Chapter 69
The VARCOMP Procedure

Chapter Table of Contents

Chapter 69
The VARCOMP Procedure

Overview

The VARCOMP procedure handles general linear models that have random effects. Random effects are classification effects with levels that are assumed to be randomly selected from an infinite population of possible levels. PROC VARCOMP estimates the contribution of each of the random effects to the variance of the dependent variable.

A single MODEL statement specifies the dependent variables and the effects: main effects, interactions, and nested effects. The effects must be composed of class variables; no continuous variables are allowed on the right side of the equal sign.

You can specify certain effects as fixed (nonrandom) by putting them first in the MODEL statement and indicating the number of fixed effects with the FIXED= option. An intercept is always fitted and assumed fixed. Except for the effects specified as fixed, all other effects are assumed to be random, and their contribution to the model can be thought of as an observation from a distribution that is normally and independently distributed.

The dependent variables are grouped based on the similarity of their missing values. Each group of dependent variables is then analyzed separately. The columns of the design matrix **X** are formed in the same order in which the effects are specified in the MODEL statement. No reparameterization is done. Thus, the columns of **X** contain only 0s and 1s.

You can specify four methods of estimation in the PROC VARCOMP statement using the METHOD= option. They are TYPE1 (based on computation of Type I sum of squares for each effect), MIVQUE0, Maximum Likelihood (METHOD=ML), and Restricted Maximum Likelihood (METHOD=REML).

Getting Started

Analyzing the Cure Rate of Rubber

This example, using data from Hicks (1973), concerns an experiment to determine the sources of variability in cure rates of rubber. The goal of the experiment was to find out if the different laboratories contributed more to the variance of cure rates than did the different batches of raw materials. This information would be useful in trying to control the cure rate of the final product because it would provide insights into the sources of the variability in cure rates. The rubber used was cured at three temperatures, which were taken to be fixed. Three laboratories were chosen at random, and three different batches of raw material were tested at each combination of temperature and laboratory. The following statements read the data into the SAS data set Cure.

```
title 'Analyzing the Cure Rate of Rubber';
data Cure;
   input Lab Temp Batch $ Cure @@;
   datalines;
1 145 A 18.6   1 145 A 17.0   1 145 A 18.7   1 145 A 18.7
1 145 B 14.5   1 145 B 15.8   1 145 B 16.5   1 145 B 17.6
1 145 C 21.1   1 145 C 20.8   1 145 C 21.8   1 145 C 21.0
1 155 A  9.5   1 155 A  9.4   1 155 A  9.5   1 155 A 10.0
1 155 B  7.8   1 155 B  8.3   1 155 B  8.9   1 155 B  9.1
1 155 C 11.2   1 155 C 10.0   1 155 C 11.5   1 155 C 11.1
1 165 A  5.4   1 165 A  5.3   1 165 A  5.7   1 165 A  5.3
1 165 B  5.2   1 165 B  4.9   1 165 B  4.3   1 165 B  5.2
1 165 C  6.3   1 165 C  6.4   1 165 C  5.8   1 165 C  5.6
2 145 A 20.0   2 145 A 20.1   2 145 A 19.4   2 145 A 20.0
2 145 B 18.4   2 145 B 18.1   2 145 B 16.5   2 145 B 16.7
2 145 C 22.5   2 145 C 22.7   2 145 C 21.5   2 145 C 21.3
2 155 A 11.4   2 155 A 11.5   2 155 A 11.4   2 155 A 11.5
2 155 B 10.8   2 155 B 11.1   2 155 B  9.5   2 155 B  9.7
2 155 C 13.3   2 155 C 14.0   2 155 C 12.0   2 155 C 11.5
2 165 A  6.8   2 165 A  6.9   2 165 A  6.0   2 165 A  5.7
2 165 B  6.0   2 165 B  6.1   2 165 B  5.0   2 165 B  5.2
2 165 C  7.7   2 165 C  8.0   2 165 C  6.6   2 165 C  6.3
3 145 A 19.7   3 145 A 18.3   3 145 A 16.8   3 145 A 17.1
3 145 B 16.3   3 145 B 16.7   3 145 B 14.4   3 145 B 15.2
3 145 C 22.7   3 145 C 21.9   3 145 C 19.3   3 145 C 19.3
3 155 A  9.3   3 155 A 10.2   3 155 A  9.8   3 155 A  9.5
3 155 B  9.1   3 155 B  9.2   3 155 B  8.0   3 155 B  9.0
3 155 C 11.3   3 155 C 11.0   3 155 C 10.9   3 155 C 11.4
3 165 A  6.7   3 165 A  6.0   3 165 A  5.0   3 165 A  4.8
3 165 B  5.7   3 165 B  5.5   3 165 B  4.6   3 165 B  5.4
3 165 C  6.6   3 165 C  6.5   3 165 C  5.9   3 165 C  5.8
;
```

