Introduction to mixed models in R and Rstudio

Arturo Marroquín

2024-03-20

First framing statement:

"Mixed models" is a term often used as a surrogate of any kind of analysis that takes into account correlation between measurements. However, in this document we will make reference specifically to the method that is an extension of simple and multiple regression. As this label might be confusing and even misleading, researchers should always confirm what type of the analysis is performed by softwares proclaiming to run "mixed models".

Summary

First: Quick scan on R proficiency

Libraries:

```r
# Install packages only if you have not already

#install.packages("tidyverse",
#                 "truncnorm",
#                 "gtsummary",
#                 "psych",
#                 "ggplot2",
#                 "ggpubr",
#                 "dplyr",
#                 "broom",
#                 "tidyr",
#                 "lme4",
#                 "sjPlot",
#                 "lmertest",
#                 "performance"
#                 )

# Loading libraries

library(tidyverse)# simulated data
library(truncnorm)# N distribution simulated (rtruncnorm)
library(gtsummary)# Summary tables (tbl_summary)
library(psych)# for matrix of scatter plots
library(ggplot2)# additional figures
library(ggpubr)# for combined plots
library(dplyr)# for %>%
library(broom)# for augment
library(tidyr)# reformatting data set (pivot_longer)
library(lme4)# mixed models
library(sjPlot)# tab_model, nice output
library(lmerTest)# P-values in mixed models
library(performance)
```

Steps that we will skip but that should be always taken into account:

1) This study answers a pertinent question that follows a theoretical rationale.
2) The design of the study should be able to answer the question and the statistical analysis should have been discussed with a statistician or the like while writing the protocol.
3) The **sampling** was made in accordance to the objectives and the generalizability intended.

Let us start!

Case study construction

Background

Omayra Sánchez, 1982

Nevado del Ruiz

Armero, 2022

A research group is very interested in evaluating the **impact** of experiencing a catastrophic event on the severity of symptoms and the changes of such severity throughout time in adults diagnosed with depression.

The classification of the severity was done with the PHQ-9.

Several measurements were made for each individual in different time points.

Armero has 200 000 inhabitants and 12 000 individuals have depression (6% prevalence).

This is a simulated scenario, so let us create a population with certain characteristics (a.k.a, variables).

```
# We establish a seed to obtain everytime the same results
set.seed(14)

# We create a dataset of the population of armero
population <- data.frame(phq9_baseline=round(rtruncnorm(n=12000, a=1, b=27, mean=13,
                                                        sd=6),0),
                 condition=sample(c("Non-exposed","Exposed"),12000, replace = T,
                                  prob = c(0.5,0.5)),
                 sex=sample(c("Female","Male"),12000, replace = T,
                                  prob = c(0.5,0.5)),
                 age_baseline=round(rtruncnorm(n=12000, a=12, b=90, mean=50, sd=25),0),
                 neighbourhood=sample(c("Paloquemao","Vanier","The heights","Montmartre"),
                                      12000, replace = T, prob =c(rep(0.25,4))),
                 education=sample(c("School","No education"), 12000, replace = T,
                                  prob = c(0.5,0.5)),
                 phq9_M1=round(rtruncnorm(n=12000, a=0, b=27, mean=12, sd=6),0),
                 phq9_M2=round(rtruncnorm(n=12000, a=0, b=27, mean=11, sd=6),0),
                 phq9_M3=round(rtruncnorm(n=12000, a=0, b=27, mean=10, sd=6),0))

# We create some tendencies in the data (essentially, females, exposed and
# two neighborhoods increase PHQ-9)
population[population$condition=="Non-exposed" &
               population$phq9_baseline>4,]$phq9_baseline <-
  population[population$condition=="Non-exposed" &
                 population$phq9_baseline>4,]$phq9_baseline-4
```

```
population[population$phq9_M1=="Non-exposed" & population$phq9_M1>4,]$phq9_M1 <-
  population[population$phq9_M1=="Non-exposed" & population$phq9_M1>4,]$phq9_M1-4

population[population$phq9_M2=="Non-exposed" & population$phq9_M2>4,]$phq9_M2 <-
  population[population$phq9_M2=="Non-exposed" & population$phq9_M2>4,]$phq9_M2-4

population[population$phq9_M3=="Non-exposed" & population$phq9_M3>4,]$phq9_M3 <-
  population[population$phq9_M3=="Non-exposed" & population$phq9_M3>4,]$phq9_M3-4

population[population$condition=="Exposed" &
               population$phq9_baseline<27,]$phq9_baseline <-
  population[population$condition=="Exposed" &
                 population$phq9_baseline<27,]$phq9_baseline+0

population[population$condition=="Exposed" &
               population$phq9_M1<27,]$phq9_M1 <-
  population[population$condition=="Exposed" &
                 population$phq9_M1<27,]$phq9_M1+1

population[population$condition=="Exposed" &
               population$phq9_M2<27,]$phq9_M2 <-
  population[population$condition=="Exposed" &
                 population$phq9_M2<27,]$phq9_M2+0

population[population$condition=="Exposed" &
               population$phq9_M3<27,]$phq9_M3 <-
  population[population$condition=="Exposed" &
                 population$phq9_M3<27,]$phq9_M3+0

population[population$sex=="Female",]$phq9_baseline <-
  population[population$sex=="Female",]$phq9_baseline+
  round(rtruncnorm(n=nrow(population[population$sex=="Female",]),
             a=3, b=7, mean=5,sd=1))

population[population$sex=="Female",]$phq9_M1 <-
  population[population$sex=="Female",]$phq9_M1+
  round(rtruncnorm(n=nrow(population[population$sex=="Female",]),
             a=3, b=7, mean=5,sd=1))

population[population$sex=="Female",]$phq9_M2 <-
  population[population$sex=="Female",]$phq9_M2+
  round(rtruncnorm(n=nrow(population[population$sex=="Female",]),
             a=2, b=6, mean=4,sd=1))

population[population$sex=="Female",]$phq9_M3 <-
  population[population$sex=="Female",]$phq9_M3+
  round(rtruncnorm(n=nrow(population[population$sex=="Female",]),
             a=2, b=6, mean=4,sd=1))

population[population$neighbourhood=="Monmartre"|
             population$neighbourhood=="The heights",]$phq9_baseline <-
  population[population$neighbourhood=="Monmartre"|
             population$neighbourhood=="The heights",]$phq9_baseline+
```

```
  round(rtruncnorm(n=nrow(population[population$neighbourhood=="Monmartre"|
              population$neighbourhood=="The heights",]),
              a=3, b=7, mean=5,sd=1))

population[population$neighbourhood=="Monmartre"|
              population$neighbourhood=="The heights",]$phq9_M1 <-
  population[population$neighbourhood=="Monmartre"|
              population$neighbourhood=="The heights",]$phq9_M1+
  round(rtruncnorm(n=nrow(population[population$neighbourhood=="Monmartre"|
              population$neighbourhood=="The heights",]),
              a=3, b=7, mean=5,sd=1))

population[population$neighbourhood=="Monmartre"|
              population$neighbourhood=="The heights",]$phq9_M2 <-
  population[population$neighbourhood=="Monmartre"|
              population$neighbourhood=="The heights",]$phq9_M2+
  round(rtruncnorm(n=nrow(population[population$neighbourhood=="Monmartre"|
              population$neighbourhood=="The heights",]),
              a=2, b=6, mean=5,sd=1))

population[population$neighbourhood=="Monmartre"|
              population$neighbourhood=="The heights",]$phq9_M3 <-
  population[population$neighbourhood=="Monmartre"|
              population$neighbourhood=="The heights",]$phq9_M3+
  round(rtruncnorm(n=nrow(population[population$neighbourhood=="Monmartre"|
              population$neighbourhood=="The heights",]),
              a=2, b=6, mean=5,sd=1))

population$phq9_baseline[population$phq9_baseline>27] <- 27
population$phq9_M1[population$phq9_M1>27] <- 27
population$phq9_M2[population$phq9_M2>27] <- 27
population$phq9_M3[population$phq9_M3>27] <- 27
```

After this chunk of code we count with our population. This would represent the "truth", seldom available to the researcher. In this case, we can corroborate our estimates (more details below).

What researchers usually do is sample a part of the population. If this is done properly (a.ka., in a *probabilistic* and *pertinent* manner), the sample is a good representative of the population and the **statistics** are good **estimators** of the population **parameters**.

We are going to skip the sample size calculation, but... well, you know. Also, eligibility criteria, but also a think to consider. In the following chunk of code we sample in a stratified manner: 100 individuals that were exposed to the natural disaster and 100 individuals that were not.

```
# Sampling procedure, every individual has the same probability of being sampled.
sample_armero <-
  rbind(population[population$condition=="Exposed",]
        [sample(nrow(population[population$condition=="Exposed",]), 100),],
                      population[population$condition=="Non-exposed",]
        [sample(nrow(population[population$condition=="Non-exposed",]), 100),])
```

Sample exploration

First we want to check how the exposure relates to the **outcome** of interest.

