Further Statistical Methods: Practical HT 2009

Department of Statistics, University of Oxford

Tom A.B. Snijders

February 11, 2009

This practical shows you how to generate many of the results used in lectures. It gives indications on how to employ R to apply a number of the methods treated in the lectures about multilevel analysis, and it is a preparation for the assessed practical later in the term.

The following literature is useful for this practical. Only the first one is perhaps fit to be printed, the others (and perhaps the first) are useful mainly as reference material. Just glance at them – do not waste paper to print them out. The literature and links are also given at the website of the course,
`http://www.stats.ox.ac.uk/~snijders/fsm.htm`

- John Fox, *Linear Mixed Models. Appendix to 'An R and S-PLUS Companion to Applied Regression'*.
  `http://cran.r-project.org/doc/contrib/Fox-Companion/appendix-mixed-models.pdf`
  (one line!)
- Douglas Bates, *Examples from Multilevel Software Comparative Reviews.*
  `http://finzi.psych.upenn.edu/R/library/mlmRev/doc/MlmSoftRev.pdf`
- Description of the lme4 package.
  `http://www.stats.ox.ac.uk/~snijders/lme4.pdf`
- Description of the nlme package.
  `http://www.stats.ox.ac.uk/~snijders/nlme.pdf`

1 Data exploration

This session will show you how to fit multilevel models using the nlme and lme4 packages of R.[1] It may be noted that the nlme package is more extensive, and documented in Pinheiro and Bates (2000). However, lme4 is the more recent package and is expected to be extended further.

The data set is the data used in Snijders & Bosker (1999) which was used also in the lectures. It is available in Stata format, and can be imported into R as follows:

```
library(foreign)
mlbook <- read.dta("mlbook1.dta")
```

Have a look at what is in the data set:

```
mlbook[1:10,]
names(mlbook)
```

The description of the data set is given in Snijders & Bosker (1999), Section 4.3. We shall be making some of the tables given also in the handout (file `MLB_S.pdf`).

But first let us explore the data. A main aspect of applying multilevel models is to distinguish between within-group and between-group regressions (see lecture notes, Section 4.5 of Snijders & Bosker 1999); this should help you avoid falling into the trap called the *ecological fallacy* (Snijders & Bosker 1999, Section 3.1-3.2). This is carried out in practice mainly by calculating group means of level-one predictors, and using these as additional predictor variables.

In this data set we calculate school means of some of the variables as follows.

```
attach(mlbook)
schdata <- aggregate(mlbook, by = list(schoolnr), mean)
schoolsb <- subset(schdata,
   select=c(schoolnr,langpost,iq.verb,ses,mixedgra,groupsiz))
names(schoolsb)<-c("school","meanlp",
   "meaniq","meanses","combi", "grs")
schoolsz <- tapply(1:2287,schoolnr,length)
mlbook$meanlp <- rep(schoolsb$meanlp,schoolsz)
mlbook$meaniq <- rep(schoolsb$meaniq,schoolsz)
mlbook$meanses <- rep(schoolsb$meanses,schoolsz)
attach(mlbook)
```

[1] In preparing the session I made use of the text *Linear Mixed Models. Appendix to 'An R and S-PLUS Companion to Applied Regression'* by John Fox, available at `http://cran.r-project.org/doc/contrib/Fox-Companion/appendix-mixed-models.pdf`, and R exercises using this data set put together by Marijtje van Duijn.

The data frames schdata and schoolsb are intermediate data frames used for storing the school means, which are then added to the mlbook data frame. Look up the definition of tapply to understand what this function does. Have a look at the data produced and understand the interpretation of the variables meanlp, meaniq, and meanses.

What is the total number of pupils in the data set (length of the vector 'schoolnr') and what is the total number of schools (length of the vector 'school')?

To have an exploratory look at the data, we take a sample of 25 schools and define a grouped data set. (It is possible also to use all schools but then the sheer number of plots-per-school becomes too large.) To understand the function groupedData, look up this command in the description of the nlme library (which is available at the website of the course).

```
library(lattice)
library(nlme)
samp25<-sample(unique(schoolnr),25)
sample25 <- groupedData(langpost ~ iq.verb | schoolnr,
          data = mlbook[is.element(schoolnr, samp25), ])
xyplot(langpost~iq.verb|schoolnr, data=sample25,
    main="Exploring 25 random schools",
    panel=function(x,y){
    panel.xyplot(x,y)
    panel.loess(x,y,span=1)
    panel.lmline(x,y,lty=2)
    }
       )
```

The xyplot makes plots for each value of schoolnr separately. This is the reason for taking no more than 25 schools. What are the school numbers in your random sample?

What impression does the graph give you with respect to the variability between schools of intercept and slope?

