

10

Multi-level Models and Repeated Measures

Use of `lme()` (*nlme*) instead of `lmer()` (*lme4*)

Here is demonstrated the use of `lme()`, from the *nlme* package, to handle the calculations in Chapter 10 of the 2nd edition of “Data Analysis & Graphics Using R” (Cambridge Univ Press, January 2007). The code is introduced with a minimum of comment.

10.1 A One-Way Random Effects Model

10.1.1 Analysis with `aov()`

10.1.2 A More Formal Approach

10.1.3 Analysis using `lme`:

The modeling command takes the form:

```
library(nlme)
ant111b.lme <- lme(fixed=harvwt ~ 1, random = ~1|site,
                  data=ant111b)
```

The only fixed effect is the overall mean. The parameter setting `random = ~1|site` fits random variation between sites. Variation between the individual units that are nested within sites, i.e., between parcels, are by default treated as random. Here is the default output:

```
> options(digits=4)
> ant111b.lme
Linear mixed-effects model fit by REML
  Data: ant111b
  Log-restricted-likelihood: -47.21
  Fixed: harvwt ~ 1
  (Intercept)
           4.292

Random effects:
  Formula: ~1 | site
           (Intercept) Residual
StdDev:      1.539      0.76
```

Number of Observations: 32
 Number of Groups: 8

Notice that `lme()` gives, not the components of variance, but the standard deviations (StdDev) which are their square roots. Observe that, according to `lme()`, $\widehat{\sigma}_W^2 = 0.76^2 = 0.578$, and $\widehat{\sigma}_L^2 = 1.539^2 = 2.369$.

Notice also that `lme()` does not give the mean square at any level higher than level 0, not even in this balanced case.

Fitted values and residuals in lme()

The level 1 fitted values from `lme()` are known as BLUPs (Best Linear Unbiased Predictors). Among linear unbiased predictors of the site means, the BLUPs are designed to have the smallest expected error mean square.

Relative to the site means, the BLUPs are pulled in toward the overall mean. The most extreme site means will on average, because of random variation, be more extreme than the corresponding “true” means for those sites. For the simple model considered here, the i th fitted value is given by

$$\hat{y}_i = \hat{y}_i + \frac{n\hat{\sigma}_L^2}{n\hat{\sigma}_L^2 + \hat{\sigma}_W^2}(\hat{y} - \hat{y}_i).$$

Using the variance component estimates, we can obtain the BLUPs for the corn yield data at each site as follows:

```
> s2W <- 0.578; s2L <- 2.37; n <- 4
> sitemeans <- with(ant111b, sapply(split(harvwt, site), mean))
> grandmean <- mean(sitemeans)
> shrinkage <- (n*s2L)/(n*s2L+s2W)
> yhat <- grandmean + shrinkage*(sitemeans - grandmean)
> yhat
  DBAN  LFAN  NSAN  ORAN  OVAN  TEAN  WEAN  WLAN
4.851 4.212 2.217 6.764 4.801 3.108 5.455 2.925
> ##
> ## More directly, use predict() with the lme object
> unique(predict(ant111b.lme))
[1] 4.851 4.212 2.217 6.764 4.801 3.108 5.455 2.925
> ##
> ## Compare with site means
> sitemeans
  DBAN  LFAN  NSAN  ORAN  OVAN  TEAN  WEAN  WLAN
4.885 4.207 2.090 6.915 4.832 3.036 5.526 2.841
```

Note that, by default, the fitted values adjust for all random effects except the residual, i.e., they are calculated at level 1. The level 0 fitted effect, i.e., not adjusting for any of the random effects, is the overall mean.

Residuals can be also defined on several levels. At level 0, they are the differences between the observed responses and the overall mean. At level 1, they are the differences

between the observed responses and the fitted values at level 1 (which are the BLUPs for the sites).

