Multilevel Modeling

Answer Key: Homework 3

Andy Stone

October 06, 2016

11.4

a.

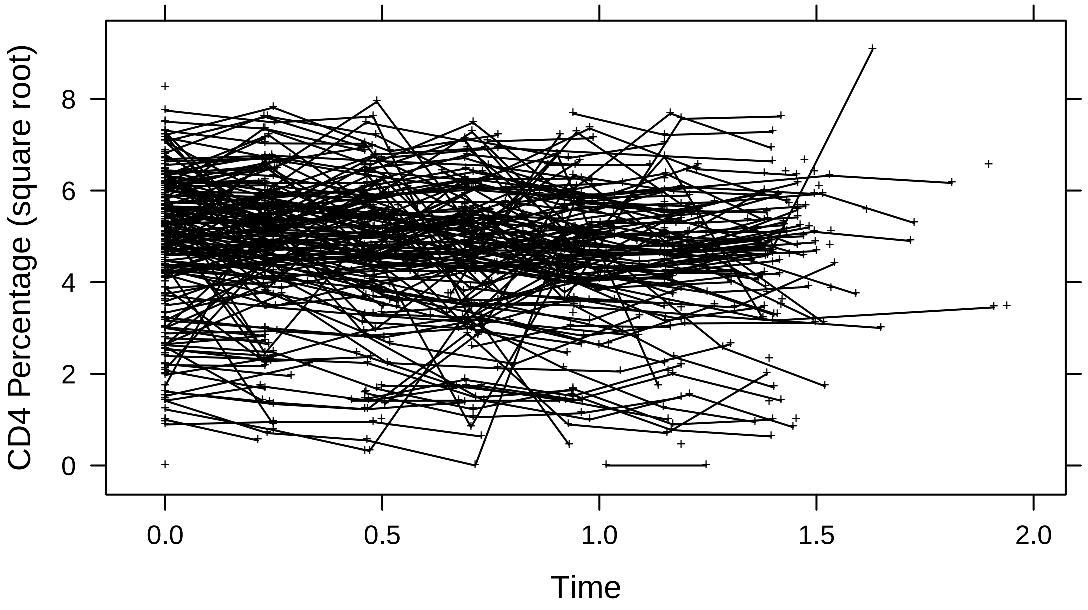
```
# Loading HIV data
hiv.data <- read.csv("allvar.csv", header=TRUE)
# Square root transformation of the CD4PCT
hiv.data$rootCD4 <- sqrt(hiv.data$CD4PCT)
# Creation of time variable
hiv.data$time <- hiv.data$visage - hiv.data$baseage
```

I graph the square root of CD4 for each child against time in Figure 1. I connect the tick marks corresponding to each child. This graph is very crowded. Some of you plotted a sample of these lines, that is a nice strategy.

It appears, from looking at these lines, that there is a slight trend downwards in `rootCD4`, which suggests that the patients' HIV status improves over time.

```
# Code for Figure 1
library(lattice)
xyplot(rootCD4 ~ time, data=hiv.data, pch="+", group=newpid, type="b", col="black",
       scales=list(tick.number=5), xlab="Time", ylab="CD4 Percentage (square root)",
       main="Figure 1: CD4 percentage (square root) v. Time")
```

Figure 1: CD4 percentage (square root) v. Time



b.

To summarize each child's linear fit, we can run individual linear regressions for each child, regressing `rootCD4` on `time`. For those childern with only one observation, their linear regression line will have a slope of zero and be equal to the `rootCD4` value of their lone observation.

We cannot estimate these regressions for individuals who do not have a measurement for the outcome variable (`rootCD4`) or the `time` variable (this does not include those individuals with a `time` observation of zero). I omit those observations from the dataset using the `complete.cases` function.

```r
# Removing those cases that have NAs for the DV or the time variable
data.noNA.CD4 <- hiv.data[complete.cases(hiv.data[,4]),]
data.noNA.CD4 <- data.noNA.CD4[complete.cases(data.noNA.CD4[,11]),]
```

To estimate the linear fit of each child's time course, I run a for-loop that takes the observations of each unique child in the dataset and conducts a simple linear regression of `rootCD4` on `time`, and then save the coefficient estimate on the intercept and slope from this regression in the matrix I created.[1]