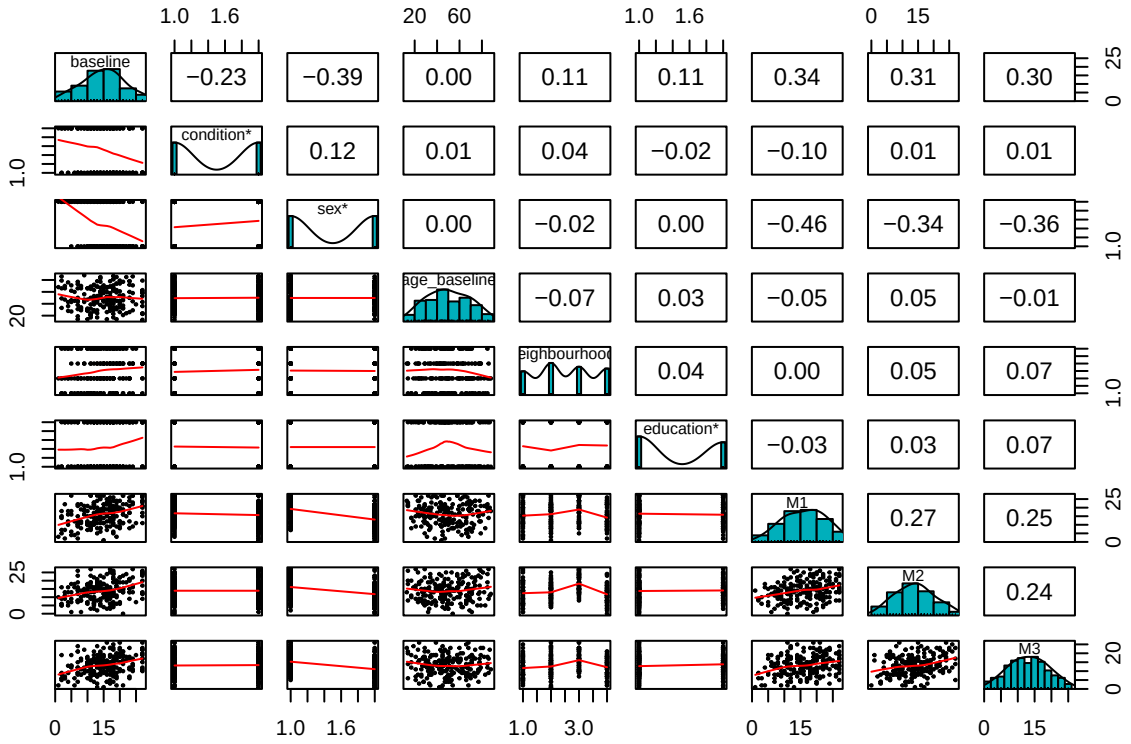
```r
# The categorical variables are described as relative frequencies
# (proportions) and continuous with the median and the IQR.
tbl_summary(sample_armero, by=c("condition"))
```

Characteristic	**Exposed**, N = 100	**Non-exposed**, N = 100
phq9_baseline	17 (12, 20)	14 (10, 18)
sex		
Female	56 (56%)	44 (44%)
Male	44 (44%)	56 (56%)
age_baseline	49 (36, 64)	49 (34, 66)
neighbourhood		
Montmartre	26 (26%)	17 (17%)
Paloquemao	28 (28%)	30 (30%)
The heights	18 (18%)	33 (33%)
Vanier	28 (28%)	20 (20%)
education		
No education	54 (54%)	56 (56%)
School	46 (46%)	44 (44%)
phq9_M1	16 (11, 22)	17 (10, 20)
phq9_M2	14 (9, 18)	14 (10, 18)
phq9_M3	13.0 (9.0, 17.0)	14.0 (9.0, 18.0)

This describes the dependent variable according to exposure, but nothing else. Only how the exposure is distributed in the rest of variables can be observed.

In this line, we would like to see how the continuous variable distributes in those levels. To do so, we will use visual aids.

```r
# change names just for the figure
sample_armero2 <- sample_armero
names(sample_armero2) <- gsub("phq9_","",names(sample_armero2))
# plots matrix
pairs.panels(sample_armero2,
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
             density = TRUE,  # show density plots
             ellipses = FALSE, # do not show ellipses
             cex.cor = 1, # change correlation text size
             cex=0.4, # change point size
             )
```

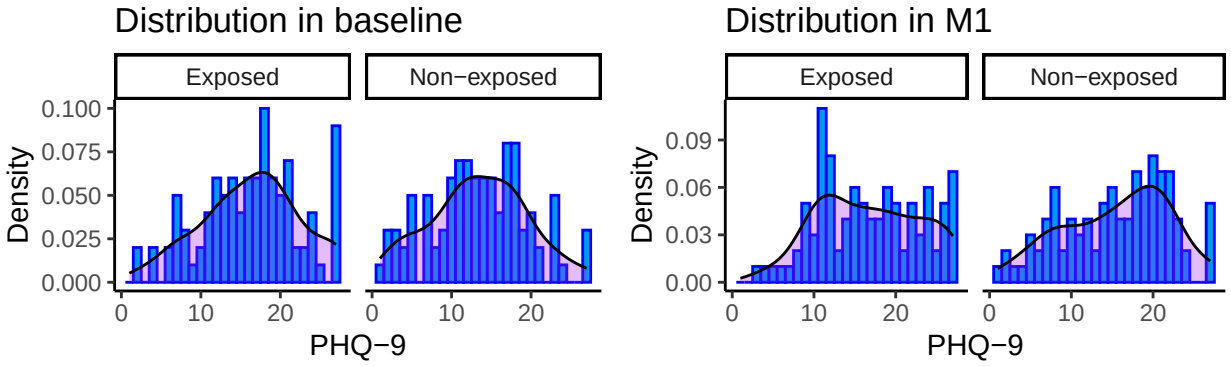


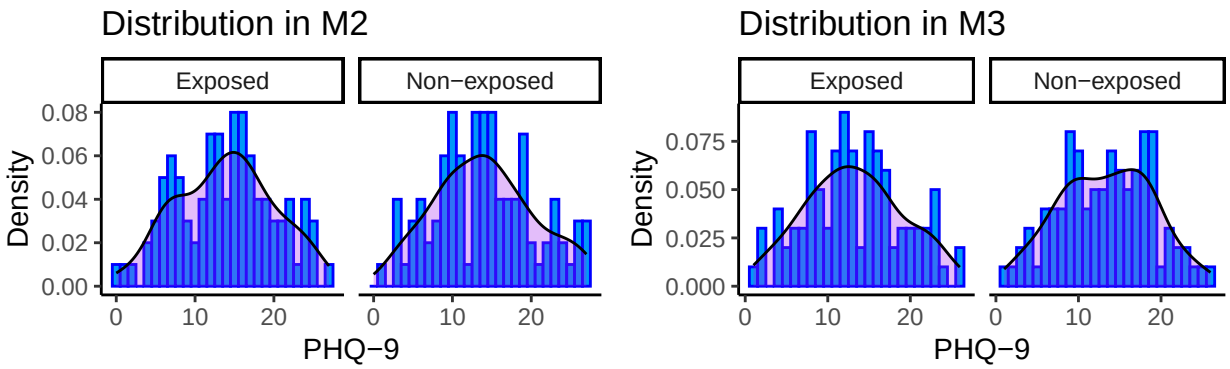

```
rm(sample_armero2)
```

In a more thorough way, we can check the distribution of our variable interest in the different levels of the exposure variable. There is a lot to discuss about the proper test to utilize according to the nature of the variables of interest.

```
# create histograms for baseline and follow-up measurements
p1 <- ggplot(sample_armero, aes(x=phq9_baseline))+
      geom_histogram(aes(y =..density..),col="blue", fill = "#0099F8",binwidth = 1) +
      geom_density(color = "black", fill = "purple", alpha = 0.3) +
      labs(title='Distribution in baseline',
           x='PHQ-9', y='Density')+
      theme_classic()+
     theme(legend.position = c(.95, .95),legend.justification = c("right", "top"),
           legend.box.just = "right",
    legend.margin = margin(6, 6, 6, 6),legend.title = element_text(face = "bold"))+
    guides(fill=guide_legend(title="New Legend Title"))+
  facet_wrap(~condition)

p2 <- ggplot(sample_armero, aes(x=phq9_M1))+
      geom_histogram(aes(y =..density..),col="blue", fill = "#0099F8",binwidth = 1) +
      geom_density(color = "black", fill = "purple", alpha = 0.3) +
      labs(title='Distribution in M1',
           x='PHQ-9', y='Density')+
```

```
      theme_classic()+
     theme(legend.position = c(.95, .95),legend.justification = c("right", "top"),
           legend.box.just = "right",
    legend.margin = margin(6, 6, 6, 6),legend.title = element_text(face = "bold"))+
    guides(fill=guide_legend(title="New Legend Title"))+
  facet_wrap(~condition)

p3 <- ggplot(sample_armero, aes(x=phq9_M2))+
      geom_histogram(aes(y =..density..),col="blue", fill = "#0099F8",binwidth = 1) +
      geom_density(color = "black", fill = "purple", alpha = 0.3) +
      labs(title='Distribution in M2',
           x='PHQ-9', y='Density')+
      theme_classic()+
     theme(legend.position = c(.95, .95),legend.justification = c("right", "top"),
           legend.box.just = "right",
    legend.margin = margin(6, 6, 6, 6),legend.title = element_text(face = "bold"))+
    guides(fill=guide_legend(title="New Legend Title"))+
  facet_wrap(~condition)

p4 <- ggplot(sample_armero, aes(x=phq9_M3))+
      geom_histogram(aes(y =..density..),col="blue", fill = "#0099F8",binwidth = 1) +
      geom_density(color = "black", fill = "purple", alpha = 0.3) +
      labs(title='Distribution in M3',
           x='PHQ-9', y='Density')+
      theme_classic()+
     theme(legend.position = c(.95, .95),legend.justification = c("right", "top"),
           legend.box.just = "right",
    legend.margin = margin(6, 6, 6, 6),legend.title = element_text(face = "bold"))+
    guides(fill=guide_legend(title="New Legend Title"))+
  facet_wrap(~condition)

ggarrange(p1,p2,p3,p4, ncol = 2)
```





Checking normality through statistical tests (Shapiro-Wilk test):

```
lapply(sample_armero[sapply(sample_armero, is.numeric)],shapiro.test)
```

```
## $phq9_baseline
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.98191, p-value = 0.01119
##
##
## $age_baseline
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.97828, p-value = 0.003412
##
##
## $phq9_M1
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.97611, p-value = 0.001728
##
##
## $phq9_M2
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.98617, p-value = 0.04785
##
##
## $phq9_M3
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.98754, p-value = 0.07679
```

Finally, we add an identifier to each individual (participant). This will be fundamental for the mixed models analysis. The importance will be evident below.

```
sample_armero$id <- c(1:nrow(sample_armero))
```

Linear regression: review and ground rules

At this point, we can start answering some of the questions that the research group had:

What is the effect of exposure on the severity of depressive symptoms?