**Uncertainty in the variance components*

It is straightforward to get 95% confidence intervals for the model parameters, using the function `intervals()`:

```
> intervals(ant111b.lme)
Approximate 95% confidence intervals

Fixed effects:
              lower est. upper
(Intercept)  3.14  4.29  5.45
attr(,"label")
[1] "Fixed effects:"

Random Effects:
Level: site
              lower est. upper
sd((Intercept)) 0.882 1.54  2.68

Within-group standard error:
lower est. upper
0.573 0.760 1.008
```

The only fixed effect is the overall mean yield. The standard error for the overall mean, allowing the calculation of a confidence interval, was discussed above.

Details for the site random effect then follow, though for the standard deviation rather than the variance. Based on these 95% confidence intervals, σ_L^2 could be anywhere between $0.882^2 \simeq 0.78$ and $2.68^2 \simeq 7.2$.

Similarly, based on the 95% confidence interval for σ_W , the component of variance σ_W^2 could be anywhere between $0.573^2 \simeq 0.33$ and $1.008^2 \simeq 1.02$.

Handling more than two levels of random variation

The `lme()` function can handle variation at several levels. Suppose, for example, that house prices (`price`) were available at samples of 3 bedroom bungaloes within samples of suburbs (`suburb`) located within a number of different American cities (`city`). We now have three levels of variation: level 3 is house, level 2 is suburb, and level 1 is city. Prices differ between cities, and between suburbs within cities, and between houses within suburbs.

Since level 1 and 2 variation must be reflected in the `lme()` function call, we would analyze such data using

```
## house.lme <- lme(price ~ 1, random= ~1|city/suburb)
```

10.2 Survey Data, with Clustering

10.2.1 Alternative models

We specify sex (`sex`) and school type (`PrivPub`) as fixed effects, while school (`school`) and class (`class`) are specified as random effects. Class is *nested* within school; within each school there are several classes. The model is

$$y = \begin{array}{c} \text{sex effect} \\ \text{(fixed)} \end{array} + \begin{array}{c} \text{type (private or public)} \\ \text{(fixed)} \end{array} + \begin{array}{c} \text{school effect} \\ \text{(random)} \end{array} + \begin{array}{c} \text{class effect} \\ \text{(random)} \end{array} + \begin{array}{c} \text{pupil effect} \\ \text{(random)} \end{array}.$$

The table of estimates and standard errors for the coefficients of the fixed component is similar to that from an `lm()` (single level) analysis, but with an extra column that is headed `DF`.

```
library(nlme)
> science.lme <- lme(fixed = like ~ sex + PrivPub,
+                   data = science, random = ~ 1 | school/class,
+                   na.action=na.omit)
> summary(science.lme)$tTable # Print coefficients.
```

	Value	Std.Error	DF	t-value	p-value
(Intercept)	4.722	0.1625	1316	29.07	6.79e-144
sexm	0.182	0.0982	1316	1.86	6.35e-02
PrivPubpublic	0.412	0.1857	39	2.22	3.25e-02

Note that we have 1316 degrees of freedom for the comparison between males and females, but only 39 degrees of freedom for the comparison between private and public schools (there are 12 private schools, and 29 public schools. The degrees of freedom are calculated as $12 - 1 + 29 - 1 = 39$). The comparison is between different schools of the different types. On the other hand, schools are made up of classes, each of which includes both males and females. Thus the between pupils level of variation is relevant for the comparison between sexes.

There are three variance components:¹

```
Between schools 0.00105
Between classes 0.318
Between students 3.05
```

Note that the standard deviations that `VarCorr(science.lme)` gives for the variance components use an approximation that can be quite inaccurate. It is important to note that the between classes variance component is additional to the variation between students component in the residual row, i.e. that labeled `Between students`.

This table is interesting in itself. It tells us that differences between classes are greater than would be expected from differences between students alone. It also suggests we can do the following simpler analysis that ignores the effect of schools:

¹`VarCorr(science.lme)` # Displayed output differs slightly
Also, try `intervals(science.lme)`, which gives SDs that are
the square roots of the variance components.

```
> science1.lme <- lme(fixed = like ~ sex + PrivPub, data = science,
+                    random = ~ 1 | Class, na.action=na.exclude)
> summary(science1.lme)$tTable # Table of coefficients
```