The variables Lab, Temp, and Batch contain levels of laboratory, temperature, and batch, respectively. The Cure variable contains the response values.

The following SAS statements perform a restricted maximum-likelihood variance component analysis.

```
proc varcomp method=reml;
   class Temp Lab Batch;
   model Cure=Temp|Lab Batch(Lab Temp) / fixed=1;
run;
```

The FIXED=1 option indicates that the first factor, Temp, is fixed. The effect specification Temp|Lab is equivalent to putting the three terms Temp, Lab, and Temp*Lab in the model. Batch(Lab Temp) is equivalent to putting Batch(Temp*Lab) in the MODEL statement. The results of this analysis are displayed in Figure 69.1 through Figure 69.4.

```
                    Analyzing the Cure Rate of Rubber

                 Variance Components Estimation Procedure

                         Class Level Information

                 Class         Levels    Values

                 Temp               3    145 155 165

                 Lab                3    1 2 3

                 Batch              3    A B C


                     Number of observations    108


                      Dependent Variable:    Cure
```

Figure 69.1. Class Level Information

Figure 69.1 provides information about the variables used in the analysis and the number of observations and specifies the dependent variable.

```
                    Analyzing the Cure Rate of Rubber

                 Variance Components Estimation Procedure

                              REML Iterations

                                                      Var(Batch(Temp*
Iteration        Objective         Var(Lab)    Var(Temp*Lab)          Lab))       Var(Error)

        0    13.4500060254     0.5094464340                0     2.4004888633     0.5787185225
        1    13.0898262160     0.3194348317                0     2.0869636935     0.6016005334
        2    13.0893125570     0.3176048001                0     2.0738906134     0.6026217204
        3    13.0893125555     0.3176017115                0     2.0738685461     0.6026234568


               Convergence criteria met.
```

Figure 69.2. Iteration History

The “REML Iterations” table, shown in Figure 69.2, displays the iteration history, which includes the value of the objective function associated with REML and the values of the variance components at each iteration.

Analyzing the Cure Rate of Rubber

Variance Components Estimation Procedure

REML Estimates

Variance Component	Estimate
Var(Lab)	0.31760
Var(Temp*Lab)	0
Var(Batch(Temp*Lab))	2.07387
Var(Error)	0.60262

Figure 69.3. REML Estimates

Figure 69.3 displays the REML estimates of the variance components.

Analyzing the Cure Rate of Rubber

Variance Components Estimation Procedure

Asymptotic Covariance Matrix of Estimates

	Var(Lab)	Var(Temp*Lab)	Var(Batch(Temp*Lab))	Var(Error)
Var(Lab)	0.32452	0	-0.04998	1.026E-12
Var(Temp*Lab)	0	0	0	0
Var(Batch(Temp*Lab))	-0.04998	0	0.45042	-0.0022417
Var(Error)	1.026E-12	0	-0.0022417	0.0089668

Figure 69.4. Covariance Matrix for REML Estimates

The “Asymptotic Covariance Matrix of Estimates” table in Figure 69.4 displays the asymptotic covariance matrix of the REML estimates.