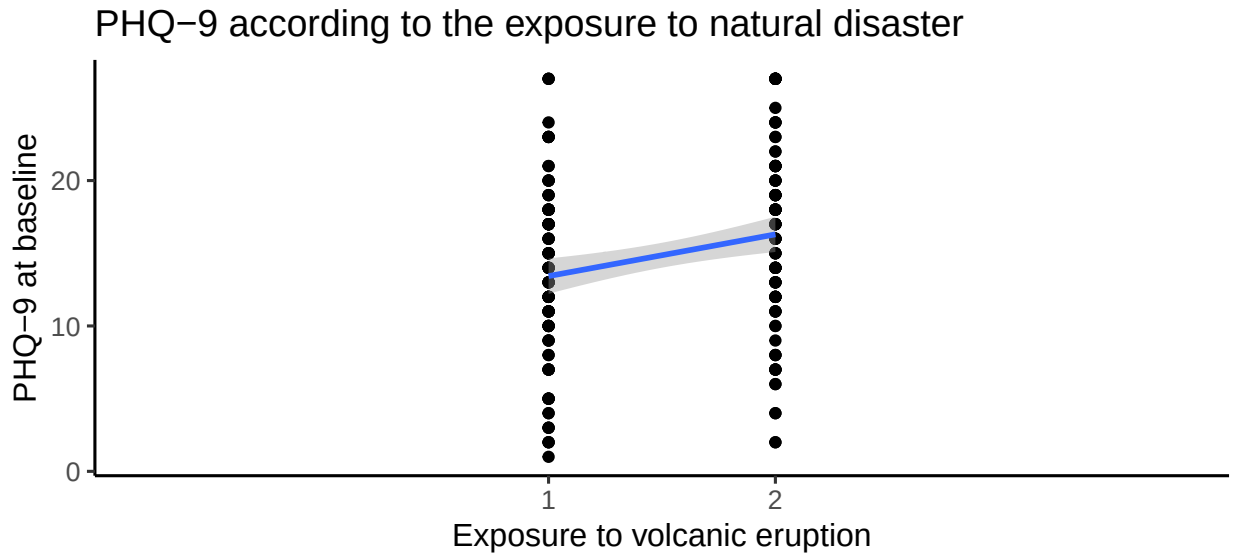
To evaluate this behavior we could simply run a t-test, which can be considered equivalent to the simple linear regression or a one-way ANOVA if we were to add a covariate with more than three levels. Nonetheless, as the data becomes more complex, using regression analysis has several advantages over the other methods.

In this case, as we will follow the way to eventually reach mixed models, we will stick with regression analysis since the beginning.

Theory

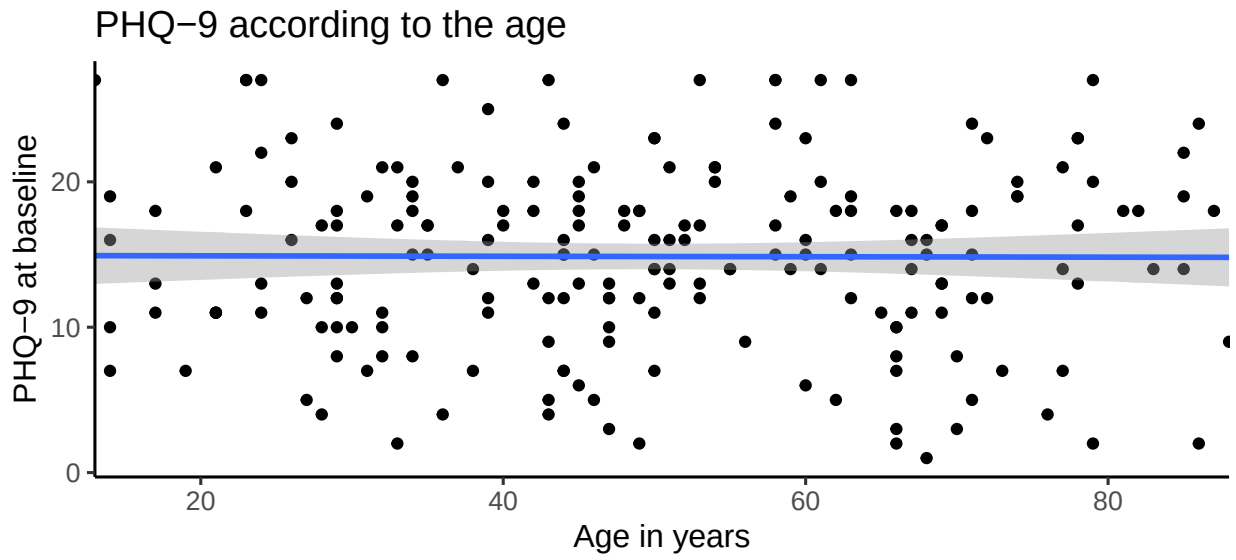
We want to describe the relationship between two variables. To do so, generally, you can have a good idea of what model could fit (represent) the data looking at a scatter plot. These plots will however be more useful for two numeric variables. When we explore the relationship between a categorical independent variable and a numeric continuous dependent variable we will not see a cloud of points but rather two columns like in the case below.

```
ggplot(sample_armero, aes(x=as.numeric(factor(sample_armero$condition,
                                              levels = c("Non-exposed","Exposed")))
                          , y=phq9_baseline))+
  geom_point()+
  geom_smooth(method=lm)+
  labs(title = "PHQ-9 according to the exposure to natural disaster",
       y = "PHQ-9 at baseline", x = "Exposure to volcanic eruption")+
  scale_x_continuous(expand = c(1,1),breaks = c(1,2))+
  theme_classic2()
```



As can be seen, despite the categorical (dichotomous) nature of the relationship, a line can "in average" summarize adequately the distribution between the two levels. A second case, would be the description of the relationship between two numeric variables, for instance PHQ-9 at baseline and age at baseline.

```
ggplot(sample_armero, aes(x=age_baseline, y=phq9_baseline))+
  geom_point()+
  geom_smooth(method=lm)+
  labs(title = "PHQ-9 according to the age",
       y = "PHQ-9 at baseline", x = "Age in years")+
  scale_x_continuous(expand = c(0,0),breaks = c(0,20,40,60,80,100))+
  theme_classic2()
```



We can appreciate there seems to be no relationship between the age and the PHQ-9 of the individuals. Indeed, we are testing visually if a line could be a good model of this relationship. If we were to consider such model, then there are a couple of assumptions that we, well, must assume. It is like flat earthers: if a model (a representation) of earth is flat they need to assume a LOT of things for it to be true. For instance, that something else gradually hides distancing bodies from bottom to top.

Well, linear models have an advantage over this kind of claims. In mathematics, lines are pretty simply described. A simple equation can give you all the information you need to draw such line.

Mathematical equation of a line:

$$y = mx + b$$

m is the slope and b the intecept (the value of y when $x = 0$)

Now, in statistical jargon, they decided that it would be better to write it in a more sophisticated way:

Statistical transform of the mathematical equation of a line:

$$y = \beta_0 + \beta_1 x$$

Why? Because in statistics we deal with inference (conclusions about populations from samples), and so the parts of the equation are no longer only mathematical descriptors dwelling in the mathematical world. They try to represent something in the real world, philosophical stuff... that is fundamental! This is at the very

basis of inferential statistics and for these to be able to represent reality a lot of assumptions have to be, well, assumed. But do not worry, many tools and strategies exist that allow us to test if the assumptions seem to be met by our models. For now, its important to understand that these letters, these specific letters and symbols, are representations of real phenomena in a population.

Going back to linear regression. We have a line that can be described by an equation and that seems to describe adequately a relationship between two variables, but not perfectly! In mathematics a line might be enough to described such relationship, one variable x **determines** y, but this is the real world! PHQ-9, a surrogate of depressive symptomatology is not only **determined** by a another variable. Well, we do not know exactly these other aspects, but the data exhibits it, so we have to include it in our equation.