	Value	Std.Error	DF	t-value	p-value
(Intercept)	4.722	0.1624	1316	29.07	6.20e-144
sexm	0.182	0.0982	1316	1.86	6.36e-02
PrivPubpublic	0.412	0.1857	64	2.22	3.02e-02

The variance components are, to two significant digits, the same as before. On this occasion, the `intervals()` function will be used – this has the advantage of giving approximate 95% confidence intervals.²

	lower	est.	upper
Variance: Class	0.190	0.321	0.54
Variance: Within	2.828	3.052	3.29

We can plot standardized residuals from the class means against fitted values, thus:

```
plot(science1.lme, resid(.) ~ fitted(.) | PrivPub)
# Invokes plot.lme(), which gives a lattice-like plot
# By default, fitted values and residuals are at level 1
```

The residuals have a banded pattern that makes them hard to interpret. There is, however, no obvious trend about the line $y = 0$.

10.2.2 Instructive, though faulty, analyses

Ignoring class as the random effect

We fit `school`, ignoring `class`, as a random effect. The estimates of the fixed effects change little.

```
> science2.lme <- lme(fixed = like ~ sex + PrivPub, data = science,
+                    random = ~ 1 | school, na.action=na.exclude)
> summary(science2.lme)$tTable
```

	Value	Std.Error	DF	t-value	p-value
(Intercept)	4.738	0.163	1341	29.00	5.95e-144
sexm	0.197	0.101	1341	1.96	5.07e-02
PrivPubpublic	0.417	0.185	39	2.25	3.02e-02

This analysis suggests, wrongly, that the between schools component of variance is substantial. The estimated variance components are³

```
2## The numerical values were extracted thus:
compvars <- rbind(intervals(science1.lme, which="var-cov")[[1]]$Class,
                  intervals(science1.lme, which="var-cov")[[2]])^2
# NB: intervals() gives SDs, which were squared to give variances.
rownames(compvars) <- c("Variance: Class", "Variance: Within")
compvars
3## The numerical values were extracted from
VarCorr(science2.lme)
## Alternatively, they can be extracted from
intervals(science2.lme)
```

```
Between schools 0.166
Between students 3.219
```

This is misleading. From our earlier investigation, it is clear that the difference is between classes, not between schools!

10.2.3 Predictive accuracy

10.3 A Multi-level Experimental Design

10.3.1 The anova table

10.3.2 Expected values of mean squares

10.3.3* The sums of squares breakdown

10.3.4 The variance components

10.3.5 The mixed model analysis

For a mixed model analysis, we specify that treatment (`shade`) is a fixed effect, that `block` and `plot` are random effects, and that `plot` is nested in `block`. The software works out for itself that the remaining part of the variation is associated with differences between vines. It is not necessary to specify it separately.

For using `lme()`, the command is

```
kiwishade.lme <- lme(fixed = yield ~ shade, random = ~ 1 |
                    block/plot, data=kiwishade)
# block/shade is an alternative to block/plot
```

For comparison purposes with the results from the preceding section, consider:

```
> VarCorr(kiwishade.lme)
              Variance  StdDev
block =      pdLogChol(1)
(Intercept)  4.04          2.01
plot =       pdLogChol(1)
(Intercept)  2.19          1.48
Residual    12.18          3.49
```

Notice that the estimate for the block component of variance differs slightly from the estimate that was obtained above.

Plots of residuals

Recall that by default, fitted values adjust for all except random variation between individual vines, i.e., they account for treatment, block and plot effects. For this, set `level=2` when calculating fitted values, or the equivalent residuals. Other choices are to calculate fitted values as treatment plus block (`level=1`) or as treatment effects only (`level=0`). The block *effects* are differences between fitted values at level 1 and fitted values at level 0, while the plot effects are differences between fitted values at level 2 and fitted values at level 1.

