


Module 5: Introduction to Multilevel Modelling

R Practical

Camille Szmaragd and George Leckie¹
Centre for Multilevel Modelling

Some of the sections within this module have online quizzes for you to test your understanding. To find the quizzes:

EXAMPLE
From within the LEMMA learning environment

- Go down to the section for **Module 5: Introduction to Multilevel Modelling**
- Click "[5.1 Comparing Groups Using Multilevel Modelling](#)" to open Lesson 5.1
- Click  to open the first question

Pre-requisites

- Modules 1-4

Contents

Introduction to the Scottish Youth Cohort Trends Dataset	2
P5.1 Comparing Groups using Multilevel Modelling.....	4
P5.1.1 A multilevel model of attainment with school effects.....	5
P5.1.2 Examining school effects (residuals)	9
P5.2 Adding Student-level Explanatory Variables: Random Intercept Models	13
P5.3 Allowing for Different Slopes across Schools: Random Slope Models	18
P5.3.1 Testing for random slopes	19
P5.3.2 Interpretation of random cohort effects across schools	20
P5.3.3 Examining intercept and slope residuals for schools	20
P5.3.4 Between-school variance as a function of cohort	24
P5.3.5 Adding a random coefficient for gender (dichotomous x)	26
P5.3.6 Adding a random coefficient for social class (categorical x)	28
P5.4 Adding Level 2 Explanatory Variables	33
P5.4.1 Contextual effects	36
P5.4.2 Cross-level interactions	39
P5.5 Complex Level 1 Variation	42
P5.6 References	42

Introduction to the Scottish Youth Cohort Trends Dataset

You will be analysing data from the Scottish School Leavers Survey (SSLS), a nationally representative survey of young people. We use data from seven cohorts of young people collected in the first sweep of the study, carried out at the end of the final year of compulsory schooling (aged 16-17) when most sample members had taken Standard grades.²

In the practical for Module 3 on multiple regression, we considered the predictors of attainment in Standard grades (subject-based examinations, typically taken in up to eight subjects). In this practical, we extend the (previously single-level) multiple regression analysis to allow for dependency of exam scores within schools and to examine the extent of between-school variation in attainment. We also consider the effects on attainment of several school-level predictors.

The dependent variable is a total attainment score. Each subject is graded on a scale from 1 (highest) to 7 (lowest) and, after recoding so that a high numeric value denotes a high grade, the total is taken across subjects. The analysis dataset contains the student-level variables considered in Module 3 together with a school identifier and three school-level variables:

Variable name	Description and codes
caseid	Anonymised student identifier
schoolid	Anonymised school identifier
score	Point score calculated from awards in Standard grades taken at age 16. Scores range from 0 to 75, with a higher score indicating a higher attainment
cohort90	The sample includes the following cohorts: 1984, 1986, 1988, 1990, 1996 and 1998. The cohort90 variable is calculated by subtracting

² We are grateful to Linda Croxford (Centre for Educational Sociology, University of Edinburgh) for providing us with these data. The dataset was constructed as part of an ESRC-funded project on Education and Youth Transitions in England, Wales and Scotland 1984-2002. Further analyses of the data can be found in Croxford and Raffe (2006).

¹ This R practical is adapted from the corresponding MLwiN practical: Steele, F. (2008) Module 5: Introduction to Multilevel Modelling. LEMMA VLE, Centre for Multilevel Modelling. Accessed at <http://www.cmm.bris.ac.uk/lemma/course/view.php?id=13>.

	1990 from each value. Thus values range from -6 (corresponding to 1984) to 8 (1998), with 1990 coded as zero
female	Sex of student (1 = female, 0 = male)
sclass	Social class, defined as the higher class of mother or father (1 = managerial and professional, 2 = intermediate, 3 = working, 4 = unclassified)
sctype	School type, distinguishing independent schools from state-funded schools (1 = independent, 0 = state-funded)
schurban	Urban-rural classification of school (1 = urban, 0 = town or rural)
schdenom	School denomination (1 = Roman Catholic, 0 = non-denominational)

There are 33,988 students in 508 schools.