Population regression line:

$$Y = \beta_0 + \beta_1 X_1 + E$$

The E term represents everything that describes the behavior of y and that is not explained by the exposure x. This part is often labeled as the **random component** of the model, whereas $\beta_0 + \beta_1$ is labeled as the **deterministic component**.

If everything is clear there is only one last thing to discuss before going into the line-drawing part: the above equation corresponds to phenomena in the real world, but usually we do not have access to this precious information. Instead, we use samples to try to aproximate this equation and generate conclusions at the population level. That is why the equation that we actually work on is the following:

Sample regression line:

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\epsilon}$$

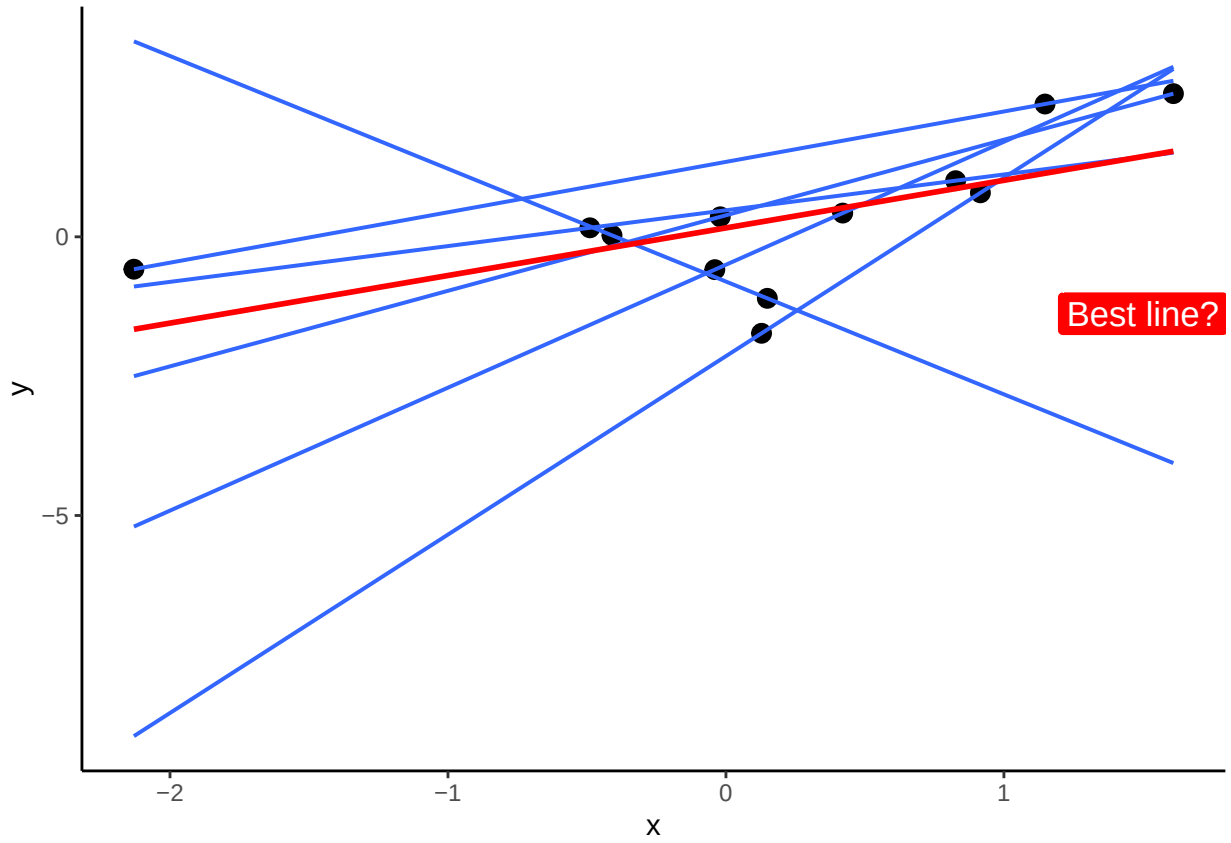
Other way to describe the sample regression line:

$$\hat{Y} = b_0 + b_1 X_1 + e$$

These are the actual values that we see in our data (sample), but they are so much more. They are **estimates** of the true **parameters** of the population and whatever formula used to get their value is an **estimator**. This means that the values will change from sample to sample, building something called a **sampling distribution**. A creature that surely deserves its own pdf file. Know for now that is this property that allows us to make **inferences (a.k.a, hypothesis testing and confidence intervals estimation)**.

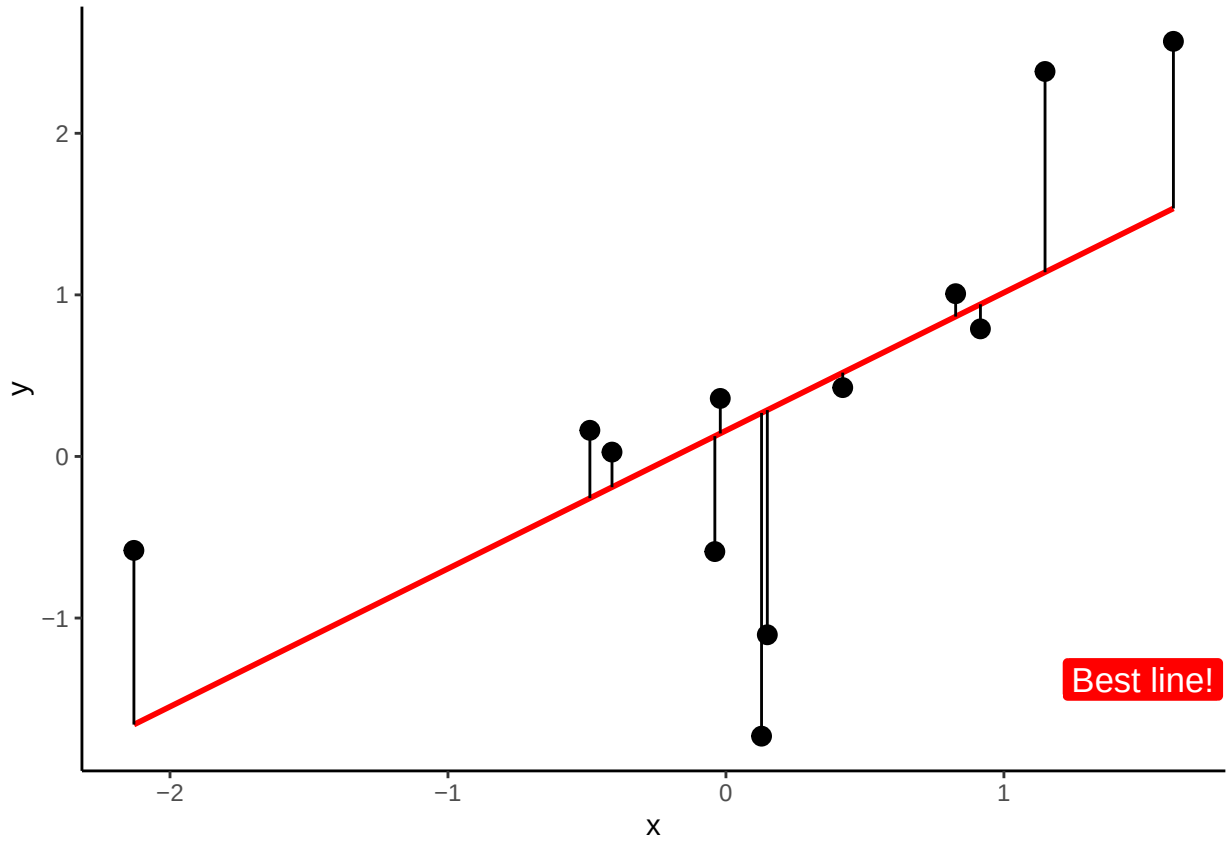
So, apparently a line can give us a lot of information, but so many lines from which to choose. How do we select the line that better describes a particular relationship? What is the main characteristic that this line should have if it were to describe the relationship? Well, intuitively it should cross most points of the scatter plot or at least be the closest as possible to every point.

```
set.seed(26)
ggplot(data.frame(x=rnorm(12,0,1),y=rnorm(12,0,1),
                  z=c(rep("A",2),rep("B",2),
                      rep("C",2),rep("D",2),
                      rep("E",2),rep("F",2))),
       aes(x=x, y=y)) +
  geom_point(size=3) +
  stat_smooth(aes(x=x, y=y, group = z),method = "lm", se=FALSE, fullrange = T, size=0.7)+
  geom_smooth(aes(x=x,y=y), method = "lm", se=FALSE, fullrange = T, size=1, color="red")+
  geom_label(aes(x = 1.5, y = -1.38, label = "Best line?"), size=4.4,colour="white",fill = "red")+
  theme_classic()
```



Visually, we can create an approximate line, its relatively easy but inaccurate. Fortunately, stats enable us to choose the best line according to more stringent criteria. Remember the error term that we were talking about a couple of paragraphs above? Let us dig into that. Let us take the graph above and suppose we are trying to see the relationship between two variables x and y, you choose what they represent, both numeric though. Some lines cross one, two, three points and some others are closer to some points. But, which? Good lord, which?

```
set.seed(26)
data.frame(x=rnorm(12,0,1),y=rnorm(12,0,1),
                  z=c(rep("A",2),rep("B",2),
                      rep("C",2),rep("D",2),
                      rep("E",2),rep("F",2))) %>%
lm(y ~ x, data = .) %>%
augment() %>%
ggplot(aes(x=x, y=y)) +
  geom_point(size=3) +
  geom_smooth(aes(x=x,y=y), method = "lm", se=FALSE, fullrange = T, size=1, color="red")+
  geom_label(aes(x = 1.5, y = -1.38, label = "Best line!"), size=4.4,colour="white",
             fill = "red")+
  geom_segment(aes(xend = x, yend = .fitted)) +
  theme_classic()
```



These segments that you can observe represent the distance between the observed data from the sample and their corresponding **fitted** values from the line. This distance is called the error (**residual**), ϵ or e in our formula. Indeed, the best fitting line will have the least error, but how to find this mathematically instead of measuring every segment with a ruler?

