

# Linear Mixed Models

Appendix to *An R and S-PLUS Companion to Applied Regression*

John Fox

May 2002

## 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*, the error term  $\varepsilon_i$ . The parameters of the model are the regression coefficients,  $\beta_1, \beta_2, \dots, \beta_p$ , and the error variance,  $\sigma^2$ . Usually,  $x_{1i} = 1$ , and so  $\beta_1$  is a constant or intercept.

For comparison with the linear mixed model of the next section, I 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  $\mathbf{y} = (y_1, y_2, \dots, y_n)'$  is the response vector;  $\mathbf{X}$  is the model matrix, with typical row  $\mathbf{x}'_i = (x_{1i}, x_{2i}, \dots, x_{pi})$ ;  $\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_p)'$  is the vector of regression coefficients;  $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)'$  is the vector of errors;  $\mathbf{N}_n$  represents the  $n$ -variable multivariate-normal distribution;  $\mathbf{0}$  is an  $n \times 1$  vector of zeroes; 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 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 and S-PLUS for fitting mixed models to data, the most ambitious of which is the `nlme` library (an acronym for *non-linear mixed effects*), described in detail by Pinheiro and Bates (2000). Despite its name, this library includes facilities for fitting *linear* mixed models (along with *nonlinear* mixed models), the subject of the present appendix. There are plans to incorporate *generalized* linear mixed models (for example, for logistic and Poisson regression) in the `nlme` library. In the interim, the reader may wish to consult the documentation for the `glmmPQL` function in Venables and Ripley's (1999) `MASS` library.<sup>1</sup>

Mixed models are a large and complex subject, and I will only scrape the surface here. I recommend Raudenbush and Bryk (2002) as a general, relatively gentle, introduction to the subject for social scientists, and Pinheiro and Bates (2000), which I have already mentioned, as the definitive reference for the `nlme` library.

## 2 The Linear Mixed Model

Linear mixed models 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 explained in the next section. The

---

<sup>1</sup>Version 6.3-2 of the `MASS` library (or, I assume, a newer version) is required.

`lme` (linear mixed effects) function in the `nlme` library, however, employs the *Laird-Ware form* of the linear mixed model (after a seminal paper on the topic published by Laird and Ware, 1982):

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

where

- $y_{ij}$  is the value of the response variable for the  $j$ th of  $n_i$  observations in the  $i$ th of  $M$  groups or clusters.
- $\beta_1, \dots, \beta_p$  are the fixed-effect coefficients, which are identical for all groups.
- $x_{1ij}, \dots, x_{pij}$  are the fixed-effect regressors for observation  $j$  in group  $i$ ; the first regressor is usually for the constant,  $x_{1ij} = 1$ .
- $b_{i1}, \dots, b_{iq}$  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  $\varepsilon_{ij}$ .
- $z_{1ij}, \dots, z_{qij}$  are the random-effect regressors.
- $\psi_k^2$  are the variances and  $\psi_{kk'}$  the covariances among the random effects, assumed to be constant across groups. In some applications, the  $\psi$ 's are parametrized in terms of a relatively small number of fundamental parameters.
- $\varepsilon_{ij}$  is the error for observation  $j$  in group  $i$ . The errors for group  $i$  are assumed to be multivariately normally distributed.
- $\sigma^2 \lambda_{ijj'}$  are the covariances between errors 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 the following section),  $\lambda_{ijj} = \sigma^2$ ,  $\lambda_{ijj'} = 0$  (for  $j \neq j'$ ), and thus the only free parameter to estimate is the common error variance,  $\sigma^2$ . Similarly, if the observations in a “group” represent longitudinal data on a single individual, then the structure of the  $\lambda$ 's may be specified to capture autocorrelation among the errors, as is common in observations collected over time.

Alternatively but equivalently, in matrix form,

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

where

- $\mathbf{y}_i$  is the  $n_i \times 1$  response vector for observations in the  $i$ th 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  $i$ .

### 3 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 Bryk and Raudenbush's influential (1992) text on hierarchical linear models, and also appears in a paper by Singer (1998), which shows how such models can be fit by the MIXED procedure in SAS.<sup>2</sup>

The data for the example, from the 1982 “High School and Beyond” survey, and pertain to 7185 high-school students from 160 schools, are present in the data frames `MathAchieve` and `MathAchSchool`, distributed with the `nlme` library:<sup>3</sup>

```
> library(nlme)
Loading required package: nls

> library(lattice) # for Trellis graphics
Loading required package: grid

> data(MathAchieve)
> MathAchieve[1:10,] # 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
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

> data(MathAchSchool)
> MathAchSchool[1:10,] # 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
```

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 shall require the following variables:

- **School:** an identification number for the student's school. Although it is not required by `lme`, students in a specific school are in consecutive rows of the data frame, a convenient form of data organiza-

---

<sup>2</sup>The data were obtained from Singer. There is now a second edition of Bryk and Raudenbush's text, Raudenbush and Bryk (2002).