The results of the analysis show that the variance attributable to Batch(Temp*Lab) (with a variance component of 2.0739) is considerably larger than the variance attributable to Lab (0.3176). Therefore, attempts to reduce the variability of cure rates should concentrate on improving the homogeneity of the batches of raw material used rather than standardizing the practices or equipment within the laboratories. Also, note that since the Batch(Temp*Lab) variance is considerably larger than the experimental error (Var(Error)=0.6026), the Batch(Temp*Lab) variability plays an important part in the overall variability of the cure rates.

Syntax

The following statements are available in PROC VARCOMP.

> **PROC VARCOMP** < *options* > **;**
> **CLASS** *variables* **;**
> **MODEL** *dependent* = < *effects* > < / *options* > **;**
> **BY** *variables* **;**

Only one MODEL statement is allowed. The BY, CLASS, and MODEL statements are described after the PROC VARCOMP statement.

PROC VARCOMP Statement

> **PROC VARCOMP** < *options* >**;**

This statement invokes the VARCOMP procedure. You can specify the following options in the PROC VARCOMP statement.

DATA=*SAS-data-set*
specifies the input SAS data set to use. If this option is omitted, the most recently created SAS data set is used.

EPSILON=*number*
specifies the convergence value of the objective function for METHOD=ML or METHOD=REML. By default, EPSILON=1E−8.

MAXITER=*number*
specifies the maximum number of iterations for METHOD=ML or METHOD=REML. By default, MAXITER=50.

METHOD=TYPE1 | MIVQUE0 | ML | REML
specifies which of the four methods (TYPE1, MIVQUE0, ML, or REML) you want to use. By default, METHOD= MIVQUE0. For more information see the “Computational Methods” section on page 3631.

BY Statement

> **BY** *variables* **;**

You can specify a BY statement with PROC VARCOMP to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives.

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the VARCOMP procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure (in base SAS software).

For more information on the BY statement, see the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, see the discussion in the *SAS Procedures Guide*.

CLASS Statement

CLASS *variables* **;**

The CLASS statement specifies the classification variables to be used in the analysis. All effects in the MODEL statement must be composed of effects that appear in the CLASS statement. Class variables can be either numeric or character; if they are character, only the first 16 characters are used.

Numeric class variables are not restricted to integers since a variable's format determines the levels. For more information, see the discussion of the FORMAT statement in *SAS Language Reference: Dictionary*.

MODEL Statement

MODEL *dependent* = < *effects* > < / *option* > **;**

The MODEL statement gives the dependent variables and independent effects. If you specify more than one dependent variable, a separate analysis is performed for each one. The independent effects are limited to main effects, interactions, and nested effects; no continuous effects are allowed. All independent effects must be composed of effects that appear in the CLASS statement. Effects are specified in the VARCOMP procedure in the same way as described for the ANOVA procedure. Only one MODEL statement is allowed.

Only one option is available in the MODEL statement.

FIXED=*n*

tells the VARCOMP procedure that the first *n* effects in the MODEL statement are fixed effects. The remaining effects are assumed to be random. By default, PROC VARCOMP assumes that all effects are random in the model. Keep in mind that if you use bar notation and, for example, specify Y=A|B / FIXED=2, then A*B is considered a random effect.

Details

Missing Values

If an observation has a missing value for any variable used in the independent effects, then the analyses of all dependent variables omit this observation. An observation is deleted from the analysis of a given dependent variable if the observation's value for that dependent variable is missing. Note that a missing value in one dependent variable does not eliminate an observation from the analysis of the other dependent variables.

During processing, PROC VARCOMP groups the dependent variables on their missing values across observations so that sums of squares and cross products can be computed in the most efficient manner.

