Changes to the Mixed Effects Models chapters in ELM

Julian Faraway

September 29, 2008

1 Introduction

The book "Extending the Linear Model with R" (ELM) [5] first appeared in 2005 and was based on R version 2.2.0. R is updated regularly and so it is natural that some incompatibilities with the current version have been introduced. For most of the chapters, these changes have been minor and have been addressed in the errata and/or subsequent reprintings of the text. However, for chapter 8 and 9, the changes have been much more substantial. These chapters are based on the `lme4` package [2]. The package author, Doug Bates of the University of Wisconsin has made some significant changes to this software, most particularly in the way that inference is handled for mixed models. Fitting mixed effects models is a complex subject because of the large range of possible models and because the statistical theory still needs some development. `lme4` is perhaps the best software generally available for fitting such models, but given the state of the field, there will be scope for significant improvements for some time. It is important to understand the reason behind these changes.

For standard linear models (such as those considered in "Linear Models with R" [4]), the recommended way to compare an alternative hypothesis of a larger model compared to a null hypothesis of a smaller model nested within this larger model, is to use an F-test. Under the standard assumptions and when the errors are normally distributed, the F-statistic has an exact F-distribution with degrees of freedom that can be readily computed given the sample size and the number of parameters used by each model.

For linear mixed effects models, that is models having some random effects, we might also wish to test fixed effect terms using an F-test. One way of approaching this is to assume that the estimates of the parameters characterizing the random effects of the model are in fact the true values. This reduces the mixed models to fixed effect models where the error has a particular covariance structure. Such models can be fit using generalized least squares and F-tests can be conducted using standard linear models theory. Several statistics software packages take this approach including the `nlme` package developed earlier and still available within R. Earlier versions of `lme4` also took this approach and hence the output seen in the current version of ELM.

However, there are two serious problems with this test. Firstly, the random effects are not actually known, but estimated. This means that the F-statistic does not follow an F-distribution exactly. In some cases, it may be a good approximation, but not in general.

Secondly, even if one were to assume that the ***F***-distribution was a sufficiently good approximation, there remains the problem of degrees of freedom. The concept of "degrees of freedom", as used in statistics, is not as well defined as many people believe. Perhaps one might think of it as the effective number of independent observations on which an estimate or test is based. Often, this is just the sample size minus the number of free parameters. However, this notion becomes more difficult when considering the dependent and hierarchical data found in mixed effects models. There is no simple way in which the degrees of freedom can be counted. The degrees of freedom are used here just to select the null distribution for a test-statistic i.e. they are used as a mathematical convenience rather than as a concept of standalone value. As such, the main concern is whether they produce the correct null distribution for the test statistic. In the case of F-statistics for mixed models, there has been substantial research on this — see [7] and related — but there is no simple and general solution. Even if there were, this would still not avoid the problem of the dubious approximation to an ***F***-distribution.

The ***t***-statistics presented in the `lme4` model summary outputs are based on the square roots of ***F***-statistics and so the same issues with testing still arise. In some cases, one may appeal to asymptotics to allow for simple normal and chi-squared approximations to be used. But it is not simply a matter of sample size — the number of random effects parameters and the model structure all make a difference to the quality of the approximation. There is no simple rule to say when the approximation would be satisfactory.

All this poses a problem for the writers of statistical software for such models. One approach is to simply provide the approximate solution even though it is known to be poor in some cases. Or one can take the approach that no answer (at least for now) is better than a possibly poor answer, which is the approach currently taken in `lme4`. In some simpler models, specialized solutions are possible. For example, in [8], ***F***-tests for a range of simple balanced (i.e. equal numbers of observations per group) designs are provided. For some simple but unbalanced datasets, some progress has been made — see [3]. However, such straightforward solutions are not available for anything more complex. Such partial specialized solutions are not satisfactory for a package as general in scope as `lme4` — we need a solution that works reasonable well in all cases.

We do have some viable alternatives. The parametric bootstrap approach based on the likelihood ratio statistic is discussed in ELM. We can add to this methods based on Markov chain Monte Carlo (MCMC). For some simple balanced models, solutions are available using the `aov()` command.

We present here the changes to the text for chapters 8 and 9. The intent here is to list the changes and suggests replacements to achieve much the same result. In section 4, we discuss MCMC methods that might provided a major alternative to the likelihood-based testing presented in the book. In section 5, we present the simpler and partial `aov` based solution. Note that `lme4` is under active development and further changes to this document are likely to be necessary. In particular, one can expect that there will be more convenient ways to access components of the model than current exist.

To help keep this document up-to-date, it is written using Sweave which inserts the output from R commands directly into the text. However, this does mean all the output which is sometimes more than the edited version that appears in the book. In particular, I have modified the printing of some summaries of `lmer` fits to prevent the printing of correlation matrices.

Let's start by loading the packages we need and verifying the version of R and `lme4`:

```
> library(faraway)
> library(lme4)
> sessionInfo("lme4")

R version 2.7.2 (2008-08-25)
i386-apple-darwin8.11.1

locale:
C

attached base packages:
character(0)

other attached packages:
[1] lme4_0.999375-24

loaded via a namespace (and not attached):
 [1] Matrix_0.999375-11 base_2.7.2         faraway_1.0.3      grDevices_2.7.2
 [5] graphics_2.7.2     grid_2.7.2         lattice_0.17-13    methods_2.7.2
 [9] stats_2.7.2        tools_2.7.2        utils_2.7.2
```

We also set the seed on the random number generator to ensure exact repeatability for the bootstraps and other simulations:

```
> set.seed(123)
```