<sup>3</sup>These are actually *grouped-data objects*, which include some additional information along with the data. See the discussion below.

tion. 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'. Note that 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*, to distinguish it from an *inner variable* (such as SES), which varies within groups. Because the 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.
- **MEANSES**: another outer variable, giving the mean SES for students in each school. 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 — and I will therefore recompute it (using `tapply` — see Section 8.4 of the text) and replace it in the student data set:

```
> attach(MathAchieve)
> mses <- tapply(SES, School, mean) # school means
> mses[as.character(MathAchSchool$School[1:10])] # for first 10 schools
      1224      1288      1296      1308      1317      1358
-0.434383  0.121600 -0.425500  0.528000  0.345333 -0.019667
      1374      1433      1436      1461
-0.012643  0.712000  0.562909  0.677455
> detach(MathAchieve)
>
```

### The `nlme` and `trellis` Libraries in S-PLUS

In S-PLUS, the `nlme` library (called `nlme3`) and the `trellis` library are in the search path at the beginning of the session and need not be attached via the `library` function.

I 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) <- c("school", "ses", "mathach")
> sample20 <- sort(sample(7185, 20)) # 20 randomly sampled students
> Bryk[sample20, ]
      school      ses mathach
20      1224 -0.078  16.405
280     1433  0.952  19.033
1153    2526 -0.708   7.359
1248    2629 -0.598  17.705
1378    2655 -0.548  17.205
1939    3152 -0.628  15.751
2182    3499  0.592  10.491
2307    3610 -0.678   8.565
2771    4042  1.082  15.297
2779    4042  0.292  20.672
2885    4223 -0.188  17.984
3016    4292  0.292  23.619
```

```

4308  6144 -0.848  2.799
4909  6816 -0.148 16.446
5404  7345 -1.528  8.788
5704  7890  0.802  3.744
5964  8193 -0.828 10.748
6088  8477 -0.398 16.305
6637  9021  1.512  9.930
7154  9586  1.082 14.266

```

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

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

```

> Bryk$meanses <- mses[as.character(Bryk$school)]
> Bryk$cses <- Bryk$ses - Bryk$meanses

> sector <- MathAchSchool$Sector
> names(sector) <- row.names(MathAchSchool)
> Bryk$sector <- sector[as.character(Bryk$school)]

> Bryk[sample20,]
   school    ses mathach  meanses    csese  sector
20    1224 -0.078  16.405 -0.434383  0.35638  Public
280    1433  0.952  19.033  0.712000  0.24000  Catholic
1153    2526 -0.708   7.359  0.326912 -1.03491  Catholic
1248    2629 -0.598  17.705 -0.137649 -0.46035  Catholic
1378    2655 -0.548  17.205 -0.701654  0.15365  Public
1939    3152 -0.628  15.751  0.031038 -0.65904  Public
2182    3499  0.592  10.491  0.449895  0.14211  Catholic
2307    3610 -0.678   8.565  0.120125 -0.79813  Catholic
2771    4042  1.082  15.297  0.402000  0.68000  Catholic
2779    4042  0.292  20.672  0.402000 -0.11000  Catholic
2885    4223 -0.188  17.984 -0.094000 -0.09400  Catholic
3016    4292  0.292  23.619 -0.486154  0.77815  Catholic
4308    6144 -0.848   2.799 -0.437535 -0.41047  Public
4909    6816 -0.148  16.446  0.528909 -0.67691  Catholic
5404    7345 -1.528   8.788  0.033250 -1.56125  Public
5704    7890  0.802   3.744 -0.522706  1.32471  Public
5964    8193 -0.828  10.748 -0.176605 -0.65140  Catholic
6088    8477 -0.398  16.305 -0.196108 -0.20189  Public
6637    9021  1.512   9.930  0.626643  0.88536  Catholic
7154    9586  1.082  14.266  0.621153  0.46085  Catholic

```

The following steps are a bit tricky:

- The students' school numbers (in `Bryk$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 Bryk and Raudenbush, we will ask whether math achievement is related to socioeconomic status; whether this relationship varies systematically by sector; and whether the relationship varies randomly across schools within the same sector.

### 3.1 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 I start by selecting samples of 20 public and 20 Catholic schools, storing each sample as a grouped-data object:

```
> attach(Bryk)
> cat <- sample(unique(school[sector=='Catholic']), 20)
> Cat.20 <- groupedData(mathach ~ ses | school,
+   data=Bryk[is.element(school, cat),])

> pub <- sample(unique(school[sector=='Public']), 20)
> Pub.20 <- groupedData(mathach ~ ses | school,
+   data=Bryk[is.element(school, pub),])
>
```

Grouped-data objects are enhanced data frames, provided by the `nlme` library, incorporating a model formula that gives information about the structure of the data. In this instance, the formula `mathach ~ ses | school`, read as “`mathach` is modeled as `ses` given `school`,” indicates that `mathach` is the response variable, `ses` is the principal within-group (i.e., inner) covariate, and `school` is the grouping variable.

Although `nlme` provides a `plot` method for grouped-data objects, which makes use of Trellis graphics, the graphs are geared more towards longitudinal data than hierarchical data. In the present context, therefore, I prefer to use Trellis graphics directly, as follows:

```
> trellis.device(color=F)
> xyplot(mathach ~ ses | 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 ~ ses | 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, I have specified monochrome graphics (`color = F`) 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. I created one display 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:

