a default. If there is no entry made the computer clock time is used.

`side` determines whether upper-tailed (`side = U`), lower-tailed (`side = L`) or two-

tailed (`side = B`) are needed. The default is `side = B`.

Note: it is important to write the letters U, L and B as capitals.

`type` It can be chosen between two methods of the closed testing procedure. One is the unconstrained method of Holm (`type = FREE`) and the other one is the logical constrained method of Shaffer and Westfall (see Westfall (199) for details) (`type = LOGICAL`). Though `type = LOGICAL` is more powerful than `type = FREE`, the last one is the default because when there are more than 20 contrasts the computer needs much time to calculate. Although `type = FREE` is less powerful than `LOGICAL` it still uses step-down testing and correlations thus it is more powerful than single-step procedures. Note: the parameters `TYPE` and `LOGICAL` have to be written in capitals!If the parameters are written in small letters then SAS will print out a failure warning.

Note: if `side=U` and `type=LOGICAL` is entered in the invocation the procedure is no longer closed. However this can be still used, Westfall explains this: "The tests are technically still called closed two-sided in that case (with high power for the anticipated alternative), but since there seem to be no directional errors (...), it seems reasonable to make directional claims as well."

3.3 Invocation of the *%Sim** macros.

Up to now the macro is not able to analyze a data set because there is no option to select one or to choose in which way it has to be computed.
Both *%Sim** macros use three marcos which contain the data (*%Estimates*), the interesting contrasts (*%Contrasts*) or both (*%MakeGLMStats*). These macros can be used in several ways:

- The *%MakeGLMStats* macro creates the *%Estimates* macro for the degree of freedom, the lsmeans, the covariance matrix and the means square error (short: summary statistics) and the *%Contrasts* macro for the contrasts of interest automatically.

- The contrasts are created by the user and the summary statistics are computed by the macro *%MakeGLMStats*.
- With the summary statistics, the macros *%Estimates* and *%Contrasts* can be specified directly, writing SAS programming statements to identify the needed values.

The following three examples show the specifying of the macros by calculating multiple comparisons of means in the balanced one-way ANOVA using the *rust inhibitor* data set from page 4. It will be analyzed by an all-pair comparison. Supplementary there are additional ways to invoke the macros, these are shown in the next chapter among the analysis of various agricultural designs.

3.3.1 Using *%MakeGLMStats*

In the rare case of having a simple model we just use the macro *%MakeGLMStats* to create *%Estimates* and *%Contrasts* automatically. *%MakeGLMStats* is invoked as follows:

```
%MakeGLMStats  (dataset =      ,
               classvar =      ,
                 yvar   =      ,
                 model  =      ,
               contrasts=      );
```

where

`dataset` is the name of the data set. An input is necessary.

`classvar` is the listing of the `class`-variables. If there are two or more factors, they have to be separated by a space. Again this is a required input.

`yvar` is the endpoint (response variable). There's no default.

`model` is the statement `model` in SAS. However nested effects as in PROC GLM are not possible. Again there is no default.

`contrasts` define the contrasts of interest and the `class`-variables (factors) to which the contrasts are applied. One can choose among all-pair comparisons: `all`(`class`-variable), many-to-one comparisons: `control`(`class`-variable)

where the first level of the `class`-variables is the control. And the default is `user`. When `user` is specified then the macro *%Contrasts* has to be created by hand.

Invocation

Analyzing all-pair comparisons of means with the example *rust inhibitor* this would be:

```
%MakeGLMStats(dataset   = rust,
              classvar  = brand,
              yvar      = effectiveness,
              model     = brand,
              contrasts = all(brand));

%SimTests(seed=100177, type=LOGICAL);
%SimIntervals(seed=100177);
```

Output

%SimTests

```
Logically Constrained (Restricted Combinations) Step-Down Tests

                    Standard    ----- Pr > |t| -----
Contrast    Estimate    Error       Raw    Bon    Adj    SE(AdjP)

1-2         -46.3000   1.1081    <.0001 <.0001 <.0001           0
1-3         -24.8100   1.1081    <.0001 <.0001 <.0001           0
1-4           2.6700   1.1081    0.0212 0.0212 0.0212           0
2-3          21.4900   1.1081    <.0001 <.0001 <.0001           0
2-4          48.9700   1.1081    <.0001 <.0001 <.0001           0
3-4          27.4800   1.1081    <.0001 <.0001 <.0001           0
```

Interpretation

`Contrasts`	is the label of the contrasts. They can be defined by the *%Contrasts* macro (see below) or generated automatically by *%MakeGLMStats* (as in this example).
`Estimate`	is the difference of the lsmeans.
`Standard Error`	is the standard error of the difference.
`Pr> |t| Raw`	is the unadjusted (raw) p-value from the marginal t-distribution.
`Pr> |t| Bon`	is the p-value with Bonferroni adjustment.
`Pr> |t| Adj`	is the p-value with closed testing procedure adjustment of Holm or of Shaffer.
`SE(AdjP)`	is the standard error for the adjusted p-value.

In the example all differences between the brands are statistically significant. Note that Bonferroni adjusted and closed testing procedure adjusted p-value in one row are the same. In this case, it is the last step of the step-down procedure; thus no adjustment is needed.

%SimIntervals

```
Estimated 95% Quantile = 2.66997

                    Standard                --- Pr > |t| --      95% Confidence
Contrast    Estimate    Error t Value       Raw Adjusted         Interval

1-2         -46.3000   1.1081  -41.78    <.0001   <.0001    -49.2587 -43.3413
1-3         -24.8100   1.1081  -22.39    <.0001   <.0001    -27.7687 -21.8513
1-4           2.6700   1.1081    2.41    0.0212   0.0892     -0.2887   5.6287
2-3          21.4900   1.1081   19.39    <.0001   <.0001     18.5313  24.4487
2-4          48.9700   1.1081   44.19    <.0001   <.0001     46.0113  51.9287
3-4          27.4800   1.1081   24.80    <.0001   <.0001     24.5213  30.4387
```

Interpretation

`Estimated 95% Quantile`	is the simulated critical value from the multivariate t distribution.
`Contrast`	is the label of the contrast.
`Estimate`	is the difference of the lsmeans.
`Standard error`	is the standard error of the difference.
`t-value`	is the test statistic.
`Pr> |t| Raw`	is the unadjusted p-value.
`Pr> |t| Adjusted`	is the adjusted p-value.
95% `Confidence Interval`	is the adjusted 95% Confidence interval, commonly denoted as *simultaneous confidence intervals*, see Hochberg and Tamhane (1987, p.4) for details.

In contrast to *%SimTests* the comparison "1 vs. 4" is not statistically significant. However the output from *%SimIntervals* contains simultaneous confidence intervals. This can be a great advantage if we accept a less powerful method than the closed testing procedure. The intervals remain in the dimension of the endpoint. These two outputs are the standard ones, they are no modifications.

3.3.2 Using *%Contrasts* and *%MakeGLMStats*

As already mentioned the invocation via *%MakeGLMStats* can only be used when we are interested in either all-pairs or many-to-one comparisons. However in many situations we want to examine any combinations of contrasts. Then *%MakeGLMStats* and *%Contrasts* have to be used. In the macro *%Contrasts* the desired comparisons can be defined in the first part. Note that the first column of the **C** matrix contains only 0's. It belongs to the intercept parameter β_1 in the PROC GLM model. Note that all factors which are listed in the model statement of the *%MakeGLMStats* macro have to be specified in the *%Contrast* macro! In the second part of the macro *%Contrasts* the labels of the contrasts are defined.

Invocation

```
%MakeGLMStats(dataset   = rust,
              classvar  = brand,
              yvar      = effectiveness,
              model     = brand);

%macro Contrasts;
   C = { 0  1 -1  0  0 ,              /* The contrasts of interest. */
         0  1  0 -1  0 ,
         0  1  0  0 -1 ,
         0  0  1 -1  0 ,
         0  0  1  0 -1 ,
         0  0  0  1 -1 };
   C = C' ;

   Clab = {"1-2", "1-3", "1-4",       /* And the labels of the contrasts. */
                  "2-3", "2-4",
                         "3-4"};
%mend;
%SimTests(seed=100177, type=LOGICAL);
%SimIntervals(seed=100177);
```

We abandon an output because we get the same results as analyzing the data set with the macros *%MakeGLMStats*.

3.3.3 Using *%Contrasts* and *%Estimates*

In some situations PROC GLM is not sufficient for calculations, for example when random variables are included. Then PROC MIXED has to be used. This can be done by using the macro *%Estimates* to select the summary statistics from other procedures than PROC GLM. Again the contrasts have to be defined by *%Contrasts*, however we do not use

%MakeGLMStats to create the summary statistics thus there is no first column for the intercept. As already mentioned the summary statistics are set due *%Estimates*:

`EstPar` are the lsmeans.

`Mse` is the mean square error.

`Cov` is the covariance matrix. Here it is `Mse*I(4)/10` were `I` is the identity

matrix, which is multiplied by the number of groups. The denominator `10` is the sample size per group.

`df` is the degree of freedom.

All the parameters have to be declared. There are no defaults. The values can be taken from the outputs of PROC GLM / MIXED, the output delivery system (ODS) or somewhere else. Without the ODS it is not possible to analyze more than one factor while using *%Estimates* thus an analysis can be only done per factor. Therefore simultaneous adjustments for more than on factor are not possible.

Invocation

```
%macro Contrasts;
   C = { 1 -1  0  0 ,
         1  0 -1  0 ,
         1  0  0 -1 ,
         0  1 -1  0 ,
         0  1  0 -1 ,
         0  0  1 -1 };
   C = C‘ ;

   Clab = {"1-2", "1-3", "1-4",
                  "2-3", "2-4",
                         "3-4"};
%mend;

%macro Estimates;
   EstPar = { 43.14 , 89.44 , 67.95 , 40.47 };
   Mse    = 6.13983;
   Cov    = Mse * I(4)/10 ;
   df     = 36;
%mend;

%SimTests(seed=100177, type=LOGICAL);
%SimIntervals(seed=100177);
```

So all the three methods to invoke the *%Sim** macros will calculate the same results except slight deviations in the results from *%Estimates* and *%Contrasts* due to roundoff errors.

4 Invocation of *%SimIntervals* and *%SimTests*

In this section the invocation of the two macros *%Sim** for some agricultural designs will be shown. For most designs any pair-wise comparisons are used because this shows the advantage of the analysis of any combination of contrasts of the two macros in contrast to ordinary procedures of SAS. Supplementary for this kind of comparisons *%Contrasts* have to be used, not only *%MakeGLMStats*. For the following designs an invocation will be shown:

- one-way analysis of variance,
- two-way analysis of variance,
- three-way analysis of variance,
- randomized complete block design,
- Latin square design,
- Youden design,
- Hierarchical designs: split-plot design and
- Analysis of covariance.

If not stated otherwise an introduction of a design starts with some notes and the model is shown. Afterwards an example for the design is explained and the data set appendant to the example is provided. Then the invocation follows and in the end the output is shown with a brief interpretation. An FWE of 5% is used in all designs. Note that all calculations are valid for both balanced and unbalanced layouts if not stated otherwise. Unbalanced designs are problematic in current statistic programs, for example for SAS which uses the Tukey-Kramer test instead of the Tukey-test when analyzing unbalanced all-pairs comparisons.

4.1 One-way analysis of variance

Although we introduced already the one-way analysis of variance we show the invocation again for completion of this section. Supplementary we indicate another way to specify the macro *%Estimates* by using SAS programming code. As already mentioned in section 2 we deviate in this section from the general model and adapt it to the particular model. Thus our model for an one-way analysis of variance is:

$$y_{ij} = \mu + \alpha_i + \varepsilon_{ij}$$

where

y_{ij} = observation of replication j on level i of factor A,

μ = mean of the population,

α_i = effect of level i from factor A,

ε_{ij} = error term.

Moore (2000, p.524-525) uses a study where the effect of logging on the number of trees in the following years was studied. In the study forest plots in Borneo are compared. Some have never been logged and others have been logged for a year respectively eight years before the study.

Data

never	27	22	29	21	19	33	16	20	24	27	28	19
1 year	12	12	15	9	20	18	17	14	14	2	17	19
8 years	18	4	22	15	18	19	22	12	12	.	.	.

Invocation

```
%MakeGLMStats(dataset   = rainforest,
              classvar  = logged    ,
              yvar      = trees     ,
              model     = logged    );

%macro Contrasts;
   C =  { 0 -1  1  0 ,
          0 -1  0  1 ,
          0  0 -1  1 };
   C=C‘;
   clab = {"never-one", "never-eight", "one-eight"};
%mend;

%SimTests(seed=100177, type=LOGICAL);
%SimIntervals(seed=100177);
```

Alternatively we can use the following invocation. However this is just possible for a one-way design because the statement `cov` in the procedure GLM allows just one factor.

```
proc glm data=rainforest outstat=stat;
   class logged;
   model trees=logged;
   lsmeans logged /out=ests cov;
run;

%macro Contrasts;
   C =  { -1  1  0 ,
          -1  0  1 ,
           0 -1  1 };
   C=C‘;
   clab = {"never-one", "never-eight", "one-eight"};
%mend;

%macro Estimates;
 use ests;
 read all var {lsmean} into EstPar;
 read all var {cov1 cov2 cov3} into Cov;
 use stat (where=(_TYPE_=’ERROR’));
 read all var {df} into df;
%mend;

%SimTests(seed=100177, type=LOGICAL);
%SimIntervals(seed=100177);
```

Output

%SimIntervals

```
Estimated 95% Quantile = 2.469223
```

Contrast	Estimate	Standard Error	t Value	--- Pr > |t| -- Raw	Adjusted	95% Confidence Interval	
never-one	1.6944	2.3064	0.73	0.4682	0.7435	-4.0006	7.3895
never-eight	9.6667	2.1353	4.53	<.0001	0.0002	4.3941	14.9392
one-eight	7.9722	2.3064	3.46	0.0017	0.0038	2.2772	13.6672

Only the difference between "never" and "one year ago" logged are not significant because the p-value is greater than 5% and the confidence interval includes 0. In comparison to "eight" the number of trees in "never" is at least 4.4 and up to 14.94 greater and the number of "one" is at least 2.28 and up to 13.67 greater than in "eight".

%SimTests

```
Logically Constrained (Restricted Combinations) Step-Down Tests
```

Contrast	Estimate	Standard Error	----- Pr > |t| ----- Raw	Bon	Adj	SE(AdjP)
never-one	1.6944	2.3064	0.4682	0.4682	0.4682	0
never-eight	9.6667	2.1353	<.0001	0.0003	0.0003	0
one-eight	7.9722	2.3064	0.0017	0.0017	0.0017	0

As we can see the p-values calculated by *%SimTests* are less than the ones computed by *%SimIntervals*. But "never-one" still shows no statistically significant difference in the number of trees. There are no confidence intervals because *%SimTests* calculates a closed testing procedure.

4.2 Two-way analysis of variance

More common than one-way designs are multi-way classifications. Therefore a two-way ANOVA and after that, in 4.3, a three-way ANOVA are shown. How to invoke the macros for a two-way (unbalanced) ANOVA will be shown by analyzing the design with and without interaction term and afterwards the same calculations with the interaction term will be done by the cell means model. At last interaction contrasts for a two-way ANOVA are shown. For all computations except the interaction contrasts a data set from Köhler *et al.* (1996, p.139-140) is used. Thus we will provide the notes to the example and its data before showing the model: In a greenhouse experiment the effect of two types of fertilizer and of three types of pesticide on the yield [kg] of wine is studied.

Data

pesticide	fertilizer		
	F1	F2	F3
P1	21.3	22.3	23.8
	20.9	21.6	23.7
	20.4	21.0	22.6
P2	12.7	12.0	14.5
	14.9	14.2	16.7
	12.9	12.1	14.5

4.2.1 Without interaction term

The model for a two-way analysis of variance without interaction is:

$$y_{ijk} = \mu + \alpha_i + \beta_j + \varepsilon_{ijk},$$

where

y_{ijk} = observation of replication k on level i of factor A and level j of factor B,

μ = mean of the population,

α_i = effect of level i of factor A,

β_j = effect of level j of factor B,

ε_{ijk} = error term.