2 Revisions to Chapter 8

8.1 Estimation

The first call to `lmer` occurs on p157 where the output now becomes:

```
> mmod <- lmer(bright ~ 1 + (1 | operator), pulp)
> summary(mmod)

Linear mixed model fit by REML
Formula: bright ~ 1 + (1 | operator)
   Data: pulp
```

```
  AIC  BIC logLik deviance REMLdev
 24.6 27.6  -9.31     16.6    18.6
Random effects:
 Groups   Name        Variance Std.Dev.
 operator (Intercept) 0.0681   0.261
 Residual             0.1063   0.326
Number of obs: 20, groups: operator, 4

Fixed effects:
            Estimate Std. Error t value
(Intercept)   60.400      0.149     404
```

There are two changes to note. The AIC and BIC have changed due to changes in the computation of the number of parameters. In the fixed effects part of the output, there is no longer a degrees of freedom and a *p*-value. In this case, we do not miss the test because the *t*-value is so large and the intercept so obviously different from zero. The following maximum likelihood based version of the output is also changed:

```
> smod <- lmer(bright ~ 1 + (1 | operator), pulp, REML = FALSE)
> summary(smod)
```

```
Linear mixed model fit by maximum likelihood
Formula: bright ~ 1 + (1 | operator)
   Data: pulp
  AIC  BIC logLik deviance REMLdev
 22.5 25.5  -8.26     16.5    18.7
Random effects:
 Groups   Name        Variance Std.Dev.
 operator (Intercept) 0.0457   0.214
 Residual             0.1063   0.326
Number of obs: 20, groups: operator, 4

Fixed effects:
            Estimate Std. Error t value
(Intercept)   60.400      0.129     467
```

In addition to the the changes to the AIC, BIC, df and *p*-value, there are also smaller numerical changes to the random effects estimates due to improvements in the fitting algorithm. Also the way to specify that maximum likelihood estimates are required has changed from `method="ML"` to `REML=FALSE`.

8.2 Inference

Nested hypotheses can still be tested using the likelihood ratio statistic. The chi-squared approximation can be quite innaccurate, giving *p*-values that tend to be too small. The parametric bootstrap requires much more computation, but gives better results.

The current text also proposed the use of $F-$ or $t-$ statistics, but as explained above, these are no longer provided in the current version of `lme4`. It would be possible to fit most of the models in this chapter using the older `nlme` package, which has a somewhat different syntax, and thereby obtain these F-statistic. Alternatively, it is possible, as we shall demonstrate, to reconstruct these tests from the output. However, one should realize that these may give poor results and we do not recommend doing this in general.

8.3 Blocks as Random Effects

The output of the model on p164 becomes:

```
> op <- options(contrasts = c("contr.sum", "contr.poly"))
> mmod <- lmer(yield ~ treat + (1 | blend), penicillin)
> print(summary(mmod), correlation = FALSE)

Linear mixed model fit by REML
Formula: yield ~ treat + (1 | blend)
   Data: penicillin
 AIC BIC logLik deviance REMLdev
 119 125  -53.3      117     107
Random effects:
 Groups   Name        Variance Std.Dev.
 blend    (Intercept) 11.8     3.43
 Residual             18.8     4.34
Number of obs: 20, groups: blend, 5

Fixed effects:
            Estimate Std. Error t value
(Intercept)    86.00       1.82    47.3
treat1         -2.00       1.68    -1.2
treat2         -1.00       1.68    -0.6
treat3          3.00       1.68     1.8

> options(op)
```

Again we note the changes in the AIC, BIC and the lack of p-values for the t-statistics. If one still wanted to perform a t-test, we could use the normal approximation on the t-statistics. Since the three treatment statistics here are well below 2 in absolute value, we might conclude that these treatment effects are not significant. However, providing a more precise p-value is problematic and for t-statistics around 2 or so, some better method of testing would be needed.

The ANOVA table at the top of p165 becomes:

```
> anova(mmod)
```

```
Analysis of Variance Table
      Df Sum Sq Mean Sq F value
treat  3   70.0    23.3    1.24
```

We no longer have an *F*-statistic or *p*-value so we can no longer perform the test in this way. The LRT-based test that follows remains unchanged except that the AIC and BIC values are different (not that it matters here).

8.5 Split plots

On p168, the first model fails with an error. The `try` command is useful when you suspect a command may cause an error and you do not want to interrupt the execution of a batch of commands.

```
> tt <- try(lmod <- lmer(yield ~ irrigation * variety + (1 | field) +
+     (1 | field:variety), data = irrigation))
> cat(tt)

Error : length(levels(dm$flist[[1]])) < length(Y) is not TRUE
```

The reason, as explained in the text, is that this model is over-parameterised. Except now the model fails to fit at all, which is actually helpful since it alerts us to the problem. The second (correct) model does fit:

```
> lmodr <- lmer(yield ~ irrigation * variety + (1 | field), data = irrigation)
> logLik(lmodr)

'log Lik.' -22.697 (df=10)
```

Note the change in the degrees of freedom. The subsequent model summary is:

```
> print(summary(lmodr), correlation = FALSE)

Linear mixed model fit by REML
Formula: yield ~ irrigation * variety + (1 | field)
   Data: irrigation
  AIC  BIC logLik deviance REMLdev
 65.4 73.1  -22.7     68.6    45.4
Random effects:
 Groups   Name        Variance Std.Dev.
 field    (Intercept) 16.20    4.02
 Residual              2.11    1.45
Number of obs: 16, groups: field, 8

Fixed effects:
                       Estimate Std. Error t value
```

```
(Intercept)              38.50       3.03   12.73
irrigationi2              1.20       4.28    0.28
irrigationi3              0.70       4.28    0.16
irrigationi4              3.50       4.28    0.82
varietyv2                 0.60       1.45    0.41
irrigationi2:varietyv2   -0.40       2.05   -0.19
irrigationi3:varietyv2   -0.20       2.05   -0.10
irrigationi4:varietyv2    1.20       2.05    0.58
```

As before, the AIC and BIC are changed while the *p*-values are gone. Note that the *t*-statistics for the fixed effects are all small and give us a good indication that there are no significant fixed effects here. The subsequent ANOVA table is:

```
> anova(lmodr)

Analysis of Variance Table
                   Df Sum Sq Mean Sq F value
irrigation          3  2.455   0.818    0.39
variety             1  2.250   2.250    1.07
irrigation:variety  3  1.550   0.517    0.25
```

Again, no F-statistics or *p*-values. For this dataset, the small *t*-values are sufficient to conclude that there are no significant fixed effects. This can be confirmed by computing the LRT and estimating its *p*-value via the parametric bootstrap.