Fixed and Random Effects

Central to the idea of variance components models is the idea of fixed and random effects. Each effect in a variance components model must be classified as either a fixed or a random effect. Fixed effects arise when the levels of an effect constitute the entire population about which you are interested. For example, if a plant scientist is comparing the yields of three varieties of soybeans, then Variety would be a fixed effect, providing that the scientist was concerned about making inferences on only these three varieties of soybeans. Similarly, if an industrial experiment focused on the effectiveness of two brands of a machine, Machine would be a fixed effect only if the experimenter's interest did not go beyond the two machine brands.

On the other hand, an effect is classified as a random effect when you want to make inferences on an entire population, and the levels in your experiment represent only a sample from that population. Psychologists comparing test results between different groups of subjects would consider Subject as a random effect. Depending on the psychologists' particular interest, the Group effect might be either fixed or random. For example, if the groups are based on the sex of the subject, then Sex would be a fixed effect. But if the psychologists are interested in the variability in test scores due to different teachers, then they might choose a random sample of teachers as being representative of the total population of teachers, and Teacher would be a random effect. Note that, in the soybean example presented earlier, if the scientists are interested in making inferences on the entire population of soybean varieties and randomly choose three varieties for testing, then Variety would be a random effect.

If all the effects in a model (except for the intercept) are considered random effects, then the model is called a *random effects model*; likewise, a model with only fixed effects is called a *fixed-effects model*. The more common case, where some factors are fixed and others are random, is called a *mixed model*. In PROC VARCOMP, by default, effects are assumed to be random. You specify which effects are fixed by using the FIXED= option in the MODEL statement. In general, if an interaction or nested effect contains any effect that is random, then the interaction or nested effect should be considered as a random effect as well.

In the linear model, each level of a fixed effect contributes a fixed amount to the expected value of the dependent variable. What makes a random effect different is that each level of a random effect contributes an amount that is viewed as a sample from a population of normally distributed variables, each with mean 0, and an unknown variance, much like the usual random error term that is a part of all linear models. The estimate of the variance associated with the random effect is known as the *variance component* because it is measuring the part of the overall variance contributed by that effect. Thus, PROC VARCOMP estimates the variance of the random variables that are associated with the random effects in your model, and the variance components tell you how much each of the random factors contributes to the overall variability in the dependent variable.

Negative Variance Component Estimates

The variance components estimated by PROC VARCOMP should theoretically be nonnegative because they are assumed to represent the variance of a random variable. Nevertheless, when you are using METHOD=MIVQUE0 (the default) or METHOD=TYPE1, some estimates of variance components may become negative. (Due to the nature of the algorithms used for METHOD=ML and METHOD=REML, negative estimates are constrained to zero.) These negative estimates may arise for a variety of reasons:

- The variability in your data may be large enough to produce a negative estimate, even though the true value of the variance component is positive.
- Your data may contain outliers. Refer to Hocking (1983) for a graphical technique for detecting outliers in variance components models using the SAS System.
- A different model for interpreting your data may be appropriate. Under some statistical models for variance components analysis, negative estimates are an indication that observations in your data are negatively correlated. Refer to Hocking (1984) for further information about these models.

Assuming that you are satisfied that the model PROC VARCOMP is using is appropriate for your data, it is common practice to treat negative variance components as if they are zero.

Computational Methods

Four methods of estimation can be specified in the PROC VARCOMP statement using the METHOD= option. They are described in the following sections.

The Type I Method

This method (METHOD=TYPE1) computes the Type I sum of squares for each effect, equates each mean square involving only random effects to its expected value, and solves the resulting system of equations (Gaylor, Lucas, and Anderson 1970). The $\mathbf{X}'\mathbf{X} \mid \mathbf{X}'\mathbf{Y}$ matrix is computed and adjusted in segments whenever memory is not sufficient to hold the entire matrix.

