Mixed-Effects Models in R

An Appendix to *An R Companion to Applied Regression, Second Edition*

John Fox & Sanford Weisberg

last revision (DRAFT): November 6, 2014

Abstract

1 Introduction

The normal linear model (described, for example, in Chapter 4 of the text),

$$\begin{aligned} y_i &= \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_p x_{pi} + \varepsilon_i \\ \varepsilon_i &\sim \text{NID}(0, \sigma^2) \end{aligned}$$

has one *random effect* (in the terminology of mixed-effects models, the subject of this appendix), the error term ε_i. The parameters of the model are the regression coefficients, $\beta_1, \beta_2, ..., \beta_p$, and the error variance, σ^2 (a *variance component*). Usually, $x_{1i} = 1$, and so β_1 is a constant or intercept.

For comparison with the linear mixed model of the next section, we rewrite the linear model in matrix form,

$$\begin{aligned} \mathbf{y} &= \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \\ \boldsymbol{\varepsilon} &\sim \mathbf{N}_n(\mathbf{0}, \sigma^2 \mathbf{I}_n) \end{aligned}$$

where $\mathbf{y} = (y_1, y_2, ..., y_n)'$ is the response vector; $\mathbf{X}$ is the model matrix, with typical row $\mathbf{x}_i' = (x_{1i}, x_{2i}, ..., x_{pi})$; $\boldsymbol{\beta} = (\beta_1, \beta_2, ..., \beta_p)'$ is the vector of regression coefficients; $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, ..., \varepsilon_n)'$ is the vector of errors; $\mathbf{N}_n$ represents the n-variable multivariate-normal distribution; $\mathbf{0}$ is an $n \times 1$ vector of 0s; and $\mathbf{I}_n$ is the order-n identity matrix.

So-called *mixed-effect models* (or just *mixed models*) include additional random-effect terms (and associated variance and covariance components), and are often appropriate for representing clustered, and therefore dependent, data — arising, for example, when data are collected hierarchically, when observations are taken on related individuals (such as siblings), or when data are gathered over time on the same individuals.

There are several facilities in R for fitting mixed models to data, the most commonly used of which are the **nlme** (Pinheiro and Bates, 2000; Pinheiro et al., 2014) and **lme4** (Bates et al., 2014) packages, and which we discuss in this appendix.[1] The **nlme** package is a part of the standard R distribution, and the **lme4** package is available on CRAN.

[1] **nlme** stands for **n**on**l**inear **m**ixed **e**ffects, even though the package also includes the `lme` function for fitting *linear* mixed models. Similarly, **lme4** stands for **l**inear **m**ixed **e**ffects with S**4** classes, but also includes functions for fitting *generalized linear* and *nonlinear* mixed models.

Section 2 describes how to fit linear mixed models in R. Sections 3 and 4 deal respectively with generalized linear mixed models and nonlinear mixed models. Mixed models are a large and complex subject, and we will only scratch the surface here. Bayesian approaches, which we do not cover, are also common and are available in R: See the complementary readings in Section 5.

2 Linear Mixed Models

Linear mixed models (*LMMs*) may be expressed in different but equivalent forms. In the social and behavioral sciences, it is common to express such models in hierarchical form, as illustrated in Section 2.1. The `lme` (`l`inear `m`ixed `e`ffects) function in the **nlme** package and the `lmer` (`l`inear `m`ixed-`e`ffects `r`egression, pronounced "elmer") function in the **lme4** package, however, employ the *Laird-Ware form* of the LMM (after a seminal paper on the topic published by Laird and Ware, 1982):

$$\begin{aligned} y_{ij} &= \beta_1 x_{1ij} + \cdots + \beta_p x_{pij} \\ &\quad + b_{i1} z_{1ij} + \cdots + b_{iq} z_{qij} + \varepsilon_{ij} \\ b_{ik} &\sim \mathrm{N}(0, \psi_k^2), \mathrm{Cov}(b_k, b_{k'}) = \psi_{kk'} \\ \varepsilon_{ij} &\sim \mathrm{N}(0, \sigma^2 \lambda_{ijj}), \mathrm{Cov}(\varepsilon_{ij}, \varepsilon_{ij'}) = \sigma^2 \lambda_{ijj'} \end{aligned} \tag{1}$$

where

- y_{ij} is the value of the response variable for the jth of n_i observations in the ith of M groups or clusters.
- $\beta_1, \ldots, \beta_p$ are the fixed-effect coefficients, which are identical for all groups.
- $x_{1ij}, \ldots, x_{pij}$ are the fixed-effect regressors for observation j in group i; the first regressor is usually for the regression constant, $x_{1ij} = 1$.
- $b_{i1}, \ldots, b_{iq}$ are the random-effect coefficients for group i, assumed to be multivariately normally distributed. The random effects, therefore, vary by group. The b_{ik} are thought of as random variables, not as parameters, and are similar in this respect to the errors ε_{ij}.
- $z_{1ij}, \ldots, z_{qij}$ are the random-effect regressors.
- ψ_k^2 are the variances and $\psi_{kk'}$ the covariances among the random effects, assumed to be constant across groups. In some applications, the ψs are parametrized in terms of a relatively small number of fundamental parameters.
- ε_{ij} is the error for observation j in group i. The errors for group i are assumed to be multivariately normally distributed.
- $\sigma^2 \lambda_{ijj'}$ is the covariance between errors ε_{ij} and $\varepsilon_{ij'}$ in group i. Generally, the $\lambda_{ijj'}$ are parametrized in terms of a few basic parameters, and their specific form depends upon context. For example, when observations are sampled independently within groups and are assumed to have constant error variance (as in the application developed in Section 2.1), $\lambda_{ijj} = \sigma^2$, $\lambda_{ijj'} = 0$ (for $j \neq j'$), and thus the only free parameter to estimate is the common error variance, σ^2. The `lmer` function in the **lme4** package handles only models of this form. In contrast, if the observations in a "group" represent longitudinal data on a single individual, then the structure of the λs may be specified to capture autocorrelation among the errors, as

is common in observations collected over time. The `lme` function in the **nlme** package can handle autocorrelated and heteoscedastic errors.

Alternatively but equivalently, in matrix form,

$$\begin{aligned} \mathbf{y}_i &= \mathbf{X}_i\boldsymbol{\beta} + \mathbf{Z}_i\mathbf{b}_i + \boldsymbol{\varepsilon}_i \\ \mathbf{b}_i &\sim \mathbf{N}_q(\mathbf{0}, \boldsymbol{\Psi}) \\ \boldsymbol{\varepsilon}_i &\sim \mathbf{N}_{n_i}(\mathbf{0}, \sigma^2\boldsymbol{\Lambda}_i) \end{aligned}$$

where

- $\mathbf{y}_i$ is the $n_i \times 1$ response vector for observations in the ith group.
- $\mathbf{X}_i$ is the $n_i \times p$ model matrix for the fixed effects for observations in group i.
- $\boldsymbol{\beta}$ is the $p \times 1$ vector of fixed-effect coefficients.
- $\mathbf{Z}_i$ is the $n_i \times q$ model matrix for the random effects for observations in group i.
- $\mathbf{b}_i$ is the $q \times 1$ vector of random-effect coefficients for group i.
- $\boldsymbol{\varepsilon}_i$ is the $n_i \times 1$ vector of errors for observations in group i.
- $\boldsymbol{\Psi}$ is the $q \times q$ covariance matrix for the random effects.
- $\sigma^2\boldsymbol{\Lambda}_i$ is the $n_i \times n_i$ covariance matrix for the errors in group i; for the `lmer` function the error covariance matrix for group i is $\sigma^2\mathbf{I}_{n_i}$.

2.1 An Illustrative Application to Hierarchical Data

Applications of mixed models to hierarchical data have become common in the social sciences, and nowhere more so than in research on education. The following example is borrowed from Raudenbush and Bryk's influential text on hierarchical linear models (Raudenbush and Bryk, 2002), and also appears in a paper by Singer (1998), which shows how such models can be fit by the `MIXED` procedure in SAS. In this section, we will show how to model Raudenbush and Bryk's data using the `lme` function in the **nlme** package and the `lmer` function in the **lme4** package.

The data for the example, from the 1982 "High School and Beyond" survey, are for 7185 high-school students from 160 schools. There are, therefore, on average $7185/160 \approx 45$ students per school. The data are conveniently available in the data frames `MathAchieve` and `MathAchSchool` in the **nlme** package:[2]

```
> library(nlme)
> head(MathAchieve, 10)  # first 10 students

Grouped Data: MathAch ~ SES | School
   School Minority    Sex    SES MathAch MEANSES
1    1224       No Female -1.528   5.876  -0.428
2    1224       No Female -0.588  19.708  -0.428
3    1224       No   Male -0.528  20.349  -0.428
```

[2]These are actually *grouped-data objects*, which inherit from data-frame objects, and which include some additional information along with the data. We briefly discuss grouped-data objects later in this appendix.

```
4    1224        No   Male -0.668   8.781  -0.428
5    1224        No   Male -0.158  17.898  -0.428
6    1224        No   Male  0.022   4.583  -0.428
7    1224        No Female -0.618  -2.832  -0.428
8    1224        No   Male -0.998   0.523  -0.428
9    1224        No Female -0.888   1.527  -0.428
10   1224        No   Male -0.458  21.521  -0.428
```

```
> dim(MathAchieve)
```

```
[1] 7185    6
```

```
> head(MathAchSchool, 10)  # first 10 schools
```

```
     School Size   Sector PRACAD DISCLIM HIMINTY MEANSES
1224   1224  842   Public   0.35   1.597       0  -0.428
1288   1288 1855   Public   0.27   0.174       0   0.128
1296   1296 1719   Public   0.32  -0.137       1  -0.420
1308   1308  716 Catholic   0.96  -0.622       0   0.534
1317   1317  455 Catholic   0.95  -1.694       1   0.351
1358   1358 1430   Public   0.25   1.535       0  -0.014
1374   1374 2400   Public   0.50   2.016       0  -0.007
1433   1433  899 Catholic   0.96  -0.321       0   0.718
1436   1436  185 Catholic   1.00  -1.141       0   0.569
1461   1461 1672   Public   0.78   2.096       0   0.683
```

```
> dim(MathAchSchool)
```

```
[1] 160   7
```

The first data frame pertains to students, and there is therefore one row in the data frame for each of the 7185 students; the second data frame pertains to schools, and there is one row for each of the 160 schools. We will use the following variables:

- `School`: an identification number for the student's school. Although it is not required by `lme` or `lmer`, students in a specific school are in consecutive rows of the data frame, a convenient form of data organization. The schools define groups — it is unreasonable to suppose that students in the same school are independent of one-another.

- `SES`: the socioeconomic status of the student's family, centered to an overall mean of 0 (within rounding error).

- `MathAch`: the student's score on a math-achievement test.

- `Sector`: a factor coded `"Catholic"` or `"Public"`. This is a school-level variable and hence is identical for all students in the same school. A variable of this kind is sometimes called an *outer variable* or a *contextual variable*, to distinguish it from an *inner variable* or *individual-level variable* (such as `SES`), which varies within groups. Because the `Sector` variable resides in the school data set, we need to copy it over to the appropriate rows of the student data set. Such data-management tasks are common in preparing data for mixed-modeling.[3]

- MEANSES: another outer variable, giving the mean SES for students in each school; we call outer variables that aggregate individual-level data to the group level *compositional variables*. Notice that this variable already appears in both data sets. The variable, however, seems to have been calculated incorrectly — that is, its values are slightly different from the school means computed directly from the MathAchieve data set — and we will therefore recompute it (using tapply — see Section 8.4 of the text) and replace it in the student data set:[4]

```
> mses <- with(MathAchieve, tapply(SES, School, mean))
> mses[as.character(MathAchSchool$School[1:10])]  # for first 10 schools

    1224     1288     1296     1308     1317     1358     1374     1433
-0.43438  0.12160 -0.42550  0.52800  0.34533 -0.01967 -0.01264  0.71200
    1436     1461
 0.56291  0.67745
```

We begin by creating a new data frame, called Bryk, containing the inner variables that we require:

```
> Bryk <- as.data.frame(MathAchieve[, c("School", "SES", "MathAch")])
> names(Bryk) <- tolower(names(Bryk))
> set.seed(12345)  # for reproducibility
> (sample20 <- sort(sample(nrow(Bryk), 20)))  # 20 randomly sampled students

 [1]    9  248 1094 1195 1283 2334 2783 2806 2886 3278 3317 3656 5180 5223 5278
[16] 5467 6292 6365 6820 7103

> Bryk[sample20, ]

     school    ses mathach
9      1224 -0.888   1.527
248    1433  1.332  18.496
1094   2467  0.062   6.415
1195   2629  0.942  11.437
1283   2639 -1.088  -0.763
2334   3657 -0.288  13.156
2783   4042  0.792  14.500
2806   4042  0.482   3.687
2886   4223  1.242  20.375
3278   4511 -0.178  15.550
3317   4511  0.342   7.447
```

[3]This data-management task is implied by the Laird-Ware form of the LMM. Some software that is specifically oriented towards modeling hierarchical data employs two data files — one for contextual variables and one for individual-level variables — corresponding respectively to the MathAchieveSchool and MathAchieve data sets in the present example.

[4]We are not sure why the school means given in the MathAchieveSchool and MathAchieve data sets differ from the values that we compute directly. It is possible that the values in these data sets were computed from larger populations of students in the sampled schools.

```
3656   5404  0.902  18.802
5180   7232  0.442  23.591
5223   7276 -1.098  -1.525
5278   7332 -0.508  16.114
5467   7364 -0.178  20.325
6292   8707 -0.228  18.463
6365   8800 -0.658  11.928
6820   9198 -0.538   2.349
7103   9550  0.752   4.285
```

Using `as.data.frame`, we make `Bryk` an ordinary data frame rather than a grouped-data object. We rename the variables to lower-case in conformity with our usual practice — data frames start with upper-case letters, variables with lower-case letters.

