Mixed Models For Continuous Outcomes Using R: Practical

Shaun Scholes

05/03/2021

# Learning objectives

The learning objectives for this Session are as follows:

* Learn concept/structure of hierarchical data
* Learn how random effects or mixed models handle hierarchical data by decomposing variance into different levels of the hierarchy
* Be able to build and interpret different random effects models suitable for hierarchical data

## Datasets

Download the Stata datasets from the course moodle page:

* Wave\_1\_elsa.dta
* Wave\_2\_elsa.dta
* Wave\_3\_elsa.dta
* Wave\_4\_elsa.dta
* Wave\_5\_elsa.dta
* Wave\_6\_elsa.dta
* Wave\_7\_elsa.dta

and save them in your working directory.

The datasets for this practical (used also in other sessions) come from the English Longitudinal Study of Ageing (ELSA): a study of persons aged 50+ living in private households in England at April 2002. Each of the seven datasets are a reduced dataset, containing just the variables we need.

For the purposes of this session, the variables we will be analysing include the time-varying outcome variable:a quality of life (QoL) score measured using the CASP-19 scale (**QoLscore**): i.e. QoLscore0 for the score at wave 1 (baseline); QoLscore1 for the computed score at wave 2, etc. QoLScore at each wave ranges from 0 to 57: higher scores indicate a higher reported quality of life.

Other variables important to consider in this session are:

* sex (**dhsex**)
* a memory test score as a measure of cognitive function (**cflisen**). Memory at each wave was measured using a word-list learning test, in which 10 words were presented orally to study participants who were then asked to recall as many as possible immediately after reading them. Plausible values range therefore from (0 to 10): higher scores indicate better cognitive function.
* The variable **idauniq** is the unique personal identifier that we will use to merge datasets and inform Stata that the data is structured into different levels of the hierarchy: that is, measurement occasions are nested within individuals.

The goals of this analysis are to explore the:

**Fixed part of the model** (population-average): Build a model to examine whether quality of life scores are associated with memory scores and sex, and whether the population average slope of the (cognitive function - QoL) association differs by sex.

**Random part of the model**: As this is hierarchical data, we will examine the suitability of adding different random effect terms to accommodate unobserved heterogeneity across persons in both the initial levels of Y (when X=0), and in the slope of the X-Y association (i.e. random intercepts and random slopes, respectively).

# Part 1 - Preparing the data for longitudinal analysis

## Install relevant packages.

#REMINDER: you do not need to do this every time if you are using   
#your personal laptop and you have already installed the packages;   
#BUT you will need to do this every time if   
#you are using a UCL computer  
#install.packages("readstata13")  
#install.packages("psych")  
#install.packages("tidyverse")  
#install.packages("dplyr")  
#install.packages("rlang")  
#install.packages("tidyr")  
#install.packages("lme4")  
#install.packages("lmtest")  
#install.packages("effects")  
#install.packages("summarytools")  
#install.packages("Hmisc")  
#install.packages("sjstats")

Load the relevant packages.

#Load relevant packages   
library(readstata13)  
library(psych)  
library(tidyverse)  
library(dplyr)  
library(rlang)  
library(tidyr)  
library(lme4)  
library(lmtest)  
library(effects)  
library(summarytools)  
library(Hmisc)  
library(sjstats)

## Read in ELSA data

ELSA\_wave1 <- haven::read\_dta("C:/Temp/ELSAdatasets/Wave\_1\_elsa.dta")  
ELSA\_wave2 <- haven::read\_dta("C:/Temp/ELSAdatasets/Wave\_2\_elsa.dta")  
ELSA\_wave3 <- haven::read\_dta("C:/Temp/ELSAdatasets/Wave\_3\_elsa.dta")  
ELSA\_wave4 <- haven::read\_dta("C:/Temp/ELSAdatasets/Wave\_4\_elsa.dta")  
ELSA\_wave5 <- haven::read\_dta("C:/Temp/ELSAdatasets/Wave\_5\_elsa.dta")  
ELSA\_wave6 <- haven::read\_dta("C:/Temp/ELSAdatasets/Wave\_6\_elsa.dta")  
ELSA\_wave7 <- haven::read\_dta("C:/Temp/ELSAdatasets/Wave\_7\_elsa.dta")