Invocation

The specification of the contrasts of *%Contrasts* is dependent from the model statement of *%MakeGLMStats*. As already shown, in the first part of the macro *%Contrasts* the first row is the mean of the population. Here the second and the third row belong to factor A and the fourth, fifth and sixth row belong to factor B. When levels of factor A are compared then the levels of factor B are set to 0. Although they are not part of the comparison the levels of factor B are needed for the calculation of the least square means (see 4.2.2 for more details.

In the following tabular the contrasts for the macro *%Contrasts* are shown:

comparison	intercept	factor A (fertilizer)		factor B (pesticide)		
	μ	α_1	α_2	β_1	β_2	β_3
factor A: 1 vs. 2	0	1	-1	0	0	0
factor B: 1 vs. 2	0	0	0	1	-1	0
factor B: 1 vs. 3	0	0	0	1	0	-1
factor B: 2 vs. 3	0	0	0	0	1	-1

Thus the invocation of the *%Sim** is:

```
%MakeGLMStats(dataset= twoway              ,
           classvar  = fertilizer pesticide,
           yvar      = yield              ,
           model     = fertilizer pesticide);

%macro Contrasts;
   C =  {0 1 -1     0  0  0,
         0 0  0     1 -1  0,
         0 0  0     1  0 -1,
         0 0  0     0  1 -1};

   C = C' ;

   Clab = {"fert1-fert2",
           "pest1-pest2",     "pest1-pest3",     "pest2-pest3"};
%mend;

%SimTests(seed=100177, type=LOGICAL);
%SimIntervals(seed=100177);
```

Output

%SimIntervals

```
Estimated 95% Quantile = 2.749074

                        Standard            --- Pr > |t| --      95% Confidence
Contrast       Estimate    Error t Value       Raw Adjusted         Interval

fert1-fert2      8.1222   0.4560   17.81    <.0001   <.0001      6.8686   9.3759
pest1-pest2     -0.0167   0.5585   -0.03    0.9766   1.0000     -1.5520   1.5187
pest1-pest3     -2.1167   0.5585   -3.79    0.0020   0.0070     -3.6520  -0.5813
pest2-pest3     -2.1000   0.5585   -3.76    0.0021   0.0074     -3.6354  -0.5646
```

From the output we can see that the difference between the two fertilizers is statistically significant. Type 1 results in a yield which is at least 6.87kg and up to 9.38kg greater than the yield of type 2. There is no significant difference between the types 1 and 2 of the pesticides because the confidence interval includes 0. The two other comparisons of pesticides, 1 vs. 3 and 2 vs. 3, are significant. The α is controlled for both factors

simultaneous. If this is not desired each factor has to be analyzed alone thus for each one the macros have to be invoked.

%SimTests

```
Logically Constrained (Restricted Combinations) Step-Down Tests

                        Standard    ----- Pr > |t| -----
Contrast       Estimate    Error       Raw    Bon    Adj    SE(AdjP)

fert1-fert2      8.1222   0.4560    <.0001 <.0001 <.0001           0
pest1-pest2     -0.0167   0.5585    0.9766 0.9766 0.9766           0
pest1-pest3     -2.1167   0.5585    0.0020 0.0060 0.0053    0.000128
pest2-pest3     -2.1000   0.5585    0.0021 0.0060 0.0053    0.000128
```

As expected the p-values are less than the ones of *%SimIntervals*. However the decisions stay the same.

4.2.2 With interaction term

In most cases a two-way ANOVA will be analyzed when the interaction term is included. The model is:

$$y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{ijk},$$

where

y_{ijk} = observation of replication k on level i of factor A and level j of factor B,

μ = mean of the population,

α_i = effect of level i of factor A,

β_j = effect of level j of factor B,

$(\alpha\beta)_{ij}$ = interaction between level i of factor A and level j of factor B,

ε_{ijk} = error term.

Invocation

By adding the interaction term the number of coefficients of the contrasts for the factors A and B have to be altered. When averaged for the three pesticides the average gain for the first fertilizer is

$$E(y_{1j}) = \mu + \alpha_1 + \frac{1}{3}(\beta_1 + \beta_2 + \beta_3) + \frac{1}{3}\{(\alpha\beta)_{11} + (\alpha\beta)_{12} + (\alpha\beta)_{13}\}.$$

And for the second fertilizer the average gain is

$$E(y_{2j}) = \mu + \alpha_2 + \frac{1}{3}(\beta_1 + \beta_2 + \beta_3) + \frac{1}{3}\{(\alpha\beta)_{21} + (\alpha\beta)_{22} + (\alpha\beta)_{23}\}.$$

Thus the difference between them is:

$$E(y_{1j} - y_{2j}) = \alpha_1 - \alpha_2 + \frac{1}{3}\{(\alpha\beta)_{11} + (\alpha\beta)_{12} + (\alpha\beta)_{13} - (\alpha\beta)_{21} - (\alpha\beta)_{22} - (\alpha\beta)_{23}\}.$$

This is the combination which has to be specified to compute the contrasts. Alternatively the contrast statements can be standardized to yield integer coefficients:

$$E(3y_{1j} - 3y_{2j}) = 3(\alpha_1 - \alpha_2) + (\alpha\beta)_{11} + (\alpha\beta)_{12} + (\alpha\beta)_{13} - (\alpha\beta)_{21} - (\alpha\beta)_{22} - (\alpha\beta)_{23},$$

which is used in this thesis. Thus the contrasts for the example are specified as in the following table:

comparison	intercept	factors				
		A (fertilizer)		B (pesticide)		
	μ	α_1	α_2	β_1	β_2	β_3
A: 1 vs. 2	0	3	−3	0	0	0
B: 1 vs. 2	0	0	0	2	-2	0
B: 1 vs. 3	0	0	0	2	0	-2
B: 2 vs. 3	0	0	0	0	2	-2

comparison	interactions					
	$(\alpha\beta)_{11}$	$(\alpha\beta)_{12}$	$(\alpha\beta)_{13}$	$(\alpha\beta)_{21}$	$(\alpha\beta)_{22}$	$(\alpha\beta)_{23}$
A: 1 vs. 2	1	1	1	-1	-1	-1
B: 1 vs. 2	1	-1	0	1	-1	0
B: 1 vs. 3	1	0	-1	1	0	-1
B: 2 vs. 3	0	1	-1	0	1	-1

Thus the invocation is:

```
%MakeGLMStats(dataset= twoway                                    ,
           classvar  = fertilizer pesticide                      ,
           yvar      = yield                                     ,
           model     = fertilizer pesticide fertilizer*pesticide);

%macro Contrasts;
   C =  {0  3 -3    0  0  0   1  1  1 -1 -1 -1};
   C = C/3;

   C1 = {0  0  0    2 -2  0   1 -1  0  1 -1  0,
         0  0  0    2  0 -2   1  0 -1  1  0 -1,
         0  0  0    0  2 -2   0  1 -1  0  1 -1};
   C1 = C1/2;

   C = C//C1;
C = C‘ ;
 Clab = {"fert1-fert2",
         "pest1-pest2",     "pest1-pest3",     "pest2-pest3"};
%mend;

%SimTests(seed=100177, type=LOGICAL);
%SimIntervals(seed=100177);
```

Output

%SimIntervals

```
Estimated 95% Quantile = 2.80968

                         Standard            --- Pr > |t| --      95% Confidence
Contrast       Estimate    Error t Value       Raw Adjusted         Interval

fert1-fert2      8.1222   0.4597   17.67    <.0001   <.0001      6.8305   9.4139
pest1-pest2     -0.0167   0.5631   -0.03    0.9769   1.0000     -1.5987   1.5654
pest1-pest3     -2.1167   0.5631   -3.76    0.0027   0.0089     -3.6987  -0.5346
pest2-pest3     -2.1000   0.5631   -3.73    0.0029   0.0092     -3.6820  -0.5180
```

By adding the interaction the p-values become larger (Although the sum of squares of the error term is less in the design containing the interaction term the degrees of freedom are less, too. Thus the mean square error in the design with the interaction term is larger than in the design without the interaction.).

%SimTests

```
Logically Constrained (Restricted Combinations) Step-Down Tests

                        Standard    ----- Pr > |t| -----
Contrast       Estimate    Error       Raw    Bon    Adj    SE(AdjP)

fert1-fert2     8.1222   0.4597    <.0001 <.0001 <.0001           0
pest1-pest2    -0.0167   0.5631    0.9769 0.9769 0.9769           0
pest1-pest3    -2.1167   0.5631    0.0027 0.0082 0.0071    0.000168
pest2-pest3    -2.1000   0.5631    0.0029 0.0082 0.0071    0.000168
```

Also the p-values computed by the *%SimIntervals* macro are equal to or greater than by the *%SimTests* macro. However the decisions stay the same.

4.2.3 Using the cell means model

The same results can be calculated by a different model representation, the *cell means model* (see e.g. Dean and Voss (1999, p.138-139) for further details). The cell means model for a two-way ANOVA is

$$y_{ijk} = \mu + \tau_{ij} + \varepsilon_{ijk},$$

where

y_{ijk} = observation of replication k on level i of factor A and level j of factor B,

μ = parametric mean of the population,

τ_{ij} =treatment combination ij, which is the sum of the individual effect of the two, factors and their joint interaction,

ε_{ijk} = error term.

The name "cell" refers to a cell in a tabular where the levels of factor A are the rows and the columns represent the levels of factor B. Although we get the same results as using the "ordinary" ANOVA representation, the invocation is different. In the macro *%MakeGLMStats* the `model` statement comprises only the interaction term thus in the macro *%Contrasts* just the interaction term has to be written, that is

```
%MakeGLMStats(dataset  = twoway              ,
              classvar  = fertilizer pesticide,
              yvar      = yield              ,
              model     = fertilizer*pesticide);

%macro Contrasts;
   C =  {0  1  1  1 -1 -1 -1};
   C = C/3;

   C1 = {0  1 -1  0  1 -1  0,
         0  1  0 -1  1  0 -1,
         0  0  1 -1  0  1 -1};
   C1 = C1/2;

   C = C//C1;
C = C' ;

 Clab = {"fert1-fert2",
         "pest1-pest2",     "pest1-pest3",     "pest2-pest3"};
%mend;


%SimTests(seed=100177, type=LOGICAL);
%SimIntervals(seed=100177);
```

4.2.4 Interaction Contrasts

When a model contains more than one factor, these factors may have an influence on each other. This interaction can be measured when the effect of one factor depends on the level of the other factor. The ordinary procedures of SAS provide a general predication whether the model contains interactions. However it is difficult to perform pairwise differences for one factor at different levels of the other factor, which can be done by using the *%Sim** macros. Although there are several types of these interaction contrasts we use only the tetrad differences. Further types of interaction contrasts as the product-type interaction contrasts or generalized interaction contrasts are described by Hochberg and Tamhane (1987 p.294-299). Tetrades have the form

$$(\mu_{ij} - \mu_{ij'}) - (\mu_{i'j} - \mu_{i'j'}).$$

As an example we use a data set according to Dean and Voss (1999, p.98). A stimulus was presented to a subject and the reaction time which was needed to press a key was measured. However the subject was warned that the stimulus was followed by an auditory or visual cue. We are interested in the interaction contrasts of the factor A (type of cues: auditory or visual) and of the factor B (the elapsed times between cue and stimulus: 5, 10 or 15 seconds).

Data

cue	Stimulus in seconds		
	5	10	15
auditory	0.204	0.182	0.202
	0.17	0.167	.
	.	0.187	.
visual	0.257	0.235	0.258
	0.279	0.26	0.281
	.	0.283	0.256

Invocation

```
%MakeGLMStats(dataset  = time          ,
              classvar = stimulus time ,
              yvar     = reaction      ,
              model    = stimulus*time);

%macro Contrasts;
   C = {0    1 -1  0 -1  1  0,
        0    1  0 -1 -1  0  1,
        0    0  1 -1  0 -1  1};
   C = C' ;

   Clab = {"(11-12)-(21-22)",
           "(11-13)-(21-23)",
           "(12-13)-(22-23)"};
%mend;

%SimTests(seed=100177, type=LOGICAL);
%SimIntervals(seed=100177);
```

Westfall *et al.* (1999, p.182) use another way to invoke *%Contrasts*. Although this way seems to be more complicated only the number of the levels per factor between the macros *MakeGLMStats* and *Contrasts* have to be adapted for a data set. In our example factor A has two levels, (`%let a =2`) and factor B has three levels (`%let b = 3`).

```
%MakeGLMStats(dataset  = time          ,
              classvar = stimulus time ,
              yvar     = reaction      ,
              model    = stimulus*time);

%let a=2;   /* Levels of first CLASS variable */
%let b=3;   /* Levels or second CLASS variable */

%macro Contrasts;
   start tlc(n); return(trim(left(char(n,20)))); finish;

   idi=(1:&a);
   idj=(1:&b);
   free C clab;
   do i1=1 to &a-1; do i2=i1+1 to &a;
      do j1=1 to &b-1; do j2=j1+1 to &b;
         C    = C    // (0 || ( ((idi=i1) - (idi=i2))
                              @((idj=j1) - (idj=j2))));
         clab = clab //   "("+tlc(i1)+tlc(j1)+"-"+tlc(i1)+tlc(j2)+")"
                        +"-("+tlc(i2)+tlc(j1)+"-"+tlc(i2)+tlc(j2)+")";
      end; end;
   end; end;
   C=C';
%mend;

%SimTests(seed=100177, type = LOGICAL);
%SimIntervals(seed=100177);
```

Output

%SimIntervals

```
Estimated 95% Quantile = 2.860682

                              Standard                --- Pr > |t| --      95% Confidence
Contrast            Estimate     Error t Value       Raw Adjusted          Interval

(11-12)-(21-22)    -0.00033    0.0232   -0.01    0.9889   0.9999     -0.0666   0.0659
(11-13)-(21-23)     -0.0180    0.0274   -0.66    0.5298   0.7903     -0.0964   0.0604
(12-13)-(22-23)     -0.0177    0.0254   -0.70    0.5060   0.7698     -0.0903   0.0549
```

%SimTests

```
Logically Constrained (Restricted Combinations) Step-Down Tests

                          Standard    ----- Pr > |t| -----
Contrast           Estimate    Error       Raw    Bon    Adj    SE(AdjP)

(11-12)-(21-22)    -0.00033   0.0232    0.9889 1.0000 0.9889           0
(11-13)-(21-23)     -0.0180   0.0274    0.5298 1.0000 0.7717     0.00163
(12-13)-(22-23)     -0.0177   0.0254    0.5060 1.0000 0.7717     0.00163
```

Neither *%SimIntervals* nor *%SimTests* show significant interactions because the p-values are greater than the α of 5%.

4.3 Three-way analysis of variance

As already mentioned multi-way classifications are more common as agricultural designs than one-way classifications. Thus a three-way balanced ANOVA with interactions is presented. If an experiment contains more than two factors the analysis via the *%Sim** macros follows the same principles as a two-way analysis of variance. The model for a three-way analysis of variance with three two-factorial and one three-factorial interaction is:

$$y_{ijmk} = \mu + \alpha_i + \beta_j + \gamma_m + (\alpha\beta)_{ij} + (\alpha\gamma)_{im} + (\beta\gamma)_{jm} + (\alpha\beta\gamma)_{ijm} + \varepsilon_{ijkm},$$

where

y_{ijmk} = observation of replication k on level i of factor A, level j of factor B and level m of factor C,
μ = mean of the population,
α_i = effect of level i of factor A,
β_j = effect of level j of factor B,
γ_m = effect of level m of factor C,
$(\alpha\beta)_{ij}$ = interaction between level i of factor A and level j of factor B,
$(\alpha\gamma)_{im}$ = interaction between level i of factor A and level m of factor C,
$(\beta\gamma)_{jm}$ = interaction between level j of factor B and level m of factor C,
$(\alpha\beta\gamma)_{ijm}$ = interaction between level i of factor A, level j of factor B and level m of factor C,
ε_{ijmk} = error term.