Next, we add the outer variables to the data frame, in the process computing a version of SES, called `cses`, that is centered at the school means:

```
> sector <- MathAchSchool$Sector
> names(sector) <- row.names(MathAchSchool)
> Bryk <- within(Bryk,{
+     meanses <- as.vector(mses[as.character(school)])
+     cses <- ses - meanses
+     sector <- sector[as.character(school)]
+ })
> Bryk[sample20, ]
```

```
     school    ses mathach   sector     cses  meanses
9      1224 -0.888   1.527   Public -0.45362 -0.43438
248    1433  1.332  18.496 Catholic  0.62000  0.71200
1094   2467  0.062   6.415   Public  0.39173 -0.32973
1195   2629  0.942  11.437 Catholic  1.07965 -0.13765
1283   2639 -1.088  -0.763   Public -0.12357 -0.96443
2334   3657 -0.288  13.156   Public  0.36118 -0.64918
2783   4042  0.792  14.500 Catholic  0.39000  0.40200
2806   4042  0.482   3.687 Catholic  0.08000  0.40200
2886   4223  1.242  20.375 Catholic  1.33600 -0.09400
3278   4511 -0.178  15.550 Catholic -0.07086 -0.10714
3317   4511  0.342   7.447 Catholic  0.44914 -0.10714
3656   5404  0.902  18.802 Catholic  0.07702  0.82498
5180   7232  0.442  23.591   Public  0.53212 -0.09012
5223   7276 -1.098  -1.525   Public -1.17623  0.07823
5278   7332 -0.508  16.114 Catholic -0.80500  0.29700
5467   7364 -0.178  20.325 Catholic -0.08864 -0.08936
6292   8707 -0.228  18.463   Public -0.38313  0.15513
6365   8800 -0.658  11.928 Catholic  0.05125 -0.70925
6820   9198 -0.538   2.349 Catholic -1.03000  0.49200
7103   9550  0.752   4.285   Public  0.69897  0.05303
```

These steps are a bit tricky:

- The students' school numbers (in `school`) are converted to character values, used to index the outer variables in the school dataset. This procedure assigns the appropriate values of `meanses` and `sector` to each student.

- To make this indexing work for the `Sector` variable in the school data set, the variable is assigned to the global vector `sector`, whose names are then set to the row names of the school data frame.

Following Raudenbush and Bryk, we will ask whether students' math achievement is related to their socioeconomic status; whether this relationship varies systematically by sector; and whether the relationship varies randomly across schools within the same sector.

2.1.1 Examining the Data

As in all data analysis, it is advisable to examine the data before embarking upon statistical modeling. There are too many schools to look at each individually, so we start by selecting samples of 20 public and 20 Catholic schools, storing each sample in a data frame:

```
> cat <- with(Bryk, sample(unique(school[sector == "Catholic"]), 20))
> Cat.20 <- Bryk[is.element(Bryk$school, cat), ]
> dim(Cat.20)

[1] 1027    6

> pub <- with(Bryk, sample(unique(school[sector == "Public"]), 20))
> Pub.20 <- Bryk[is.element(Bryk$school, pub), ]
> dim(Pub.20)

[1] 739   6
```

We use Trellis graphics (provided by the **lattice** package — see Section 7.3.1 of the text) to visualize the relationship between math achievement and school-centered SES in the sampled schools:

```
> library(lattice) # for Trellis graphics
> trellis.device(color=FALSE)
> xyplot(mathach ~ cses | school, data=Cat.20, main="Catholic",
+     panel=function(x, y){
+         panel.xyplot(x, y)
+         panel.loess(x, y, span=1)
+         panel.lmline(x, y, lty=2)
+     }
+ )
> xyplot(mathach ~ cses | school, data=Pub.20, main="Public",
+     panel=function(x, y){
+         panel.xyplot(x, y)
+         panel.loess(x, y, span=1)
+         panel.lmline(x, y, lty=2)
+     }
+ )
```

- The call to `trellis.device` creates a graphics-device window appropriately set up for Trellis graphics; in this case, we specified monochrome graphics (`color = FALSE`) so that this appendix will print well in black-and-white; the default is to use color.

- The `xyplot` function draws a Trellis display of scatterplots of math achievement against socioeconomic status, one scatterplot for each school, as specified by the formula `mathach ~ ses | school`. The school number appears in the strip label above each plot. We created one graph for Catholic schools (Figure 1) and another for public schools (Figure 2). The argument `main` to `xyplot` supplies the title of each graph.

- The content of each cell (or *panel*) of the Trellis display is determined by the `panel` argument to `xyplot`, here an anonymous function defined "on the fly." This function takes two arguments, `x` and `y`, giving respectively the horizontal and vertical coordinates of the points in a panel, and successively calls three standard panel functions:

  - `panel.xyplot` (which is the default panel function for `xyplot`) creates a basic scatterplot.
  - `panel.loess` draws a local regression line on the plot. Because there is a modest number of observations for each school, we set the span of the local-regression smoother to 1. (See the Appendix on nonparametric regression for details.)
  - `panel.lmline` similarly draws a least-squares line; the argument `lty=2` produces a broken line.

Examining the scatterplots in Figures 1 and 2, there is a weak positive relationship between math achievement and SES in most Catholic schools, although there is variation among schools: In some schools the slope of the regression line is near 0 or even negative. There is also a positive relationship between the two variables for most of the public schools, and here the average slope is larger. Considering the moderate number of students in each school, linear regressions appear to do a reasonable job of capturing the within-school relationships between math achievement and SES.

The **nlme** package includes the function `lmList` for fitting a linear model to the observations in each group, returning a list of linear-model objects, which is itself an object of class `"lmList"`.[5] Here, we fit the regression of math-achievement scores on centered socioeconomic status for each school, creating separate `"lmList"` objects for Catholic and public schools:

```
> cat.list <- lmList(mathach ~ cses | school, subset = sector=="Catholic",
+      data=Bryk)
> pub.list <- lmList(mathach ~ cses | school, subset = sector=="Public",
+      data=Bryk)
```

Several methods exist for manipulating `"lmList"` objects. For example, the generic `intervals` function has a method for objects of this class that returns (by default) 95-percent confidence intervals for the regression coefficients; the confidence intervals can be plotted, as follows:

```
> plot(intervals(cat.list), main="Catholic")

> plot(intervals(pub.list), main="Public")
```

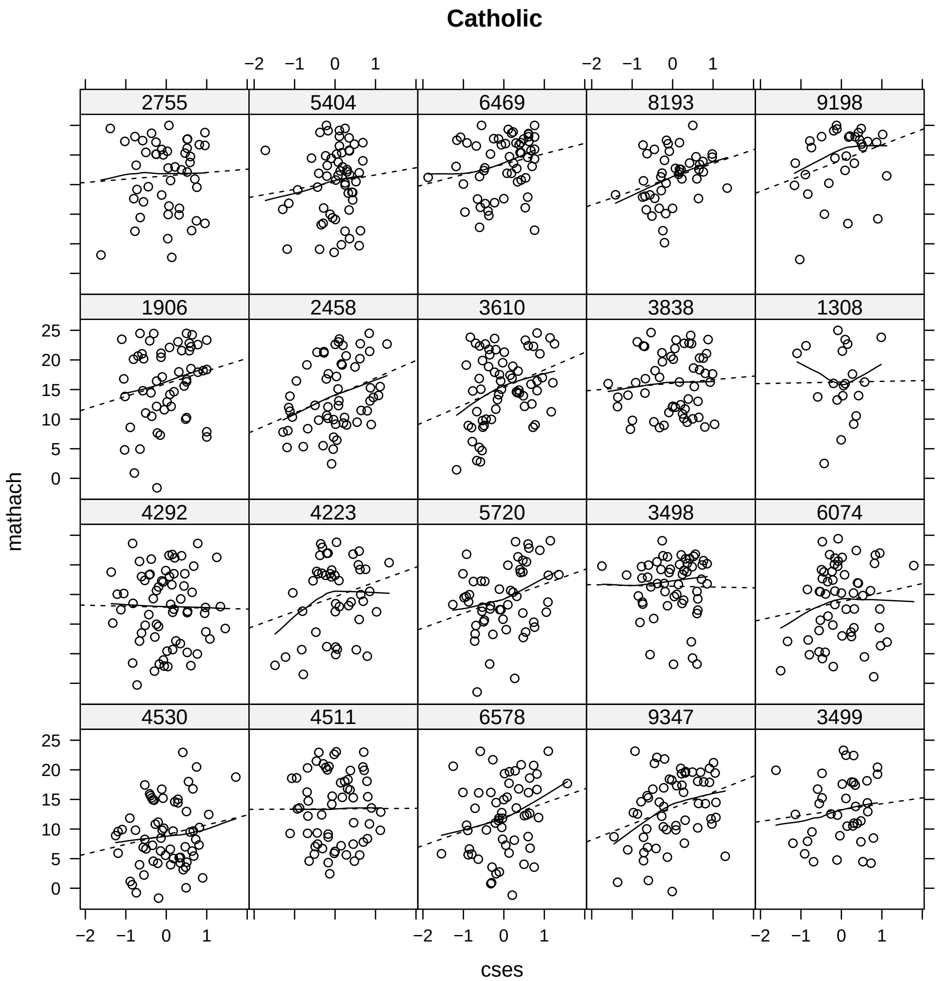

Figure 1: Trellis display of math achievement by socio-economic status for 20 randomly selected Catholic schools. The broken lines give linear least-squares fits, the solid lines local-regression fits.

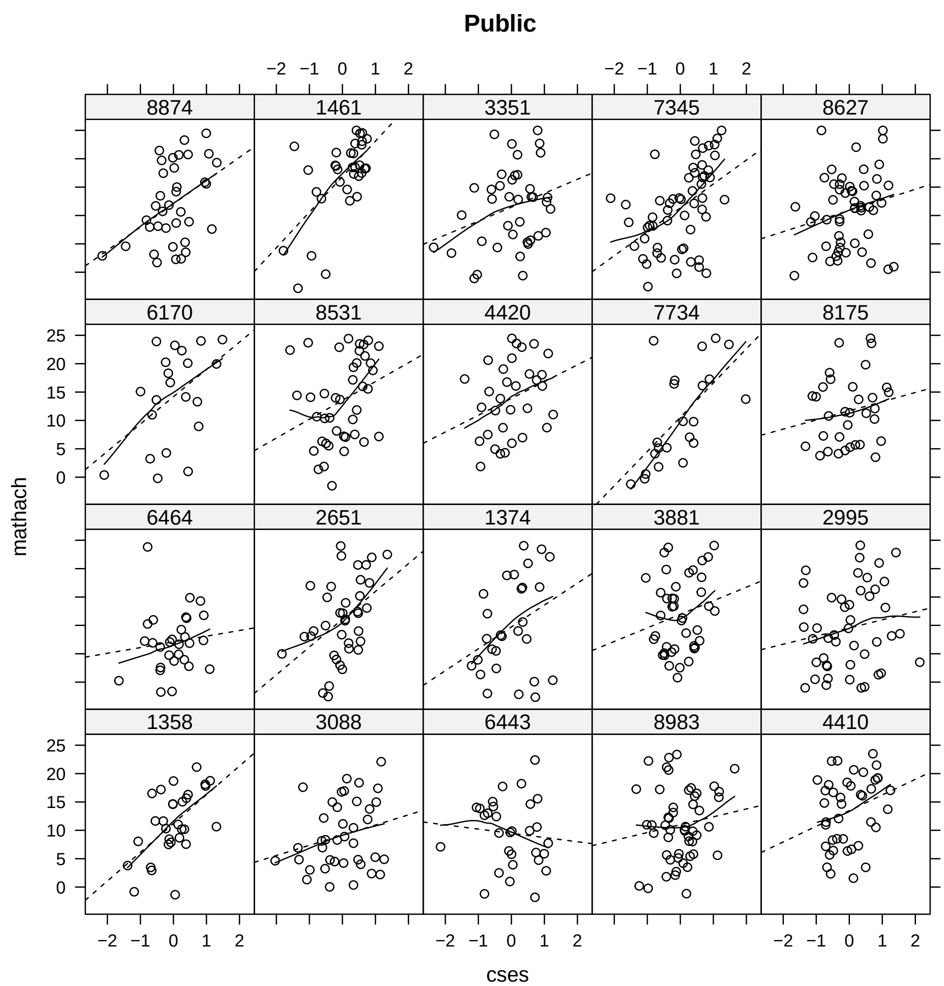

Figure 2: Trellis display of math achievement by socio-economic status for 20 randomly selected public schools.

