Random and Mixed Effects Macros for MacAnova

Technical Report #624

School of Statistics

University of Minnesota

Gary W. Oehlert

Random and Mixed Effects Macros for MacAnova
Technical Report #624
School of Statistics
University of Minnesota

Gary W. Oehlert

March 1998

Abstract

MacAnova is a freely available program for Macintosh, Windows, and Unix platforms with broad data analysis and linear algebra capabilities. The built in ANOVA facilities are generally quite good for fixed effects models, but there are no built in functions for random or mixed effects ANOVA. There are now three macros (`ems`, `varcomp`, and `mixed`) available in the `design.mac` macro file. These macros compute expected mean squares, estimate variance components, and perform mixed effects ANOVA. This note announces the availability of the macros, gives background for when they can be used, and describes their use.

1 Introduction

The MacAnova program (`http://www.stat.umn.edu/~gary/macanova/macanova.home.html`) has several functions for fitting linear models and generalized linear models. The `anova()` function computes an analysis of variance table with sources, degrees of freedom, sums of squares, and mean squares. For example,

```
Cmd> anova("y = A + C + A.C + B + A.B")
```

computes and prints the ANOVA decomposition of the variation in y for the main effects of factors A, B, and C, and the two factor interactions AB and AC. Any variation and degrees of freedom not explained by the model are put in a final term labeled ERROR1.

By default, `anova()` computes no F-statistics or p-values. These can be requested by adding the keyword phase `fstats:T` to the `anova()` command:

```
Cmd> anova("y = A + C + A.C + B + A.B",fstats:T)
```

The output will now include columns for F-statistics and p-values. The denominator for any mean square is the next term labeled ERRORk in the model. For this model, there is only a single error term (ERROR1, the last term), so all F-statistics are computed using ERROR1 in the denominator. This use of ERROR1 is appropriate for fixed effects models.

We may choose to label terms in the ANOVA as error terms. For example,

```
Cmd> anova("y = A + C + E(A.C) + B + A.B",fstats:T)
```

will compute the sums of squares and df exactly as before, but the line for the AC interaction will be labeled as ERROR1, and the last line of the ANOVA (formerly labeled ERROR1) will now be labeled ERROR2. This would be appropriate for a split plot design, where factor A is the whole plot treatment and C is a blocking factor for whole plots. The term AC is then the whole plot error. F-statistics in `anova()` are computed using the next error term in the model. Thus A would be tested against ERROR1 (AC), and B would be tested against ERROR2.

This is the extent to which errors can be specified using `anova()` in MacAnova. (Any term may be specified as the source of error for the `contrast()` and `secoefs()` commands, which compute contrast estimates, standard errors, and sums of squares, and model coefficients and their standard errors respectively.) Clearly, this is inadequate for general random and mixed models.

The macro file `design.mac` contains numerous MacAnova macros, three of which are relevant here: `ems`, `varcomp`, and `mixed`. These macros compute expected mean squares, estimate variance components, and perform general mixed effects ANOVA. The macro ems is the heart of all the approaches. Once the expected mean squares are known, variance components can be estimated and appropriate tests constructed. The following three sections of this note describe the ems, `varcomp`, and `mixed` macros.

2 ems

2.1 Background

Each line of an ANOVA table contains a mean square. This mean square is a random variable, because it depends on the random experimental error, and possibly on random effects that are in the model. The expected value of this random variable, averaging across all the potential values of the random effects and errors, is the expected mean square or EMS.

The EMS for a term depends on how we compute sums of squares, the assumptions we make about our model, and the layout of the data (amount of replication, balance, and so on). The ems macro automatically accounts for replication, layout, and lack of balance. In fact, ems does not even look for balance; it uses the computations necessary for unbalanced data in all situations.

When data are unbalanced, sums of squares can be computed in several ways. Since sums of squares can be computed in different ways, their expectations will also differ. MacAnova has two methods for computing sums of squares. First, and by default, sums of squares can be computed sequentially. This means that the sum of squares for a given term is the reduction in error sum of squares that is achieved when adding the given term to a model that already contains the terms that precede the given term in the model. Thus for a model `"y=A+B+C+A.B"`, we have that A is adjusted for the constant; B is adjusted for the constant and A; C is adjusted for the constant A, and B; and AB is adjusted for the constant, A, B, and C.

A second method for computing sums of squares is used when the phrase `marginal:T` (or just `marg:T`) is used in the `anova()` command. In this case, each term is adjusted for every other term in the model. This is equivalent to determining the increase in error sum of squares that would occur if the given term was removed from the full model. This approach is sometimes called SAS Type III[1].

Model assumptions can also affect expected mean squares. We will consider two assumptions: mixed effects restrictions and hierarchy. In mixed effects models, we may have an interaction between a fixed factor A (with a levels) and a random factor B (with b levels). What should we assume about the interaction

[1] Please note that the marginal sums of squares produced by MacAnova may not equal the SAS Type III sums of squares when there are model degrees of freedom that are linearly dependent on other model degrees of freedom. This could happen, for example, when there are empty cells in a factorial structure. MacAnova determines which model degrees of freedom are linearly dependent on other model degrees of freedom only once, sequentially, with terms entered in the order specified in the model. The adjustment for other terms is then done with the set of columns chosen in the first sequential pass.

effects $\alpha\beta_{ij}$? The restricted model assumes that the mixed interaction effects $\alpha\beta_{ij}$ are normal, with mean zero and variance $\sigma^2_{\alpha\beta}(a-1)/a$. These effects are independent for different values of j, but sum to zero across the levels of the fixed effect. That is, we have the restriction that

$$0 = \sum_{i=1}^{a} \alpha\beta_{ij} \ .$$

The alternative is an unrestricted model. For this model, the mixed interaction effects $\alpha\beta_{ij}$ are independent, normal, with mean zero and variance $\sigma^2_{\alpha\beta}$.