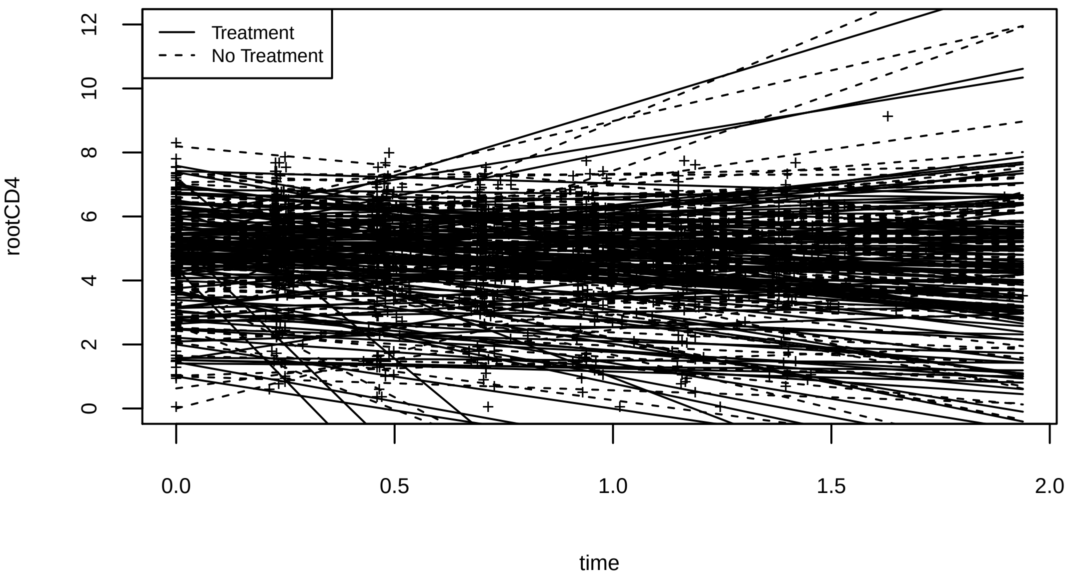
```r
# Determining number of unique patients that have no NAs for DV/time
npatients <- length(unique(data.noNA.CD4$newpid))

# Empty matrix to fill with regression coefficients
lm.mat <- matrix(data=NA, nrow=max(unique(data.noNA.CD4$newpid)), ncol=2)

# Running separate regressions for each individual
# Then, finding individual regression coefficients for each individual
for(i in unique(data.noNA.CD4$newpid)){
    # Subsetting the dataset to only observations related to child i
    tempdata <- data.noNA.CD4[which(data.noNA.CD4$newpid == i),]
    # Running the regression for child i
    templm <- lm(tempdata$rootCD4 ~ tempdata$time)
    # Saving the coefficients from this regression to the matrix
    lm.mat[i,] <- unname(templm$coefficients)
  }

# Omitting empty matrix observations with no intercept (necessary due to how I wrote the loop)
lm.mat <- lm.mat[complete.cases(lm.mat[,1]),]
```

In Figure 2 I plot the summaries of the linear fit of each child's data. As we are interested in the effect of the treatment, I differentiate between children who recieved the treatment (solid lines) and those who did not (dashed lines). At this point, there are no clear distinguishable differences in the trends of those children who recieved the treatment and those who did not.

```r
# Plotting the regression lines. Code for Figure 2
plot(rootCD4 ~ time, data=data.noNA.CD4, pch="+", ylim=c(0,12),
     main="Figure 2: Child-Specific Linear Fits", cex=0.7, cex.main=0.7,
     cex.lab=0.7, cex.axis=0.7)
for(i in 1:length(lm.mat[,1])){
    curve(lm.mat[i,1] + lm.mat[i,2]*x, lty=hiv.data$treatmnt[i], add=TRUE)
}
legend("topleft",lty=c(1,2),legend=c("Treatment", "No Treatment"), cex=0.6)
```

[1] Because of how I wrote the for-loop, there end up being 4 empty rows in the matrix corresponding to those observations with NAs I omitted above, so I omit them with the line of code following the loop.

Figure 2: Child−Specific Linear Fits



c.

To conduct the first step, I create an indicator variable for each of the individual children in the model. Then, I regress `rootCD4` on time, the indicators for each child (to allow the intercept to vary for each child), as well as the interaction between time and the groups to allow the slope on time to vary for each child. Another way to do this is to run separate regressions for each child (like in part (b) above), and getting the point estimates from these regressions.

```
# Creating indicators for each of the children
inddummies <- as.factor(data.noNA.CD4$newpid)
# The first step of the analysis
first <- lm(rootCD4 ~ time + inddummies + time*inddummies - 1, data=data.noNA.CD4)
```