The MIVQUE0 Method

Based on the technique suggested by Hartley, Rao, and LaMotte (1978), the MIVQUE0 method (METHOD=MIVQUE0) produces unbiased estimates that are invariant with respect to the fixed effects of the model and that are locally best quadratic unbiased estimates given that the true ratio of each component to the residual error component is zero. The technique is similar to TYPE1 except that the random effects are adjusted only for the fixed effects. This affords a considerable timing advantage over the TYPE1 method; thus, MIVQUE0 is the default method used in PROC VARCOMP. The $\mathbf{X}'\mathbf{X}|\mathbf{X}'\mathbf{Y}$ matrix is computed and adjusted in segments whenever memory is not sufficient to hold the entire matrix. Each element (i, j) of the form

$$\mathrm{SSQ}(\mathbf{X}_i'\mathbf{M}\mathbf{X}_j)$$

is computed, where

$$\mathbf{M} = \mathbf{I} - \mathbf{X}_0(\mathbf{X}_0'\mathbf{X}_0)^-\mathbf{X}_0'$$

and where $\mathbf{X}_0$ is part of the design matrix for the fixed effects, $\mathbf{X}_i$ is part of the design matrix for one of the random effects, and SSQ is an operator that takes the sum of squares of the elements. For more information refer to Rao (1971, 1972) and Goodnight (1978).

The Maximum Likelihood Method

The Maximum Likelihood method (METHOD=ML) computes maximum-likelihood estimates of the variance components; refer to Searle, Casella, and McCulloch (1992). The computing algorithm makes use of the W-transformation developed by Hemmerle and Hartley (1973). The procedure uses a Newton-Raphson algorithm, iterating until the log-likelihood objective function converges.

The objective function for METHOD=ML is $\ln(|\mathbf{V}|) + \mathbf{r}'\mathbf{V}^{-1}\mathbf{r}$, where

$$\mathbf{V} = \sigma_0^2\mathbf{I} + \sum_{i=1}^{n_r} \sigma_i^2\mathbf{X}_i\mathbf{X}_i'$$

and where σ_0^2 is the residual variance, n_r is the number of random effects in the model, σ_i^2 represents the variance components, $\mathbf{X}_i$ is part of the design matrix for one of the random effects, and

$$\mathbf{r} = \mathbf{y} - \mathbf{X}_0(\mathbf{X}_0'\mathbf{V}^{-1}\mathbf{X}_0)^{-}\mathbf{X}_0'\mathbf{V}^{-1}\mathbf{y}$$

is the vector of residuals.

The Restricted Maximum Likelihood Method

The Restricted Maximum Likelihood Method (METHOD=REML) is similar to the maximum likelihood method, but it first separates the likelihood into two parts: one that contains the fixed effects and one that does not (Patterson and Thompson 1971). The procedure uses a Newton-Raphson algorithm, iterating until convergence is reached for the log-likelihood objective function of the portion of the likelihood that does not contain the fixed effects. Using notation from earlier methods, the objective function for METHOD=REML is $\ln(|\mathbf{V}|)+\mathbf{r}'\mathbf{V}^{-1}\mathbf{r}+\ln(|\mathbf{X}_0'\mathbf{V}^{-1}\mathbf{X}_0|)$. Refer to Searle, Casella, and McCulloch (1992) for additional details.

Displayed Output

PROC VARCOMP displays the following items:

- Class Level Information for verifying the levels and number of observations in your data
- for METHOD=TYPE1, an analysis-of-variance table with Source, DF, Type I Sum of Squares, Type I Mean Square, and Expected Mean Square, and a table of Type I variance component estimates
- for METHOD=MIVQUE0, the SSQ Matrix containing sums of squares of partitions of the $\mathbf{X}'\mathbf{X}$ crossproducts matrix adjusted for the fixed effects
- for METHOD=ML and METHOD=REML, the iteration history, including the objective function, as well as variance component estimates
- for METHOD=ML and METHOD=REML, the estimated Asymptotic Covariance Matrix of the variance components
- a table of variance component estimates

ODS Table Names

PROC VARCOMP assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 15, "Using the Output Delivery System."