The choice between these models depends on what we believe about interactions. Consider repeating the experiment over and over; this involves random sampling for levels of the random factor B from some population of levels. Suppose that in these repeats of the experiment we choose the same level of B twice; that is, the kth element of the B population occurs in two of these experiments. If you believe that the interaction effects $\alpha\beta_{ij}$ corresponding to this kth element of B will be the same in both repeats of the experiment, then you would use the restricted model. If you believe that the two repeats of the experiment would have different, independent sets of interaction effects, then you would use the unrestricted model.

The second assumption regards hierarchy. The MacAnova `anova()` command enforces hierarchy. That is, for a model

```
Cmd> anova("y = A + B + C + A.B.C")
```

the two factor interactions AB, AC, and BC will be included in the ABC interaction term. The three factor interaction implies the presence of main effects and two factor interactions. If these effects are not already present in the model, they will be included in the three factor interaction term. Some may wish to compute expected mean squares for nonhierarchical models.

2.2 ems Syntax

ems has two mandatory arguments, three optional keyword phrase arguments that alter how the EMS are computed, and two optional keyword phrase arguments that alter return values and what is printed. The basic form is

```
Cmd> ems(Model,Randomvars)
```

This computes the expected mean squares for the terms in the ANOVA for the model given in the CHARACTER scalar `Model`. `Randomvars` is a CHARACTER vector specifying the names of factors in the model that are random. Thus

```
Cmd> ems("y = A + B + A.B + C","C")
```

requests expected mean squares for a three-way factorial model with main effects and AB interaction where factor C is random. Alternatively, `Randomvars` can also be REAL with integer elements giving the index of a factor in the model. In the previous example, `"C"` could be replaced by `3`, since C is the third factor in the model (factor, not term). If both A and C were random, we would use

```
Cmd> ems("y = A + B + A.B + C",vector("A","C"))
```

or

```
Cmd> ems("y = A + B + A.B + C",vector(1,3))
```

If there are no random factors, Randomvars should be NULL. Thus

```
Cmd> ems("y = A + B + A.B + C",NULL)
```

would compute EMS for a fixed effects model.

In this default use, ems computes sequential (Type I) sums of squares for the the restricted (mixed effects add to zero across fixed factors) model, with hierarchy enforced. It prints these expected means squares for each term, and returns no value. Contributions from random terms are shown as multiples of the variance component (for example, `16V(C)`); contributions from fixed terms are shown as a multiple of a quadratic function for the term, for example, `32Q(B)`. In a balanced design, the `Q()` function is the sum of the squared effects divided by degrees of freedom, for example, $\sum_{j=1}^{b}(\beta_j^2)/(b-1)$. In an unbalanced situation, `Q(c)` is a more complicated quantity.

ems works only for factors — no variates are allowed in the model. ems works for both balanced and unbalanced data.

ems assumes that if a factor first appears in an interaction, then that factor is nested in the other terms of the interaction. For example, if the first appearance of factor C is in the term A.B.C, then C is assumed nested in the A.B combinations. This nesting is assumed in the remainder of the model. That is, continuing the example, if there is a later term C.D, it will be interpreted as A.B.C.D even though A.B.C.D is not specifically in the model.

When a term contains the first appearance in the model of more than one factor, ems assumes that the new factors are merged to make a single factor, whose number of levels is the product of the numbers of levels in the factors being merged. For example, if the first appearance of factors B and C with 5 and 3 levels, respectively is in the term A.B.C, then B and C together are considered a single factor with 15 levels. This grouping is assumed in the remainder of the model. That is, continuing the example, if there is a later term C.D, it will be interpreted as B.C.D even though B.C.D is not specifically in the model. This grouped factor is interpreted as random if any of the factors in the group is random.

We may use keyword phrases to alter the computation of expected mean squares. By default, ems computes expectations for mean squares based on sequential sums of squares. You may specify marginal (Type III) computations by including the phrase `marg:T` in the ems command. For example,

```
Cmd> ems("y = A + B + A.B + A.B.C",vector("B","C"),marg:T)
```

By default, ems uses the restricted model for mixed effects. You may specify the unrestricted model by including the phrase `restrict:F` in the ems command. For example,

```
Cmd> ems("y = A + B + A.B","B",restrict:F)
```

Finally, by default, ems enforces hierarchy. You may specify a nonhierarchical model by including the phrase `nonhier:T` in the ems command. For example,

```
Cmd> ems("y = A + B + A.B + A.B.C","C",nonhier:T)
```

In this model, the ABC interaction would not include the degrees of freedom from the AC or BC interactions. (Note, you cannot use `anova()` to compute such an analysis, though it can be done (if you know how) using `swp()`).