Then, I save the child-specific intercept coefficients from this first model, to be used as the y_i in the second model, to the `intercepts` object. I then run a for-loop over each child to create a single variable for each child expressing whether they recieved the treatment or not, and what their `baseage` was. Then, I regress the intercept coefficients on the child-specific factors from the first model on the `treated` and `baseage` variables. The results from this model are presented below. We see that the child's base age (with a coefficient of -0.121) seems to be influential in decreasing the intercepts of each child, but not the treatment. This is to be expected, as the treatment may only be expected to affect the slope of the regression, whereas children of a higher age may just generally have lower levels of CD4.

```
# Intercept coefficients of the children
intercepts <- first$coefficients[2:(npatients + 1)]

# Creating child-specific measures of the child's treatment and baseage
treated <- NULL
baseage <- NULL

for (i in unique(inddummies)){
  treated[i] <- unique(data.noNA.CD4$treatmnt[which(data.noNA.CD4$newpid==i)])
  baseage[i] <- unique(data.noNA.CD4$baseage[which(data.noNA.CD4$newpid==i)])
}

# Second step
second <- lm(intercepts ~ treated + baseage)
display(second, digits=3)
lm(formula = intercepts ~ treated + baseage)
            coef.est coef.se
(Intercept)  4.994    0.324
treated      0.124    0.187
baseage     -0.121    0.041
---
n = 250, k = 3
residual sd = 1.480, R-Squared = 0.04
```

Finally, I regress the slope coefficients obtained in step one on the group-level predictors. We can see that neither the treatment nor the base age are significant predictors of the slope of each child's fit. This provides suggestive evidence that the treatment was not effective.

```
# Coefficients on slope term
slopes <- first$coefficients[(npatients + 1):500]

# Third step
third <- lm(slopes ~ treated + baseage)
display(third, digits=3)
lm(formula = slopes ~ treated + baseage)
            coef.est coef.se
(Intercept)  0.662    0.463
treated     -0.166    0.271
baseage     -0.041    0.060
---
n = 224, k = 3
residual sd = 2.018, R-Squared = 0.00
```

12.2

a.

I first write a model that predicts `rootCD4` as a function of `time`, allowing for the intercept to vary across children. This is done using the `lmer` function, specifying varying intercepts using the child-specific indicator variable `inddummies`. From the output of this model, we can see that the estimated coefficient on `time` is given as -0.366, with a standard error of 0.054, rendering it statistically significant at all traditional levels of significance. This means that, for an average child, a one-unit increase in `time` (meaning one year having progressed from the base age) is associated with a 0.366 unit decrease in the child's square root measure of CD4.

```
mlm.1 <- lmer(rootCD4 ~ time + (1 | inddummies), data=data.noNA.CD4)
display(mlm.1, digits=3)
lmer(formula = rootCD4 ~ time + (1 | inddummies), data = data.noNA.CD4)
            coef.est coef.se
(Intercept)  4.763    0.096
```

```
time        -0.366    0.054

Error terms:
 Groups     Name        Std.Dev.
 inddummies (Intercept) 1.399
 Residual               0.772
---
number of obs: 1072, groups: inddummies, 250
AIC = 3148.8, DIC = 3126.9
deviance = 3133.9
```

b.

Now, I extend the above model to include group-level predictors (meaning child-level predictors) for treatment and baseline age. We know the `lmer` function will only accept predictors at the individual level, so I include the individual-level measures of treatment and baseline age.

