Introduction to linear mixed-effects models

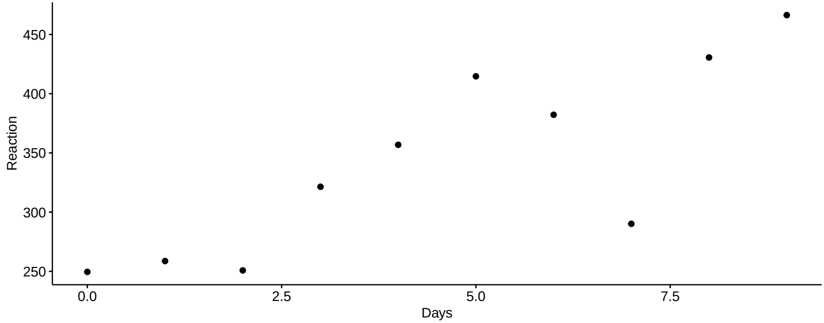
Advanced statistical methods and models in experimental design

Bartosz Maćkiewicz

sleepstudy data

Let's take a closer look at the sleepstudy data. The dataset contains eighteen participants from the three-hour sleep condition. Each day, over 10 days, participants performed a ten-minute "psychomotor vigilance test" where they had to monitor a display and press a button as quickly as possible each time a stimulus appeared. The dependent measure in the dataset is the participant's average response time (RT) on the task for that day.

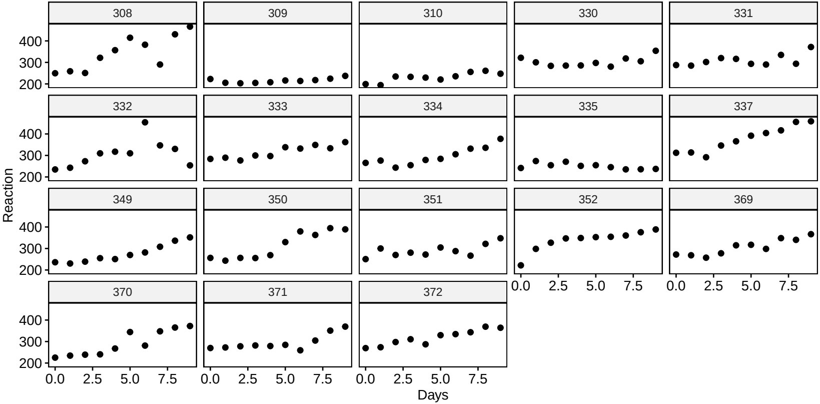
```r
library(tidyverse); library(lme4); library(ggpubr)
participant_308  <- sleepstudy %>% filter(Subject == "308")
ggscatter(participant_308, x =  "Days", y = "Reaction")
```



sleepstudy data

Using ggplot we can create the plot which shows data for all 18 subjects.

```
ggscatter(sleepstudy, x =  "Days", y = "Reaction",
          facet.by = "Subject")
```



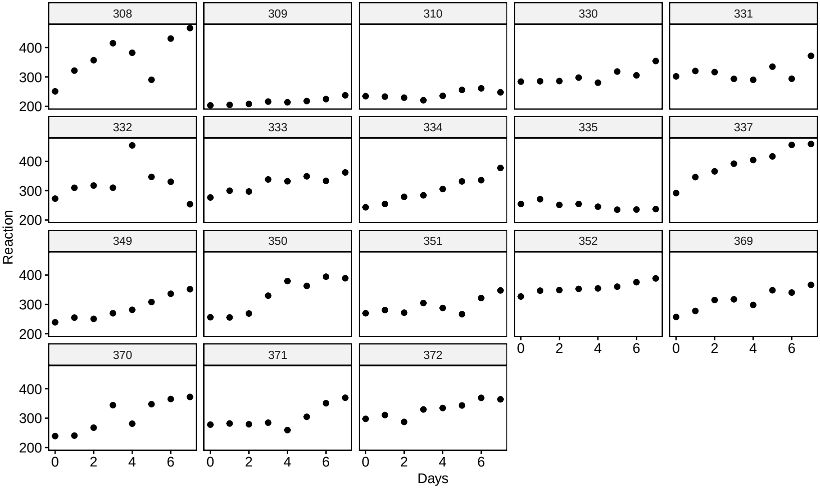
sleepstudy data

This is how Belenky et al. (2003) describe their study (p. 2):