Table 69.1. ODS Tables Produced in PROC VARCOMP

ODS Table Name	Description	Statement
ANOVA	Type 1 analysis of variance	METHOD=TYPE1
AsyCov	Asymptotic covariance matrix of estimates	METHOD=ML or REML
ClassLevels	Class level information	default
ConvergenceStatus	Convergence status	default
DepVar	Dependent variable (one variable)	METHOD=TYPE1, REML, or ML
DepVar$n$	Dependent variable n (multiple variables)	METHOD=TYPE1, ML, or REML
DependentInfo	Dependent variable info (multiple variables)	METHOD=MIVQUE0
Estimates	Variance component estimates (one variable)	default
Estimates$n$	Variance component estimates (multiple variables)	default
IterHistory	Iteration history	METHOD=ML or REML
ML	ML or REML estimates	METHOD=ML or REML
NObs	Number of observations	default
SSCP	Sum of squares matrix (one variable)	METHOD=MIVQUE0
SSCPn	Sum of squares matrix (multiple variables)	METHOD=MIVQUE0
Type1	Type 1 estimates	METHOD=TYPE1

Relationship to PROC MIXED

The MIXED procedure effectively performs the same analyses as PROC VARCOMP and many others, including Type I, Type II, and Type III tests of fixed effects, confidence limits, customized contrasts, and least-squares means. Furthermore, continuous variables are permitted as both fixed and random effects in PROC MIXED, and numerous other covariance structures besides variance components are available.

To translate PROC VARCOMP code into PROC MIXED code, move all random effects to the RANDOM statement in PROC MIXED. For example, the syntax for the example in the "Getting Started" section on page 3624 is as follows:

```
proc mixed;
   class Temp Lab Batch;
   model Cure = Temp;
   random Lab Temp*Lab Batch(Lab Temp);
run;
```

REML is the default estimation method in PROC MIXED, and you can specify other methods using the METHOD= option.

Example

Example 69.1. Using the Four Estimation Methods

In this example, a and b are classification variables and y is the dependent variable. a is declared fixed, and b and a*b are random. Note that this design is unbalanced because the cell sizes are not all the same. PROC VARCOMP is invoked four times, once for each of the estimation methods. The data are from Hemmerle and Hartley (1973). The following statements produce Output 69.1.1.

```
data a;
   input a b y @@;
   datalines;
1 1 237   1 1 254   1 1 246   1 2 178   1 2 179
2 1 208   2 1 178   2 1 187   2 2 146   2 2 145   2 2 141
3 1 186   3 1 183   3 2 142   3 2 125   3 2 136
;

proc varcomp method=type1;
   class a b;
   model y=a|b / fixed=1;
run;

proc varcomp method=mivque0;
   class a b;
   model y=a|b / fixed=1;
run;

proc varcomp method=ml;
   class a b;
   model y=a|b / fixed=1;
run;
```

```
proc varcomp method=reml;
   class a b;
   model y=a|b / fixed=1;
run;
```

Output 69.1.1. VARCOMP Procedure: Method=TYPE1

```
                    Variance Components Estimation Procedure

                           Class Level Information

                        Class         Levels    Values

                        a                  3    1 2 3

                        b                  2    1 2


                         Number of observations    16


                           Dependent Variable:    y
```

The “Class Level Information” table displays the levels of each variable specified in the CLASS statement. You can check this table to make sure the data are input correctly.

```
                      Variance Components Estimation Procedure

                            Type 1 Analysis of Variance

                                 Sum of
Source                  DF      Squares   Mean Square  Expected Mean Square

a                        2        11736   5868.218750  Var(Error) + 2.725 Var(a*b) + 0.1 Var(b) + Q(a)
b                        1        11448         11448  Var(Error) + 2.6308 Var(a*b) + 7.8 Var(b)
a*b                      2   299.041026    149.520513  Var(Error) + 2.5846 Var(a*b)
Error                   10   786.333333     78.633333  Var(Error)
Corrected Total         15        24270             .  .
```