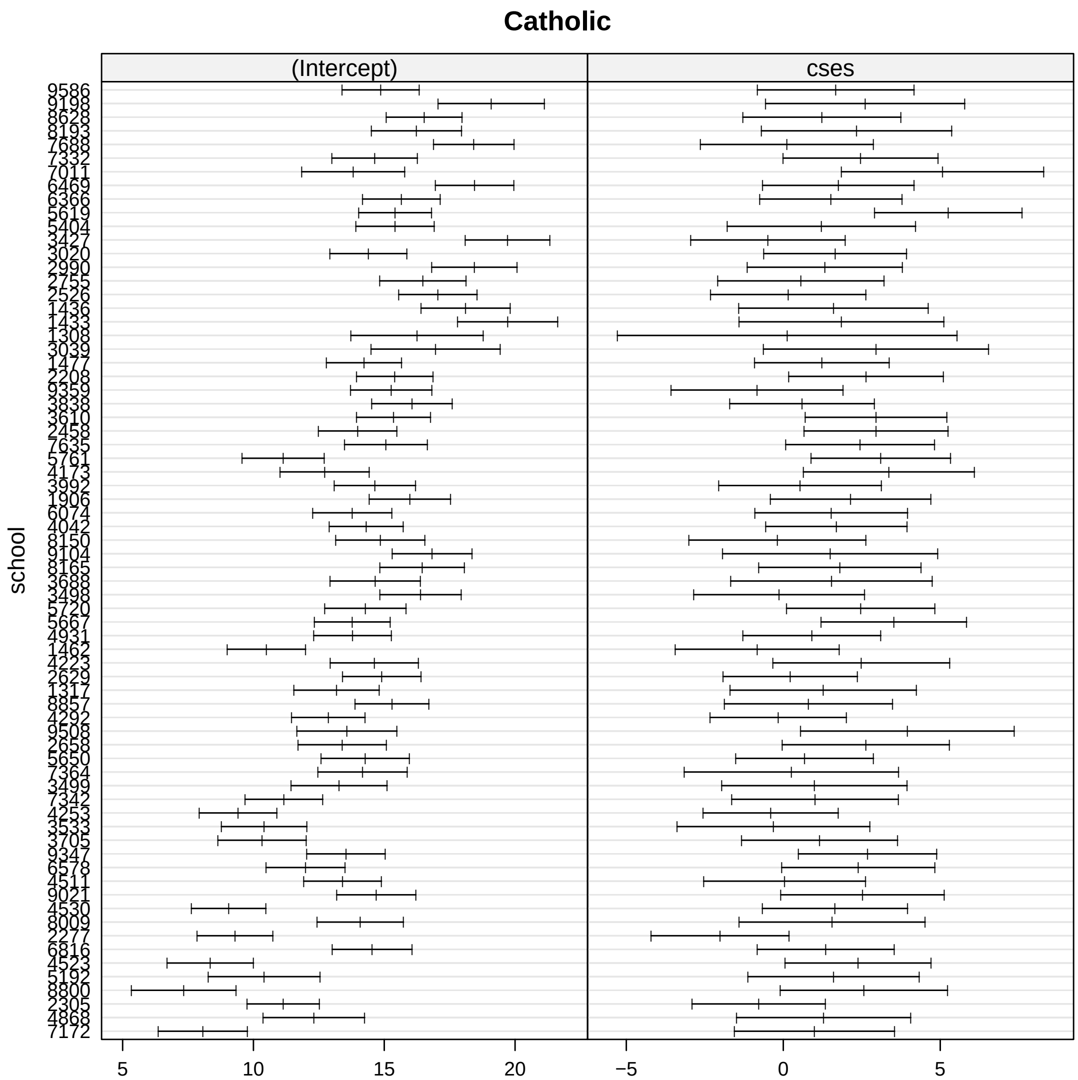

Figure 3: 95-percent confidence intervals for the intercepts and slopes of the within-schools regressions of math achievement on centered SES, for Catholic schools.

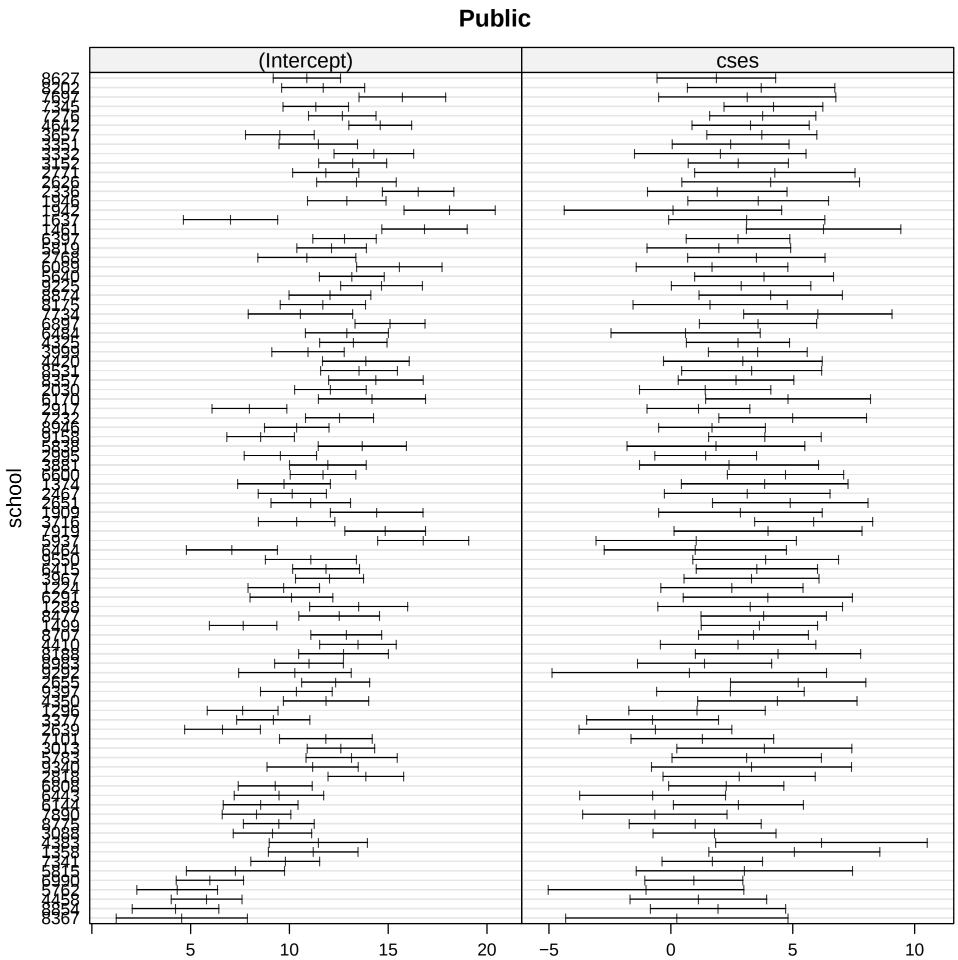

Figure 4: 95-percent confidence intervals for the intercepts and slopes of the within-schools regressions of math achievement on centered SES, for public schools.

The resulting graphs are shown in Figures 3 and 4. In interpreting these graphs, we need to be careful to take into account that we have not constrained the scales for the plots to be the same, and indeed the scales for the intercepts and slopes in the public schools are wider than in the Catholic schools. Because the SES variable is centered to 0 within schools, the intercepts are interpretable as the average level of math achievement in each school. It is clear that there is substantial variation in the intercepts among both Catholic and public schools; the confidence intervals for the slopes, in contrast, overlap to a much greater extent, but there is still apparent school-to-school variation.

To facilitate comparisons between the distributions of intercepts and slopes across the two sectors, we draw parallel boxplots of the coefficients:

```
> cat.coef <- coef(cat.list)
> head(cat.coef, 10)

     (Intercept)    cses
7172       8.067  0.9945
4868      12.310  1.2865
2305      11.138 -0.7821
8800       7.336  2.5681
5192      10.409  1.6035
4523       8.352  2.3808
6816      14.538  1.3527
2277       9.298 -2.0150
8009      14.085  1.5569
4530       9.056  1.6474

> pub.coef <- coef(pub.list)
> head(pub.coef, 10)

     (Intercept)    cses
8367       4.553  0.2504
8854       4.240  1.9388
4458       5.811  1.1318
5762       4.325 -1.0141
6990       5.977  0.9477
5815       7.271  3.0180
7341       9.794  1.7037
1358      11.206  5.0680
4383      11.466  6.1802
3088       9.146  1.7913

> old <- par(mfrow=c(1, 2))
> boxplot(cat.coef[, 1], pub.coef[, 1], main="Intercepts",
+     names=c("Catholic", "Public"))
> boxplot(cat.coef[, 2], pub.coef[, 2], main="Slopes",
+     names=c("Catholic", "Public"))
> par(old) # restore
```

[5] A similar function is included in the **lme4** package.

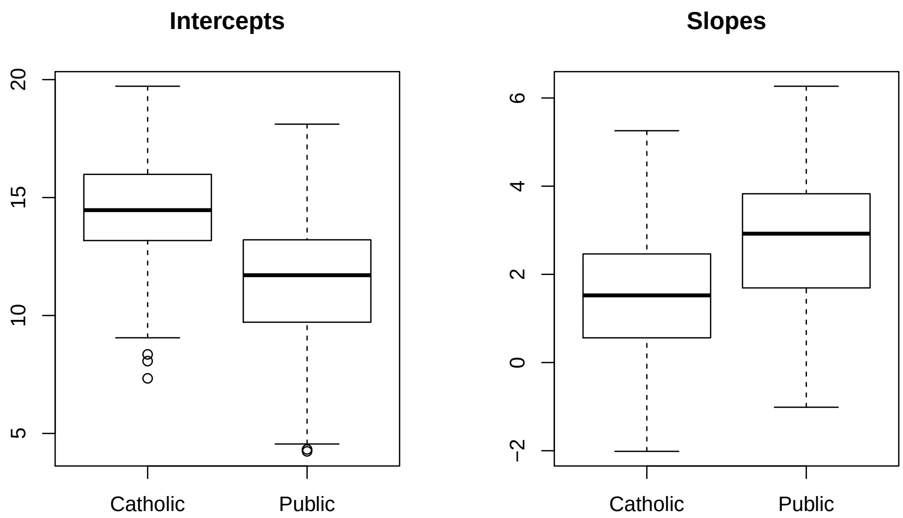

Figure 5: Boxplots of intercepts and slopes for the regressions of math achievement on centered SES in Catholic and public schools..

The calls to `coef` extract matrices of regression coefficients from the `lmList` objects, with rows representing schools. Setting the plotting parameter `mfrow` to `1` row and `2` columns produces the side-by-side pairs of boxplots in Figure 5; `mfrow` is then returned to its previous value. The Catholic schools have a higher average level of math achievement than the public schools, while the average slope relating math achievement to SES is larger in the public schools than in the Catholic schools.

2.1.2 Fitting a Hierarchical Linear Model with `lme`

Following Raudenbush and Bryk (2002) and Singer (1998), we will fit a hierarchical linear model to the math-achievement data. This model consists of two equations: First, within schools, we have the regression of math achievement on the individual-level covariate SES; it aids interpretability of the regression coefficients to center SES at the school average; then the intercept for each school estimates the average level of math achievement in the school.

Using centered SES, the individual-level equation for individual j in school i is

$$\text{mathach}_{ij} = \alpha_{0i} + \alpha_{1i}\text{cses}_{ij} + \varepsilon_{ij} \tag{2}$$

At the school level, also following Raudenbush, Bryk, and Singer, we will entertain the possibility that the school intercepts and slopes depend upon sector and upon the average level of SES in the schools:

$$\begin{aligned} \alpha_{0i} &= \gamma_{00} + \gamma_{01}\text{meanses}_i + \gamma_{02}\text{sector}_i + u_{0i} \qquad (3) \\ \alpha_{1i} &= \gamma_{10} + \gamma_{11}\text{meanses}_i + \gamma_{12}\text{sector}_i + u_{1i} \end{aligned}$$

This kind of formulation is sometimes called a *coefficients-as-outcomes* model.[6]

[6] This coefficients-as-outcomes model assumes that the regressions of the within-school intercepts and slopes on school mean SES are linear. We invite the reader to examine this assumption by creating scatterplots of the within-

Substituting the school-level Equation 3 into the individual-level Equation 2 produces

$$\begin{aligned}\text{mathach}_{ij} &= \gamma_{00} + \gamma_{01}\text{meanses}_i + \gamma_{02}\text{sector}_i + u_{0i} \\ &\quad +(\gamma_{10} + \gamma_{11}\text{meanses}_i + \gamma_{12}\text{sector}_i + u_{1j})\text{cses}_{ij} + \varepsilon_{ij}\end{aligned}$$

Rearranging terms,

$$\begin{aligned}\text{mathach}_{ij} &= \gamma_{00} + \gamma_{01}\text{meanses}_i + \gamma_{02}\text{sector}_i + \gamma_{10}\text{cses}_{ij} \\ &\quad +\gamma_{11}\text{meanses}_i\text{cses}_{ij} + \gamma_{12}\text{sector}_i\text{cses}_{ij} \\ &\quad +u_{0i} + u_{1i}\text{cses}_{ij} + \varepsilon_{ij}\end{aligned}$$

Here, the γs are fixed effects, while the us (and the individual-level errors ε_{ij}) are random effects.

Finally, rewriting the model in the notation of the LMM (Equation 1),

$$\begin{aligned}\text{mathach}_{ij} &= \beta_1 + \beta_2\text{meanses}_i + \beta_3\text{sector}_i + \beta_4\text{cses}_{ij} \\ &\quad +\beta_5\text{meanses}_i\text{cses}_{ij} + \beta_6\text{sector}_i\text{cses}_{ij} \\ &\quad +b_{i1} + b_{i2}\text{cses}_{ij} + \varepsilon_{ij}\end{aligned} \tag{4}$$

The change is purely notational, using βs for fixed effects and bs for random effects. (In the data set, however, the school-level variables — that is, `meanses` and `sector` — are attached to the observations for the individual students, as previously described.) We place no constraints on the covariance matrix of the random effects, so

$$\mathbf{\Psi} = V\begin{bmatrix} b_{i1} \\ b_{i2}\end{bmatrix} = \begin{bmatrix} \psi_1^2 & \psi_{12} \\ \psi_{12} & \psi_2^2\end{bmatrix}$$

but assume that the individual-level errors are independent within schools, with constant variance:

$$V(\boldsymbol{\varepsilon}_i) = \sigma^2\mathbf{I}_{n_i}$$

As mentioned in Section 2, LMMs are fit with the `lme` function in the **nlme** package. Specifying the fixed effects in the call to `lme` is identical to specifying a linear model in a call to `lm` (see Chapter 4 of the text). Random effects are specified via the `random` argument to `lme`, which takes a one-sided model formula.