We could sum all the error segments, but the sum would be 0 (negatives annul positives). One way out is to sum the squared errors, a value known as **the sum squared error (SSE)**.

$$SSE = \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 = \sum_{i=1}^{n}(Y_i - \hat{\beta}_0 + \hat{\beta}_1 X_{i1})^2$$

Indeed, statisticians, made fancy experiments and trials to solve for the $\hat{\beta}_0$ and $\hat{\beta}_1$ that minimizes the SSE. This method is known as **the least square estimation** method and the line that results from it **the least square regression line**.

Least squares **estimators**:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sum_{i=1}^{n}(X_i - \overline{X})^2}$$

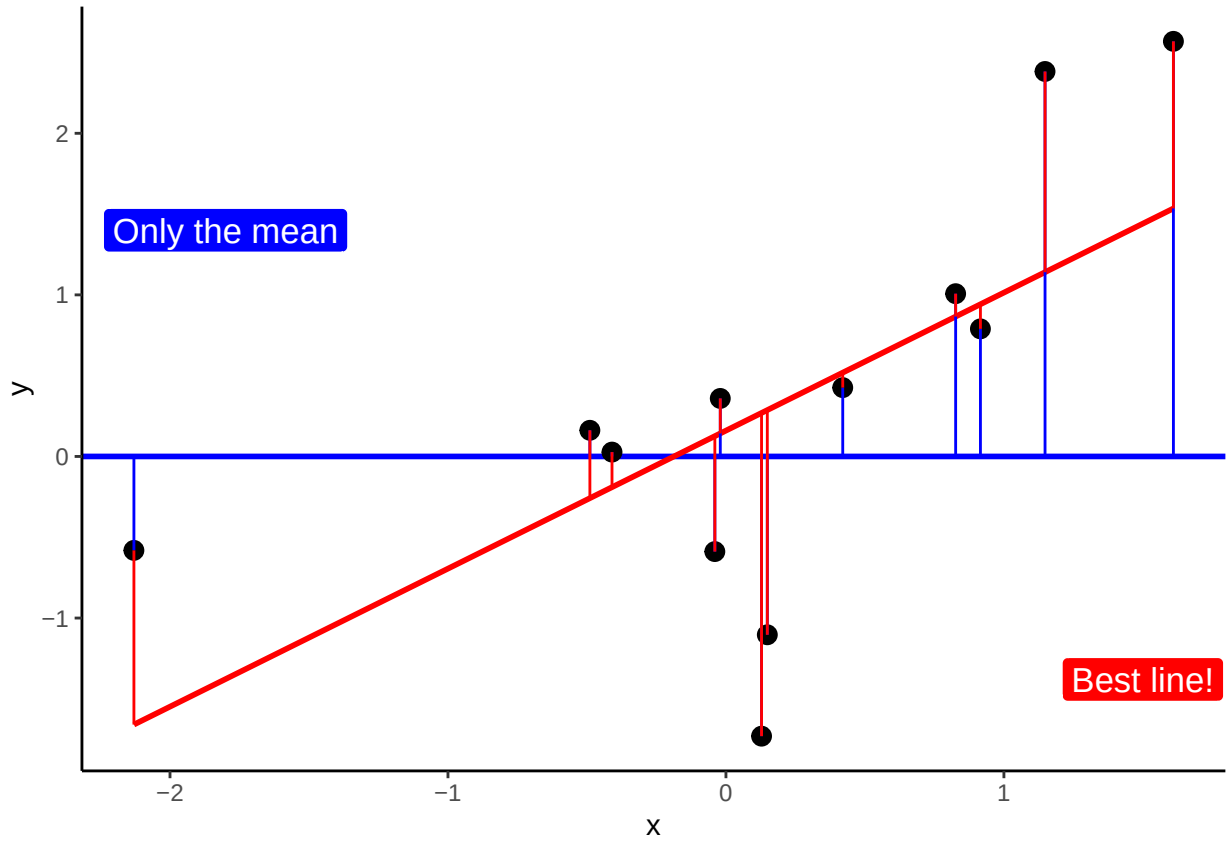
$$\hat{\beta}_0 = \overline{Y} - \hat{\beta}_1 \overline{X}$$

That is it, we have our line. Now, inferences!!

Remember, we said that this values would change from sample to sample creating a fantastic beast. Well, we can estimate some characteristic of the distribution of this parameters and decide if the estimate observed in our data is likely to be seen in such a sample were the corresponding ***parameter*** to be 0 (no linear relationship). This is called a Wald test. This is the first p-value that we will talk about.

For the second one, we need another measure. In general, an univariate description of a variable can give you a lot of information. The mean is already telling you a lot if the variable is symmetric (i.e, gaussian). That is the minimum information that you count only with the variable, so one wants to know, not only if the relationship exists (β_1 is different from 0), but if your model is actually useful. Interestingly, with only one independent variable, these two p-values would be equal. But let us talk about the second.

```
set.seed(26)
data.frame(x=rnorm(12,0,1),y=rnorm(12,0,1),
                  z=c(rep("A",2),rep("B",2),
                      rep("C",2),rep("D",2),
                      rep("E",2),rep("F",2))) %>%
lm(y ~ x, data = .) %>%
augment() %>%
ggplot(aes(x=x, y=y)) +
  geom_hline(yintercept=0,color="blue", size=1)+
  geom_point(size=3) +
  geom_smooth(aes(x=x,y=y), method = "lm", se=FALSE, fullrange = T, size=1, color="red")+
  geom_label(aes(x = 1.5, y = -1.38, label = "Best line!"), size=4.4,colour="white",
             fill = "red")+
  geom_label(aes(x = -1.8, y = 1.4, label = "Only the mean"), size=4.4,colour="white",
             fill = "blue")+
   geom_segment(aes(xend =x , yend = 0), color="blue") +
  geom_segment(aes(xend = x, yend = .fitted), color="red") +
  theme_classic()
```



With the mean we are already explaining some of the variation, but then, how much of the variation in y is better understood thanks to x. This measure will correspond to the segments between the mean of y and the fitted line $\hat{y_i}$.The sum of this squared distance is known as **the sum squared about the regression (SSR)**. And,

$$SSR = \sum_{i=1}^{n}(\hat{y_i} - \bar{y_i})^2 = \sum_{i=1}^{n}(\hat{\beta}_0 + \hat{\beta}_1 X_{i1} - \bar{y_i})^2$$

Great! Now, all the variability in y is its numerator of the variance: **The total sum of squares**. And intuitively:

$$SST = SSE + SSR$$

so,

$$\frac{SST - SSE}{SST} = \frac{SSR}{SST} = R^2$$

This is also called $R^2 \ldots$

Finally, just like in ANOVA we test with an F statistic taking into account the **degrees of freedom**:

$$F = \frac{Mean\ squares\ about\ the\ regression}{Mean\ squares\ about\ the\ error} = \frac{\frac{(SST-SSE)}{k}}{\frac{SSE}{n-k-1}}$$

If F is big enough we can reject the null hypothesis, claiming that the model, a.k.a, the regression line, explains more than 0.

This is it. You know simple linear regression now. But life is not that easy. We have to check certain things before claiming truth or discovery. In this case we have to check if the assumptions hold and the behavior of residuals will light the path. They have to be normal, have a constant variance in the different values of x, be linear and **NOT BE CORRELATED** or better yet, **BE INDEPENDENT** from one another. We already saw them... residual plot.