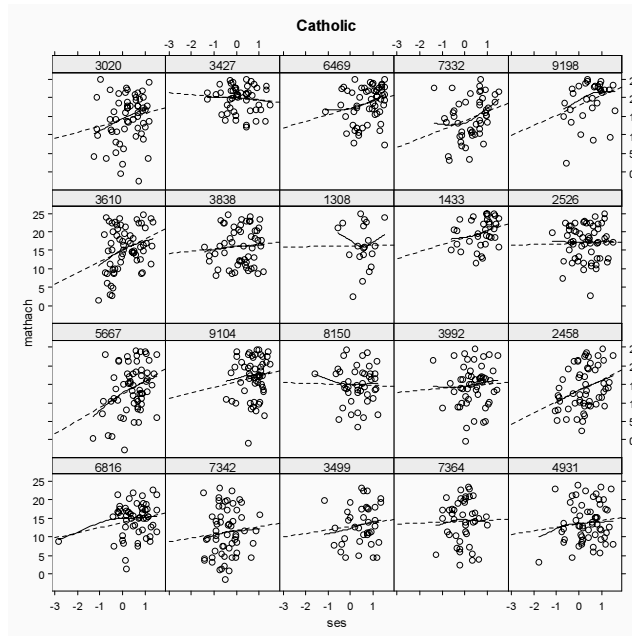


Figure 1: Trellis display of math achievement by socio-economic status for 20 randomly selected Catholic schools. The broken lines give linear least-squares fits, the solid lines local-regression fits.

- `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 relatively small number of observations for each school, I 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 zero 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 small 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` library 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`". Here, I fit the regression of math-achievement scores on socioeconomic status for each school, creating separate `lmList` objects for Catholic and public schools:

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

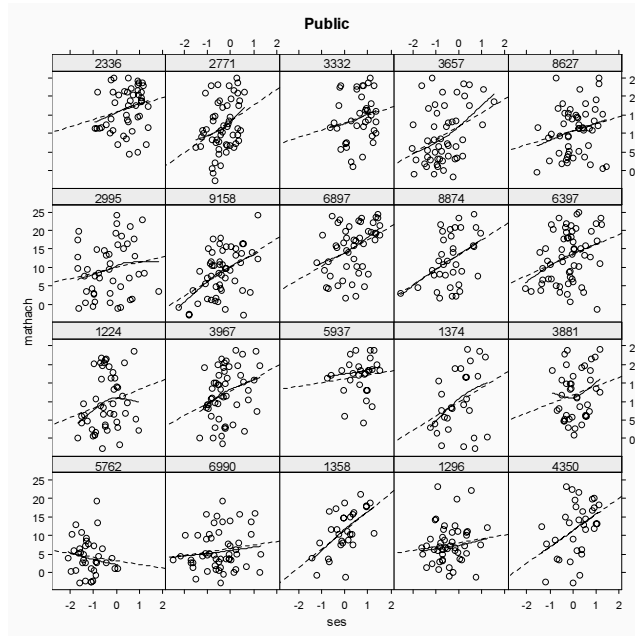


Figure 2: Trellis display of math achievement by socio-economic status for 20 randomly selected public schools.

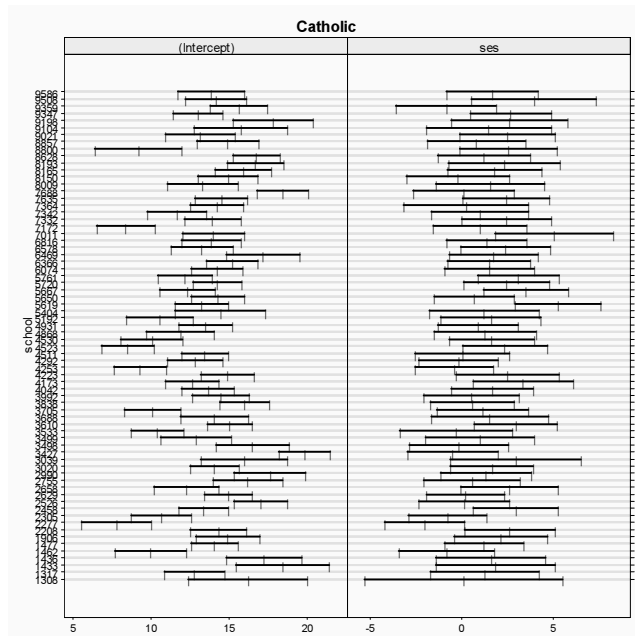


Figure 3: 95-percent confidence intervals for the intercepts and slopes of the within-schools regressions of math achievement on SES, for Catholic schools.