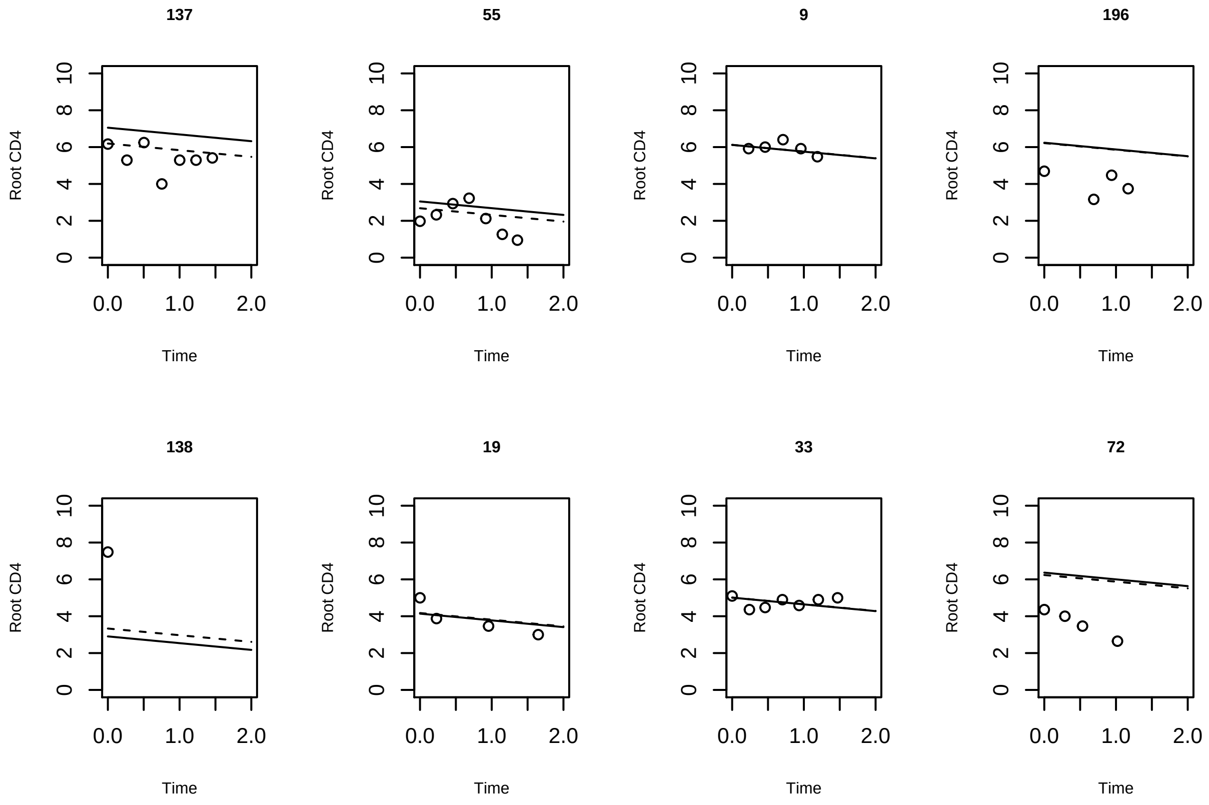
The output from this model provides us with coefficient estimates and standard errors on the individual-level `time` variable, as well as on the group-level variables `treatmnt` and `baseage`. The coefficient on `time`, -0.362, is statistically significant and similar to that in the previous model. Its interpretation is that a *ceteris paribus* one-unit increase in `time` is associated with a 0.362 unit decrease in a child's square root CD4 level. The coefficient on `treatmnt` is 0.180, and with a standard error of 0.183, is not statistically distinguishable from zero and thus we should not put much stock in the directionality of the effect. If we did, the interpretation would be that treatment increases a child's rootCD4 level. The coefficient on `baseage` is -0.119, and statistically distinguishable from zero. This suggests that a one-unit change in a child's base age is associated with a *ceteris paribus* 0.119 unit decrease in a child's root CD4 count – that is, older children are, all else equal, expected to have lower root CD4 counts.

```
mlm.2 <- lmer(rootCD4 ~ time + treatmnt + baseage + (1 | inddummies), data=data.noNA.CD4)
display(mlm.2, digits=3)
lmer(formula = rootCD4 ~ time + treatmnt + baseage + (1 | inddummies),
    data = data.noNA.CD4)
            coef.est coef.se
(Intercept)  4.906    0.317
time        -0.362    0.054
treatmnt     0.180    0.183
baseage     -0.119    0.040

Error terms:
 Groups     Name        Std.Dev.
 inddummies (Intercept) 1.375
 Residual               0.773
---
number of obs: 1072, groups: inddummies, 250
AIC = 3149.2, DIC = 3110.9
deviance = 3124.1
```

c.

Both of the models estimated above are "partial pooling" multilevel models. The difference in the two models is that the first includes no group-level predictors, while the second includes a child's base age and the child's treatment. We can first compare these two models using graphical summaries. To do so, I take a random sample of 8 of the children from the dataset. Then, I plot the observations with `rootCD4` on the y-axis and `time` on the x-axis. Then, I overplot the regression lines from both models – the first model, without the group-level predictors, is the solid line, and the second model is the dashed line. We can see that, in general, the second model (with group level predictors) seems to help the regression line to fit better to the data points than the first model. Nonetheless, without varying slope parameters, any of these individual fits of the regression line will only be a coarse approximation of the true relationship between the individual's root CD4 level and time.

```
# Code for figures displaying two models' fit
set.seed(55)
display <- as.numeric(sample(unique(inddummies), size=8)) # Sample 8 observations
par(mfrow=c(2,4))
for (i in display){
    age <- unique(data.noNA.CD4$baseage[which(data.noNA.CD4$newpid == i)])
    treat <- unique(data.noNA.CD4$treatmnt[which(data.noNA.CD4$newpid == i)])
    plot(rootCD4[which(data.noNA.CD4$newpid == i)] ~ time[which(data.noNA.CD4$newpid == i)],
        data=data.noNA.CD4, ylim=c(0,10), xlim=c(0, 2), xlab="Time", ylab="Root CD4", main=i,
        cex.lab=0.7, cex.main=0.7)
    curve(coef(mlm.1)$inddummies[i,1] + coef(mlm.1)$inddummies[i,2]*x, add=TRUE)
    curve(coef(mlm.2)$inddummies[i,1] + coef(mlm.2)$inddummies[i,2]*x +
            coef(mlm.2)$inddummies[i,3]*treat + coef(mlm.2)$inddummies[i,4]*age, add=TRUE, lty=2)
}
```



