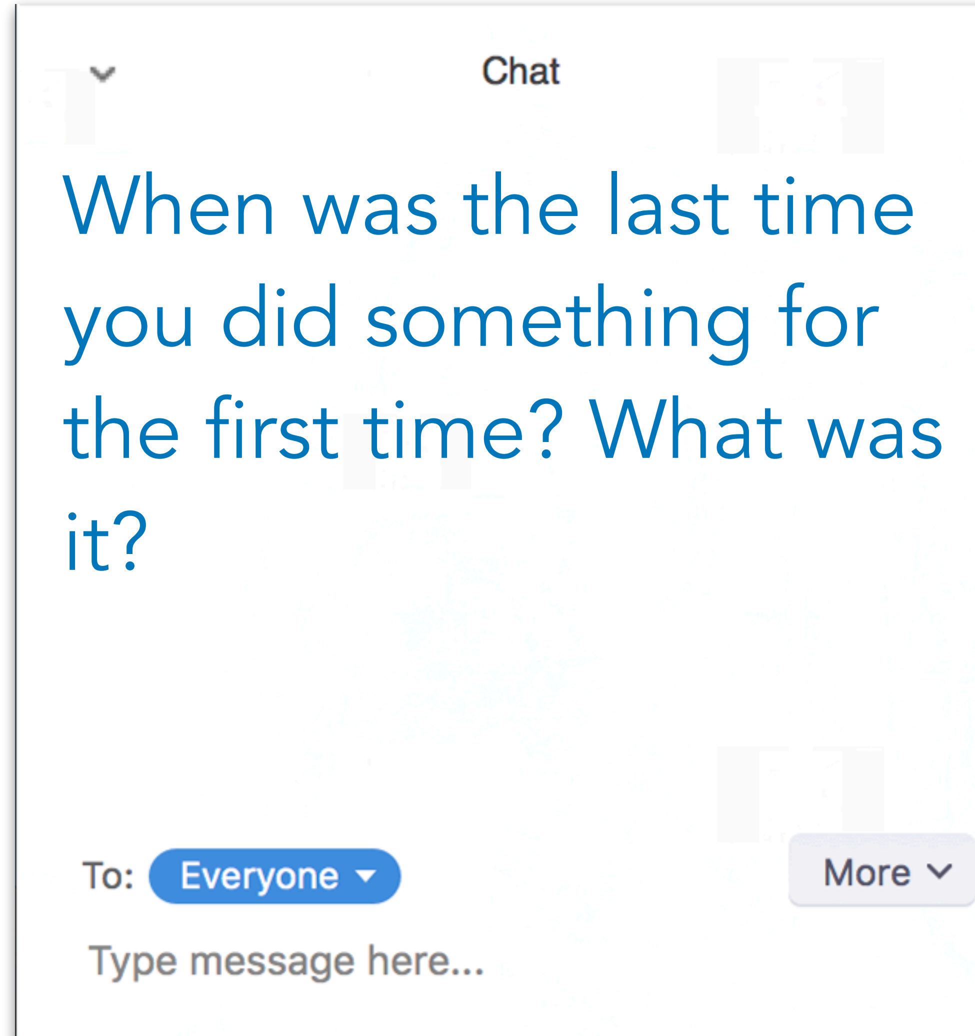
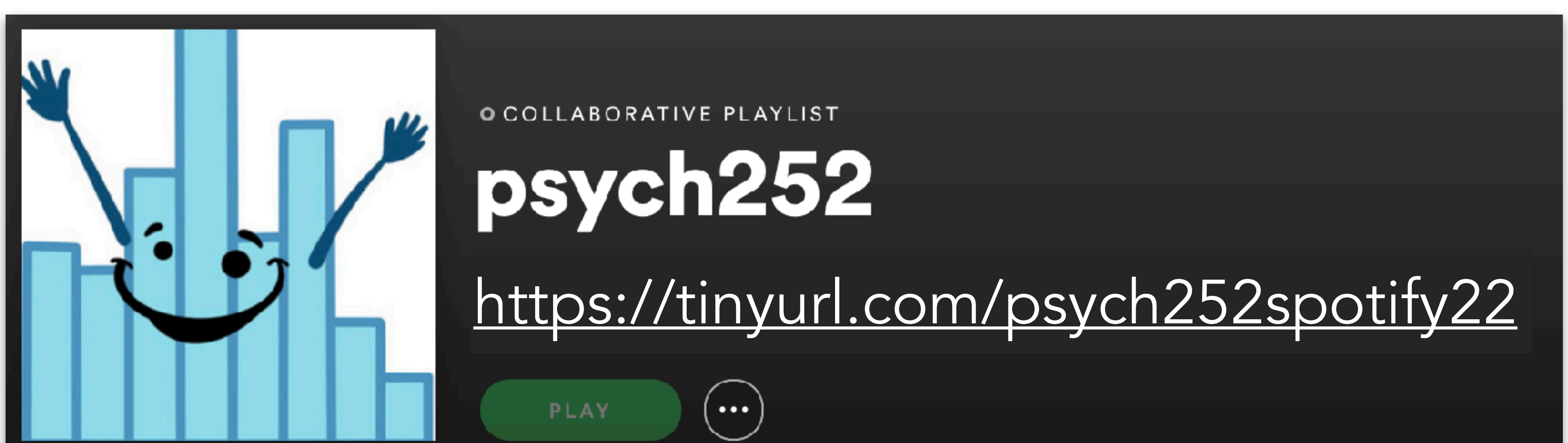
PSYCH 252: Statistical Methods for Behavioral and Social Sciences

Linear mixed effects models 3





We're listening to "Pundela" by "Titi Robin" submitted by Tobi

02/08/2022

Things that came up

Brenton Wiernik
@bmwiernik

A student in my R programming class shared this and I can't stop laughing about it

8:34 AM · Mar 1, 2021 · Twitter for iPhone

273 Retweets **29** Quote Tweets **2,012** Likes

Logistics

Homework 5

Part 1: Causal graphs (1.5 points)

For each graph, determine whether different variables are independent of each other. In addition to writing *Yes* or *No*, please also write the resulting graph of performing d-separation by listing the vertices and edges in alphabetical order, ignoring redundancies. For instance, the graph below in *Figure 1* would be described as

Vertices: A, B, C, D

Edges: A-B, A-C, A-D, B-D

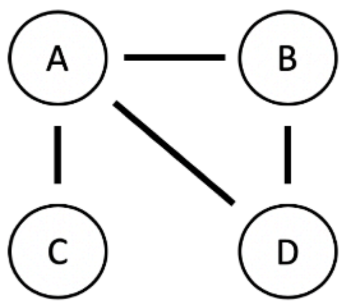


Figure 1: Figure 1

Part 2: Correlation constrains Causation (2.5 points)

It's intuitive to believe that additional years of compulsory education would increase yearly earnings, but a causal relationship is difficult to establish due to both practical and ethical concerns of randomly assigning years of required education. In this section, we will explore the dataset of a seminal work in economics by Angrist and Krueger that established a causal link between education and income through some very clever use of an instrumental variable. Here's the first paragraph from the paper.

> Every developed country in the world has a compulsory schooling requirement, yet little is known about the effect these laws have on educational attainment and earnings. This paper exploits an unusual natural experiment to estimate the impact of compulsory schooling laws in the United States. The experiment stems from the fact that children born in different months of the year start school at different ages, while compulsory schooling laws generally require students to remain in school until their sixteenth or seventeenth birthday. In effect, the interaction of school-entry requirements and compulsory schooling laws compel students born in certain months to attend school longer than students born in other months. Because one's birthday is unlikely to be correlated with personal attributes other than age at school entry, season of birth generates exogenous variation in education that can be used to estimate the impact of compulsory schooling on education and earnings.

```
df.qob = read_tsv("data/asciiqob.tab") %>%
  head(50000) # We don't need all 330000
head(df.qob) %>%
  kable()
```

log_weekly_wage	education	year_of_birth	quarter_of_birth	place_of_birth
5.790019	12	30	1	45
5.952494	11	30	1	45
5.315949	12	30	1	45
5.595926	12	30	1	45
6.068915	12	30	1	37
5.793871	11	30	1	45

Part 3: Cross-Validation (11 points)

In this section, we will be using some US Census data to find relationships between demographic data and average income within cities.

Although there are many different models we could construct from all these variables, we will be focusing on just the following 4 models:

1. `model_med_age: mean_income ~ median_age`
2. `model_med_age10: mean_income ~ median_age + median_age^2 + median_age^3 + ... + median_age^10`
3. `model_edu: mean_income ~ less9thgrade + grade9to12 + highschool + somecollege + assoc + bachelors + grad`
4. `model_race: mean_income ~ percent_white + percent_black + percent_amindian_alaskan + percent_asian + percent_nativeandother + percent_other_nativeandother + percent_hispanicorlatino + percent_race_other`

due Thursday 24th at 8pm

No class on Monday

Have a nice long weekend :)

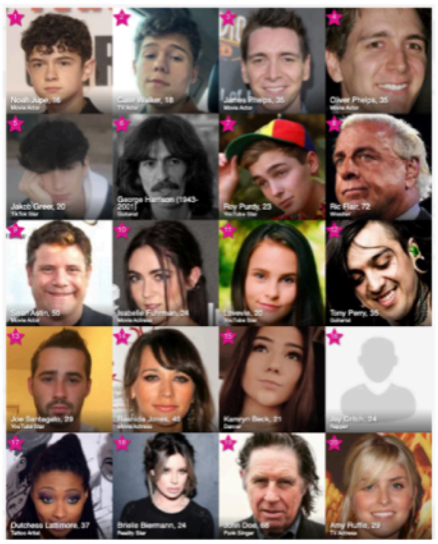
Plan for today

- Quick recap
- Understanding `lmer()` summary
- A worked example
- Reporting results
- Understanding `lmer()` syntax
- Let's simulate some `lmer()`s
- `lmer()` standard operating procedures
- Some more examples

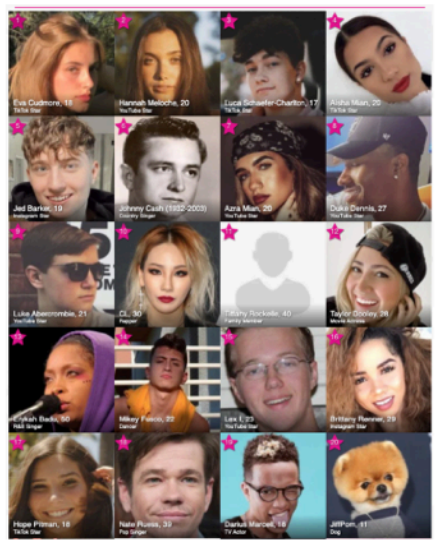
Quick recap

Quick recap: Linear mixed effects models

Are faces of people born on February 25th more trustworthy than faces of people born on February 26th?



born on February 25th



born on February 26th

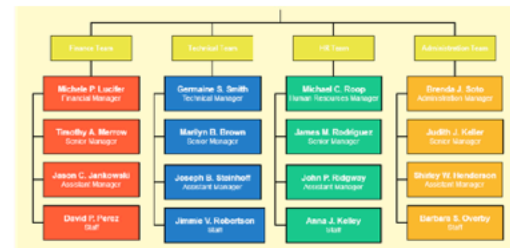
N = 100 participants

```
lmer(formula = trustworthy ~ 1 + birthday + (1 | item),
     data = df.birthday)
```

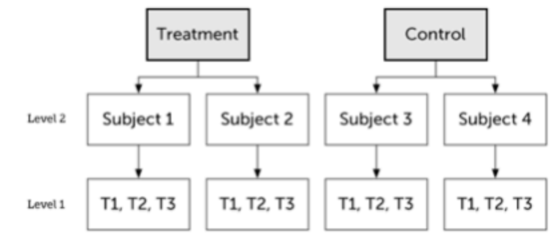
13

Linear mixed effects models

Hierarchical models



Longitudinal models



- allow us to account for dependencies in our data
- **hierarchical models**: schools > teachers > students
- **longitudinal models**: repeated observations from the same people

17

Random Slopes + Intercepts

It's reasonable to imagine that the most realistic situation is a combination of the scenarios described above:

> Faculty salaries start at different levels *and* increase at different rates depending on their department.

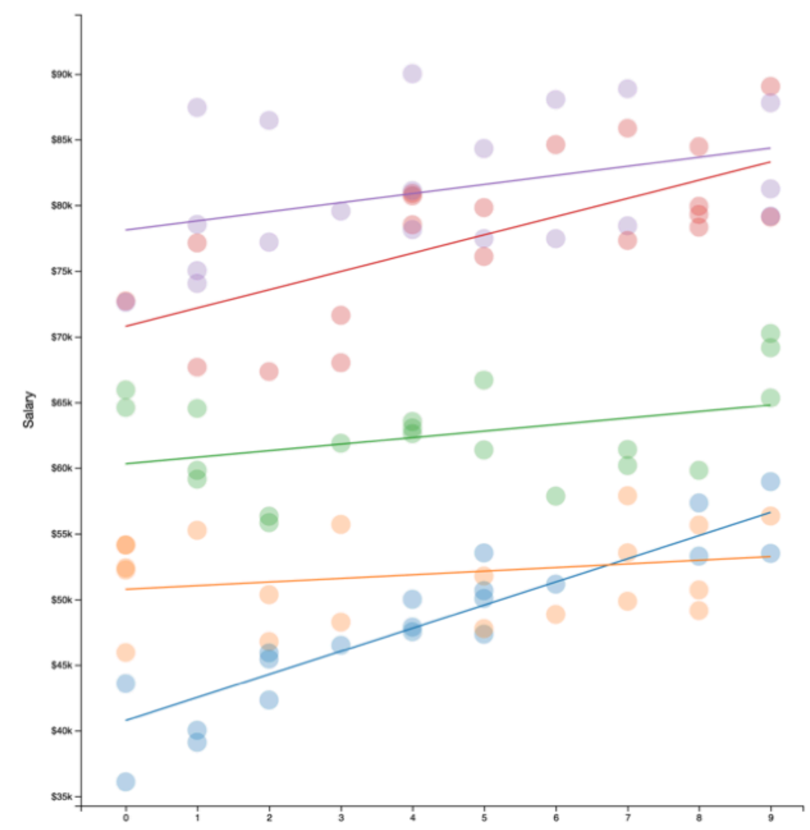
To incorporate both of these realities into our model, we want both the slope and the intercept to vary depending on the department of the faculty member. We can describe this with the following notation:

$$\hat{y}_i = \alpha_{j[i]} + \beta_{j[i]} x_i$$

Thus, the *starting salary* for faculty member i depends on their department ($\alpha_{j[i]}$), and their annual raise also varies by department ($\beta_{j[i]}$):

$$salary_i = \beta_{0j[i]} + \beta_{1j[i]} * experience_i$$

In order to implement any of these methods, you'll need to have a strong understanding of the phenomenon you're modeling, and how that is captured in the data. And, of course, you'll need to assess the performance of your models (not described here).

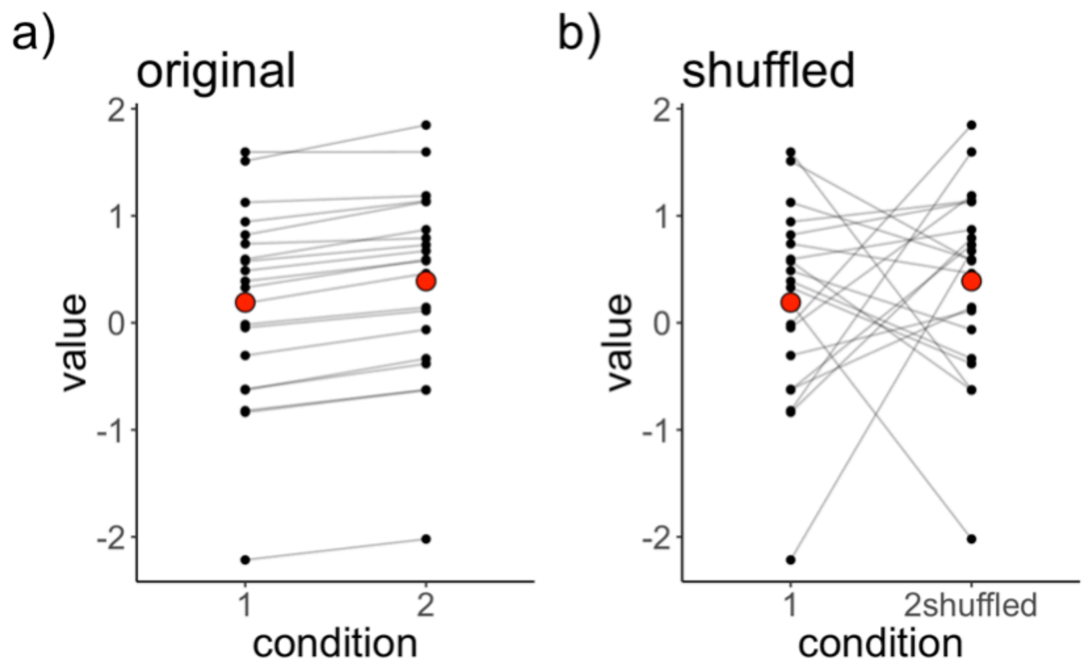


Quick recap: Dependence matters

Dependence

Does it really matter?

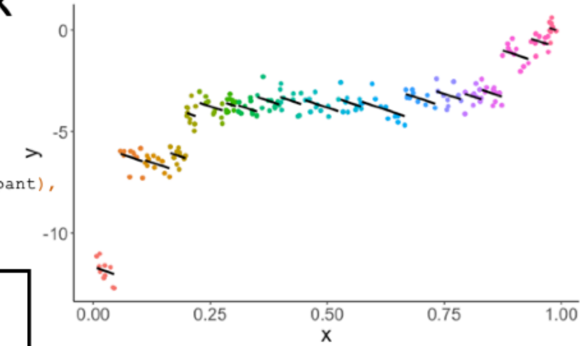
Is there a significant difference between conditions 1 and 2?



20

Simpson's paradox

```r
lmer(formula = y ~ 1 + x + (1 | participant),
     data = df.simpson) %>%
  summary()
```




```
Linear mixed model fit by REML ['lmerMod']
Formula: y ~ 1 + x + (1 | participant)
   Data: df.simpson

REML criterion at convergence: 345.1

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.43394 -0.59687  0.04493  0.62694  2.68828

Random effects:
 Groups      Name        Variance Std.Dev.
 participant (Intercept) 21.4898  4.6357
 Residual                 0.1661  0.4075
Number of obs: 200, groups:  participant, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept)  -0.1577     1.3230  -0.119
x            -7.6678     1.6572  -4.627

Correlation of Fixed Effects:
  (Intr)
x -0.621
```

negative (!) relationship between x and y

(once we take into account individual differences)

76

Linear model

```r
lm(formula = value ~ 1 + condition,
   data = df.original)
```

$\text{value}_i = b_0 + b_1 \cdot \text{condition}_i + e_i$

i = observation

$e_i \sim \mathcal{N}(\text{mean} = 0, \text{sd} = s_{\text{error}})$

3 parameters: b_0, b_1, s_{error}

Linear mixed effects model

```r
lmer(formula = value ~ 1 + condition +
       (1 | participant),
     data = df.original)
```

$\text{value}_{i,j} = b_0 + b_1 \cdot \text{condition}_{i,j} + U_i + e_i$

i = participant,
j = time point

$e_i \sim \mathcal{N}(\text{mean} = 0, \text{sd} = s_{\text{error}})$

$U_i \sim \mathcal{N}(\text{mean} = 0, \text{sd} = s_U)$

b_0, b_1 = fixed effects
U_i = random effect

here: random intercept

4 parameters: b_0, b_1, s_{error}, s_U

33

Model coefficients

Linear model

```r
fit = lm(formula = value ~ 1 + condition,
         data = df.original)
coef(fit)
```

```
(Intercept)  condition2
  0.1905239   0.1993528
```

- one intercept
- one slope for condition

Linear mixed effects model

```r
fit = lmer(formula = value ~ 1 + condition +
             (1 | participant),
           data = df.original)
coef(fit)
```

```
$participant
   (Intercept) condition2
1  -0.57839428  0.1993528
2   0.22299824  0.1993528
3  -0.82920677  0.1993528
4   1.49310938  0.1993528
5   0.36042775  0.1993528
6  -0.82060123  0.1993528
7   0.47929171  0.1993528
8   0.66401020  0.1993528
9   0.55135879  0.1993528
10 -0.28306703  0.1993528
11  1.57681676  0.1993528
12  0.38457642  0.1993528
13 -0.59969682  0.1993528
14 -2.21148391  0.1993528
15  1.05439374  0.1993528
16 -0.06476643  0.1993528
17 -0.03505690  0.1993528
18  0.93945348  0.1993528
19  0.87495531  0.1993528
20  0.63135911  0.1993528

attr(,"class")
[1] "coef.mer"
```

- different intercept for each participant
- one slope for condition

34

general points about `lmer()`

- **fixed effects**:
  - often: factors that we manipulate experimentally
  - parameters are estimated --> we are interested in characterizing the relationship between this variable and the outcome
- **random effects**:
  - variation we want to control for
  - often: differences between participants or items in our experiment

Understanding the `lmer()` summary

Understanding the `lmer()` summary

```r
# fit a linear mixed effects model
lmer(formula = value ~ condition + (1 | participant),
                       data = df.original) %>%
  summary()
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: value ~ condition + (1 | participant)
   Data: df.original

REML criterion at convergence: 17.3

Scaled residuals:
     Min       1Q   Median       3Q      Max 
-1.55996 -0.36399 -0.03341  0.34400  1.65823 

Random effects:
 Groups      Name        Variance Std.Dev.
 participant (Intercept) 0.816722 0.90373 
 Residual                0.003796 0.06161 
Number of obs: 40, groups:  participant, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept)  0.19052    0.20255   0.941
condition2   0.19935    0.01948  10.231

Correlation of Fixed Effects:
           (Intr)
condition2 -0.048
```

REML = restricted maximum likelihood

method for fitting models with **random effects**

Understanding the **lmer()** summary

```r
# fit a linear mixed effects model
lmer(formula = value ~ condition + (1 | participant),
                     data = df.original) %>%
  summary()
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: value ~ condition + (1 | participant)
   Data: df.original

REML criterion at convergence: 17.3

Scaled residuals:
     Min       1Q   Median       3Q      Max
-1.55996 -0.36399 -0.03341  0.34400  1.65823

Random effects:
 Groups      Name        Variance Std.Dev.
 participant (Intercept) 0.816722 0.90373
 Residual                0.003796 0.06161
Number of obs: 40, groups:  participant, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept)  0.19052    0.20255   0.941
condition2   0.19935    0.01948  10.231

Correlation of Fixed Effects:
           (Intr)
condition2 -0.048
```

fitting **lmer()** doesn't always work …

lmer() complains when it didn't work

Understanding the **lmer()** summary

```r
# fit a linear mixed effects model
lmer(formula = value ~ condition + (1 | participant),
                          data = df.original) %>%
  summary()
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: value ~ condition + (1 | participant)
   Data: df.original

REML criterion at convergence: 17.3

Scaled residuals: 
     Min       1Q   Median       3Q      Max 
-1.55996 -0.36399 -0.03341  0.34400  1.65823 

Random effects:
 Groups      Name        Variance Std.Dev.
 participant (Intercept) 0.816722 0.90373 
 Residual                0.003796 0.06161 
Number of obs: 40, groups:  participant, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept)  0.19052    0.20255   0.941
condition2   0.19935    0.01948  10.231

Correlation of Fixed Effects:
           (Intr)
condition2 -0.048
```

summary information about residuals

Understanding the **lmer()** summary

```r
# fit a linear mixed effects model
lmer(formula = value ~ condition + (1 | participant),
                          data = df.original) %>%
  summary()
```

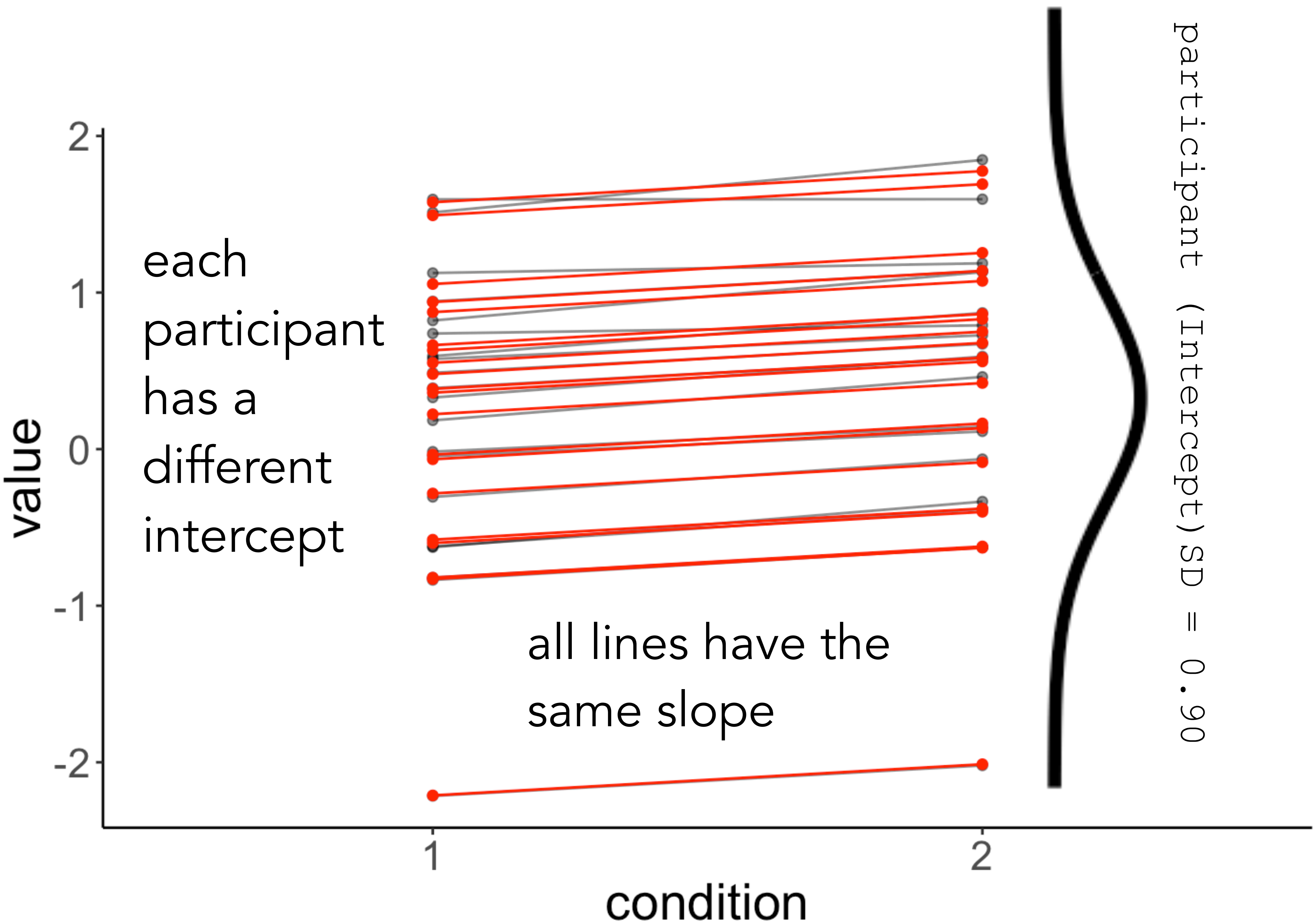
```
Linear mixed model fit by REML ['lmerMod']
Formula: value ~ condition + (1 | participant)
   Data: df.original

REML criterion at convergence: 17.3

Scaled residuals:
     Min       1Q   Median       3Q      Max 
-1.55996 -0.36399 -0.03341  0.34400  1.65823 

Random effects:
 Groups      Name        Variance Std.Dev.
 participant (Intercept) 0.816722 0.90373 
 Residual                0.003796 0.06161 
Number of obs: 40, groups:  participant, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept)  0.19052    0.20255   0.941
condition2   0.19935    0.01948  10.231

Correlation of Fixed Effects:
           (Intr)
condition2 -0.048
```

Random effects

one parameter to capture the variance between participants (gives us a sense for whether there are interindividual differences)

one parameter to capture the residual variance (just like sigma in an **lm()**)

Understanding the **`lmer()`** summary



Understanding the **lmer()** summary

```r
# fit a linear mixed effects model
lmer(formula = value ~ condition + (1 | participant),
                      data = df.original) %>%
  summary()
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: value ~ condition + (1 | participant)
   Data: df.original

REML criterion at convergence: 17.3

Scaled residuals:
     Min       1Q   Median       3Q      Max
-1.55996 -0.36399 -0.03341  0.34400  1.65823

Random effects:
 Groups      Name        Variance Std.Dev.
 participant (Intercept) 0.816722 0.90373
 Residual                0.003796 0.06161
Number of obs: 40, groups:  participant, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept)  0.19052    0.20255   0.941
condition2   0.19935    0.01948  10.231

Correlation of Fixed Effects:
           (Intr)
condition2 -0.048
```

Fixed effects

one parameter for the global intercept (value for the baseline condition)

one parameter for the condition effect (difference between the two conditions)

interpretation the same as for **lm()**, also: we can use contrasts!