According to Neter *et al.* (1996, 942-943) the following data set is used as the example. In an experiment the following effects on the exercise time of a subject were measured: body fat (factor A), smoking history (factor B) and gender of the subject (factor C). The exercise time is the time [minutes] until a subject, who is performing on a bicycle apparatus, gets tired. Note that all three factors have two levels thus it is actually a 2^3 factorial layout.

Data

body fat	gender	smoking history	
		light	heavy
low fat	male	24.1	17.6
		29.2	18.8
		24.6	23.2
	female	20.0	14.8
		21.9	10.3
		17.6	11.3
high fat	male	14.6	14.9
		15.3	20.4
		12.3	12.8
	female	16.1	10.1
		9.3	14.4
		10.8	6.1

Invocation

To demonstrate the specification of a contrast the levels 1 (low) and 2 (heavy) of factor A (body fat) are compared. The average time until tiredness for level 1 is:

$$E(y_{1jm}) = \mu + \alpha_1 + \frac{1}{2}(\beta_1 + \beta_2) + \frac{1}{2}(\gamma_1 + \gamma_2)$$
$$+\frac{1}{8}\{(\alpha\beta)_{11} + (\alpha\beta)_{12} + (\alpha\gamma)_{11} + (\alpha\gamma)_{12} + (\alpha\beta\gamma)_{111} + (\alpha\beta\gamma)_{112} + (\alpha\beta\gamma)_{121} + (\alpha\beta\gamma)_{122}\}.$$

And the average time for level 2 is:

$$E(y_{2jm}) = \mu + \alpha_2 + \frac{1}{2}(\beta_1 + \beta_2) + \frac{1}{2}(\gamma_1 + \gamma_2)$$
$$+\frac{1}{8}\{(\alpha\beta)_{21} + (\alpha\beta)_{22} + (\alpha\gamma)_{21} + (\alpha\gamma)_{22} + (\alpha\beta\gamma)_{211} + (\alpha\beta\gamma)_{212} + (\alpha\beta\gamma)_{221} + (\alpha\beta\gamma)_{222}\}.$$

The difference of the two equations is:

$$E(y_{1jm} - y_{2jm}) = \alpha_1 - \alpha_2$$
$$+\frac{1}{8}\{(\alpha\beta)_{11} + (\alpha\beta)_{12} + (\alpha\gamma)_{11} + (\alpha\gamma)_{12} + (\alpha\beta\gamma)_{111} + (\alpha\beta\gamma)_{112} + (\alpha\beta\gamma)_{121} + (\alpha\beta\gamma)_{122}$$
$$-(\alpha\beta)_{21} - (\alpha\beta)_{22} - (\alpha\gamma)_{21} - (\alpha\gamma)_{22} - (\alpha\beta\gamma)_{211} - (\alpha\beta\gamma)_{212} - (\alpha\beta\gamma)_{221} - (\alpha\beta\gamma)_{222}\}.$$

Alternatively it can be written as:

$$
\begin{aligned}
E(8y_{1jm} - 8y_{2jm}) &= 8\alpha_1 - 8\alpha_2 \\
&+(\alpha\beta)_{11} + (\alpha\beta)_{12} + (\alpha\gamma)_{11} + (\alpha\gamma)_{12} + (\alpha\beta\gamma)_{111} + (\alpha\beta\gamma)_{112} + (\alpha\beta\gamma)_{121} + (\alpha\beta\gamma)_{122} \\
&-(\alpha\beta)_{21} - (\alpha\beta)_{22} - (\alpha\gamma)_{21} - (\alpha\gamma)_{22} - (\alpha\beta\gamma)_{211} - (\alpha\beta\gamma)_{212} - (\alpha\beta\gamma)_{221} - (\alpha\beta\gamma)_{222},
\end{aligned}
$$

which is used here. However in the macros only the three-way interaction term is specified. Although the same p-values will be calculated by using all four interaction terms in the *%Contrasts* macro, the differences of the lsmeans and the standard errors are incorrect. Thus the contrasts for the example are specified as in the following tables:

Main factors

comparison	intercept	factor A		factor B		factor C	
		α_1	α_2	β_1	β_2	γ_1	γ_2
$fat_h - fat_l$	0	4	-4	0	0	0	0
$smoke_h - smoke_l$	0	0	0	4	-4	0	0
$gender_f - gender_m$	0	0	0	0	0	4	-4

Three-factorial interaction

comparison	interaction terms							
	$(\alpha\beta\gamma)_{111}$	$(\alpha\beta\gamma)_{112}$	$(\alpha\beta\gamma)_{121}$	$(\alpha\beta\gamma)_{122}$	$(\alpha\beta\gamma)_{211}$	$(\alpha\beta\gamma)_{212}$	$(\alpha\beta\gamma)_{221}$	$(\alpha\beta\gamma)_{222}$
$fat_h - fat_l$	1	1	1	1	-1	-1	-1	-1
$smoke_h - smoke_l$	1	1	-1	-1	1	1	-1	-1
$gender_f - gender_m$	1	-1	1	-1	1	-1	1	-1

Thus the invocation of the macros is:

```
%MakeGLMStats(dataset  = threeway                          ,
              classvar = fat smoke gender                  ,
              yvar     = minutes                           ,
              model    = fat smoke gender fat*smoke*gender);
%macro Contrasts;
   C =  {0   4 -4    0  0    0  0   1  1  1  1 -1 -1 -1 -1};
   C = C/4;

   C1 = {0   0  0    4 -4    0  0   1  1 -1 -1  1  1 -1 -1};
   C1 = C1/4;

   C2 = {0   0  0    0  0    4 -4   1 -1  1 -1  1 -1  1 -1};
   C2 = C2/4;


C = C//C1//C2;
C = C‘;

   Clab = {"fat_h-fat_l",   "smoke_h-smoke_l",   "gender_f-gender_m"};
%mend;

%SimTests(seed=100177, type=LOGICAL);
%SimIntervals(seed=100177);
```

As already shown in 4.2.3 another model representation, the cell means model, can be used. This cell means model can be applied here, too. However the results would be the same except the Monte Carlo error.

Output

%SimIntervals

```
Estimated 95% Quantile = 2.625904
```

Contrast	Estimate	Standard Error	t Value	--- Pr > |t| -- Raw	Adjusted	95% Confidence Interval	
fat_h-fat_l	-6.3583	1.2474	-5.10	0.0001	0.0004	-9.6338	-3.0829
smoke_h-smoke_l	-3.4250	1.2474	-2.75	0.0144	0.0391	-6.7004	-0.1496
gender_f-gender_m	-5.4250	1.2474	-4.35	0.0005	0.0013	-8.7004	-2.1496

As it can be seen from the p-values, which are all less than 5%, or from the confidence intervals, they exclude the 0, all comparisons are significant.

%SimTests

```
Logically Constrained (Restricted Combinations) Step-Down Tests
```

Contrast	Estimate	Standard Error	----- Pr > |t| ----- Raw	Bon	Adj	SE(AdjP)
fat_h-fat_l	-6.3583	1.2474	0.0001	0.0003	0.0003	0
smoke_h-smoke_l	-3.4250	1.2474	0.0144	0.0144	0.0144	0
gender_f-gender_m	-5.4250	1.2474	0.0005	0.0010	0.0010	0

Again the powerful macro *%SimTests* calculates p-values which are less than the ones of *%SimIntervals*.

4.4 Randomized complete block design

In the randomized complete block design the experimental units are grouped in blocks thus every level of the factor is represented with one ore more units in a block. The block factor is used as an adjustment against possible heterogeneities. As Piepho (2000, p.103) describes the goal is to minimize the variance within and to maximize it between the blocks. From that it follows a reduction of the mean square error. The model of the randomized complete block design is equivalent to the model of the two-way ANOVA (see 4.2). However the randomized complete block design is common in agricultural experiments and for completion of this thesis we show the analysis of this design. The model is:

$$y_{ij} = \mu + \alpha_i + b_j + \varepsilon_{ijk},$$

where

y_{ijk} = observation of replication k of level i of the factor and level j of the block,
μ = mean of the population,
α_i = effect of level i of the factor,
b_j = effect of level j of the block,
ε_{ijk} = error term.

In an experiment of Gomez *et al.* (1984, p.164) the effect of six sowing densities on the yield [kg/ha] of a rice type is studied. Every sowing density appears once in one of the four blocks.

Data

sowing	block			
density	1	2	3	4
25	5113	5398	5307	4678
50	5346	5952	4719	4264
75	5272	5713	5483	4749
100	5164	4831	4986	4410
125	4804	4848	4432	4748
150	5254	4542	4919	4098

Invocation

To demonstrate the construction of the contrasts the average yield over all blocks of the sowing densities 25[kg/ha] and 50[kg/ha] are shown. The average yield of the first sowing density (25[kg/ha]) is:

$$y_{1j} = \mu + \alpha_1 + \frac{1}{4}(b_1 + b_2 + b_3 + b_4)$$

And the average yield of the second one (50[kg/ha] is :

$$y_{2j} = \mu + \alpha_2 + \frac{1}{4}(b_1 + b_2 + b_3 + b_4)$$

The difference of the two equations is:

$$y_{1j} - y_{2j} = \alpha_1 - \alpha_2$$

which is used here. Therefore the contrasts are:

compa-	inter-	Levels: factor						Levels: block			
rison	cept	α_1	α_2	α_3	α_4	α_5	α_6	b_1	b_2	b_3	b_4
50 - 25	0	-1	1	0	0	0	0	0	0	0	0
75 - 50	0	0	-1	1	0	0	0	0	0	0	0
⋮	⋮										
150 -125	0	0	0	0	0	-1	1	0	0	0	0

Though the block factor does not appear in the calculations of the lsmeans it is listed in the *%Contrasts* macro. The block factor is needed to reduce the mean square error. Thus it is specified in the model statement of the *%MakeGLMStats* macro and therefore it appears in the *%Contrasts* macro. So the invocation is:

```
%MakeGLMStats(dataset  = fixedblock   ,
              classvar = density block,
              yvar     = yield        ,
              model    = density block);

%macro Contrasts;
   C =  {0  -1  1  0  0  0  0      0 0 0 0,
         0   0 -1  1  0  0  0      0 0 0 0,
         0   0  0 -1  1  0  0      0 0 0 0,
         0   0  0  0 -1  1  0      0 0 0 0,
         0   0  0  0  0 -1  1      0 0 0 0};


C = C';

   Clab = {"50-25",   "75-50",   "100-75",
           "125-100", "150-125"};
%mend;

%SimTests(seed=100177, type=LOGICAL);
%SimIntervals(seed=100177);
```

As described on page 24 all factors listed in the `model` statement have to appear in the *Contrasts* macro. If the last four rows, which belong to the block factor, are omitted, then SAS will print out a failure warning.

Output

%SimIntervals

```
Estimated 95% Quantile = 3.031517
```

Contrast	Estimate	Standard Error	t Value	--- Pr > |t| -- Raw	Adjusted	95% Confidence Interval	
50-25	-19.4845	264.1	-0.07	0.9426	1.0000	-820.1	781.1
75-50	243.4	266.5	0.91	0.3826	0.8454	-564.5	1051.2
100-75	-314.7	277.8	-1.13	0.2838	0.7246	-1156.8	527.5
125-100	-271.3	292.2	-0.93	0.3750	0.8379	-1157.2	614.6
150-125	175.0	292.2	0.60	0.5626	0.9649	-710.9	1060.9

As we can see from the p-values or confidence intervals none of the comparisons show a statistical significance.

%SimTests

Logically Constrained (Restricted Combinations) Step-Down Tests

Contrast	Estimate	Standard Error	Pr > |t| Raw	Bon	Adj	SE(AdjP)
50-25	-19.4845	264.1	0.9426	1.0000	0.9426	0
75-50	243.4	266.5	0.3826	1.0000	0.7977	0.00204
100-75	-314.7	277.8	0.2838	1.0000	0.7253	0.00217
125-100	-271.3	292.2	0.3750	1.0000	0.7977	0.00204
150-125	175.0	292.2	0.5626	1.0000	0.8038	0.00174

Again the differences are not significant, which is a matter of course because the raw p-values are not significant, too.

4.5 Latin square design

The randomized complete block design is used to eliminate one disturbing effect. If two of these effects appear then row-column-designs are used. Two of them are shown here, the first one is called Latin Square design and the second one, shown in the next subsection, is the Youden design. The difference between them is the number of levels per block. In a Latin Square design every level of the factor is represented once in each block. Whereas in a Youden design the blocks are incomplete, see page 4.6 for details. To analyze the design the following model is used:

$$y_{ijhk} = \mu + \alpha_i + b_j + c_h + \varepsilon_{ijhk},$$

where

y_{ijhk} = observation of the replication k of level i of factor A, level j of row B and level
= h of column C,
μ = mean of the population,
α_i = effect of level i of factor A,
b_j = effect of level j of row B, which is represented as rows,
c_h = effect of level h of column C, which is represented as columns,
ε_{ijhk} = error term.

Piepho (2000, p.175) uses an example where the contents of bacteria in milk of five companies were compared. The content was measured on five different days and day times, both of them are disturbing effects.

Data

	Day				
Daytime	1	2	3	4	5
08:30	A	B	C	D	E
	1.9	1.2	0.7	2.2	2.3
10:00	D	C	E	B	A
	2.3	2.0	0.6	2.6	2.3
11:30	C	A	D	E	B
	2.1	1.5	1.7	1.1	3.0
14:00	B	E	A	C	D
	2.9	1.1	1.2	1.8	2.6
15:30	E	D	B	A	C
	1.8	2.1	2.0	2.4	2.5

Invocation

If we are interested in comparisons company 1 vs. 2, 2 vs. 5 and 3 vs. 4 then the invocation is:

```
%MakeGLMStats(dataset  = latinsq           ,
              classvar = company row column,
              yvar     = content           ,
              model    = company row column);

%macro Contrasts;
   C =  {0   1 -1  0  0  0    0  0  0  0  0     0  0  0  0  0,
         0   0  1  0  0 -1    0  0  0  0  0     0  0  0  0  0,
         0   0  0  1 -1  0    0  0  0  0  0     0  0  0  0  0};

C = C';

   Clab = {"1-2",   "2-5",   "3-4"};
%mend;

%SimTests(seed=100177, type=LOGICAL);
%SimIntervals(seed=100177);
```

Output

%SimIntervals

```
Estimated 95% Quantile = 2.699672
```

Contrast	Estimate	Standard Error	t Value	--- Pr > |t| -- Raw	Adjusted	95% Confidence Interval	
1-2	-0.4800	0.2185	-2.20	0.0484	0.1227	-1.0700	0.1100
2-5	0.9600	0.2185	4.39	0.0009	0.0027	0.3700	1.5500
3-4	-0.3600	0.2185	-1.65	0.1254	0.2993	-0.9500	0.2300

Only the contents of company 2 and 5 show a significant difference. The p-values of the two other comparisons are greater than the α.

%SimTests

```
Logically Constrained (Restricted Combinations) Step-Down Tests
```

Contrast	Estimate	Standard Error	----- Pr > |t| ----- Raw	Bon	Adj	SE(AdjP)
1-2	-0.4800	0.2185	0.0484	0.0969	0.0923	0.000427
2-5	0.9600	0.2185	0.0009	0.0026	0.0025	0.000087
3-4	-0.3600	0.2185	0.1254	0.1254	0.1254	0

Using the *%SimTests* macro the decisions stay the same. Note that the comparison "2-5" should have in both analyses the same adjusted p-value. However they differ because of the Monte Carlo error.