The following code gives a figure (A) that shows residuals after fitting the block and treatment effects, and takes residuals from those.

```
library(lattice)
xyplot(residuals(kiwishade.lme) ~ fitted(kiwishade.lme)|block,
       groups=shade, pch=1:4, layout=c(3,1), data=kiwishade,
       xlab="Level 2 fitted values (Treatment + block + plot effects)",
       ylab="Residuals (level 2)", aspect=1,
       key=list(
         points=list(pch=1:4, col=trellis.par.get("superpose.symbol")$col[1:4]),
         text=list(levels(kiwishade$shade)), space="top", columns=4))
# By default, residuals and fitted values are calculated at
# the highest available of the random factors, here level 2
```

In the following figure (B), level 1 fitted values are subtracted from level 2 fitted values and plotted against the treatment means.

```
## Simplified version of graph that shows the plot effects
ploteffects <- fitted(kiwishade.lme) - fitted(kiwishade.lme, level=1)
xyplot(ploteffects ~ fitted(kiwishade.lme, level = 1)|block, layout=c(3,1),
       data=kiwishade, groups=shade, pch=1:4, aspect=1,
       key=list(
         points=list(pch=1:4, col=trellis.par.get("superpose.symbol")$col[1:4]),
         text=list(levels(kiwishade$shade)), space="top", columns=4))
```

It will help in interpreting Figure B to remember that the treatment means are, in order,

Dec2Feb	Feb2May	none	Aug2Dec
89.92	92.77	100.20	103.23

Notice that the plot-specific effects go in opposite directions, relative to the overall treatment means, in the east and north blocks.

10.3.6 Predictive accuracy

10.3.7 Different sources of variance – complication or focus of interest?

10.4 Within and Between Subject Effects

10.4.1 Model selection

For conceptual clarity, and in order to keep inference as simple as possible, we limit initial attention to three models:

1. All possible interactions (this is likely to be more complex than we need):

```
## Change initial letters of levels of tinting$agegp to upper case
levels(tinting$agegp) <- toupper.initial(levels(tinting$agegp))
## Fit all interactions: data frame tinting (DAAG)
itstar.lme <- lme(log(it) ~ tint*target*agegp*sex,
                 random= ~ 1 | id, data=tinting, method="ML")
```

2. All two-factor interactions (this is a reasonable guess; two-factor interactions may be all we need):

```
it2.lme <- lme(log(it) ~ (tint+target+agegp+sex)^2,
              random = ~1|id, data=tinting,
              method="ML")
```

3. Main effects only (this is a very simple model):

```
it1.lme <- lme(log(it) ~ (tint+target+agegp+sex),
              random = ~1|id, data=tinting,
              method= "ML")
```

The reason for specifying `method="ML"`, i.e. use the maximum likelihood estimation criterion, is that we can then do the equivalent of an analysis of variance comparison:

```
anova(itstar.lme, it2.lme, it1.lme)
```

Here is the outcome:

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
itstar.lme	1	26	8.15	91.5	21.93			
it2.lme	2	17	-3.74	50.7	18.87	1 vs 2	6.11	0.7288
it1.lme	3	8	1.14	26.8	7.43	2 vs 3	22.88	0.0065

Notice that `df` is now used for degrees of freedom, where `Df` was used in connection with `summary.aov()` earlier. Such inconsistencies should, in time, be removed.

The p -value for comparing model 1 with model 2 is 0.73, while that for comparing model 2 with model 3 is 0.0065. This suggests that the model that limits attention to two-factor interactions is adequate. (Note also that the AIC statistic favors model 2. The BIC statistic, which is an alternative to AIC, favors the model that has main effects only. The BIC's penalty for model complexity can be unduly severe when the number of residual degrees of freedom is small. See Hastie et al., 2001, p. 208.)

10.4.2 Estimates of model parameters

For exploration of parameter estimates in the model that includes all two-factor interactions, we re-fit the model used for `it2.lme`, but now using `method="REML"` (restricted maximum likelihood estimation), and examine the estimated effects. The parameter estimates that come from the REML analysis are in general preferable, because they avoid or reduce the biases of maximum likelihood estimates.)

```
> it2.reml <- update(it2.lme, method="REML")
> summary(it2.reml)$tTable
```