Understanding the **lmer**() summary

```r
# fit a linear mixed effects model
lmer(formula = value ~ condition + (1 | participant),
                    data = df.original) %>%
  summary()
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: value ~ condition + (1 | participant)
   Data: df.original

REML criterion at convergence: 17.3

Scaled residuals:
     Min       1Q   Median       3Q      Max
-1.55996 -0.36399 -0.03341  0.34400  1.65823

Random effects:
 Groups      Name        Variance Std.Dev.
 participant (Intercept) 0.816722 0.90373
 Residual                0.003796 0.06161
Number of obs: 40, groups:  participant, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept)  0.19052    0.20255   0.941
condition2   0.19935    0.01948  10.231

Correlation of Fixed Effects:
           (Intr)
condition2 -0.048
```

correlation between intercept and condition2

The "correlation of fixed effects" output doesn't have the intuitive meaning that most would ascribe to it. Specifically, is not about the correlation of the variables. It is in fact about the expected correlation of the regression coefficients.

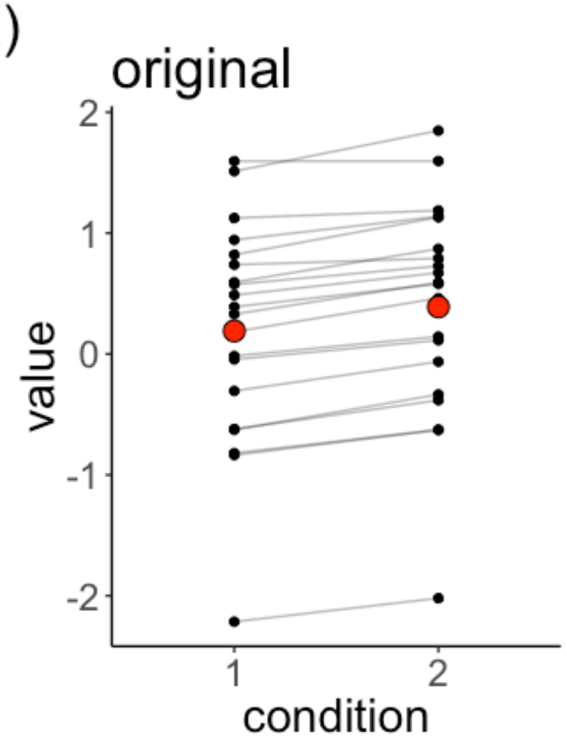
we just performed a paired t-test …

```r
t.test(df.original$value[df.original$condition == "1"],
       df.original$value[df.original$condition == "2"],
       alternative = "two.sided",
       paired = T)
```

```
Paired t-test

data:  df.original$value[df.original$condition == "1"] and
df.original$value[df.original$condition == "2"]
t = -10.231, df = 19, p-value = 3.636e-09
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.2401340 -0.1585717
sample estimates:
mean of the differences
            -0.1993528
```

If we take the differences for each participant between condition 1 and condition 2, are these difference scores significantly different from 0?

a)



we just performed a paired t-test …

```r
lmer(formula = value ~ condition + (1 | participant),
     data = df.original)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: value ~ condition + (1 | participant)
   Data: df.original

REML criterion at convergence: 17.3

Scaled residuals:
     Min       1Q   Median       3Q      Max
-1.55996 -0.36399 -0.03341  0.34400  1.65823

Random effects:
 Groups      Name        Variance Std.Dev.
 participant (Intercept) 0.816722 0.90373
 Residual                0.003796 0.06161
Number of obs: 40, groups:  participant, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept)  0.19052    0.20255   0.941
condition2   0.19935    0.01948  10.231

Correlation of Fixed Effects:
           (Intr)
condition2 -0.048
```

- explicitly models the interindividual variation
- much more flexible …

```r
t.test(df.original$value[df.original$condition == "1"],
       df.original$value[df.original$condition == "2"],
       alternative = "two.sided",
       paired = T)
```

```
Paired t-test

data:  df.original$value[df.original$condition == "1"] and
df.original$value[df.original$condition == "2"]
t = -10.231, df = 19, p-value = 3.636e-09
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.2401340 -0.1585717
sample estimates:
mean of the differences
             -0.1993528
```

general points about `lmer()`

- Why don't we just run individual regressions?
  - overfitting ...
  - inflating type 1 error
  - larger uncertainty in parameter estimates because only few data points are used for each model
  - unclear how to aggregate the results to make an overall statement
- Why don't we just run a regression on the means?
  - we throw away a lot of information
  - what to do when the design is unbalanced?
- Mixed effects model:
  - makes use of all available information
  - addresses the main problems of the other two approaches

let's take a look at an example

A worked example

Higher Order Functions   Posts   About Me   CV

Tristan Mahr

Language and data scientist

- Madison, WI
- Email
- Twitter
- GitHub
- Stackoverflow
- R Bloggers

Plotting partial pooling in mixed-effects models

In this post, I demonstrate a few techniques for plotting information from a relatively simple mixed-effects model fit in R. These plots can help us develop intuitions about what these models are doing and what "partial pooling" means.

The `sleepstudy` dataset

For these examples, I'm going to use the `sleepstudy` dataset from the lme4 package. The outcome measure is reaction time, the predictor measure is days of sleep deprivation, and these measurements are nested within participants—we have 10 observations per participant. I am also going to add two fake participants with incomplete data to illustrate partial pooling.

https://www.tjmahr.com/plotting-partial-pooling-in-mixed-effects-models/

Data set

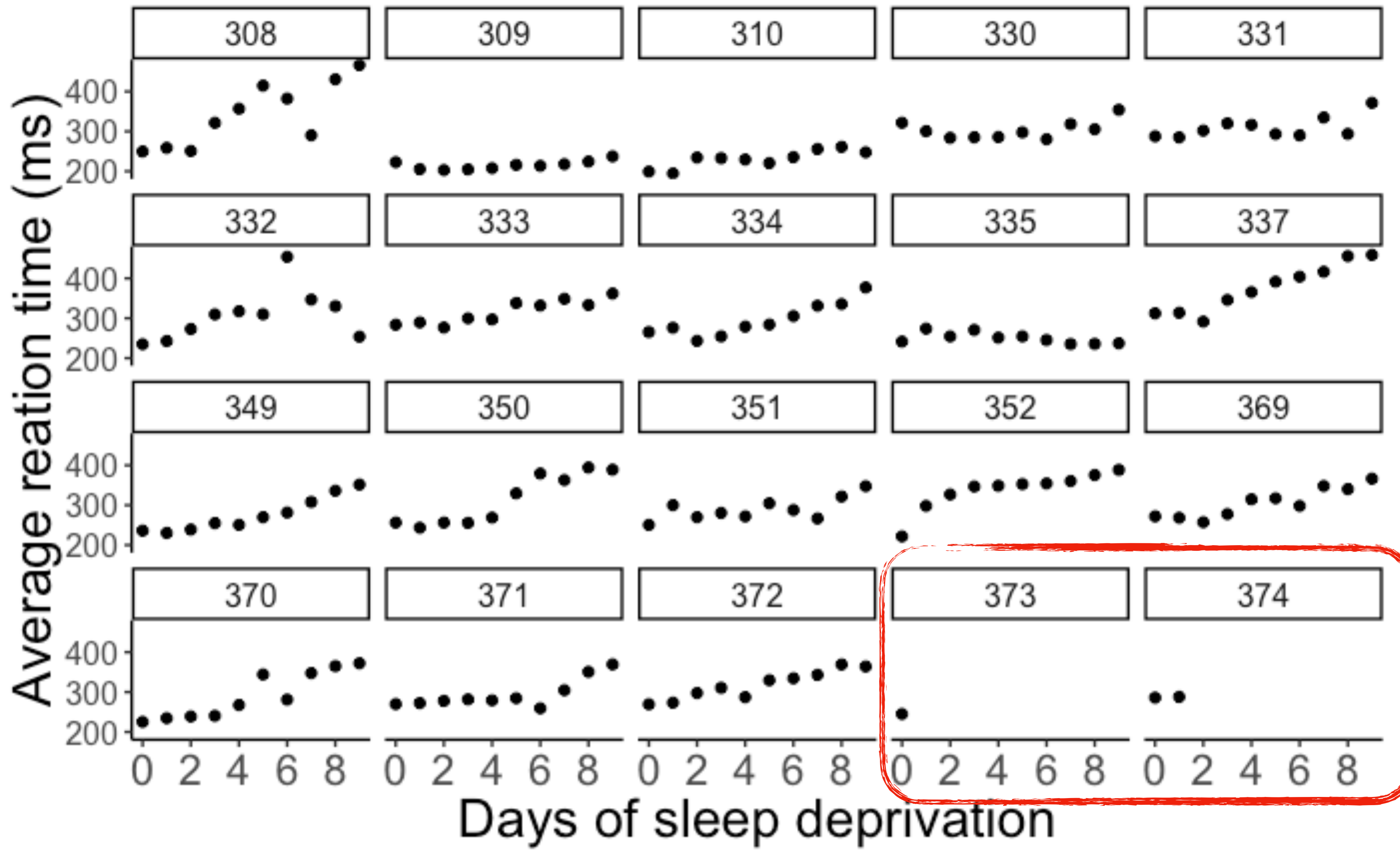
How does sleep deprivation affect reaction time?

subject	days	reaction
308	0	249.56
308	1	258.70
308	2	250.80
308	3	321.44
308	4	356.85
309	0	222.73
309	1	205.27
309	2	202.98
309	3	204.71
309	4	207.72

Data set

How does sleep deprivation affect reaction time?

subject	days	reaction
308	0	249.56
308	1	258.70
308	2	250.80
308	3	321.44
308	4	356.85
309	0	222.73
309	1	205.27
309	2	202.98
309	3	204.71
309	4	207.72



20 participants

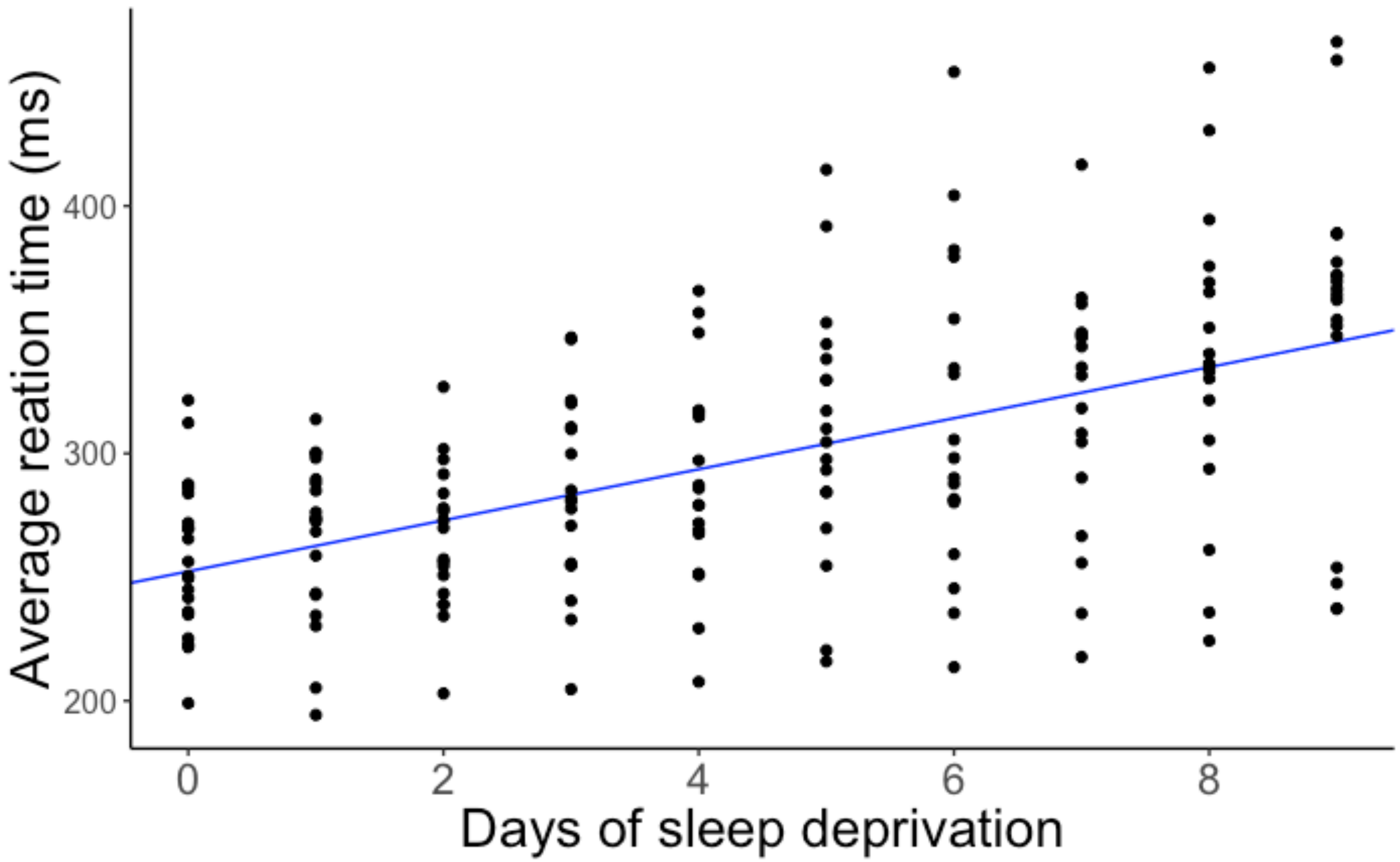
2 with incomplete information

Pooling information

- **complete pooling**
  - combine data from all participants and fit one global regression
- **no pooling**
  - don't combine any of the data and fit a separate regression to each individual participant
- **partial pooling**
  - take into account all information by explicitly modeling the variation between participants

Complete pooling: Fit one global regression

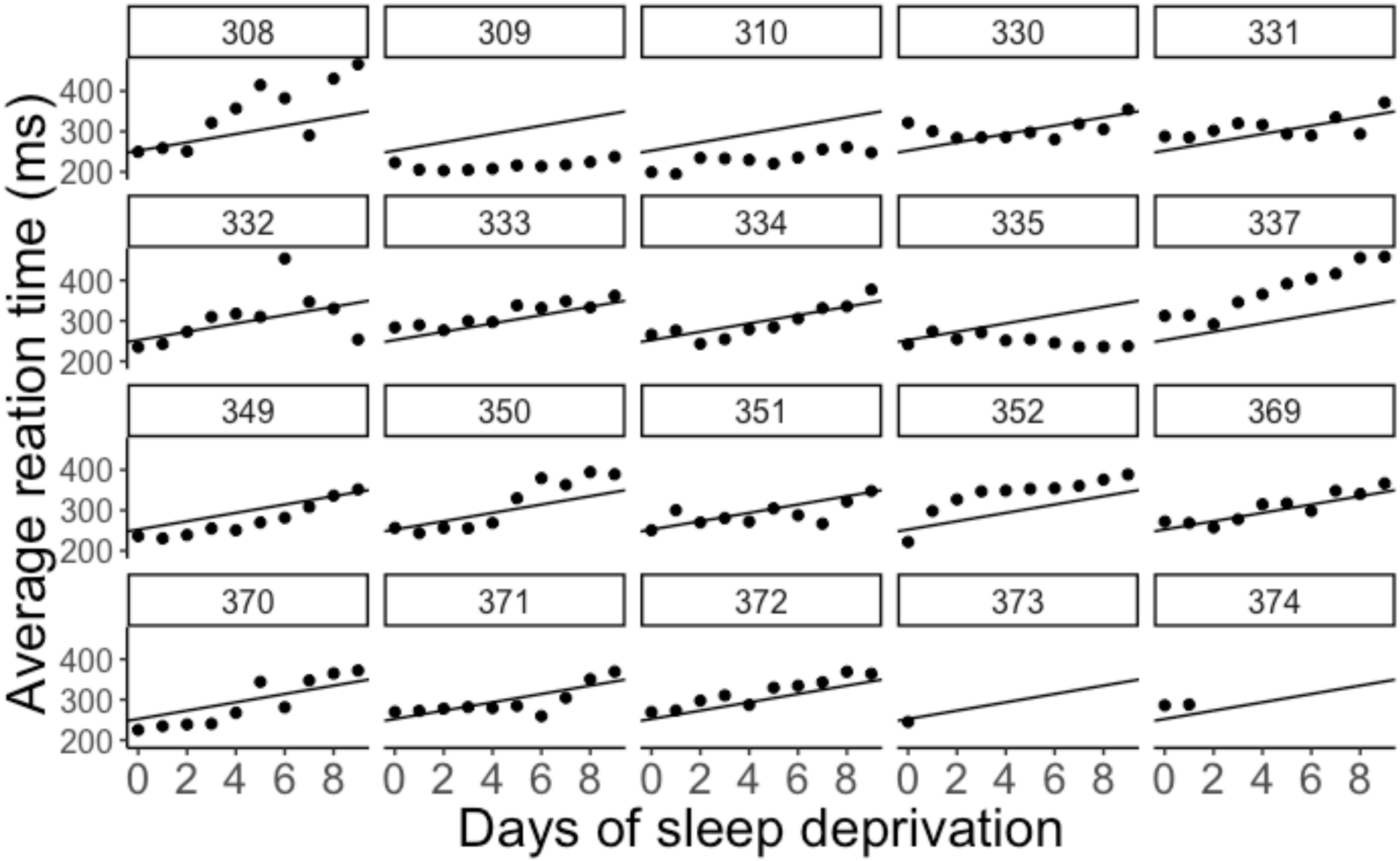
```
lm(formula = reaction ~ days,
   data = df.sleep)
```



Complete pooling: Fit one global regression

```
lm(formula = reaction ~ days,
   data = df.sleep)
```

same line for
each participant



No pooling: Fit separate regressions

```
df.no_pooling = df.sleep %>%
  group_by(subject) %>%
  nest(data = c(days, reaction)) %>%
  mutate(fit = map(data, ~ lm(reaction ~ days, data = .)),
         params = map(fit, tidy)) %>%
  unnest(c(params)) %>%
  select(subject, term, estimate) %>%
  complete(subject, term, fill = list(estimate = 0)) %>%
  pivot_wider(names_from = term, values_from = estimate) %>%
  clean_names()
```

data | regression fit | extracted parameters

	subject	data	fit	params
1	308	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(2...	list(coefficients = c(\`(Intercept)\` = 244.1926690909...	list(term = c("(Intercept)", "days"), estimate = c(244.1...
2	309	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(2...	list(coefficients = c(\`(Intercept)\` = 205.0549454545...	list(term = c("(Intercept)", "days"), estimate = c(205.0...
3	310	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(1...	list(coefficients = c(\`(Intercept)\` = 203.4842254545...	list(term = c("(Intercept)", "days"), estimate = c(203.4...
4	330	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(3...	list(coefficients = c(\`(Intercept)\` = 289.6850927272...	list(term = c("(Intercept)", "days"), estimate = c(289.6...
5	331	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(2...	list(coefficients = c(\`(Intercept)\` = 285.7389654545...	list(term = c("(Intercept)", "days"), estimate = c(285.7...
6	332	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(2...	list(coefficients = c(\`(Intercept)\` = 264.2516145454...	list(term = c("(Intercept)", "days"), estimate = c(264.2...
7	333	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(2...	list(coefficients = c(\`(Intercept)\` = 275.0191054545...	list(term = c("(Intercept)", "days"), estimate = c(275.0...
8	334	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(2...	list(coefficients = c(\`(Intercept)\` = 240.1629145454...	list(term = c("(Intercept)", "days"), estimate = c(240.1...
9	335	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(2...	list(coefficients = c(\`(Intercept)\` = 263.0346927272...	list(term = c("(Intercept)", "days"), estimate = c(263.0...
10	337	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(3...	list(coefficients = c(\`(Intercept)\` = 290.1041272727...	list(term = c("(Intercept)", "days"), estimate = c(290.1...
11	349	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(2...	list(coefficients = c(\`(Intercept)\` = 215.1117727272...	list(term = c("(Intercept)", "days"), estimate = c(215.1...
			⋮	
19	374	list(days = c(0, 1), reaction = c(286, 288))	list(coefficients = c(\`(Intercept)\` = 286, days = 2.000...	list(term = c("(Intercept)", "days"), estimate = c(286, 2...
20	373	list(days = 0, reaction = 245)	list(coefficients = c(\`(Intercept)\` = 245, days = NA), r...	list(term = "(Intercept)", estimate = 245, std.error = ...