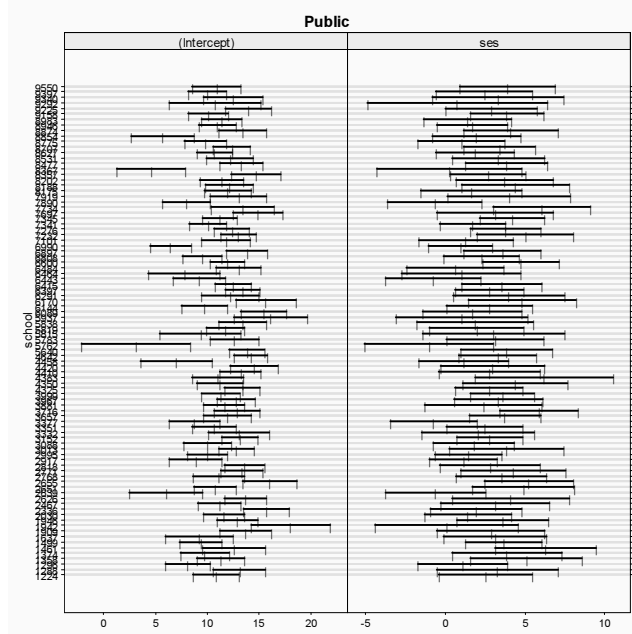


Figure 4: 95-percent confidence intervals for the intercepts and slopes of the within-schools regressions of math achievement on SES, for public schools.

The resulting graphs are shown in Figures 3 and 4. In interpreting these graphs, we need to be careful to remember that I 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 zero, the intercepts are interpretable as the predicted levels of math achievement in each school at an overall average level of SES. 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, I draw parallel boxplots for the coefficients:

```
> cat.coef <- coef(cat.list)
> cat.coef[1:10,]
      (Intercept)      ses
1308      16.1890  0.12602
1317      12.7378  1.27391
1433      18.3989  1.85429
1436      17.2106  1.60056
1462       9.9408 -0.82881
1477      14.0321  1.23061
1906      14.8855  2.14551
2208      14.2889  2.63664
2277       7.7623 -2.01503
2305      10.6466 -0.78211

> pub.coef <- coef(pub.list)
> pub.coef[1:10,]
      (Intercept)      ses
1224      10.8051  2.508582
1288      13.1149  3.255449
```

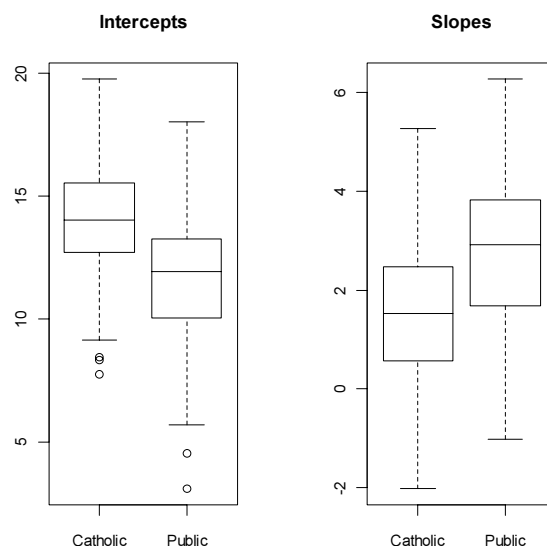


Figure 5: Boxplots of intercepts and slopes for the regressions of math achievement on SES in Catholic and public schools.

```
1296      8.0938 1.075959
1358     11.3059 5.068009
1374      9.7772 3.854323
1461     12.5974 6.266497
1499      9.3539 3.634734
1637      9.2227 3.116806
1909     13.7151 2.854790
1942     18.0499 0.089383
```

```
> 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)
>
```

The calls to `coef` extract matrices of regression coefficients from the `lmList` objects, with rows representing schools. Setting `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. At an average level of SES, 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.

### 3.2 Fitting a Hierarchical Linear Model with `lme`

Following Bryk and Raudenbush (1992) and Singer (1998), I 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

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

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

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

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

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

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

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  $\gamma$ 's are fixed effects, while the  $u$ 's (and the individual-level errors  $\varepsilon_{ij}$ ) are random effects.

Finally, rewriting the model in the notation of the linear mixed model (equation 1),

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

The change is purely notational, using  $\beta$ 's for fixed effects and  $b$ 's 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.) I place no constraints on the covariance matrix of the random effects, so

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

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

$$V(\varepsilon_i) = \sigma^2 \mathbf{I}_{n_i}$$

As mentioned in Section 2, linear mixed models are fit with the **lme** function in the **nlme** library. 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, I reorder the levels of the factor **sector** so that the contrast coding the **sector** effect will use the value 0 for the public sector and 1 for the Catholic sector, in conformity with the coding employed by Bryk and Raudenbush (1992) and by Singer (1998):

```
> Bryk$sector <- factor(Bryk$sector, levels=c('Public', 'Catholic'))
> contrasts(Bryk$sector)
      Catholic
Public      0
Catholic    1
```

## Reminder: Default Contrast Coding in S-PLUS

Recall that in S-PLUS, the default contrast function for unordered factors is `contr.helmert` rather than `contr.treatment`. As described in Section 4.2 of the text, there are several ways to change the contrast coding, including resetting the global default: `options(contrasts=c('contr.treatment', 'contr.poly'))`.