2 Fitting a multilevel model

For the estimation of multilevel models we will first use the lme4 library developed by Douglas Bates. The name stands for Linear Mixed Effects, but the package also contains non-linear models. Mixed Effects refers to models with both fixed and random effects. Thus we need the command

```
library(lme4)
```

If this package is not available you first have to install it:

```
install("lme4")
```

For all fitted models in this session, determine from the output the outcome of the tests for the various included fixed effects, and the parameter estimates with their interpretations. With 'outcome of the test' is meant: test statistic, of which you should know the reference distribution (i.e., distribution or approximate distribution under the null hypothesis), p-value, and the conclusion whether the null hypothesis that the parameter is 0 is rejected at the 5% significance level.

The basic functions for fitting a mixed model are lmer() and lmer2(). They resemble the function lm() used to estimate a regression model, but are more elaborate due to the more complex model. These functions use the same notation to define a model, but also need the specification of a random part. Use the help function (or the command ?lmer) to get a little more insight into this function. More documentation of the lme4 library can be found at the CRAN website. The list of commands for lme4 is also at the website of the course.

2.1 Empty model and Random Intercept model

To fit the empty model we use the commands

```
print(model0 <- lmer2(langpost ~ 1 + (1|schoolnr)))
show(vc <- VarCorr(model0))
```

Note that 1 denotes the intercept; this is implied when other, non-trivial, variables are included in the model. The notation (1|schoolnr) denotes the random intercept at the school level.
Look up the meaning of the function VarCorr in the lme4 documentation.

It is possible that the call to VarCorr leads to an error message because of a conflict with the library nlme. In that case, one solution is to exit from R and start again, do everything again except attaching the nlme library and the calls to groupedData, sample, and xyplot. Then VarCorr will not lead to problems.

The lme4 package produces REML estimates whereas the tables in the handouts give the ML estimates; further, the standard errors are calculated in a different way. This leads to some minor differences.

ML estimates can be requested by adding model = "ML". Compare the results in Table 4.1 which is contained both in the handout and in Snijders & Bosker (1999), with the results from lme4.

The model with a fixed effect of IQ is fitted by

```
print(model1 <- lmer2(langpost ~ iq.verb + (1|schoolnr),
                      data=mlbook))
show(vc <- VarCorr(model1))
```

This is different from the result in Table 4.2 because the IQ variable is centered there, and not in this data set. To calculate a centered IQ variable, compute

```
mlbook$iqdev <- mlbook$iq.verb - mean(mlbook$iq.verb)
print(model2 <- lmer2(langpost ~ iqdev + (1|schoolnr),
                       data=mlbook))
show(vc <- VarCorr(model2))
```

How large is the overall mean of iq.verb? Give the explanation why the intercept is 11.17 first and 40.61 now.

To obtain the values of the estimated fixed effects and the level-2 empirical Bayes residuals (the usual residuals for the level-two random effects, hence the name 'ranef'), use the commands

```
fixef(model1)
ranef(model1)
```

The ('diagnostic') covariance matrix of the empirical Bayes residuals, in the form of the diagonal blocks (covariances between residuals of different groups are zero, as the fact that parameters are estimated is not accounted for in these covariances) can be obtained from lmer, but not from lmer2, as

```
model1@bVar
```

However, bVar is a list of variance covariance matrices so the vector still must be taken out; this is done here by model2n@bVar$schoolnr. Hence we use lmer and give the commands

```
print(model2n <- lmer(langpost ~ iqdev +  meanIQ
               + ses + sex +(1|schoolnr),data=mlbook))
ebe2 <- model2n@ranef
bv2 <- model2n@bVar$schoolnr
sebe2 <- ebe2/sqrt(bv2)
qqmath(sebe2)
```

which produce a normal quantile plot for the standardized residuals. Does this plot give support for the normality of the level-2 random effects?

2.2 Random slope models

The random slope model of Table 5.1 can be fitted by

```
print(model3 <- lmer2(langpost ~ iqdev + iqmean
                + (iqdev|schoolnr)), data=mlbook)
show(vc <- VarCorr(model3))
```

To fit the model of Table 5.2, note that the variable called Z is group size, centered at the global mean. Also the group mean of IQ is centered. These centered variables are obtained by

```
mean(mlbook$meanIQ)
mean(mlbook$groupsiz)
mlbook$meanIQ <- mlbook$meanIQ - mean(mlbook$meanIQ)
mlbook$groupsiz <- mlbook$groupsiz - mean(mlbook$groupsiz)
```

This also gives us the original values of the global means for future reference, if needed. What is the difference between the variables iqdev and meanIQ?