Before fitting a mixed model to the math-achievement data, we reorder the levels of the factor `sector` so that the contrast for `sector` will use the value `0` for the public sector and `1` for the Catholic sector, in conformity with the coding employed by Raudenbush and Bryk (2002) and by Singer (1998):

```
> Bryk$sector <- factor(Bryk$sector, levels=c("Public", "Catholic"))
> contrasts(Bryk$sector)

         Catholic
Public          0
Catholic        1
```

Having established the contrast-coding for `sector`, the LMM in Equation 4 is fit as follows:

school regression coefficients for Catholic and public schools, computed in the previous section, against school mean SES, modifying the hierarchical model in light of these graphs if the relationships appear nonlinear. For an analysis along these lines, see the discussion of the High School and Beyond data in Fox (ress).

```
> bryk.lme.1 <- lme(mathach ~ meanses*cses + sector*cses,
+     random = ~ cses | school,
+     data=Bryk)
> summary(bryk.lme.1)

Linear mixed-effects model fit by REML
 Data: Bryk
    AIC   BIC logLik
  46524 46592 -23252

Random effects:
 Formula: ~cses | school
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev Corr
(Intercept) 1.5426 (Intr)
cses        0.3182 0.391
Residual    6.0598

Fixed effects: mathach ~ meanses * cses + sector * cses
                     Value Std.Error   DF t-value p-value
(Intercept)         12.128    0.1993 7022   60.86  0.0000
meanses              5.333    0.3692  157   14.45  0.0000
cses                 2.945    0.1556 7022   18.93  0.0000
sectorCatholic       1.227    0.3063  157    4.00  0.0001
meanses:cses         1.039    0.2989 7022    3.48  0.0005
cses:sectorCatholic -1.643    0.2398 7022   -6.85  0.0000
 Correlation:
                    (Intr) meanss cses   sctrCt mnss:c
meanses              0.256
cses                 0.075  0.019
sectorCatholic      -0.699 -0.356 -0.053
meanses:cses         0.019  0.074  0.293 -0.026
cses:sectorCatholic -0.052 -0.027 -0.696  0.077 -0.351

Standardized Within-Group Residuals:
     Min       Q1      Med       Q3      Max
-3.15926 -0.72319  0.01705  0.75445  2.95822

Number of Observations: 7185
Number of Groups: 160
```

Notice that the formula for the random effects includes only the term for centered SES; as in a linear-model formula, a random intercept is implied unless it is explicitly excluded (by specifying `-1` in the `random` formula). By default, `lme` fits the model by *restricted maximum likelihood* (*REML*), which in effect corrects the maximum-likelihood estimator for degrees of freedom (see the complementary readings).

The output from the `summary` method for `lme` objects consists of several panels:

- The first panel gives the AIC (Akaike information criterion) and BIC (Bayesian information criterion), which can be used for model selection (see Section 4.5 of the text), along with the log of the maximized restricted likelihood.

- The next panel displays estimates of the variance and covariance parameters for the random effects, in the form of standard deviations and correlations. The term labelled `Residual` is the estimate of σ. Thus, $\widehat{\psi}_1 = 1.543$, $\widehat{\psi}_2 = 0.318$, $\widehat{\sigma} = 6.060$, and $\widehat{\psi}_{12} = 0.391 \times 1.543 \times 0.318 = 0.192$.

- The table of fixed effects is similar to output from `lm`; to interpret the coefficients in this table, refer to the hierarchical form of the model given in Equations 2 and 3, and to the Laird-Ware form of the LMM in Equation 4 (which orders the coefficients differently from the `lme` output). In particular:
  - The fixed-effect intercept coefficient $\widehat{\beta}_1 = 12.128$ represents an estimate of the average level of math achievement in public schools, which are the baseline category for the dummy regressor for `sector`.
  - Likewise, the coefficient labelled `sectorCatholic`, $\widehat{\beta}_4 = 1.227$, represents the difference between the average level of math achievement in Catholic schools and public schools.
  - The coefficient for `cses`, $\widehat{\beta}_3 = 2.945$, is the estimated average slope for SES in public schools, while the coefficient labelled `cses:sectorCatholic`, $\widehat{\beta}_6 = -1.643$, gives the difference in average slopes between Catholic and public schools. As we noted in our exploration of the data, the average level of math achievement is higher in Catholic than in public schools, and the average slope relating math achievement to students' SES is larger in public than in Catholic schools.
  - Given the parametrization of the model, the coefficient for `meanses`, $\widehat{\beta}_2 = 5.333$, represents the relationship of schools' average level of math achievement to their average level of SES
  - The coefficient for the interaction `meanses:cses`, $\widehat{\beta}_5 = 1.039$, gives the average change in the within-school SES slope associated with a one-unit increment in the school's mean SES. All of the coefficients are highly statistically significant.[7]

- The panel labelled `Correlation` gives the estimated sampling correlations among the fixed-effect coefficient estimates. These coefficient correlations are not usually of direct interest. Very large correlations, however, are indicative of an ill-conditioned model.

- Some information about the standardized within-group residuals ($\widehat{\varepsilon}_{ij}/\widehat{\sigma}$), the number of observations, and the number of groups, appears at the end of the output.

In addition to estimating and testing the fixed effects, it is of interest to determine whether there is evidence that the variances of the random effects in the model are different from 0. We can test hypotheses about the variances and covariances of random effects by deleting random-effects terms from the model and noting the change in the log of the maximized restricted likelihood, calculating log likelihood-ratio statistics. When LMMs are fit by REML, we must be careful, however, to compare models that are identical in their fixed effects.

For the current illustration, we may proceed as follows:

[7] See Section 2.2 for more careful hypothesis tests of fixed-effects coefficients in LMMs.

```
> bryk.lme.2 <- update(bryk.lme.1,
+     random = ~ 1 | school)  # omitting random effect of cses
> anova(bryk.lme.1, bryk.lme.2)

           Model df   AIC   BIC logLik   Test L.Ratio p-value
bryk.lme.1     1 10 46524 46592 -23252
bryk.lme.2     2  8 46521 46576 -23252 1 vs 2   1.124    0.57

> bryk.lme.3 <- update(bryk.lme.1,
+     random = ~ cses - 1 | school)  # omitting random intercept
> anova(bryk.lme.1, bryk.lme.3)

           Model df   AIC   BIC logLik   Test L.Ratio p-value
bryk.lme.1     1 10 46524 46592 -23252
bryk.lme.3     2  8 46740 46795 -23362 1 vs 2   220.6  <.0001
```

Each of these likelihood-ratio tests is on 2 degrees of freedom, because excluding one of the random effects removes not only its variance from the model but also its covariance with the other random effect. There is strong evidence, then, that the average level of math achievement (as represented by the intercept) varies from school to school, but not that the coefficient of SES varies, once differences between Catholic and public schools are taken into account, and the average level of SES in the schools is held constant.

A more careful formulation of these tests takes account of the fact that each null hypothesis places a variance (but not covariance) component on a boundary of the parameter space. Consequently, the null distribution of the LR test statistic is not simply chisquare with 2 degrees of freedom, but rather a mixture of chisquare distributions.[8] Moreover, it is reasonably simple to compute the corrected p-value:

```
> pval <- function(chisq, df){
+     (pchisq(chisq, df, lower.tail=FALSE) +
+             pchisq(chisq, df - 1, lower.tail=FALSE))/2
+ }
> pval(1.124, df=2)

[1] 0.4296

> pval(220.6, df=2)

[1] 6.59e-49
```

Here, therefore, the corrected p-values are similar to the uncorrected ones.

Model `bryk.lme.2`, fit above, omits the non-significant random effects for `cses`; the fixed-effects estimates are nearly identical to those for the initial model `bryk.lme.1`, which includes these random effects:

```
> summary(bryk.lme.2)
```

[8]See the complementary readings for discussion of this point.

```
Linear mixed-effects model fit by REML
 Data: Bryk
    AIC   BIC logLik
  46521 46576 -23252

Random effects:
 Formula: ~1 | school
        (Intercept) Residual
StdDev:       1.541    6.064

Fixed effects: mathach ~ meanses * cses + sector * cses
                     Value Std.Error   DF t-value p-value
(Intercept)         12.128    0.1992 7022   60.88  0.0000
meanses              5.337    0.3690  157   14.46  0.0000
cses                 2.942    0.1512 7022   19.46  0.0000
sectorCatholic       1.225    0.3061  157    4.00  0.0001
meanses:cses         1.044    0.2910 7022    3.59  0.0003
cses:sectorCatholic -1.642    0.2331 7022   -7.05  0.0000
 Correlation:
                    (Intr) meanss cses   sctrCt mnss:c
meanses              0.256
cses                 0.000  0.000
sectorCatholic      -0.699 -0.356  0.000
meanses:cses         0.000  0.000  0.295  0.000
cses:sectorCatholic  0.000  0.000 -0.696  0.000 -0.351

Standardized Within-Group Residuals:
     Min       Q1      Med       Q3      Max
-3.17012 -0.72488  0.01485  0.75424  2.96551

Number of Observations: 7185
Number of Groups: 160
```

This model is sufficiently simple, despite the interactions, to interpret the fixed effects from the estimated coefficients, but even here it is likely easier to visualize the model in effect plots (as discussed for linear models in Section 4.3.3 of the text). Our **effects** package has methods for mixed models fit by functions in the **nlme** and **lme4** packages. In the present example, we can use the `allEffects` function to graph the high-order fixed effects in the LMM we fit to the High School and Beyond Data – that is, the interactions between mean and centered SES and between mean SES and sector – producing Figure 6:

```
NULL
```

```
> library(effects)
> plot(allEffects(bryk.lme.2), rug=FALSE)
```

It is clear from these graphs that the impact of a student's SES on math achievement rises as the mean level of math achievement in his or her school rises, and is larger in public schools than in Catholic schools.

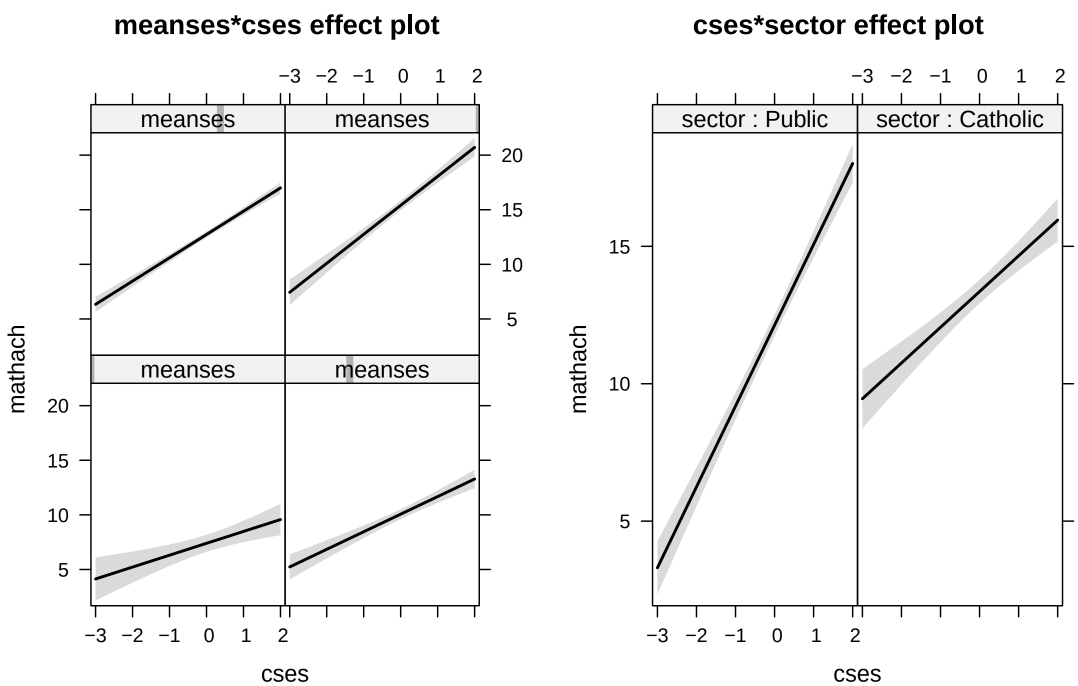

Figure 6: Effect displays for the high-order terms in the LMM fit to the High School and Beyond data, `bryk.lme.2`.

2.1.3 Fitting a Hierarchical Linear Model with `lmer`

We can perform the same analysis employing `lmer` in the **lme4** package. For example, to fit the initial hiearchical model considered in the previous section:

```
> library(lme4)
> bryk.lmer.1 <- lmer(mathach ~ meanses*cses + sector*cses + (cses | school),
+     data=Bryk)
> summary(bryk.lmer.1)

Linear mixed model fit by REML ['lmerMod']
Formula: mathach ~ meanses * cses + sector * cses + (cses | school)
   Data: Bryk

REML criterion at convergence: 46504

Scaled residuals:
   Min     1Q Median     3Q    Max
-3.159 -0.723  0.017  0.754  2.958

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 school   (Intercept)  2.380   1.543
          cses         0.101   0.318    0.39
 Residual             36.721   6.060
```

```
Number of obs: 7185, groups:  school, 160

Fixed effects:
                     Estimate Std. Error t value
(Intercept)            12.128      0.199    60.9
meanses                 5.333      0.369    14.4
cses                    2.945      0.156    18.9
sectorCatholic          1.227      0.306     4.0
meanses:cses            1.039      0.299     3.5
cses:sectorCatholic    -1.643      0.240    -6.9

Correlation of Fixed Effects:
            (Intr) meanss cses   sctrCt mnss:c
meanses      0.256
cses         0.075  0.019
sectorCthlc -0.699 -0.356 -0.053
meanses:css  0.019  0.074  0.293 -0.026
css:sctrCth -0.052 -0.027 -0.696  0.077 -0.351
```

The estimates of the fixed effects and variance/covariance components are the same as those obtained from `lme` (see page 15), but the specification of the model is slightly different: Rather than using a `random` argument as in `lme`, the random effects in `lmer` are given directly in the model formula, enclosed in parentheses; as in `lme`, a random intercept is implied if it is not explicitly removed. An important difference between `lme` and `lmer`, however, is that `lmer` can accommodate *crossed* random effects, while `lme` cannot: Suppose, for example, that we were interested in teacher effects on students' achievement. Each student in a high school has several teachers, and so students would not be strictly nested within teachers.

A subtle difference between the `lme` and `lmer` output is that the former includes p-values for the Wald t-tests of the estimated coefficients while the latter does not. The p-values in `lmer` are suppressed because the Wald tests can be inaccurate. We address this issue in Section 2.2.