Having established the contrast-coding for `sector`, the linear mixed model in equation 4 is fit as follows:

```
> 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
46525 46594 -23252

Random effects:
Formula: ~cses | school
Structure: General positive-definite, Log-Cholesky parametrization
           StdDev  Corr
(Intercept) 1.541177 (Intr)
cses         0.018174 0.006
Residual     6.063492

Fixed effects: mathach ~ meanses * cses + sector * cses
              Value Std.Error   DF t-value p-value
(Intercept)   12.1282   0.19920 7022   60.886 <.0001
meanses        5.3367   0.36898  157   14.463 <.0001
cses           2.9421   0.15122 7022   19.456 <.0001
sectorCatholic  1.2245   0.30611  157    4.000 1e-04
meanses:cses   1.0444   0.29107 7022    3.588 3e-04
cses:sectorCatholic -1.6421 0.23312 7022   -7.044 <.0001
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
-3.170106 -0.724877  0.014892  0.754263  2.965498

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 Pinheiro and Bates (2000) for details on this

and other points.

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 6.5.2 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  $\sigma$ . Thus,  $\hat{\psi}_1 = 1.541$ ,  $\hat{\psi}_2 = 0.018$ ,  $\hat{\sigma} = 6.063$ , and  $\hat{\psi}_{12} = 0.006 \times 1.541 \times 0.018 = 0.0002$ .
- 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 linear mixed model in equation 4 (which orders the coefficients differently from the `lme` output). In particular:
  - The fixed-effect intercept coefficient  $\hat{\beta}_1 = 12.128$  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**,  $\hat{\beta}_3 = 1.225$ , represents the difference between the average level of math achievement in Catholic schools and public schools.
  - The coefficient for **cses**,  $\hat{\beta}_4 = 2.942$ , is the estimated average slope for SES in public schools, while the coefficient labelled **cses:sectorCatholic**,  $\hat{\beta}_6 = -1.642$ , 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**,  $\hat{\beta}_2 = 5.337$ , represents the relationship of schools' average level of math achievement to their average level of SES
  - The coefficient for the interaction **meanses:cse**s,  $\hat{\beta}_5 = 1.044$ , gives the average change in the within-school SES slope associated with a one-unit increment in the school's mean SES. Notice that all of the coefficients are highly statistically significant.
- The panel labelled **Correlation** gives the estimated sampling correlations among the fixed-effect coefficient estimates, which 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 ( $\hat{\varepsilon}_{ij}/\hat{\sigma}$ ), the number of observations, and the number of groups, appears at the end of the output.

## Different Solutions from `lme` in R and S-PLUS

The results for  $\hat{\psi}_2$  (the standard deviation of the random effects of `cse`s) and  $\hat{\psi}_{12}$  (the covariance between the random intercepts and slopes) are different in S-PLUS, where `lme` uses a different (and more robust) optimizer than in R.

Comparing the maximized log-likelihoods for the two programs suggests that the R version of `lme` has not converged fully to the REML solution. Checking confidence intervals for the variance components in the R solution [by the command `intervals(bryk.lme.1)`] shows that the confidence interval for  $\psi_2$  in particular is extremely wide, raising the possibility that the problem is ill-conditioned. (These normal-theory confidence intervals for variance components should not be trusted, but they do help to reveal estimates that are not well determined by the data.)

In the S-PLUS solution, in contrast, the confidence interval for  $\psi_2$  is much narrower, but the confidence interval for the correlation between the two sets of random effects is very wide. Not coincidentally, it turns out that  $\psi_2$  and  $\psi_{12}$  can be dropped from the model.

I am grateful to Douglas Bates and José Pinheiro for clarifying the source of the differences in the results from the R and S-PLUS versions of `nlme`.

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. We must be careful, however, to compare models that are identical in their fixed effects.

For the current illustration, we may proceed as follows:

```
> 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 46525 46594 -23252
bryk.lme.2    2  8 46521 46576 -23252 1 vs 2 0.0032069 0.9984

> 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 46525 46594 -23252
bryk.lme.3    2  8 46740 46795 -23362 1 vs 2 219.44 <.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.

Model `bryk.lme.2`, fit above, omits the non-significant random effects for `cse`s; 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)
> summary(bryk.lme.2)
Linear mixed-effects model fit by REML
Data: Bryk
      AIC    BIC logLik
46521 46576 -23252
```

```

Random effects:
Formula: ~1 | school
(Intercept) Residual
StdDev:      1.5412   6.0635

Fixed effects: mathach ~ meanses * cses + sector * cses
              Value Std.Error   DF t-value p-value
(Intercept)  12.1282   0.19920 7022  60.885 <.0001
meanses      5.3367   0.36898  157  14.463 <.0001
cses         2.9421   0.15121 7022  19.457 <.0001
sectorCatholic 1.2245   0.30612  157   4.000 1e-04
meanses:cses 1.0444   0.29105 7022   3.589 3e-04
cses:sectorCatholic -1.6422 0.23309 7022  -7.045 <.0001
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
-3.170115 -0.724877 0.014845 0.754242 2.965513

Number of Observations: 7185
Number of Groups: 160

```

## 4 An Illustrative Application to Longitudinal Data

To illustrate the use of linear mixed models for longitudinal research, I draw on as-yet unpublished data collected by Blackmoor and Davis on the exercise histories of 138 teenaged girls hospitalized for eating disorders, and on a group of 93 ‘control’ subjects.<sup>4</sup> The data are in the data frame `Blackmoor` in the `car` library:

```

> library(car) # for data only
. . .