These keywords can be used together. For example,

```
Cmd> ems(Model,Randomvars,marg:T,restrict:F)
```

provides answers equivalent to the EMS in SAS PROC GLM.

Using the phrase `keep:T` suppresses printed output but returns a structure (described below) containing the results. If you want the printed output too, use `keep:T,print:T`.

When `keep:T` is an argument to ems , ems returns a structure with five components. These components are

`df` REAL vector of degrees of freedom for all terms in model;

ss REAL vector of sums of squares for all terms in model;

termnames CHARACTER vector of labels for each term;

coefs REAL matrix with coefs[i,j] the coefficient for term j in the EMS of term i;

rterms LOGICAL vector with T indicating that a term is random.

Components ss and df are just those computed from a MacAnova anova() command (possibly with marg:T as needed), and may not be in conformance with the model as used by ems for the following reasons:

1. anova() computes only hierarchical models, while you may specify nonhierarchical models in ems by using nonhier:T.
2. ems enforces nesting and grouping. If b first appears in a.b then b is nested in a and any appearance of b in a later term implies the presence of a. anova() does no such enforcing. For example, in "y=a+a.b+c+b.c", b.c would be interpreted by ems as a.b.c while anova() would not include a.b.c in the model. If b and c first appear together, then "y=b.c+d+c.d" is interpreted in ems as "y=b.c+d+b.c.d".

2.3 ems Examples

The examples below are based on balanced two factor and three factor models with a total of 64 responses. All factors have 2 levels, so two factor and three factor models have 16 and 8 replications, respectively. In some examples, one of the responses is set to MISSING to destroy balance.

Here is a fully nested model; d is nested in c, and e is nested in d. Factor c is fixed, and both d and e are random.

```
Cmd> ems("y=c/d/e",vector("d","e"))
EMS(CONSTANT) = V(ERROR1) + 8V(c.d.e) + 16V(c.d) + 64Q(CONSTANT)
EMS(c) = V(ERROR1) + 8V(c.d.e) + 16V(c.d) + 32Q(c)
EMS(c.d) = V(ERROR1) + 8V(c.d.e) + 16V(c.d)
EMS(c.d.e) = V(ERROR1) + 8V(c.d.e)
EMS(ERROR1) = V(ERROR1)
```

Here is a 3 factor crossed model, with c and d fixed, e random.

```
Cmd> ems("y=c*d*e",3) # e is factor 3
EMS(CONSTANT) = V(ERROR1) + 32V(e) + 64Q(CONSTANT)
EMS(c) = V(ERROR1) + 16V(c.e) + 32Q(c)
EMS(d) = V(ERROR1) + 16V(d.e) + 32Q(d)
EMS(c.d) = V(ERROR1) + 8V(c.d.e) + 16Q(c.d)
EMS(e) = V(ERROR1) + 32V(e)
EMS(c.e) = V(ERROR1) + 16V(c.e)
EMS(d.e) = V(ERROR1) + 16V(d.e)
EMS(c.d.e) = V(ERROR1) + 8V(c.d.e)
EMS(ERROR1) = V(ERROR1)
```

Here is a 2 factor crossed model with unbalanced data. Factor c is fixed and factor d is random. We create a new response with one missing value to illustrate ems for with unbalanced data.

```
Cmd> y1 <- y[1]; y[1] <- ? # make data unbalanced

Cmd> ems("y=c*d",2) # d is factor 2
EMS(CONSTANT) = V(ERROR1) + 0.0080645V(c.d) + 31.508V(d) +
  0.0079365Q(c) + 63Q(CONSTANT)
EMS(c) = V(ERROR1) + 15.746V(c.d) + 0.0081925V(d) + 31.492Q(c)
EMS(d) = V(ERROR1) + 0.0042316V(c.d) + 31.484V(d)
EMS(c.d) = V(ERROR1) + 15.742V(c.d)
EMS(ERROR1) = V(ERROR1)
```

We see that a multiple of the c.d variance component appeared in the EMS for d; this would not occur for balanced data.

In this example, we continue with the unbalanced data, but now we use marg:T. Note how the EMS for main effects differ from the preceding example.

```
Cmd> ems("y=c*d",2,marg:T) # crossed with d random
EMS(CONSTANT) = V(ERROR1) + 31.475V(d) + 62.951Q(CONSTANT)
EMS(c) = V(ERROR1) + 15.742V(c.d) + 31.475Q(c)
EMS(d) = V(ERROR1) + 31.475V(d)
EMS(c.d) = V(ERROR1) + 15.742V(c.d)
EMS(ERROR1) = V(ERROR1)
```

In this example, we illustrate the unrestricted model. Now the c.d variance component appears with a much larger coefficient in the EMS for d.

```
Cmd> ems("y=c*d",2,restrict:F) # crossed with d random
EMS(CONSTANT) = V(ERROR1) + 15.762V(c.d) + 31.508V(d) + 0.0079365Q(c)
   + 63Q(CONSTANT)
EMS(c) = V(ERROR1) + 15.754V(c.d) + 0.0081925V(d) + 31.492Q(c)
EMS(d) = V(ERROR1) + 15.746V(c.d) + 31.484V(d)
EMS(c.d) = V(ERROR1) + 15.738V(c.d)
EMS(ERROR1) = V(ERROR1)
```