4.6 Youden design

As already mentioned in 4.5 the Youden and the Latin square design are used to eliminate two disturbing factors via two blocks. In certain cases it may be impossible to generate complete blocks where every level of a factor is represented in each row and column. Here the Latin square is not the appropriate design. However the Youden design is a special row-column-design where the rows or columns are incomplete blocks. The model is:

$$y_{ijhk} = \mu + \alpha_i + b_j + c_h + \varepsilon_{ijhk},$$

where

y_{ijhk} = observation of replication k of the level i of factor A, level j of row B and level
= h of column C,

μ = parametric mean of the population,

α_i = effect of level i of factor A,

b_j = effect of level j of row B, which is represented as rows,

c_h = effect of level h of column C, which is represented as columns,

ε_{ijhk} = error term.

Piepho (2000, p.178) uses an example with five treatments, five rows and four columns.

Data

Row (Block)	Column 1	2	3	4
1	B	D	C	E
	158	107	94	68
2	D	A	E	B
	140	156	106	130
3	E	B	A	C
	141	155	99	97
4	C	E	D	A
	156	126	99	86
5	A	C	B	D
	136	140	131	102

Invocation

If we are interested in the comparisons 1 vs. 3, 1 vs. 4, 3 vs. 4 and 4 vs. 5 then the invocation will be:

```
%MakeGLMStats(dataset  = youden        ,
              classvar = trt row column,
              yvar     = y             ,
              model    = trt row column);

%macro Contrasts;
   C =  {0   1  0 -1  0  0   0  0  0  0  0    0  0  0  0,
         0   1  0  0 -1  0   0  0  0  0  0    0  0  0  0,
         0   0  0  1 -1  0   0  0  0  0  0    0  0  0  0,
         0   0  0  0  1 -1   0  0  0  0  0    0  0  0  0};

C = C';

   Clab = {"trt1-trt3",   "trt1-trt4",
           "trt3-trt4",   "trt4-trt5"};
%mend;

%SimTests(seed=100177, type=LOGICAL);
%SimIntervals(seed=100177);
```

Output

%SimIntervals

```
Estimated 95% Quantile = 2.988913

                      Standard            --- Pr > |t| --      95% Confidence
Contrast     Estimate    Error t Value       Raw Adjusted         Interval

trt1-trt3     -9.6667   6.4588   -1.50    0.1728   0.4155    -28.9713   9.6380
trt1-trt4      3.4000   6.4588    0.53    0.6129   0.9298    -15.9047  22.7047
trt3-trt4     13.0667   6.4588    2.02    0.0777   0.2046     -6.2380  32.3713
trt4-trt5      0.7333   6.4588    0.11    0.9124   0.9993    -18.5713  20.0380
```

All p-values are greater than the α and all confidence intervals include 0. Thus there is no significant difference.

%SimTests

```
Logically Constrained (Restricted Combinations) Step-Down Tests

                     Standard    ----- Pr > |t| -----
Contrast     Estimate    Error       Raw    Bon    Adj    SE(AdjP)

trt1-trt3     -9.6667   6.4588    0.1728 0.3457 0.3033     0.00113
trt1-trt4      3.4000   6.4588    0.6129 1.0000 0.8255     0.00168
trt3-trt4     13.0667   6.4588    0.0777 0.3107 0.2092     0.00132
trt4-trt5      0.7333   6.4588    0.9124 1.0000 0.9124           0
```

Again there are no statistically significant differences.

4.7 Hierarchical designs: split-plot design

This two-way design is used when it is impossible to randomize completely the levels of a factor in an experiment. The factor, called main factor, which can not be randomized completely is arranged in complete blocks. In each unit of the main factor in the blocks the second (sub) factor is represented with each of its levels once. The difference to a completely randomized two-way design is the estimate of the variance. While the sub factor uses the ordinary mean square error, the main factor uses a combination of the mean squares of the block and of the interaction between block and main factor (see below for details). In our model the block and the interaction between block and main factor are random. Note that the model does not contain the interaction between block and main factor. In the split-plot design this interaction term is at the same time the error term of the main factor. The model of the split-plot design is:

$$y_{ijh} = \mu + \alpha_i + \beta_j + b_h + (\alpha\beta)_{ij} + \phi_{ih} + \varepsilon_{ijh},$$

where

y_{ijh} = observation of level i of the factor A, of level j of factor B and level h of the
= block,
μ = mean of the population,
α_i = effect of level i of the main factor,
β_j = effect of level j of the sub factor,
b_h = effect of level h of the block,
$(\alpha\beta)_{ij}$ = interaction between level i of factor A and level j of factor B,
ϕ_{ih} = error term of the main factor, which is alternatively $(\alpha b)_{ih}$,
ε_{ijh} = error term of the sub factor.

In the example from Piepho (2000, p.218) the effect of four types of rice and six levels of nitrogen fertilizer is examined. The main factor, which is the fertilizer, is brought out in a randomized way in three blocks. Whereas the types of rice as the sub factor are brought out in a completely randomized way in each unit per block.

Data

	fertilizer 1				2			
block	type 1	2	3	4	1	2	3	4
1	4430	3944	3464	4126	5418	6502	4768	5192
2	4478	5314	2944	4482	5166	5858	6004	4604
3	3850	3660	3124	4836	6432	5586	5556	4652
	fertilizer 3				4			
block	type 1	2	3	4	1	2	3	4
1	6076	6008	6244	4546	6462	7139	5792	2774
2	6420	6127	5724	5744	7056	6982	5880	5036
3	6704	6642	6014	4146	6680	6564	6370	3638
	fertilizer 5				6			
block	type 1	2	3	4	1	2	3	4
1	7290	7682	7080	1414	8452	6228	5594	2248
2	7848	6594	6662	1960	8832	7387	7122	1380
3	7552	6576	6320	2766	8818	6006	5480	2014

4.7.1 Split plot analysis using PROC MIXED and PROC GLM

Originally PROC GLM was used to analyze a split-plot design until PROC MIXED became available. Though there are a number of similarities some important differences occur where PROC GLM calculates incorrect values. Comparing the two procedures, the main points are:

False values

The default standard error for the lsmeans from the main factor in an unbalanced design in PROC GLM is

$$stderr(\bar{x}_{i..}) = \sqrt{\frac{\sigma^2}{n_i}}, \forall\ i,$$

which is incorrect. According to Littell *et al.*(1996, p.61) The correct standard error, which is calculated by PROC MIXED, is:

$$stderr(\bar{x}_{i..}) = \sqrt{\frac{1}{n_i} \cdot (4\sigma^2_{block} + 4\sigma^2_{block \cdot residual} + \sigma^2)}.$$

Or expressed in terms of ANOVA means squares:

$$stderr(\bar{x}_{i..}) = \sqrt{\frac{1}{n_i}\left[\left(\frac{1}{2}\right)MS_{block} + \left(\frac{1}{2}\right)MS_{block \cdot main}\right]}.$$

According to our example for the wrong calculations for the main effect (fertilizer) this would be:

$$stderr(\bar{x}_{i..}) = \sqrt{\frac{349442}{12}} = 170.646.$$

To get the right standard error we have to use the estimates of σ^2, σ^2_{block} and $\sigma^2_{block \cdot residual}$ from e.g. the output of PROC MIXED:

```
 Covariance Parameter
       Estimates

Cov Parm       Estimate

block             16667
block*main       -51758
Residual         349442
```

This becomes

$$stderr(\bar{x}_{i..}) = \sqrt{\frac{1}{12} \cdot 4 \cdot 16667 + 4 \cdot (-51758) + 349442} = 131.997.$$

A further problem is the specification of the mean square error. The statement `contrast` in PROC GLM use the split-plot mean square error σ^2 as the default denominator for all F-statistics. This is not correct in every case. Although PROC GLM allows to define a single different mean square as denominator it is not able to calculate a linear combination of mean squares to be used with contrasts. Thus there is no way to get the correct results. The problem of PROC GLM not computing linear combinations of mean squares is shown in the follwing table:

Source of Variation	Standard Error
Main	$\sqrt{\frac{1}{n_i} \cdot 4\sigma^2_{block} + 4\sigma^2_{block \cdot residual} + \sigma^2}$
Sub	$\sqrt{\frac{1}{n_i} \cdot 4\sigma^2_{block} + \sigma^2_{block \cdot residual} + \sigma^2}$
Main · Sub	$\sqrt{\frac{1}{n_i} \cdot \sigma^2_{block} + \sigma^2_{block \cdot residual} + \sigma^2}$

All of the correct standard errors are functions of all three variance components and are estimated by linear combinations of mean squares.

Correct values

- The F-statistics and the probability values of the fixed effects, their interaction, the random effects block and the interaction between block and main factor are correct calculated by PROC GLM.
- In some cases PROC GLM estimates and tests main effect and sub effect contrasts and the resulting standard errors and denominator degrees of freedom correctly. All results are correct in the special case of a balanced design. However in an unbalanced design PROC GLM uses the correct defaults only in certain cases; in other cases, where the standard error involves a single mean square, PROC GLM allows to override the default by using the correct mean square to obtain the correct result. If the standard involves a linear combination of mean squares then PROC GLM has no available option to calculate in a correct way.

4.7.2 SAS-syntax for PROC MIXED

In the following calculations we use certain estimates from the output of PROC MIXED. For completion of the analysis we show the SAS-syntax of this procedure:

```
proc mixed method = reml nobound;
 class block main sub;
 model yield =  main sub main * sub /ddfm = satterth;
 random block block * main ;
 lsmeans main /pdiff adjust = dunnett cl;
 lsmeans sub  /pdiff adjust = dunnett cl;
run;
```

4.7.3 Analysis using *%SimIntervals* and *%SimTests*

As already shown it is not possible that PROC GLM does calculate the correct standard errors for the contrasts of the main effect. When using *%MakeGLMStats* for calculating the summary statistics the same error appears (because this macro uses the GLM procedure). Also it is not possible to calculate the pairwise contrasts for both factors at the same time with the *%Sim** because the macro *%Estimates* accepts only one mean square error. In a balanced design all the standard errors within one factor are equal. Therefore it is possible to split the analysis for each factor. On the other hand the analysis of an unbalanced design is difficult and time-consuming because the standard errors within one factor will be different.Thus the analysis via the *%Sim** macros is not recommendable for unbalanced designs. By using the output delivery system of SAS it may be possible to analyze both factors in one invocation simultaneously. However the use of the *%Sim** macros in combination with the ODS would go too far for this thesis. Westfall *et al.* (1999) shows the application of the ODS with the macros.

Main factor

As we use %Estimates we do not have a dummy variable in %Contrasts.
%Estimates needs a mean square error to calculate the correct standard error of the difference of the lsmeans. PROC MIXED does not compute this value. However it provides the required standard errors. Thus we use the following formula from Piepho (2000, p.224) to compute the needed standard errors from the mean square error (so that *%Estimates* can calculate from the mean square error the standard error):

$$stderr_{diff(main))} = \sqrt{\frac{2 \cdot \sigma^2_{main}}{n_{block} \cdot n_{sub}}} \Leftrightarrow \sigma^2_{main} = \frac{stderr^2_{diff(main)} \cdot n_{block} \cdot n_{sub}}{2}.$$

This becomes:

$$s^2_{main} = \frac{154,06^2 \cdot 12}{2} = 142406,9016.$$

Additional the *%Estimates* macro needs the degree of freedom. Again we use a formula from Piepho (2000, p.220). The degree of freedom of the error term of the main factor is

$df = (a-1)(r-1) = (6-1)(3-1) = 10$, with a levels of the main factor and r blocks.

Invocation

```
%macro Contrasts;
   C = { 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 -1};


   C = C‘ ;

   Clab = {"1-2", "1-3", "1-4", "1-5", "1-6"};
%mend;

%macro Estimates;
   EstPar = {4054.33 ,5478.17, 5866.25, 5864.42, 5812.00, 5796.75  };
   Mse    = 142406.9016;
   Cov    = Mse * I(6)/12 ;
   df     = 10;
%mend;
%SimIntervals(seed=100177);
%SimTests(seed=100177, type=LOGICAL);
```

Output

%SimIntervals

```
Estimated 95% Quantile = 2.991591

                    Standard            --- Pr > |t| --      95% Confidence
Contrast    Estimate    Error t Value       Raw Adjusted         Interval

1-2          -1423.8    154.1   -9.24    <.0001   <.0001     -1884.7   -963.0
1-3          -1811.9    154.1  -11.76    <.0001   <.0001     -2272.8  -1351.0
1-4          -1810.1    154.1  -11.75    <.0001   <.0001     -2271.0  -1349.2
1-5          -1757.7    154.1  -11.41    <.0001   <.0001     -2218.6  -1296.8
1-6          -1742.4    154.1  -11.31    <.0001   <.0001     -2203.3  -1281.5
```

%SimTests

```
Logically Constrained (Restricted Combinations) Step-Down Tests
```

Contrast	Estimate	Standard Error	Pr > |t| Raw	Bon	Adj	SE(AdjP)
1-2	-1423.8	154.1	<.0001	<.0001	<.0001	0
1-3	-1811.9	154.1	<.0001	<.0001	<.0001	0
1-4	-1810.1	154.1	<.0001	<.0001	<.0001	0
1-5	-1757.7	154.1	<.0001	<.0001	<.0001	0
1-6	-1742.4	154.1	<.0001	<.0001	<.0001	0

All fertilizers show significant differences.

Sub factor

Again there is no dummy variable in the *%Contrasts* macro. The required mean square error can be taken directly from the output of PROC MIXED. It is the error variance component, called *Residual*. According to Piepho (2000, p.220) the degree of freedom of the error term of the sub factor is $df = a(b-1)(r-1) = 6(4-1)(3-1) = 36$ with b levels of the sub factor.

Invocation

```
%macro Contrasts;
   C = { 1 -1  0  0,
         1  0 -1  0,
         1  0  0 -1};
   C = C‘ ;

   Clab = {"1-2", "1-3", "1-4"};
%mend;

%macro Estimates;
   EstPar = {6553.56 ,6155.50 , 5563.44 , 3642.11 };
   Mse    = 349422;
   Cov    = Mse * I(4)/18 ;
   df     = 36;
%mend;
%SimIntervals(seed=100177);
%SimTests(seed=100177, type=LOGICAL);
```

Output

%SimIntervals

```
Estimated 95% Quantile = 2.445971
```

Contrast	Estimate	Standard Error	t Value	--- Pr > |t| -- Raw	Adjusted	95% Confidence Interval	
1-2	398.1	197.0	2.02	0.0509	0.1221	-83.8936	880.0
1-3	990.1	197.0	5.02	<.0001	0.0001	508.2	1472.1
1-4	2911.5	197.0	14.78	<.0001	<.0001	2429.5	3393.4

%SimTests

```
Logically Constrained (Restricted Combinations) Step-Down Tests
```

Contrast	Estimate	Standard Error	----- Pr > |t| ----- Raw	Bon	Adj	SE(AdjP)
1-2	398.1	197.0	0.0509	0.0509	0.0509	0
1-3	990.1	197.0	<.0001	<.0001	<.0001	0
1-4	2911.5	197.0	<.0001	<.0001	<.0001	0

The types 3 and 4 differ significantly from type 1.

4.8 Analysis of covariance

If the disturbing factor can be quantified before the experiment or controlled during the experiment then blocks can be used to eliminate this factor. In the analysis of covariance the means are adjusted for the nuisance factor, which either cannot be controlled with blocks or cannot be measured before the experiment. We use the following model:

$$y_{ij} = \mu + \alpha_i + \beta x_{ij} + \gamma x_{ij} + \varepsilon_{ij},$$

where

y_{ij} = observation of the replication j on the level i of factor A and of the regression coefficients βx_{ij} and γx_{ij},

μ = mean of the population,

α_i = effect of level i of factor A,

βx_{ij}= first regression coefficient (initial weight),

γx_{ij}= second regression coefficient (initial age),

ε_{ij} = error term.

Snedecor and Cochran (1967, p.438-443) use an example where the effect of four types of feed on the weight gain of pigs was analyzed. The disturbing factors are initial weight and age.