> *The first 3 days (T1, T2 and B) were adaptation and training (T1 and T2) and baseline (B) and subjects were required to be in bed from 23:00 to 07:00 h [8 h required time in bed (TIB)]. On the third day (B), baseline measures were taken. Beginning on the fourth day and continuing for a total of 7 days (E1–E7) subjects were in one of four sleep conditions [9 h required TIB (22:00–07:00 h), 7 h required TIB (24:00–07:00 h), 5 h required TIB (02:00–07:00 h), or 3 h required TIB (04:00–07:00 h)], effectively one sleep augmentation condition, and three sleep restriction conditions.*

sleepstudy data

```
ggscatter(sleepstudy, x =  "Days", y = "Reaction",
          facet.by = "Subject")
```



Complete pooling

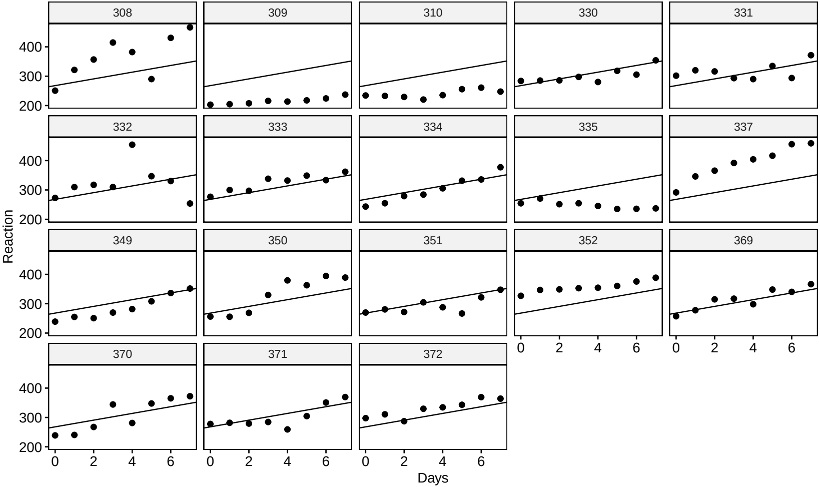
The complete pooling approach is a “one-size-fits-all” model: it estimates a single intercept and slope for the entire dataset, ignoring the fact that different subjects might vary in their intercepts or slopes.

```r
complete_pooling_model <- lm(Reaction ~ Days, data = sleepstudy)
summary(complete_pooling_model)
```

```
## 
## Call:
## lm(formula = Reaction ~ Days, data = sleepstudy)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -112.284  -26.732    2.143   27.734  140.453 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  267.967      7.737  34.633  < 2e-16 ***
## Days          11.435      1.850   6.183 6.32e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 50.85 on 142 degrees of freedom
## Multiple R-squared:  0.2121, Adjusted R-squared:  0.2066 
## F-statistic: 38.23 on 1 and 142 DF,  p-value: 6.316e-09
```

Complete pooling

```
ggscatter(sleepstudy, x =  "Days", y = "Reaction", facet.by = "Subject") +
  geom_abline(
    intercept = complete_pooling_model$coefficients[["(Intercept)"]],
    slope = complete_pooling_model$coefficients[["Days"]]
    )
```



No pooling

Pooling all the information to get just one intercept and one slope estimate seems inappropriate. Another approach would be to fit separate lines for each participant. This means that the estimates for each participant will be completely uninformed by the estimates for the other participants. In other words, we can separately estimate 18 individual intercept/slope pairs.