> data(Blackmoor) # from car
> Blackmoor[1:20,] # first 20 observations
  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

```

---

<sup>4</sup>These data were generously made available to me by Elizabeth Blackmoor and Caroline Davis of York University.

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 two years, starting at age eight.
- **exercise**: the amount of exercise in which the subject engaged, expressed as estimated hours per week.
- **group**: a factor indicating whether the subject is a 'patient' or a 'control'.<sup>5</sup>

## 4.1 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**. Because there are some 0 values of **exercise**, I use “started” logs in the analysis reported below (see Section 3.4 of the text on transforming data), adding five minutes (5/60 of an hour) to each value of **exercise** prior to taking logs (and using logs to the base 2 for interpretability):

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

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

```
> pat <- sample(unique(subject[group=='patient']), 20)
> Pat.20 <- groupedData(log.exercise ~ age | subject,
+   data=Blackmoor[is.element(subject, pat),])

> con <- sample(unique(subject[group=='control']), 20)
> Con.20 <- groupedData(log.exercise ~ age | subject,
+   data=Blackmoor[is.element(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, 1, .5), more=T)
```

---

<sup>5</sup>To avoid the possibility of confusion, I 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**.



```

> 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, 1, 1))
>

```

The graphs appear in Figure 6.

- Each Trellis plot is constructed by using the default `plot` method for grouped-data objects.
- To make the two plots comparable, I 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)<sup>6</sup>, and the aspect ratio of the plot (the ratio of the 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,” I 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=T` 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.

I 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) by pooling information across subjects.<sup>7</sup>

```

> pat.list <- lmList(log.exercise ~ I(age - 8) | subject,
+   subset = group=='patient', data=Blackmoor)
> con.list <- lmList(log.exercise ~ I(age - 8) | subject,
+   subset = group=='control', data=Blackmoor)
> 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)
>

```

The boxplots of regression coefficients are shown in Figure 7. I 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.

<sup>6</sup>Notice the unusual ordering in specifying the layout — columns first, then rows.

<sup>7</sup>Pooled estimates of the random effects provide so-called *best-linear-unbiased predictors* (or *BLUPs*). See `help(predict.lme)` and Pinheiro and Bates (2000).

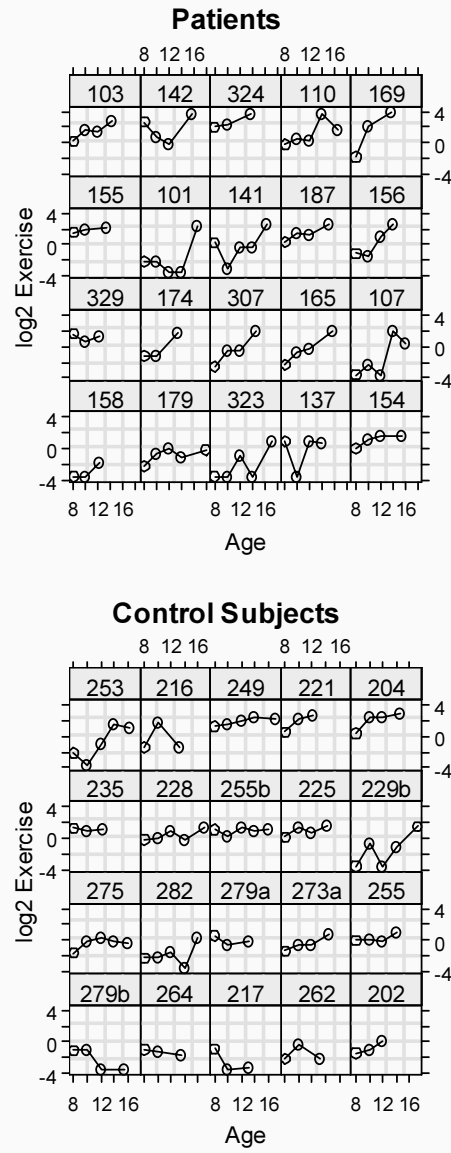


Figure 6:  $\log_2$  exercise by age for 20 randomly selected patients and 20 randomly selected control subjects.

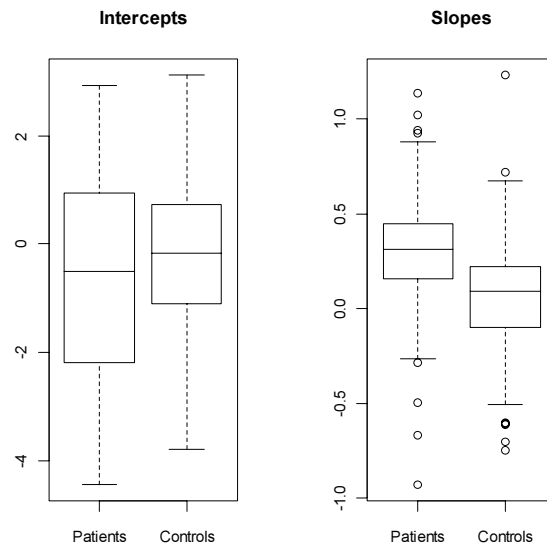


Figure 7: Coefficients for the within-subject regressions of  $\log_2$  exercise on age, for patients and control subjects

## 4.2 Fitting the Mixed Model

I proceed to fit a linear mixed model 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=Blackmoor)
> summary(bm.lme.1)
Linear mixed-effects model fit by REML
Data: Blackmoor
      AIC      BIC logLik
3630.1 3668.9 -1807.1

Random effects:
Formula: ~I(age - 8) | subject
Structure: General positive-definite, Log-Cholesky parametrization
          StdDev  Corr
(Intercept) 1.44356 (Intr)
I(age - 8)   0.16480 -0.281
Residual     1.24409

Fixed effects: log.exercise ~ I(age - 8) * group
              Value Std.Error DF t-value p-value
(Intercept) -0.27602  0.182368 712 -1.5135  0.1306
I(age - 8)   0.06402  0.031361 712  2.0415  0.0416
grouppatient -0.35400  0.235291 229 -1.5045  0.1338
I(age - 8):grouppatient 0.23986  0.039408 712  6.0866 <.0001
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
-2.73486 -0.42451  0.12277  0.52801  2.63619