Now we can fit the model; again, the results are slightly different from those in the table because we now use REML estimation.

```
print(model4 <- lmer2(langpost ~ iqdev + meanIQ +
   groupsiz  + iqdev*groupsiz + (iqdev|schoolnr)))
```

The random slope of IQ can be tested by a likelihood ratio test. Note that this has to be based on the ML deviances, not on the REML deviances. The deviances can be obtained from functions such as

```
deviance(model1)
logLik(model1)
```

A direct likelihood ratio test can also be requested from

```
print(model5 <- lmer2(langpost ~ iqdev + meanIQ +
   groupsiz  + iqdev*groupsiz + (1|schoolnr)))
anova(model5,model4)
```

Report both the test obtained from the anova function and the one-sided likelihood ratio test obtained from the two deviances as treated in the lecture. Make sure that the ML deviances are being used. Which is the better of these tests, and why?

2.3 Model checking with nlme

Package nlme still has more model checking capacities than lme4. Therefore we continue with the earlier package. If this gives any problems, it may be best to exit R and then start a session again where you load only nlme as a library and not lme4.

Take care that the variables iqdev and meanIQ are available again. Fitting and comparing mixed models in nlme is done by the lme function as follows.

```
summary(model1n <- lme(langpost ~ iqdev + ses + meanIQ
    + sex, random = ~ 1 |schoolnr))
summary(model2n <- lme(langpost ~ iqdev + ses + meanIQ
    + sex, random = ~ iqdev |schoolnr))
anova(model1n, model2n)
```

Note the difference in model syntax between nlme and lme4. If you are not sure how exactly these models are specified, consult the help function or the description of lme in nlme.

Assumptions about normal distributions of residuals can be checked graphically by the normal quantile plots, using ∼ranef for the level-two random effects and ∼resid for the level-one residuals:

```
qqnorm(model2n, ~ranef(.))
qqnorm(model2n, ~resid(.))
```

It is instructive to make comparisons with the fits obtained by fitting OLS models to each school separately. These fits, and an object comparing the mixed effects model and the separate OLS models, are requested and compared by the following functions. Note that the coefficients obtained in coef(model2n), given that model2n is a lme object, are the sum of the fixed effects and the empirical Bayes estimates of the level-2 random effects.

```
olse2list <- lmList(langpost ~ iqdev +  ses + meanIQ
          + sex   |  schoolnr, data=mlbook)
compm2 <- compareFits(coef(olselist), coef(model2n))
olse2list[1:3]
```

The last command shows you how the contents of the olse2list object are organized. Why are the coefficients for meanIQ all NA?

The sets of coefficients are compared by the function

```
plot(compm2, mark=fixef(model2n))
```

where the mark=fixef(model2n) parameter requests a line at the fixed effect estimates of the mixed model. The plots are a bit too crowded, with 131 schools, but it is clearly visible that the OLS estimates are much more variable than the mixed model estimates. If this were desired, a smaller part of the data could be obtained by taking a sample as above and then using only the data specified by

```
data = mlbook[is.element(schoolnr, samp25), ]
```

This will, however, not be pursued here.

The within-group OLS residuals can be obtained now by the command

```
resi2 <- residuals(olse2list)
```

These can be used for checking linearity of effects as follows.

First, plots of the relation between the language test and IQ can be obtained from

```
plot(iq.verb,langpost)
plot(jitter(iq.verb),jitter(langpost))
lines(lowess(iq.verb,langpost, f=.2), lwd=3)
```

The first, 'raw' plot is not so helpful because many pupils may have the same values on iq.verb and langpost; these are not really continuous variables. And several individuals with the same values give just one plotting symbol. Adding the jitter, which is a small random perturbation to each data point, gives a picture with more black where there are more data. The lowess line suggests a slight departure from linearity. However, it is possible that this is remedied already by controlling for the other variables (ses and sex), as we do in the model. Therefore we make a similar plot, but now for the residuals.

```
resi2 <- residuals(olse2list)
plot(jitter(iqdev),resi2,
       lines(lowess(iqdev,resi2, f=.2), lwd=3))
```

Here we see that the slight departure from linearity remains. This could be resolved by considering a non-linear transform of iqdev. This was done also in the handouts for the course and in Snijders & Bosker (1999). The plots produced here point to a slightly different function. There is no time, however, to pursue this in this class.

3 And finally....

It is important to be able communicate results of a statistical analysis to an audience that does not primarily have a statistical interest. A final question of the assessed practical will be as follows.

"Write a summary, with a maximum length of one page, of your substantive conclusions of the whole data analysis in a non-technical way that is understandable to persons who know nothing about multilevel analysis."

Try to write such a conclusion for the analysis you have completed now. You can discuss with a friend who is not a statistician about your text, and try to write it in such a way that your friend indeed understands some of it!

Another aspect of reporting is to present a good table with estimates and standard errors. The tables given in the handouts are a bit extensive. Look in the paper by Levels, Dronkers, and Kraaykamp (2008) (see course website) for a more application-oriented way of presenting the results in tables.