No pooling: Fit separate regressions

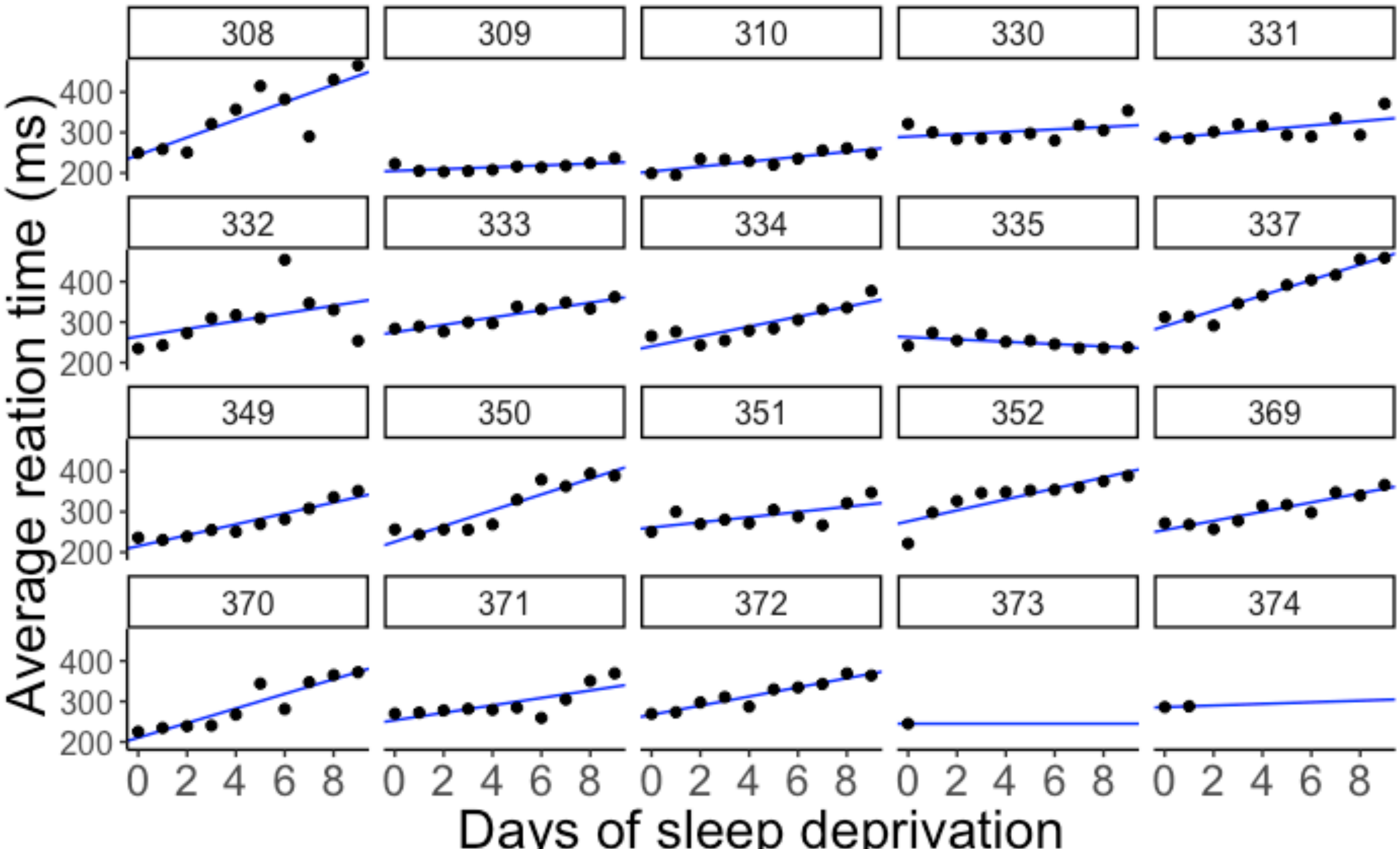
```r
df.no_pooling = df.sleep %>%
  group_by(subject) %>%
  nest(data = c(days, reaction)) %>%
  mutate(fit = map(data, ~ lm(reaction ~ days, data = .)),
         params = map(fit, tidy)) %>%
  unnest(c(params)) %>%
  select(subject, term, estimate) %>%
  complete(subject, term, fill = list(estimate = 0)) %>%
  pivot_wider(names_from = term, values_from = estimate) %>%
  clean_names()
```

	subject	intercept	days
1	308	244.1927	21.764702
2	309	205.0549	2.261785
3	310	203.4842	6.114899
4	330	289.6851	3.008073
5	331	285.7390	5.266019
6	332	264.2516	9.566768
7	333	275.0191	9.142045
8	334	240.1629	12.253141
9	335	263.0347	-2.881034
10	337	290.1041	19.025974
11	349	215.1118	13.493933
12	350	225.8346	19.504017
13	351	261.1470	6.433498
14	352	276.3721	13.566549
15	369	254.9681	11.348109
16	370	210.4491	18.056151
17	371	253.6360	9.188445
18	372	267.0448	11.298073
19	373	245.0000	0.000000
20	374	286.0000	2.000000

separate intercept and slope for each participant

No pooling: Fit separate regressions

different line for
each participant



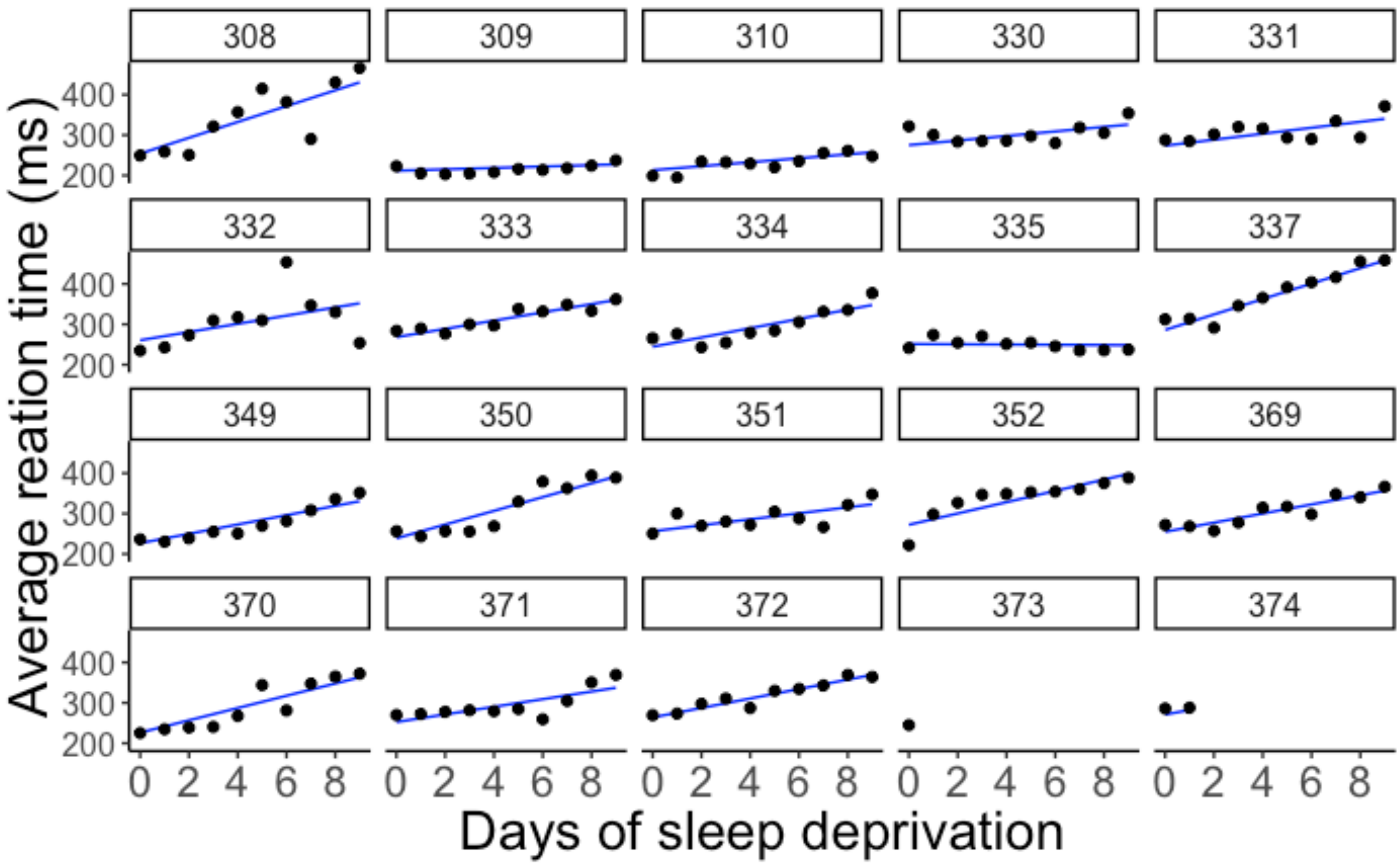
Partial pooling: Fit mixed effects model

intercepts and slopes differ between participants

random intercept

random slope

```
lmer(formula = reaction ~ 1 + days + (1 + days | subject),
     data = df.sleep)
```

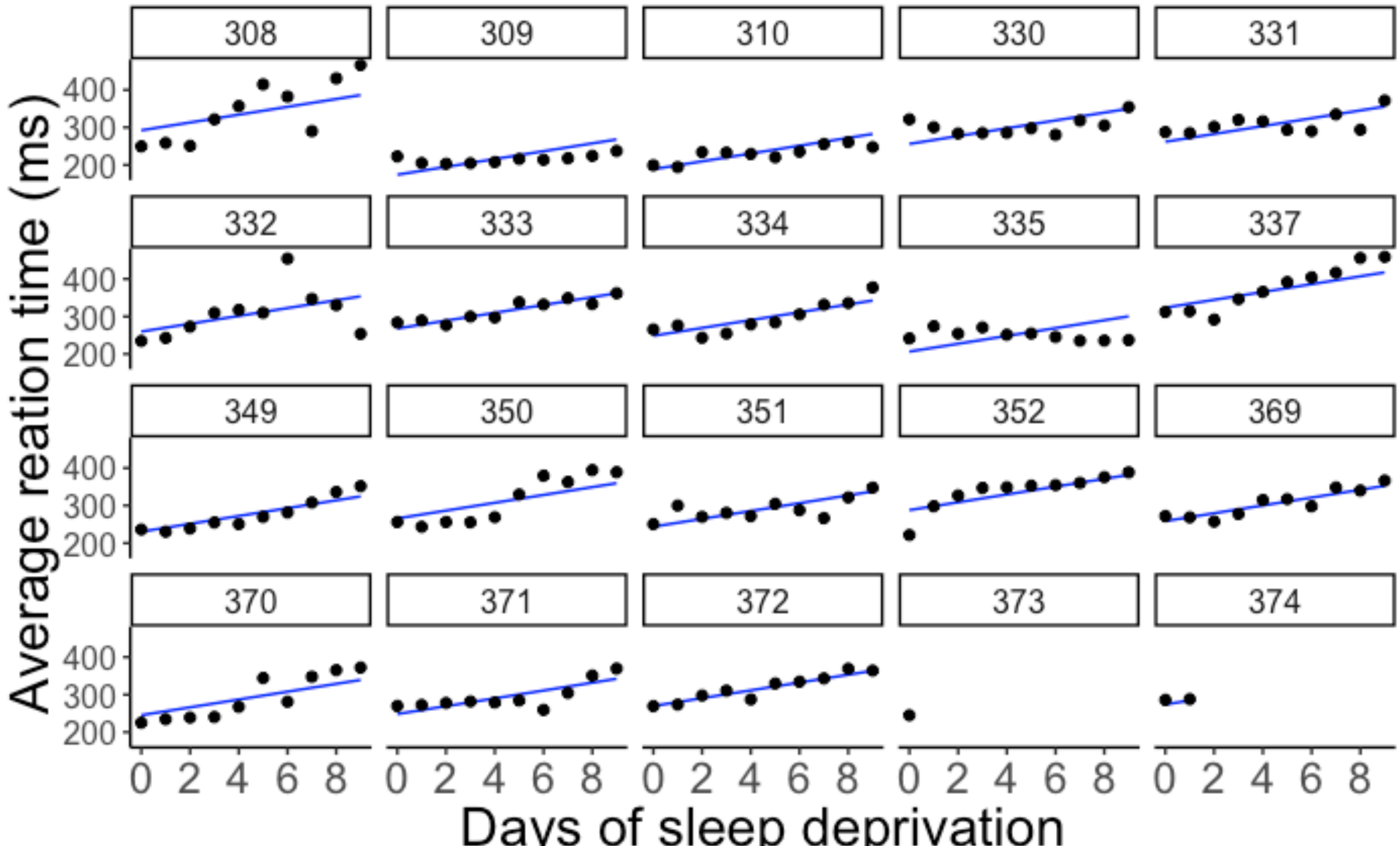


Partial pooling: Fit mixed effects model

only intercepts differ
between participants

random intercept

```
lmer(formula = reaction ~ 1 + days + (1 | subject),
     data = df.sleep)
```

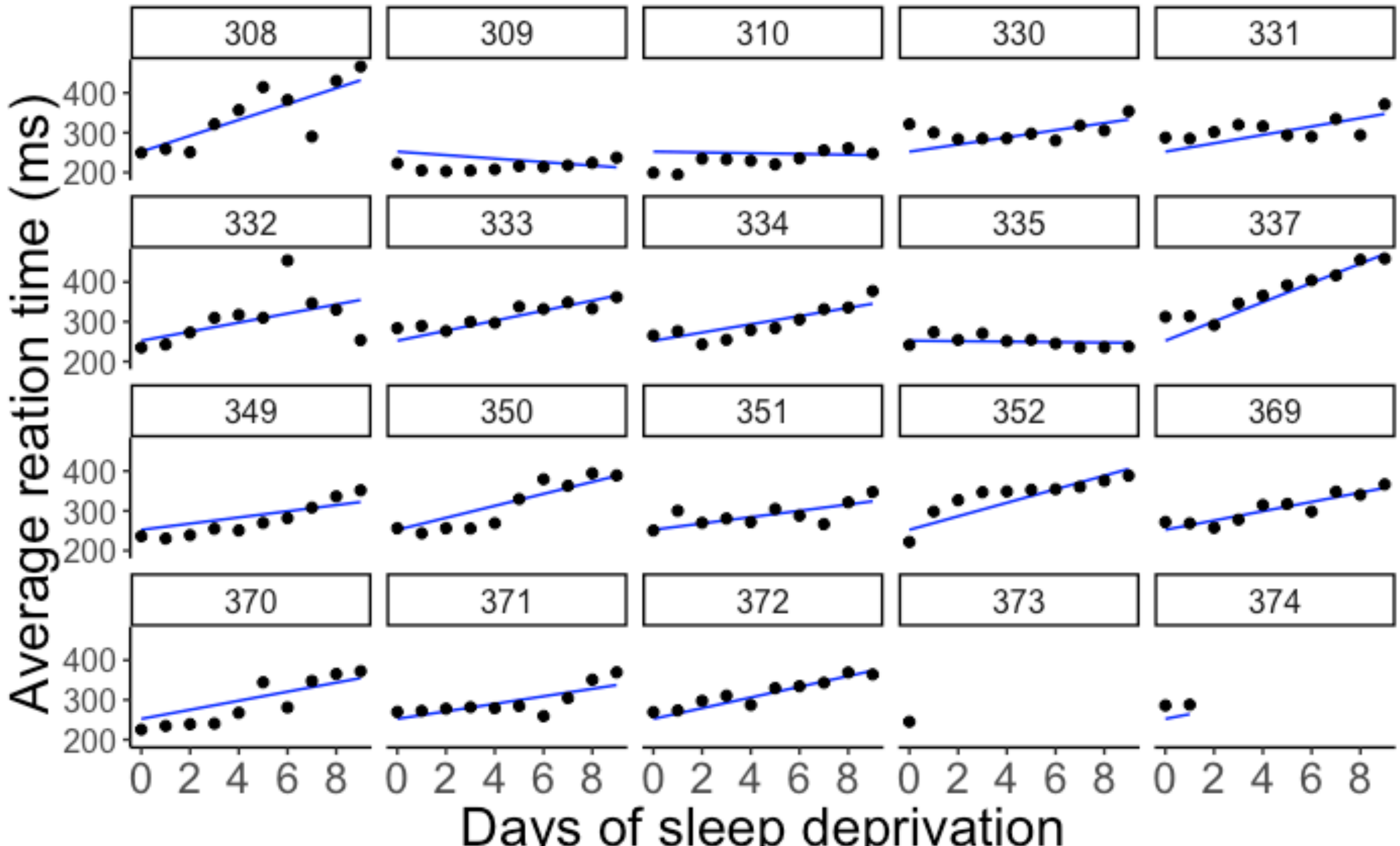


Partial pooling: Fit mixed effects model

only slopes differ between participants

random slope

```
lmer(formula = reaction ~ 1 + days + (0 + days | subject),
     data = df.sleep)
```



Coefficients

```
lmer(formula = reaction ~ 1 + days + ...,
     data = df.sleep)
```

`(1 | subject)`

random intercepts

`(0 + days | subject)`

random slopes

`(1 + days | subject)`

random intercepts and slopes (+ correlation)

$subject

	(Intercept)	days
308	292.2749	10.43191
309	174.0559	10.43191
310	188.7454	10.43191
330	256.0247	10.43191
331	261.8141	10.43191
332	259.8262	10.43191
333	268.0765	10.43191
334	248.6471	10.43191
335	206.5096	10.43191
337	323.5643	10.43191
349	230.5114	10.43191
350	265.6957	10.43191
351	243.7988	10.43191
352	287.8850	10.43191
369	258.6454	10.43191
370	245.2931	10.43191
371	248.3508	10.43191
372	269.6861	10.43191
373	248.2086	10.43191
374	273.9400	10.43191

$subject

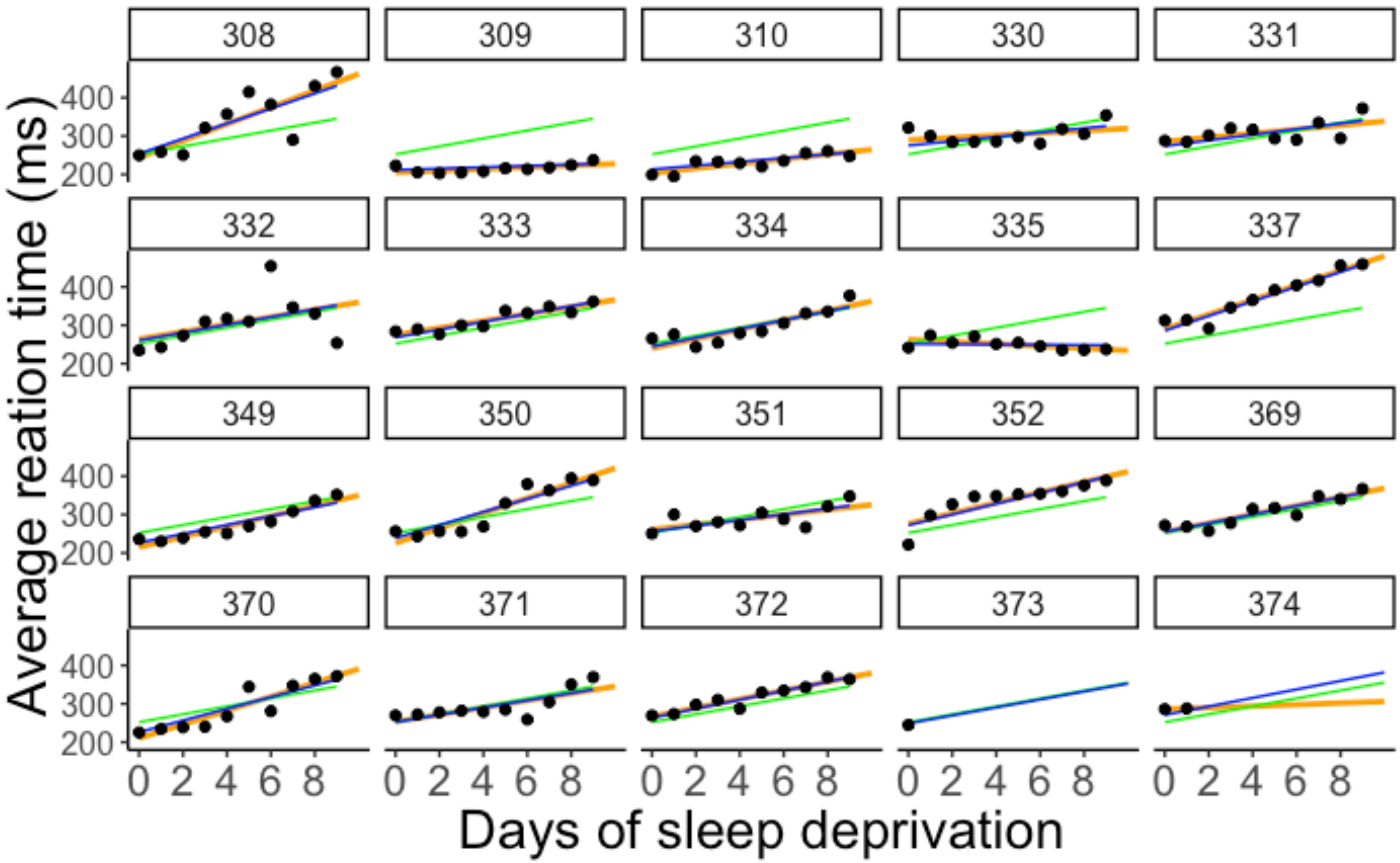
	(Intercept)	days
308	252.2965	19.9526801
309	252.2965	-4.3719650
310	252.2965	-0.9574726
330	252.2965	8.9909957
331	252.2965	10.5394285
332	252.2965	11.3994289
333	252.2965	12.6074020
334	252.2965	10.3413879
335	252.2965	-0.5722073
337	252.2965	24.2246485
349	252.2965	7.7702676
350	252.2965	15.0661415
351	252.2965	7.9675415
352	252.2965	17.0002999
369	252.2965	11.6982767
370	252.2965	11.3939807
371	252.2965	9.4535879
372	252.2965	13.4569059
373	252.2965	10.4142695
374	252.2965	11.9097917

$subject

	(Intercept)	days
308	253.9479	19.6264139
309	211.7328	1.7319567
310	213.1579	4.9061843
330	275.1425	5.6435987
331	273.7286	7.3862680
332	260.6504	10.1632535
333	268.3684	10.2245979
334	244.5523	11.4837825
335	251.3700	-0.3355554
337	286.2321	19.1090061
349	226.7662	11.5531963
350	238.7807	17.0156766
351	256.2344	7.4119501
352	272.3512	13.9920698
369	254.9484	11.2985741
370	226.3701	15.2027922
371	252.5051	9.4335432
372	263.8916	11.7253342
373	248.9752	10.3915245
374	271.1451	11.0782697

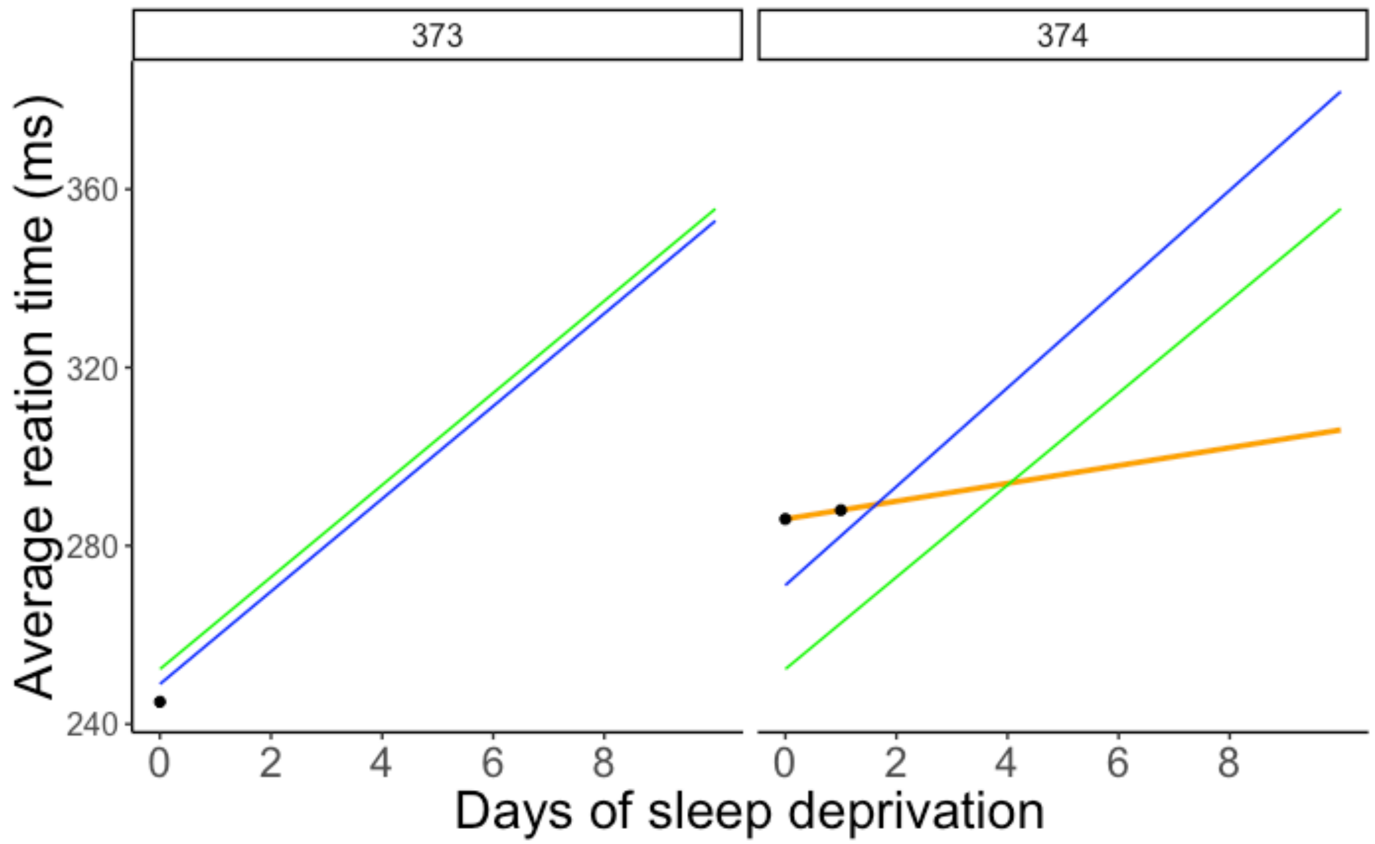
Comparison

complete pooling
no pooling
partial pooling



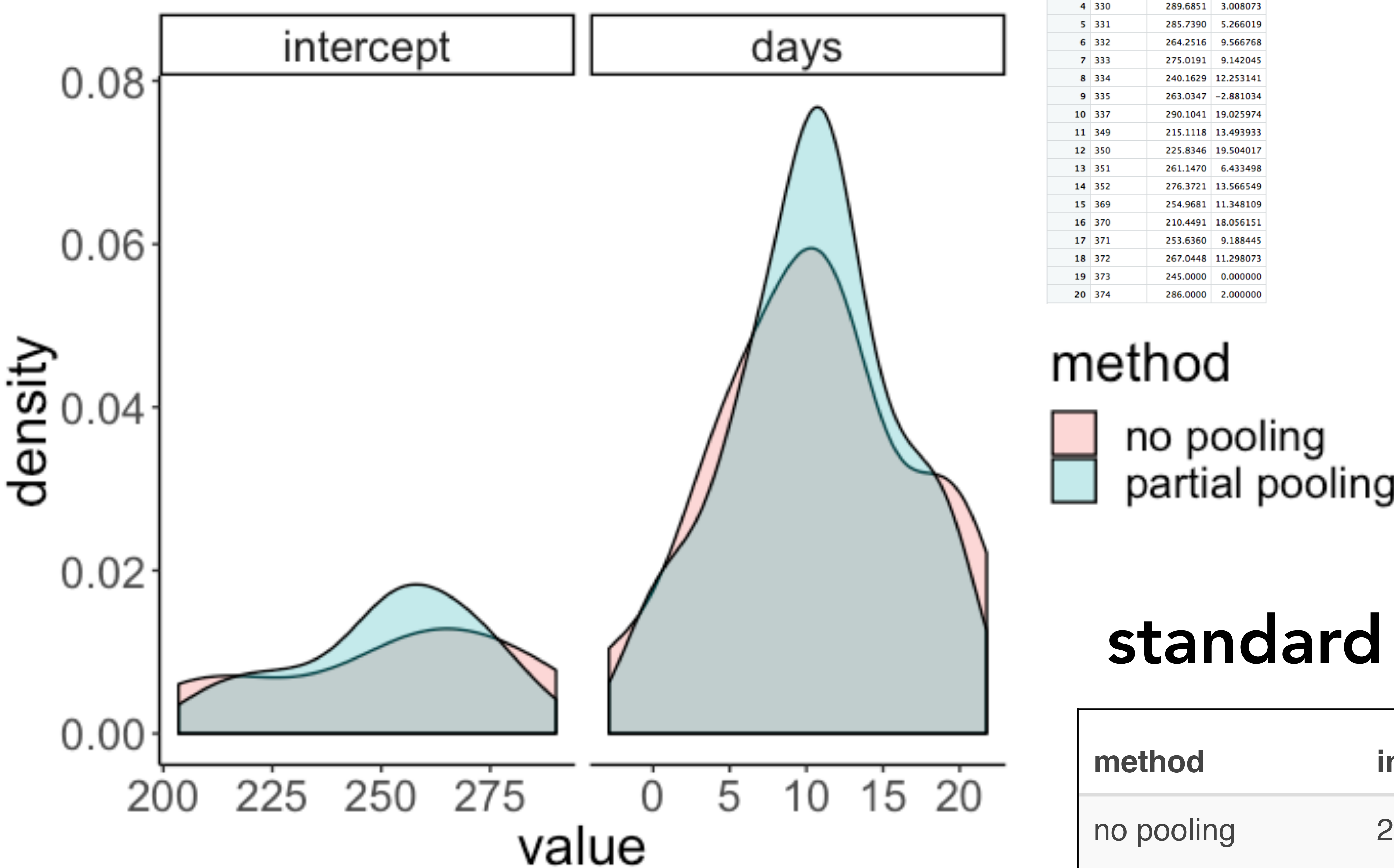
Comparison

complete pooling
no pooling
partial pooling



- **complete pooling**:
  - doesn't account for any individual variation
- **no pooling**:
  - doesn't yield predictions when we only have observation
  - doesn't consider the general effect of sleep deprivation when making predictions
- **partial pooling**:
  - draws on all the information in the data
  - extrapolates based on information about the individual participants, as well as information based on the whole sample