#Look at one of the datasets in more detail  
#Look at the first 6 rows   
head(ELSA\_wave1)

## # A tibble: 6 x 9  
## idauniq dhsex wpdes edqual indager depression depressed cflisen0  
## <dbl> <dbl+l> <dbl+l> <dbl+lb> <dbl+l> <dbl> <dbl> <dbl>  
## 1 100035 2 [Fem~ 1 [ret~ 2 [A-L~ 67 3 0 NA  
## 2 100036 2 [Fem~ 2 [in ~ 2 [A-L~ 53 1 0 NA  
## 3 100037 2 [Fem~ 1 [ret~ 2 [A-L~ 68 1 0 NA  
## 4 100038 2 [Fem~ 1 [ret~ NA 64 NA 1 NA  
## 5 100039 1 [Mal~ 2 [in ~ 2 [A-L~ 53 0 0 NA  
## 6 100040 2 [Fem~ 2 [in ~ 2 [A-L~ 48 0 0 NA  
## # ... with 1 more variable: QoLscore0 <dbl>

#Check column names  
colnames(ELSA\_wave1)

## [1] "idauniq" "dhsex" "wpdes" "edqual" "indager"   
## [6] "depression" "depressed" "cflisen0" "QoLscore0"

#Check class of variables  
lapply(ELSA\_wave1, class)

## $idauniq  
## [1] "numeric"  
##   
## $dhsex  
## [1] "haven\_labelled" "vctrs\_vctr" "double"   
##   
## $wpdes  
## [1] "haven\_labelled" "vctrs\_vctr" "double"   
##   
## $edqual  
## [1] "haven\_labelled" "vctrs\_vctr" "double"   
##   
## $indager  
## [1] "haven\_labelled" "vctrs\_vctr" "double"   
##   
## $depression  
## [1] "numeric"  
##   
## $depressed  
## [1] "numeric"  
##   
## $cflisen0  
## [1] "numeric"  
##   
## $QoLscore0  
## [1] "numeric"

## Preparing and merging the Data

Each of the 7 ELSA datasets contains 1 row for each participant who was interviewed at that particular Wave: the datasets therefore are in ‘wide’ form. To keep things simple I anticipated that we were going to reshape the data from wide to long: therefore the original variables were renamed to have the wave as the suffix: e.g. cflisen0; QoLscore0. First, drop participants who did not have a valid memory score (cflisen0) at baseline using the filter function.

ELSA\_wave1\_new <- ELSA\_wave1 %>% filter(!is.na(cflisen0))

Recode sex from “Female” and “Male” to 1 and 0 respectively.

ELSA\_wave1\_new$dhsex <- ifelse(ELSA\_wave1\_new$dhsex == 1, 0, 1)  
ELSA\_wave1\_new$dhsex <- as.integer(ELSA\_wave1\_new$dhsex)

At this stage we should have a ‘wide’ dataset with N=11,035 rows.

### Check the length of the new dataframe ###  
length(ELSA\_wave1\_new$idauniq)

## [1] 11035

Keep just the variables we need in new dataframes using select.

ELSA\_wave1\_new <- ELSA\_wave1\_new %>%   
 dplyr::select(idauniq,cflisen0,QoLscore0,dhsex)   
ELSA\_wave2\_new <- ELSA\_wave2 %>%   
 dplyr::select(idauniq,cflisen1,QoLscore1)   
ELSA\_wave3\_new <- ELSA\_wave3 %>%   
 dplyr::select(idauniq,cflisen2,QoLscore2)   
ELSA\_wave4\_new <- ELSA\_wave4 %>%   
 dplyr::select(idauniq,cflisen3,QoLscore3)  
ELSA\_wave5\_new <- ELSA\_wave5 %>%   
 dplyr::select(idauniq,cflisen4,QoLscore4)  
ELSA\_wave6\_new <- ELSA\_wave6 %>%   
 dplyr::select(idauniq,cflisen5,QoLscore5)  
ELSA\_wave7\_new <- ELSA\_wave7 %>%   
 dplyr::select(idauniq,cflisen6,QoLscore6)

## Create a wide (merged) dataframe

To have the data ready to reshape we must merge the seven datasets (using the personal identifier **idauniq**) so that we have all the available information for each participant on the same row. In R this is achieved by the leftjoin command as follows:

ELSA\_wide <- dplyr::left\_join(ELSA\_wave1\_new, ELSA\_wave2\_new,   
 by='idauniq') %>%  
 dplyr::left\_join(.,ELSA\_wave3\_new , by='idauniq') %>%  
 dplyr::left\_join(.,ELSA\_wave4\_new , by='idauniq') %>%  
 dplyr::left\_join(.,ELSA\_wave5\_new , by='idauniq') %>%  
 dplyr::left\_join(.,ELSA\_wave6\_new , by='idauniq') %>%  
 dplyr::left\_join(.,ELSA\_wave7\_new , by='idauniq')  
#View(ELSA\_wide)

# Part 2 - Descriptive Statistics and Reshaping the Data

Obtain summary statistics for the wave-specific values of **QoLscore**, and inspect the correlation among the scores, using the following commands:

psych::describe(ELSA\_wide$QoLscore0)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 9231 42.51 8.69 44 43.31 8.9 7 57 50 -0.85 0.52 0.09

psych::describe(ELSA\_wide$QoLscore1)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 6753 42.74 8.69 44 43.52 8.9 6 57 51 -0.81 0.42 0.11

psych::describe(ELSA\_wide$QoLscore2)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 5987 41.1 8.58 42 41.71 8.9 4 57 53 -0.67 0.23 0.11

psych::describe(ELSA\_wide$QoLscore3)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 5258 40.97 8.61 42 41.58 8.9 5 57 52 -0.68 0.29 0.12

psych::describe(ELSA\_wide$QoLscore4)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 5114 40.9 8.83 42 41.52 8.9 8 57 49 -0.64 0.15 0.12

psych::describe(ELSA\_wide$QoLscore5)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 4410 40.83 8.6 42 41.39 8.9 3 57 54 -0.61 0.16 0.13

psych::describe(ELSA\_wide$QoLscore6)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 3749 41.74 8.58 43 42.32 8.9 7 57 50 -0.62 0.1 0.14

cor <- ELSA\_wide %>%  
 select(QoLscore0,QoLscore1,QoLscore2,  
 QoLscore3,QoLscore4,QoLscore5,QoLscore6) %>%   
 as.matrix() %>%  
 Hmisc::rcorr(type = "pearson")  
cor[1]

## $r  
## QoLscore0 QoLscore1 QoLscore2 QoLscore3 QoLscore4 QoLscore5 QoLscore6  
## QoLscore0 1.0000000 0.7413105 0.6850436 0.6391521 0.6208704 0.5777250 0.5659726  
## QoLscore1 0.7413105 1.0000000 0.7531615 0.7068383 0.6825553 0.6359879 0.6032860  
## QoLscore2 0.6850436 0.7531615 1.0000000 0.7845977 0.7507372 0.7038995 0.6834387  
## QoLscore3 0.6391521 0.7068383 0.7845977 1.0000000 0.7959708 0.7385970 0.7157178  
## QoLscore4 0.6208704 0.6825553 0.7507372 0.7959708 1.0000000 0.7877537 0.7552994  
## QoLscore5 0.5777250 0.6359879 0.7038995 0.7385970 0.7877537 1.0000000 0.7890160  
## QoLscore6 0.5659726 0.6032860 0.6834387 0.7157178 0.7552994 0.7890160 1.0000000

cor[2]

## $n  
## QoLscore0 QoLscore1 QoLscore2 QoLscore3 QoLscore4 QoLscore5 QoLscore6  
## QoLscore0 9231 6155 5439 4792 4655 4033 3453  
## QoLscore1 6155 6753 5017 4397 4259 3716 3213  
## QoLscore2 5439 5017 5987 4469 4235 3693 3194  
## QoLscore3 4792 4397 4469 5258 4278 3718 3220  
## QoLscore4 4655 4259 4235 4278 5114 3958 3407  
## QoLscore5 4033 3716 3693 3718 3958 4410 3314  
## QoLscore6 3453 3213 3194 3220 3407 3314 3749

**Q1**

How many participants had plausible values of QoLscore at each wave? Would you expect the scores to be correlated across the waves? What pattern if any do you see in the correlations?