For another time... Let us run the linear model in R.

```
# the only line of code you need to run it
#summary(lm(phq9_baseline~condition,sample_armero))
# but is good practice to save it in an object
model_simple_lm <- lm(phq9_baseline~condition,sample_armero)
summary(model_simple_lm)
```

```
##
## Call:
## lm(formula = phq9_baseline ~ condition, data = sample_armero)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -14.30  -4.30   0.57   4.57  13.57
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            16.3000     0.6147  26.517  < 2e-16 ***
## conditionNon-exposed  -2.8700     0.8693  -3.301  0.00114 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.147 on 198 degrees of freedom
## Multiple R-squared:  0.05218,    Adjusted R-squared:  0.04739
## F-statistic:  10.9 on 1 and 198 DF,  p-value: 0.001141
```

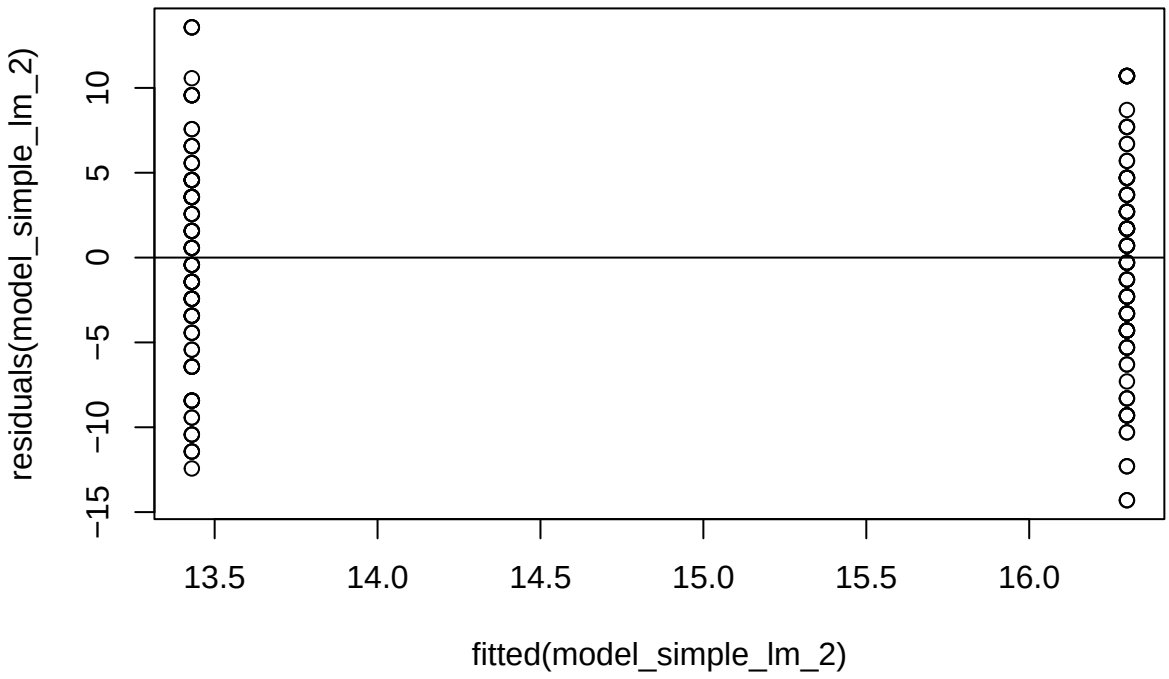
Usually, people like to factor the other way around.

```
model_simple_lm_2 <- lm(phq9_baseline~factor(condition,
                                             levels = c("Non-exposed","Exposed")),
                        sample_armero)
summary(model_simple_lm_2)
```

```
##
## Call:
## lm(formula = phq9_baseline ~ factor(condition, levels = c("Non-exposed",
##     "Exposed")), data = sample_armero)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -14.30  -4.30   0.57   4.57  13.57
##
## Coefficients:
##                                                              Estimate
```

```
## (Intercept)                                                      13.4300
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed   2.8700
##                                                                 Std. Error
## (Intercept)                                                         0.6147
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed      0.8693
##                                                                 t value Pr(>|t|)
## (Intercept)                                                      21.848  < 2e-16
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed   3.301  0.00114
##
## (Intercept)                                                     ***
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.147 on 198 degrees of freedom
## Multiple R-squared:  0.05218,    Adjusted R-squared:  0.04739
## F-statistic:  10.9 on 1 and 198 DF,  p-value: 0.001141
```

First: Interpretation



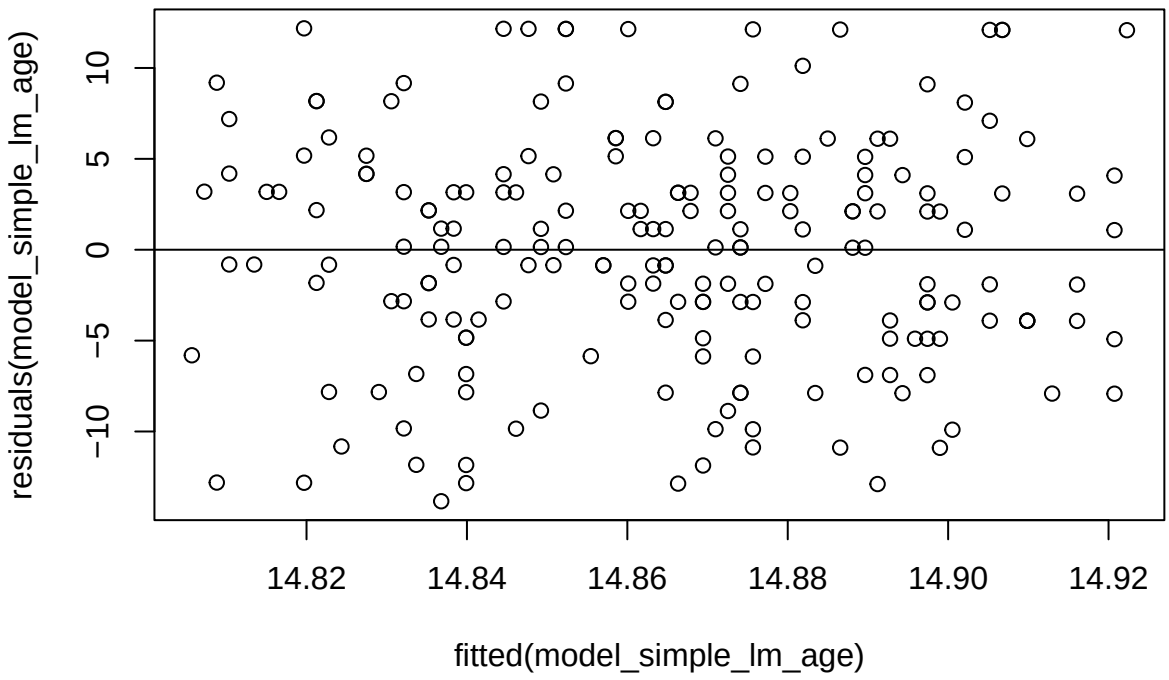
```r
#plot(model_simple_lm_2)
```

Honorable mention: **Transformations**

```r
model_simple_lm_age <- lm(phq9_baseline~age_baseline
                          ,sample_armero)
summary(model_simple_lm_age)
```

```
##
## Call:
## lm(formula = phq9_baseline ~ age_baseline, data = sample_armero)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.8368  -3.9099   0.1492   4.1329  12.1803
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  14.942504   1.270269  11.763   <2e-16 ***
## age_baseline -0.001555   0.023854  -0.065    0.948
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.314 on 198 degrees of freedom
## Multiple R-squared:  2.145e-05,  Adjusted R-squared:  -0.005029
## F-statistic: 0.004247 on 1 and 198 DF,  p-value: 0.9481
```

```r
#explore model
plot(fitted(model_simple_lm_age), residuals(model_simple_lm_age))+
  abline(a = 0,b = 0)
```



Example of multiple regression

```
model_simple_mlm <- lm(phq9_baseline~factor(condition,
                                            levels = c("Non-exposed","Exposed"))+
                         age_baseline+neighbourhood,sample_armero)
summary(model_simple_mlm)
```

```
##
## Call:
## lm(formula = phq9_baseline ~ factor(condition, levels = c("Non-exposed",
##     "Exposed")) + age_baseline + neighbourhood, data = sample_armero)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.6401  -3.8231   0.1057   3.8550  12.2462
##
## Coefficients:
##                                                                  Estimate
## (Intercept)                                                    11.064626
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed  3.804242
## age_baseline                                                   -0.003196
## neighbourhoodPaloquemao                                         1.046132
## neighbourhoodThe heights                                        6.380014
## neighbourhoodVanier                                             0.530448
##                                                                Std. Error
## (Intercept)                                                      1.447319
```

```
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed   0.816612
## age_baseline                                                     0.021631
## neighbourhoodPaloquemao                                          1.148828
## neighbourhoodThe heights                                         1.190506
## neighbourhoodVanier                                              1.190251
##                                                                 t value Pr(>|t|)
## (Intercept)                                                       7.645 9.44e-13
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed   4.659 5.90e-06
## age_baseline                                                     -0.148    0.883
## neighbourhoodPaloquemao                                           0.911    0.364
## neighbourhoodThe heights                                          5.359 2.36e-07
## neighbourhoodVanier                                               0.446    0.656
##
## (Intercept)                                                     ***
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed ***
## age_baseline
## neighbourhoodPaloquemao
## neighbourhoodThe heights                                        ***
## neighbourhoodVanier
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.662 on 194 degrees of freedom
## Multiple R-squared:  0.2121, Adjusted R-squared:  0.1918
## F-statistic: 10.44 on 5 and 194 DF,  p-value: 6.897e-09
```