```

Number of Observations: 945

Number of Groups: 231

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-eight intercepts for the two groups.

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

```

> bm.lme.2 <- update(bm.lme.1, random = ~ 1 | subject)
> anova(bm.lme.1, bm.lme.2)
      Model df      AIC      BIC logLik  Test L.Ratio p-value
bm.lme.1    1  8 3630.1 3668.9 -1807.1
bm.lme.2    2  6 3644.3 3673.3 -1816.1 1 vs 2  18.122  1e-04

> bm.lme.3 <- update(bm.lme.1, random = ~ I(age - 8) - 1 | subject)
> anova(bm.lme.1, bm.lme.3)
      Model df      AIC      BIC logLik  Test L.Ratio p-value
bm.lme.1    1  8 3630.1 3668.9 -1807.1
bm.lme.3    2  6 3834.1 3863.2 -1911.0 1 vs 2  207.95 <.0001

```

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

Let us next consider the possibility that the within-subject errors (the  $\varepsilon_{ij}$ 's in the mixed model) are auto-correlated — as may well be the case, since the observations are taken longitudinally on the same subjects. The **lme** function incorporates a flexible mechanism for specifying correlation structures, and supplies constructor functions for several such structures.<sup>8</sup> 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  $\varepsilon_{it}$  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 the following results:

```

> bm.lme.4 <- update(bm.lme.1, correlation = corCAR1(form = ~ age |subject))
> summary(bm.lme.4)
Linear mixed-effects model fit by REML
Data: Blackmoor
      AIC      BIC logLik
3607.1 3650.8 -1794.6

```

Random effects:

---

<sup>8</sup>A similar mechanism is provided for modeling non-constant variance, via the **weights** argument to **lme**. See the documentation for **lme** for details.

```

Formula: ~I(age - 8) | subject
Structure: General positive-definite, Log-Cholesky parametrization
          StdDev   Corr
(Intercept) 1.053381 (Intr)
I(age - 8)   0.047939 0.573
Residual    1.514138

```

Correlation Structure: Continuous AR(1)

```

Formula: ~age | subject
Parameter estimate(s):
  Phi
0.6236

```

```

Fixed effects: log.exercise ~ I(age - 8) * group
              Value Std.Error DF t-value p-value
(Intercept)  -0.306202  0.182027 712 -1.6822  0.0930
I(age - 8)    0.072302  0.032020 712  2.2580  0.0242
grouppatient  -0.289267  0.234968 229 -1.2311  0.2196
I(age - 8):grouppatient 0.230054  0.040157 712  5.7289 <.0001
Correlation:
              (Intr) I(g-8) grpptn
I(age - 8)    -0.509
grouppatient  -0.775  0.395
I(age - 8):grouppatient 0.406 -0.797 -0.511

```

Standardized Within-Group Residuals:

	Min	Q1	Med	Q3	Max
	-2.67829	-0.46383	0.16530	0.58823	2.11817

```

Number of Observations: 945
Number of Groups: 231

```

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`). The estimated autocorrelation,  $\hat{\phi} = 0.62$ , is quite large, but the fixed-effects estimates and their standard errors have not changed much.<sup>9</sup> A likelihood-ratio test establishes the statistical significance of the error autocorrelation:

```

> anova(bm.lme.1, bm.lme.4)
      Model df    AIC    BIC logLik  Test L.Ratio p-value
bm.lme.1   1  8 3630.1 3668.9 -1807.1
bm.lme.4   2  9 3607.1 3650.8 -1794.6 1 vs 2  25.006 <.0001

```

Because the specification of the model has changed, we can re-test whether random intercepts and slopes are required; as it turns out, the random `age` term may now be removed from the model, but not the random intercepts:

```

> bm.lme.5 <- update(bm.lme.4, random = ~ 1 | subject)
> anova(bm.lme.4, bm.lme.5)
      Model df    AIC    BIC logLik  Test L.Ratio p-value
bm.lme.4   1  9 3607.1 3650.8 -1794.6
bm.lme.5   2  7 3605.0 3638.9 -1795.5 1 vs 2  1.8374  0.399

> bm.lme.6 <- update(bm.lme.4, random = ~ I(age - 8) - 1 | subject)