The Type I analysis of variance consists of sequentially partitioning the total sum of squares. The mean square is the sum of squares divided by the degrees of freedom, and the expected mean square is the expected value of the mean square under the mixed model. The “Q” notation in the expected mean squares refers to a quadratic form in parameters of the parenthesized effect.

```
                    Variance Components Estimation Procedure

                               Type 1 Estimates

                      Variance Component        Estimate

                      Var(b)                      1448.4
                      Var(a*b)                  27.42659
                      Var(Error)                78.63333
```

The Type I estimates of the variance components result from solving the linear system of equations established by equating the observed mean squares to their expected values.

Output 69.1.2. VARCOMP Procedure: Method=MIVQUE0

```
                 Variance Components Estimation Procedure

                        Class Level Information

                     Class         Levels    Values

                     a                  3    1 2 3

                     b                  2    1 2


                      Number of observations    16
```

The "Class Level Information" is the same as before.

```
                 Variance Components Estimation Procedure

                          MIVQUE(0) SSQ Matrix

Source                b             a*b           Error               y

b              60.84000        20.52000         7.80000         89295.4
a*b            20.52000        20.52000         7.80000         30181.3
Error           7.80000         7.80000        13.00000         12533.5
```

The MIVQUE0 sums-of-squares matrix is displayed in the previous table.

```
                 Variance Components Estimation Procedure

                          MIVQUE(0) Estimates

                    Variance Component             y

                    Var(b)                    1466.1
                    Var(a*b)               -35.49170
                    Var(Error)             105.73660
```

The MIVQUE0 estimates result from solving the equations established by the MIVQUE0 SSQ matrix. Note that the estimate of the variance component for the interaction effect, Var(a*b), is negative for this example.

Output 69.1.3. VARCOMP Procedure: Method=ML

```
                    Variance Components Estimation Procedure

                            Class Level Information

                         Class         Levels    Values

                         a                  3    1 2 3

                         b                  2    1 2


                          Number of observations    16


                            Dependent Variable:    y
```

The “Class Level Information” is the same as before.

```
                    Variance Components Estimation Procedure

                         Maximum Likelihood Iterations

Iteration        Objective            Var(b)         Var(a*b)       Var(Error)

        0    78.3850371200        1031.49070                0    74.3909717935
        1    78.2637043807    732.3606453635                0    77.4011688154
        2    78.2635471161    723.6867470850                0    77.5301774839
        3    78.2635471152    723.6658365289                0    77.5304926877


       Convergence criteria met.
```

The Newton-Raphson algorithm used by PROC VARCOMP requires three iterations to converge to the maximum likelihood estimates.

```
                    Variance Components Estimation Procedure

                               Maximum Likelihood
                                   Estimates

                            Variance
                            Component       Estimate

                            Var(b)        723.66584
                            Var(a*b)              0
                            Var(Error)     77.53049
```

The ML estimate of Var(a*b) is zero for this example, and the other two estimates are smaller than their Type I and MIVQUE0 counterparts.

```
                    Variance Components Estimation Procedure

                   Asymptotic Covariance Matrix of Estimates

                              Var(b)        Var(a*b)      Var(Error)

          Var(b)           537826.1               0      -107.33905
          Var(a*b)                0               0               0
          Var(Error)      -107.33905              0       858.71104
```

One benefit of using likelihood-based methods is that an approximate covariance matrix is available from the matrix of second derivatives evaluated at the ML solution. This covariance matrix is valid asymptotically and can be unreliable in small samples.

Here the variance component estimates for B and the Error are negatively correlated and the elements for Var(a*b) are set to zero because the estimate equals zero. Also, the very large variance for Var(b) indicates a lot of uncertainty about the estimate for Var(b), and one contributing explanation is that B has only two levels in this data set.