Here we use both Type III sums of squares and the unrestricted model. This is equivalent to the method used in SAS PROC GLIM.

```
Cmd> ems("y=c*d",2,marg:T,restrict:F) # same as SAS PROC GLM
EMS(CONSTANT) = V(ERROR1) + 15.738V(c.d) + 31.475V(d) + 62.951Q(CONSTANT)
EMS(c) = V(ERROR1) + 15.738V(c.d) + 31.475Q(c)
EMS(d) = V(ERROR1) + 15.738V(c.d) + 31.475V(d)
EMS(c.d) = V(ERROR1) + 15.738V(c.d)
EMS(ERROR1) = V(ERROR1)
```

For the remaining examples, we restore the first response value so that the data are balanced.

```
Cmd> y[1] <- y1 # restore value for y[1] to regain balance
```

The next two examples illustrate the role of hierarchy. We use a model with c and d crossed, e random, and the c.d.e interaction. By default, ems assumes hierarchy, so the c.d.e term includes the c.e and d.e interactions. In the fully crossed model, the c.e interaction would appear in the EMS of c, and the d.e interaction would appear in the EMS of d, but c.d.e would not appear in either (under the restricted assumptions). However, since we have hierarchy, c.d.e includes c.e and d.e, so c.d.e appears in the EMS of both c and d.

```
Cmd> ems("y=c*d+e+c.d.e",3)
EMS(CONSTANT) = V(ERROR1) + 32V(e) + 64Q(CONSTANT)
EMS(c) = V(ERROR1) + 8V(c.d.e) + 32Q(c)
EMS(d) = V(ERROR1) + 8V(c.d.e) + 32Q(d)
EMS(c.d) = V(ERROR1) + 8V(c.d.e) + 16Q(c.d)
EMS(e) = V(ERROR1) + 32V(e)
EMS(c.d.e) = V(ERROR1) + 8V(c.d.e)
EMS(ERROR1) = V(ERROR1)
```

In contrast to the above example, consider what happens with a nonhierarchical model (still restricted). Here, c.d.e drops out of the EMS of both c and d, since it no longer contains the c.e and d.e terms that would appear in the c and d expected mean squares.

```
Cmd> ems("y=c*d+e+c.d.e",3,nonhier:T)
EMS(CONSTANT) = V(ERROR1) + 32V(e) + 64Q(CONSTANT)
EMS(c) = V(ERROR1) + 32Q(c)
EMS(d) = V(ERROR1) + 32Q(d)
EMS(c.d) = V(ERROR1) + 8V(c.d.e) + 16Q(c.d)
EMS(e) = V(ERROR1) + 32V(e)
EMS(c.d.e) = V(ERROR1) + 8V(c.d.e)
EMS(ERROR1) = V(ERROR1)
```

The final example shows the form of the output structure that can be returned by `ems` . We use a 2^2 design with 16 replicates, with the second factor (d) random. First, the table of expected mean squares.

```
Cmd> ems("y=c*d",2)
EMS(CONSTANT) = V(ERROR1) + 32V(d) + 64Q(CONSTANT)
EMS(c) = V(ERROR1) + 16V(c.d) + 32Q(c)
EMS(d) = V(ERROR1) + 32V(d)
EMS(c.d) = V(ERROR1) + 16V(c.d)
EMS(ERROR1) = V(ERROR1)
```

Here is the structure that is returned. The first three components are the degrees of freedom, sum of squares, and names of the terms in the model. The fourth component (`coefs`), is a matrix with element x_{ij}; x_{ij} is the coefficient of the representing element (either V() or Q()) of term j in the EMS for term i. For example, $x_{2,4}$ is 16, since V(c.d) appears with multiplier 16 in EMS(c); c is term 2, and c.d is term 4. The last component tells which terms are random. It has one fewer element than df or ss, because the last term (an ERROR term) is always random.

```
Cmd> ems("y=c*d",2,keep:T)
component: df
(1)          1          1          1          1         60
component: ss
(1)    0.76155   0.036871    0.31116      1.623     56.318
component: termnames
(1) "CONSTANT"
(2) "c"
(3) "d"
(4) "c.d"
(5) "ERROR1"
component: coefs
(1,1)         64          0         32          0          1
(2,1)          0         32          0         16          1
(3,1)          0          0         32          0          1
(4,1)          0          0          0         16          1
(5,1)          0          0          0          0          1
component: rterms
(1) F       F       T       T
```

2.4 ems Computational Method

ems() uses the "synthesis" method of Hartley (1967), as explained in 10.5.2 of R. R. Hocking (1985). Roughly speaking, we set up basis vectors for each term in the model (call these X), and then we form $X'X$. We now calculate the change in the diagonal of $X'X$ as we sweep out terms (groups of columns). For example, if we sweep columns 2-6 for factor A, and the diagonal terms of factor B in columns 7-10 change, then the total change in the diagonal terms in columns 7-10 is the coefficient for V(B) in the EMS of A.

Clearly, setting up the columns is the major issue. This is reasonably straightforward for purely random and/or purely fixed terms, though some care is needed to get the scaling right. The tedious bits arise for mixed terms when using the restricted model assumptions, and for fixing up nonstandard nesting and/or hierarchy.