P5.1 Comparing Groups using Multilevel Modelling

Download the R dataset for this lesson:

From within the LEMMA Learning Environment

- Go to **Module 5: Introduction to Multilevel Modelling**, and scroll down to **R Datasets and R files**
- Right click "5.1.txt" and select **Save Link As ...** to save the dataset to your computer

Read the dataset into R using the `read.table` command and create a dataframe object named `mydata`³:

```
> mydata <- read.table(file = "5.1.txt", sep = ",", header = TRUE)
```

and use the `str` command to produce a summary of the dataset:

```
> str(mydata)
'data.frame': 33988 obs. of 9 variables:
 $ caseid : int 18 17 19 20 21 13 16 14 15 12 ...
 $ schoolid: int 1 1 1 1 1 1 1 1 1 1 ...
 $ score : int 0 10 0 40 42 4 0 0 14 27 ...
 $ cohort90: int -6 -6 -6 -6 -6 -6 -6 -6 -6 -6 ...
 $ female : int 1 1 1 1 1 1 1 1 1 1 ...
 $ sclass : int 2 2 4 3 2 2 3 4 3 2 ...
 $ sctype : int 0 0 0 0 0 0 0 0 0 0 ...
 $ schurban: int 1 1 1 1 1 1 1 1 1 1 ...
 $ schdenom: int 0 0 0 0 0 0 0 0 0 0 ...
```

³ At the beginning of your R session, you will need to set R's working directory to the file location where you saved the dataset. This can be done using the command line and the function `setwd`:

```
> setwd("C:/userdirectory")
```

Or through selecting Change dir... on the File menu.

P5.1.1 A multilevel model of attainment with school effects

We will start with the simplest multilevel model which allows for school effects on attainment, but without explanatory variables. This ‘null’ model may be written:

$$\text{score}_{ij} = \beta_0 + u_{0j} + e_{ij}$$

where score_{ij} is the attainment of student i in school j , β_0 is the overall mean across schools, u_{0j} is the effect of school j on attainment, and e_{ij} is a student-level residual. The school effects u_{0j} , which we will also refer to as school (or level 2) residuals, are assumed to follow a normal distribution with mean zero and variance σ_{u0}^2 .

R’s main command for fitting multilevel models is part of the additional `lme4`⁴ library which can be installed through the R Packages menu; select Install Package(s) and then select the correct Mirror and package from the scroll-down menus. As you will see, there is a variety of additional packages that can be installed with R. You only need to install a package once to your own computer. If you then want to use the package, you simply need to call it from within R prior to using the command for the first time in each R session. This can be done with the `library()` function and in this case `library(lme4)`.

```
> library(lme4)
Loading required package: Matrix
Loading required package: lattice

Attaching package: 'lme4'
```

```
The following object(s) are masked from package:stats :
```

```
AIC
```

The output informs us that R has loaded two additional packages `Matrix` and `lattice` which are required for the `lme4` package to work. We are also told that the `AIC` object is masked from a third package `stats`. This means that when you call these commands you need to specify from which packages you are calling them from. This is done by using the name of the package followed by two ‘:’ and then the name of the command; for instance in this case `stats::AIC`.

We will use the `lmer()` function from the `lme4` library to fit the above model. The syntax for this function is very similar to the syntax used for the `lm()` function for multiple regression which we introduced in Module 3.⁵ Below we choose to store the model as a new object called `nullmodel`:

```
> nullmodel <- lmer(score ~ (1 | schoolid), data = mydata, REML = FALSE)
```

⁴ `lme4` is a package developed by Douglas Bates and Martin Maechler for fitting linear and generalized linear mixed-effect models.

⁵ To obtain details of the different options available for the `lmer()` function, just type `help("lmer")`

The response variable (`score`) follows the command which is then followed by a `~` and then by a list of fixed part explanatory variables (excluding the constant as this is included by default)⁶. The above model contains only an intercept and so no fixed part explanatory variables are specified. The level 2 random part of the model is specified in brackets by the list of random part explanatory variables (the constant has to be explicitly specified by `1`, followed by a single vertical bar `|` and then by the level 2 identifier (`schoolid`). The `data` option specifies the dataframe being used to fit the model. The `REML = FALSE` option is used to request maximum likelihood estimation (as opposed to the default of restricted maximum likelihood estimation).

⁶ Note, to omit the constant you need to add `-1` to the right-hand side of the “`~`” sign.

We then display the results using the `summary` command, which gives the following output:

```
> summary(nullmodel)
Linear mixed model fit by maximum likelihood

Formula: score ~ (1 | schoolid)
Data: mydata
AIC      BIC    logLik deviance REMLdev
286545 286570 -143270  286539  286539

Random effects:
Groups   Name      Variance Std.Dev.
schoolid (Intercept) 61.024   7.8118
Residual                258.357 16.0735
Number of obs: 33988, groups: schoolid, 508

Fixed effects:
              Estimate Std. Error t value
(Intercept)  30.6006     0.3693   82.85
```

Before interpreting the model, we will discuss the estimation procedure that `lmer` uses.⁷ The estimation procedure used by both MLE and REML is based on optimizing a function of the log likelihood using penalized iteratively re-weighted least squares. The log-likelihood is evaluated using an adaptive Gauss-Hermite approximation, which, when using the default value of one, reduces to the Laplacian approximation. This default approximation can be changed by using the `nAGQ = n` option, where `n` is an integer greater than one, representing the number of points used for evaluating the adaptive Gauss-Hermite approximation. The greater the value of `n`, the more accurate the evaluation of the log-likelihood, but the longer it takes to fit the model.