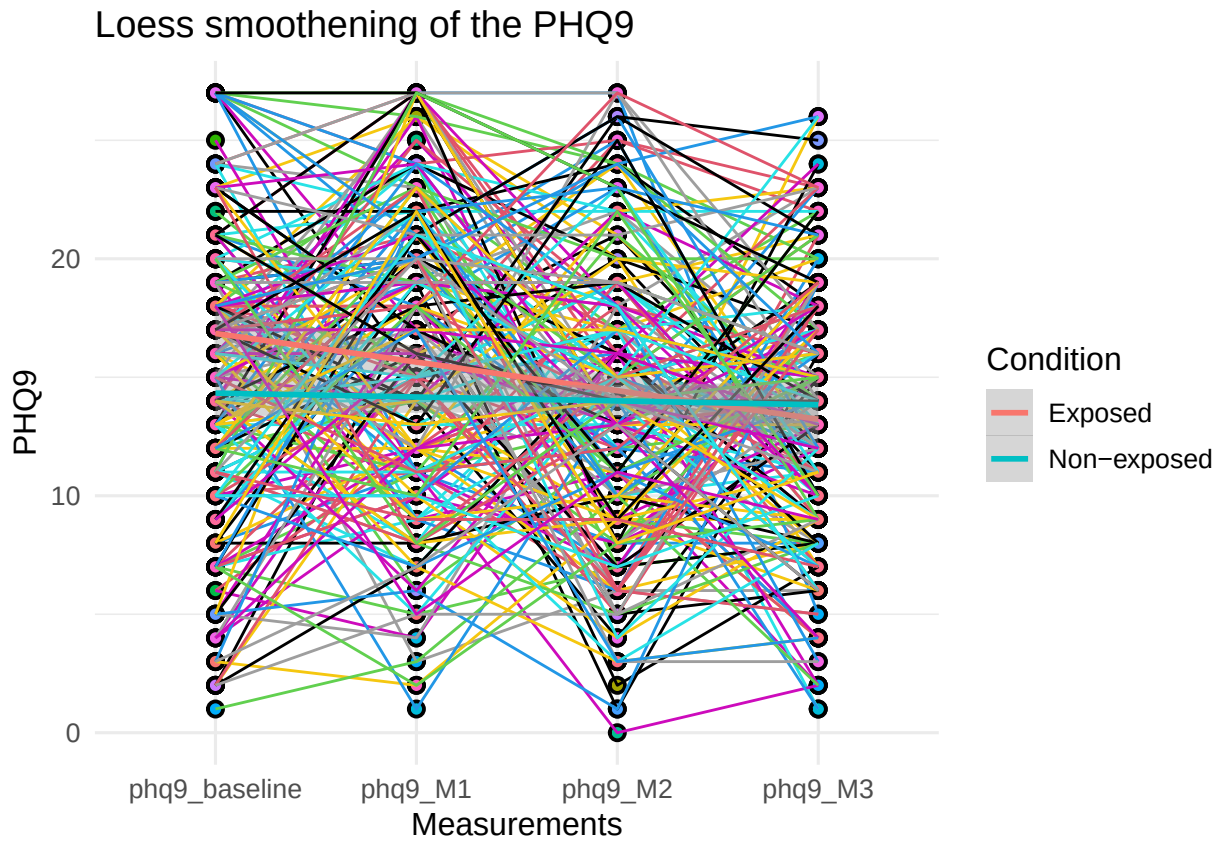
Linear mixed models: an introduction

First let us reshape the dataset:

```
# reshape database
sample_armero_longf <- pivot_longer(sample_armero,
                                    c(phq9_baseline,phq9_M1,
                                      phq9_M2,phq9_M3),names_to = "timepoint",
                                    values_to = "PHQ9")
```

Let us plot:

```
# Base template: time_num for numeric and time_f for factor
ggplot(sample_armero_longf, aes(x = timepoint, y = PHQ9))+
geom_point(aes(fill=factor(id)), pch=21, size=2, stroke=1, show.legend = F)+
geom_line(aes(group=id), col=factor(sample_armero_longf$id))+
  geom_smooth(aes(group=condition,color=condition), se=T, method = lm)+
  theme_minimal(base_size = 12)+#theme_classic() or
  #theme(axis.text.x = element_text(angle = 40, vjust = 1, hjust=1))+
  labs(title="Loess smoothening of the PHQ9",
       color="Condition", y="PHQ9", x= "Measurements")
```




```
multpiple_viol <- lm(PHQ9~factor(condition,
                                          levels = c("Non-exposed","Exposed")),
                     sample_armero_longf)

summary(multpiple_viol)
```

```
##
## Call:
## lm(formula = PHQ9 ~ factor(condition, levels = c("Non-exposed",
##     "Exposed")), data = sample_armero_longf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.0375  -4.0850  -0.0375   4.9150  12.9150
##
## Coefficients:
##                                                                  Estimate
## (Intercept)                                                       14.0850
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed     0.9525
##                                                                  Std. Error
## (Intercept)                                                          0.3114
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed       0.4404
##                                                                  t value Pr(>|t|)
## (Intercept)                                                       45.234   <2e-16
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed     2.163   0.0308
```

```
##
## (Intercept)                                                    ***
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.228 on 798 degrees of freedom
## Multiple R-squared:  0.005829,   Adjusted R-squared:  0.004583
## F-statistic: 4.679 on 1 and 798 DF,  p-value: 0.03084
```

Show error for not adding RE:

```
mix_null <- lmer(PHQ9~
                 # fixed effects
                 1 +
                 # random effects
                 (1|id), sample_armero_longf)

summary(mix_null)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: PHQ9 ~ 1 + (1 | id)
##    Data: sample_armero_longf
##
## REML criterion at convergence: 5131.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.15665 -0.69478 -0.00377  0.68416  2.33447
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept) 10.44    3.232
##  Residual             28.56    5.344
## Number of obs: 800, groups:  id, 200
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  14.5612     0.2965 199.0000   49.11   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
icc(mix_null)
```

```
## # Intraclass Correlation Coefficient
##
##     Adjusted ICC: 0.268
##   Unadjusted ICC: 0.268
```

```
mix1 <- lmer(PHQ9~
                 # fixed effects
```

```
               factor(condition,levels = c("Non-exposed","Exposed")) +
               # random effects
               (1|id), sample_armero_longf)

summary(mix1)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: PHQ9 ~ factor(condition, levels = c("Non-exposed", "Exposed")) +
##     (1 | id)
##    Data: sample_armero_longf
##
## REML criterion at convergence: 5128.3
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.18052 -0.69976  0.01481  0.68074  2.37091
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept) 10.30    3.210
##  Residual             28.56    5.344
## Number of obs: 800, groups:  id, 200
##
## Fixed effects:
##                                                                  Estimate
## (Intercept)                                                       14.0850
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed     0.9525
##                                                                  Std. Error
## (Intercept)                                                         0.4177
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed      0.5907
##                                                                        df t value
## (Intercept)                                                       198.0000  33.724
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed 198.0000   1.613
##                                                                  Pr(>|t|)
## (Intercept)                                                       <2e-16 ***
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed      0.108
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## f(,l=c("N-" -0.707
```

```
library(lmerTest)
```

```
mix1 <- lmer(PHQ9~
               # fixed effects
               factor(condition,levels = c("Non-exposed","Exposed")) +
               # random effects
               (1|id), sample_armero_longf)

summary(mix1)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: PHQ9 ~ factor(condition, levels = c("Non-exposed", "Exposed")) +
##     (1 | id)
##    Data: sample_armero_longf
##
## REML criterion at convergence: 5128.3
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.18052 -0.69976  0.01481  0.68074  2.37091
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept) 10.30    3.210
##  Residual             28.56    5.344
## Number of obs: 800, groups:  id, 200
##
## Fixed effects:
##                                                                  Estimate
## (Intercept)                                                       14.0850
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed     0.9525
##                                                                  Std. Error
## (Intercept)                                                          0.4177
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed       0.5907
##                                                                        df t value
## (Intercept)                                                       198.0000  33.724
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed 198.0000   1.613
##                                                                  Pr(>|t|)
## (Intercept)                                                       <2e-16 ***
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed      0.108
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## f(,l=c("N-" -0.707
```

Let us add time as a fixed effect:

```r
sample_armero_longf$time_num <- as.numeric(factor(sample_armero_longf$timepoint))-1

mix2 <- lmer(PHQ9~
               # fixed effects
               factor(condition,levels = c("Non-exposed","Exposed")) +
               time_num+
               # random effects
               (1|id), sample_armero_longf)

summary(mix2)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
```

```
## Formula: PHQ9 ~ factor(condition, levels = c("Non-exposed", "Exposed")) +
##     time_num + (1 | id)
##    Data: sample_armero_longf
##
## REML criterion at convergence: 5113.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.13614 -0.69467 -0.01393  0.69202  2.33629
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept) 10.49    3.239
##  Residual             27.82    5.274
## Number of obs: 800, groups:  id, 200
##
## Fixed effects:
##                                                                  Estimate
## (Intercept)                                                       15.1133
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed     0.9525
## time_num                                                          -0.6855
##                                                                  Std. Error
## (Intercept)                                                          0.4869
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed       0.5907
## time_num                                                             0.1668
##                                                                        df t value
## (Intercept)                                                       350.6733  31.042
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed 198.0000   1.613
## time_num                                                          599.0000  -4.110
##                                                                  Pr(>|t|)
## (Intercept)                                                       < 2e-16 ***
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed      0.108
## time_num                                                         4.51e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) f(l=c"
## f(,l=c("N-" -0.607
## time_num    -0.514  0.000
```