	Value	Std.Error	DF	t-value	p-value
(Intercept)	3.6191	0.130	145	27.82	5.3e-60
tint.L	0.1609	0.044	145	3.64	3.8e-04
tint.Q	0.0210	0.045	145	0.46	6.4e-01
targethicon	-0.1181	0.042	145	-2.79	6.0e-03
agegpOlder	0.4712	0.233	22	2.02	5.5e-02
sexm	0.0821	0.233	22	0.35	7.3e-01
tint.L:targethicon	-0.0919	0.046	145	-2.00	4.8e-02
tint.Q:targethicon	-0.0072	0.048	145	-0.15	8.8e-01


```
tint.L:agegpOlder      0.1308      0.049 145      2.66 8.7e-03
tint.Q:agegpOlder      0.0697      0.052 145      1.34 1.8e-01
tint.L:sexm            -0.0979      0.049 145     -1.99 4.8e-02
tint.Q:sexm            0.0054      0.052 145      0.10 9.2e-01
targethicon:agegpOlder -0.1389      0.058 145     -2.38 1.9e-02
targethicon:sexm       0.0779      0.058 145      1.33 1.8e-01
agegpOlder:sexm        0.3316      0.326  22      1.02 3.2e-01
```

Consider the relative amounts of evidence for the different sets of comparisons, and the consequences for the standard errors in the computer output.

- **Numbers of individuals:**

```
> attach(tinting)
> uid <- unique(tinting$id)
> subs <- match(uid, tinting$id)
> table(sex[subs], agegp[subs])
```

```
      Younger Older
f  9          4
m  4          9
```

Standard errors in the computer output, for comparisons made at this level, are in the range 0.23 - 0.32.

- **Numbers of comparisons between levels of `tint` or `target`:** Each of these comparisons is made at least as many times as there are individuals, i.e., at least 26 times. Standard errors in the computer output, for comparisons made at this level, are in the range 0.042 - 0.058.

10.5 Repeated Measures in Time

10.5.1 Example – random variation between profiles

The following code shows data from investigations designed to assess the feasibility of a proposed 119 kilometer human powered flight from the island of Crete – in the initial phase of the *Daedalus* project. After an initial 5-minute warm-up period and 5-minute recovery period, the power requirements from the athletes were increased, at two-minute intervals, in steps of around 30 watts.

```
## Plot points and fitted lines (panel A)
library(lattice)
xyplot(o2 ~ wattsPerKg, groups=id, data=humanpower1,
       panel=function(x, y, subscripts, groups, ...) {
         u <- lm(y~groups*x);
         hat <- fitted(u)
         panel.superpose(x, y, subscripts, groups)
         panel.superpose(x, hat, subscripts, groups, type="l")
       },
       xlab="Watts per kilogram",
       ylab=expression("Oxygen intake ("*ml.min^{-1}*kg^{-1}*")"))
```

The model that will be fitted is

$$y_{ij} = \alpha + \beta x_{ij} + a + b x_{ij} + e_{ij}$$

where i refers to individual, and j to observation j for that individual. α and β are fixed, a and b have a joint bivariate normal distribution, each with mean 0, independently of the e_{ij} which are i.i.d. normal. Each point in the figure given by the following code is a realization of an $(\alpha + a, \beta + b)$ pair.

```
## Calculate intercepts and slopes; plot Slopes vs Intercepts
## Uses the function lmList() from the nlme package
hp.lmList <- lmList(o2 ~ wattsPerKg|id, data=humanpower1)
coefs <- data.frame(t(sapply(hp.lmList, coef)))
names(coefs) <- c("Intercept", "Slope")
plot(Slope ~ Intercept, data=coefs)
abline(lm(Slope~Intercept, data=coefs))
```

The model allows, for each different athlete, a random slope (for wattsPerKg) and random intercept. We expect the correlation between the realizations of the random intercept and the random slope to be close to 1. As it will turn out, this will not create any difficulty. This is in fact easier than trying to work with one random parameter. The necessary code is:

```
hp.lme <- lme(o2 ~ I(wattsPerKg-3.214),
              random = ~ I(wattsPerKg-3.214) | id,
              data=humanpower1)
```