```

<sup>9</sup>The large correlation between the random effects for the intercept and slope, however, suggests that the problem may be ill-conditioned.

```
> anova(bm.lme.4, bm.lme.6)
      Model df      AIC      BIC logLik   Test L.Ratio p-value
bm.lme.4    1  9 3607.1 3650.8 -1794.6
bm.lme.6    2  7 3619.9 3653.8 -1802.9 1 vs 2  16.742   2e-04
```

### Differences in the Results in R and S-PLUS

As in the analysis of the Bryk and Raudenbush data, the `lme` function in S-PLUS produces different estimates of the standard deviation of the slopes [here, the random effects for  $I(\text{age} - 8)$ ] and of the correlation between the slopes and intercepts. This time, however, the R version of the software does a slightly better job of maximizing the restricted likelihood.

As before, this difference is evidence of an ill-conditioned problem, as may be verified by examining confidence intervals for the estimates: In R, the confidence interval for the correlation runs from nearly  $-1$  to almost  $+1$ , while `lme` in S-PLUS is unable to calculate the confidence intervals because the estimated covariance matrix for the random effects is not positive-definite. Also as before, the random slopes can be removed from the model (see below).

To get a more concrete sense of the fixed effects, using model `bm.lme.5` (which includes autocorrelated errors, random intercepts, but not random slopes), I 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  -0.590722  0.58068
2  10 patient   0.009735  0.92344
3  12 patient   0.610192  1.44313
4  14 patient   1.210650  2.23109
5  16 patient   1.811107  3.42578
6  18 patient   2.411565  5.23718
7   8 control  -0.307022  0.72498
8  10 control  -0.161444  0.81080
9  12 control  -0.015866  0.90573
10 14 control   0.129712  1.01074
11 16 control   0.275291  1.12690
12 18 control   0.420869  1.25540
```

Specifying `level=0` in the call to `predict` produces estimates of the fixed effects. The expression

$$(2^{\text{pdata\$log.exercise}}) - 5/60$$

translates the fitted values of exercise from the  $\log_2$  scale back to hours/week.

Finally, I plot the fitted values (Figure 8):

```
> 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(locator(1), c('Patients', 'Controls'), pch=c(19, 22), lty=c(1,2), lwd=2)
>
```

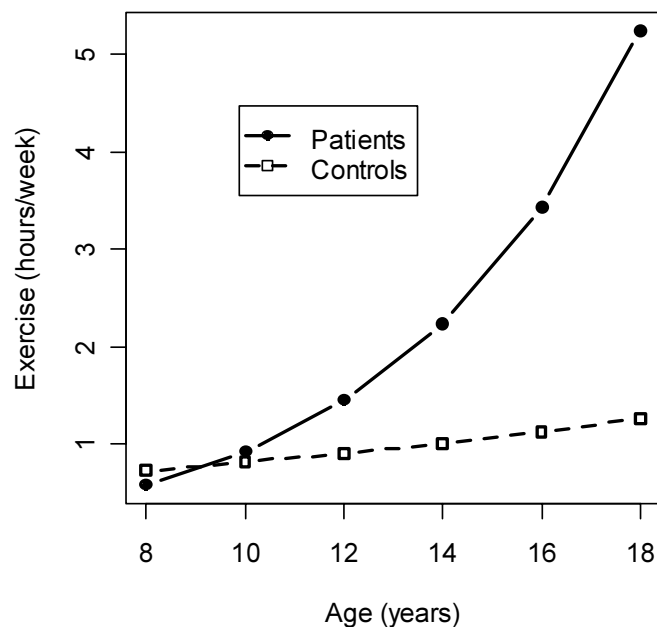


Figure 8: Fitted values representing estimated fixed effects of **group**, **age**, and their interaction.

Notice that the legend is placed interactively with the mouse. 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.

### Reminder: Use of **legend** in S-PLUS

The plotting characters for the **legend** function in S-PLUS are specified via the **marks** argument, rather than via **pch** as in R. As well, in S-PLUS, you may wish to use plotting characters 16 and 0 in place of 19 and 22.

## References

- Bryk, A. S. & S. W. Raudenbush. 1992. *Hierarchical Linear Models: Applications and Data Analysis Methods*. Newbury Park CA: Sage.
- Laird, N. M. & J. H. Ware. 1982. "Random-Effects Models for Longitudinal Data." *Biometrics* 38:963–974.
- Pinheiro, J. C. & D. M. Bates. 2000. *Mixed-Effects Models in S and S-PLUS*. New York: Springer.
- Raudenbush, S. W. & A. S. Bryk. 2002. *Hierarchical Linear Models: Applications and Data Analysis Methods*. 2nd ed. Thousand Oaks CA: Sage.
- 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.

Venables, W. N. & B. D. Ripley. 1999. *Modern Applied Statistics with S-PLUS*. 3rd ed. New York: Springer.