Fitting random effects / mixed models in R requires the dataset to be in long form. First, order the variables before reshaping using **relocate**: the order of the variables is then idauniq (column 1), followed in columns 2 to 8 by the CF variables (waves 1 to 7); and then in columns 9 to 15 by the QoL variables (waves 1 to 7).

ELSA\_wide2 <- ELSA\_wide %>%   
 relocate(starts\_with(c("cflisen","QoLscore")),.after = idauniq)  
#View(ELSA\_wide2)

## Reshape the data

Here we reshape the data from wide to long using **reshape**.

ELSA\_long <- stats::reshape(as.data.frame(ELSA\_wide2),   
idvar = "idauniq",   
direction="long",  
varying=list(c(2:8),c(9:15)),  
timevar = "time",  
v.names=c("cflisen","QoLscore"))  
#number of records in this long dataset.  
ELSA\_long %>% summarise(n=n())

## n  
## 1 77245

#View(ELSA\_long)

Observations with QoLscore==NA or cflisen==NA will be automatically dropped from any regression model. To run our analyses on the same number of observations drop these observations from the dataset.

#Drop any missing values  
ELSA\_long <- ELSA\_long %>% filter(!is.na(QoLscore))  
ELSA\_long <- ELSA\_long %>% filter(!is.na(cflisen))

This leaves us with a dataset of N=40,437 rows.

ELSA\_long %>% summarise(n=n())

## n  
## 1 40437

Sort the data by idauniq and time using **arrange**.

ELSA\_long <- ELSA\_long %>%  
 dplyr::arrange(ELSA\_long, idauniq,time)  
head(ELSA\_long)

## idauniq dhsex time cflisen QoLscore  
## 103712.1 103712 1 1 6 34  
## 103712.2 103712 1 2 6 42  
## 103712.3 103712 1 3 10 37  
## 103712.4 103712 1 4 7 42  
## 103712.5 103712 1 5 7 40  
## 103712.6 103712 1 6 6 36

Summarise memory scores:

describe(ELSA\_long$cflisen)

## ELSA\_long$cflisen Format:%9.0g   
## n missing distinct Info Mean Gmd .05 .10   
## 40437 0 11 0.965 5.762 1.892 3 4   
## .25 .50 .75 .90 .95   
## 5 6 7 8 8   
##   
## lowest : 0 1 2 3 4, highest: 6 7 8 9 10  
##   
## Value 0 1 2 3 4 5 6 7 8 9 10  
## Frequency 257 777 636 1885 4647 8246 10053 8165 4182 1343 246  
## Proportion 0.006 0.019 0.016 0.047 0.115 0.204 0.249 0.202 0.103 0.033 0.006

Examine distribution of QoL scores:

describe(ELSA\_long$QoLscore)

## ELSA\_long$QoLscore Format:%9.0g   
## n missing distinct Info Mean Gmd .05 .10   
## 40437 0 55 0.999 41.69 9.692 25 30   
## .25 .50 .75 .90 .95   
## 36 43 48 52 54   
##   
## lowest : 3 4 5 6 7, highest: 53 54 55 56 57

Then we create a variable which indicates how many waves each participant had valid values of the exposure and outcome variables. I group the long dataset by the participant identifier and create a new variable (**nwaves**) which is equal to the number of records per id.

#number of waves (as new variable).  
ELSA\_long <- ELSA\_long %>%  
 group\_by(idauniq) %>%  
 mutate(nwaves = n())  
#View(ELSA\_long)

With this new variable, I create an idauniq-level dataset (called IDdata) using same principle as tag in Stata (using **filter** and **row\_number**), and obtain a frequency of this variable across the N=10,173 participants.

IDdata <- ELSA\_long %>%  
 group\_by(idauniq) %>%  
 arrange(time) %>%  
 filter(row\_number()==1)   
IDdata$nwaves<-as.factor(IDdata$nwaves)  
summarytools::freq(IDdata$nwaves)

## Frequencies   
## IDdata$nwaves   
## Type: Factor   
##   
## Freq % Valid % Valid Cum. % Total % Total Cum.  
## ----------- ------- --------- -------------- --------- --------------  
## 1 2195 21.58 21.58 21.58 21.58  
## 2 1420 13.96 35.54 13.96 35.54  
## 3 1080 10.62 46.15 10.62 46.15  
## 4 925 9.09 55.24 9.09 55.24  
## 5 1024 10.07 65.31 10.07 65.31  
## 6 1361 13.38 78.69 13.38 78.69  
## 7 2168 21.31 100.00 21.31 100.00  
## <NA> 0 0.00 100.00  
## Total 10173 100.00 100.00 100.00 100.00

**Q2**

Is this a balanced dataset, or is it unbalanced? Balanced = all individuals have the same number of repeated measurements, obtained on a common set of occasions. Unbalanced = the sequence of observation times is not common to all individuals.