Data

feed 1			feed 2			feed 3			feed 4		
age	weight	gain	age	weight	gain	age	weight	gain	age	weight	gain
78	61	1.40	78	74	1.61	78	80	1.67	77	62	1.40
90	59	1.79	99	75	1.31	83	61	1.41	71	55	1.47
94	76	1.72	80	64	1.12	79	62	1.73	78	62	1.37
71	50	1.47	75	48	1.35	70	47	1.23	70	43	1.15
99	61	1.26	94	62	1.29	85	59	1.49	95	57	1.22
80	54	1.28	91	42	1.24	83	42	1.22	96	51	1.48
83	57	1.34	75	52	1.29	71	47	1.39	71	41	1.31
75	45	1.55	63	43	1.43	66	42	1.39	63	40	1.27
62	41	1.57	62	50	1.29	67	40	1.56	62	45	1.22
67	40	1.26	67	40	1.26	67	40	1.36	67	39	1.36

Invocation

If we are interested in an all-pair comparison then the invocation is:

```
%MakeGLMStats(dataset   = covariate                  ,
              classvar  = treatment                  ,
              yvar      = gain                       ,
              model     = treatment in_age in_weight);

%macro Contrasts;
   C = {0    1 -1  0  0     0   0,
        0    1  0 -1  0     0   0,
        0    1  0  0 -1     0   0,
        0    0  1 -1  0     0   0,
        0    0  1  0 -1     0   0,
        0    0  0  1 -1     0   0};
   C = C' ;

   Clab = {"1-2", "1-3", "1-4",
                  "2-3", "2-4",
                         "3-4"};
%mend;

%SimTests(seed=100177, type=LOGICAL);
%SimIntervals(seed=100177);
```

Alternatively we can use only *%MakeGLMStats*:

```
%MakeGLMStats(dataset   = covariate                ,
              classvar  = treatment                ,
              yvar      = gain                     ,
              model     = treatment in_age in_weight,
              contrasts = all (treatment));

%SimIntervals(seed=100177);
%SimTests(seed=100177, type=LOGICAL);
```

Output

%SimIntervals

```
Estimated 95% Quantile = 2.683334
```

Contrast	Estimate	Standard Error	t Value	--- Pr > |t| -- Raw	Adjusted	95% Confidence Interval	
1-2	0.1546	0.0630	2.45	0.0194	0.0831	-0.0144	0.3237
1-3	0.0185	0.0637	0.29	0.7735	0.9914	-0.1524	0.1893
1-4	0.1196	0.0638	1.88	0.0694	0.2521	-0.0515	0.2907
2-3	-0.1362	0.0632	-2.15	0.0385	0.1501	-0.3059	0.0336
2-4	-0.0350	0.0638	-0.55	0.5863	0.9458	-0.2061	0.1361
3-4	0.1011	0.0631	1.60	0.1184	0.3844	-0.0683	0.2705

None of the comparisons show significant differences.

%SimTests

```
Logically Constrained (Restricted Combinations) Step-Down Tests
```

Contrast	Estimate	Standard Error	----- Pr > |t| ----- Raw	Bon	Adj	SE(AdjP)
1-2	0.1546	0.0630	0.0194	0.1162	0.0864	0.000844
1-3	0.0185	0.0637	0.7735	1.0000	0.8261	0.00172
1-4	0.1196	0.0638	0.0694	0.2081	0.1622	0.000932
2-3	-0.1362	0.0632	0.0385	0.1162	0.0949	0.000693
2-4	-0.0350	0.0638	0.5863	1.0000	0.8261	0.00172
3-4	0.1011	0.0631	0.1184	0.2081	0.1622	0.000932

Although the powerful *%SimTests* is used the decisions stay the same.

5 Examples

In this section four examples of the analysis of agricultural designs via the *%Sim** macros are shown. The first example contains a three-way analysis of variances, where the cell means model is used. Afterwards we present an analysis of covariances. As the third example we analyze a split plot design and the last example contains the analysis of tetrades. We assume that the requirements of the analysis of variances, e.g. variance homogeneity, normal distribution and uncorrelated data are given. Note that the number of replications in all four examples is very small, it is between three and twelve.

5.1 Licensure of carrots

After breeding a plant the new type has to be permitted so that it can be launched on the market. In Germany the licensure of new types of plants takes place at the Bundessortenamt. To get a licensure the new type has to pass a test method (Wertprüfung). In this test method the new type has to show its superiority (see below) above a standard. A Wertprüfung takes place on several locations in several years. In this time the new type has to be better than the standard in the totality of all characteristics from the mean of the different locations. The test method goes over two years (the period of time is dependent from the plant species). Our data set is a small part from a Wertprüfung with carrots (*Daucus carota L. ssp. sativus Umbellifarae*). Here the superiority contains the yield (marketable ware: very good and not marketable: deformed, cracked, too small), intensity of the color inside the carrot, intensity of the green color inside the carrot, the tensile strength, number of shoots and the density of the circles on the outside of the carrot. In total these are nine characteristics. The observations of the four new types ("Jeanette", "Napoli", "Splendid" and "Yukon") and the standard type ("Bolero") were measured at five locations (shortcuts: lemgr, quedl, unihh, unibe, olvst) in the years 1996 and 1997. Every factor combination has four replications. To simplify the analysis we deviate from the original guidelines of the Bundessortenamt. For us a new type gets a licensure if it shows superiority above the standard in the totality of all characteristics. However

this superiority has to appear in both years on at least one location. To represent all characteristics in one endpoint we use the parametric version of the multivariate endpoint analysis of O'Brien (1984). Therefore we standardize each endpoint by subtracting the overall variable mean from each observation and afterwards the differences are divided by the standard deviation of the endpoint. Finally we calculate the response variable by adding each of the observations of the endpoints. However the algebraic sign from the endpoints with negative specification (such as the green color inside the carrot or the not marketable ware: deformed) are switched. Thus we receive an univariate design. Note that we are not interested in the difference between the years or the locations. We have to analyze each combination of the year, type and location. So we get the following model:

$$y_{ijmk} = \mu + \alpha_i + \beta_j + \gamma_m + (\alpha\beta)_{ij} + (\alpha\gamma)_{im} + (\beta\gamma)_{jm} + (\alpha\beta\gamma)_{ijm} + \varepsilon_{ijkm},$$

where

y_{ijmk} = observation of replication k on level i of factor A (type), level j of factor B (location) and level m of factor C (year),

μ = mean of the population,

α_i = effect of level i of factor A,

β_j = effect of level j of factor B,

γ_m = effect of level m of factor C,

$(\alpha\beta)_{ij}$ = interaction between level i of factor A and level j of factor B,

$(\alpha\gamma)_{im}$ = interaction between level i of factor A and level m of factor C,

$(\beta\gamma)_{jm}$ = interaction between level j of factor B and level m of factor C,

$(\alpha\beta\gamma)_{ijm}$ = interaction between level i of factor A, level j of factor B and level m of factor C,

ε_{ijmk} = error term.

Data

This is a small part of the data set:

location	year	type	replication	shoots	tensile	...	small	endpoint
lemgr	96	bolero	1	-0.653	1.454	...	-0.470	2.144
lemgr	96	bolero	2	-0.653	1.196	...	-0.470	2.631
lemgr	96	bolero	3	-0.653	1.196	...	-0.470	2.520
lemgr	96	bolero	4	-0.653	1.454	...	-0.322	1.800
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮
olvst	97	napoli	1	2.897	-1.387	...	-0.421	-3.416
olvst	97	napoli	2	0.767	-1.645	...	-0.480	1.499
olvst	97	napoli	3	0.057	-1.387	...	-0.539	1.599
olvst	97	napoli	4	0.767	-1.387	...	-0.539	1.002

Invocation

We are only interested in the decision whether a new type is better than the standard in both years on at least two locations. Thus we use the *%SimTest* macro. Note that there are 40 comparisons, which are too many to use the logical constraint method of Shaffer. So we use the unconstrained method of Holm, which is not so time consuming. To test the superiority we have to use one-sided tests. As already mentioned we are only interested in the comparisons of types in a specific year and at a specific location. Thus we do not compare the years or locations. For this specific invocation we use the cell means model. This is a brief general overview how the contrasts have to be specified:

		bolero									
contrast	intercept	lemgr		olvst		quedl		unibe		unihh	
96	97	96	97	96	97	96	97	96	97		
"96 lemgr jeanette"	0	-1	0	0	0	0	0	0	0	0	0
"97 lemgr jeanette"	0	0	-1	0	0	0	0	0	0	0	0

jeanette											yukon	
lemgr		olvst		quedl		unibe		unihh		...	unihh	
96	97	96	97	96	97	96	97	96	97		96	97
1	0	0	0	0	0	0	0	0	0	...	0	0
0	1	0	0	0	0	0	0	0	0	...	0	0

Therefore the invocation is:

```
%MakeGLMStats(dataset   = licensure              ,
              classvar  = types location year    ,
              yvar      = endpoint               ,
              model     = type * location * year );
%macro Contrasts;
   C =  {0 -1  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
         0  0 -1  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
         0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
         0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
         0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
         0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
         0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
         0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0,
         0  0  0 -1  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
         0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0,
         0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
         0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
         0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
         0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
         0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0,
         0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0,
         0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
         0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
         0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
         0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
         0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
         0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

```
           0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
        0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0,
        0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0,
        0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0,
        0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0,
        0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
        0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
        0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0,
        0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0,
        0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0,
        0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0,
        0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
        0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
        0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
        0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
        0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0,
        0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0,
        0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0,
        0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0};
C=C';
   clab = {"96 lemgr jeanette", "97 lemgr jeanette", "96 lemgr napoli",
           "97 lemgr napoli",   "96 lemgr splendid", "97 lemgr splendid",
           "96 lemgr yukon",    "97 lemgr yukon",    "96 olvst jeanette",
           "97 olvst jeanette", "96 olvst napoli",   "97 olvst napoli",
           "96 olvst splendid", "97 olvst splendid", "96 olvst yukon",
           "97 olvst yukon",    "96 quedl jeanette", "97 quedl jeanette",
           "96 quedl napoli",   "97 quedl napoli",   "96 quedl splendid",
           "97 quedl splendid", "96 quedl yukon",    "97 quedl yukon",
           "96 unibe jeanette", "97 unibe jeanette", "96 unibe napoli",
           "97 unibe napoli",   "96 unibe splendid", "97 unibe splendid",
           "96 unibe yukon",    "97 unibe yukon",    "96 unihh jeanette",
           "97 unihh jeanette", "96 unihh napoli",   "97 unihh napoli",
           "96 unihh splendid", "97 unihh splendid", "96 unihh yukon",
```

```
          "97 unihh yukon"};
%mend;

%SimTests(seed=100177, side=U, type=FREE);
```

Output

Unconstrained (Free Combinations) Step-Down Tests

Contrast	Estimate	Standard Error	Pr > t Raw	Bon	Adj	SE(AdjP)
96 lemgr jeanette	-1.3231	1.0390	0.8976	1.0000	1.0000	0
97 lemgr jeanette	1.5013	1.0390	0.0753	1.0000	0.7616	0.00240
96 lemgr napoli	0.4542	1.0390	0.3313	1.0000	0.9854	0.000822
97 lemgr napoli	2.6845	1.0390	0.0054	0.1556	0.1251	0.00108
96 lemgr splendid	-4.0990	1.0390	0.9999	1.0000	1.0000	0
97 lemgr splendid	3.8680	1.0390	0.0001	0.0046	0.0045	0.000068
96 lemgr yukon	-7.3462	1.0390	1.0000	1.0000	1.0000	0
97 lemgr yukon	4.0666	1.0390	<.0001	0.0024	0.0023	0.000065
96 olvst jeanette	1.9346	1.0390	0.0323	0.8392	0.4845	0.00232
97 olvst jeanette	3.6958	1.0390	0.0003	0.0078	0.0074	0.000129
96 olvst napoli	1.2573	1.0390	0.1141	1.0000	0.8601	0.00210
97 olvst napoli	0.6369	1.0390	0.2704	1.0000	0.9802	0.000946
96 olvst splendid	-0.4119	1.0390	0.6538	1.0000	0.9999	0.000050
97 olvst splendid	-0.5696	1.0390	0.7078	1.0000	0.9999	0.000050
96 olvst yukon	1.0269	1.0390	0.1623	1.0000	0.9300	0.00164
97 olvst yukon	1.9838	1.0390	0.0291	0.7847	0.4612	0.00227
96 quedl jeanette	0.7356	1.0390	0.2400	1.0000	0.9749	0.00106
97 quedl jeanette	-3.2620	1.0390	0.9990	1.0000	1.0000	0
96 quedl napoli	4.0174	1.0390	<.0001	0.0028	0.0027	0.000078
97 quedl napoli	-1.5686	1.0390	0.9334	1.0000	1.0000	0
96 quedl splendid	1.4502	1.0390	0.0824	1.0000	0.7725	0.00238
97 quedl splendid	-0.8157	1.0390	0.7832	1.0000	1.0000	0
96 quedl yukon	0.8123	1.0390	0.2178	1.0000	0.9681	0.00118
97 quedl yukon	-0.5114	1.0390	0.6883	1.0000	0.9999	0.000050
96 unibe jeanette	6.7784	1.0390	<.0001	<.0001	<.0001	0
97 unibe jeanette	-0.4570	1.0390	0.6697	1.0000	0.9999	0.000050
96 unibe napoli	0.5870	1.0390	0.2865	1.0000	0.9802	0.000946
97 unibe napoli	-1.3666	1.0390	0.9048	1.0000	1.0000	0
96 unibe splendid	3.1000	1.0390	0.0017	0.0499	0.0433	0.000518
97 unibe splendid	0.1815	1.0390	0.4308	1.0000	0.9966	0.000404
96 unibe yukon	5.2782	1.0390	<.0001	<.0001	<.0001	0
97 unibe yukon	0.5184	1.0390	0.3093	1.0000	0.9817	0.000912
96 unihh jeanette	-0.8553	1.0390	0.7942	1.0000	1.0000	0
97 unihh jeanette	2.0106	1.0390	0.0274	0.7680	0.4489	0.00228
96 unihh napoli	5.9212	1.0390	<.0001	<.0001	<.0001	0
97 unihh napoli	-3.9470	1.0390	0.9999	1.0000	1.0000	0
96 unihh splendid	6.8160	1.0390	<.0001	<.0001	<.0001	0

```
97 unihh splendid     -4.1369   1.0390    0.9999 1.0000 1.0000           0
96 unihh yukon         3.7571   1.0390    0.0002 0.0065 0.0063    0.000104
97 unihh yukon         5.7490   1.0390    <.0001 <.0001 <.0001           0
```

The only type, which is significant in both years on at least one location is the type "Yukon" on the location "unihh". The first adjusted p-value is 0.0063 and the second is smaller than 0.0001. Thus only the type "Yukon" gets the licensure.

5.2 Uptake of a fungicide into apple leaves

The aim of this experiment was to find an effective fungicide against a fungi, which harms apples. In the study 25 fungicides with five to ten replications were tested on their ability to penetrate the cuticle from leaves of *Gloster*. The endpoint is the uptake [%] in the leaves, which was measured after several hours. However the time [hours] between the application of the fungicide and the measuring of the penetration is not constant. Thus we consider this time as a covariate. Because we cannot assume which fungicide is better than the other, we are interested whether there are differences between the fungicides. Thus we have two-sided tests. Also we want to know how large the differences between the fungicides are so we use *%SimIntervals* to get simultaneous confidence intervals. We use the following model:

$$y_{ij} = \mu + \alpha_i + \beta x_{ij} + \varepsilon_{ij},$$

where

y_{ij} = observation of the replication j on the level i of the factor (fungicide) and of the regression coefficient βx_{ij} (hours),

μ = mean of the population,

α_i = effect of level i of the factor (fungicide),

βx_{ij}= regression coefficient (hours),

ε_{ij} = error term.

Data

A part of the data set is provided in the following tabular:

fungicide	replication	hours	penetration
38-1	1	18.91	16.60
38-1	2	19.01	11.94
38-1	3	19.10	7.79
38-1	4	19.18	10.05
⋮	⋮	⋮	⋮
66-6	7	27.63	3.49
66-6	8	27.73	1.14
66-6	9	27.84	3.32

Invocation