The output of `lmer` consists of three parts. The first part under `Formula:` and `Data:` reports a range of summary statistics (AIC, BIC, LogLik,...). The second part under `Random effects:` summarises the variance and standard deviation of each random effect (including the level 1 model residuals). Underneath the random effects table, the total number of observations is provided along with the number of units (or groups) for each higher level in the model. Here, schools are our only higher level and the output reports that we have 508 different schools. The final part of the output is the `Fixed effects:` table which reports the parameter estimate (Estimate) standard error (Std. Error) and t-value (t value), for each parameter in the model. For models with more than one fixed part explanatory variable (including the intercept), a correlation table between these variables is also provided underneath the table of parameter estimates (see later examples).

The overall mean attainment (across schools) is estimated as 30.60. The mean for school j is estimated as $30.60 + \hat{u}_{0j}$, where \hat{u}_{0j} is the school residual which we will estimate in a moment. A school with $\hat{u}_{0j} > 0$ has a mean that is higher than

⁷ For further details see the PDF vignettes available on the `lme4` website <http://cran.r-project.org/web/packages/lme4>, in particular the vignette entitled "Computational Methods" which deals with the statistical theory.

average, while $\hat{u}_{0j} < 0$ for a below-average school. (We will obtain confidence intervals for residuals to determine whether differences from the overall mean can be considered 'real' or due to chance.)

Partitioning variance

The between-school (level 2) variance `schoolid` (Intercept) in attainment is estimated as $\hat{\sigma}_{u0}^2 = 61.02$, and the within-school between-student (level 1) variance `Residual` is estimated as $\hat{\sigma}_e^2 = 258.36$. Thus the total variance is $61.02 + 258.36 = 319.38$.

The variance partition coefficient (VPC) is $61.02/319.38 = 0.19$, which indicates that 19% of the variance in attainment can be attributed to differences between schools. Note, however, that we have not accounted for intake ability (measured by exams taken on entry to secondary school) so the school effects are not value-added. Previous studies have found that between-school variance in *progress*, i.e. after accounting for intake attainment, is close to 10%.

Testing for school effects

To test the significance of school effects, we can carry out a likelihood ratio test comparing the null multilevel model with a null single-level model. To fit the null single-level model, we need to remove the random school effect:

```
scoreij = β0 + eij

> fit <- lm(score ~ 1, data = mydata)
> summary(fit)

Call:
lm(formula = score ~ 1, data = mydata)

Residuals:
    Min       1Q   Median       3Q      Max
-31.095 -12.095  1.905  13.905  43.905

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  31.09462     0.09392   331.1  <2e-16 ***

Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 17.31 on 33987 degrees of freedom
```

The likelihood ratio test statistic is calculated as two times the difference in the log likelihood values for the two models.

You can obtain the log likelihood value for a model with the `logLik` command:

```
> logLik(nullmodel)
'log Lik.' -143269.5 (df=3)

> logLik(fit)
'log Lik.' -145144.4 (df=2)
```

$LR = 2(-143269.5 - -145144.4) = 3750$ on 1 d.f. (because there is only one parameter difference between the models, σ_{u0}^2).

Bearing in mind that the 5% point of a chi-squared distribution on 1 d.f. is 3.84, there is overwhelming evidence of school effects on attainment. We will therefore revert to the multilevel model with school effects.⁸

P5.1.2 Examining school effects (residuals)

To estimate the school-level residuals \hat{u}_{0j} and their associated standard errors, we use the `ranef` command with the `postVar` option. This creates a random effects object, containing the variance-covariance matrix in the `postVar` attribute.

```
> u0 <- ranef(nullmodel, postVar = TRUE)

> u0se <- sqrt(attr(u0[[1]], "postVar")[1, , ])
```

The 508 school level residuals are stored in `u0`, a type of R object called a list. It is actually a list of lists. The first and unique element of the list, `u0[1]`, is the list corresponding to the first set of random effects. We can obtain a description of `u0[1]` by using the `str` command:

```
> str(u0[1])
list of 1
 $ schoolid:'data.frame':      508 obs. of  1 variable:
  ..$ (Intercept): num [1:508] -11.84 3.21 3.4 -7.42 3.43 ...
  ..- attr(*, "postVar")= num [1, 1, 1:508] 5.71 1.7 2.24 4.29 2.66 ...
```

The first line of the output confirms that there are 508 schools in the data. The second line (`$ (Intercept)`) lists the school effects while the third line corresponding to the `"postVar"` attribute lists their associated posterior variances.