8.6 Nested Effects

The first model output on p171 becomes:

```
> cmod <- lmer(Fat ~ 1 + (1 | Lab) + (1 | Lab:Technician) + (1 |
+     Lab:Technician:Sample), data = eggs)
> summary(cmod)

Linear mixed model fit by REML
Formula: Fat ~ 1 + (1 | Lab) + (1 | Lab:Technician) + (1 | Lab:Technician:Sample)
   Data: eggs
   AIC   BIC logLik deviance REMLdev
 -54.2 -44.9   32.1    -68.7   -64.2
Random effects:
 Groups                Name        Variance Std.Dev.
 Lab:Technician:Sample (Intercept) 0.00306  0.0554
 Lab:Technician        (Intercept) 0.00698  0.0835
 Lab                   (Intercept) 0.00592  0.0769
 Residual                          0.00720  0.0848
Number of obs: 48, groups: Lab:Technician:Sample, 24; Lab:Technician, 12; Lab, 6
```

```
Fixed effects:
            Estimate Std. Error t value
(Intercept)    0.388      0.043    9.02
```

Again the same changes as seen before. The output of the random effects at the top of p172 becomes:

```
> cmodr <- lmer(Fat ~ 1 + (1 | Lab) + (1 | Lab:Technician), data = eggs)
> VarCorr(cmodr)
```

```
$`Lab:Technician`
            (Intercept)
(Intercept)   0.0080018
attr(,"stddev")
(Intercept)
   0.089453
attr(,"correlation")
            (Intercept)
(Intercept)           1

$Lab
            (Intercept)
(Intercept)   0.0059191
attr(,"stddev")
(Intercept)
   0.076936
attr(,"correlation")
            (Intercept)
(Intercept)           1

attr(,"sc")
sigmaREML
  0.09612
```

which is effectively the same information as in the text, but displayed in a less pleasant way.

8.7 Crossed Effects

On p173, the ANOVA becomes:

```
> mmod <- lmer(wear ~ material + (1 | run) + (1 | position), abrasion)
> anova(mmod)
```

```
Analysis of Variance Table
         Df Sum Sq Mean Sq F value
material  3   4622    1541    25.1
```

which is again without the F-statistic and *p*-values. The model output is:

```
> print(summary(mmod), correlation = FALSE)
```

```
Linear mixed model fit by REML
Formula: wear ~ material + (1 | run) + (1 | position)
   Data: abrasion
 AIC BIC logLik deviance REMLdev
 114 120  -50.1      120     100
Random effects:
 Groups   Name        Variance Std.Dev.
 run      (Intercept)  66.9     8.18
 position (Intercept) 107.1    10.35
 Residual              61.3     7.83
Number of obs: 16, groups: run, 4; position, 4

Fixed effects:
            Estimate Std. Error t value
(Intercept)   265.75       7.67    34.7
materialB     -45.75       5.53    -8.3
materialC     -24.00       5.53    -4.3
materialD     -35.25       5.53    -6.4
```

Again, the *p*-values are gone. However, note that the large size of the t-statistics means that we can be confident that there are significant material effects here. This could verified with an LRT with parametric bootstrap to estimate the *p*-value but is hardly necessary given the already convincing level of evidence.

8.8 Multilevel Models

The linear models analysis remains unchanged. The first difference occurs at the top of p177 where the ANOVA table becomes:

```
> jspr <- jsp[jsp$year == 2, ]
> mmod <- lmer(math ~ raven * social * gender + (1 | school) +
+     (1 | school:class), data = jspr)
> anova(mmod)
```

```
Analysis of Variance Table
                    Df Sum Sq Mean Sq F value
raven                1  10218   10218  374.40
social               8    616      77    2.82
gender               1     22      22    0.79
raven:social         8    577      72    2.64
raven:gender         1      2       2    0.09
social:gender        8    275      34    1.26
raven:social:gender  8    187      23    0.86
```

We no longer have F-statistics and their associated degrees of freedom and p-values. Note that we can reconstruct the ANOVA table by finding the residual standard error from the model:

```
> attributes(VarCorr(mmod))$sc

sigmaREML
   5.2241
```

and then recomputing the $F-$statistics:

```
> (fstat <- anova(mmod)[, 3]/5.2241^2)

[1] 374.405559   2.820309   0.791794   2.644944   0.090217   1.260889   0.857225
```

The p-values are then:

```
> round(pf(fstat, anova(mmod)[, 1], 917, lower.tail = FALSE), 4)

[1] 0.0000 0.0043 0.3738 0.0072 0.7640 0.2605 0.5524
```

The degrees of freedom for the denominator of 917 can be obtained by summing the degrees of freedom from the ANOVA table and subtracting an extra one for the intercept:

```
> nrow(jspr) - sum(anova(mmod)[, 1]) - 1

[1] 917
```

Now, as pointed out in Section 1, there is good reason to question the results of such F-tests. In this case, the nominal degrees of freedom is large. Given that the number of random effects is not particularly large, the "true" degrees of freedom will still be large. This suggests that these particular p-values will be fairly accurate.