```
%MakeGLMStats(dataset   = appleleaf      ,
              classvar  = fungicide      ,
              yvar      = penetration    ,
              model     = fungicide hours,
              contrasts = all (fungicide));

%SimIntervals(seed=100177);
```

Output

In total we have 300 comparisons. So we abandon the whole output. This is a part of it:

```
Estimated 95% Quantile = 3.601726

                    Standard                --- Pr > |t| --      95% Confidence
Contrast    Estimate    Error t Value       Raw Adjusted         Interval

1-2           2.0235   3.1525    0.64    0.5217   1.0000     -9.3311  13.3781
1-3           2.6892   4.5534    0.59    0.5555   1.0000    -13.7108  19.0892
1-4           9.7953   6.6033    1.48    0.1396   0.9974    -13.9880  33.5786
1-5           0.4186   7.2461    0.06    0.9540   1.0000    -25.6800  26.5173
 :               :        :       :       :        :           :       :
23-24        -1.3567   3.1150   -0.44    0.6637   1.0000    -12.5760   9.8627
23-25         0.4057   3.4646    0.12    0.9069   1.0000    -12.0730  12.8843
24-25         1.7623   3.1767    0.55    0.5797   1.0000     -9.6792  13.2038
```

To shorten the output only the significant comparisons are summarized in the following tabular:

contrast	95% confidence interval		contrast	95% confidence interval		contrast	95% confidence interval	
38-1 − 64-1	0.8694	23.0270	38-6 − 66-4	5.3767	31.1301	39-4 − 39-5	2.6900	26.3151
38-1 − 64-2	4.0565	26.2687	38-6 − 66-5	5.2300	28.5635	39-4 − 39-6	1.6554	26.7618
38-1 − 64-5	2.1408	30.9946	38-6 − 66-6	7.6203	29.6979	39-4 − 64-1	2.9136	37.8492
38-1 − 66-1	2.2156	30.0016				39-4 − 64-2	8.0831	39.1086
38-1 − 66-2	0.5313	33.1158	39-1 − 64-1	1.7128	27.1373	39-4 − 64-5	13.5763	36.4256
38-1 − 66-4	0.7532	44.6443	39-1 − 64-2	5.8749	29.4041	39-4 − 64-7	13.2022	39.4043
			39-1 − 64-5	6.7192	31.3700	39-4 − 66-1	13.2396	35.8440
38-2 − 64-2	2.0456	24.2326	39-1 − 64-7	4.1658	36.5280	39-4 − 66-2	14.5218	35.9918
38-2 − 64-5	1.7271	27.3614	39-1 − 66-1	6.6982	30.4728	39-4 − 66-3	11.8403	34.3238
38-2 − 66-1	1.7596	26.4106	39-1 − 66-2	5.7847	32.8161	39-4 − 66-4	18.4332	43.8307
38-2 − 66-2	0.4699	29.1302	39-1 − 66-3	1.3864	32.8650	39-4 − 66-5	14.8678	44.6828
38-2 − 66-4	1.0925	40.2580	39-1 − 66-4	6.8284	43.5229	39-4 − 66-6	14.3869	48.6884
			39-1 − 66-5	2.6356	45.0024			
38-3 − 39-3	-23.4151	-0.3051	39-1 − 66-6	1.7223	49.4403	39-5 − 66-4	5.0020	28.2568
38-3 − 64-5	2.2505	25.5066				39-5 − 66-5	2.0490	28.4966
38-3 − 64-7	1.7310	28.6307	39-2 − 64-2	3.9031	28.7204	39-5 − 66-6	1.9840	32.0862
38-3 − 66-1	1.9343	24.9046	39-2 − 64-5	6.4916	28.9421			
38-3 − 66-2	3.0444	25.2244	39-2 − 64-7	4.4303	33.6080	39-6 − 64-4	5.7499	28.0968
38-3 − 66-3	0.2248	23.6945	39-2 − 66-1	6.3939	28.1216	39-6 − 64-5	3.2841	27.8493
38-3 − 66-4	6.7364	33.2828	39-2 − 66-2	6.0851	29.8603	39-6 − 64-6	3.5612	31.0968
38-3 − 66-5	3.1432	34.1627	39-2 − 66-3	2.0765	29.5194			
38-3 − 66-6	2.6475	38.1830	39-2 − 66-4	7.7529	39.9429	64-2 − 64-3	-22.8928	-0.2450
			39-2 − 66-5	3.6761	41.3064	64-2 − 64-4	-25.0837	-0.6734
38-4 − 39-3	-34.8161	-3.1162	39-2 − 66-6	2.8476	45.6596	64-2 − 64-6	-33.6992	-2.9443
38-4 − 39-4	-32.2779	-4.1790						
38-4 − 66-4	1.9953	23.8117	39-3 − 39-5	2.4275	28.0529	64-3 − 64-5	0.7719	25.1762
38-4 − 66-5	0.4542	22.6395	39-3 − 39-6	1.0391	28.8535	64-3 − 66-1	0.7747	24.2551
38-4 − 66-6	1.4403	25.1780	39-3 − 64-1	5.6358	36.6024	64-3 − 64-4	0.5982	37.6119
			39-3 − 64-2	10.5732	38.0938			
38-5 − 64-7	1.0537	33.8491	39-3 − 64-3	0.4021	25.1271	64-4 − 64-5	2.9614	25.6060
38-5 − 66-2	0.7426	32.0672	39-3 − 64-4	0.0431	22.8667	64-4 − 64-7	0.7864	30.3856
38-5 − 66-3	0.8629	27.5974	39-3 − 64-5	14.7193	36.7580	64-4 − 66-1	2.8813	24.7679
38-5 − 66-4	10.6327	33.9275	39-3 − 64-7	13.4018	40.6800	64-4 − 66-2	2.4488	26.6302
38-5 − 66-5	9.8385	32.0084	39-3 − 66-1	14.5150	36.0486	64-4 − 66-4	3.9768	36.8526
38-5 − 66-6	11.5193	33.8523	39-3 − 66-2	14.9554	37.0335			
			39-3 − 66-3	11.5152	36.1242	64-5 − 64-6	-31.3481	-8.1056
			39-3 − 66-4	17.5616	46.1777			
			39-3 − 66-5	13.6859	47.3401	64-6 − 64-7	7.5749	34.4834
			39-3 − 66-6	12.9970	51.5537	64-6 − 66-1	7.7908	30.7447
						64-6 − 66-2	8.8915	31.0739
						64-6 − 66-3	6.0627	29.5532
						64-6 − 66-4	12.5674	39.1483
						64-6 − 66-5	8.9701	40.0323
						64-6 − 66-6	8.4716	44.0555

From the 300 comparisons 104 show significant differences. By using the simultaneous confidence intervals we get more information about the fungicides than by using the p-value. For example a cuticle assimilates more of 38-1 than of 64-1. The uptake is at least 0.87% and at most 23.03% higher.

5.3 Effectiveness of fertilizers and substrates on spinach

In this greenhouse experiment the effect of five fertilizers and two substrates on the dry weight [g] of spinach is studied. Each spinach plant was planted into a flowerpot with one of the substrates, later one of the fertilizers was applied. The pots were placed one behind the other on a rotating conveyor belt. To eliminate a possible nuisance factor blocks containing five pots were created and in each block each level of the factor fertilizer appeared once. To simplify the lay out of the experiment the experimenter used a split plot design. The main factor contains the substrates and therefore the fertilizers are the sub factor. We are interested in the differences between the fertilizers and the substrates and how large these differences are. Thus we use the *%SimIntervals* macro. However we are not able to predict how large the effects of each of the fertilizers and substrates are, so we use two-sided tests. The model of our design is:

$$y_{ijh} = \mu + \alpha_i + \beta_j + b_h + (\alpha\beta)_{ij} + \phi_{ih} + \varepsilon_{ijh},$$

where

y_{ijh} = observation of level i of the main factor (substrate), of level j of the sub
= factor (fertilizer) and level h of the block,
μ = mean of the population,
α_i = effect of level i of the main factor,
β_j = effect of level j of the sub factor,
b_h = effect of level h of the block,
$(\alpha\beta)_{ij}$ = interaction between level i of the main factor and level j of the sub factor,
ϕ_{ih} = error term of the main factor, which is alternatively $(\alpha b)_{ih}$,
ε_{ijh} = error term of the sub factor.

Data

The data set is provided in the following table:

substrate	block	fertilizer				
		1	2	3	4	5
a	1	10.6	10.5	10.2	10.6	10.2
	2	10.4	10.4	10.3	11.1	10.6
	3	10.7	10.8	11.0	10.8	10.9
	4	10.5	10.7	10.6	11.1	10.8
	5	10.3	10.7	11.0	9.4	10.7
	6	10.3	10.4	10.8	10.6	11.4
b	1	10.6	10.0	9.1	10.3	9.9
	2	10.1	9.3	10.3	9.6	10.4
	3	8.7	9.3	9.1	10.4	10.3
	4	9.7	10.1	11.1	10.3	9.8
	5	11.5	9.6	9.8	10.0	10.0
	6	8.2	11.8	10.0	6.2	10.0

Main factor

As shown in 4.7 the analysis of the split plot design via the *%Sim** macros is divided into two parts, for both factors the macros have to be invoked separately. Two control the FWE over both invocations we use an $\alpha = 2.5\%$ for the analysis of each factor. We begin with main factor. For the invocation of the *%SimIntervals* macro we have to use the macros *%Contrasts* and *%Estimates*.

Invocation

%Estimates requires several values for the mean square error, the lsmeans, the degree of freedom and the covariance matrix. The lsmeans can directly be taken e.g. from the PROC MIXED output. To get the mean square error we use the standard error from the PROC MIXED output and compute the needed value:

$$\sigma^2_{main} = \frac{stderr^2_{diff(main)} \cdot n_{block} \cdot n_{sub}}{2}$$

This becomes:

$$s^2_{main} = \frac{0.2246^2 \cdot 30}{2} = 0.7567$$

The degree of freedom is $df = (a-1)(r-1) = 5$ and the covariance matrix is $Cov = Mse * I(2)/30$ with two levels of the main factor (substrate) and 30 observations per level. Note that we do not use *%MakeGLMStats*, thus the contrasts in the *%Contrasts* macro do not have 0 as the first value for the intercept parameter.

```
%macro Contrasts;
   C = {1 -1};

   C = C' ;

   Clab = {"1-2"};
%mend;

%macro Estimates;

   EstPar = {10.6467, 9.85};
   Mse    = 0.7566774;
   Cov    = Mse * I(2)/30;
   df     = 5;
%mend;
%SimIntervals(seed=100177, conf=0.975);
```

Output

```
Estimated 97.5% Quantile = 3.092336

                      Standard             --- Pr > |t| --     97.5% Confidence
Contrast    Estimate    Error t Value       Raw Adjusted         Interval

1-2           0.7967   0.2246    3.55    0.0164   0.0148      0.1022   1.4912
```

The adjusted p-value is smaller than 2.5%, thus the difference between the two substrates is significant. If spinach is planted in substrate 1 it brings at least 0.102[g] and at most 1.491[g] more yield than if it is planted in substrate 2.

Sub factor

Now the difference between the five fertilizers are analyzed. Again we use $\alpha = 2.5\%$ to control the FWE.

Invocation

For the *%Estimates* macro the lsmeans and the mean square error can be directly taken from e.g. the PROC MIXED output. The degree of freedom is calculated by $df = a(b-1)(r-1) = 40$ and the covariance matrix is $Cov = Mse * I(5)/12$ with five levels of the factor fertilizer and twelve observations per level.

```
%macro Contrasts;
   C = { 1 -1  0  0  0,
         1  0 -1  0  0,
         1  0  0 -1  0,
         1  0  0  0 -1,
         0  1 -1  0  0,
         0  1  0 -1  0,
         0  1  0  0 -1,
         0  0  1 -1  0,
         0  0  1  0 -1,
         0  0  0  1 -1};
   C = C' ;

   Clab = {"1-2", "1-3", "1-4", "1-5",
           "2-3", "2-4", "2-5",
           "3-4", "3-5",
           "4-5"};
%mend;

%macro Estimates;
   EstPar = {10.2167, 10.3, 10.275, 10.0333, 10.4167};
   Mse    = 0.6915;
   Cov    = Mse * I(5)/12 ;
   df     = 40;
%mend;
%SimIntervals(seed=100177, conf=0.975);
```

Output

```
Estimated 97.5% Quantile = 3.133614
```

Contrast	Estimate	Standard Error	t Value	--- Pr > |t| -- Raw	Adjusted	97.5% Confidence Interval	
1-2	-0.0833	0.3395	-0.25	0.8074	0.9992	-1.1471	0.9805
1-3	-0.0583	0.3395	-0.17	0.8645	0.9999	-1.1221	1.0055
1-4	0.1834	0.3395	0.54	0.5920	0.9807	-0.8804	1.2472
1-5	-0.2000	0.3395	-0.59	0.5591	0.9745	-1.2638	0.8638
2-3	0.0250	0.3395	0.07	0.9417	1.0000	-1.0388	1.0888
2-4	0.2667	0.3395	0.79	0.4367	0.9320	-0.7971	1.3305
2-5	-0.1167	0.3395	-0.34	0.7328	0.9974	-1.1805	0.9471
3-4	0.2417	0.3395	0.71	0.4806	0.9516	-0.8221	1.3055
3-5	-0.1417	0.3395	-0.42	0.6786	0.9930	-1.2055	0.9221
4-5	-0.3834	0.3395	-1.13	0.2655	0.7896	-1.4472	0.6804

None of the comparisons show significant differences because all p-values are larger than 2.5% and all simultaneous confidence intervals contain 0.

5.4 Effects of water and rotation of crops on lettuce

This experiment shows the effect of the application of water and a rotation of crops on the fresh weight [g] of lettuce. The study took place in a greenhouse and the lettuce was planted into flowerpots. The first factor, the water application, has two levels: normal and reduced doses. A rotation of crops is the second factor. It has the levels 1, 2 and 3, so that 1 means the lettuce was planted into unused substrate, position 2 is already used substrate (once) and when the substrate was used twice it is position 3. For each combination of the factors twelve plants were used. We are interested in possible interactions between the levels of the two factors.

Data

This is the data set:

water application	position 1	2	3
1	168.9	154.6	63.1
	180.5	150.9	75.9
	177.3	158.5	77.2
	129.5	174.8	83.7
	183.4	149.4	98.3
	187.7		90.2
	177.7	167.5	97.2
	177.9	166.2	87.2
	182.1	178.9	78.5
	177.0	167.3	87.7
	169.0	158.2	79.1
	181.7	140.4	87.9
2	98.8	101.4	94.4
	91.4	109.3	89.9
	90.0	103.0	86.4
	94.6	90.8	85.8
	97.7		93.7
	91.7	105.0	91.7
	105.3	105.6	83.9
	102.2		79.7
	103.0	105.2	90.7
	96.6	104.8	72.6
	98.9	106.5	72.9
	76.3	.	89.2

Invocation

```
%MakeGLMStats(dataset  = tetrade       ,
              classvar = water position,
              yvar     = freshweight   ,
              model    = water*position);

%macro Contrasts;
   C = {0    1 -1  0 -1  1  0,
        0    1  0 -1 -1  0  1,
        0    0  1 -1  0 -1  1};
   C = C' ;

   Clab = {"(11-12)-(21-22)",
           "(11-13)-(21-23)",
           "(12-13)-(22-23)"};
%mend;

%SimTests(seed=100177, type=LOGICAL);
```

Output

```
Logically Constrained (Restricted Combinations) Step-Down Tests

                            Standard    ----- Pr > |t| -----
Contrast           Estimate    Error       Raw    Bon    Adj    SE(AdjP)