Numerically, we can compare the standard deviations of the individual-level data ($\hat{\sigma_y}$) and of the group-level data ($\hat{\sigma_\alpha}$) for each regression. The first model (without group-level predictors) has a $\hat{\sigma_y} = 0.772$ and a $\hat{\sigma_\alpha} = 1.399$. The second model (with the group-level predictors) has a $\hat{\sigma_y} = 0.773$ and a $\hat{\sigma_\alpha} = 1.375$. As the standard deviation at the group level is slightly smaller in the second model, adding the group predictors has slightly reduced the variation across groups. The variance ratio of the first model is $\frac{1.399^2}{0.772^2} = 3.284$ and the variance ratio of the second model is $\frac{1.375^2}{0.773^2} = 3.164$. There is little difference between the two models in terms of the ratio of group-level variance to individual-level varaince.

```
(sigma.hat(mlm.1)$sigma$data) # SE of data, model 1
[1] 0.7724971
(sigma.hat(mlm.1)$sigma$inddummies) # SE of group-level, model 1
(Intercept)
   1.398894

(sigma.hat(mlm.2)$sigma$data) # SE of data, model 2
[1] 0.7725771
(sigma.hat(mlm.2)$sigma$inddummies) # SE of group-level, model 2
(Intercept)
   1.374657
```

12.5

First, I subset the data as to only look at the houses in Minnesota. To fit a varying-intercept model predicting radon levels, we model logged `activity` as a function of a house-level predictor `floor` and county sample size.

Examining the output of the model, we see that the variable for `floor` is negative and statistically significant, with a coefficient of -0.700. This is sensible, as this variable takes on a value of 1 if the measurement was conducted on the first floor of the home, so we expect that moving from the basement to the first floor makes it harder to detect radon. The model also estimates a substantively small but statistically significant negative effect of county sample size on radon level, with a coefficient of -0.004, suggesting as county sample size increases by one unit (i.e., one house), a house is predicted to have a -0.004 unit decrease in log radon level.

```
# Loading the data
radon <- read.table("http://stat.columbia.edu/~gelman/arm/examples/radon/srrs2.dat",
                    header=TRUE, sep=",")
# Subsetting to Minnesota only
mn.radon <- radon[which(radon$state == 'MN'),]

# Defining radon variable
radon <- mn.radon$activity

# Logging radon, first changing zero values to 0.1 so they can be logged
log.radon <- log(ifelse(radon==0, .1, radon))

floor <- mn.radon$floor # Defining floor variable
n <- length(log.radon) # Defining sample size

# Creating unique indicator for each county
county.indicator <- as.factor(mn.radon$county)

# Creating variable determining sample size of county. Group-level variable, however,
# specified at individual level as lmer() only accepts individual-level predictors
countysize <- NULL
for(i in 1:n){
    number <- mn.radon$cntyfips[i]
    countysize[i] <- length(which(mn.radon$cntyfips == number))
  }
```

```
# Varying-intercept model of radon w/ county sample size as group-level predictor
radon.mod <- lmer(log.radon ~ floor + countysize + (1 | county.indicator))
summary(radon.mod)
Linear mixed model fit by REML ['lmerMod']
Formula: log.radon ~ floor + countysize + (1 | county.indicator)

REML criterion at convergence: 2177

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.4625 -0.6170  0.0248  0.6300  3.4371

Random effects:
 Groups           Name        Variance Std.Dev.
 county.indicator (Intercept) 0.09389  0.3064
 Residual                     0.57211  0.7564
Number of obs: 919, groups:  county.indicator, 85

Fixed effects:
             Estimate Std. Error t value
(Intercept)  1.532301   0.059245  25.864
floor       -0.700124   0.070431  -9.941
countysize  -0.004479   0.001960  -2.285

Correlation of Fixed Effects:
           (Intr) floor
floor      -0.281
countysize -0.543  0.059
```