3 varcomp

3.1 Background

One common goal when analyzing data with random effects is to estimate the variance components. There are many estimating techniques available, including maximum likelihood, restricted maximum likelihood (reml), MINQUE, and the method of moments. The so-called ANOVA estimates for variance components are method of moments estimators: we set the observed mean squares equal to their expectations and solve for the unknown variance components. The `varcomp` macro implements method of moments (ANOVA) estimates of variance components.

ANOVA estimates of variance components are linear combinations of the observed mean squares that give unbiased estimates of the variance components. Coefficients in these linear combinations may be negative, so the resulting estimates may be negative.

Each individual mean square for a random term is assumed to follow a multiple of a chi square distribution. Thus, the mean square for term i is distributed as $\lambda_i w_i/\nu_i$, where λ_i is the expected mean square for term i, ν_i is the degrees of freedom for term i, and w_i is a chi square random variable with ν_i degrees of freedom. This mean square has expected value λ_i and variance $2\lambda_i^2/\nu_i$.

The estimate for variance component k is a linear combination of mean squares:

$$\begin{aligned}\hat{\sigma}_k^2 &= \sum_{j=1}^{J} z_{kj} MS_j \\ &= \sum_{j=1}^{J} z_{kj}\lambda_j w_j/\nu_j\end{aligned}$$

where J is the total number of terms in the model. Many of the z_{kj} parameters are 0; in particular, those corresponding to fixed effects are zero. The expected value of $\hat{\sigma}_k^2$ is $\sum_j z_{kj}\lambda_j = \sigma_k^2$. The variance of $\hat{\sigma}_k^2$ is

$$V(\hat{\sigma}_k^2) = \sum_{j=1}^{J} 2z_{kj}^2\lambda_j^2/\nu_j \, .$$

Again, we estimate this by replacing λ_k by MS_k obtaining

$$\hat{V}(\hat{\sigma}_k^2) = \sum_{j=1}^{J} 2z_{kj}^2 MS_j^2/\nu_j \, .$$

Important Notes:

- varcomp assumes that the EMS for random terms have no contributions from fixed factors. This happens naturally for balanced data and some special cases, and may be guaranteed in general by using marg:T as an option to ems .

- varcomp uses the SS and DF returned in the output of ems . As discussed in the ems section above, these SS and DF may not match the EMS if the nesting and/or grouping patterns assumed by ems are not followed in the model.

3.2 varcomp Syntax

varcomp may be called with two kinds of arguments. First, varcomp can take a single structure argument that contains the output from an ems(...,keep:T) command. For example,

```
Cmd> emsstuff<-ems("y = A/B/C",vector("B",C"),keep:T)

Cmd> varcomp(emsstuff)
```

The second form is that varcomp can take the arguments that you would ordinarily give to ems and use them directly. For example,

```
Cmd> varcomp("y = A/B/C",vector("B",C"))
```

If you are going to do ems , and varcomp , and mixed , it is generally most efficient to do ems once, using keep:T, print:T to save the results and print them as well. Then the saved results of ems can be used as an argument to varcomp and/or mixed . This is more efficient because the vast majority of the calculation required is in computing the expected mean square information.

The output from varcomp is a matrix with two columns. The first column is the estimate variance component, and the second column is the (estimated) standard error of the variance component estimate. The rows are labeled with the term names.

3.3 varcomp Examples

This example is synthetic data. There are three populations of a species of plant. Four random males are chosen from each population and crossed with four random females from the same population. From each cross, we get six seeds. These six seeds are randomly split into three groups of two and grown in three different environments. Thus we have males and females random and nested separately in population, and then crossed with each other. This whole structure is then crossed with environment.

Here are and analysis of variance and the expected mean squares for this experiment:

```
Cmd> anova("y=(pop+m.pop+f.pop+m.f.pop)*env")
Model used is y=(pop+m.pop+f.pop+m.f.pop)*env
                 DF          SS          MS
CONSTANT          1      5.4299      5.4299
pop               2      2091.4      1045.7
pop.m             9       112.5        12.5
pop.f             9      370.02      41.113
pop.m.f          27      56.774      2.1027
env               2      206.15      103.08
pop.env           4     0.16527    0.041316
pop.m.env        18      3.4185     0.18992
pop.f.env        18      8.2354     0.45752
pop.m.f.env      54      17.117     0.31698
ERROR1          144      30.448     0.21144
```

```
Cmd> emsstuff<-ems("y=(pop+m.pop+f.pop+m.f.pop)*env",vector("m","f"),
   keep:T,print:T)
EMS(CONSTANT) = V(ERROR1) + 6V(pop.m.f) + 24V(pop.f) + 24V(pop.m) +
   288Q(CONSTANT)
EMS(pop) = V(ERROR1) + 6V(pop.m.f) + 24V(pop.f) + 24V(pop.m) + 96Q(pop)
EMS(pop.m) = V(ERROR1) + 6V(pop.m.f) + 24V(pop.m)
EMS(pop.f) = V(ERROR1) + 6V(pop.m.f) + 24V(pop.f)
EMS(pop.m.f) = V(ERROR1) + 6V(pop.m.f)
EMS(env) = V(ERROR1) + 2V(pop.m.f.env) + 8V(pop.f.env) +
   8V(pop.m.env) + 96Q(env)
EMS(pop.env) = V(ERROR1) + 2V(pop.m.f.env) + 8V(pop.f.env) +
   8V(pop.m.env) + 32Q(pop.env)
EMS(pop.m.env) = V(ERROR1) + 2V(pop.m.f.env) + 8V(pop.m.env)
EMS(pop.f.env) = V(ERROR1) + 2V(pop.m.f.env) + 8V(pop.f.env)
EMS(pop.m.f.env) = V(ERROR1) + 2V(pop.m.f.env)
EMS(ERROR1) = V(ERROR1)
```