Another possibility is to compute LRTs. For example, we can test the three-way interaction term by fitting the model with and without this term and computing the test:

```
> mmod <- lmer(math ~ (raven * social * gender)^2 + (1 | school) +
+     (1 | school:class), data = jspr, REML = FALSE)
> mmod2 <- lmer(math ~ (raven + social + gender)^2 + (1 | school) +
+     (1 | school:class), data = jspr, REML = FALSE)
> anova(mmod, mmod2)

Data: jspr
Models:
mmod2: math ~ (raven + social + gender)^2 + (1 | school) + (1 | school:class)
mmod: math ~ (raven * social * gender)^2 + (1 | school) + (1 | school:class)
      Df   AIC   BIC logLik Chisq Chi Df Pr(>Chisq)
mmod2 31  5958  6108  -2948
mmod  39  5967  6156  -2944   7.1      8       0.53
```

We notice that the p-value of 0.53 is quite similar to the 0.55 produced by the F-test. For larger datasets where the residual standard error is estimated fairly precisely, the denominator of the F-statistic has little variability so that the test statistic becomes close to chi-squared distributed, just like the LRT. They will not be numerically identical, but we might expect them to be close.

Implementing the parametric bootstrap to estimate the p-value is possible here:

```
> nrep <- 1000
> lrstat <- numeric(nrep)
> for (i in 1:nrep) {
+     rmath <- unlist(simulate(mmod2))
+     rmmod <- lmer(rmath ~ (raven * social * gender)^2 + (1 |
+         school) + (1 | school:class), data = jspr, REML = FALSE)
+     rmmod2 <- lmer(rmath ~ (raven + social + gender)^2 + (1 |
+         school) + (1 | school:class), data = jspr, REML = FALSE)
+     lrstat[i] <- 2 * (logLik(rmmod) - logLik(rmmod2))
+ }
> 2 * (logLik(mmod) - logLik(mmod2))
```

```
'log Lik.' 7.0954 (df=39)
```

```
> mean(lrstat > 7.0954)
```

```
[1] 0.543
```

Unsurprisingly, given the sample size, the results are very similar to that obtained by the chi-squared approximation.

If you need to consider more than just two models and wish to select a model, it is typically better to use a criterion-based variable selection methods. Here we can use the AIC to select the model. The computation of the AIC does require the specification of the number of parameters, which could be problematic if we also consider the random effects parameters. However, if we only consider models where the fixed effect are different, the issue does not arise when comparing such models. We fit a sequence of such models here. (There is some ugliness in the extraction of the AIC — this is likely to change in future versions of `lme4`).

```
> mmod <- lmer(math ~ raven * social * gender + (1 | school) +
+     (1 | school:class), data = jspr)
> attributes(summary(mmod))$AICtab["AIC"]
```

```
  AIC
 5970
```

```
> mmod <- lmer(math ~ raven * social + (1 | school) + (1 | school:class),
+     data = jspr)
> attributes(summary(mmod))$AICtab["AIC"]
```

```
   AIC
 5963.2
```

```
> mmod <- lmer(math ~ raven + social + (1 | school) + (1 | school:class),
+     data = jspr)
> attributes(summary(mmod))$AICtab["AIC"]
```

```
   AIC
 5949.7
```

```
> mmod <- lmer(math ~ raven + (1 | school) + (1 | school:class),
+     data = jspr)
> attributes(summary(mmod))$AICtab["AIC"]
```

```
   AIC
 5965.9
```

```
> mmod <- lmer(math ~ social + (1 | school) + (1 | school:class),
+     data = jspr)
> attributes(summary(mmod))$AICtab["AIC"]
```

```
   AIC
 6226.6
```

We see that the main effects model that uses `raven` and `social` gives the lowest AIC. We can examine this fit using the same centering of the Raven score as used in the book:

```
> jspr$craven <- jspr$raven - mean(jspr$raven)
> mmod <- lmer(math ~ craven + social + (1 | school) + (1 | school:class),
+     jspr)
> print(summary(mmod), correlation = FALSE)
```

```
Linear mixed model fit by REML
Formula: math ~ craven + social + (1 | school) + (1 | school:class)
   Data: jspr
  AIC  BIC logLik deviance REMLdev
 5950 6013  -2962     5928    5924
Random effects:
 Groups       Name        Variance Std.Dev.
 school:class (Intercept)  1.03    1.02
 school       (Intercept)  3.23    1.80
 Residual                 27.57    5.25
Number of obs: 953, groups: school:class, 90; school, 48
```

```
Fixed effects:
            Estimate Std. Error t value
(Intercept)  32.0107     1.0350   30.93
craven        0.5841     0.0321   18.21
social2      -0.3611     1.0948   -0.33
social3      -0.7767     1.1649   -0.67
social4      -2.1196     1.0396   -2.04
social5      -1.3632     1.1585   -1.18
social6      -2.3703     1.2330   -1.92
social7      -3.0482     1.2703   -2.40
social8      -3.5472     1.7027   -2.08
social9      -0.8863     1.1031   -0.80
```

Now that there are only main effects without interactions, the interpretation is simpler but essentially similar to that seen in the book. We do not have *p*-values in the table of coefficients. However, the sample size is large here and so a normal approximation could be used to compute reasonable *p*-values:

```
> tval <- attributes(summary(mmod))$coef[, 3]
> pval <- 2 * pnorm(abs(tval), lower = FALSE)
> cbind(attributes(summary(mmod))$coef, "p value" = round(pval,
+     4))
```

```
            Estimate Std. Error  t value p value
(Intercept) 32.01073   1.034987 30.92862  0.0000
craven       0.58412   0.032085 18.20531  0.0000
social2     -0.36106   1.094769 -0.32980  0.7415
social3     -0.77672   1.164888 -0.66677  0.5049
social4     -2.11964   1.039635 -2.03883  0.0415
social5     -1.36317   1.158500 -1.17667  0.2393
social6     -2.37031   1.233021 -1.92236  0.0546
social7     -3.04824   1.270275 -2.39967  0.0164
social8     -3.54723   1.702737 -2.08325  0.0372
social9     -0.88633   1.103141 -0.80346  0.4217
```

Of course, there are the usual concerns with multiple comparisons for the nine-level factor, `social`. The reference level is social class I and we can see significant differences between this level and levels IV, VII and VIII.