Shrinkage



separate regression
for each participant

	subject	intercept	days
1	308	244.1927	21.764702
2	309	205.0549	2.261785
3	310	203.4842	6.114899
4	330	289.6851	3.008073
5	331	285.7390	5.266019
6	332	264.2516	9.566768
7	333	275.0191	9.142045
8	334	240.1629	12.253141
9	335	263.0347	-2.881034
10	337	290.1041	19.025974
11	349	215.1118	13.493933
12	350	225.8346	19.504017
13	351	261.1470	6.433498
14	352	276.3721	13.566549
15	369	254.9681	11.348109
16	370	210.4491	18.056151
17	371	253.6360	9.188445
18	372	267.0448	11.298073
19	373	245.0000	0.000000
20	374	286.0000	2.000000

lmer()

```
(1 + days | subject)
```

random intercepts and slopes (+ correlation)

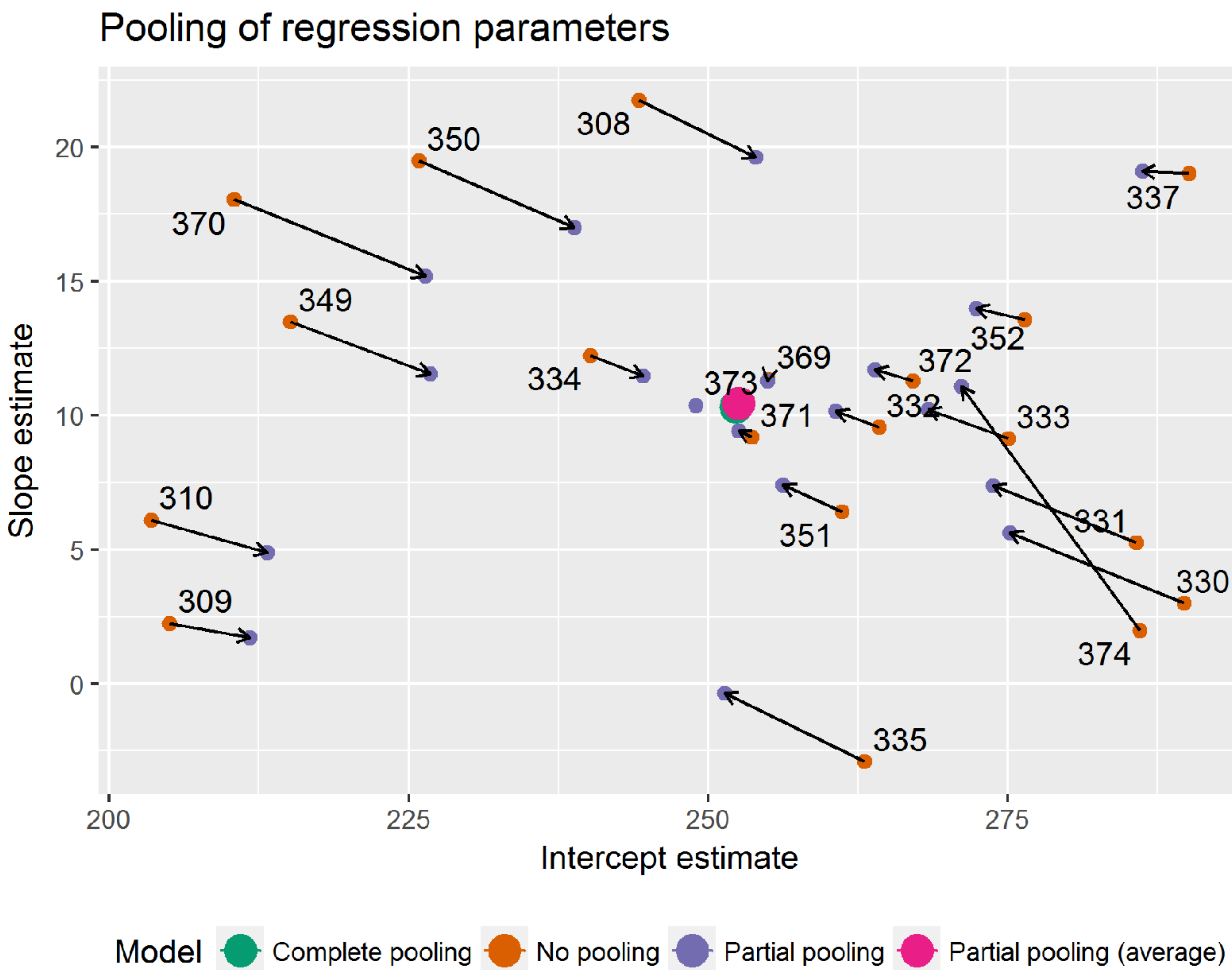
```
$subject
     (Intercept)       days
308     253.9479 19.6264139
309     211.7328  1.7319567
310     213.1579  4.9061843
330     275.1425  5.6435987
331     273.7286  7.3862680
332     260.6504 10.1632535
333     268.3684 10.2245979
334     244.5523 11.4837825
335     251.3700 -0.3355554
337     286.2321 19.1090061
349     226.7662 11.5531963
350     238.7807 17.0156766
351     256.2344  7.4119501
352     272.3512 13.9920698
369     254.9484 11.2985741
370     226.3701 15.2027922
371     252.5051  9.4335432
372     263.8916 11.7253342
373     248.9752 10.3915245
374     271.1451 11.0782697
```

standard deviation

method	intercept	days
no pooling	28.95	6.56
partial pooling	21.59	5.46

variance "shrinks"

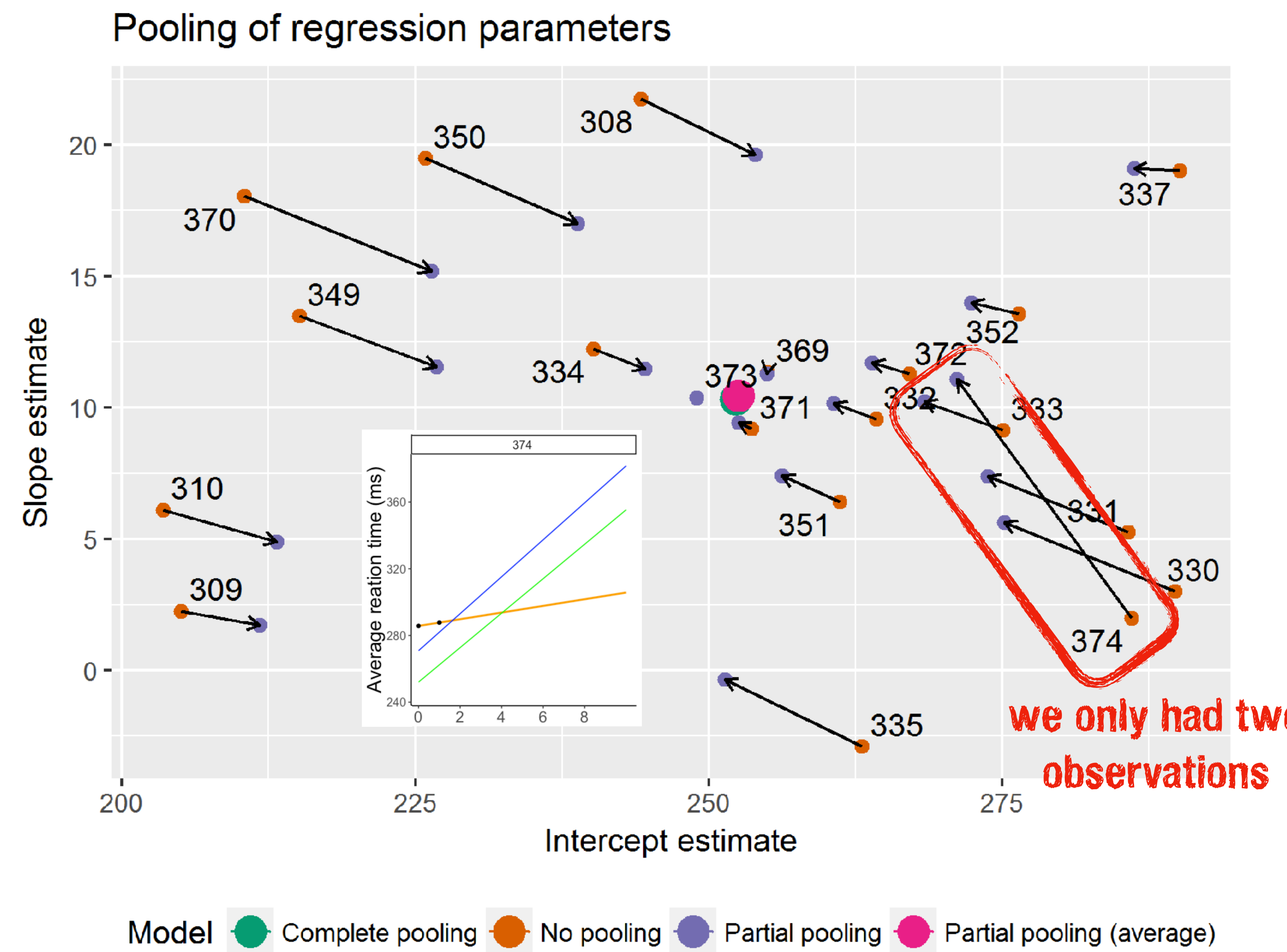
Shrinkage



Pooling of regression parameters
Slope estimate
Intercept estimate
20
15
10
5
0
200
225
250
275
308
350
337
370
349
334
373
369
372
352
371
332
333
310
351
331
330
309
374
335
Model
Complete pooling
No pooling
Partial pooling
Partial pooling (average)

Shrinkage

- more shrinkage when estimate is further from the average
- more shrinkage when estimate is more uncertain (based on fewer observations); more information "borrowed" from other clusters

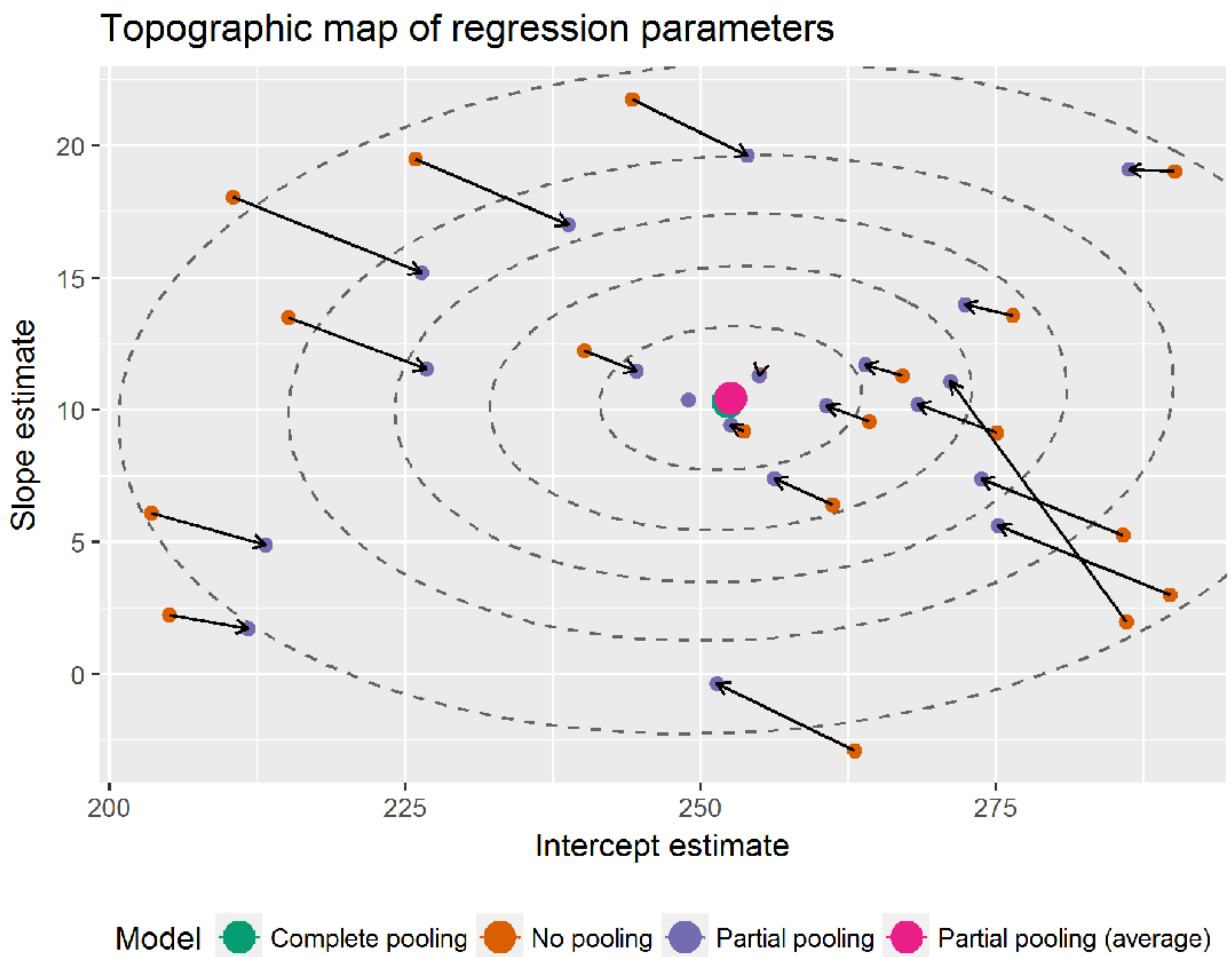


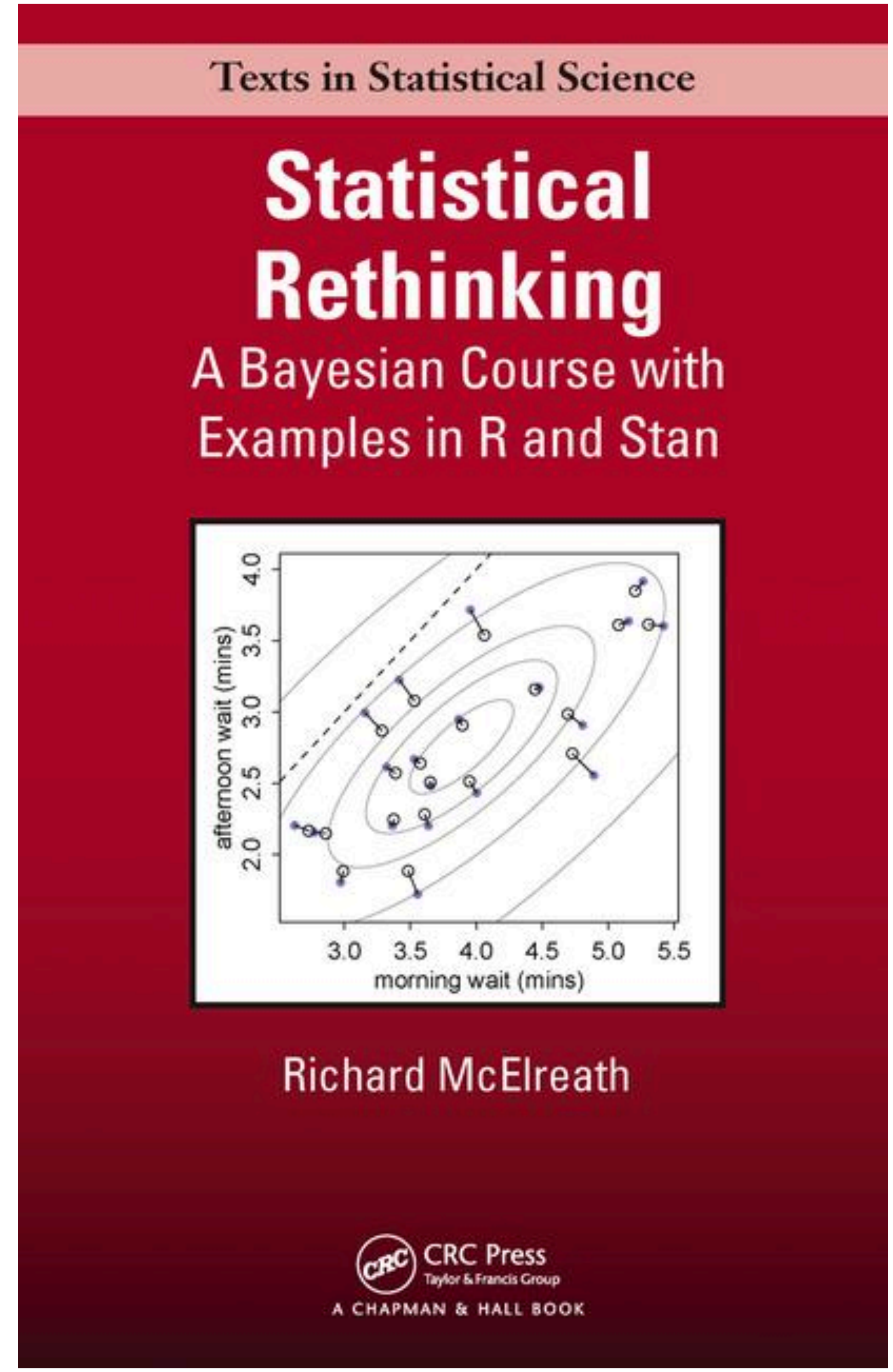
In the lme4 book, Douglas Bates provides an alternative to *shrinkage*:

The term "shrinkage" may have negative connotations. John Tukey preferred to refer to the process as the estimates for individual subjects **"borrowing strength" from each other**.

This is a fundamental difference in the models underlying mixed-effects models versus strictly fixed effects models. In a mixed-effects model we assume that the levels of a grouping factor are a selection from a population and, as a result, can be expected to share characteristics to some degree. Consequently, the predictions from a mixed-effects model are attenuated relative to those from strictly fixed-effects models.

Shrinkage

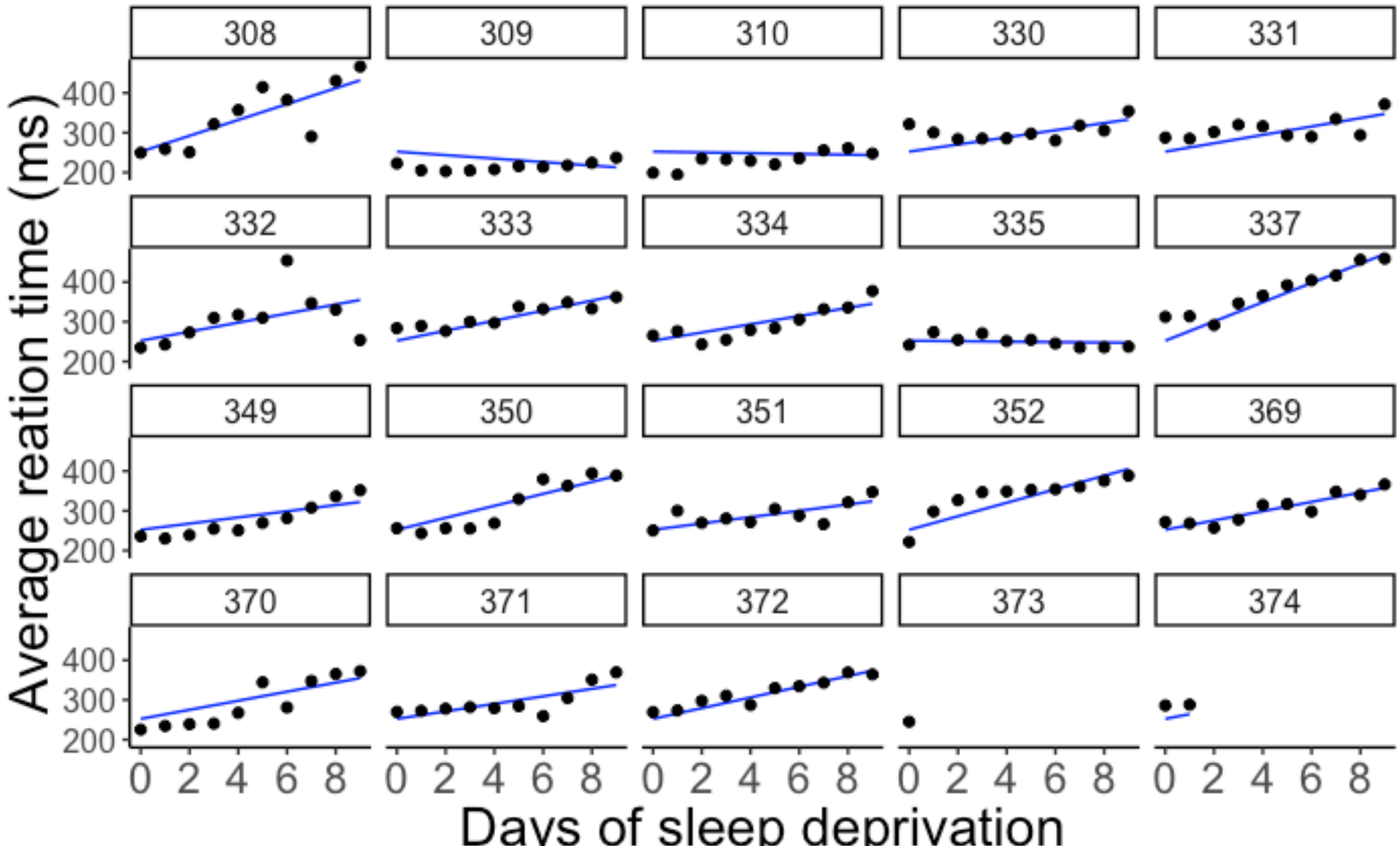




mixed effects model estimates a multi-variate Gaussian to account for (possible) correlations between intercepts and slopes

Reporting results

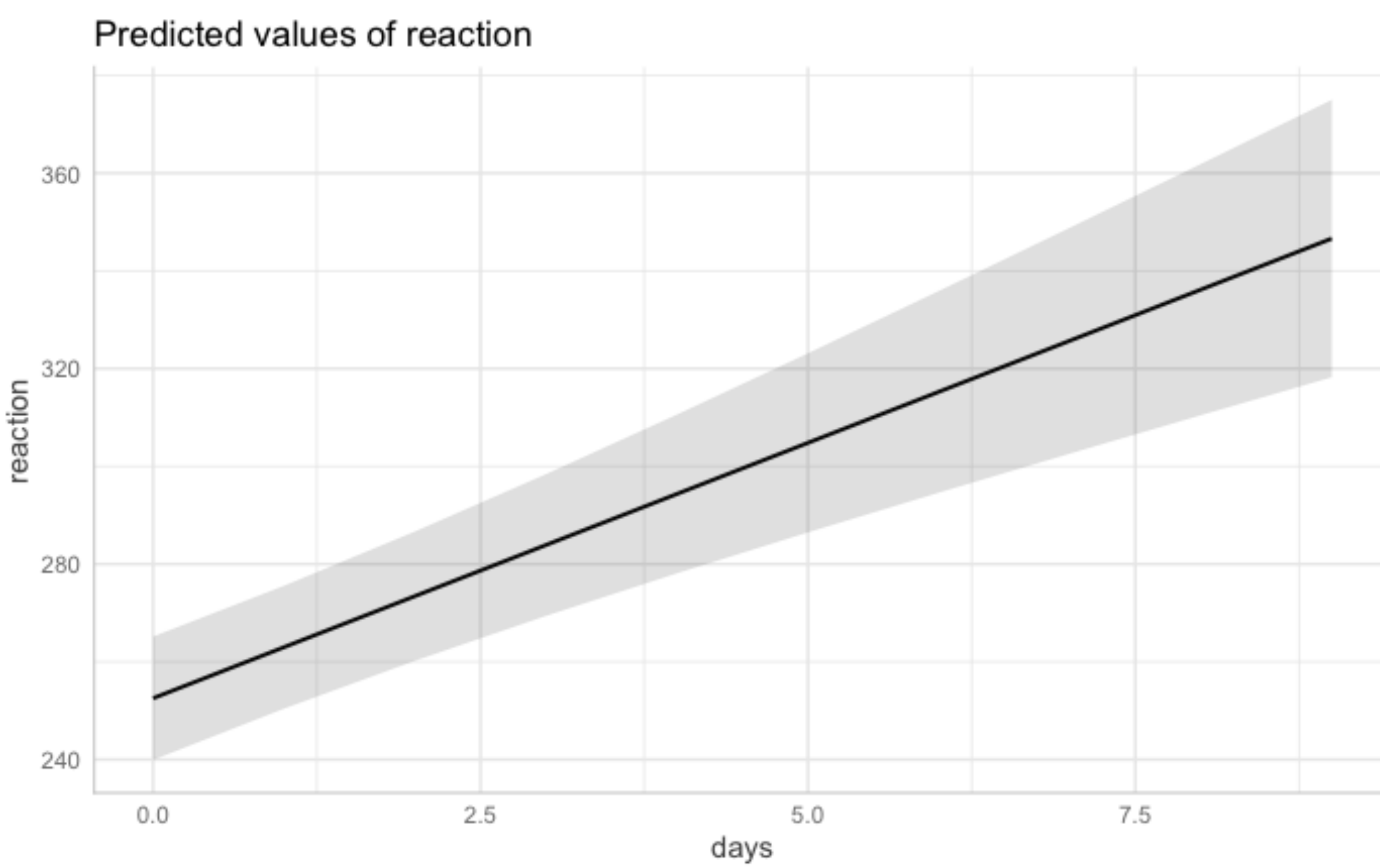
Visualization



show the data together with the model predictions

Visualization

```r
library("ggeffects")

ggpredict(model = fit.random_intercept_slope,
          terms = "days",
          type = "fe") %>%
  plot()
```



- the relationship between the variables of interest (marginalizing over other variables)

show the (marginalized) model prediction

Reporting results

The following is a reproduced excerpt:

7.1. In Writing

Our reports include a description of the following parts (also see Meteyard & Davies, 2019; Barr et al., 2013):

- Model specification, including:
  - Dependent variable, and all fixed and random effects (intercepts, slopes, correlations), both in words and possibly also by providing the model equation/ R-pseudo code (so-called Wilkinson notation)
  - Transformation of variables, e.g., standardizing or centering variables
  - Contrast coding (typically sum-to-zero coding)
- Inference:
  - Description of how *p*-values were obtained (in case of a frequentist approach) or what other (Bayesian) decision rule was used for inference.
  - Description of what post-hoc or follow-up tests were performed
  - Any convergence issues that may arise while running the model (in particular if they require adjustments in the model specification) and how they were dealt with should be described, as well as the subsequent adjustments that were made.
- Model output, at minimum the following:
  - Model results: (un)standardized regression coefficients, standard errors and/or confidence / credible intervals, test statistics, degrees of freedom, *p*-values

Meteyard, L. & Davies, R. A. I. (2020). Best practice guidance for linear mixed-effects models in psychological science. *Journal of Memory and Language*.