From the EMS, we see than (MS(pop.m.f.env)-MS(ERROR1))/2 is an unbiased estimate of $\sigma^2_{m.f.env}$; here, we have (.31698 - .21144)/2 = .05277. Similarly, (MS(pop.f.env)-MS(pop.m.f.env))/8 is an unbiased estimate of $\sigma^2_{f.env}$; here, we have (.45752 - .31698)/8 = .01757. varcomp automates these calculations, as well as providing the standard error.

```
Cmd> varcomp(emsstuff)
                Estimate           SE
pop.m            0.43323      0.24668
pop.f             1.6254      0.80788
pop.m.f          0.31521     0.095472
pop.m.env      -0.015883     0.010989
pop.f.env       0.017568     0.020532
pop.m.f.env     0.052766     0.032948
ERROR1           0.21144     0.024919
```

Note that variance component estimates can be negative; varcomp does not truncate the estimates at 0.

To illustrate what can happen with unbalanced data, we make a new response vector with one missing value. The EMS are now much more complicated, as seen below.

```
Cmd> ems("y2=(pop+m.pop+f.pop+m.f.pop)*env",vector("m","f"))
EMS(CONSTANT) = V(ERROR1) + 0.0034722V(pop.m.f.env) +
   0.0025253V(pop.f.env) + 0.0025253V(pop.m.env) + 0.00087108Q(pop.env) +
   0.0017422Q(env) + 5.9792V(pop.m.f) + 23.917V(pop.f) + 23.917V(pop.m) +
   0.0017422Q(pop) + 287Q(CONSTANT)
EMS(pop) = V(ERROR1) + 0.0035088V(pop.m.f.env) + 0.0025518V(pop.f.env) +
   0.0025518V(pop.m.env) + 0.00088025Q(pop.env) + 0.0017605Q(env) +
   5.9792V(pop.m.f) + 23.917V(pop.f) + 23.917V(pop.m) + 95.749Q(pop)
EMS(pop.m) = V(ERROR1) + 0.0036486V(pop.m.f.env) + 0.0026535V(pop.f.env) +
   0.0026535V(pop.m.env) + 0.00091533Q(pop.env) + 0.0018307Q(env) +
   5.9792V(pop.m.f) + 0.0028257V(pop.f) + 23.917V(pop.m)
EMS(pop.f) = V(ERROR1) + 0.0038946V(pop.m.f.env) + 0.0028324V(pop.f.env) +
   0.0028324V(pop.m.env) + 0.00097704Q(pop.env) + 0.0019541Q(env) +
   5.9792V(pop.m.f) + 23.914V(pop.f)
EMS(pop.m.f) = V(ERROR1) + 0.0044788V(pop.m.f.env) + 0.0032573V(pop.f.env) +
   0.0032573V(pop.m.env) + 0.0011236Q(pop.env) + 0.0022472Q(env) +
   5.9792V(pop.m.f)
```

```
EMS(env) = V(ERROR1) + 1.9918V(pop.m.f.env) + 7.9667V(pop.f.env) +
   7.9667V(pop.m.env) + 0.0012605Q(pop.env) + 95.7Q(env)
EMS(pop.env) = V(ERROR1) + 1.9918V(pop.m.f.env) + 7.9666V(pop.f.env) +
   7.9666V(pop.m.env) + 31.924Q(pop.env)
EMS(pop.m.env) = V(ERROR1) + 1.9916V(pop.m.f.env) + 0.0030327V(pop.f.env) +
   7.966V(pop.m.env)
EMS(pop.f.env) = V(ERROR1) + 1.9914V(pop.m.f.env) + 7.963V(pop.f.env)
EMS(pop.m.f.env) = V(ERROR1) + 1.9905V(pop.m.f.env)
EMS(ERROR1) = V(ERROR1)
```

In particular, the EMS for male, female, and male.female all include quadratic effects of environment. The variance components estimates produced by `varcomp` assume that the EMS for random effects have no contributions from fixed effects. Thus this set of expected mean squares cannot be used appropriately by `varcomp`. If so used, `varcomp` prints a warning.

```
Cmd> varcomp("y2=(pop+m.pop+f.pop+m.f.pop)*env",vector("m","f"))
WARNING: fixed effects contribute to some random terms
                Estimate          SE
pop.m            0.44839     0.25251
pop.f             1.6396     0.81325
pop.m.f          0.30377    0.092456
pop.m.env      -0.016848    0.010589
pop.f.env       0.016768    0.020114
pop.m.f.env     0.049917    0.032691
ERROR1           0.21253    0.025134
```