As in the previous section, let us proceed to remove the random slopes from the model, comparing the resulting model to the initial model by a likelihood-ratio text:

```
> bryk.lmer.2 <- lmer(mathach ~ meanses*cses + sector*cses + (1 | school),
+     data=Bryk)
> anova(bryk.lmer.1, bryk.lmer.2)

Data: Bryk
Models:
bryk.lmer.2: mathach ~ meanses * cses + sector * cses + (1 | school)
bryk.lmer.1: mathach ~ meanses * cses + sector * cses + (cses | school)
            Df   AIC   BIC logLik deviance Chisq Chi Df Pr(>Chisq)
bryk.lmer.2  8 46513 46568 -23249    46497
bryk.lmer.1 10 46516 46585 -23248    46496     1      2       0.61
```

Notice that, out of an abundance of caution, `anova` refits the models using ML rather than REML, because LR ratio tests of models that differ in their *fixed* effects are inappropriate. In our case, however, the models compared have identical fixed effects and differ only in the *random* effects. A likelihood-ratio test is therefore appropriate even if the models are fit by REML. We can obtain this test by specifying the argument `refit=FALSE`:

```
> anova(bryk.lmer.1, bryk.lmer.2, refit=FALSE)

Data: Bryk
Models:
bryk.lmer.2: mathach ~ meanses * cses + sector * cses + (1 | school)
bryk.lmer.1: mathach ~ meanses * cses + sector * cses + (cses | school)
            Df   AIC   BIC logLik deviance Chisq Chi Df Pr(>Chisq)
bryk.lmer.2  8 46521 46576 -23252    46505
bryk.lmer.1 10 46524 46592 -23252    46504  1.12      2       0.57
```

The results are identical to those using `lme`.

2.2 Wald Tests for Linear Mixed Models

As we mentioned, it is inappropriate to perform likelihood-ratio tests for fixed effects when a LMM is fit by REML. Though it is sometimes recommended that ML be used instead to obtain LR tests of fixed effects, ML estimates can be substantially biased when there are relatively few higher-level units. Wald tests can be performed, however, for the fixed effects in a LMM estimated by REML, but as we also mentioned, Wald tests obtained for individual coefficients by dividing estimated fixed effects by their standard errors can be inaccurate. The same is true of more complex Wald tests on several degrees of freedom — for example, F-tests for terms in a linear mixed model.

One approach to obtaining more accurate inferences in LMMs fit by REML is to adjust the estimated covariance matrix of the fixed effects to reduce the typically downward bias of the coefficient standard errors, as suggested by Kenward and Roger (1997), and to adjust degrees of freedom for t and F tests (applying a method introduced by Satterthwaite, 1946). These adjustments are available for linear mixed models fit by `lmer` in the `Anova` and `linearhypothesis` functions in the **car** package, employing infrastructure from the **pbkrtest** package. For example,

```
> library(car)
> Anova(bryk.lmer.2, test="F")

Analysis of Deviance Table (Type II Wald F tests with Kenward-Roger df)

Response: mathach
                 F Df Df.res  Pr(>F)
meanses      209.2  1    156 < 2e-16
cses         409.4  1   7023 < 2e-16
sector        16.0  1    154 9.8e-05
meanses:cses  12.9  1   7023 0.00033
cses:sector   49.6  1   7023 2.0e-12
```

In this case, with many schools and a moderate number of students within each school, the KR tests are essentially the same as Wald chisquare tests using the naively computed covariance matrix for the fixed effects:

```
> Anova(bryk.lmer.2)

Analysis of Deviance Table (Type II Wald chisquare tests)

Response: mathach
```

```
              Chisq Df Pr(>Chisq)
meanses       209.2  1    < 2e-16
cses          409.4  1    < 2e-16
sector         16.0  1    6.3e-05
meanses:cses   12.9  1    0.00033
cses:sector    49.6  1    1.9e-12
```

2.3 An Illustrative Application to Longitudinal Data

To illustrate the use of linear mixed models in longitudinal research, we draw on data described by Davis et al. (2005) on the exercise histories of 138 teenaged girls hospitalized for eating disorders and of 93 comparable "control" subjects.[9] The data are in the data frame `Blackmore` in the **car** package:

```
> head(Blackmore, 20)

   subject   age exercise   group
1      100  8.00     2.71 patient
2      100 10.00     1.94 patient
3      100 12.00     2.36 patient
4      100 14.00     1.54 patient
5      100 15.92     8.63 patient
6      101  8.00     0.14 patient
7      101 10.00     0.14 patient
8      101 12.00     0.00 patient
9      101 14.00     0.00 patient
10     101 16.67     5.08 patient
11     102  8.00     0.92 patient
12     102 10.00     1.82 patient
13     102 12.00     4.75 patient
15     102 15.08    24.72 patient
16     103  8.00     1.04 patient
17     103 10.00     2.90 patient
18     103 12.00     2.65 patient
20     103 14.08     6.86 patient
21     104  8.00     2.75 patient
22     104 10.00     6.62 patient
```

The variables are:

- `subject`: an identification code; there are several observations for each subject, but because the girls were hospitalized at different ages, the number of observations and the age at the last observation vary.
- `age`: the subject's age in years at the time of observation; all but the last observation for each subject were collected retrospectively at intervals of 2 years, starting at age 8.
- `exercise`: the amount of exercise in which the subject engaged, expressed as estimated hours per week.

[9]These data were generously made available to me by Elizabeth Blackmore and Caroline Davis of York University.

- `group`: a factor indicating whether the subject is a `"patient"` or a `"control"`.[10]

2.3.1 Examining the Data

Initial examination of the data suggested that it is advantageous to take the log of `exercise`: Doing so makes the `exercise` distribution for both groups of subjects more symmetric and linearizes the relationship of `exercise` to `age`.[11] Because there are some 0 values of `exercise`, we use "started" logs in the analysis reported below (see Section 3.4 of the text on transforming data), adding 5 minutes (5/60 of an hour) to each value of `exercise` prior to taking logs (and using logs to the base 2 for interpretability):

```
> Blackmore$log.exercise <- log(Blackmore$exercise + 5/60, 2)
```

As in the analysis of the math-achievement data in the preceding section, we begin by sampling 20 subjects from each of the patient and control groups, plotting `log.exercise` against `age` for each `subject`:

```
> pat <- with(Blackmore, sample(unique(subject[group=="patient"]), 20))
> Pat.20 <- groupedData(log.exercise ~ age | subject,
+     data=Blackmore[is.element(Blackmore$subject, pat),])
> con <- with(Blackmore, sample(unique(subject[group=="control"]), 20))
> Con.20 <- groupedData(log.exercise ~ age | subject,
+     data=Blackmore[is.element(Blackmore$subject, con),])
> print(plot(Con.20, main="Control Subjects",
+         xlab="Age", ylab="log2 Exercise",
+         ylim=1.2*range(Con.20$log.exercise, Pat.20$log.exercise),
+         layout=c(5, 4), aspect=1.0),
+     position=c(0, 0, 0.5, 1), more=TRUE)
> print(plot(Pat.20, main="Patients",
+         xlab="Age", ylab="log2 Exercise",
+         ylim=1.2*range(Con.20$log.exercise, Pat.20$log.exercise),
+         layout=c(5, 4), aspect=1.0),
+     position=c(0.5, 0, 1, 1))
```

The graphs appear in Figure 7.

- Each Trellis plot is constructed by using the default `plot` method for grouped-data objects. Grouped-data objects, provided by the **nlme** package, are enhanced data frames, incorporating a model formula that gives information about the structure of the data. In this instance, the formula `log.exercise ~ age | subject`, read as "`log.exercise` depends on `age` given `subject`," indicates that `log.exercise` is the response variable, `age` is the principal within-subject covariate (actually, in this application, it is the *only* within-subject covariate), and the data are grouped by `subject`.

- To make the two plots comparable, we have exercised direct control over the scale of the vertical axis (set to slightly larger than the range of the combined log-exercise values), the layout of the plot (`5` columns, `4` rows),[12] and the aspect ratio of the plot (the ratio of the

[10] To avoid the possibility of confusion, we point out that the variable `group` represents groups of independent patients and control subjects, and is *not* a factor defining clusters. Clusters in this longitudinal data set are defined by the variable `subject`.

[11] We invite the read to examine the distribution of the `exercise` variable, before and after log-transformation.

[12] Notice the unusual ordering in specifying the layout — columns first, then rows.

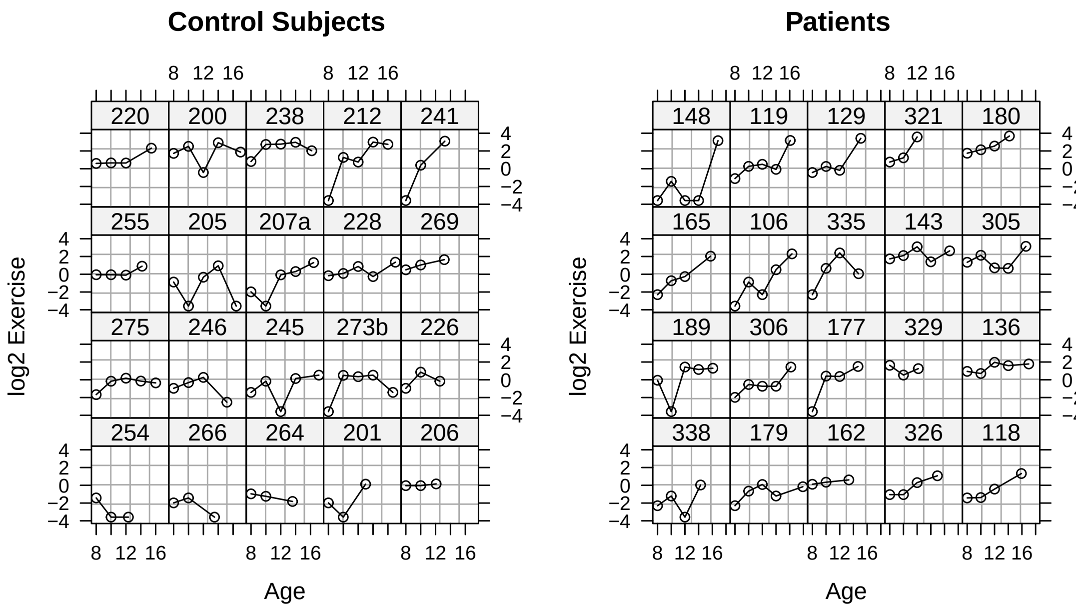

Figure 7: $\log_2$ exercise by age for 20 randomly selected patients and 20 randomly selected control subjects.

vertical to the horizontal size of the plotting region in each panel, set here to `1.0`).

- The `print` method for Trellis objects, normally automatically invoked when the returned object is not assigned to a variable, simply plots the object on the active graphics device. So as to print both plots on the same "page," we have instead called `print` explicitly, using the `position` argument to place each graph on the page. The form of this argument is `c(xmin, ymin, xmax, ymax)`, with horizontal (`x`) and vertical (`y`) coordinates running from `0, 0` (the lower-left corner of the page) to `1, 1` (the upper-right corner). The argument `more=TRUE` in the first call to `print` indicates that the graphics page is not yet complete.

There are few observations for each subject, and in many instances, no strong within-subject pattern. Nevertheless, it appears as if the general level of exercise is higher among the patients than among the controls. As well, the trend for exercise to increase with age appears stronger and more consistent for the patients than for the controls.

We pursue these impressions by fitting regressions of `log.exercise` on `age` for each subject. Because of the small number of observations per subject, we should not expect very good estimates of the within-subject regression coefficients. Indeed, one of the advantages of mixed models is that they can provide improved estimates of the within-subject coefficients (the random effects plus the fixed effects) by pooling information across subjects.[13]

```
> pat.list <- lmList(log.exercise ~ I(age - 8) | subject,
+     subset = group=="patient", data=Blackmore)
> con.list <- lmList(log.exercise ~ I(age - 8) | subject,
+     subset = group=="control", data=Blackmore)
> pat.coef <- coef(pat.list)
> con.coef <- coef(con.list)
> old <- par(mfrow=c(1, 2))
> boxplot(pat.coef[,1], con.coef[,1], main="Intercepts",
+     names=c("Patients", "Controls"))
> boxplot(pat.coef[,2], con.coef[,2], main="Slopes",
+     names=c("Patients", "Controls"))
> par(old)
```

Boxplots of the within-subjects regression coefficients are shown in Figure 8. We changed the origin of `age` to `8` years, which is the initial observation for each subject, so the intercept represents level of exercise at the start of the study. As expected, there is a great deal of variation in both the intercepts and the slopes. The median intercepts are similar for patients and controls, but there is somewhat more variation among patients. The slopes are higher on average for patients than for controls, for whom the median slope is close to 0.

2.3.2 Fitting a Mixed Model with Autocorrelated Errors

We proceed to fit a LMM to the data, including fixed effects for `age` (again, with an origin of `8`), `group`, and their interaction, and random intercepts and slopes:

```
> bm.lme.1 <- lme(log.exercise ~ I(age - 8)*group,
+                         random = ~ I(age - 8) | subject, data=Blackmore)
> summary(bm.lme.1)
```

[13]Pooled estimates of the random effects provide so-called *best-linear-unbiased predictors* (or *BLUPs*) of the regression coefficients for individual subjects. See `help(predict.lme)` and the complementary readings.