Let us add the other fixed effects of interest:

```
mix3 <- lmer(PHQ9~
               # fixed effects
               factor(condition,levels = c("Non-exposed","Exposed")) + time_num+
               sex+neighbourhood+education+age_baseline+
               # random effects
               (1|id), sample_armero_longf)
summary(mix3)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
```

```
## Formula: PHQ9 ~ factor(condition, levels = c("Non-exposed", "Exposed")) +
##     time_num + sex + neighbourhood + education + age_baseline +      (1 | id)
##    Data: sample_armero_longf
##
## REML criterion at convergence: 4943.2
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.43112 -0.76233 -0.01591  0.74872  2.58853
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept)  0.4239  0.6511
##  Residual             27.8203  5.2745
## Number of obs: 800, groups:  id, 200
##
## Fixed effects:
##                                                                  Estimate
## (Intercept)                                                     16.206373
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed   1.165481
## time_num                                                        -0.685500
## sexMale                                                         -4.773840
## neighbourhoodPaloquemao                                          0.462052
## neighbourhoodThe heights                                         4.942984
## neighbourhoodVanier                                             -0.557341
## educationSchool                                                  0.376695
## age_baseline                                                    -0.004872
##                                                                 Std. Error
## (Intercept)                                                       0.789768
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed   0.394732
## time_num                                                          0.166794
## sexMale                                                           0.387696
## neighbourhoodPaloquemao                                           0.553215
## neighbourhoodThe heights                                          0.571441
## neighbourhoodVanier                                               0.571519
## educationSchool                                                   0.388208
## age_baseline                                                      0.010387
##                                                                        df
## (Intercept)                                                    236.282375
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed 191.999962
## time_num                                                       598.999951
## sexMale                                                        191.999962
## neighbourhoodPaloquemao                                        191.999962
## neighbourhoodThe heights                                       191.999962
## neighbourhoodVanier                                            191.999962
## educationSchool                                                191.999962
## age_baseline                                                   191.999962
##                                                                t value Pr(>|t|)
## (Intercept)                                                     20.520  < 2e-16
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed   2.953  0.00354
## time_num                                                        -4.110 4.51e-05
## sexMale                                                        -12.313  < 2e-16
## neighbourhoodPaloquemao                                          0.835  0.40464
## neighbourhoodThe heights                                         8.650 2.04e-15
```

```
## neighbourhoodVanier                                                  -0.975  0.33069
## educationSchool                                                       0.970  0.33310
## age_baseline                                                         -0.469  0.63957
##
## (Intercept)                                                                 ***
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed              **
## time_num                                                                    ***
## sexMale                                                                     ***
## neighbourhoodPaloquemao
## neighbourhoodThe heights                                                    ***
## neighbourhoodVanier
## educationSchool
## age_baseline
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) f(l=c" tim_nm sexMal nghbrP nghbTh nghbrV edctnS
## f(,l=c("N-" -0.327
## time_num    -0.317  0.000
## sexMale     -0.298  0.120  0.000
## nghbrhdPlqm -0.389  0.092  0.000  0.059
## nghbrhdThhg -0.420  0.175  0.000  0.025  0.560
## neghbrhdVnr -0.423  0.019  0.000  0.039  0.541  0.528
## educatnSchl -0.195 -0.021  0.000  0.001  0.062 -0.020 -0.012
## age_baselin -0.629 -0.002  0.000 -0.002 -0.096 -0.024  0.046 -0.043
```

Let us compare deviance:

```
mix3_reduced <- lmer(PHQ9~
               # fixed effects
               factor(condition,levels = c("Non-exposed","Exposed")) + time_num+
               sex+neighbourhood+
               # random effects
               (1|id), sample_armero_longf)
summary(mix3_reduced)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: PHQ9 ~ factor(condition, levels = c("Non-exposed", "Exposed")) +
##     time_num + sex + neighbourhood + (1 | id)
##    Data: sample_armero_longf
##
## REML criterion at convergence: 4937
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.48814 -0.78229 -0.02358  0.76114  2.58025
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept)  0.3906  0.625
##  Residual             27.8203  5.274
```

```
## Number of obs: 800, groups:  id, 200
##
## Fixed effects:
##                                                                  Estimate
## (Intercept)                                                       16.1403
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed     1.1730
## time_num                                                         -0.6855
## sexMale                                                          -4.7747
## neighbourhoodPaloquemao                                           0.4064
## neighbourhoodThe heights                                          4.9478
## neighbourhoodVanier                                              -0.5397
##                                                                  Std. Error
## (Intercept)                                                         0.5872
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed      0.3938
## time_num                                                            0.1668
## sexMale                                                             0.3868
## neighbourhoodPaloquemao                                             0.5485
## neighbourhoodThe heights                                            0.5699
## neighbourhoodVanier                                                 0.5696
##                                                                        df t value
## (Intercept)                                                       285.0742  27.488
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed 193.9999   2.979
## time_num                                                          598.9999  -4.110
## sexMale                                                           193.9999 -12.344
## neighbourhoodPaloquemao                                           193.9999   0.741
## neighbourhoodThe heights                                          193.9999   8.682
## neighbourhoodVanier                                               193.9999  -0.948
##                                                                  Pr(>|t|)
## (Intercept)                                                       < 2e-16 ***
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed  0.00326 **
## time_num                                                         4.51e-05 ***
## sexMale                                                           < 2e-16 ***
## neighbourhoodPaloquemao                                           0.45960
## neighbourhoodThe heights                                         1.58e-15 ***
## neighbourhoodVanier                                               0.34454
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) f(l=c" tim_nm sexMal nghbrP nghbTh
## f(,l=c("N-" -0.448
## time_num    -0.426  0.000
## sexMale     -0.401  0.120  0.000
## nghbrhdPlqm -0.589  0.093  0.000  0.059
## nghbrhdThhg -0.590  0.174  0.000  0.025  0.563
## neghbrhdVnr -0.533  0.019  0.000  0.040  0.550  0.530
```

Likelihood ratio test:

```
anova(mix3_reduced, mix3)
```

```
## refitting model(s) with ML (instead of REML)

## Data: sample_armero_longf
```

```
## Models:
## mix3_reduced: PHQ9 ~ factor(condition, levels = c("Non-exposed", "Exposed")) + time_num + sex + neigh
## mix3: PHQ9 ~ factor(condition, levels = c("Non-exposed", "Exposed")) + time_num + sex + neighbourhoo
##              npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## mix3_reduced    9 4952.7 4994.9 -2467.4   4934.7
## mix3           11 4955.6 5007.1 -2466.8   4933.6 1.1678  2     0.5577
```

Let us add random slope of time:

```
mix4 <- lmer(PHQ9~
               # fixed effects
               factor(condition,levels = c("Non-exposed","Exposed")) + time_num+
               sex+neighbourhood+
               # random effects
               (1+time_num|id), sample_armero_longf)
```

```
## boundary (singular) fit: see help('isSingular')
```

```
summary(mix4)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: PHQ9 ~ factor(condition, levels = c("Non-exposed", "Exposed")) +
##     time_num + sex + neighbourhood + (1 + time_num | id)
##    Data: sample_armero_longf
##
## REML criterion at convergence: 4936.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.50505 -0.77248 -0.01557  0.75670  2.54028
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  id       (Intercept)  1.2306  1.1093
##           time_num     0.0686  0.2619   -1.00
##  Residual             27.6144  5.2549
## Number of obs: 800, groups:  id, 200
##
## Fixed effects:
##                                                                  Estimate
## (Intercept)                                                       16.1576
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed     1.1374
## time_num                                                          -0.6855
## sexMale                                                           -4.7649
## neighbourhoodPaloquemao                                            0.4050
## neighbourhoodThe heights                                           4.9349
## neighbourhoodVanier                                               -0.5425
##                                                                  Std. Error
## (Intercept)                                                          0.5920
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed       0.3955
## time_num                                                             0.1672
```

```
## sexMale                                                                0.3885
## neighbourhoodPaloquemao                                                0.5509
## neighbourhoodThe heights                                               0.5724
## neighbourhoodVanier                                                    0.5721
##                                                                          df t value
## (Intercept)                                                          276.8888  27.295
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed       223.5497   2.876
## time_num                                                             489.3933  -4.100
## sexMale                                                              223.5497 -12.264
## neighbourhoodPaloquemao                                              223.5497   0.735
## neighbourhoodThe heights                                             223.5497   8.622
## neighbourhoodVanier                                                  223.5497  -0.948
##                                                                      Pr(>|t|)
## (Intercept)                                                           < 2e-16 ***
## factor(condition, levels = c("Non-exposed", "Exposed"))Exposed        0.00442 **
## time_num                                                             4.84e-05 ***
## sexMale                                                               < 2e-16 ***
## neighbourhoodPaloquemao                                               0.46303
## neighbourhoodThe heights                                             1.22e-15 ***
## neighbourhoodVanier                                                   0.34400
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) f(l=c" tim_nm sexMal nghbrP nghbTh
## f(,l=c("N-" -0.446
## time_num    -0.433  0.000
## sexMale     -0.400  0.120  0.000
## nghbrhdPlqm -0.587  0.093  0.000  0.059
## nghbrhdThhg -0.588  0.174  0.000  0.025  0.563
## neghbrhdVnr -0.531  0.019  0.000  0.040  0.550  0.530
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```
anova(mix3_reduced, mix4)
```

```
## refitting model(s) with ML (instead of REML)

## Data: sample_armero_longf
## Models:
## mix3_reduced: PHQ9 ~ factor(condition, levels = c("Non-exposed", "Exposed")) + time_num + sex + neigh
## mix4: PHQ9 ~ factor(condition, levels = c("Non-exposed", "Exposed")) + time_num + sex + neighbourhoo
##              npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## mix3_reduced    9 4952.7 4994.9 -2467.4   4934.7
## mix4           11 4956.5 5008.1 -2467.3   4934.5 0.2168  2     0.8973
```

Good bye, for now.