Reporting results

Table 2

Posterior means and 95% highest density intervals for each fixed effect in the Bayesian mixed-effects regression model. I fitted the model separately for participants' hypothetical and counterfactual judgments. The results shows that participants' hypothetical judgments are most strongly influenced by the initial position of the block, and that counterfactual judgments are mostly strongly influenced by the final position of the block. Note*: I used sum contrasts for the predictor variables with no/yes for the block variables, and miss/hit for the outcome variable.*

model specification: `judgment ~ 1 + block_initial + block_final + outcome + (1 | participant)`

name	intercept	block initial	block final	outcome
hypothetical	49.26 [44.95, 53.61]	9.67 [6.64, 12.66]	1.84 [−1.08, 4.94]	−2.08 [−5.3, 1.03]
counterfactual	51.2 [48.02, 54.46]	3.05 [0.48, 5.75]	28.51 [25.91, 31.11]	0.59 [−2.03, 3.29]

Understanding `lmer()` syntax

`lmer()` syntax summary

formula	description
`dv ~ x1 + (1 | g)`	Random **intercept** for each level of \`g\`
`dv ~ x1 + (0 + x1 | g)`	Random **slope** for each level of \`g\`
`dv ~ x1 + (x1 | g)`	**Correlated** random slope and intercept for each level of \`g\`
`dv ~ x1 + (x1 || g)`	**Uncorrelated** random slope and intercept for each level of \`g\`
`dv ~ x1 + (1 | part) + (1 | item)`	Random intercept for each level of \`participant\` and for each level of \`item\` (**crossed**)
`dv ~ x1 + (1 | school/class)`	Random intercept for each level of \`school\` and for each level of \`class\` in \`school\` (**nested**)

Coefficients

```
lmer(formula = reaction ~ 1 + days + ...,
     data = df.sleep)
```

`(1 | subject)` `(0 + days | subject)` `(1 + days | subject)`

random intercepts

```
$subject
    (Intercept)     days
308    292.2749 10.43191
309    174.0559 10.43191
310    188.7454 10.43191
330    256.0247 10.43191
331    261.8141 10.43191
332    259.8262 10.43191
333    268.0765 10.43191
334    248.6471 10.43191
335    206.5096 10.43191
337    323.5643 10.43191
349    230.5114 10.43191
350    265.6957 10.43191
351    243.7988 10.43191
352    287.8850 10.43191
369    258.6454 10.43191
370    245.2931 10.43191
371    248.3508 10.43191
372    269.6861 10.43191
373    248.2086 10.43191
374    273.9400 10.43191
```

random slopes

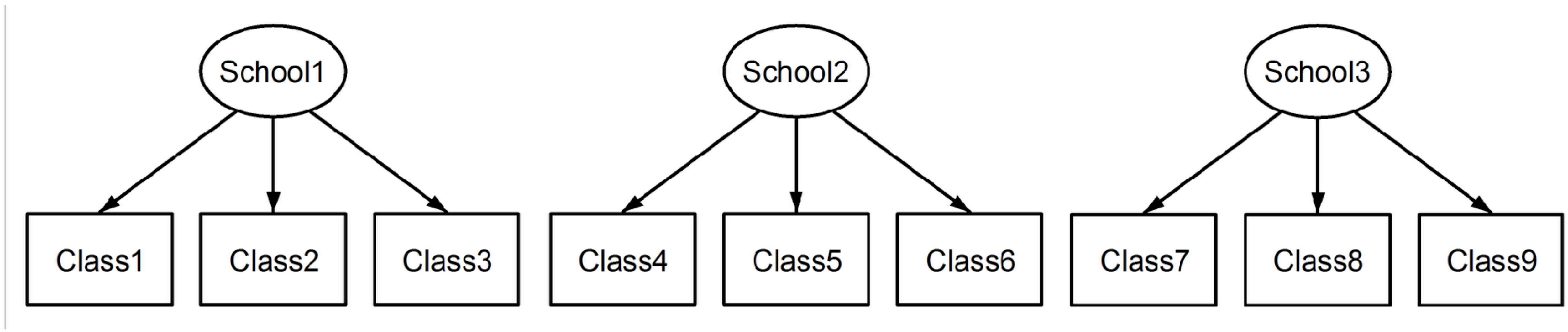
```
$subject
    (Intercept)       days
308    252.2965 19.9526801
309    252.2965 -4.3719650
310    252.2965 -0.9574726
330    252.2965  8.9909957
331    252.2965 10.5394285
332    252.2965 11.3994289
333    252.2965 12.6074020
334    252.2965 10.3413879
335    252.2965 -0.5722073
337    252.2965 24.2246485
349    252.2965  7.7702676
350    252.2965 15.0661415
351    252.2965  7.9675415
352    252.2965 17.0002999
369    252.2965 11.6982767
370    252.2965 11.3939807
371    252.2965  9.4535879
372    252.2965 13.4569059
373    252.2965 10.4142695
374    252.2965 11.9097917
```

random intercepts and slopes (+ correlation)

```
$subject
    (Intercept)       days
308    253.9479 19.6264139
309    211.7328  1.7319567
310    213.1579  4.9061843
330    275.1425  5.6435987
331    273.7286  7.3862680
332    260.6504 10.1632535
333    268.3684 10.2245979
334    244.5523 11.4837825
335    251.3700 -0.3355554
337    286.2321 19.1090061
349    226.7662 11.5531963
350    238.7807 17.0156766
351    256.2344  7.4119501
352    272.3512 13.9920698
369    254.9484 11.2985741
370    226.3701 15.2027922
371    252.5051  9.4335432
372    263.8916 11.7253342
373    248.9752 10.3915245
374    271.1451 11.0782697
```

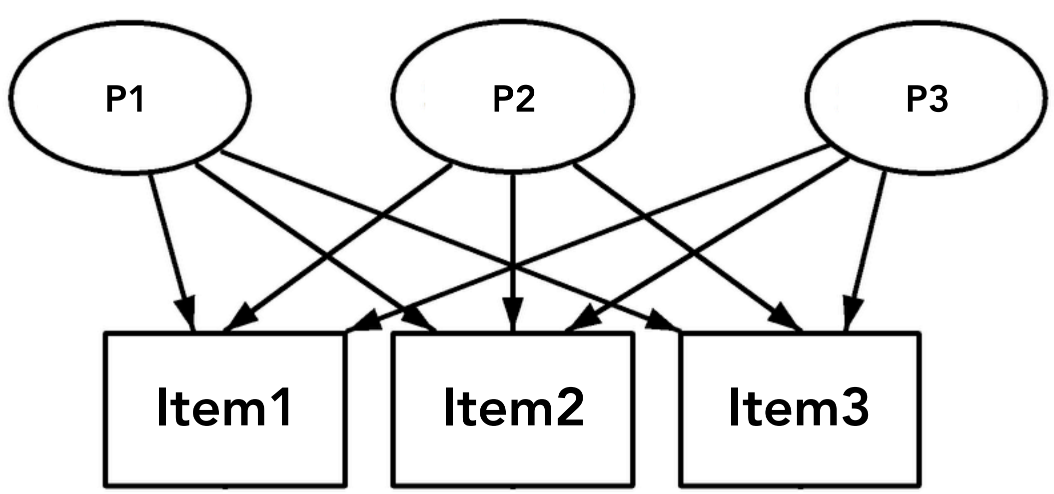
Multi-level models

nested `(1|School/Class)`



each class only appears within one school

crossed `(1|participant) + (1|item)`

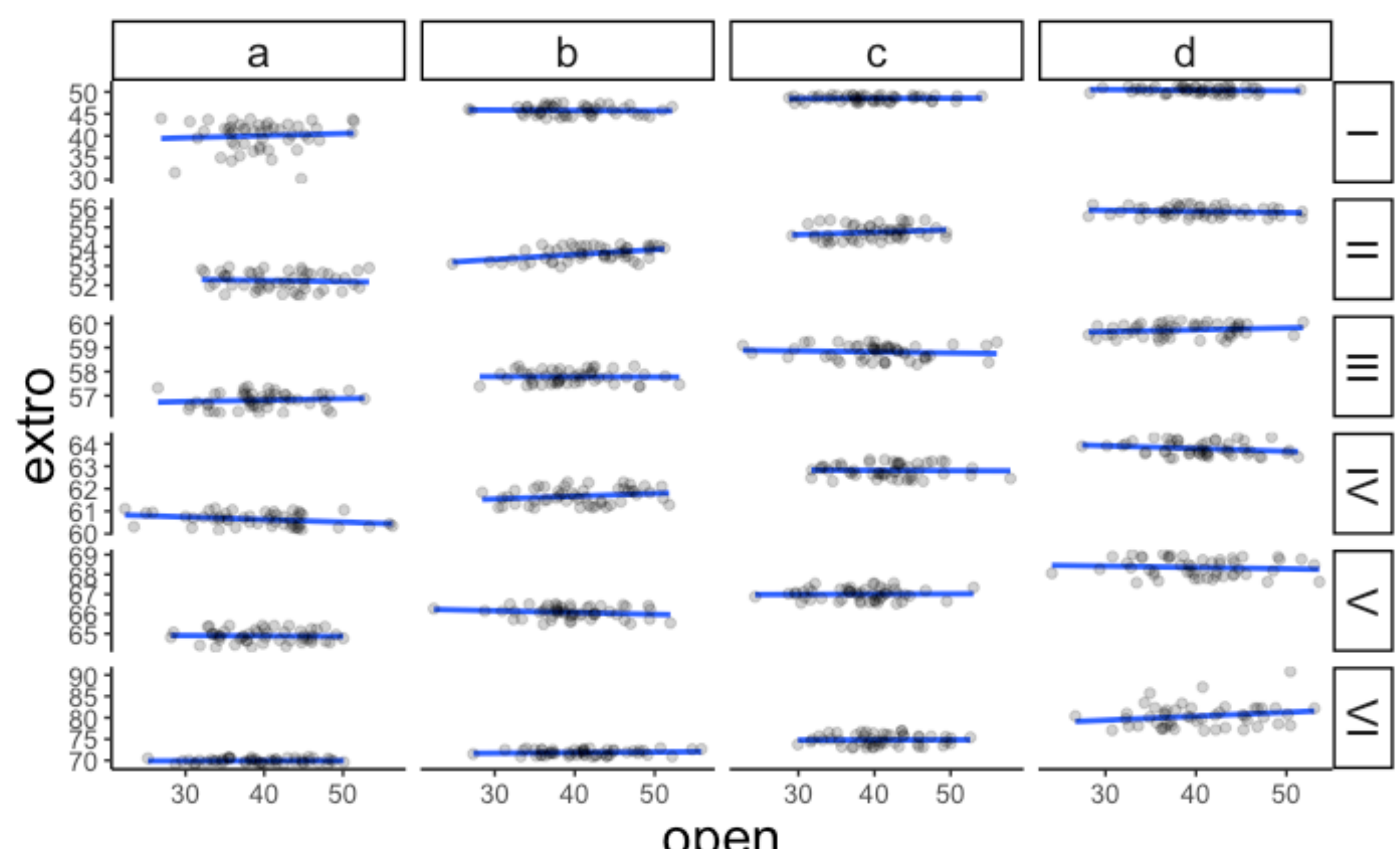


each participant rates each item

Multi-level models

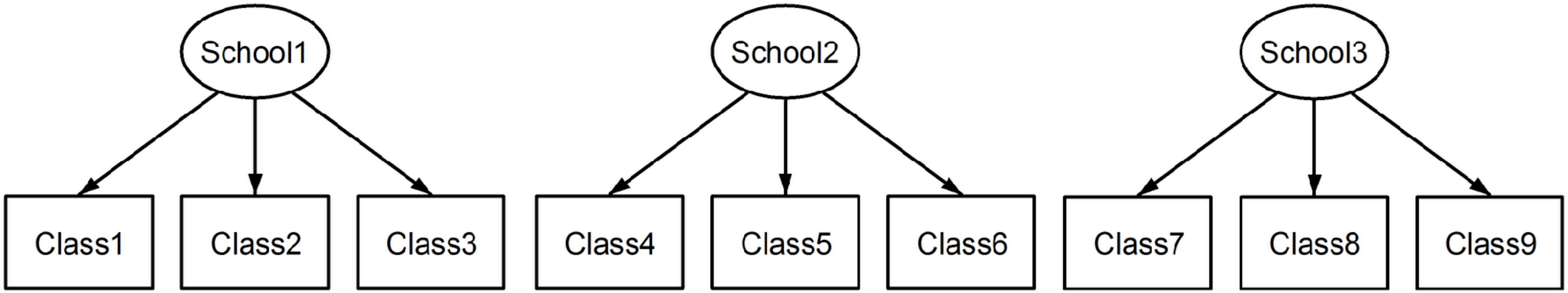
	id	extro	open	agree	social	class	school
1	1	63.69356	43.43306	38.02668	75.05811	d	IV
2	2	69.48244	46.86979	31.48957	98.12560	a	VI
3	3	79.74006	32.27013	40.20866	116.33897	d	VI
4	4	62.96674	44.40790	30.50866	90.46888	c	IV
5	5	64.24582	36.86337	37.43949	98.51873	d	IV
6	6	50.97107	46.25627	38.83196	75.21992	d	I
7	7	60.14740	37.04243	38.55959	95.91299	d	III
8	8	64.17886	42.16530	34.88235	91.45257	d	IV
9	9	56.67670	32.84933	31.68027	115.25167	a	III
10	10	47.23914	44.25764	24.99970	122.70848	b	I

relationship between openness and extraversion



Multi-level models

nested `(1|School/Class)`



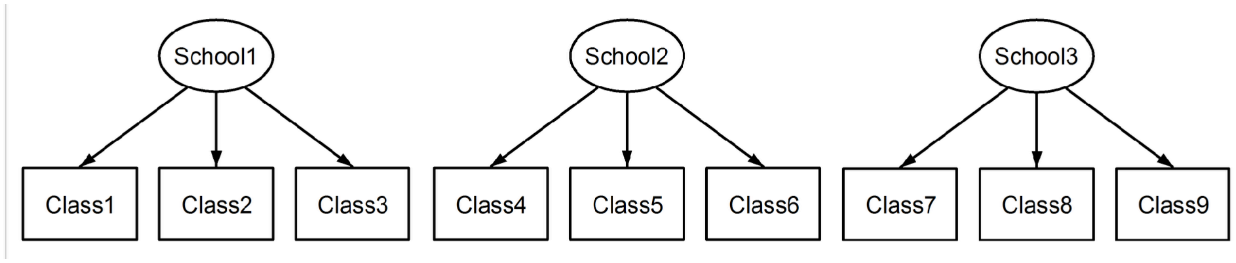

```r
# fit nested model
fit.nested = lmer(extro ~ open + agree + social + (1 | school/class), data = df.school)

# print model summary
fit.nested %>% summary()

# model coefficients
fit.nested %>% coef()
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: extro ~ open + agree + social + (1 | school/class)
   Data: df.school

REML criterion at convergence: 3554.6

Scaled residuals:
    Min      1Q  Median      3Q     Max
-9.9949 -0.3348  0.0057  0.3394 10.6476

Random effects:
 Groups       Name        Variance Std.Dev.
 class:school (Intercept)  8.2046  2.8644
 school       (Intercept) 93.8433  9.6873
 Residual                  0.9684  0.9841
Number of obs: 1200, groups:  class:school, 24; school, 6
```

Multi-level models

random intercepts

nested `(1|School/Class)`



random intercepts of class within each school

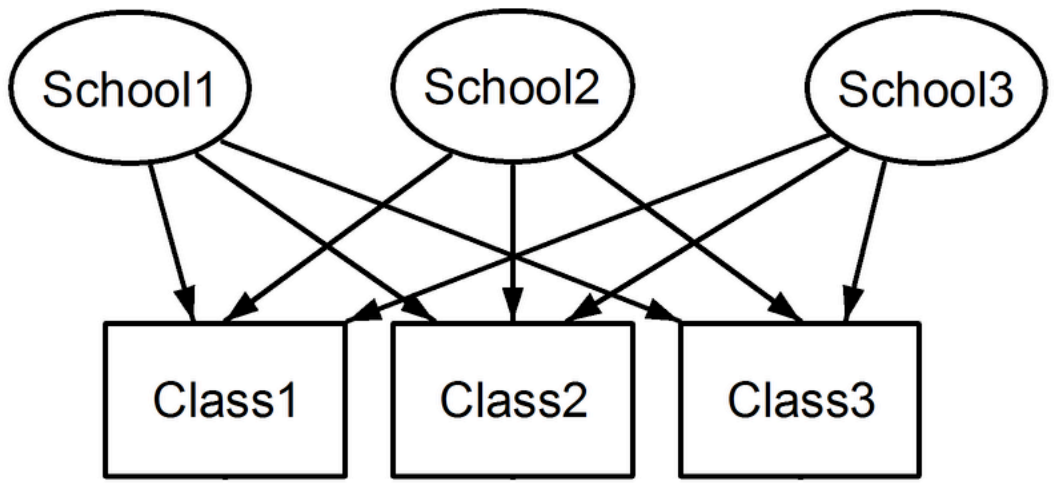
random intercepts of schools

```
$`class:school`
       (Intercept)        open        agree       social
a:I       53.77106 0.006106514 -0.007665927 0.0005404069
a:II      58.23842 0.006106514 -0.007665927 0.0005404069
a:III     58.71819 0.006106514 -0.007665927 0.0005404069
a:IV      58.68813 0.006106514 -0.007665927 0.0005404069
a:V       58.68268 0.006106514 -0.007665927 0.0005404069
a:VI      56.23088 0.006106514 -0.007665927 0.0005404069
b:I       59.54852 0.006106514 -0.007665927 0.0005404069
b:II      59.62643 0.006106514 -0.007665927 0.0005404069
b:III     59.70219 0.006106514 -0.007665927 0.0005404069
b:IV      59.73276 0.006106514 -0.007665927 0.0005404069
b:V       59.87036 0.006106514 -0.007665927 0.0005404069
b:VI      58.14865 0.006106514 -0.007665927 0.0005404069
c:I       62.28460 0.006106514 -0.007665927 0.0005404069
c:II      60.74743 0.006106514 -0.007665927 0.0005404069
c:III     60.70970 0.006106514 -0.007665927 0.0005404069
c:IV      60.86062 0.006106514 -0.007665927 0.0005404069
c:V       60.80225 0.006106514 -0.007665927 0.0005404069
c:VI      61.10164 0.006106514 -0.007665927 0.0005404069
d:I       64.14113 0.006106514 -0.007665927 0.0005404069
d:II      61.81189 0.006106514 -0.007665927 0.0005404069
d:III     61.65165 0.006106514 -0.007665927 0.0005404069
d:IV      61.83703 0.006106514 -0.007665927 0.0005404069
d:V       62.13593 0.006106514 -0.007665927 0.0005404069
d:VI      66.66561 0.006106514 -0.007665927 0.0005404069

$school
    (Intercept)        open        agree       social
I      46.44407 0.006106514 -0.007665927 0.0005404069
II     54.20862 0.006106514 -0.007665927 0.0005404069
III    58.29847 0.006106514 -0.007665927 0.0005404069
IV     62.15074 0.006106514 -0.007665927 0.0005404069
V      66.41348 0.006106514 -0.007665927 0.0005404069
VI     73.91156 0.006106514 -0.007665927 0.0005404069
```

```
# model coefficients
fit.nested %>% coef()
```

Multi-level models

crossed (1|School) + (1|Class)



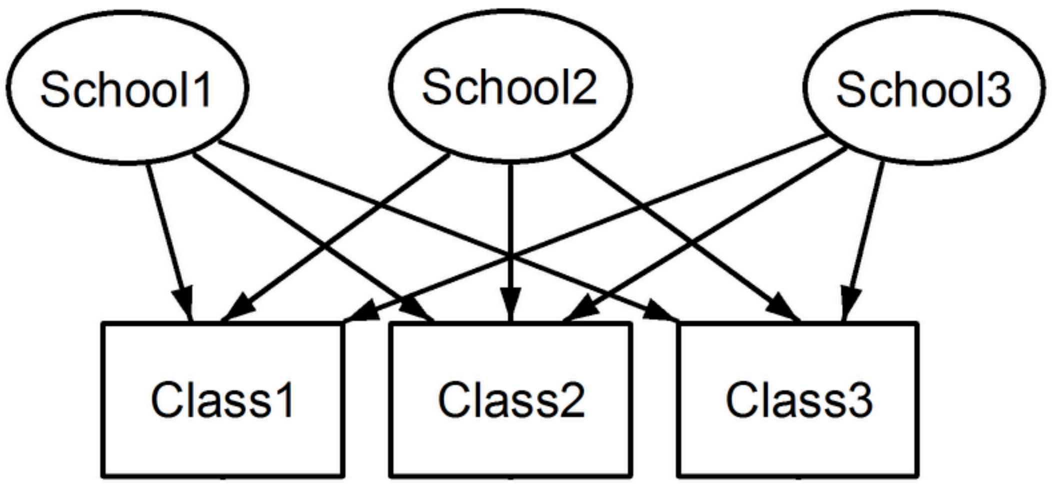
each class appears in each of the schools

```r
# fit crossed model
fit.crossed = lmer(extro ~ open + agree + social + (1 | school) + (1 | class), data = df.school)

# print model summary
fit.crossed %>% summary()
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: extro ~ open + agree + social + (1 | school) + (1 | class)
   Data: df.school

REML criterion at convergence: 4723.9

Scaled residuals:
    Min      1Q  Median      3Q     Max
-7.8677 -0.5421  0.0101  0.5218  8.2282

Random effects:
 Groups   Name        Variance Std.Dev.
 school   (Intercept) 95.914   9.794
 class    (Intercept)  5.787   2.406
 Residual              2.787   1.669
Number of obs: 1200, groups:  school, 6; class, 4
```

Multi-level models

crossed `(1|School) + (1|Class)`



each class is in
each of the schools

random intercepts

random intercepts of school

random intercepts of class

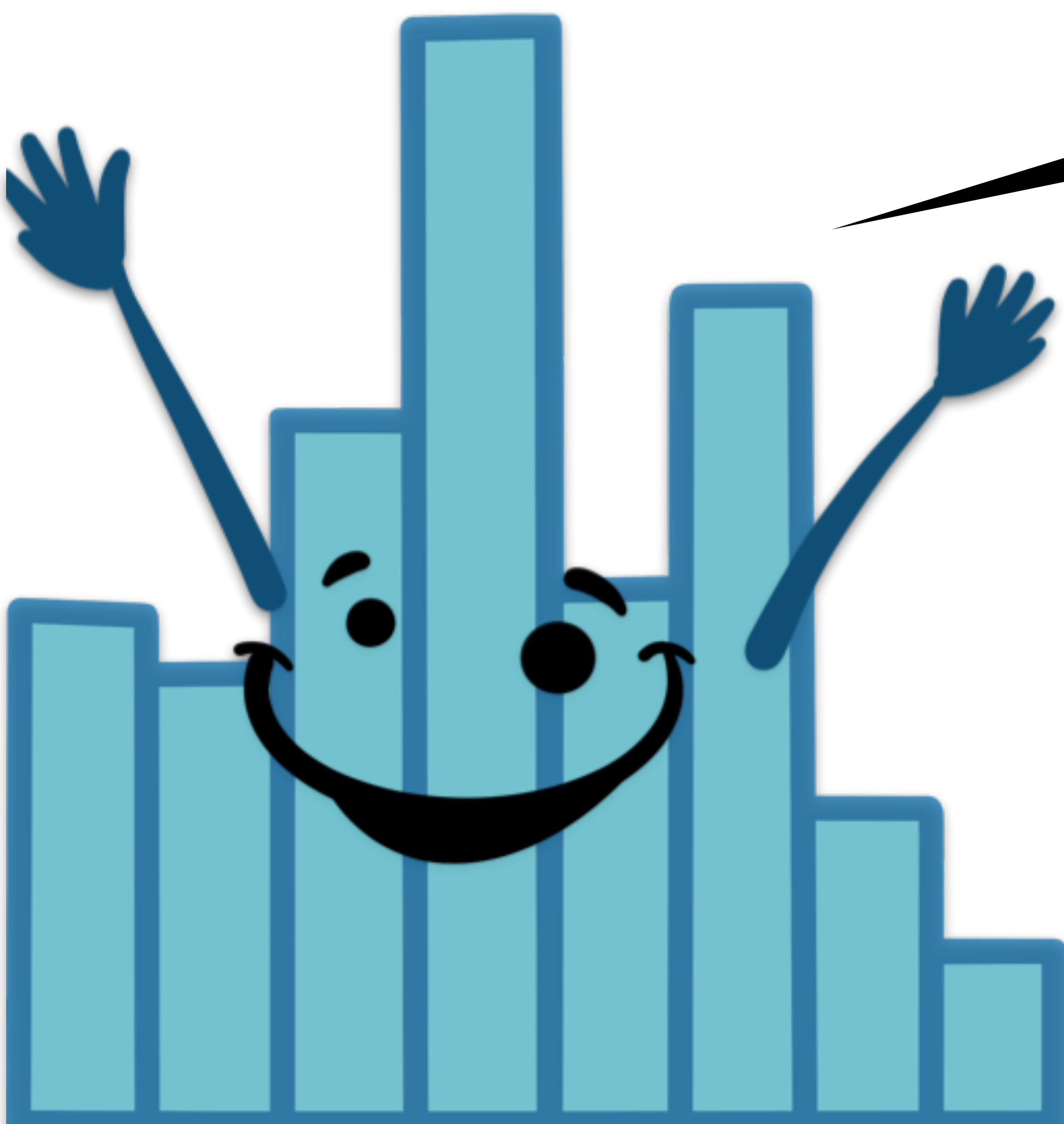
```
$school
    (Intercept)       open        agree       social
I      46.10663 0.01083374 -0.005420032 -0.001761963
II     54.02956 0.01083374 -0.005420032 -0.001761963
III    58.22277 0.01083374 -0.005420032 -0.001761963
IV     62.15508 0.01083374 -0.005420032 -0.001761963
V      66.51062 0.01083374 -0.005420032 -0.001761963
VI     74.16838 0.01083374 -0.005420032 -0.001761963

$class
  (Intercept)       open        agree       social
a    57.35175 0.01083374 -0.005420032 -0.001761963
b    59.39261 0.01083374 -0.005420032 -0.001761963
c    61.04758 0.01083374 -0.005420032 -0.001761963
d    63.00342 0.01083374 -0.005420032 -0.001761963
```

```
# model coefficients
fit.crossed %>% coef()
```

We're listening to
"Tcheren Deya" by
"Mathias Duplessy"
submitted by Tobi

02:00

stretch break!