When testing the compositional effects, we need to make two changes. Firstly, we have decided not to have the interaction between Raven score and social class, consistent with the analysis above. Secondly, we cannot use the *F*-test to make the comparison. We replace this with an LRT:

```
> schraven <- lm(raven ~ school, jspr)$fit
> mmod <- lmer(math ~ craven + social + (1 | school) + (1 | school:class),
+     jspr, REML = FALSE)
> mmodc <- lmer(math ~ craven + social + schraven + (1 | school) +
+     (1 | school:class), jspr, REML = FALSE)
> anova(mmod, mmodc)

Data: jspr
Models:
mmod: math ~ craven + social + (1 | school) + (1 | school:class)
mmodc: math ~ craven + social + schraven + (1 | school) + (1 | school:class)
      Df   AIC   BIC logLik Chisq Chi Df Pr(>Chisq)
mmod  13  5954  6018  -2964
mmodc 14  5956  6024  -2964  0.18      1       0.67
```

As before, we do not find any compositional effects.

3 Revisions to Chapter 9

9.1 Longitudinal Data

On p186, the function lmList() in lme4, can be used for the computation of linear models on groups within the data simpler. For now, the computation in the book is simpler.

On p189, the output of the model becomes:

```
> psid$cyear <- psid$year - 78
> mmod <- lmer(log(income) ~ cyear * sex + age + educ + (cyear |
+     person), psid)
> print(summary(mmod), correlation = FALSE)

Linear mixed model fit by REML
Formula: log(income) ~ cyear * sex + age + educ + (cyear | person)
   Data: psid
  AIC  BIC logLik deviance REMLdev
 3840 3894  -1910     3786    3820
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 person   (Intercept) 0.2817   0.531
          cyear       0.0024   0.049    0.187
 Residual             0.4673   0.684
Number of obs: 1661, groups: person, 85

Fixed effects:
            Estimate Std. Error t value
(Intercept)   6.6742     0.5433   12.28
```

```
cyear         0.0853     0.0090    9.48
sexM          1.1503     0.1213    9.48
age           0.0109     0.0135    0.81
educ          0.1042     0.0214    4.86
cyear:sexM   -0.0263     0.0122   -2.15
```

Some changes in the AIC and BIC as well as the omission of *p*-values are noted. Given the sample size, the normal approximation for the computation of *p*-values for the t-statistics would be acceptable.

9.2 Repeated Measures

The model output on p193 becomes:

```
> mmod <- lmer(acuity ~ power + (1 | subject) + (1 | subject:eye),
+     vision)
> print(summary(mmod), correlation = FALSE)
```

```
Linear mixed model fit by REML
Formula: acuity ~ power + (1 | subject) + (1 | subject:eye)
   Data: vision
 AIC BIC logLik deviance REMLdev
 343 357   -164      339     329
Random effects:
 Groups      Name        Variance Std.Dev.
 subject:eye (Intercept) 10.3     3.21
 subject     (Intercept) 21.5     4.64
 Residual                16.6     4.07
Number of obs: 56, groups: subject:eye, 14; subject, 7

Fixed effects:
            Estimate Std. Error t value
(Intercept)  112.643      2.235    50.4
power6/18      0.786      1.540     0.5
power6/36     -1.000      1.540    -0.6
power6/60      3.286      1.540     2.1
```

Some changes in the AIC and BIC as well as the omission of *p*-values are noted. The ANOVA table becomes:

```
> anova(mmod)
```

```
Analysis of Variance Table
      Df Sum Sq Mean Sq F value
power  3  140.8    46.9    2.83
```

We would like to know whether the power is statistically significant but no longer have the p-value from F-statistic available. We can use the LRT and parametric bootstrap as follows:

```
> mmod <- lmer(acuity ~ power + (1 | subject) + (1 | subject:eye),
+     vision, REML = FALSE)
> nmod <- lmer(acuity ~ 1 + (1 | subject) + (1 | subject:eye),
+     vision, REML = FALSE)
> as.numeric(2 * (logLik(mmod) - logLik(nmod)))

[1] 8.2624

> pchisq(8.2625, 3, lower = FALSE)

[1] 0.040887

> lrstat <- numeric(1000)
> for (i in 1:1000) {
+     racuity <- unlist(simulate(nmod))
+     rnull <- lmer(racuity ~ 1 + (1 | subject) + (1 | subject:eye),
+         vision, REML = FALSE)
+     ralt <- lmer(racuity ~ power + (1 | subject) + (1 | subject:eye),
+         vision, REML = FALSE)
+     lrstat[i] <- as.numeric(2 * (logLik(ralt) - logLik(rnull)))
+ }
> (pval <- mean(lrstat > 8.2625))

[1] 0.057
```

Using the chi-squared approximation gives a p-value of 0.041 while the parametric bootstrap gives 0.057. These are close to the p-value of 0.048 from the F-statistic. Thus the result is borderline.

We can repeat the calculations for when the 43rd observation is omitted:

```
> mmodr <- lmer(acuity ~ power + (1 | subject) + (1 | subject:eye),
+     vision, subset = -43)
> anova(mmodr)

Analysis of Variance Table
      Df Sum Sq Mean Sq F value
power  3   89.2    29.7     3.6
```

Again we lack the p-values we had before. However, we do have have sufficient sample size to conclude that a t-statistic of 3 is sufficient to indicate the significance of the highest power relative to the baseline. The same would be true for the calculations based on the Helmert contrasts.