In our case there is only one set of random effects and therefore `u0[1]` is a list of only one object, `u0[[1]]`. `u0[[1]]` is itself a dataframe containing the school-level residuals and the "posterior variances" of these residuals within the attribute `postVar`. To access the elements of this dataframe, we need to use two sets of square brackets as opposed to one set of square bracket.

⁸ Note that this test statistic has a non-standard sampling distribution as the null hypothesis of a zero variance is on the boundary of the parameter space; we do not envisage a negative variance. In this case the correct p-value is half the one obtained from the tables of chi-squared distribution with 1 degree of freedom.

You can see the difference the second pair of square brackets makes by using the `str` command:

```
> str(u0[[1]])
'data.frame':      508 obs. of  1 variable:
 $ (Intercept): num -11.84 3.21 3.4 -7.42 3.43 ...
 - attr(*, "postVar")= num [1, 1, 1:508] 5.71 1.7 2.24 4.29 2.66 ...
```

The output seems similar to the previous one, except that the name of the higher level random-effect is no longer specified.

R uses lists in particular to associate specific attributes to data objects such as dataframes or vectors. By default, R returns a dataframe for the random effects, even when there is only one set of random effect.

As there is only one set of random effects, the `postVar` attribute only contains the "posterior variance" of each school-level residual. To access this set of variances, we look into the attribute `postVar` of the dataframe `u0[[1]]`. This returns a three-dimensional array with the third dimension referring to each individual residual. To reduce this array into a simple vector containing the "posterior variances" for each residual, we use `attr(u0[[1]], "postVar")[1, ,]`. To view the first few elements of this vector, we can use the `head` command:

```
> head(attr(u0[[1]], "postVar")[1, , ])
[1] 5.714615 1.698067 2.243282 4.291562 2.658550 1.969786
```

The school-level residuals and their standard errors have been calculated and stored for each individual school. We can therefore calculate summary statistics and produce graphs based on these data.

Next we create a dataframe containing an identifier, residual and standard error for every school:

```
> schoolid <- as.numeric(rownames(u0[[1]]))

> u0tab <- cbind(schoolid, u0[[1]], u0se)

> colnames(u0tab) <- c("schoolid", "u0", "u0se")
```

We then sort this table by ascending order based on the values of `u0`:

```
> u0tab <- u0tab[order(u0tab$u0), ]
```

and create a new column containing the ranks:

```
> u0tab <- cbind(u0tab, c(1:dim(u0tab)[1]))

> colnames(u0tab)[4] <- "u0rank"
```

We finally reorder the table based on the school identifier:

```
> u0tab <- u0tab[order(u0tab$schoolid), ]
```

To see the school residual, standard error and ranking for a particular school, we can list the data by using the indexing structure of the R dataframe. Here we do this for the first 10 schools in the data.

```
> u0tab[1:10, ]
  schoolid      u0      u0se u0rank
1         1 -11.844059 2.390526    37
2         2  3.207216 1.303099   337
3         3  3.396920 1.497759   344
4         4 -7.416852 2.071609    73
5         5  3.427138 1.630506   345
6         6 12.437109 1.403491   487
7         7 -1.652372 1.460226   199
8         8 20.984041 2.021872   508
9         9 -8.693975 6.438403    59
10        10  1.737830 1.904961   291
```

From these values we can see, for example, that school 1 had an estimated residual of -11.84 which was ranked 37, i.e. 37 places from the bottom. For this school, we estimate a mean score of $30.60 - 11.84 = 18.76$. In contrast, the mean for school 8 (ranked 508, the highest) is estimated as $30.60 + 20.98 = 51.58$.

Finally, we use the `plot` and `segments` commands to produce a 'caterpillar plot' to show the school effects in rank order together with 95% confidence intervals.

We start by creating the plot but without plotting any data

```
> plot(u0tab$u0rank, u0tab$u0, type = "n", xlab = "u_rank", ylab = "conditional
modes of r.e. for school_id:_cons")
```

By using the `type = "n"` option, we create the axis for the plot but prevent any data from being plotted.

We then add to the plot the 95% confidence intervals by using the `segments` command:

```
> segments(u0tab$u0rank, u0tab$u0 - 1.96*u0tab$u0se, u0tab$u0rank, u0tab$u0 +
1.96*u0tab$u0se)
```

The `segments` command takes at least four arguments corresponding to the pair of (x,y) coordinates, corresponding to the two end points of the segments, or in this case the lower and upper values of the 95% confidence intervals.

This document is only the first few pages of the full version.

To see the complete document please go to learning materials and register:

<http://www.cmm.bris.ac.uk/lemma>

The course is completely free. We ask for a few details about yourself for our research purposes only. We will not give any details to any other organisation unless it is with your express permission.