(11-12)-(21-22)     22.5020   6.1739    0.0005 0.0005 0.0005           0
(11-13)-(21-23)     81.6750   5.8705    <.0001 <.0001 <.0001           0
(12-13)-(22-23)     59.1730   6.1739    <.0001 <.0001 <.0001           0
```

As we can see from the output all three tetrades show a significant interaction. For example the second one, (11-12)-(21-22), means that the fresh weight of the lettuce for the first position of the rotation of crops with normal water application is 81.68[g] higher than with reduced water application.

6 Summary

The macros *%SimTests* and *%SimIntervals* are powerful and very practical tools to analyze agricultural designs. They can be used with comparatively little effort and are a valuable addition to the analysis of many designs. All advantages of both macros, e.g. the simultaneous confidence intervals of *%SimIntervals* or the simultaneous control over all contrast test over all factors, fill some of the gaps in the current statistical analysis software.

The aim of this thesis is to show the applications of the two macros for agricultural designs. However here only a small part of the possibilities of the *%Sim** macros is shown. They can also be used for repeated measurements, multiple endpoints and much more, see Westfall *et al.* (1999) for details. Further the use of the macros is improved when the invocation is combined with the output delivery system of SAS. With Bretz's (1999) calculations the exact critical values of the multivariate t distribution have become available now. Soon two altered versions of the *%Sim** macros will be completed by Bretz where the exact critical values instead of the simulated values are used.

Furthermore in this thesis only pairwise tests are treated. Westfall *et al.* (1999) show the invocation for combinations of means. Thus e.g. test on trend, as dose-response tests, are available.

Possible future improvements of the macros could be e.g. tests on equivalence and/or ratio tests.

7 Listings

In this section the original source code of the macros *%SimIntervals*, *%SimTests* and *%MakeGLMStats* from Westfall *et al.* (1999) is provided. All three macros are given on the disk.

7.1 The *%SimIntervals* Macro

```
/*-----------------------------------------------------------------*/
/* Name:      SimIntervals                                         */
/* Title:     Simultaneous Confidence Intervals for General        */
/*            Linear Functions                                     */
/* Author:    Randy Tobias, sasrdt@sas.com,                        */
/* Reference: Edwards and Berry (1987). The efficiency of          */
/*            simulation-based multiple comparisons.               */
/*            Biometrics 43, 913-928.                              */
/* Release:   Version 7.01                                         */
/*-----------------------------------------------------------------*/
/* Inputs:                                                         */
/*                                                                 */
/*      NSAMP =  simulation size, with 20000 as default            */
/*                                                                 */
/*      SEED  =  random number seed, with 0 (clock time)           */
/*               as default                                        */
/*                                                                 */
/*      CONF  =  desired confidence level, with 0.95 as default    */
/*                                                                 */
/*      SIDE  = U, L or B, for upper-tailed, lower-tailed          */
/*              or two-tailed, respectively. SIDE=B is default.    */
/*                                                                 */
/*  Additionally, %SimIntervals requires two further macros to     */
/*  be defined that use SAS/IML to construct the estimates and     */
/*  the contrasts of interest.  In particular, make sure the       */
/*  following two macros are defined before invoking               */
/*  %SimIntervals:                                                 */
/*                                                                 */
/*     %Estimate: Uses SAS/IML code to define                      */
/*        EstPar - (column) vector of estimated parameters         */
/*        Cov    - covariance matrix for the for the estimates     */
/*        df     - error degrees of freedom; set to 0 for          */
/*                 asymptotic analysis                             */
/*                                                                 */
/*     %Contrasts: Uses SAS/IML code to define                     */
/*        C      - matrix whose columns define the contrasts of    */
/*                 interest between the parameters                 */
/*        CLab   - (column) character vector whose elements        */
/*                 label the respective contrasts in C             */
/*                                                                 */
```

```
/*  You can either define these macros directly, or use the     */
/*  %MakeGLMStats macro to define them.                         */
/*                                                              */
/*--------------------------------------------------------------*/
/* Output:                                                      */
/*   The output is a dataset with one observation for each      */
/*   contrast and the following variables:                      */
/*                                                              */
/*     Contrast - contrast label                                */
/*     Estimate - contrast estimated value                      */
/*     StdErr   - standard error of estimate                    */
/*     tValue   - normalized estimate, Estimate/StdErr          */
/*     RawP     - non-multiplicity-adjusted p-value             */
/*     OneP     - one-step multiplicity-adjusted p-value        */
/*     LowerCL  - multiplicity-adjusted lower confidence limit  */
/*     UpperCL  - multiplicity-adjusted upper confidence limit  */
/*                                                              */
/*   This dataset is also displayed as a formatted table, using */
/*   the ODS system.                                            */
/*--------------------------------------------------------------*/

%macro SimIntervals(nsamp   = 20000,
                    seed    =     0,
                    conf    =  0.95,
                    side    =     B,
                    options =      );
%global ANORM quant;

options nonotes;

proc iml;
   %Estimates;
   if (df <= 0) then call symput('ANORM','1');
   else              call symput('ANORM','0');
   %Contrasts;

   Cov = C`*Cov*C;
   D   = diag(1/sqrt(vecdiag(Cov)));
   R   = D*Cov*D;

   evec = eigvec(R);
   eval = eigval(R) <> 0;
   U = (diag(sqrt(eval))*evec`)`;
   dimU = sum(eval > 1e-8);

   U    = U[,1:dimU];

   ests = C`*EstPar;
   ses  = sqrt(vecdiag(Cov));
   tvals = ests/ses;
   %if       (&side = B) %then %do;
```

```
   if df>0 then rawp = 2*(1-probt(abs(tvals),df));
      else rawp = 2*(1-probnorm(abs(tvals)));
   %end;
%else %if (&side = L) %then %do;
   if df>0 then rawp =      probt(    tvals ,df) ;
   else rawp = probnorm( tvals);
   %end;
%else                        %do;
   if df>0 then rawp =    1-probt(    tvals ,df) ;
   else rawp = 1-probnorm(tvals);
   %end;

adjp = j(ncol(C),1,0);
maxt=j(&nsamp,1,0);
do isim = 1 to &nsamp;
   Z = U*rannor(j(dimU,1,&seed));
   if df>0 then do;
      V = cinv(ranuni(&seed),df);
      tvalstar = Z / sqrt(V/df);
      end;
   else do; tvalstar = Z; end;
   %if (&side = B) %then %do; mx = max(abs(tvalstar)); %end;
   %else                 %do; mx = max(    tvalstar ); %end;
   maxt[isim] = mx;

   %if       (&side = B) %then %do; adjp = adjp + (mx>abs(tvals)); %end;
   %else %if (&side = L) %then %do; adjp = adjp + (mx>   -tvals ); %end;
   %else                       %do; adjp = adjp + (mx>    tvals ); %end;
   end;
adjp = adjp/&nsamp;

confindx = round(&nsamp*&conf,1);
sorttemp = maxt;
maxt[rank(maxt),] = sorttemp;
c_alpha = maxt[confindx];

start tlc(n,d); return(trim(left(char(n,d)))); finish;

%if (&side = B) %then %do;
   LowerCL = ests - c_alpha*ses;
   UpperCL = ests + c_alpha*ses;
   %end;
%else %if (&side = L) %then %do;
   LowerCL = j(ncol(C),1,.M);
   UpperCL = ests + c_alpha*ses;
   %end;
%else %do;
   LowerCL = ests - c_alpha*ses;
   UpperCL = j(ncol(C),1,.I);
   %end;
```

```
   create SimIntOut
      var {"Estimate" "StdErr" "tValue" "RawP"
           "OneP" "LowerCL" "UpperCL"};
   data = ests || ses || tvals || rawp || adjp || LowerCL || UpperCL;
   append from data;
   call symput('confpct',tlc(100*&conf,4));
   call symput('quant'  ,tlc(c_alpha  ,8));

   create labels from clab; append from clab;

data SimIntOut; merge labels(rename=(COL1=Contrast)) SimIntOut; run;

%if (^%index(%upcase(&options),NOPRINT)) %then %do;

proc template;
delete MCBook.SimIntervals;
define table MCBook.SimIntervals;
   column Contrast Estimate StdErr tValue RawP OneP LowerCL UpperCL;

   define header h1;
      text "Estimated &confpct% Quantile = &quant";
      spill_margin;
%if (^&ANORM) %then %do;
      space=1;
%end;
      end;

%if (&ANORM) %then %do;
   define header h2;
      text "Asymptotic Normal Approximations";
      space=1;
      end;
%end;

   define column Contrast;
      header="Contrast";
      end;
   define column Estimate;
      header="Estimate"       format=D8. space=1;
      translate _val_ = ._ into '';
      end;
   define column StdErr;
      header="Standard Error" format=D8. space=1;
      translate _val_ = ._ into '';
      end;
   define column tValue;
      header="#t Value"       format=7.2;
      translate _val_ = .I into '  Infty',
         _val_ = .M into ' -Infty',
         _val_ = ._ into '';
      end;
```

```
   %if (&side = B) %then %do;
      define header ProbtHead;
         text " Pr > |t| ";
         start=RawP end=OneP just=c expand='-';
         end;
      %end;
   %else %if (&side = L) %then %do;
      define header ProbtHead;
         text " Pr < t ";
         start=RawP end=OneP just=c expand='-';
         end;
      %end;
   %else %do;
      define header ProbtHead;
         text " Pr > t ";
         start=RawP end=OneP just=c expand='-';
         end;
      %end;

   define column RawP;
      space=1 glue=10
      parent=Common.PValue header="Raw";
      translate _val_ = ._ into '';
      end;
   define column OneP;
      parent=Common.PValue header="Adjusted";
      translate _val_ = ._ into '';
      end;

   define header CLHead;
      text "&confpct% Confidence Interval";
      start=LowerCL end=UpperCL just=c;
      end;
   define LowerCL;
      translate _val_ = .M into '  -Infty';
      space=1 glue=10 format=D8. print_headers=off;
      end;
   define UpperCL;
      format=D8. print_headers=off;
      translate _val_ = .I into '   Infty';
      end;

   end;
run;

data _null_; set SimIntOut;
   file print ods=(template='MCBook.SimIntervals');
   put _ods_;
   run;
```

```
%end;

options notes;

%mend;
```

7.2 The *%SimTests* Macro

```
/*-----------------------------------------------------------------*/
/* Name:      SimTests                                             */
/* Title:     Simultaneous Hypothesis Tests for General Linear     */
/*            Functions, using Correlations and Constraints        */
/* Author:    Peter Westfall, westfall@ttu.edu                     */
/* Reference: Westfall, P.H. (1997). Multiple testing of           */
/*            general contrasts using logical constraints and      */
/*            correlations.  JASA 92, 299-306                      */
/* Release:   Version 7.01                                         */
/*-----------------------------------------------------------------*/
/* Inputs:                                                         */
/*                                                                 */
/*      NSAMP =  simulation size, with 20000 as default            */
/*                                                                 */
/*      SEED  =  random number seed, with 0 (clock time)           */
/*               as default                                        */
/*                                                                 */
/*      SIDE  = U, L or B, for upper-tailed, lower-tailed          */
/*              or two-tailed, respectively. SIDE=B is default.    */
/*                                                                 */
/*      TYPE  = LOGICAL or FREE, for logically constrained or      */
/*              unconstrained tests, respectively. TYPE=FREE       */
/*              is the default.                                    */
/*                                                                 */
/*  Additionally, %SimTests requires two further macros to be      */
/*  defined that use SAS/IML to construct the estimates and        */
/*  the contrasts of interest.  In particular, make sure the       */
/*  following two macros are defined before invoking               */
/*  %SimTests:                                                     */
/*                                                                 */
/*     %Estimate: Uses SAS/IML code to define                      */
/*        EstPar - (column) vector of estimated parameters         */
/*        Cov    - covariance matrix for the for the estimates     */
/*        df     - error degrees of freedom; set to 0 for          */
/*                 asymptotic analysis                             */
/*                                                                 */
/*     %Contrasts: Uses SAS/IML code to define                     */
/*        C      - matrix whose columns define the contrasts of    */
/*                 interest between the parameters                 */
/*        CLab   - (column) character vector whose elements        */
/*                 label the respective contrasts in C             */
/*                                                                 */
/*  You can either define these macros directly, or use the        */
/*  %MakeGLMStats macro to define them.                            */
/*                                                                 */
/*-----------------------------------------------------------------*/
/* Output:                                                         */
/*   The primary output is a dataset with one observation for      */
/*   each contrast and the following variables:                    */
/*                                                                 */
```

```
/*     Contrast - contrast label                                  */
/*     Estimate - contrast estimated value                        */
/*     StdErr   - standard error of estimate                      */
/*     tValue   - normalized estimate, Estimate/StdErr            */
/*     RawP     - non-multiplicity-adjusted p-value               */
/*     BonP     - Bonferroni multiplicity-adjusted p-value        */
/*     BonMult  - corresponding Bonferroni multiplier             */
/*     AdjP     - stepwise multiplicity-adjusted p-value          */
/*     SEAdjP   - standard error for AdjP                         */
/*                                                                */
/*   This dataset is also displayed as a formatted table, using */
/*   the ODS system.                                              */
/*                                                                */
/*  This macro also produces a data set called SUBSETS that has */
/*  has a variable STEPJ indicating the particular (ordered)    */
/*  hypothesis being considered; as well as variables           */
/*  (TEST1--TESTk) identifying the particular subset hypotheses */
/*  that contain the hypothesis indicated by the STEPJ variable,*/
/*  that do not contradict falsehood of the previous hypotheses.*/
/*  The order of the TEST1--TESTk variables is from most to     */
/*  least significant.                                           */
/*--------------------------------------------------------------*/

%macro SimTests(nsamp   = 20000  ,
                seed    = 0      ,
                side    = B      ,
                type    = FREE   ,
                options =        );
%global ANORM;

options nonotes;

proc iml;
%Estimates;
if (df <= 0) then call symput('ANORM','1');
else              call symput('ANORM','0');
%Contrasts;
C = C‘;
side="&side";
type="&type";
if side = "U" then C=-C;

EstCont = C*EstPar;
CovCont = C*Cov*C‘;
SECont = sqrt(vecdiag(CovCont));
tvals = EstCont/SECont;
if side = "B" then do;
   tvals = -abs(tvals);
   if df=0 then pvals = 2*probnorm(tvals);
      else pvals = 2*probt(tvals,df);
   end;
```

```
else do;
   if df=0 then pvals=probnorm(tvals);
      else pvals = probt(tvals,df);
   end;

k = nrow(c);
nests = nrow(EstPar);
call symput('k',char(k));
call symput('g',char(nests));
r  = rank(Pvals`);
ir = r;
ir[,r] = 1:nrow(PVals);
origord  = ir`         ;
cord     = c      [ir,];
clabord  = clab   [ir,];
tvalsord = tvals  [ir,];
pvalsord = pvals  [ir,];
ccord    = CovCont[ir,ir];
crrccord = inv(sqrt(diag(ccord)))*ccord*inv(sqrt(diag(ccord)));
ct = t(cord);

start ztrail;
   ii=1;
   zz=kk;
   do while(mod(zz,2)=0);
      ii=ii+1;
      zz=zz/2;
      end;
finish;

if type = "LOGICAL" then do;
do iout  = 1 to k-2;
   limit = 2**(k-iout-1);
   in=J(k-iout-1,1,0);
   zero =J(k,1,0);
   in1 =zero;
   y = ct[,1:iout];

   do kk=1 to limit;
      if kk=limit then in=j(k-iout-1,1,0);
      else do;
         run ztrail;
         in[ii,]=^in[ii,];
         end;

      locbin  = j(iout, 1, 0) // {1} // in;
      loc1 = loc(locbin);
      x = ct[,loc1];

      res = y - x*ginv(x`*x)*x`*y;
      ssemat = vecdiag(res`*res);
```

```
      if ssemat > .00000001 then do;
         if in1=0 then in1 = locbin;
         else do;
            check = in1 - repeat(locbin, 1, ncol(in1));
            diff = check[<>,] - check[><,];
            if min(diff) = 2 then  in1 = in1||locbin;
            else do;
               mindx = diff[,>:<];
               if check[+,mindx]=-1 then in1[,mindx] = locbin;
               end;
            end;
         end;
      end;
   in1  = in1‘;
   ncont = nrow(in1);
   in1 = j(ncont,1,iout+1)||in1;
   if iout = 1 then inbig = in1;
   else inbig = inbig//in1;
   end;
end;

big = j(1,k+1,1)//inbig;
lastset = j(1,1,k)||j(1,k-1,0)||{1};
big = big//lastset;
stepj = big[,1];
if type="FREE" then do;
   stepj = 1:k;
   stepj = stepj‘;
end;
SubsetK = big[,2:ncol(big)];
if type="FREE" then do;
   m = j(1,k,1);
   do i = 2 to k;
      r = j(1,i-1,0)||j(1,k-i+1,1);
      m= m//r;
   end;
SubsetK = m;
end;


subsets = subsetk||stepj;
create subsets var (("t1":"t&k")||"StepJ");
append from subsets;
nbig = nrow(big);
if type="LOGICAL" then des = design(big[,1]);
   else des=design(stepj);
if type="LOGICAL" then contonly = big[,2:k+1];
   else contonly = subsetk;
tcmpr = des*tvalsord;
h = root(crrccord);
```