8.3 Multiple Response Multilevel Models

The ANOVA table at the top of page 197 can be reconstructed as follows:

```
> jspr <- jsp[jsp$year == 2, ]
> mjspr <- data.frame(rbind(jspr[, 1:6], jspr[, 1:6]), subject = factor(rep(c("english
+     "math"), c(953, 953))), score = c(jspr$english/100, jspr$math/40))
> mjspr$craven <- mjspr$raven - mean(mjspr$raven)
> mmod <- lmer(score ~ subject * gender + craven * subject + social +
+     (1 | school) + (1 | school:class) + (1 | school:class:id),
+     mjspr)
> sigmaerr <- attributes(VarCorr(mmod))$sc
> (fstat <- anova(mmod)[, 3]/sigmaerr^2)
```

```
[1] 3953.6598    7.4644  444.6302    6.4690   28.2253   15.9866
```

```
> nrow(mjspr) - sum(anova(mmod)[, 1]) - 1
```

```
[1] 1892
```

```
> (pvals <- pf(fstat, anova(mmod)[, 1], 1892, lower.tail = FALSE))
```

```
[1] 0.0000e+00 6.3515e-03 8.0171e-89 2.4689e-08 1.2072e-07 6.6244e-05
```

```
> cbind(anova(mmod), fstat, pvalue = round(pvals, 3))
```

```
               Df   Sum Sq   Mean Sq   F value     fstat pvalue
subject         1 53.73683 53.736832 3953.6598 3953.6598  0.000
gender          1  0.10145  0.101454    7.4644    7.4644  0.006
craven          1  6.04327  6.043266  444.6302  444.6302  0.000
social          8  0.70339  0.087924    6.4690    6.4690  0.000
subject:gender  1  0.38363  0.383630   28.2253   28.2253  0.000
subject:craven  1  0.21728  0.217284   15.9866   15.9866  0.000
```

In this case, there are a large number of degrees of freedom for the error and the approximation will be good here, just as in the analysis of this data in the previous chapter. In any case, the interaction terms are clearly significant. The subsequent model summary will lack *p*-values but these are not necessary for our interpretation. If we wanted them, a normal approximation would suffice.

4 Inference via MCMC

An alternative way of conducting inference is via Bayesian methods implemented via Markov chain Monte Carlo (MCMC). A general introduction to these methods may be found in texts such as [6]. The idea is to assign a non-informative prior on the parameters of the mixed model and then generate a sample from their posterior distribution. We use MCMC methods starting from the REML estimates to generate this sample. More details and other examples of data analysis with MCMC from `lme4` can be found in [1].

To illustrate these methods, consider the `penicillin` data analyzed in Chapter 8. We fit the model:

```
> mmod <- lmer(yield ~ treat + (1 | blend), penicillin)
```

We can generate 10,000 MCMC samples as in:

```
> pens <- mcmcsamp(mmod, n = 10000, saveb = TRUE)
```

The `saveb=TRUE` option asks that the random effects also be saved. We can examine the fixed effect samples and summarize them:

```
> ff <- as.data.frame(pens)
> head(ff)

  (Intercept)    treatB  treatC treatD     ST1  sigma
1      84.000 1.0000000 5.00000 2.0000 0.79127 4.3397
2      82.569 1.4235024 4.92370 3.1100 0.90557 3.5470
3      83.874 4.1356980 4.74026 4.6649 0.96193 3.6846
4      83.557 0.0093179 3.53468 1.2031 0.75773 3.9401
5      83.973 2.8346079 5.01643 3.5657 0.48614 3.2749
6      88.652 1.5378192 0.90691 2.8383 0.56695 5.9062

> colMeans(ff)

(Intercept)      treatB      treatC      treatD         ST1       sigma
   83.96355     1.09684     5.06267     2.03114     0.48562     5.10608
```

Notice that the first MCMC sample corresponds to the REML estimates. The parameter `ST1` is the ratio $\sigma_b/\sigma_\varepsilon$. The posterior means for the fixed effects of the MCMC samples are similar to the REML values. The posterior mean for σ_ε is somewhat larger than the REML estimate. We may compute the posterior mean for σ_b as:

```
> mean(ff[, 5] * ff[, 6])

[1] 2.3168
```

The standard deviations for the posterior distributions are:

```
> sd(ff)

(Intercept)      treatB      treatC      treatD         ST1       sigma
    2.70538     3.35078     3.33229     3.34932     0.38841     1.10593
```

where we see somewhat larger values than for the standard errors of the REML estimates. In light of the larger posterior mean for σ_ε, this is expected. We can compute highest posterior density (HPD) intervals: (95% by default)

```
> HPDinterval(pens)
```

```
$fixef
               lower   upper
(Intercept) 78.6888 89.4739
treatB      -6.0399  7.2130
treatC      -1.7340 11.5006
treatD      -4.6778  8.6397
attr(,"Probability")
[1] 0.95

$ST
     lower  upper
[1,]     0 1.1812
attr(,"Probability")
[1] 0.95

$sigma
      lower  upper
[1,] 3.1465 7.2202
attr(,"Probability")
[1] 0.95

$ranef
       lower  upper
[1,] -1.0333 7.9959
[2,] -6.1178 2.3489
[3,] -4.7802 3.3968
[4,] -2.8446 5.2853
[5,] -6.7036 1.8314
attr(,"Probability")
[1] 0.95
```

These are constructed as the shortest interval to contain the specified probability within the posterior distribution. This is not quite the same as taking empirical quantiles which would produce wider intervals, particularly for asymmetric distributions.