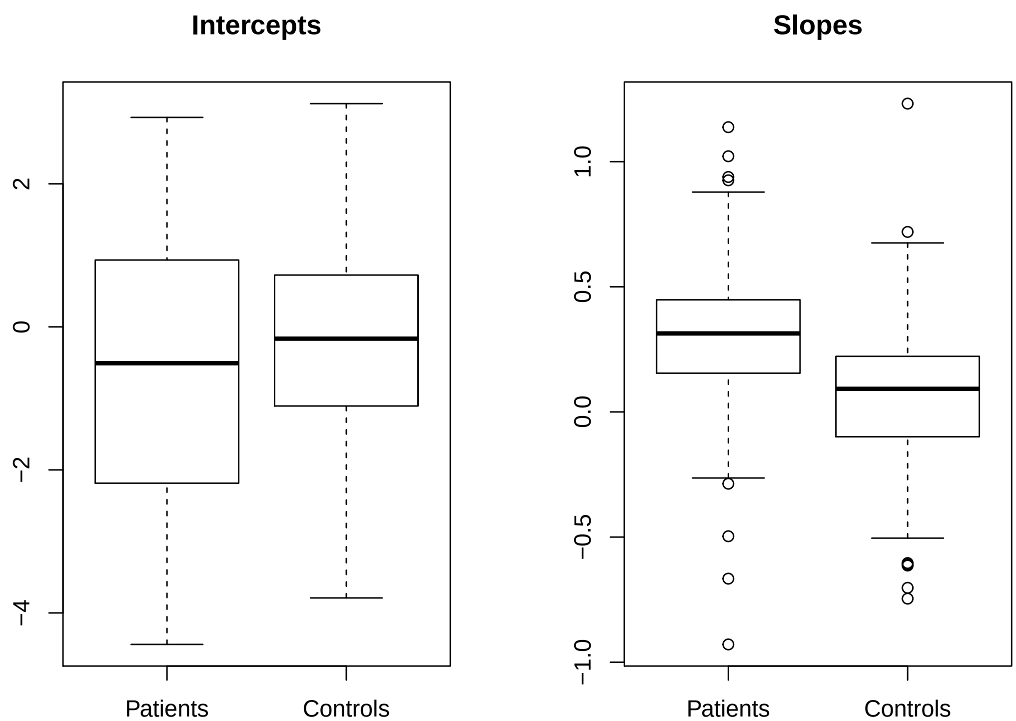

Figure 8: Coefficients for the within-subject regressions of $\log_2$ exercise on age, for patients and control subjects.

```
Linear mixed-effects model fit by REML
 Data: Blackmore
   AIC  BIC logLik
  3630 3669  -1807

Random effects:
 Formula: ~I(age - 8) | subject
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev Corr
(Intercept) 1.4436 (Intr)
I(age - 8)  0.1648 -0.281
Residual    1.2441

Fixed effects: log.exercise ~ I(age - 8) * group
                          Value Std.Error  DF t-value p-value
(Intercept)             -0.2760   0.18237 712  -1.514  0.1306
I(age - 8)               0.0640   0.03136 712   2.041  0.0416
grouppatient            -0.3540   0.23529 229  -1.504  0.1338
I(age - 8):grouppatient  0.2399   0.03941 712   6.087  0.0000
 Correlation:
                        (Intr) I(g-8) grpptn
I(age - 8)              -0.489
grouppatient            -0.775  0.379
I(age - 8):grouppatient  0.389 -0.796 -0.489

Standardized Within-Group Residuals:
    Min      Q1     Med      Q3     Max
-2.7349 -0.4245  0.1228  0.5280  2.6362

Number of Observations: 945
Number of Groups: 231
```

Examining the naive t-tests, there is a small, and marginally statistically significant, average `age` trend in the control group (represented by the fixed-effect coefficient for `age - 8`), and a highly significant interaction of `age` with `group`, reflecting a much steeper average trend in the patient group. The small and nonsignificant coefficient for `group` indicates similar age-8 intercepts for the two groups.[14]

We test whether the random intercepts and slopes are necessary, omitting each in turn from the model and calculating a likelihood-ratio statistic, contrasting the refitted model with the original model:

```
> bm.lme.2 <- update(bm.lme.1, random = ~ 1 | subject)
> anova(bm.lme.1, bm.lme.2)  # test for random slopes

         Model df  AIC  BIC logLik   Test L.Ratio p-value
bm.lme.1     1  8 3630 3669  -1807
bm.lme.2     2  6 3644 3673  -1816 1 vs 2   18.12  0.0001
```

[14] Unfortunately, the **pbkrtest** package will not provide corrected standard errors and degrees of freedom for models fit by `lme` (as opposed to `lmer`).

```
> bm.lme.3 <- update(bm.lme.1, random = ~ I(age - 8) - 1 | subject)
> anova(bm.lme.1, bm.lme.3)  # test for random intercepts

         Model df  AIC  BIC logLik   Test L.Ratio p-value
bm.lme.1     1  8 3630 3669  -1807
bm.lme.3     2  6 3834 3863  -1911 1 vs 2   207.9  <.0001
```

The tests are highly statistically significant, particularly for random intercepts, suggesting that both random intercepts and random slopes are required.

Let us next consider the possibility that the within-subject errors (the ε_{ij}s in the mixed model of Equation 1 on page 2) are autocorrelated — as may well be the case, because the observations are taken longitudinally on the same subjects. The `lme` function incorporates a flexible mechanism for specifying error-correlation structures, and supplies constructor functions for several such structures.[15] Most of these correlation structures, however, are appropriate only for equally spaced observations. An exception is the `corCAR1` function, which permits us to fit a continuous first-order autoregressive process in the errors. Suppose that ε_{it} and $\varepsilon_{i,t+s}$ are errors for subject i separated by s units of time, where s need not be an integer; then, according to the continuous first-order autoregressive model, the correlation between these two errors is $\rho(s) = \phi^{|s|}$ where $0 \leq \phi < 1$. This appears a reasonable specification in the current context, where there are at most $n_i = 5$ observations per subject.

Fitting the model with CAR1 errors to the data produces a convergence failure:

```
> bm.lme.4 <- update(bm.lme.1, correlation = corCAR1(form = ~ age | subject))

Error in lme.formula(fixed = log.exercise ~ I(age - 8) * group, data = Blackmore,  :
  nlminb problem, convergence error code = 1
  message = iteration limit reached without convergence (10)
```

The correlation structure is given in the `correlation` argument to `lme` (here as a call to `corCAR1`); the `form` argument to `corCAR1` is a one-sided formula defining the time dimension (here, `age`) and the group structure (`subject`). With so few observations within each subject, it is difficult to separate the estimated correlation of the errors from the correlations among the observations induced by clustering, as captured by subject-varying intercepts and slopes. This kind of convergence problem is a common occurrence in mixed-effects modeling.

We will therefore fit two additional models to the data, each including *either* random intercepts or random slopes (but not *both*) along with autocorrelated errors:

```
> bm.lme.5 <- update(bm.lme.1, random = ~ 1 | subject,
+     correlation = corCAR1(form = ~ age |subject)) # random intercepts (not slopes)
> bm.lme.6 <- update(bm.lme.1, random = ~ I(age - 8) - 1 | subject,
+     correlation = corCAR1(form = ~ age |subject)) # random slopes (not intercepts)
```

These models and our initial model without autocorrelated errors (`bm.lme.1`) are not properly nested for likelihood-ratio tests — indeed `bm.lme.5` and `bm.lme6` have the same number of parameters — but we can examine the maximimzed restricted log-likilihood under the models along with the AIC and BIC model-selection criteria:

[15] A similar mechanism is provided for modeling non-constant error variance, via the `weights` argument to `lme`. See the documentation for `lme` for details. In contrast, the `lmer` function in the **lme4** package does not accommodate autocorrelated errors, which is why we used `lme` for this example.

```
> table <- matrix(0, 3, 3)
> table[, 1] <- c(logLik(bm.lme.1), logLik(bm.lme.5), logLik(bm.lme.6))
> table[, 2] <- c(BIC(bm.lme.1), BIC(bm.lme.5), BIC(bm.lme.6))
> table[, 3] <- c(AIC(bm.lme.1), AIC(bm.lme.5), AIC(bm.lme.6))
> colnames(table) <- c("logLik", "BIC", "AIC")
> rownames(table) <- c("bm.lme.1", "bm.lme.5", "bm.lme.6")
> table
```

```
         logLik  BIC  AIC
bm.lme.1  -1807 3669 3630
bm.lme.5  -1795 3639 3605
bm.lme.6  -1803 3654 3620
```

All of these criteria point to model `bm.lme.5`, with random intercepts, a fixed age slope (within patient/control groups), and autocorrelated errors.

Although we expended some effort in modeling the random effects, the estimates of the fixed effects, and their standard errors, do not depend critically on the random-effect specification of the model, also a common occurrence:

```
> compareCoefs(bm.lme.1, bm.lme.5, bm.lme.6)
```

```
Call:
1: lme.formula(fixed = log.exercise ~ I(age - 8) * group, data = Blackmore,
  random = ~I(age - 8) | subject)
2: lme.formula(fixed = log.exercise ~ I(age - 8) * group, data = Blackmore,
  random = ~1 | subject, correlation = corCAR1(form = ~age | subject))
3: lme.formula(fixed = log.exercise ~ I(age - 8) * group, data = Blackmore,
  random = ~I(age - 8) - 1 | subject, correlation = corCAR1(form = ~age |
  subject))
                        Est. 1    SE 1  Est. 2    SE 2  Est. 3    SE 3
(Intercept)            -0.2760  0.1824 -0.3070  0.1895 -0.3178  0.1935
I(age - 8)              0.0640  0.0314  0.0728  0.0317  0.0742  0.0365
grouppatient           -0.3540  0.2353 -0.2838  0.2447 -0.2487  0.2500
I(age - 8):grouppatient  0.2399  0.0394  0.2274  0.0397  0.2264  0.0460
```

The summary for model `bm.lme.5` is as follows:

```
> summary(bm.lme.5)
```

```
Linear mixed-effects model fit by REML
 Data: Blackmore
   AIC  BIC logLik
  3605 3639  -1795

Random effects:
 Formula: ~1 | subject
        (Intercept) Residual
StdDev:        1.15    1.529
```

```
Correlation Structure: Continuous AR(1)
 Formula: ~age | subject
 Parameter estimate(s):
   Phi
0.6312
Fixed effects: log.exercise ~ I(age - 8) * group
                           Value Std.Error  DF t-value p-value
(Intercept)             -0.30697   0.18950 712  -1.620  0.1057
I(age - 8)               0.07278   0.03168 712   2.297  0.0219
grouppatient            -0.28383   0.24467 229  -1.160  0.2472
I(age - 8):grouppatient  0.22744   0.03974 712   5.723  0.0000
 Correlation:
                        (Intr) I(g-8) grrpptn
I(age - 8)              -0.553
grouppatient            -0.775  0.428
I(age - 8):grouppatient  0.441 -0.797 -0.556

Standardized Within-Group Residuals:
    Min      Q1     Med      Q3     Max
-2.9431 -0.4640  0.1732  0.5869  2.0220

Number of Observations: 945
Number of Groups: 231
```

There is, therefore, a moderately large estimated error autocorrelation, $\widehat{\phi} = .631$.

To get a more concrete sense of the fixed effects, using model `bm.lme.5` (which includes autocorrelated errors and random intercepts, but not random slopes), we employ the `predict` method for `lme` objects to calculate fitted values for patients and controls across the range of ages (`8` to `18`) represented in the data:

```
> pdata <- expand.grid(age=seq(8, 18, by=2), group=c("patient", "control"))
> pdata$log.exercise <- predict(bm.lme.5, pdata, level=0)
> pdata$exercise <- (2^pdata$log.exercise) - 5/60
> pdata

   age   group log.exercise exercise
1    8 patient    -0.590801   0.5806
2   10 patient     0.009641   0.9234
3   12 patient     0.610082   1.4430
4   14 patient     1.210523   2.2309
5   16 patient     1.810964   3.4254
6   18 patient     2.411405   5.2366
7    8 control    -0.306970   0.7250
8   10 control    -0.161409   0.8108
9   12 control    -0.015847   0.9057
10  14 control     0.129715   1.0107
11  16 control     0.275277   1.1269
12  18 control     0.420838   1.2554
```

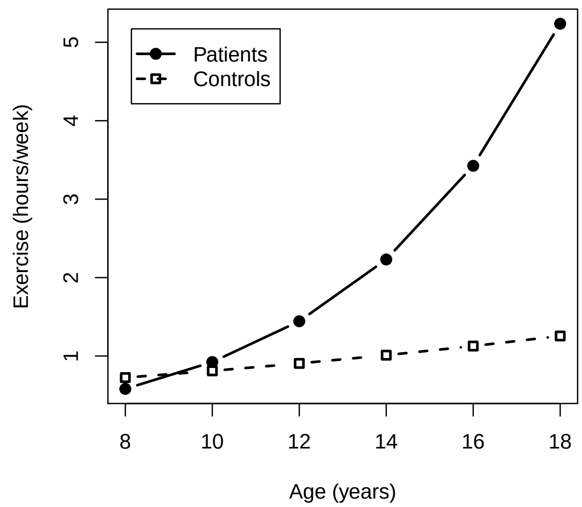

Figure 9: Fitted values representing estimated fixed effects of `group`, `age`, and their interaction.

Specifying `level=0` in the call to `predict` produces estimates of the fixed effects. The expression `(2^pdata$log.exercise) - 5/60` translates the fitted values of exercise from the $\log_2$ scale back to hours/week.

Finally, we plot the fitted values (Figure 9):

```
> plot(pdata$age, pdata$exercise, type="n",
+     xlab="Age (years)", ylab="Exercise (hours/week)")
> points(pdata$age[1:6], pdata$exercise[1:6], type="b", pch=19, lwd=2)
> points(pdata$age[7:12], pdata$exercise[7:12], type="b", pch=22, lty=2, lwd=2)
> legend("topleft", c("Patients", "Controls"), pch=c(19, 22),
+     lty=c(1,2), lwd=2, inset=0.05)
```

Essentially the same graph (Figure 10) can be constructed by the **effects** package, with the added feature of confidence intervals for the estimated effects:

```
> plot(Effect(c("age", "group"), bm.lme.5, xlevels=list(age=seq(8, 18, by=2)),
+         transformation=list(link=function(x) log2(x + 5/60),
+             inverse=function(x) 2^x - 5/60)),
+      multiline=TRUE, ci.style="bars",
+      xlab="Age (years)", ylab="Exercise (hours/week)",
+      rescale.axis=FALSE, rug=FALSE, colors=c("black", "black"),
+      key.args=list(x = 0.20, y = 0.75, corner = c(0, 0), padding.text = 1.25),
+     main="")
```

It is apparent that the two groups of subjects have similar average levels of exercise at `age` 8, but that thereafter the level of exercise increases much more rapidly for the patient group than for the controls.

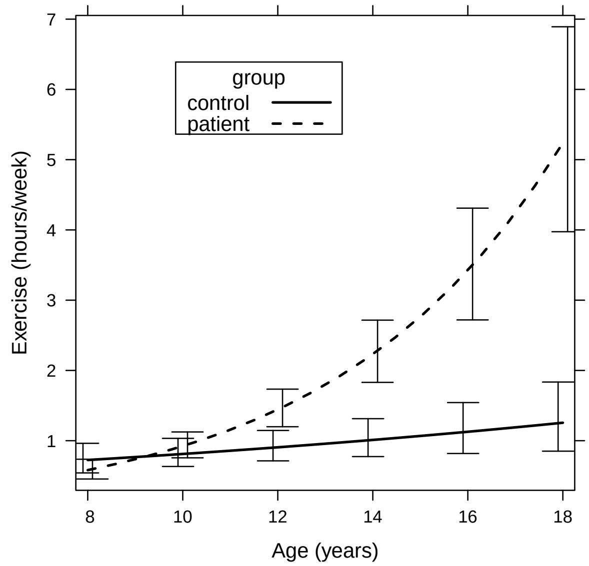

Figure 10: Plot produced using the `Effect` function in the **effects** package.