To fix this, compute the marginal (Type III) expected mean squares.

```
Cmd> emsstuff2<-ems("y2=(pop+m.pop+f.pop+m.f.pop)*env",
   vector("m","f"),keep:T,print:T,marg:T)
EMS(CONSTANT) = V(ERROR1) + 5.9627V(pop.m.f) + 23.838V(pop.f) +
   23.838V(pop.m) + 286.01Q(CONSTANT)
EMS(pop) = V(ERROR1) + 5.9628V(pop.m.f) + 23.838V(pop.f) +
   23.838V(pop.m) + 95.507Q(pop)
EMS(pop.m) = V(ERROR1) + 5.9635V(pop.m.f) + 23.842V(pop.m)
EMS(pop.f) = V(ERROR1) + 5.9635V(pop.m.f) + 23.842V(pop.f)
EMS(pop.m.f) = V(ERROR1) + 5.9652V(pop.m.f)
EMS(env) = V(ERROR1) + 1.9896V(pop.m.f.env) + 7.9474V(pop.f.env) +
   7.9474V(pop.m.env) + 95.507Q(env)
EMS(pop.env) = V(ERROR1) + 1.9897V(pop.m.f.env) + 7.9477V(pop.f.env) +
   7.9477V(pop.m.env) + 31.878Q(pop.env)
EMS(pop.m.env) = V(ERROR1) + 1.99V(pop.m.f.env) + 7.9499V(pop.m.env)
EMS(pop.f.env) = V(ERROR1) + 1.99V(pop.m.f.env) + 7.9499V(pop.f.env)
EMS(pop.m.f.env) = V(ERROR1) + 1.9905V(pop.m.f.env)
EMS(ERROR1) = V(ERROR1)
```

Here no random term has an EMS including a quadratic in fixed effects. Thus we could use

```
Cmd> varcomp(emsstuff2)
                Estimate          SE
pop.m            0.43597     0.24764
pop.f             1.6377     0.81335
pop.m.f          0.31197    0.094699
pop.m.env      -0.016186    0.010769
pop.f.env       0.017346    0.020313
pop.m.f.env     0.049917    0.032691
ERROR1           0.21253    0.025134
```

since these estimates of variance components are free of quadratic terms in the fixed effects.

3.4 `varcomp` Computational Method

`varcomp` uses the output of ems to find linear combinations of observed mean squares that are unbiased estimates of variance components. The `coefs` component of the output of ems shows the coefficients for the contribution of every term to every expected mean square. The rows and columns of this matrix corresponding to random terms are extracted; call this matrix A. Observed mean squares for random terms are extracted; call this vector y. `varcomp` then computes $x = A^{-1}y$; x is then vector of estimated variance components. The standard errors for estimated variance components are computed as explained in the Background section.

4 `mixed`

4.1 Background

A standard part of the analysis of a mixed effects model is to test the null hypotheses that the various terms in the model are zero (either $H_0 : \sigma_\alpha^2 = 0$ for a random effect or $H_0 : \beta_j \equiv 0$ for a fixed effect). This is accomplished by computing an F-ratio from two mean squares whose EMS's differ only by a multiple of the null hypothesis term of interest.

For example, consider the first ems illustration of the last Section. Suppose that we wish to test the null hypothesis that the environment by male (nested in population) interaction variance component is zero. Inspection of the table of EMS reveals the following two lines:

```
EMS(pop.f.env) = V(ERROR1) + 2V(pop.m.f.env) + 8V(pop.f.env)
EMS(pop.m.f.env) = V(ERROR1) + 2V(pop.m.f.env)
```

These lines differ only by a term in V(pop.f.env), so MS(pop.f.env)/MS(pop.m.f.env) is an appropriate F test.

For unbalanced data, and in some cases for balanced data, there can be hypotheses to be tested for which there are no two terms in the ANOVA, the EMS's of which differ only by the component of interest. In this case, we must construct a denominator that has the same expectation as the numerator when the null hypothesis is true. This denominator is some linear combination of other mean squares in the ANOVA table. One complication here is that this new denominator will not be distributed as a multiple of a chisquare. We therefore use an approximate degrees of freedom obtained via the Satterthwaite approximation (see, for example, Kuehl 1994).

Again from the first ems illustration of the last Section, consider testing the effect of population. The table of EMS includes the following:

```
EMS(pop) = V(ERROR1) + 6V(pop.m.f) + 24V(pop.f) + 24V(pop.m) + 96Q(pop)
EMS(pop.m) = V(ERROR1) + 6V(pop.m.f) + 24V(pop.m)
EMS(pop.f) = V(ERROR1) + 6V(pop.m.f) + 24V(pop.f)
EMS(pop.m.f) = V(ERROR1) + 6V(pop.m.f)
```

Furthermore, we find that no term in the ANOVA has EMS equal to V(ERROR1) + 6V(pop.m.f) + 24V(pop.f) + 24V(pop.m). However, EMS(pop.m)+EMS(pop.f)-EMS(pop.m.f) does equal V(ERROR1) + 6V(pop.m.f) + 24V(pop.f) + 24V(pop.m), so we can use this combination of three MS as a denominator for MS(pop). Note that MS(pop.m)+MS(pop.f)-MS(pop.m.f) could be negative. Thus the Satterthwaite approximation as a multiple of a chisquare could be quite inappropriate.

An alternative that avoids the possibility of negative denominators also modifies the numerator. In the preceding case, we add MS(pop.m.f) to the numerator instead of subtracting it from the denominator.

More generally, anywhere we would subtract some multiple of an MS from the denominator we instead add the corresponding (positive) multiple of the MS to the numerator. This makes the F-ratio smaller, but the approximate degrees of freedom in the denominator tend to be bigger, and the Satterthwaite approximation tends to be more accurate.