Conducting hypothesis tests using this information is problematic, not least because Bayesian methods are not sympathetic to such ideas. If you really must conduct tests and compute p-values, there are some possibilities. Firstly, you can easily check whether the point of the null hypothesis falls with the 95% interval. This treats the HPD intervals like confidence intervals although the underlying theory is rather different. For the three fixed treatment effects seen in this model, all three intervals contain zero and so this null hypotheses would not be rejected. To figure p-values, we would need to find the intervals that intersect with zero. For the treatment contrast C-A interval this would be $[0, 10]$. The fraction of samples that lie outside this interval is:

```
> mean((ff[, 3] < 0) | (ff[, 3] > 10))
```

```
[1] 0.1222
```

This would be the estimated *p*-value. Computing a *p*-value for the treatment effect as a whole is more difficult. One possible way of doing this is to construct an elliptical confidence region around the estimates that intersects the origin. The orientation of the ellipse would be determined by the covariance of the MCMC samples. The proportion of samples lying outside the ellipse could be used to estimate the *p*-value. If we assume multivariate normality for the joint posterior density (which seems OK here), then this *p*-value can be estimated by computing the Mahalanobis distance of the origin to the center of the distribution and then using the chi-squared as the reference distribution:

```
> covarm <- cov(ff[, 2:4])
> meanm <- colMeans(ff[, 2:4])
> (md <- mahalanobis(c(0, 0, 0), meanm, covarm))
```

```
[1] 2.6032
```

```
> pchisq(md, 3, lower = FALSE)
```

```
[1] 0.45692
```

However, all this is rather speculative, contrary to the spirit of Bayesian analysis and lacks solid theoretical backing. It is better to simply study the the posterior distributions with respect to the practical questions of interest concerning the particular dataset.

Turning to the blocking variation, we might question whether this is significant or not, especially since in the completely fixed effects analysis, this factor was borderline. We might consider the proportion of MCMC samples in which the blocking variation was less than 1% of the residual variation. We can calculate this as:

```
> mean(ff[, 5] < 0.01)
```

```
[1] 0.1437
```

Thus it is quite plausible that the blocking variation is rather small. This is emphatically not a *p*-value but it does address the practical question regarding the existence of a signficant blocking variation.

The MCMC approach has some advantages and disadvantages relative to the LRT with parametric bootstrap testing method. One major advantage is that it is much faster. With the parametric bootstrap, the model is refit with each sample, which is time consuming. The disadvantages are that one has to be careful about the stability and convergence of the Markov chain. This can be checked using plots of the chain such as:

We have plotted just the last 500 values of the chain to get a closer look. The diagnostics in this case are quite encouraging. The samples appear to vary randomly around some mean

```
> timeseq <- 1:10000
> plot(treatC ~ timeseq, ff, subset = (timeseq > 9500), type = "l")
```

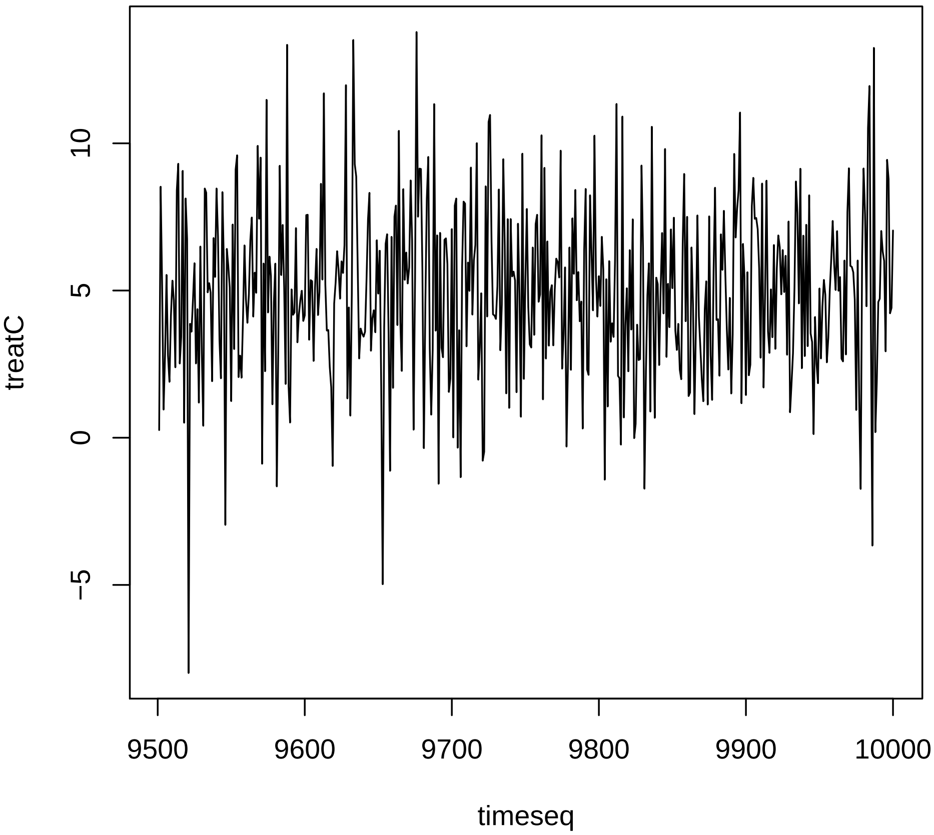

Figure 1: MCMC samples from the penicillin model for the C-A contrast

```
> plot(ST1 ~ timeseq, ff, subset = (timeseq > 9500), type = "l")
```