3 Generalized Linear Mixed Models

Generalized linear mixed models (*GLMMs*) bear the same relationship to LMMs that GLMs bear to linear models (see Chapters 4 and 5 of the text). GLMMs add random effects to the linear predictor of a GLM and express the expected value of the response conditional on the random effects: The link function $g(\cdot)$ is the same as in generalized linear models. In the GLMM, the conditional distribution of y_{ij}, the response for observation j in group i, given the random effects, is (most straightforwardly) a member of an exponential family, with mean μ_{ij}, variance

$$\mathrm{Var}(y_{ij}) = \phi V(\mu_{ij})\lambda_{ij}$$

and covariances

$$\mathrm{Cov}(y_{ij}, y_{ij'}) = \phi\sqrt{V(\mu_{ij})}\sqrt{V(\mu_{ij'})}\lambda_{ijj'}$$

where ϕ is a dispersion parameter and the function $V(\mu_{ij})$ depends on the distributional family to which Y belongs. Recall, for example, that in the binomial and Poisson families the dispersion is fixed to 1, and that in the Gaussian family $V(\mu) = 1$. Alternatively, for quasi-likelihood estimation, $V(\cdot)$ can be given directly.[16]

[16] As in the generalized linear model, see Section 5.10.3 of the text.

The GLMM may therefore be written as

$$
\begin{aligned}
\eta_{ij} &= \beta_1 + \beta_2 x_{2ij} + \cdots + \beta_p x_{pij} + b_{1i} z_{1ij} + \cdots + b_{qi} z_{qij} \\
g(\mu_{ij}) &= E(y_{ij}|b_{1i}, \ldots, b_{qi}) = \eta_{ij} \\
b_{ki} &\sim N(0, \psi_k^2), \mathrm{Cov}(b_{ki}, b_{k'i}) = \psi_{kk'} \\
b_{ki}, b_{k'i'} &\text{ are independent for } i \neq i' \\
\mathrm{Var}(y_{ij}) &= \phi V(\mu_{ij}) \lambda_{ij} \\
\mathrm{Cov}(y_{ij}, y_{ij'}) &= \phi \sqrt{V(\mu_{ij})} \sqrt{V(\mu_{ij'})} \lambda_{ijj'} \\
y_{ij}, y_{ij'} &\text{ are independent for } i \neq i'
\end{aligned}
$$

where η_{ij} is the linear predictor for observation j in cluster i; the fixed-effect coefficients (βs), random-effect coefficients (bs), fixed-effect regressors (xs), and random-effect regressors (zs) are defined as in the LMM.[17]

In matrix form, the GLMM is

$$
\begin{aligned}
\boldsymbol{\eta}_i &= \mathbf{X}_i \boldsymbol{\beta} + \mathbf{Z}_i \mathbf{b}_i && (5) \\
g(\boldsymbol{\mu}_i) &= g[E(\mathbf{y}_i|\mathbf{b}_i)] = \boldsymbol{\eta}_i \\
\mathbf{b}_i &\sim \mathbf{N}_q(\mathbf{0}, \boldsymbol{\Psi}) \\
\mathbf{b}_i, \mathbf{b}_{i'} &\text{ are independent for } i \neq i' \\
E(\mathbf{y}_i|\mathbf{b}_i) &= \boldsymbol{\mu}_i && (6) \\
V(\mathbf{y}_i|\mathbf{b}_i) &= \phi V^{1/2}(\boldsymbol{\mu}_i) \boldsymbol{\Lambda} V^{1/2}(\boldsymbol{\mu}_i) && (7) \\
\mathbf{y}_i, \mathbf{y}_{i'} &\text{ are independent for } i \neq i'
\end{aligned}
$$

where

- $\mathbf{y}_i$ is the $n_i \times 1$ response vector for observations in the ith of m groups;
- $\boldsymbol{\mu}_i$ is the $n_i \times 1$ expectation vector for the response, conditional on the random effects;
- $\boldsymbol{\eta}_i$ is the $n_i \times 1$ linear predictor for the elements of the response vector;
- $g(\cdot)$ is the link function, transforming the conditional expected response to the linear predictor;
- $\mathbf{X}_i$ is the $n_i \times p$ model matrix for the fixed effects of observations in group i;
- $\boldsymbol{\beta}$ is the $p \times 1$ vector of fixed-effect coefficients;
- $\mathbf{Z}_i$ is the $n_i \times q$ model matrix for the random effects of observations in group i;
- $\mathbf{b}_i$ is the $q \times 1$ vector of random-effect coefficients for group i;
- $\boldsymbol{\Psi}$ is the $q \times q$ covariance matrix of the random effects;
- $\boldsymbol{\Lambda}_i$ is $n_i \times n_i$ and expresses the dependence structure for the conditional distribution of the response within each group—for example, if the observations are sampled independently in each group, $\boldsymbol{\Lambda}_i = \mathbf{I}_{n_i}$;[18]

[17] The `glmer` function in the **lme4** package, which we will use to fit GLMMs, is somewhat more restrictive, setting $\lambda_{kk} = 1$ and $\lambda_{kk'} = 0$.

[18] As mentioned, this restriction is imposed by the `glmer` function in the **lme4** package. See footnote 17.

- $V^{1/2}(\boldsymbol{\mu}_i) \equiv \text{diag}[\sqrt{V(\mu_{ij})}]$; and
- ϕ is the dispersion parameter.

3.1 Example: Migraine Headaches

In an effort to reduce the severity and frequency of migraine headaches through the use of biofeedback training, Tammy Kostecki-Dillon collected longitudinal data on migraine-headache sufferers.[19] The 133 patients who participated in the study were each given four weekly sessions of biofeedback training. The patients were asked to keep daily logs of their headaches for a period of 30 days prior to training, during training, and post-training, to 100 days after training began. Compliance with these instructions was low, and there is therefore quite a bit of missing data; for example, only 55 patients kept a log prior to training. On average, subjects recorded information on 31 days, with the number of days ranging from 7 to 121. Subjects were divided into three self-selected groups: those who discontinued their migraine medication during the training and post-training phase of the study; those who continued their medication, but at a reduced dose; and those who continued their medication at the previous dose.

We will use a binomial GLMM—specifically, a binary logit mixed-effects model—to analyze the incidence of headaches during the period of the study. Examination of the data suggested that the incidence of headaches was invariant during the pre-training phase of the study, increased (as was expected by the investigator) at the start of training, and then declined at a decreasing rate. We decided to fit a linear trend prior to the start of training (before time 0), possibly to capture a trend that we failed to detect in our exploration of the data, and to transform time at day 1 and later (which, for simplicity, we term "time post-treatment") by taking the square-root.[20] In addition to the intercept, representing the level of headache incidence at the end of the pre-training period, we include a dummy regressor coded 1 post-treatment, and 0 pre-treatment, to capture the anticipated increase in headache incidence at the start of training; dummy regressors for levels of medication; and interactions between medication and treatment, and between medication and the pre- and post-treatment time trends. Thus, the fixed-effects part of the model is

$$\begin{aligned}\text{logit}(\pi_{ij}) = \beta_1 &+ \beta_2 m_{1i} + \beta_3 m_{2i} + \beta_4 p_{ij} + \beta_5 t_{0ij} + \beta_6 \sqrt{t_{1ij}} \\ &+ \beta_7 m_{1i} p_{ij} + \beta_8 m_{2i} p_{ij} + \beta_9 m_{1i} t_{0ij} + \beta_{10} m_{2i} t_{0ij} \\ &+ \beta_{11} m_{1i} \sqrt{t_{1ij}} + \beta_{12} m_{2i} \sqrt{t_{1ij}}\end{aligned}$$

where

- π_{ij} is the probability of a headache for individual $i = 1, \ldots, 133$, on occasion $j = 1, \ldots, n_i$;
- m_{1i} is a dummy regressor coded 1 if individual i continued taking migraine medication at a reduced dose post-treatment, and m_{2i} is a dummy regressor coded 1 if individual i continued taking medication at its previous dose post-treatment;
- p_{ij} is a dummy regressor coded 1 post-treatment (i.e., after time 0) and 0 pre-treatment;
- t_{0ij} is time (in days) pre-treatment, running from -29 through 0, and coded 0 after treatment began; and

[19] The data are described by Kostecki-Dillon, Monette, and Wong (1999) and were generously made available to me by Georges Monette, who performed the original data analysis. The analysis reported here is similar to his.

[20] The original analysis of the data by Georges Monette used regression splines for time-trends, with results generally similar to those reported here.

- t_{1ij} is time (in days) post-treatment, running from 1 through 99, and coded 0 pre-treatment.

We will include patient random effects for the intercept (i.e., the level of headache incidence pre-treatment), for the post-treatment dummy regressor, and for the pre- and post-treatment time-trend regressors.

The data for this example are in the `KosteckiDillon` data frame in the **car** package. We begin with a bit of data-management:

```
> KosteckiDillon$treatment <- factor(with(KosteckiDillon,
+     ifelse(time > 0, "yes", "no")))
> KosteckiDillon$pretreat <- with(KosteckiDillon, ifelse(time > 0, 0, time))
> KosteckiDillon$posttreat <- with(KosteckiDillon, ifelse(time > 0, time, 0))
> head(KosteckiDillon, 10)
```

```
   id time dos hatype age airq medication headache    sex treatment pretreat
1   1  -11 753   Aura  30    9 continuing      yes female        no      -11
2   1  -10 754   Aura  30    7 continuing      yes female        no      -10
3   1   -9 755   Aura  30   10 continuing      yes female        no       -9
4   1   -8 756   Aura  30   13 continuing      yes female        no       -8
5   1   -7 757   Aura  30   18 continuing      yes female        no       -7
6   1   -6 758   Aura  30   19 continuing      yes female        no       -6
7   1   -5 759   Aura  30   17 continuing      yes female        no       -5
8   1   22 786   Aura  30   21 continuing      yes female       yes        0
9   1   23 787   Aura  30   21 continuing      yes female       yes        0
10  1   24 788   Aura  30   18 continuing      yes female       yes        0
   posttreat
1          0
2          0
3          0
4          0
5          0
6          0
7          0
8         22
9         23
10        24
```

There are variables in the data set that we're not using in this analysis. For details, see `?KosteckiDillon`.

GLMMs may be fit by the `glmer` function (pronounced "glimmer") in the **lme4** package. As is also true for `lmer`, there is no provision for autocorrelated within-subject errors, and in the case of a GLMM, we don't have the alternative of using the `nlme` package. Even without explicit termporal autocorrelation, however, the random effects are complex for a fairly small small data set, and we will try to simplify this part of the model. Specifying fixed and random effects in `glmer` is the same as in `lmer`; additionally, as for `glm`, we may specify a distributional `family` argument, which, in turn, takes an optional `link` argument. In the current example, we use the default `logit` link for the `binomial` family, and so do not have to give the `link` explicitly.

Our initial attempt to fit a GLMM to the migraine-headaches data produces a convergence warning:

```
    > mod.mig.1 <- glmer(headache ~   # warning: time-consuming!
+         medication * (treatment + pretreat + sqrt(posttreat))
+         + (treatment + pretreat + sqrt(posttreat) | id),
+     data=KosteckiDillon, family=binomial)

Warning message:
In checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv,  :
  Model failed to converge with
    max|grad| = 0.00623063 (tol = 0.001, component 2)
```

The `glmer` function make provision for alternative optimizers, and after a bit of experimentation, we were able to obtain convergence using the `optimx` optimizer in the **optimx** package, specifying the optimization method as `"nlminb"`; `optimx` produces its own warning but nevertheless converges to a solution:

```
> library(optimx)
> mod.mig.1a <- glmer(headache ~
+         medication * (treatment + pretreat + sqrt(posttreat))
+         + (treatment + pretreat + sqrt(posttreat) | id),
+     data=KosteckiDillon, family=binomial,
+     control=glmerControl(optimizer="optimx",
+         optCtrl=list(method="nlminb")))

Warning message:
In optimx.check(par, optcfg$ufn, optcfg$ugr, optcfg$uhess, lower,  :
  Parameters or bounds appear to have different scalings.
  This can cause poor performance in optimization.
  It is important for derivative free methods like BOBYQA, UOBYQA, NEWUOA.
```

As it turns out, the two solutions are nearly identical:

```
> compareCoefs(mod.mig.1, mod.mig.1a)

Call:
1: glmer(formula = headache ~ medication * (treatment + pretreat +
  sqrt(posttreat)) + (treatment + pretreat + sqrt(posttreat) | id), data =
  KosteckiDillon, family = binomial)
2: glmer(formula = headache ~ medication * (treatment + pretreat +
  sqrt(posttreat)) + (treatment + pretreat + sqrt(posttreat) | id), data = KD,
   family = binomial, control = glmerControl(optimizer = "optimx", optCtrl =
  list(method = "nlminb")))
                                   Est. 1      SE 1    Est. 2      SE 2
(Intercept)                      2.27e-01  6.12e-01  2.27e-01  6.12e-01
medicationreduced                1.96e+00  8.86e-01  1.96e+00  8.86e-01
medicationcontinuing             2.79e-01  6.86e-01  2.79e-01  6.86e-01
treatmentyes                     3.38e-01  7.12e-01  3.40e-01  7.12e-01
pretreat                        -1.95e-02  4.18e-02 -1.97e-02  4.19e-02
sqrt(posttreat)                 -2.72e-01  9.21e-02 -2.72e-01  9.22e-02
medicationreduced:treatmentyes   4.51e-01  1.03e+00  4.47e-01  1.04e+00
```

```
medicationcontinuing:treatmentyes     1.16e+00  8.13e-01  1.16e+00  8.14e-01
medicationreduced:pretreat            6.22e-02  6.03e-02  6.26e-02  6.03e-02
medicationcontinuing:pretreat        -6.59e-06  4.76e-02  2.40e-04  4.77e-02
medicationreduced:sqrt(posttreat)    -1.05e-02  1.29e-01 -1.04e-02  1.29e-01
medicationcontinuing:sqrt(posttreat)  1.56e-02  1.13e-01  1.55e-02  1.13e-01
```

Thus the convergence warning in our intial attempt was likely a false alarm; in general, `glmer` is conservative about detecting convergence failures. Maximizing the likelihood for a GLMM is a much more formidible task than for a LMM, and numerical problems are common. Existing methods are approximations because exact evaluation of the likelihood is intractable. The `glmer` function implements various numerical methods, and by default uses a Laplace approximation, which is a compromise between accuracy and computational speed.