Let's simulate some `lmer()`s

Let's simulate an `lmer()`

```r
# make example reproducible
set.seed(1)

# parameters
sample_size = 100
b0 = 1
b1 = 2
sd_residual = 1
sd_participant = 0.5

# generate the data
df.mixed = tibble(participant = rep(1:sample_size, 2),
                  condition = rep(0:1, each = sample_size)) %>%
  group_by(participant) %>%
  mutate(intercepts = rnorm(n = 1, sd = sd_participant)) %>%
  ungroup() %>%
  mutate(value = b0 + b1 * condition + intercepts + rnorm(n(), sd = sd_residual)) %>%
  arrange(participant, condition)
```

participant	condition	intercepts	value
1	0	-0.31	0.07
1	1	-0.31	3.10
2	0	0.09	1.13
2	1	0.09	4.78
3	0	-0.42	-0.33
3	1	-0.42	4.17
4	0	0.80	1.96
4	1	0.80	3.47
5	0	0.16	0.51
5	1	0.16	0.88

$$\text{value}_{ij} = b_0 + b_1 \cdot \text{condition}_{ij} + U_i + e_{ij}$$

$$e_{ij} \sim \mathcal{N}(\text{mean} = 0, \text{sd} = s_{\text{error}})$$

$$U_i \sim \mathcal{N}(\text{mean} = 0, \text{sd} = s_{\text{U}})$$

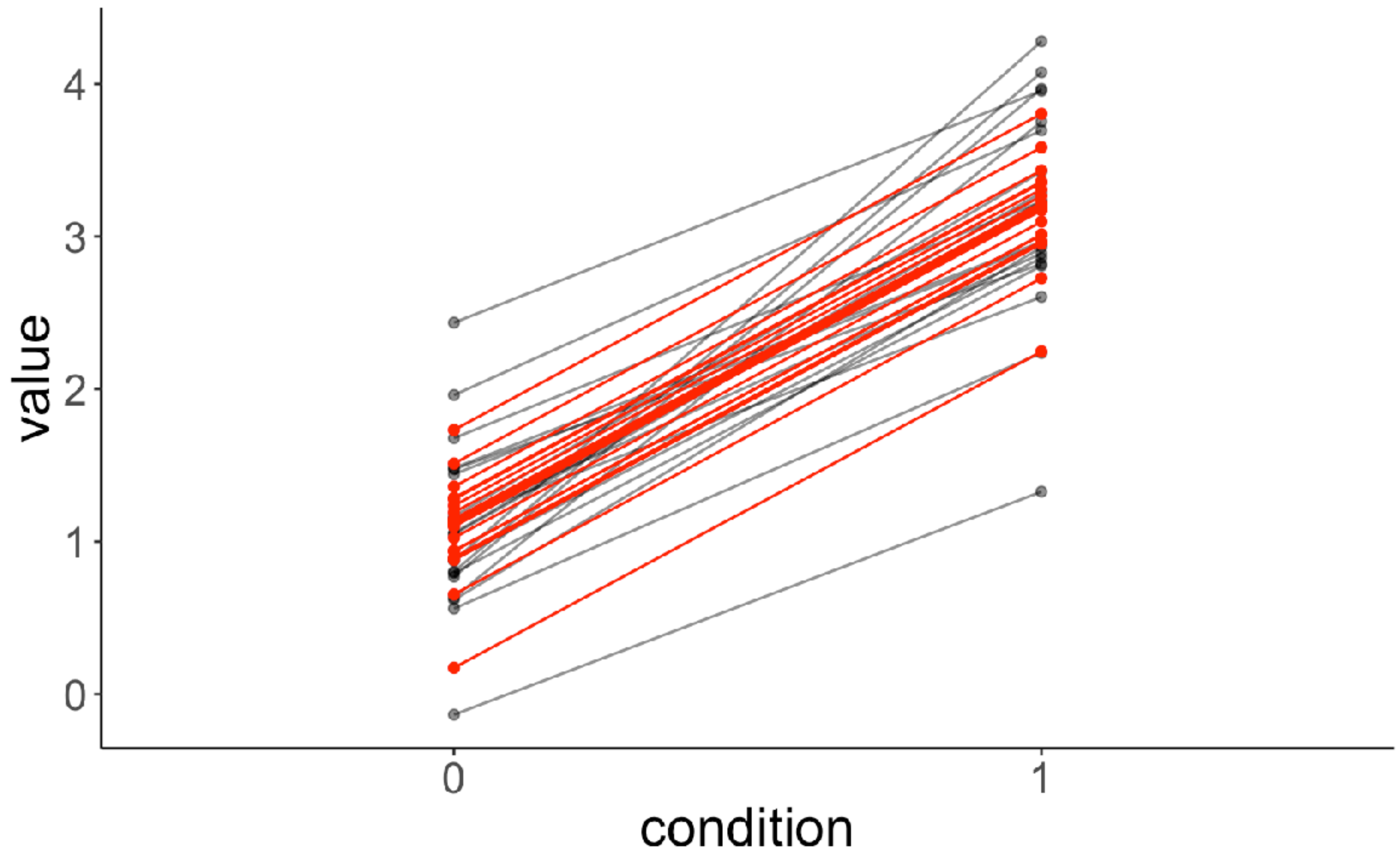
simulating data from a model and trying to recover the parameters is a great way to check one's understanding of what the model does

Let's simulate an **`lmer()`**

```
Linear mixed model fit by REML ['lmerMod']
Formula: value ~ 1 + condition + (1 | participant)
   Data: df.mixed

REML criterion at convergence: 606

Scaled residuals: 
     Min       1Q   Median       3Q      Max 
-2.53710 -0.62295 -0.04364  0.67035  2.19899 

Random effects:
 Groups      Name        Variance Std.Dev.
 participant (Intercept) 0.1607   0.4009  
 Residual                1.0427   1.0211  
Number of obs: 200, groups:  participant, 100

Fixed effects:
            Estimate Std. Error t value
(Intercept)   1.0166     0.1097   9.267
condition     2.0675     0.1444  14.317

Correlation of Fixed Effects:
          (Intr)
condition -0.658
```

```
# make example reproducible
set.seed(1)

# parameters
sample_size = 100
b0 = 1
b1 = 2
sd_residual = 1
sd_participant = 0.5

# generate the data
df.mixed = tibble(participant = rep(1:sample_size, 2),
                  condition = rep(0:1, each = sample_size)) %>% 
  group_by(participant) %>% 
  mutate(intercepts = rnorm(n = 1, sd = sd_participant)) %>% 
  ungroup() %>% 
  mutate(value = b0 + b1 * condition + intercepts + rnorm(n(), sd = sd_residual)) %>% 
  arrange(participant, condition)
```

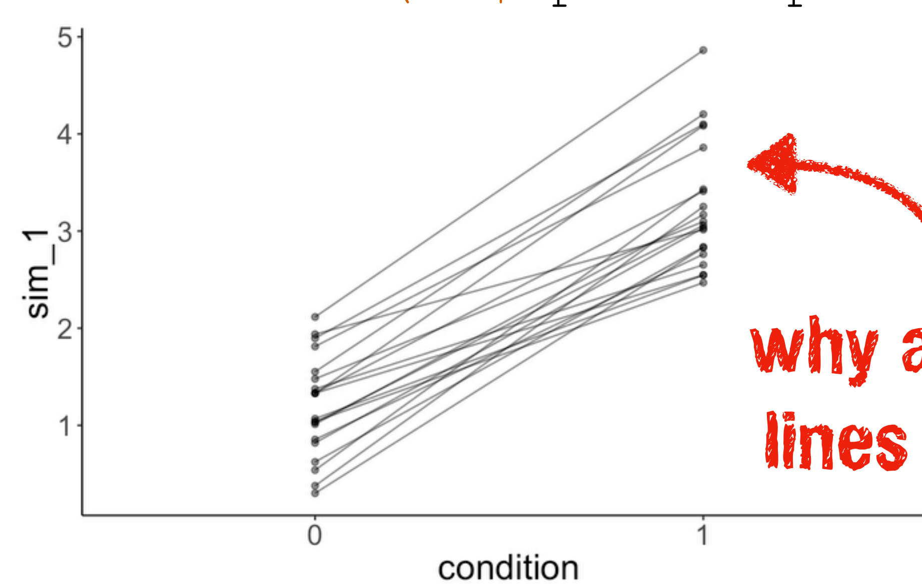
No outlier




```
Linear mixed model fit by REML ['lmerMod']
Formula: value ~ 1 + condition + (1 | participant)
   Data: df.test

REML criterion at convergence: 74.9

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.9268 -0.5412 -0.1103  0.4868  1.7747

Random effects:
 Groups      Name        Variance Std.Dev.
 participant (Intercept) 0.1702   0.4125
 Residual                0.2270   0.4764
Number of obs: 40, groups:  participant, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept)   1.0920     0.1409    7.75
condition1    2.0726     0.1507   13.76

Correlation of Fixed Effects:
           (Intr)
condition1 -0.535
```

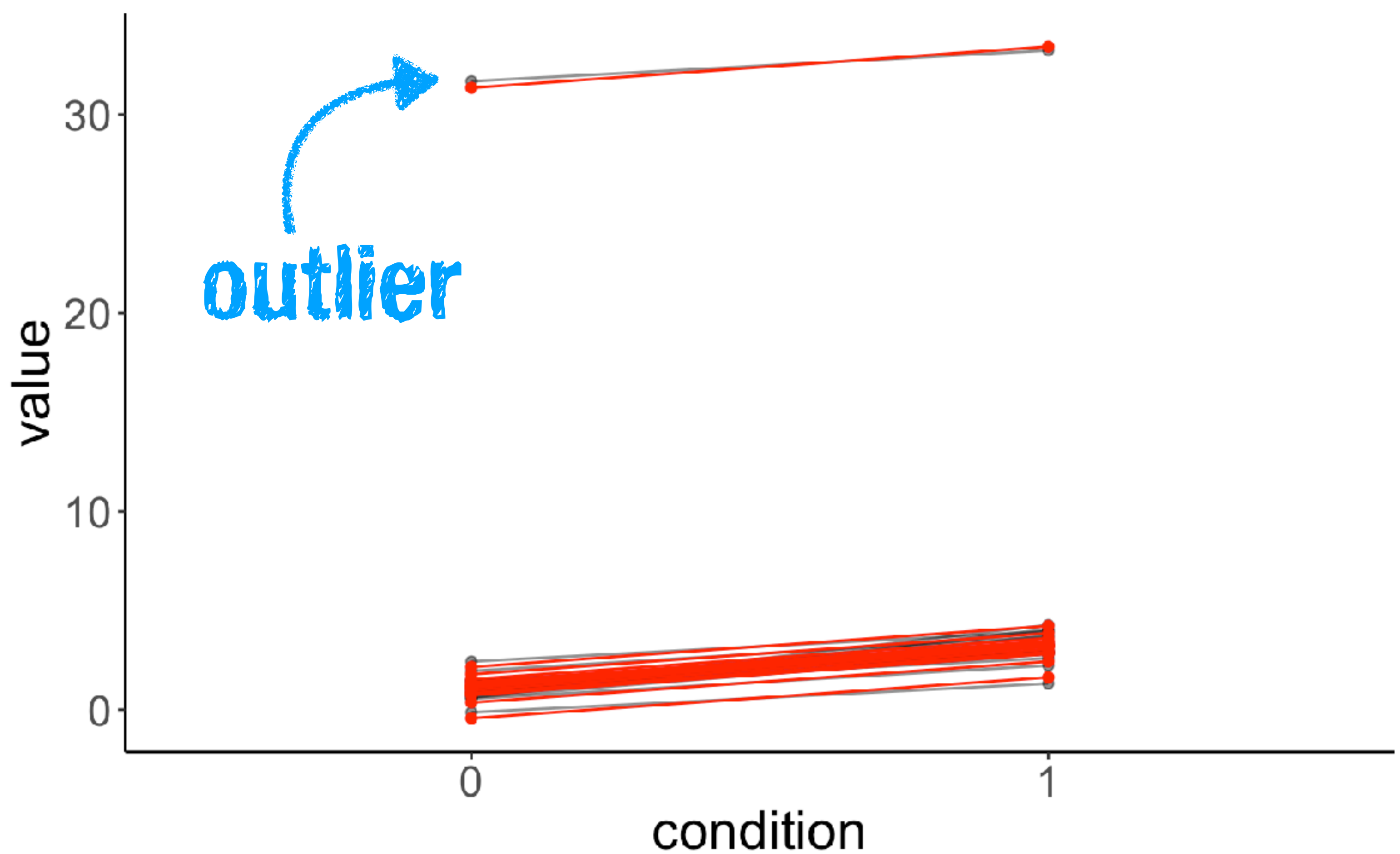
```r
# fit model
fit.test = lmer(formula = value ~ 1 + condition + (1 | participant),
                data = df.test)

# simulate data
fit.test %>%
  simulate()
```

simulated data



why aren't the lines parallel?

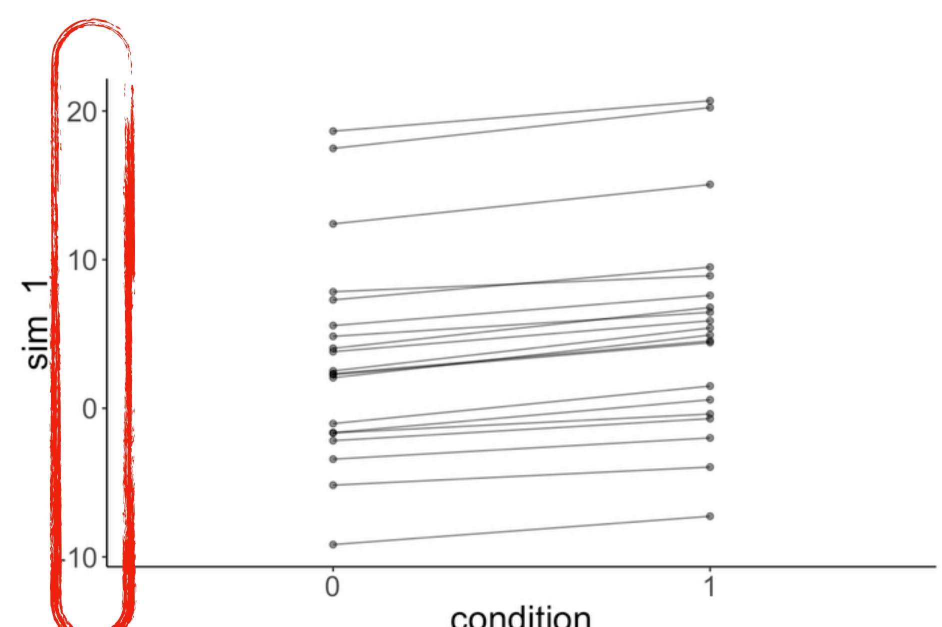
With outlier



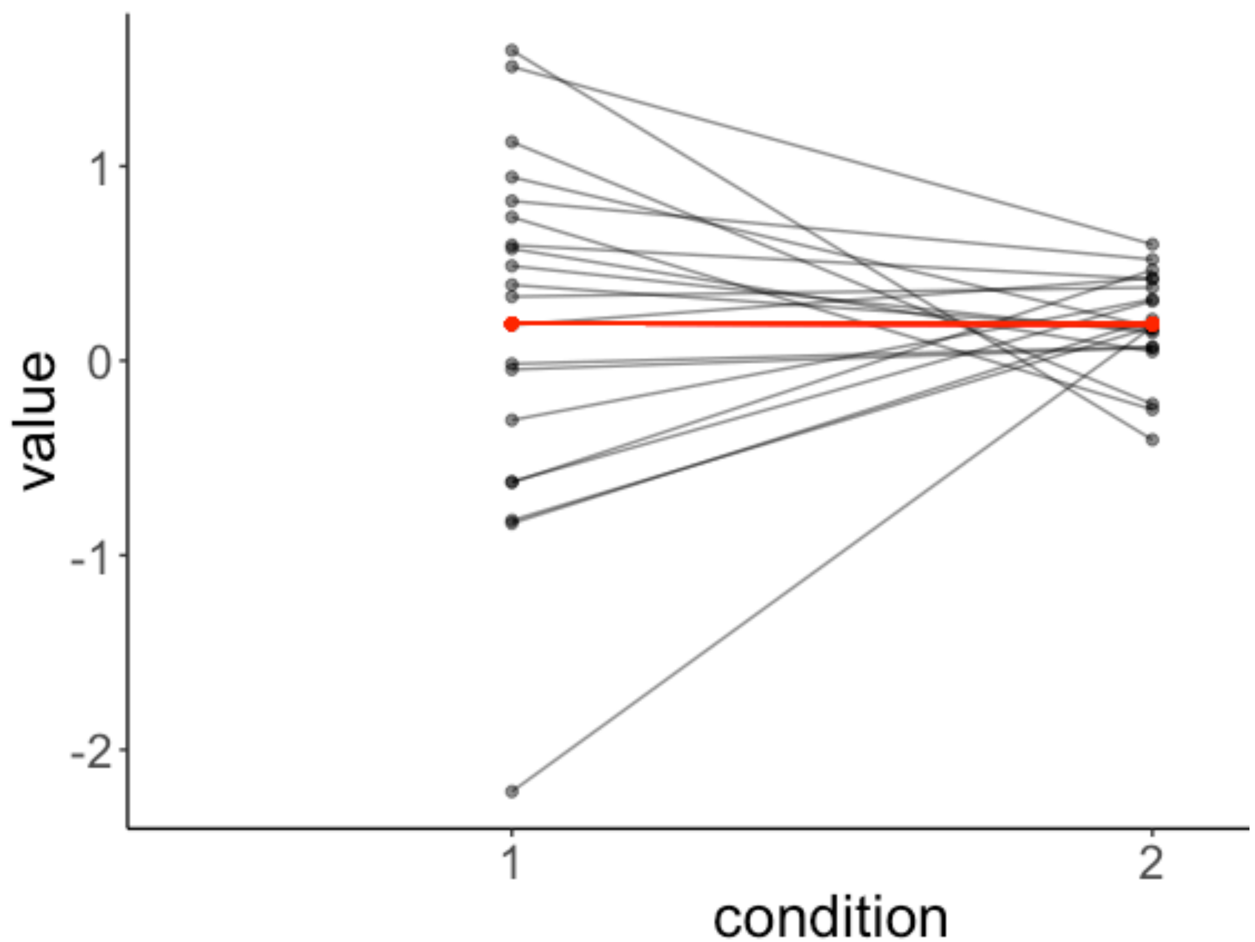

```
Linear mixed model fit by REML ['lmerMod']
Formula: value ~ 1 + condition + (1 | participant)
   Data: df.test

REML criterion at convergence: 171.7

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.4038 -0.4678 -0.0094  0.5800  1.3930

Random effects:
 Groups      Name        Variance Std.Dev.
 participant (Intercept) 46.198   6.7969
 Residual                 0.227   0.4764
Number of obs: 40, groups:  participant, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept)   2.5920     1.5236   1.701
condition1    2.0726     0.1507  13.758

Correlation of Fixed Effects:
           (Intr)
condition1 -0.049
```

```
# fit model
fit.test = lmer(formula = value ~ 1 + condition + (1 | participant),
                data = df.test)

# simulate data
fit.test %>%
  simulate()
```

simulated data



Non-equal variance




```
singular fit
Linear mixed model fit by REML ['lmerMod']
Formula: value ~ 1 + condition + (1 | participant)
   Data: df.test

REML criterion at convergence: 83.6

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.5808 -0.3184  0.0130  0.4551  2.0913

Random effects:
 Groups      Name        Variance Std.Dev.
 participant (Intercept) 0.0000   0.0000
 Residual                0.4512   0.6717
Number of obs: 40, groups:  participant, 20

Fixed effects:
             Estimate Std. Error t value
(Intercept)  0.190524   0.150197   1.268
condition2  -0.001941   0.212411  -0.009

Correlation of Fixed Effects:
           (Intr)
condition2 -0.707
convergence code: 0
singular fit
```

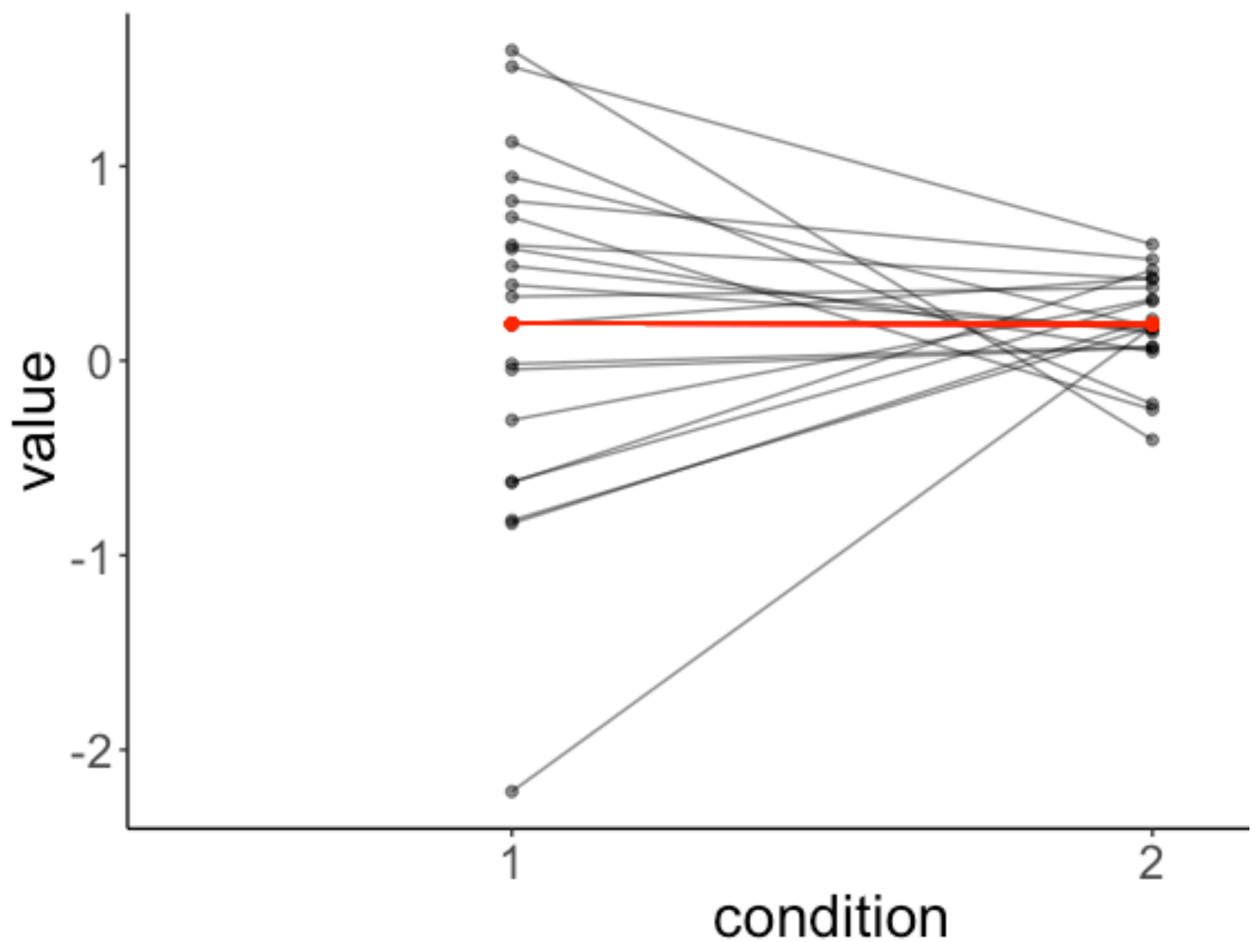
clearly there are interindividual differences though?!