4.2 `mixed` Syntax

Syntax for `mixed` is similar to that of `varcomp` . `mixed` may be called with two kinds of arguments. First, `mixed` can take a single structure argument that contains the output from an ems`(...,keep:T)` command. For example,

```
Cmd> emsstuff<-ems("y = A/B/C",vector("B",C"),keep:T)

Cmd> mixed(emsstuff)
```

The second form is that `mixed` can take the arguments that you would ordinarily give to `mixed` and use them directly. For example,

```
Cmd> mixed("y = A/B/C",vector("B",C"))
```

If you are going to do ems , and `varcomp` , and `mixed` , it is generally most efficient to do ems once, using `keep:T, print:T` to save the results and print them as well. Then the saved results of ems can be used as an argument to `varcomp` and/or `mixed` . This is more efficient because the vast majority of the calculation required is in computing the expected mean square information.

`mixed` has two optional keyword arguments. By default, `mixed` adds appropriate multiples of an MS to the numerator to be tested rather than subtracting a multiple of an MS in the denominator. If the phrase `useneg:T` is included as an argument, then `mixed` will instead subtract the multiple of the MS from the denominator.

By default, `mixed` prints out a table giving each line of the ANOVA with its (approximate) df and MS, the error MS and (approximate) error df, the F statistic and its p value. In this case, `mixed` returns a NULL value. If the keyword phrase `keepmixed:T` is used, the table will not be printed, but will instead be returned as the value of `mixed` . (The table is returned as a labeled matrix.)

4.3 `mixed` Examples

We illustrate `mixed` with the balanced plant breeding data used in the `varcomp` examples. For the full data set (note, emsstuff is the output from ems for the full data set),

```
Cmd> mixed(emsstuff)
                    DF          MS    Error DF    Error MS           F     P value
CONSTANT         1.914       7.533       14.01       53.61      0.1405      0.8617
cf               2.008        1048       14.01       53.61       19.54   8.745e-05
cf.m                 9        12.5          27       2.103       5.945   0.0001412
cf.f                 9       41.11          27       2.103       19.55   1.242e-09
cf.m.f              27       2.103         144      0.2114       9.945           0
env              2.012       103.4       30.75      0.6474       159.7           0
cf.env           56.12      0.3583       30.75      0.6474      0.5534      0.9729
cf.m.env            18      0.1899          54       0.317      0.5991      0.8844
cf.f.env            18      0.4575          54       0.317       1.443      0.1496
cf.m.f.env          54       0.317         144      0.2114       1.499     0.03044
ERROR1             144      0.2114           0           0     MISSING     MISSING
```

The test used here for environment is (MS(env)+MS(m.f.env))/(MS(f.env)+MS(m.env)) which equals 159.7. Both the numerator and the denominator have approximate degrees of freedom, in this case, 2.01 and 30.75.

If we wished to use the negative coefficients in the denominator, we get

```
Cmd> newmixed(emsstuff,useneg:T)
                       DF          MS    Error DF    Error MS          F    P value
CONSTANT                1        5.43       12.92       51.51     0.1054     0.7506
cf                      2        1046       12.92       51.51       20.3  0.0001028
cf.m                    9        12.5          27       2.103      5.945  0.0001412
cf.f                    9       41.11          27       2.103      19.55  1.242e-09
cf.m.f                 27       2.103         144      0.2114      9.945          0
env                     2       103.1       7.048      0.3305      311.9  1.322e-07
cf.env                  4     0.04132       7.048      0.3305      0.125     0.9688
cf.m.env               18      0.1899          54       0.317     0.5991     0.8844
cf.f.env               18      0.4575          54       0.317      1.443     0.1496
cf.m.f.env             54       0.317         144      0.2114      1.499    0.03044
ERROR1                144      0.2114           0           0    MISSING    MISSING
```

Now the test for environment is MS(env)/(MS(f.env)+MS(m.env)-MS(m.f.env)) which equals 311.9. Only the denominator has an approximate degrees of freedom, 7.048 (numerator has 2 degrees of freedom).

mixed computes its denominators on the assumption that no fixed term makes a contribution to the EMS of a random term. mixed will print a warning if any fixed term appears in the EMS for a random term.

4.4 mixed Computational Method

mixed first uses the coefs component of the output of ems to find the linear combination of variance components that should be used as denominator for a given term. This is the row of coefs corresponding to the term, modified by setting the entries for any fixed terms and the term itself to zero. Call this modifed row y'. mixed then uses the rows and columns of coefs corresponding to random terms (call this A) to find a linear combination of mean squares that has as its expectation the desired linear combination of variance components. This linear combination is $A^{-1}y$. When useneg is false (the default), any mean squares with negative coefficients for the denominator are instead added to the numerator. Approximate degrees of freedom for both numerator and denominator are found using Satterthwaite's approximation.

5 References

Hartley, H. O. (1967) "Expectation, Variances and Covariances of ANOVA Mean Squares by 'Synthesis'," *Biometrics* 23, 105-114.

Hocking, R. R. (1985) *The Analysis of Linear Models*, Brooks/Cole: Monterey, CA.

Kuehl, R. O. (1994) *Statistical Principles of Research Design and Analysis*, Duxbury: Belmont,CA.