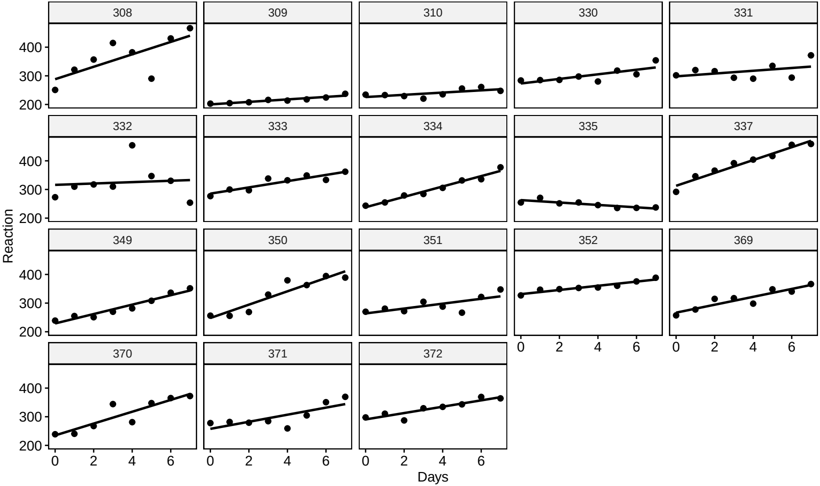
Question: What are intercepts and slopes for subjects 308 and 335?

```
no_pooling_model <- lm(Reaction ~ Days * Subject, data = sleepstudy)
summary(no_pooling_model)
```

```
##
## Call:
## lm(formula = Reaction ~ Days * Subject, data = sleepstudy)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -106.521   -8.541    1.143    8.889  128.545
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      288.2175    16.4772  17.492  < 2e-16 ***
## Days              21.6905     3.9388   5.507 2.49e-07 ***
## Subject309       -87.9262    23.3023  -3.773 0.000264 ***
## Subject310       -62.2856    23.3023  -2.673 0.008685 **
## Subject330       -14.9533    23.3023  -0.642 0.522422
## Subject331         9.9658    23.3023   0.428 0.669740
```

No pooling

```
ggscatter(sleepstudy, x =  "Days", y = "Reaction",
          facet.by = "Subject", add = "reg.line")
```



Partial pooling using mixed-effects models

Neither the complete nor no-pooling approach is satisfactory. It would be desirable to improve our estimates for individual participants by taking advantage of what we know about the other participants. This will help us better distinguish signal from error for each participant and improve generalization to the population.

In the no-pooling model, we treated `Subject` as a **fixed factor**. Each pair of intercept and slope estimates is determined by that subject's data alone. However, we are not interested in these 18 subjects in and of themselves; rather, we are interested in them **as examples drawn from a larger population of potential subjects**. This subjects-as-fixed-effects approach is suboptimal if your goal is to generalize to new participants in the population of interest.

Partial pooling happens when you treat a factor as a **random** instead of **fixed** in your analysis. A **random factor** is a factor whose levels are considered to **represent a proper subset of all the levels in the population**. Usually, you treat a factor as random when the levels you have in the data are the result of sampling, and you want to generalize beyond those levels.

In this case, we have 18 unique subjects and thus, *18* levels of the `Subject` factor, and would like to say something general about effects of sleep deprivation on the population of potential subjects.

Partial pooling using mixed-effects models

A way to include random factors in your analysis is to use a linear mixed-effects model. Rather than estimating the intercept and slope for each participant without considering the estimates for other subjects, the model estimates values for the population, and pulls the estimates for individual subjects toward those values, a statistical phenomenon known as *shrinkage*.

Partial pooling using mixed-effects models: multilevel model

The multilevel model is below. It is important that you understand the math and what it means. It looks complicated at first, but there's really nothing below that you haven't seen before.