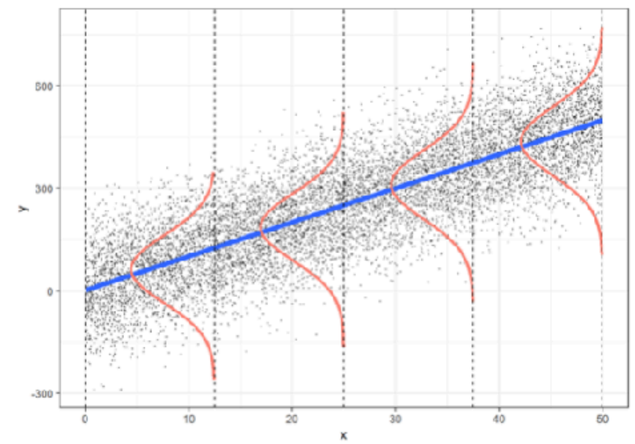
Non-equal variance

Model assumptions of simple regression

- independent observations
- Y is continuous
- errors are normally distributed
- errors have constant variance
- error terms are uncorrelated

assumption violated





the "model" would just reproduce the data

random intercept

```
# fit model
lmer(formula = value ~ 1 + condition + (1 + condition | participant),
     data = df.test)
```

random slope

won't work

```
Error: number of observations (=40) <= number of random effects (=40) for term
(1 + condition | participant); the random-effects parameters and the residual
variance (or scale parameter) are probably unidentifiable
```

`lmer()` standard operating procedures

Standard Operating Procedures For Using Mixed-Effects Models

A Principled Workflow from the Decision, Development, and Psychopathology (D2P2) Lab
document version 1.0.0 -- 28 June 2020
[This document will be continuously updated and expanded; it may contain typos and other errors--both unintentional errors and errors based on incorrect or outdated knowledge--we will try to improve these things in future versions. Feel free to let us know if you spotted such things, how to further improve this document!]

Authors (in alphabetical order except that the youngsters were so kind to put the oldest guy in the lab first; BF)

Bernd Figner, Johannes Algermissen, Floor Burghoorn, Leslie Held, Afreen Khalid, Felix Klaassen, Farnaz Mosannenzadeh, Julian Quandt

Content/Analysis Steps

http://decision-lab.org/wp-content/uploads/2020/07/SOP_Mixed_Models_D2P2_v1_0_0.pdf

What shall I include as random effects?

- mixed opinions on the topic
- go maximal!

Journal of Memory and Language 68 (2013) 255–278

Contents lists available at SciVerse ScienceDirect

Journal of Memory and Language

journal homepage: www.elsevier.com/locate/jml

Random effects structure for confirmatory hypothesis testing: Keep it maximal

Dale J. Barr [a,*], Roger Levy [b], Christoph Scheepers [a], Harry J. Tily [c]

[a] Institute of Neuroscience and Psychology, University of Glasgow, 58 Hillhead St., Glasgow G12 8QB, United Kingdom
[b] Department of Linguistics, University of California at San Diego, La Jolla, CA 92093-0108, USA
[c] Department of Brain and Cognitive Sciences, Massachussetts Institute of Technology, Cambridge, MA 02139, USA

"Through theoretical arguments and Monte Carlo simulation, we show that LMEMs generalize best when they include the maximal random effects structure justified by the design. ...

Maximal LMEMs should be the 'gold standard' for confirmatory hypothesis testing in psycholinguistics and beyond."

Barr, D. J., Levy, R., Scheepers, C., & Tily, H. J. (2013). Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory and Language*, 68(3), 255--278.

What shall I include as random effects?

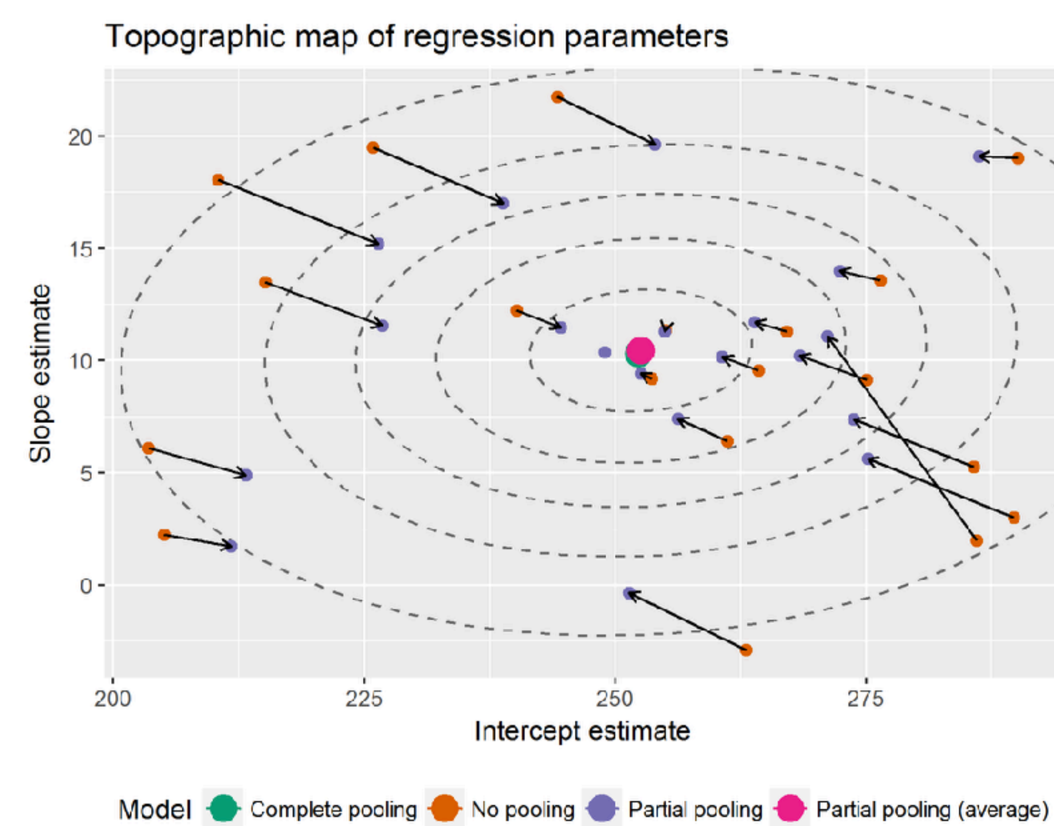
- general advice:
  - start maximal (as supported by the design)
    - random intercepts for different participants
    - random slopes when participants are tested multiple times
    - random intercepts for items
  - reduce complexity of the random effects structure step by step
    - remove the correlation between random effects first

Remove the correlation component from your model

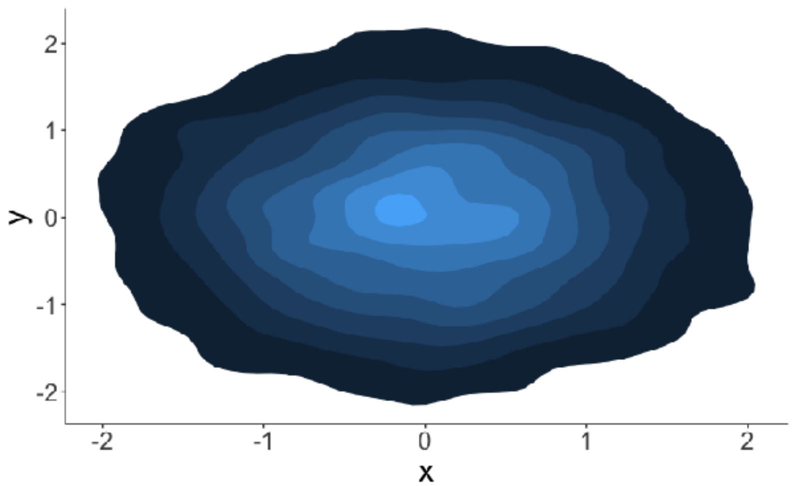
```r
# fit the model
fit.lmer = lmer(formula = reaction ~ 1 + days + (1 + days | subject),
                data = df.sleep)
# model summary
fit.lmer %>%
  summary()
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: reaction ~ 1 + days + (1 + days | subject)
   Data: df.sleep

REML criterion at convergence: 1771.4

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.9707 -0.4703  0.0276  0.4594  5.2009

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 subject  (Intercept) 582.73   24.140
          days         35.03    5.919   0.07
 Residual             649.36   25.483
Number of obs: 183, groups:  subject, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept)  252.543      6.433  39.256
days          10.452      1.542   6.778

Correlation of Fixed Effects:
     (Intr)
days -0.137
```

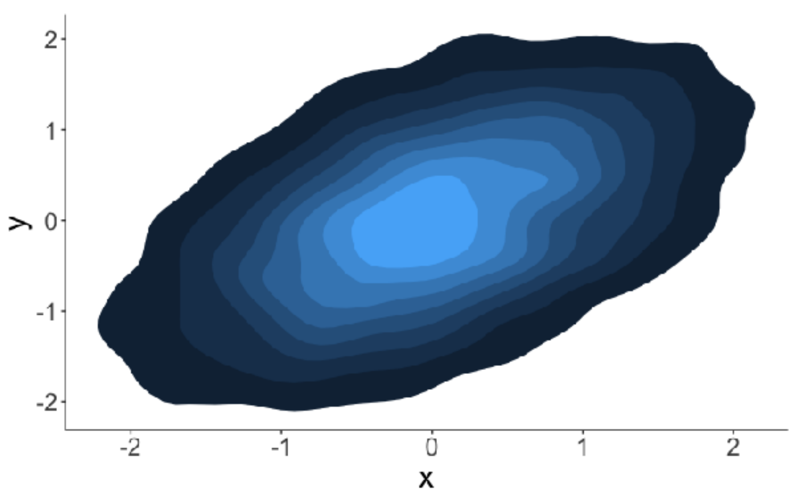
multivariate Gaussian



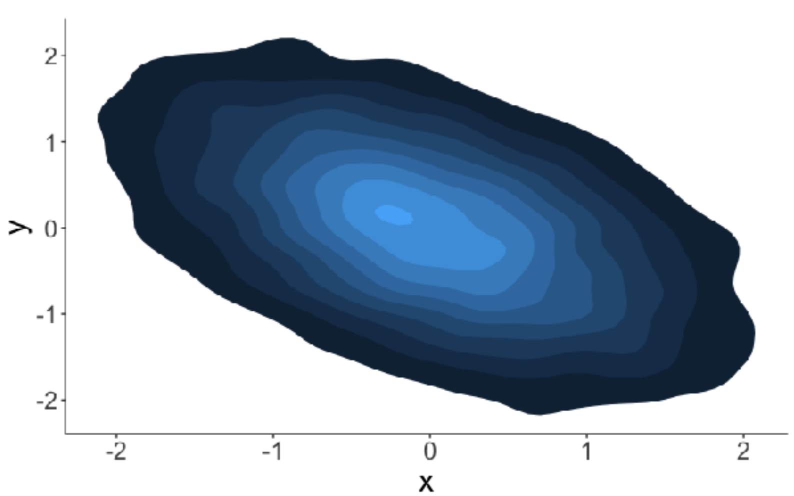
Remove the correlation component from your model



uncorrelated



positively correlated



negatively correlated

Remove the correlation component from your model

```r
# fit the model
fit.lmer = lmer(formula = reaction ~ 1 + days + (0 + days | subject) + (1 | subject),
                data = df.sleep)
# model summary
fit.lmer %>%
  summary()
```

random slopes

random intercepts

```
Linear mixed model fit by REML ['lmerMod']
Formula: reaction ~ 1 + days + (0 + days | subject) + (1 | subject)
   Data: df.sleep

REML criterion at convergence: 1771.5

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.9805 -0.4673  0.0250  0.4589  5.2083

Random effects:
 Groups    Name        Variance Std.Dev.
 subject   days         35.88    5.99
 subject.1 (Intercept) 598.11   24.46
 Residual              647.90   25.45
Number of obs: 183, groups:  subject, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept)  252.550      6.491  38.907
days          10.439      1.556   6.708

Correlation of Fixed Effects:
     (Intr)
days -0.184
```

independent Gaussians

Remove the correlation component from your model

```r
# fit the model
fit.lmer = lmer(formula = reaction ~ 1 + days + (1 + days || subject),
                data = df.sleep)
# model summary
fit.lmer %>%
  summary()
```

alternative syntax (doesn't model correlation between random effects)

```
Linear mixed model fit by REML ['lmerMod']
Formula: reaction ~ 1 + days + (0 + days | subject) + (1 | subject)
   Data: df.sleep

REML criterion at convergence: 1771.5

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.9805 -0.4673  0.0250  0.4589  5.2083

Random effects:
 Groups    Name        Variance Std.Dev.
 subject   days         35.88    5.99
 subject.1 (Intercept) 598.11   24.46
 Residual              647.90   25.45
Number of obs: 183, groups:  subject, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept)  252.550      6.491  38.907
days          10.439      1.556   6.708

Correlation of Fixed Effects:
     (Intr)
days -0.184
```

independent Gaussians

What if `lmer()` fails to converge?

```r
# fit the model
fit.lmer = lmer(formula = reaction ~ 1 + days + (1 + days | subject),
                data = df.sleep)

# explore different optimization algorithms
fit.all = allFit(fit.lmer)

# summarize result
fit.all %>% summary()
```

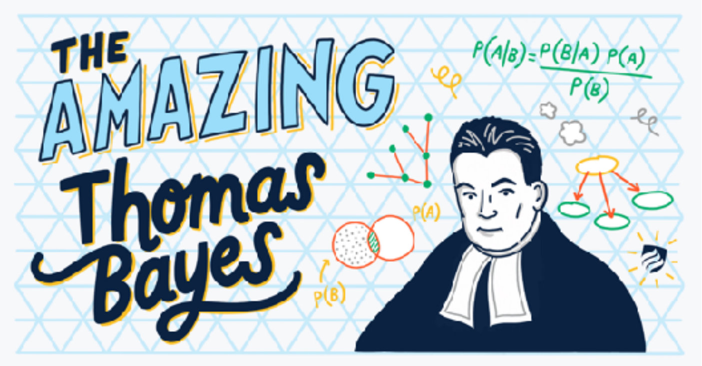
comparison of the different optimization algorithms

```
$fixef
                              (Intercept)     days
bobyqa                           252.5426 10.45212
Nelder_Mead                      252.5426 10.45212
nlminbwrap                       252.5426 10.45212
nloptwrap.NLOPT_LN_NELDERMEAD    252.5426 10.45212
nloptwrap.NLOPT_LN_BOBYQA        252.5426 10.45212

$llik
                       bobyqa                   Nelder_Mead                    nlminbwrap
                    -885.7239                     -885.7239                     -885.7239
nloptwrap.NLOPT_LN_NELDERMEAD     nloptwrap.NLOPT_LN_BOBYQA
                    -885.7239                     -885.7239

$sdcor
                              subject.(Intercept) subject.days.(Intercept) subject.days    sigma
bobyqa                                   24.13911                 5.918866   0.06927657 25.48261
Nelder_Mead                              24.13900                 5.918891   0.06928125 25.48261
nlminbwrap                               24.13911                 5.918867   0.06927628 25.48261
nloptwrap.NLOPT_LN_NELDERMEAD            24.13979                 5.918851   0.06927975 25.48255
nloptwrap.NLOPT_LN_BOBYQA                24.13979                 5.918851   0.06927975 25.48255
```

https://rdrr.io/cran/lme4/man/convergence.html

What if `lmer()` fails to converge?

1. We drop random effects in the following order: random correlations, random slopes of covariates (where significance is of no interest), random intercepts ("0+" instead "1+") (following Barr et al., 2013). We never remove the random slopes of the variables of interest (i.e., the ones for which we want to conduct significance tests).
   Please note that removing random correlation terms can be tricky if random slopes are estimated for factors with 3 or more levels. In that case, it is probably easiest to use `afex::mixed()` with `expand_re = TRUE` (an alternative option is to create manually the relevant contrasts yourself and add them as predictors to your model, which allows you to suppress the random corrections using the double pipe symbol ||).
2. We try to run separate analyses: For example, one model to only test the fixed and random effect of A (with fixed effect of B present); then one model to only test the effect of B.
   If we really have to drop random slopes, we follow the next step:
3. We follow the PCA approach suggested by **rePsychLing** (see Bates et al., 2015) that is performing a PCA on the random effects and following the guidelines described in the paper.
   a. We use a likelihood ratio test to test whether the model fit becomes significantly worse. As we prefer a more conservative approach here (i.e., rather err on the side of keeping too many random effects; we prioritize avoiding inflated Type 2 errors for this kind of decision), we use larger alpha-level of .2 (Matuschek et al., 2017).
   b. Alternatively, we suggest an Information criterion approach to avoid using a *p* value for our inclusion/exclusion decision, but choose the best model based on *BIC* or *AIC*.



3.2.2. Or we choose a Bayesian approach

As an alternative to targeting convergence issues within **lme4**, we suggest fitting the same model with **brms** and comparing it to the **lme4** fit. We assume that both provide similar results when

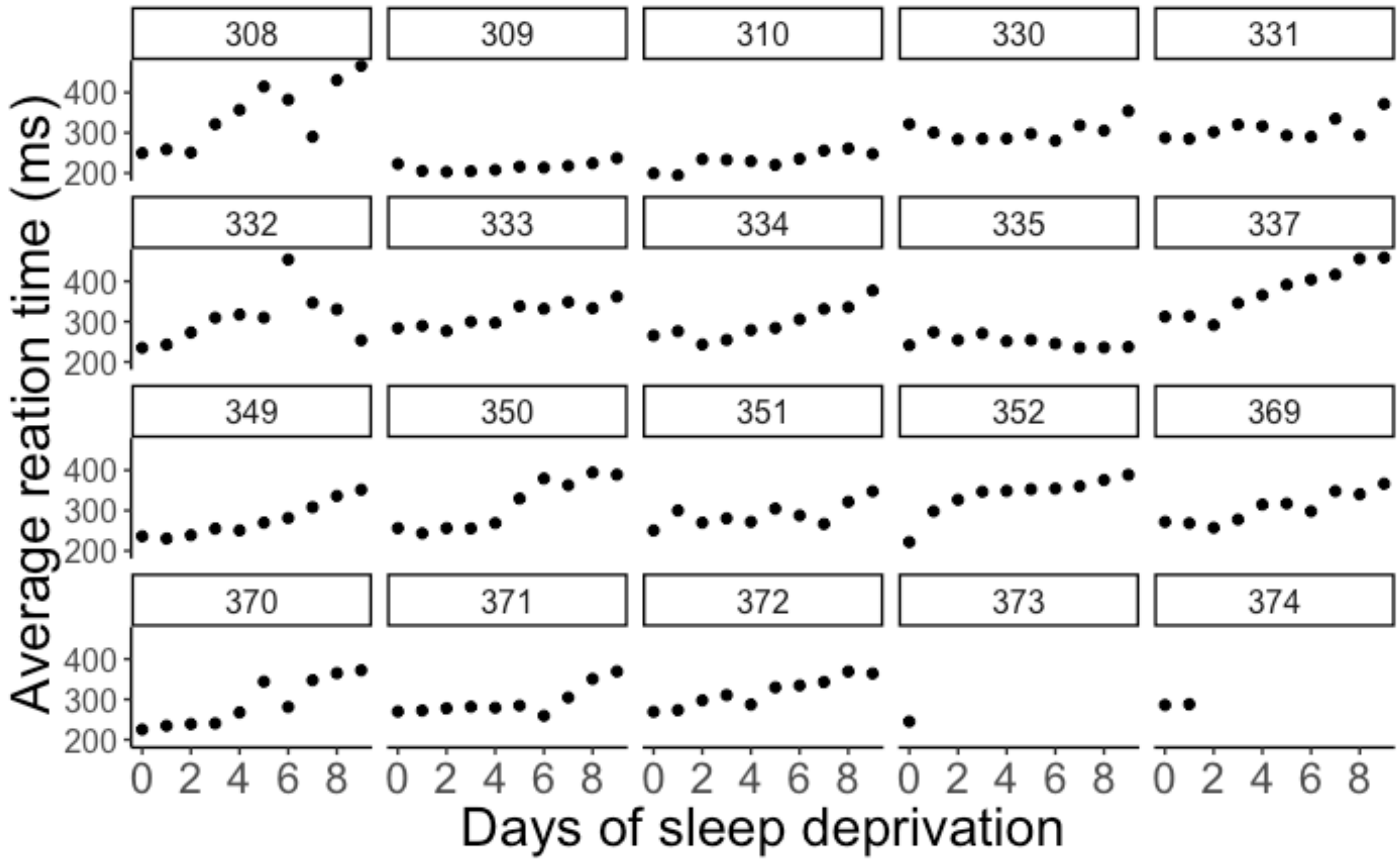
Some more examples

1. Sleep

Sleep data

How does sleep deprivation affect reaction time?

subject	days	reaction
308	0	249.56
308	1	258.70
308	2	250.80
308	3	321.44
308	4	356.85
309	0	222.73
309	1	205.27
309	2	202.98
309	3	204.71
309	4	207.72



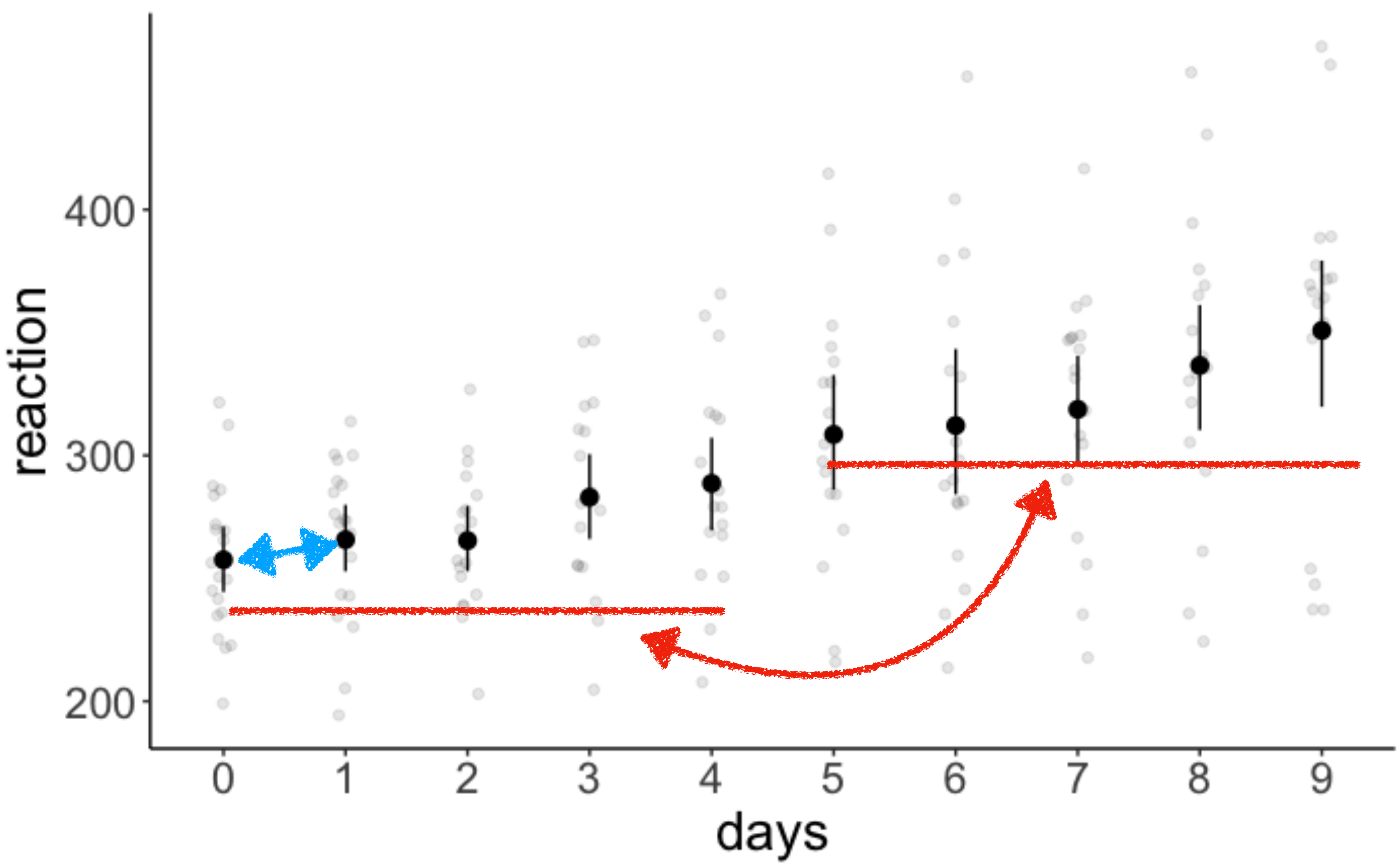
20 participants 2 with incomplete information

Testing specific hypotheses with linear contrasts

1. Is there a significant difference between day 0 and day 1?

2. Is there a significant difference between the days 0-4 and days 5-9?

subject	days	reaction
308	0	249.56
308	1	258.70
308	2	250.80
308	3	321.44
308	4	356.85
309	0	222.73
309	1	205.27
309	2	202.98
309	3	204.71
309	4	207.72



Sleep data

fit the model

```
fit = lmer(formula = reaction ~ 1 + days + (1 | subject),
           data = df.sleep %>%
             mutate(days = as.factor(days)))
```

Sleep data

fit the model

```r
fit = lmer(formula = reaction ~ 1 + days + (1 | subject),
           data = df.sleep %>%
             mutate(days = as.factor(days)))

contrast = list(first_vs_second = c(-1, 1, rep(0, 8)),
                early_vs_late = c(rep(-1, 5)/5, rep(1, 5)/5))

fit %>%
  emmeans(specs = "days",
          contr = contrast) %>%
  pluck("contrasts")
```

define the contrasts

test the contrasts

```
contrast         estimate    SE  df t.ratio p.value
first_vs_second      7.82 10.10 156   0.775  0.4398
early_vs_late       53.66  4.65 155  11.534  <.0001

Degrees-of-freedom method: kenward-roger
```

days	reaction
0	257.54
1	265.73

index	reaction
early	271.67
late	325.39

https://aosmith.rbind.io/2019/03/25/getting-started-with-emmeans/

2. Weight loss

Weight loss data

id	diet	exercises	timepoint	score
1	no	no	t1	10.43
1	no	no	t2	13.21
1	no	no	t3	11.59
1	yes	no	t1	10.20
1	yes	no	t2	12.51
1	yes	no	t3	14.60
2	no	no	t1	11.59
2	no	no	t2	10.66
2	no	no	t3	13.21
2	yes	no	t1	12.98
2	yes	no	t2	12.98
2	yes	no	t3	14.60