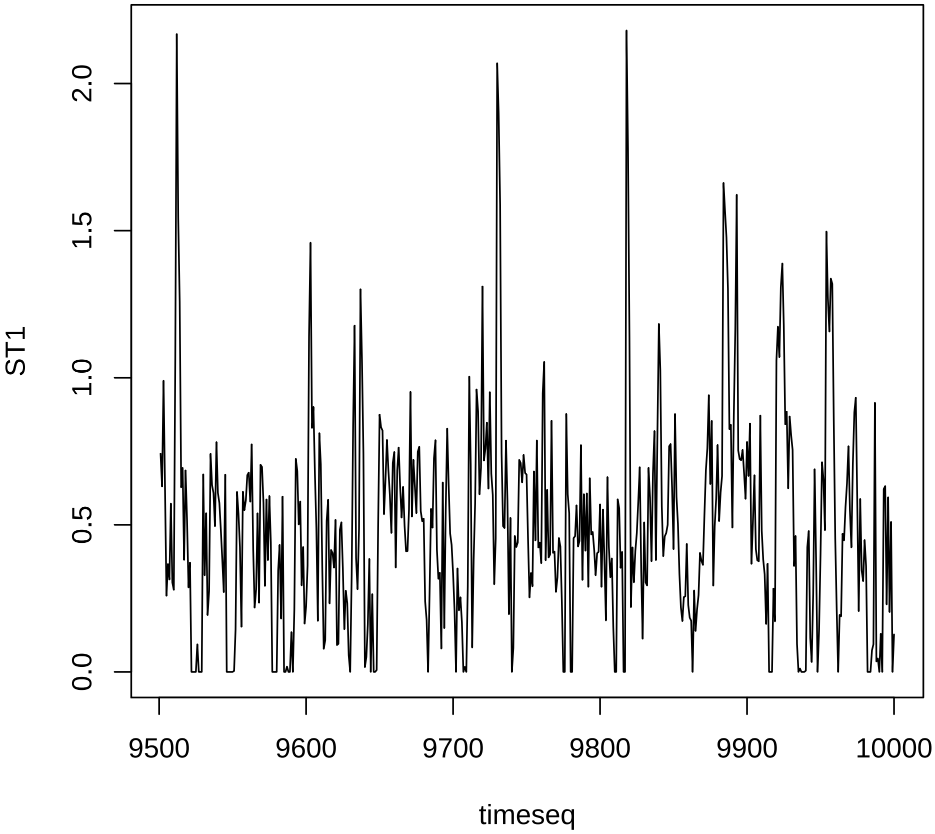

Figure 2: MCMC samples from the penicillin model for the SD ratio

value without getting stuck in any particular region which is a sign of difficulties with the mixing of the Markov chain.

We show the corresponding plot for $\sigma_b/\sigma_\varepsilon$ in Figure 2. In this case, greater correlation is evident and we note that the chain stays near zero for several steps at a time on several occasions, which leads to some concern about the mixing of the Markov chain. This has some consequences for our understanding of the block effect since the posterior distribution puts non-negligible weight around zero.

Judging the effectiveness of the MCMC method for any given problem can be difficult and goes beyond the scope of this article. The BUGS software, that can be accessed from R, allows much more control — see [9]. However, this very much a problem for the less sophisticated user since if the diagnostics for the MCMC reveal some problem, it requires some additional expertise to know how to proceed.

5 Inference with aov

The `aov()` function can be used to fit simple models with a single random effects component. The results are reliable only for balanced data. We can illustrate this with the `penicillin` data:

```
> lmod <- aov(yield ~ treat + Error(blend), penicillin)
> summary(lmod)

Error: blend
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals  4    264      66

Error: Within
          Df Sum Sq Mean Sq F value Pr(>F)
treat      3   70.0    23.3    1.24   0.34
Residuals 12  226.0    18.8
```

We see that the test of the significance for the fixed effects which is effectively the same as the original F-test presented in ELM. Note that the p-values are provided only for the fixed effects terms. The fixed effect coefficients may be obtained as

```
> coef(lmod)

(Intercept) :
(Intercept)
         86

blend :
numeric(0)

Within :
treatB treatC treatD
     1      5      2
```

The `irrigation` data can also be fit using `aov`:

```
> lmod <- aov(yield ~ irrigation * variety + Error(field), irrigation)
> summary(lmod)

Error: field
           Df Sum Sq Mean Sq F value Pr(>F)
irrigation  3   40.2    13.4    0.39   0.77
Residuals   4  138.0    34.5
```

```
Error: Within
                   Df Sum Sq Mean Sq F value Pr(>F)
variety             1   2.25    2.25    1.07   0.36
irrigation:variety  3   1.55    0.52    0.25   0.86
Residuals           4   8.43    2.11
```

The analysis takes account of the fact that the irrigation does not vary within the field. Note that the F-statistics are the same as the ANOVA table obtained originally from `lmer`.

References

[1] D.J. Baayen, R.H.and Davidson and D.M. Bates. Mixed-effects modeling with crossed random effects for subjects and items. unpublished, 2007.

[2] Douglas Bates. Fitting linear mixed models in R. *R News*, 5(1):27–30, May 2005.

[3] C. Crainiceanu and D. Ruppert. Likelihood ratio tests in linear mixed models with one variance component. *Journal of the Royal Statistical Society, Series B*, 66:165–185, 2004.

[4] J. Faraway. *Linear Models with R.* Chapman and Hall, London, 2005.

[5] J. Faraway. *Extending the Linear Model with R.* Chapman and Hall, London, 2006.

[6] A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian Data Analysis.* Chapman and Hall, London, 2 edition, 2004.

[7] MG Kenward and JH Roger. Small sample inference for fixed effects from restricted maximum likelihood. *Biometrics*, 53:983–997, Sep 1997.

[8] H. Scheffé. *The Analysis of Variance.* Wiley, New York, 1959.

[9] Andrew Thomas. The BUGS language. *R News*, 6(1):17–21, March 2006.