Level 1:

$$Y_{sd} = \beta_{0s} + \beta_{1s} X_{sd} + e_{sd}$$

Level 2:

$$\beta_{0s} = \gamma_0 + S_{0s}$$

$$\beta_{1s} = \gamma_1 + S_{1s}$$

$$\langle S_{0s}, S_{1s} \rangle \sim N\left(\langle 0, 0 \rangle, ^{\circ}\right)$$

- Y_{sd} (observed) - value of `Reaction` for subject s on day d
- X_{sd} (observed) - value of `Days` (0-7) for subject s od day d
- β_{0s} (derived) - level 1 intercept parameter for subject s
- β_{1s} (derived) - level 1 intercept parameter for subject s
- e_{sd} (derived) - Error for subject s, day d
- γ_0 (fixed) - Grand intercept ("gamma")
- γ_1 (fixed) - Grand slope ("gamma")
- S_{0s} (derived) - random intercept (offset) for subject s
- S_{1s} (derived) - random slope (offset) for subject s
- Σ (random) - Variance-covariance matrix

Partial pooling using mixed-effects models: fitting the model using lme4

To estimate parameters, we are going to use the lmer() function of the lme4 package. The basic syntax of lmer() is

```
lmer(formula, data, ...)
```

where formula expresses the structure of the underlying model in a compact format.

The general format of the model formula for N fixed effects (fix) and K random effects (ran) is:

```
DV ~ fix1 + fix2 + ... + fixN + (ran1 + ran2 + ... + ranK | random_factor1)
```

Each bracketed expression represents random effects associated with a single random factor. On the left side of the bar | you put the effects you want to allow to vary over the levels of the random factor named on the right side. Usually, the right-side variable is one whose values uniquely identify individual subjects (e.g., subject_id).

Model	Syntax
random intercepts only	`Reaction ~ Days + (1 | Subject)`
random intercepts and slopes	`Reaction ~ Days + (1 + Days | Subject)`
(alternative syntax)	`Reaction ~ Days + (Days | Subject)`
random slopes only	`Reaction ~ Days + (0 + Days | Subject)`
random intercepts and slopes + zero-covariances	`Reaction ~ Days + (Days || Subject)`

Partial pooling using mixed-effects models

```r
pp_model <- lmer(
  Reaction ~ # DV
  Days + # Fixed effect
  (Days |Subject), # random intercept and slope for each subject
  #(1|Subject), # only random intercept for each subject
  data = sleepstudy
  )
```

```r
summary(pp_model)
```

Partial pooling using mixed-effects models: interpreting fixed effects

```
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)    267.967      8.266  32.418
## days_deprived   11.435      1.845   6.197
```

This indicates that the estimated mean reaction time for participants at Day 0 was about 268 milliseconds, with each day of sleep deprivation adding an additional 11 milliseconds to the response time, on average.

Partial pooling using mixed-effects models: random effects

```
## Random effects:
##  Groups   Name          Variance Std.Dev. Corr
##  Subject  (Intercept)   958.35   30.957
##           days_deprived  45.78    6.766   0.18
##  Residual               651.60   25.526
## Number of obs: 144, groups:  Subject, 18
```

What you find here is a table with information about the variance components: the variance-covariance matrix (or matrices, if you have multiple random factors) and the residual variance.

`Residual` tells us that the residual variance was estimated at about 651.6.

The two lines above the `Residual` line give us information about the variance-covariance matrix for the `Subject` random factor. The values in the `Variance` column gives us the main diagonal of the matrix. The `Corr` column tells us the correlation between the intercept and slope.

We can pull out the estimated random effects (BLUPS) using `ranef()`:

```
ranef(pp_model)[["Subject"]] %>% head(4)
```

```
##     (Intercept)      Days
## 308   24.499289  8.602000
## 309  -59.372310 -8.127753
## 310  -39.476276 -7.429237
## 330    1.350043 -2.384598
```

Mixed-Effects Models: why?

Yarkoni (2022): Fixed effects are used to model variables that must remain constant in order for the model to preserve its meaning across replication studies; random effects are used to model indicator variables that are assumed to be stochastically sampled from some underlying population and can vary across replications without meaningfully altering the research question.