Type-II Wald tests for the fixed effects, computed by the `Anova` function in the **car** package, reveal that all of the interactions are non-significant, along with the pre-treatment trend, while the medication and treatment effects, along with the post-treatment trend, are highly statistically significant:

```
> Anova(mod.mig.1a)

Analysis of Deviance Table (Type II Wald chisquare tests)

Response: headache
                           Chisq Df Pr(>Chisq)
medication                 22.34  2    1.4e-05
treatment                  13.32  1    0.00026
pretreat                    0.38  1    0.53782
sqrt(posttreat)            34.60  1    4.1e-09
medication:treatment        2.38  2    0.30357
medication:pretreat         1.86  2    0.39392
medication:sqrt(posttreat)  0.06  2    0.96955
```

Before examining the estimated fixed effects in the model, we will attempt to simplify the random effects, removing each random effect in turn and performing a likelihood-ratio test relative to the initial model:

```
> mod.mig.2 <- update(mod.mig.1a,
+     formula = headache ~
+         medication * (treatment + pretreat + sqrt(posttreat))
+     + (-1 + as.numeric(treatment == "yes")  + pretreat + sqrt(posttreat) | id))
> anova(mod.mig.1a, mod.mig.2)

Data: KosteckiDillon
Models:
mod.mig.2: headache ~ medication + treatment + pretreat + sqrt(posttreat) +
mod.mig.2:     (-1 + as.numeric(treatment == "yes") + pretreat + sqrt(posttreat) |
mod.mig.2:         id) + medication:treatment + medication:pretreat + medication:sqrt(posttreat
mod.mig.1a: headache ~ medication * (treatment + pretreat + sqrt(posttreat)) +
mod.mig.1a:     (treatment + pretreat + sqrt(posttreat) | id)
           Df  AIC  BIC logLik deviance Chisq Chi Df Pr(>Chisq)
mod.mig.2  18 4384 4498  -2174     4348
mod.mig.1a 22 4373 4512  -2164     4329  19.7      4    0.00057
```

```
> pval(19.701, df=4) # no random intercepts

[1] 0.0003839

> mod.mig.3 <- update(mod.mig.1a,
+     formula = headache ~
+         medication * (treatment + pretreat + sqrt(posttreat))
+     + (pretreat + sqrt(posttreat) | id))
> anova(mod.mig.1a, mod.mig.3)

Data: KosteckiDillon
Models:
mod.mig.3: headache ~ medication + treatment + pretreat + sqrt(posttreat) +
mod.mig.3:     (pretreat + sqrt(posttreat) | id) + medication:treatment +
mod.mig.3:     medication:pretreat + medication:sqrt(posttreat)
mod.mig.1a: headache ~ medication * (treatment + pretreat + sqrt(posttreat)) +
mod.mig.1a:     (treatment + pretreat + sqrt(posttreat) | id)
           Df  AIC  BIC logLik deviance Chisq Chi Df Pr(>Chisq)
mod.mig.3  18 4377 4491  -2170     4341
mod.mig.1a 22 4373 4512  -2164     4329  12.1      4      0.017

> pval(12.092, df=4) # no random treatment

[1] 0.01188

> mod.mig.4 <- update(mod.mig.1a,
+     formula = headache ~
+         medication * (treatment + pretreat + sqrt(posttreat))
+     + (treatment + sqrt(posttreat) | id))
> anova(mod.mig.1a, mod.mig.4)

Data: KosteckiDillon
Models:
mod.mig.4: headache ~ medication + treatment + pretreat + sqrt(posttreat) +
mod.mig.4:     (treatment + sqrt(posttreat) | id) + medication:treatment +
mod.mig.4:     medication:pretreat + medication:sqrt(posttreat)
mod.mig.1a: headache ~ medication * (treatment + pretreat + sqrt(posttreat)) +
mod.mig.1a:     (treatment + pretreat + sqrt(posttreat) | id)
           Df  AIC  BIC logLik deviance Chisq Chi Df Pr(>Chisq)
mod.mig.4  18 4370 4484  -2167     4334
mod.mig.1a 22 4373 4512  -2164     4329   5.8      4       0.21

> pval(5.7963, df=4) # no random pretreat

[1] 0.1684

> mod.mig.5 <- update(mod.mig.1,  # this fails with mod.mig.1a
+     formula =  headache ~
+         medication * (treatment + pretreat + sqrt(posttreat))
+     + (treatment + pretreat | id))
> anova(mod.mig.1, mod.mig.5)
```

```
Data: KosteckiDillon
Models:
mod.mig.5: headache ~ medication + treatment + pretreat + sqrt(posttreat) +
mod.mig.5:     (treatment + pretreat | id) + medication:treatment + medication:pretreat +
mod.mig.5:     medication:sqrt(posttreat)
mod.mig.1: headache ~ medication * (treatment + pretreat + sqrt(posttreat)) +
mod.mig.1:     (treatment + pretreat + sqrt(posttreat) | id)
          Df  AIC  BIC logLik deviance Chisq Chi Df Pr(>Chisq)
mod.mig.5 18 4381 4495  -2172     4345
mod.mig.1 22 4373 4512  -2164     4329  16.2      4     0.0027
```

```
> pval(16.214, df=4) # no random posttreat
```

```
[1] 0.001885
```

As in LMMs, we use our `pval` function to correct chisquare tests for the variance/covariance components, reflecting the fact that the null values of variances are on the boundary of the parameter space. As well, we were unable to fit `mod.mig.5` using `optimx` without producing an error, and so we updated `mod.mig.1` rather than `mod.mig.1a` to obtain `mod.mig.5` and the corresponding test for the random post-treatment effect. The relatively convoluted specification of `mod.mig.2`, where the dummy regressor for `treatment` is generated directly, rather than putting the factor `treatment` in the random-effects formula, is necessary to suppress the random effect for the intercept; simply specifying `-1` with the factor `treatment` in the random-effects formula places *two* dummy regressors in the random-effects model, fitting different intercepts for each of the two levels of `treatment`, and producing a model equivalent to `mod.mig.1a`.

On the basis of these tests for the fixed and random effects, we specified a final model for the migraines data that eliminates the fixed-effect interactions with medication and the pre-treatment trend fixed and random effects, obtaining the following estimates for the fixed effects and variance components:

```
> mod.mig.6 <- glmer(headache ~ medication + treatment + sqrt(posttreat)
+     + (treatment + sqrt(posttreat) | id),
+     data=KosteckiDillon, family=binomial,
+     control=glmerControl(optimizer="optimx",
+         optCtrl=list(method="nlminb")))
> summary(mod.mig.6)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: headache ~ medication + treatment + sqrt(posttreat) + (treatment +
    sqrt(posttreat) | id)
   Data: KosteckiDillon
Control: glmerControl(optimizer = "optimx", optCtrl = list(method = "nlminb"))

     AIC      BIC   logLik deviance df.resid
    4369     4439    -2174     4347     4141

Scaled residuals:
```

```
   Min     1Q Median     3Q    Max
-5.182 -0.646  0.260  0.580  3.690

Random effects:
 Groups Name            Variance Std.Dev. Corr
 id     (Intercept)     1.7011   1.304
        treatmentyes    1.7126   1.309    -0.12
        sqrt(posttreat) 0.0571   0.239     0.11 -0.66
Number of obs: 4152, groups:  id, 133

Fixed effects:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)           -0.2459     0.3438   -0.72   0.4745
medicationreduced      2.0501     0.4679    4.38  1.2e-05
medicationcontinuing   1.1553     0.3838    3.01   0.0026
treatmentyes           1.0608     0.2439    4.35  1.4e-05
sqrt(posttreat)       -0.2684     0.0449   -5.98  2.2e-09

Correlation of Fixed Effects:
            (Intr) mdctnr mdctnc trtmnt
medictnrdcd -0.674
mdctncntnng -0.828  0.656
treatmentys -0.215 -0.053 -0.049
sqrt(psttr)  0.016 -0.009 -0.002 -0.685
```

Figure 11 shows the estimated fixed effects plotted on the probability scale; as a consequence, the post-treatment trends for the three medication conditions are not parallel, as they would be if plotted on the logit scale:

```
> new.1 <- expand.grid(treatment="yes", posttreat=1:99,
+     medication=c("reduced", "continuing", "none"))
> new.1$treatment <- factor("yes", levels=c("no", "yes"))
> new.2 <- expand.grid(treatment="no", posttreat=-29:0,
+     medication=c("reduced", "continuing", "none"))
> new.2$posttreat <- 0
> new.2$treatment <- factor("no", levels=c("no", "yes"))
> new <- rbind(new.2, new.1)
> new$p <- predict(mod.mig.6, newdata=new, re.form=NA, type="response")
> new$time <- c(rep(-29:0, 3),rep(1:99, 3))
> plot(p ~ time, type="n", data=new, xlab="Time (days)",
+     ylab="Fitted Probability of Headache")
> abline(v=0, col="gray")
> lines(p ~ time, subset = medication == "none", data=new,
+     lty=1, lwd=2)
> lines(p ~ time, subset = medication == "reduced", data=new,
+     lty=2, lwd=2)
> lines(p ~ time, subset = medication == "continuing", data=new,
+     lty=3, lwd=2)
> legend("topright", lty=1:3, lwd=2,
```

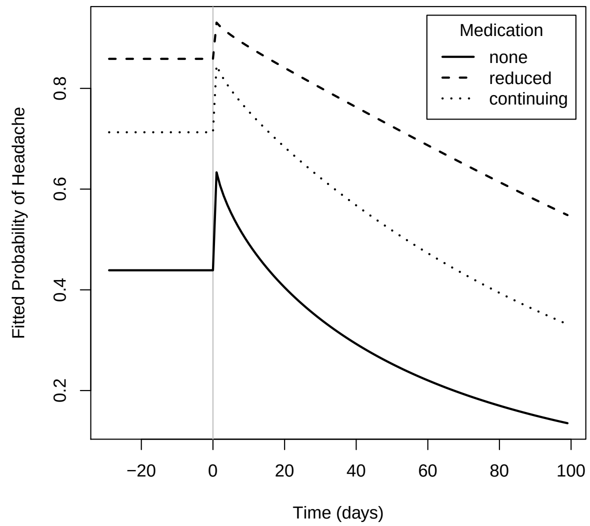

Figure 11: Fixed effects from a binomial GLMM fit to the migraines data. Treatment started at time 1.

```
+     legend=c("none", "reduced", "continuing"), title="Medication",
+     inset=.02)
```

It is apparent from this graph that after an initial increase at the start of treatment, the incidence of headaches declined to substantially below its pre-treatment level. As well, the incidence of headaches was lowest among the patients who discontinued their medication, and highest among those who reduced their medication; patients who continued their medication at pre-training levels were intermediate in headache incidence. Of course, self-selection of the medication groups renders interpretation of this pattern ambiguous.

4 Nonlinear Mixed Models

<to come>

5 Complementary Reading and References

Much of the material in this appendix is adapted from Fox (2015, Chaps. 23 and 24). A very brief treatment of mixed models may be found in Weisberg (2014, Sec. 7.4). Snijders and Bosker (2012) and Raudenbush and Bryk (2002) are two accessible books that emphasize hierarchical linear and, to a lesser extent, generalized-linear models. Gelman and Hill (2007) develop mixed models in the more general context of regression analysis. Stroup (2013) presents a more formal and comprehensive development of generalized linear mixed models, treating other regression models, such as linear models, generalized linear models, and linear mixed-effects models as special cases (and emphasizing **SAS** software for fitting these models).

References

Bates, D., Maechler, M., Bolker, B., and Walker, S. (2014). *lme4: Linear mixed-effects models using Eigen and S4*. R package version 1.1-7.

Davis, C., Blackmore, E., Katzman, D. K., and Fox, J. (2005). Female adolescents with anorexia nervosa and their parents: A case-control study of exercise attitudes and behaviours. *Psychological Medicine*, 35:377–386.

Fox, J. (2015, in press). *Applied Regression Analysis and Generalized Linear Models*. Sage, Thousand Oaks CA, third edition.

Gelman, A. and Hill, J. (2007). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, Cambridge UK.

Kenward, M. G. and Roger, J. (1997). Small sample inference for fixed effects from restricted maximum likelihood. *Biometrics*, 53:983–997.

Laird, N. M. and Ware, J. H. (1982). Random-effects models for longitudinal data. *Biometrics*, 38:963–974.

Pinheiro, J., Bates, D., DebRoy, S., Sarkar, D., and R Core Team (2014). *nlme: Linear and Nonlinear Mixed Effects Models*. R package version 3.1-117.

Pinheiro, J. C. and Bates, D. M. (2000). *Mixed-effects models in S and S-PLUS*. Springer, New York.

Raudenbush, S. W. and Bryk, A. S. (2002). *Hierarchical linear models: Applications and Data Analysis Methods*. Sage, Thousand Oaks CA, second edition.

Satterthwaite, F. E. (1946). An approximate distribution of estimates of variance components. *Biometrics Bulletin*, 2:110–114.

Singer, J. D. (1998). Using SAS PROC MIXED to fit multilevel models, hierarchical models, and individual growth models. *Journal of Educational and Behavioral Statistics*, 24:323–355.

Snijders, T. A. B. and Bosker, R. J. (2012). *Multilevel Analysis: An Introduction to Basic and Advanced Multilevel Modeling*. Sage, Thousand Oaks CA, 2nd edition.

Stroup, W. W. (2013). *Generalized Linear Mixed Models: Modern Concepts, Methods and Applications*. CRC Press, Boca Raton FL.

Weisberg, S. (2014). *Applied Linear Regression*. Wiley, Hoboken NJ, fourth edition.