**Q3**

How many persons in this dataset had at least one observed value of QoLscore over the follow-up period?

# Part 3 - Random Effects Model - Varying Intercepts

Fit the null model (i.e. just the constant as a predictor) for a random effects / mixed model that allows individuals to have randomly varying intercepts.

**Q4**

Before doing so, how many parameters would you expect to be estimated? Having fitted the null model, estimate the intra-cluster correlation (ICC). The commands are as follows:

Fit null model (“randint”) with random effects:

randint <- lmer(QoLscore ~ 1 + (1|idauniq),   
## Variable of interest, 1 denotes null, idauniq in the grouping variable  
data = ELSA\_long,   
## Dataframe  
REML = FALSE)   
## Should the estimates be chosen to optimise the restricted maximum likelihood (REML) criterion;   
##FALSE means it optimises log-liklihood instead.

Look at model table:

summary(randint)

## Linear mixed model fit by maximum likelihood ['lmerMod']  
## Formula: QoLscore ~ 1 + (1 | idauniq)  
## Data: ELSA\_long  
##   
## AIC BIC logLik deviance df.resid   
## 264345.3 264371.1 -132169.7 264339.3 40434   
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -6.5733 -0.4958 0.0544 0.5340 4.3855   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## idauniq (Intercept) 55.70 7.463   
## Residual 23.36 4.833   
## Number of obs: 40437, groups: idauniq, 10173  
##   
## Fixed effects:  
## Estimate Std. Error t value  
## (Intercept) 41.11007 0.07971 515.8

Add intercept and residual values (taken from “random effects” section in randint summary table above) into ICC equation

ICC <- (55.70)/(55.70 + 23.36)  
ICC

## [1] 0.7045282

**Q5**

What is the estimate for the population average intercept (0 in the lecture slides): and what is your interpretation of this value?

**Q6**

How do you interpret the ICC?

# Adding fixed effects to the null random effects model

Using a random effects model (to accommodate the correlation among repeated measurements for the same person due to unobserved heterogeneity) we can incorporate the effects of predictors into the model for the mean response by including them as fixed effects. Add the predictor variables **cflisen** (as a continuous variable) and sex (**dhsex**) (as a dummy / indicator variable) as fixed effects to the null model above that allowed individuals to have randomly varying intercepts.

**Q7**

Before doing so, how many parameters would you expect to see in the output?

Having fitted the model, store these current (active) estimation results under the name M1. The commands to fit model (M1) with memory (cflisen) and sex (dhsex) as fixed effects are as follows:

M1 <- lmer(QoLscore ~ 1 + (1|idauniq) + cflisen + dhsex,   
## Added memory and sex as fixed effects  
data = ELSA\_long,   
## Dataframe  
REML = FALSE)   
## Same as above  
#Look at model table  
summary(M1)

## Linear mixed model fit by maximum likelihood ['lmerMod']  
## Formula: QoLscore ~ 1 + (1 | idauniq) + cflisen + dhsex  
## Data: ELSA\_long  
##   
## AIC BIC logLik deviance df.resid   
## 263960.2 264003.3 -131975.1 263950.2 40432   
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -6.6249 -0.4924 0.0514 0.5363 4.3527   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## idauniq (Intercept) 53.73 7.330   
## Residual 23.30 4.827   
## Number of obs: 40437, groups: idauniq, 10173  
##   
## Fixed effects:  
## Estimate Std. Error t value  
## (Intercept) 38.84058 0.15845 245.122  
## cflisen 0.39817 0.02012 19.792  
## dhsex 0.13526 0.15755 0.859  
##   
## Correlation of Fixed Effects:  
## (Intr) cflisn  
## cflisen -0.682   
## dhsex -0.514 -0.036

**Q8**

How do you interpret the slope for cflisen?