Subtracting off the mean of wattsPerKg (= 3.214) is optional, but may have benefits for the numerical optimization. The output is:

```
> summary(hp.lme)
Linear mixed-effects model fit by REML
Data: humanpower1
      AIC      BIC    logLik
136.2356 143.7842 -62.11782
\fc{
Random effects:
Formula: ~I(wattsPerKg - 3.214) | id
Structure: General positive-definite, Log-Cholesky parametrization
          StdDev  Corr
(Intercept)      1.470117 (Intr)
I(wattsPerKg - 3.214) 2.662602 0.997
Residual          2.031814

Fixed effects: o2 ~ I(wattsPerKg - 3.214)
              Value Std.Error DF  t-value p-value
(Intercept)      46.78798 0.7781026 22 60.13086      0
I(wattsPerKg - 3.214) 13.90774 1.3545133 22 10.26770      0
Correlation:
              (Intr)
I(wattsPerKg - 3.214) 0.79
```

```
Standardized Within-Group Residuals:
      Min       Q1       Med       Q3       Max
-1.99508076 -0.48513743 -0.08216073  0.45981369  2.11652299

Number of Observations: 28
Number of Groups: 5
```

The standard errors that are given for the fixed effects of the intercept and slope take no account of the random between individuals components of the intercepts and slopes. The standard errors relate to the accuracy of prediction of the mean response line for the population from which the athletes were sampled. The slopes are drawn from a distribution with estimated mean 13.9 and standard error $\sqrt{1.35^2 + 2.66^2} = 2.98$. This standard deviation may be compared with the standard deviation (= 3.28) of the five slopes that were fitted to the initial fixed effects model.⁴

The predicted lines from this model are shown as dashed lines in the following figure:

```
hp.lme <- lme(o2 ~ wattsPerKg, random = ~wattsPerKg | id,
             data=humanpower1)
hat <- predict(hp.lme)
xyplot(hat ~ wattsPerKg, groups=id, type="l", data=humanpower1)
## Alternatively, plot different lines in different panels
xyplot(hat ~ wattsPerKg | id, type="b", data=humanpower1)
```

These are the BLUPs (best linear unbiased predictors) that were discussed earlier in this chapter. Also of interest is a plot of residuals, with the points for each individual athlete connected with broken lines.⁵

10.5.2 Orthodontic measurements on children

The `Orthodont` data frame (*MASS* package) has measurements on the distance between two positions on the skull, taken every two years from age 8 until age 14, on 16 males and 11 females. The following code gives a figure that shows the pattern of change for each of the 25 individuals. Lines have been added

```
xyplot(distance ~ age | Subject, groups=Sex, data=Orthodont,
       type=c("p", "r"), par.strip.text=list(cex=0.75), layout=c(8, 4))
```

A good summary of these data are the intercepts and slopes, as in the figure given by the following code: It is good practice to use the intercept at the mean value of age, i.e., at age=11. This is both the mean both over all profiles and the mean for each individual profile. Because the line for each individual profile passes through the centroid (\bar{x}, \bar{y}) for that profile, the intercept is the mean of the y -values for that profile and is estimated independently of the slope of the profile.

```
ab <- t(sapply(split(Orthodont, Orthodont$Subject),
              function(u) coef(lm(distance ~ I(age-11), data=u))))

4## Derive the sd from the data frame coefs that was calculated above
sd(coefs$Slope)
5xyplot(resid(hp.lme) ~ wattsPerKg, groups=id, type="b", data=humanpower1)
```

```

ab <- cbind(ab, data.frame(Sex=sapply(split(as.character(Orthodont$Sex),
                                         Orthodont$Subject),function(x)x[1])))
names(ab) <- c("ybar", "b", "Sex")
sex <- as.character(ab$Sex)
plot(ab[, 1], ab[, 2], col=c(Female="red", Male="blue")[sex],
      pch=c(Female=1, Male=3)[sex], xlab="Intercept", ylab="Slope")
use <- ab$ybar %in% range(ab$ybar) | ab$b %in% range(ab$b) |
      ab$b == min(ab$b[sex=="Male"])
text(ab[use, 1], ab[use, 2], rownames(ab)[use], pos=4, xpd=TRUE)