```
if type="FREE" then nbig=k;
count = j(nbig,1,0);
countc = count;
countc2 = count;

contonly = ((contonly + des)  > 0);

if type="LOGICAL" then totals =  contonly[,+];
   else do; totals=k:1; totals=totals`; end;
if side = "B" then do;
    if df = 0 then bon = 2*(probnorm(tcmpr))#totals;
       else bon = 2*(probt(tcmpr,df))#totals;
       end;
   else do;
      if df = 0 then bon = (probnorm(tcmpr))#totals;
      else bon = (probt(tcmpr,df))#totals;
      end;

if &nsamp>0 then do;
file log;
do isim = 1 to &nsamp;
   if mod(isim,5000) = 0 then put isim;
   z = h`*rannor(j(k,1,&seed));
   if df=0 then s=1; else do;
      chi = 2*rangam(&seed,df/2);
      s = sqrt(chi/df);
      end;
   t = z/s;
   if side = "B" then t = -abs(t);
   try = (contonly#(j(nbig,1,1)*t`));
   try1 = (10000*(try=0)) + try;
   maxind =    (try1[,><] <= tcmpr);

   sumind = (try1 < ((tcmpr)*j(1,ncol(try),1)))[,+];
   countc = countc + sumind;
   countc2 = countc2 + sumind##2;
   count = count +  maxind;
end;

smpl = count/&nsamp;
cv = bon + smpl - countc/&nsamp;
avec = countc/&nsamp;
avec2 = countc2/&nsamp;
varx = smpl#(j(nrow(smpl),1,1)-smpl);
varz = avec2 - avec##2 + smpl - smpl##2 -2*avec#(j(nrow(smpl),1,1)-smpl);
covzx = (avec-smpl)#(j(nrow(smpl),1,1)-smpl);
a1 = varz+covzx;
a2 = varx+covzx;
atot = a1+a2;
atot = (atot=0) + atot;
a1 = a1/atot;
```

```
a2 = a2/atot;
atot = a1+a2;
a2 = a2+(atot=0);
gls = a1#smpl + a2#cv;

stdgls = sqrt(abs((a1##2#varx + a2##2#varz -2*a1#a2#covzx)/&nsamp));
stdsmpl = sqrt(varx/&nsamp);
stdcv = sqrt(abs(varz/&nsamp));
glsbig = des#(gls*j(1,k,1));
glsp = glsbig[<>,];
glsin = glsbig[<:>,];

stdgls = stdgls[glsin,];
glsptry = glsp‘;
smplbig = des#(smpl*j(1,k,1));
smplp = smplbig[<>,];
smplin = smplbig[<:>,];
stdsmpl = stdsmpl[smplin,];
cvbig = des#(cv*j(1,k,1));
cvp = cvbig[<>,];
cvin = cvbig[<:>,];
stdcv = stdcv[cvin,];

do i = 2 to k;
   if smplp[1,i] < smplp[1,i-1] then do;
      smplp[1,i] = smplp[1,i-1];
      stdsmpl[i,1] = stdsmpl[i-1,1];
      end;
   if cvp[1,i] < cvp[1,i-1] then do;
      cvp[1,i] = cvp[1,i-1];
      stdcv[i,1] = stdcv[i-1,1];
      end;
   if glsp[1,i] < glsp[1,i-1] then do;
      glsp[1,i] = glsp[1,i-1];
      stdgls[i,1] = stdgls[i-1,1];
      end;
end;

adjpsmpl = smplp‘;
adjpcv = cvp‘;
adjpgls = glsp‘;
adjp=adjpgls;
SEAdjp = stdgls#(stdgls>.00000001);
end;

bonbig = des#(bon*j(1,k,1));
bonp = bonbig[<>,];
bonmult = bonp‘/pvalsord;

do i = 2 to k;
   if bonp[1,i] < bonp[1,i-1] then bonp[1,i] = bonp[1,i-1];
```

```
   end;

rawp = pvalsord;
estimate = EstCont[ir,];
if side ="U" then estimate=-estimate;
stderr = SECont[ir,];
contrast = cord;
if side = "U" then contrast=-contrast;
adjpbon = bonp‘;
adjpbon = (adjpbon<1)#adjpbon +(adjpbon>=1);

if &nsamp>0 then do;
   outres =   origord
            ||contrast
            ||estimate
            ||stderr
            ||rawp
            ||bonmult
            ||adjpbon
            ||adjp
            ||SEAdjp;
   create SimTestOut var (    "OrigOrd"
                          ||("Est1":"Est&g")
                          ||"Estimate"
                          ||"StdErr"
                          ||"RawP"
                          ||"BonMult"
                          ||"BonP"
                          ||"AdjP"
                          ||"SEAdjP");
   append from outres;
   end;
else do;
   outres =   origord
            ||contrast
            ||estimate
            ||stderr
            ||rawp
            ||bonmult
            ||adjpbon;
   create SimTestOut var (    "OrigOrd"
                          ||("Est1":"Est&g")
                          ||"Estimate"
                          ||"StdErr"
                          ||"RawP"
                          ||"BonMult"
                          ||"BonP");
   append from outres;
   end;

create labels from clabord; append from clabord;
```

```
data SimTestOut; merge SimTestOut labels;
   rename col1=Contrast;
proc sort data=SimTestOut out=SimTestOut; by origord;
data SimTestOut; set SimTestOut; drop origord;
   run;

%if (^%index(%upcase(&options),NOPRINT)) %then %do;

proc template;
delete MCBook.SimTests;
define table MCBook.SimTests;
   column Contrast Estimate StdErr RawP BonP AdjP SEAdjP;

   define header h1;
      spill_margin;
%if (%upcase(&type) = LOGICAL) %then %do;
      text "Logically Constrained (Restricted Combinations) Step-Down Tests";
   %end;
%else %do;
      text "Unconstrained (Free Combinations) Step-Down Tests";
   %end;
%if (^&ANORM) %then %do;
      space=1;
%end;
      end;
%if (&ANORM) %then %do;
   define header h2;
      text "Asymptotic Normal Approximations";
      space=1;
      end;
%end;

   define column Contrast;
      header="Contrast";
      end;
   define column Estimate;
      header="Estimate"       format=D8. space=1;
      translate _val_ = ._ into '';
      end;
   define column StdErr;
      header="Standard Error" format=D8.;
      translate _val_ = ._ into '';
      end;

   %if (&nsamp) %then %let LastPValCol = AdjP;
   %else              %let LastPValCol = BonP;

   %if (&side = B) %then %do;
      define header ProbtHead;
         text " Pr > |t| ";
```

```
         start=Rawp end=&LastPValCol just=c expand='-';
         end;
      %end;
   %else %if (&side = L) %then %do;
      define header ProbtHead;
         text " Pr < t ";
         start=Rawp end=&LastPValCol just=c expand='-';
         end;
      %end;
   %else %do;
      define header ProbtHead;
         text " Pr > t ";
         start=Rawp end=&LastPValCol just=c expand='-';
         end;
      %end;

   define column RawP;
      space=1 glue=10
      parent=Common.PValue header="Raw";
      translate _val_ = ._ into '';
      end;
   define column BonP;
      space=1 glue=10
      parent=Common.PValue header="Bon";
      translate _val_ = ._ into '';
      end;
   define column AdjP;
      parent=Common.PValue header="Adj";
      translate _val_ = ._ into '';
      end;

   define column SEAdjP;
      header="SE(AdjP)" format=d8.;
      translate _val_ = ._ into '';
      end;

   end;
run;

data _null_; set SimTestOut;
   file print ods=(template='MCBook.SimTests');
   put _ods_;
   run;

%end;

options notes;

%mend;
```

7.3 The *%MakeGLMStats* Macro

```
/*-----------------------------------------------------------------*/
/* Name:      MakeGLMStats                                         */
/* Title:     Macro to create %Estimates and %Contrasts macros     */
/*            needed for %SimIntervals and %SimTests               */
/* Author:    Randy Tobias, sasrdt@sas.com                         */
/* Release:   Version 7.01                                         */
/*-----------------------------------------------------------------*/
/* Inputs:                                                         */
/*                                                                 */
/*    DATASET = Data set to be analyzed (required)                 */
/*                                                                 */
/*   CLASSVAR = Listing of classification variables. If absent,    */
/*              no classification variables are assumed            */
/*                                                                 */
/*      YVAR  = response variable (required)                       */
/*                                                                 */
/*     MODEL  = GLM model specification (required)                 */
/*                                                                 */
/*  CONTRASTS = CONTROL(effect), ALL(effect), or USER. This        */
/*              creates the %Contrasts macro unless you specify    */
/*              USER (the default), in which case you create       */
/*              the %Contrasts macro yourself                      */
/*                                                                 */
/*-----------------------------------------------------------------*/
/* Output:  This macro creates the %Estimates macro needed for     */
/* the %SimIntervals and %SimTests macros.  Additionally, if       */
/* you specify CONTRASTS = ALL or CONTROL, it also creates the     */
/* %Contrasts macro.  There is no other output.                    */
/*-----------------------------------------------------------------*/

%macro MakeGLMStats(dataset= , classvar= , yvar= , model= , contrasts=USER);
   %global nx yvar1 nlev icntl;

   options nonotes;

   %let yvar1 = &yvar;
   proc glmmod data=&dataset noprint outparm=parm outdesign=design;
      %if (%length(&classvar)) %then %do;
      class &classvar;
      %end;
      model &yvar = &model;
   data _null_; set parm; call symput('nx',_n_);
      run;

   %macro Estimates;
      use design;
      read all var ("col1":"col&nx") into X;
      read all var ("&yvar1")        into Y;
      XpXi   = ginv(X`*X);
      rankX  = trace(XpXi*(X`*X));
```

```
   n      = nrow(X);
   df     = n-rankX;
   EstPar = XpXi*X`*Y;
   mse    = ssq(Y-X*EstPar)/df;
   Cov    = mse*XpXi;
%mend;

%let ctype = %upcase(%scan(&contrasts,1));
%if (&ctype ^= USER) %then %do;
   %let effect = %scan(&contrasts,2);
   %if (&ctype = CONTROL) %then %do;
      %let icntl = %scan(&contrasts,3);
      %end;
   %end;

%if (&ctype ^= USER) %then %do;
   ods listing close;
   ods output LSMeanCoef=LSMeanCoef;
   proc glm data=&dataset;
      %if (%length(&classvar)) %then %do;
       class &classvar;
       %end;
       model &yvar = &model;
       lsmeans &effect / e;
   quit;
   ods listing;
   proc transpose data=LSMeanCoef out=temp;
      var Row:;
   data _null_; set temp;
      call symput('nlev',_n_);
   run;
   %end;

%if (&ctype = ALL) %then %do;
   %macro Contrasts; %global nlev;
      use LSMeanCoef; read all var ("Row1":"Row&nlev") into L;
      free C clab;
      do i = 1 to ncol(L)-1;
         do j = i+1 to ncol(L);
            C    = C    // L[,i]` - L[,j]`;
            clab = clab // (     trim(left(char(i,5)))
                           +'-'+trim(left(char(j,5))));
            end;
         end;
      C = C`;
      %mend;
   %end;
%if (&ctype = CONTROL) %then %do;
   %macro Contrasts; %global icntl;
      use LSMeanCoef; read all var ("Row1":"Row&nlev") into L;
      free C clab;
```

```
         j = &icntl;
         do i = 1 to ncol(L);
            if (i ^= j) then do;
               C    = C    // L[,i]‘ - L[,j]‘;
               clab = clab // (     trim(left(char(i,5)))
                               +’-’+trim(left(char(j,5))));
               end;
            end;
         C = C‘;
         %mend;
      %end;
   options notes;
%mend;
```

References

[1] Biesheuvel, E., Hothorn, L.(2001): Many-to-one comparisons in a stratified design maintaining the overall α level. *Biometrical Journal* (in press).

[2] Bretz, F. (1999): *Powerful Modifications of Williams' Test on Trend*. Dissertation, University of Hannover.

[3] Dean, A., Voss, D. (1999): *Design and Analysis of Experiments*. Springer Verlag, New York.

[4] Dunnett, C. W. (1955): A multiple comparison procedure for comparing several treatments with a control. *Journal of the American Statistical Association* 50, 1096-1121.

[5] Gomez, K. A., Gomez, A. A. (1984): *Statistical procedures for agricultural research.* Wiley, New York.

[6] Hochberg, Y., Tamhane, A. C. (1987): *Multiple Comparison Procedure*. John Wiley & Sons, Inc., New York.

[7] Holm, S. (1979): A simple sequentially rejective test procedure. *Scandinavian Journal of Statistics* 6, 65-70.

[8] Köhler, W., Schachtel, G., Voleske, P. (1996): *Biostatistik*. 2^{nd} edition, Springer Verlag, Berlin.

[9] Littell, R., Milliken, G. A., Stroup, W. W., Wolfinger , R. D. (1996): *SAS System for Mixed Models*. Cary, NC: SAS Institute Inc.

[10] Moore, D. S. (2000): *The Basic Practice of Statistics*. 2^{nd} edition, W. H. Freeman and Company.

[11] Mudra, A. (1958): *Statistische Methoden für landwirtschaftliche Versuche*. Parey Verlag, Berlin.

[12] Neter, J., Kutner, M. H., Nachtsheim, C. J., Wasserman, W., (1996): *Applied Linear Statistical Models*. 4^{th} edition, Irwin.

[13] O'Brien, P. C. (1984): Procedures for comparing samples with multiple endpoints. *Biometrics*, 40, 1079-1087.

[14] Piepho, H.-P. (2000): *Planen und Auswerten von Versuchen*. Lecture notes, University of Hohenheim.

[15] Shaffer, J.P. (1986): Modified sequentially rejective multiple test procedures. *Journal of the American Statistical Association* 81, 826-831.

[16] Snedecor, G. W., Cochran, W. G. (1967): *Statistical methods*. Iowa State University Press, Ames.

[17] Tamhane, A. C., Dunlop, D. D. (2000): *Statistics and data analysis*. Prentice Hall.

[18] Tukey, J. W. (1953): The problem of multiple comparisons. *Mimeographed Notes*, Princeton University.

[19] Westfall, P. H., Tobias, R. D., Rom, D., Wolfinger, R. D., Hochberg, Y. (1999): *Multiple Comparisons and Multiple Tests Using the SAS System*. Cary, NC: SAS Institute Inc.

[20] Westfall, P., Young, S.S. (1993): *Resampling-Based Multiple Testing*. John Wiley & Sons, Inc., New York.

[21] Westfall, P. (1997): Multiple Testing of General Contrasts Using Logical Constraints and Correlations. *Journal of the American Statistical Association* 92, 299-307.

Hiermit erkläre ich an Eides Statt, daß ich diese Arbeit nur mit den aufgeführten Hilfsmitteln erstellt habe. Die Arbeit wurde bisher keiner anderen Prüfungskommission vorgelegt.

(Rethmar, den 17. April 2001, Cornelia Schratz)