**Q9**

How do you interpret the effect of sex (note: males are reference)?

**Q10**

To check your answer to Q9, save and graph the predicted values from the model using the following commands after estimation of the model:

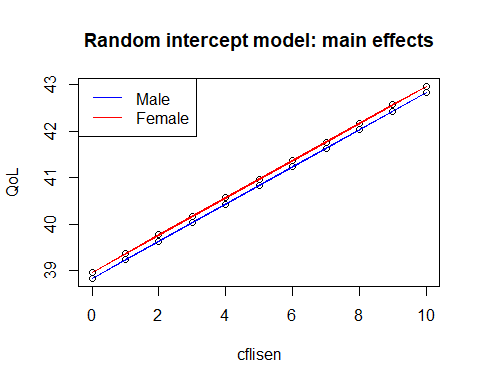
#Plot the pop.average trajectory  
#Creates new column with predicted values  
ELSA\_long$yhat <- predict(M1, re.form = NA)   
  
# Look at these values   
summary(ELSA\_long$yhat)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 38.84 40.83 41.23 41.21 41.76 42.96

#Create reduced dataframe  
fitted.data <- subset(ELSA\_long,   
 select = c("idauniq", "dhsex", "cflisen", "yhat"))  
head(fitted.data)

## # A tibble: 6 x 4  
## # Groups: idauniq [1]  
## idauniq dhsex cflisen yhat  
## <dbl+lbl> <int> <dbl> <dbl>  
## 1 103712 1 6 41.4  
## 2 103712 1 6 41.4  
## 3 103712 1 10 43.0  
## 4 103712 1 7 41.8  
## 5 103712 1 7 41.8  
## 6 103712 1 6 41.4

#Plot fitted data  
with(fitted.data,  
{plot(cflisen, yhat,  
main = "Random intercept model: main effects",  
xlab="cflisen",ylab="QoL");  
lines(cflisen[dhsex==0], yhat[dhsex==0],type = "l", col="blue")  
lines(cflisen[dhsex==1], yhat[dhsex==1],type = "l", col="red")  
legend("topleft", c("Male","Female"), lty = c(1,1), col = c("blue","red"))})



# Part 4 - Random Effects Model - Varying Intercepts and Slopes

We can relax the assumption of the previous random effect model by allowing for individual-specific random variability in the slope of the memory - quality of life association. Given your previous specification of the fixed part of the model (main effects for memory score and sex), now add a random slope to the model, and also allow estimation of the correlation between the randomly varying intercepts and slopes.

**Q10**

Before doing so, how many parameters would you expect to see in the output? Having fitted the model store these current (active) estimation results under the name M2.

**Q11**

Then perform a Likelihood Ratio (LR) test to examine the statistical significance of adding these two parameters to the random part of the model. What are your conclusions? Should we proceed with a more complex model that allows individual-specific random variability in the slope of the memory - quality of life association: or should we assume no such variability in the slope?

**Q12**

What is the estimated correlation between the random intercepts and slopes?

**Q13**

Save the estimates of the level-2 random effects (intercepts and slopes). Obtain histograms of the random effects. What distribution is assumed for these residuals? What are your conclusions about whether this assumption is reasonable in this case?

The commands that follow show you how to: fit the model; store the estimates; save the predicted values of the random intercepts and slopes; obtain histograms; and run a LR test comparing the models with (M2) and without random slopes (M1).

M2 <- lmer(QoLscore ~ 1 + (1 + cflisen|idauniq) + cflisen + dhsex,   
#Added random variabilibity in the slope for memory ("cflisen")  
data = ELSA\_long,   
REML = FALSE)   
#Note: Ignore warning messages:  
  
# Look at model table  
summary(M2)

## Linear mixed model fit by maximum likelihood ['lmerMod']  
## Formula: QoLscore ~ 1 + (1 + cflisen | idauniq) + cflisen + dhsex  
## Data: ELSA\_long  
##   
## AIC BIC logLik deviance df.resid   
## 263876.6 263936.8 -131931.3 263862.6 40430   
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -6.6566 -0.4901 0.0540 0.5313 4.3859   
##   
## Random effects:  
## Groups Name Variance Std.Dev. Corr   
## idauniq (Intercept) 73.3391 8.5638   
## cflisen 0.2737 0.5232 -0.57  
## Residual 22.8623 4.7815   
## Number of obs: 40437, groups: idauniq, 10173  
##   
## Fixed effects:  
## Estimate Std. Error t value  
## (Intercept) 38.78619 0.16745 231.63  
## cflisen 0.41627 0.02145 19.40  
## dhsex 0.12870 0.15695 0.82  
##   
## Correlation of Fixed Effects:  
## (Intr) cflisn  
## cflisen -0.723   
## dhsex -0.482 -0.038  
## optimizer (nloptwrap) convergence code: 0 (OK)  
## Model failed to converge with max|grad| = 0.00287436 (tol = 0.002, component 1)