```

One slope appears an outlier from the main body of the data. Hence, we omit the smallest and the largest values from the sample of male slopes, before doing a *t*-test.⁶The output is

Two Sample t-test

```

data:  b[Sex == "Male" & !extreme.males] and b[Sex == "Female"]
t = 2.87, df = 23, p-value = 0.008593
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.0742 0.4560
sample estimates:
mean of x mean of y
 0.745    0.480

```

Now consider a random effects model. We will fit two models, one assuming that the slopes are the same and the other allowing for different slopes. We will omit the same two males as before:

```

orthsame.lme <- lme(distance ~ Sex + I(age-11),
                   random = ~ I(age-11) | Subject,
                   data=Orthodont, subset=!extreme.males, method="ML")
orthdiff.lme <- update(orthsame.lme,
                      fixed = distance ~ Sex * I(age-11))

```

Now use the `anova()` function to compare these models:

```

> anova(orthsame.lme, orthdiff.lme)

```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
orthsame.lme	1	7	419.1	437.3	-202.6			
orthdiff.lme	2	8	414.6	435.4	-199.3	1 vs 2	6.537	0.0106

The estimates of fixed effects from the REML model are in general preferable to those from the full maximum likelihood (ML) model:

```

> orthdiffr.lme <- update(orthsame.lme, method="REML")
> summary(orthdiffr.lme)
Linear mixed-effects model fit by REML

```

```

6attach(ab)
extreme.males <- b %in% range(b[Sex=="Male"])
t.test(b[Sex=="Male" & !extreme.males], b[Sex=="Female"], var.equal=TRUE)
# Specify var.equal=TRUE, to allow comparison with anova output
detach(ab)

```

```

Data: Orthodont
  Subset: !extreme.males
      AIC   BIC logLik
418.9 439.5 -201.5

Random effects:
Formula: ~I(age - 11) | Subject
Structure: General positive-definite, Log-Cholesky parametrization
      StdDev Corr
(Intercept) 1.8636 (Intr)
I(age - 11) 0.2209 0.216
Residual    1.2772

Fixed effects: distance ~ Sex + I(age - 11) + Sex:I(age - 11)
      Value Std.Error DF t-value p-value
(Intercept) 24.933    0.4956 71  50.31 0.0000
SexFemale   -2.291    0.7760 25  -2.95 0.0068
I(age - 11)  0.856    0.0947 71   9.04 0.0000
SexFemale:I(age - 11) -0.381    0.1465 71  -2.60 0.0113
Correlation:
      (Intr) SexFml I(-11)
SexFemale      -0.639
I(age - 11)     0.110 -0.070
SexFemale:I(age - 11) -0.071 0.124 -0.646

Standardized Within-Group Residuals:
      Min      Q1      Med      Q3      Max
-3.32968 -0.38808 0.01614 0.41296 3.19724

Number of Observations: 100
Number of Groups: 27

```

The output gives 71 degrees of freedom for the difference in slopes where there should be 23, as in the *t*-test that was carried out earlier. The comparison that used `anova()` gave the degrees of freedom correctly.

Correlation between successive times

We have not checked for sequential correlation at the successive time points. This can be done with `plot(ACF(orthdiff.lme), alpha = 0.025)`, which causes the inclusion of dotted lines that show the critical values for a two-sided test at a level of $\alpha = 0.05$. The autocorrelation estimate is an average across all profiles. A correlation of -0.5 at lag one does then show up as statistically significant.

The function `corAR1()` may be used to fit a correlation at lag 1:

```

> orthdiffAR1.lme <- update(orthdiff.lme,
+                           correlation=corAR1(form=~age|Subject))
> anova(orthdiff.lme, orthdiffAR1.lme)

```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
	orthdiff.lme	1	8 414.6	435.4	-199.3			
	orthdiffAR1.lme	2	9 416.6	440.0	-199.3	1 vs 2	3.411e-13	1

The estimated correlation is zero, which sits oddly with the indications from the plot of the autocorrelation function. Where inclusion of a sequential correlation does improve the modeling at the within time covariate level, the effect may be to improve sensitivity for comparisons at the between groups level.