Output 69.1.4. VARCOMP Procedure: Method=REML

```
                    Variance Components Estimation Procedure

                            Class Level Information

                         Class         Levels    Values

                         a                  3    1 2 3

                         b                  2    1 2


                          Number of observations    16


                            Dependent Variable:    y
```

The “Class Level Information” is the same as before.

```
                    Variance Components Estimation Procedure

                                REML Iterations

Iteration        Objective          Var(b)         Var(a*b)       Var(Error)

        0    63.4134144942      1269.52701                0    91.5581191305
        1    63.0446869787      1601.84199    32.7632417174    76.9355562461
        2    63.0311530508      1468.82932    27.2258186561    78.7548276319
        3    63.0311265148      1464.33646    26.9564053003    78.8431476502
        4    63.0311265127      1464.36727    26.9588525177    78.8423898761


       Convergence criteria met.
```

The REML optimization requires four iterations to converge.

```
                    Variance Components Estimation Procedure

                                 REML Estimates

                            Variance
                            Component        Estimate

                            Var(b)             1464.4
                            Var(a*b)         26.95885
                            Var(Error)       78.84239
```

The REML estimates are all larger than the corresponding ML estimates (adjusting for potential downward bias) and are fairly similar to the Type I estimates.

```
                    Variance Components Estimation Procedure

                   Asymptotic Covariance Matrix of Estimates

                               Var(b)        Var(a*b)      Var(Error)

           Var(b)           4401703.8         1.29359      -273.39651
           Var(a*b)           1.29359          3559.1      -502.85157
           Var(Error)      -273.39651      -502.85157          1249.7
```

The Error variance component estimate is negatively correlated with the other two variance component estimates, and the estimated variances are all larger than their ML counterparts.

References

Gaylor, D.W., Lucas, H.L., and Anderson, R.L. (1970), "Calculation of Expected Mean Squares by the Abbreviated Doolittle and Square Root Methods," *Biometrics*, 26, 641–55.

Goodnight, J.H. (1978), *Computing MIVQUE0 Estimates of Variance Components*, SAS Technical Report R-105. Cary, NC: SAS Institute Inc.

Goodnight, J.H. and Hemmerle, W.J. (1979), "A Simplified Algorithm for the W-Transformation in Variance Component Estimation," *Technometrics*, 21, 265–268.

Hartley, H.O., Rao, J.N.K., and LaMotte, L. (1978), "A Simple Synthesis-Based Method of Variance Component Estimation," *Biometrics*, 34, 233–244.

Hemmerle, W.J. and Hartley, H.O. (1973), "Computing Maximum Likelihood Estimates for the Mixed AOV Model Using the W-Transformation," *Technometrics*, 15, 819–831.

Hicks, C.R. (1973), *Fundamental Concepts in the Design of Experiments*, New York: Holt, Rinehart and Winston, Inc.

Hocking, R.R. (1983), "A Diagnostic Tool for Mixed Models with Applications to Negative Estimates of Variance Components," *Proceedings of the Eighth Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc., 8, 711–716.

Hocking, R.R. (1984), *The Analysis of Linear Models*, Monterey, CA: Brooks-Cole Publishing Co.

Patterson, H.D. and Thompson, R. (1971), "Recovery of Inter-Block Information When Block Sizes Are Unequal," *Biometrika*, 58, 545–554.

Rao, C.R. (1971), "Minimum Variance Quadratic Unbiased Estimation of Variance Components," *Journal of Multivariate Analysis*, 1, 445–456.

Rao, C.R. (1972), "Estimation of Variance and Covariance Components in Linear Models," *Journal of the American Statistical Association*, 67, 112–115.

Searle, S.R., Casella, G., and McCulloch, C.E. (1992), *Variance Components*, New York: John Wiley and Sons, Inc.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/STAT® User's Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS/STAT® User's Guide, Version 8

ISBN 1–58025–494–2

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999

The Institute is a private company devoted to the support and further development of its software and related services.