# Run a Likelihood ratio test to compare the models   
lrtest(M2, M1) ## Larger model goes first

## Likelihood ratio test  
##   
## Model 1: QoLscore ~ 1 + (1 + cflisen | idauniq) + cflisen + dhsex  
## Model 2: QoLscore ~ 1 + (1 | idauniq) + cflisen + dhsex  
## #Df LogLik Df Chisq Pr(>Chisq)   
## 1 7 -131931   
## 2 5 -131975 -2 87.62 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#Extract additional variables (predicted values and random effects) ###  
ELSA\_long$person.specific <- fitted(M2)   
#Fitted values: person-specific ##  
ELSA\_long$pop.average <- predict(M2, re.form = NA)   
head(ELSA\_long)

## # A tibble: 6 x 9  
## # Groups: idauniq [1]  
## idauniq dhsex time cflisen QoLscore nwaves yhat person.specific pop.average  
## <dbl+lb> <int> <int> <dbl> <dbl> <int> <dbl> <dbl> <dbl>  
## 1 103712 1 1 6 34 7 41.4 37.7 41.4  
## 2 103712 1 2 6 42 7 41.4 37.7 41.4  
## 3 103712 1 3 10 37 7 43.0 39.3 43.1  
## 4 103712 1 4 7 42 7 41.8 38.1 41.8  
## 5 103712 1 5 7 40 7 41.8 38.1 41.8  
## 6 103712 1 6 6 36 7 41.4 37.7 41.4

# Create new dataframe containing random effects ###  
random.effects <- as.data.frame(ranef(M2)$idauniq)   
names(random.effects)

## [1] "(Intercept)" "cflisen"

# Add idauniq column  
random.effects <- cbind(idauniq =   
 rownames(random.effects), random.effects)  
names(random.effects)

## [1] "idauniq" "(Intercept)" "cflisen"

head(random.effects)

## idauniq (Intercept) cflisen  
## 103712 103712 -3.589866 -0.01879125  
## 103713 103713 -10.876169 0.31586708  
## 103714 103714 -9.497864 0.10586792  
## 103715 103715 9.161786 -0.14714829  
## 103716 103716 3.126899 -0.11665361  
## 103717 103717 -8.406910 0.34711170

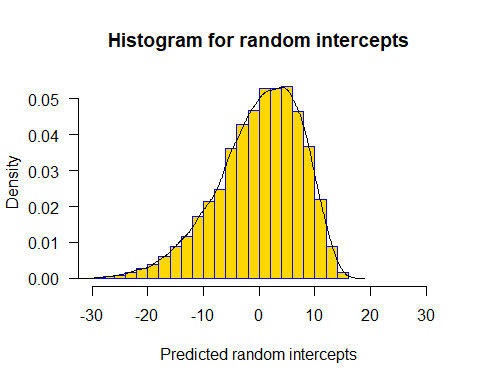
#Summarise the intercept ("(Intercept)") and slope ("cflisen") columns  
summary(random.effects$"(Intercept)")

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -31.6946 -4.6716 0.8969 0.0000 5.6278 15.7103

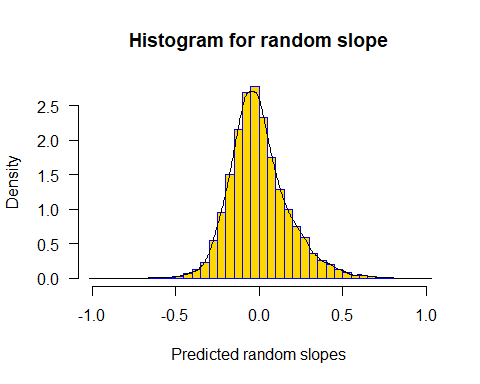
summary(random.effects$"cflisen")

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -0.95101 -0.11406 -0.01960 0.00000 0.09192 0.96568

#Plot a histogram of the random effects: Intercepts and slopes ###  
#Plot density of the intercept ###  
hist(random.effects$"(Intercept)",   
 main="Histogram for random intercepts",   
 xlab="Predicted random intercepts",   
 border="blue",   
 col="gold",   
 xlim=c(-30,30),   
 las=1,   
 breaks=30,   
 prob = TRUE)  
lines(density(random.effects$"(Intercept)"))



### Plot density of the slope ###  
hist(random.effects$"cflisen",   
 main="Histogram for random slope",   
 xlab="Predicted random slopes",   
 border="blue",   
 col="gold",   
 xlim=c(-1,1),   
 las=1,   
 breaks=30,   
 prob = TRUE)  
lines(density(random.effects$"cflisen"))



# Optional: Plotting the person-specific trajectories

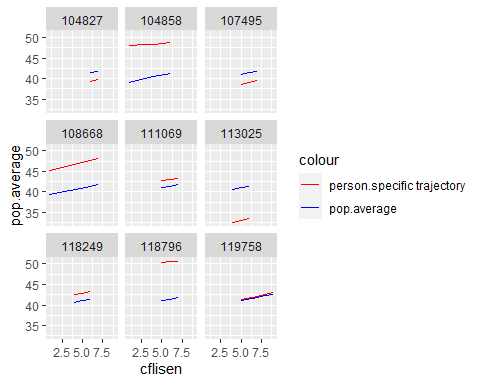
In the final part of this practical I thought I would demonstrate how to plot the person-specific trajectories and compare those to the population-average trajectory.

To keep things manageable, we focus on just nine participants.

### Create a new dataframe with a subset of nine participants ###  
ELSA\_9IDs <-subset(ELSA\_long,idauniq==104827|idauniq==104858|idauniq==107495|idauniq==108668|idauniq==111069|idauniq==113025|idauniq==118249|idauniq==118796|idauniq==119758)  
head(ELSA\_9IDs)

## # A tibble: 6 x 9  
## # Groups: idauniq [1]  
## idauniq dhsex time cflisen QoLscore nwaves yhat person.specific pop.average  
## <dbl+lb> <int> <int> <dbl> <dbl> <int> <dbl> <dbl> <dbl>  
## 1 104827 1 1 7 40 6 41.8 39.8 41.8  
## 2 104827 1 2 7 39 6 41.8 39.8 41.8  
## 3 104827 1 3 6 42 6 41.4 39.3 41.4  
## 4 104827 1 4 6 39 6 41.4 39.3 41.4  
## 5 104827 1 5 6 34 6 41.4 39.3 41.4  
## 6 104827 1 6 6 42 6 41.4 39.3 41.4

### Plot trajectories for these nine participants ###  
### using the predicted values we obtained earlier ###  
myplot <- ggplot(data=ELSA\_9IDs,aes(cflisen)) +   
 geom\_line(aes(y=pop.average,colour="pop.average")) +   
 geom\_line(aes(y=person.specific,colour="person.specific trajectory")) +   
 facet\_wrap(~idauniq) +  
 scale\_colour\_manual(values=c("red","blue"))  
myplot



#list the residuals for intercept and slope.  
mydata <- subset(random.effects,idauniq==104827|idauniq==104858|idauniq==107495|idauniq==108668|idauniq==111069|idauniq==113025|idauniq==118249|idauniq==118796|idauniq==119758)  
mydata

## idauniq (Intercept) cflisen  
## 104827 104827 -2.2582505 0.03049695  
## 104858 104858 9.0695417 -0.27380090  
## 107495 107495 -2.7211354 0.05481085  
## 108668 108668 5.9677583 0.03255818  
## 111069 111069 2.1893995 -0.09870130  
## 113025 113025 -8.3543212 0.07399377  
## 118249 118249 2.3465208 -0.08865404  
## 118796 118796 10.2827659 -0.19868729  
## 119758 119758 0.4243164 -0.01104857

**Q14**

Which of the 9 participants had the lowest predicted random intercept?

**Q15**

Which of the 9 participants had the highest predicted random slope?