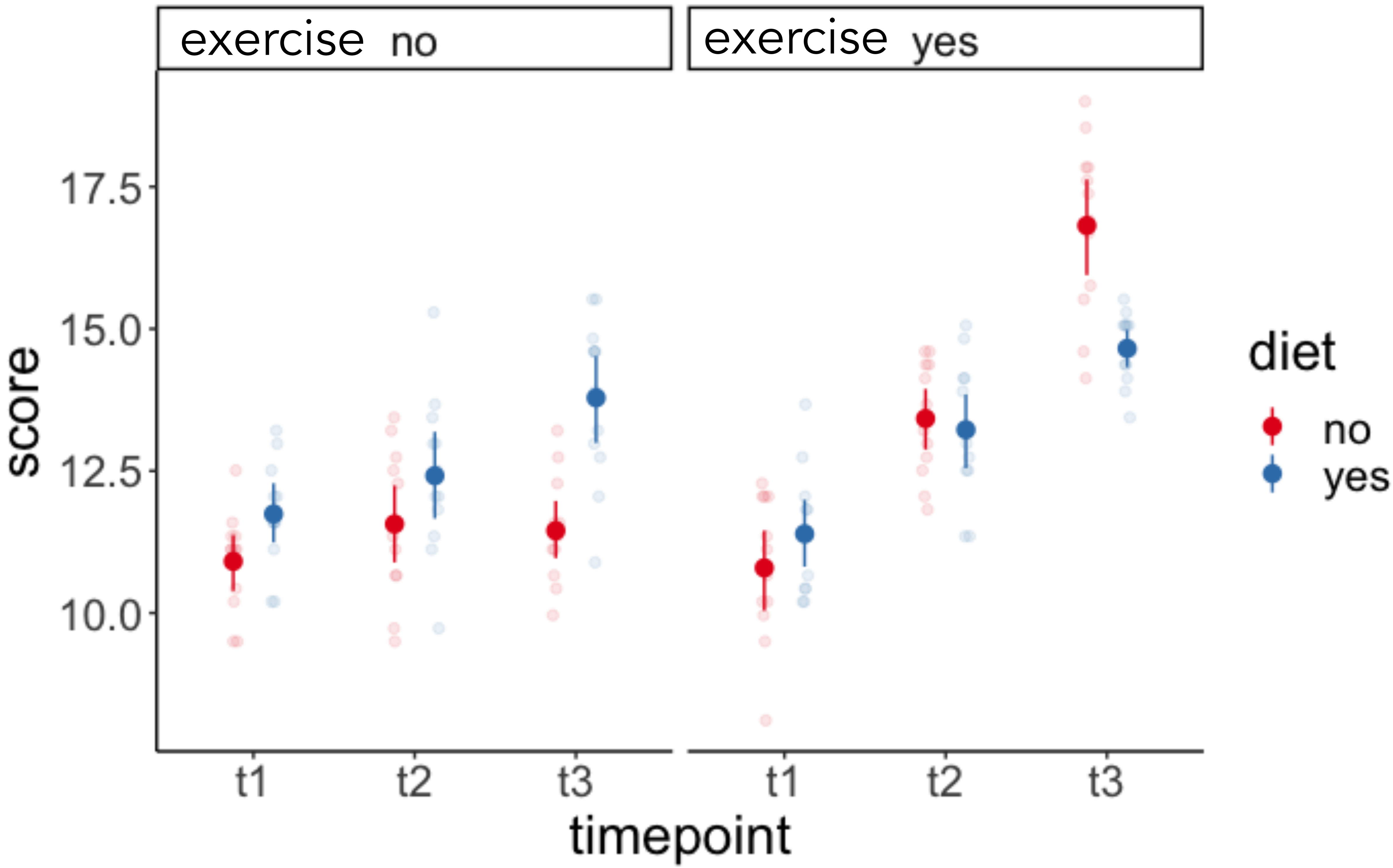
between participants: exercise yes/no

within participants: diet yes/no

within participants: time points

one observation in each cell, so we can use an ANOVA

Weight loss data



Weight loss data

```r
fit = aov_ez(id = "id",
        dv = "score",
        between = "exercises",
        within = c("diet", "timepoint"),
        data = df.weightloss)
```

df.weightloss

id	diet	exercises	timepoint	score
1	no	no	t1	10.43
1	no	no	t2	13.21
1	no	no	t3	11.59
1	yes	no	t1	10.20
1	yes	no	t2	12.51
1	yes	no	t3	14.60
2	no	no	t1	11.59

```
Anova Table (Type 3 tests)

Response: score
                  Effect          df  MSE         F  ges p.value
1              exercises       1, 22 1.84 38.77 *** .284  <.001
2                   diet       1, 22 0.65    7.91 * .028   .010
3         exercises:diet       1, 22 0.65 51.70 *** .157  <.001
4              timepoint 1.74, 38.26 1.48 82.20 *** .541  <.001
5    exercises:timepoint 1.74, 38.26 1.48 26.22 *** .274  <.001
6         diet:timepoint 1.61, 35.44 1.92      0.78 .013   .439
7 exercises:diet:timepoint 1.61, 35.44 1.92  9.97 *** .147  <.001
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

main effects and interactions

Weight loss data

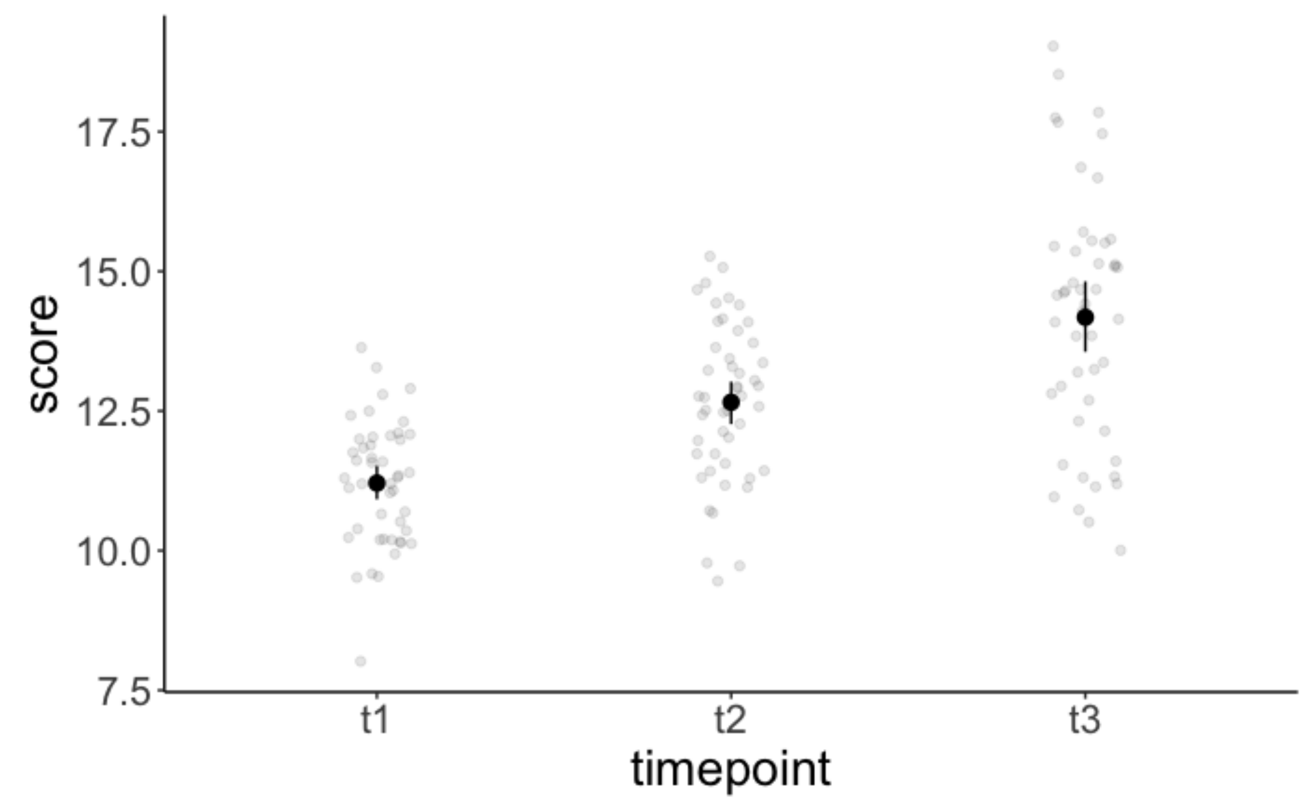
1. Is the score at the third time point different from the other two time points?
2. Is there a linear increase across time points?

```r
fit = aov_ez(id = "id",
           dv = "score",
           between = "exercises",
           within = c("diet", "timepoint"),
           data = df.weightloss)

contrasts = list(first_two_vs_last = c(-0.5, -0.5, 1),
                 linear_increase = c(-1, 0, 1))

fit %>%
  emmeans(spec = "timepoint",
          contr = contrasts)
```

```
contrast          estimate    SE df t.ratio p.value
 first_two_vs_last     2.24 0.200 44  11.194  <.0001
 linear_increase       2.97 0.231 44  12.820  <.0001
```

df.weightloss

id	diet	exercises	timepoint	score
1	no	no	t1	10.43
1	no	no	t2	13.21
1	no	no	t3	11.59
1	yes	no	t1	10.20
1	yes	no	t2	12.51
1	yes	no	t3	14.60
2	no	no	t1	11.59



3. politeness

Politeness

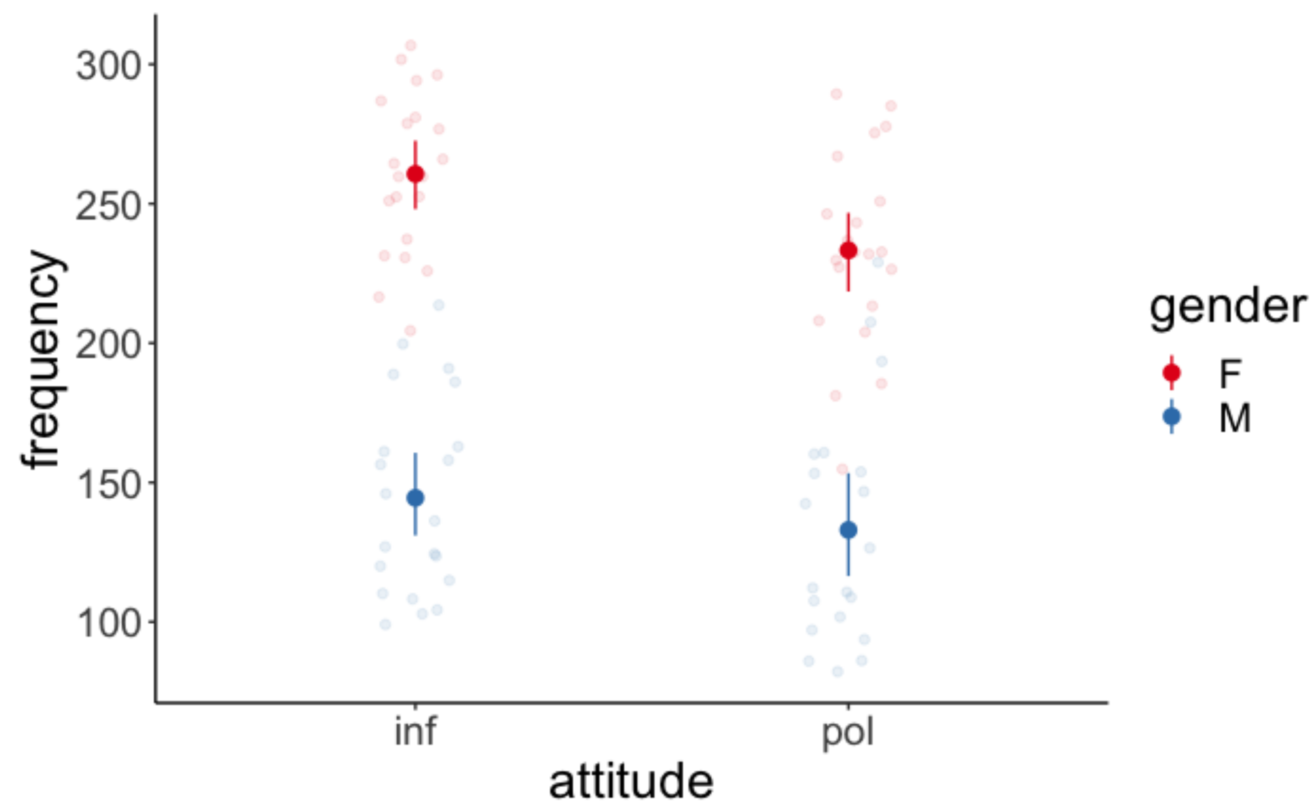
subject	gender	scenario	attitude	frequency
F1	F	1	pol	213.3
F1	F	1	inf	204.5
F1	F	2	pol	285.1
F1	F	2	inf	259.7
F1	F	3	pol	203.9
F1	F	3	inf	286.9
F1	F	4	pol	250.8
F1	F	4	inf	276.8
F1	F	5	pol	231.9
F1	F	5	inf	252.4
F1	F	6	pol	181.2
F1	F	6	inf	230.7
F1	F	7	inf	216.5
F1	F	7	pol	154.8
F3	F	1	pol	229.7

gender: female, male

scenario: different text prompt

attitude: polite vs. informal

frequency: pitch of voice

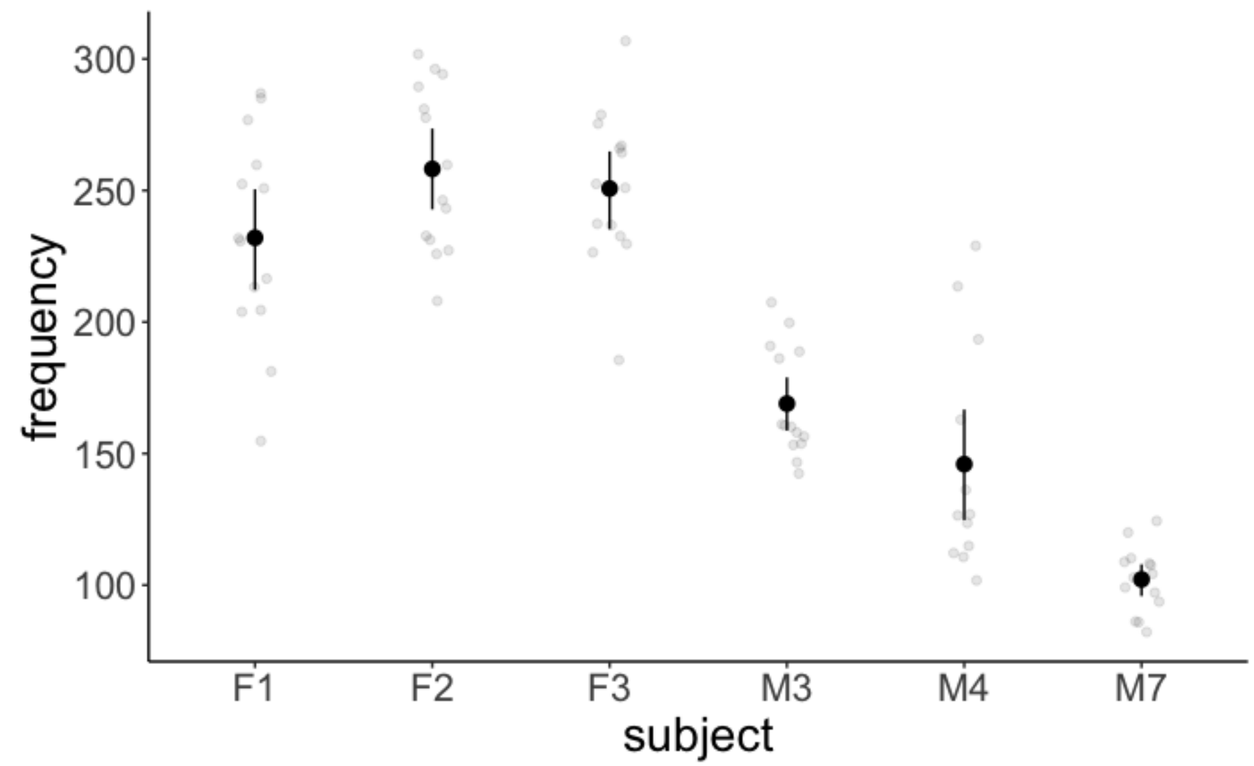


https://jontalle.web.engr.illinois.edu/MISC/lme4/bw_LME_tutorial.pdf

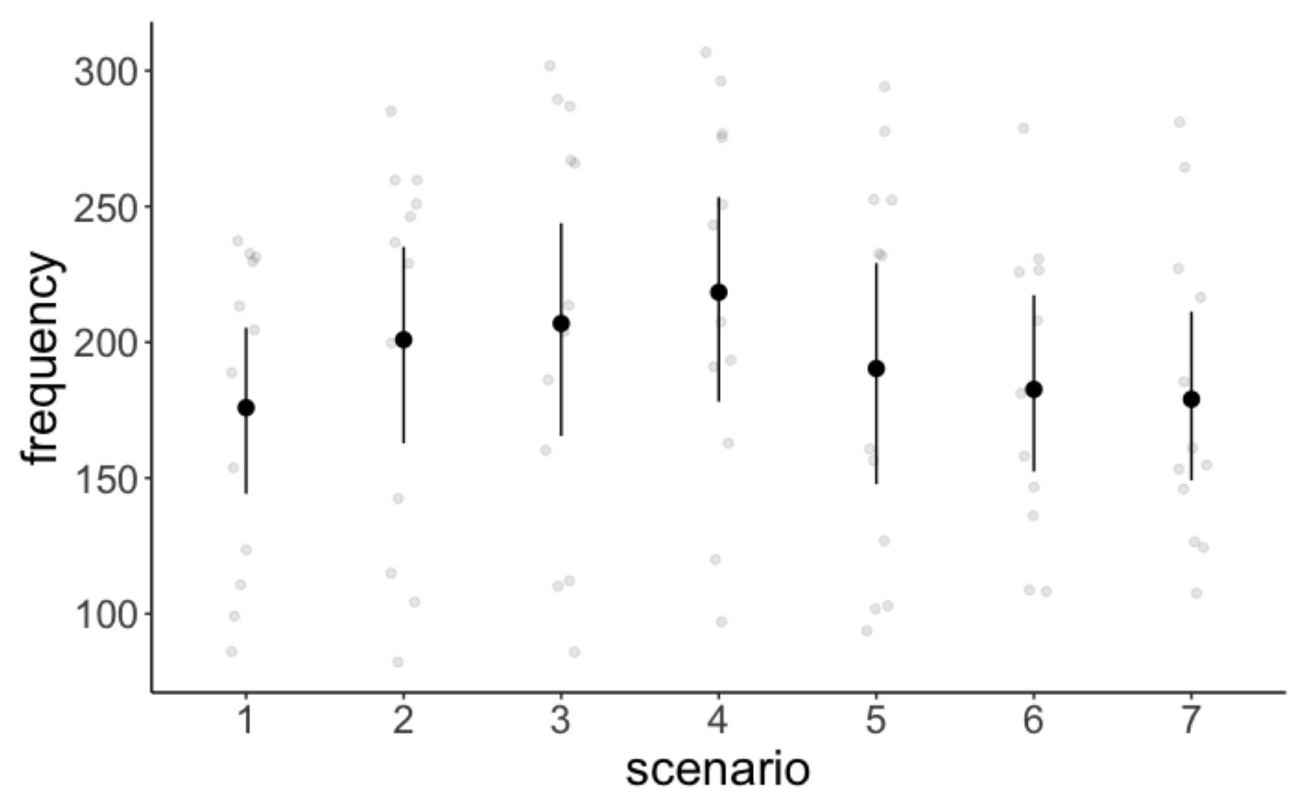
Politeness

subject	gender	scenario	attitude	frequency
F1	F	1	pol	213.3
F1	F	1	inf	204.5
F1	F	2	pol	285.1
F1	F	2	inf	259.7
F1	F	3	pol	203.9
F1	F	3	inf	286.9
F1	F	4	pol	250.8
F1	F	4	inf	276.8
F1	F	5	pol	231.9
F1	F	5	inf	252.4
F1	F	6	pol	181.2
F1	F	6	inf	230.7
F1	F	7	inf	216.5
F1	F	7	pol	154.8
F3	F	1	pol	229.7

variation across subjects



variation across scenarios

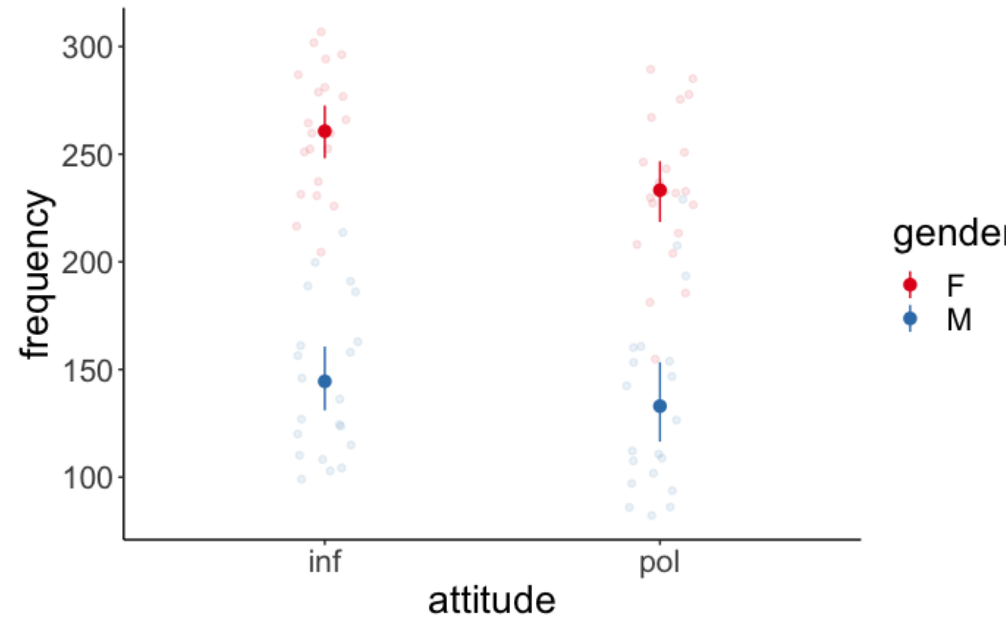


Politeness

Was there an effect of gender and attitude on pitch?

```
lmer(formula = frequency ~ 1 + attitude * gender + (1 | subject) + (1 | scenario),
     data = df.politeness) %>%
  joint_tests()
```

model term	df1	df2	F.ratio	p.value
attitude	1	69.04	12.497	0.0007
gender	1	4.00	26.578	0.0067
attitude:gender	1	69.04	1.969	0.1650



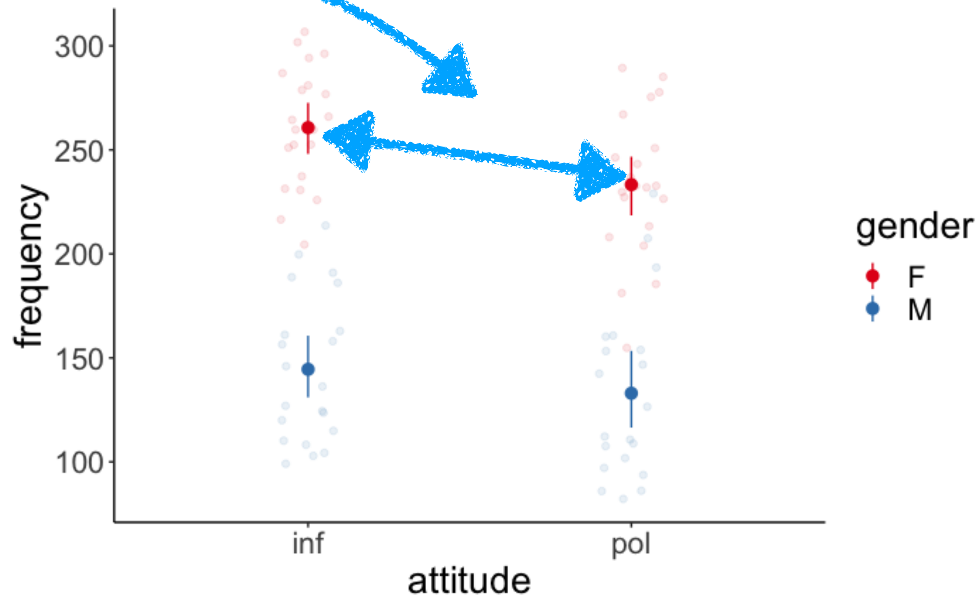
main effect of attitude, main effect of gender, no significant interaction effect

Politeness

Was there a difference between informal and polite speech for female participants?

```
fit = lmer(formula = frequency ~ 1 + attitude * gender + (1 | subject) + (1 | scenario),
           data = df.politeness)

fit %>%
  emmeans(specs = pairwise ~ attitude + gender,
          adjust = "none")
```

```
contrast        estimate    SE    df t.ratio p.value
 inf F - pol F      27.4  7.79 69.00 3.517   0.0008
 inf F - inf M     116.2 21.73  4.56 5.348   0.0040
 inf F - pol M     128.0 21.77  4.59 5.881   0.0027
 pol F - inf M      88.8 21.73  4.56 4.087   0.0115
 pol F - pol M     100.6 21.77  4.59 4.623   0.0071
 inf M - pol M      11.8  7.90 69.08 1.497   0.1390

Degrees-of-freedom method: kenward-roger
```



yes, there was significant difference in pitch for women between informal and formal speech

Politeness

Was there an effect of gender and attitude on pitch?

ANOVA

```
aov_ez(id = "subject",
       dv = "frequency",
       between = "gender",
       within = "attitude",
       data = df.politeness)
```

```
More than one observation per cell, aggregating the data using
mean (i.e, fun_aggregate = mean)!Missing values for following
ID(s):
M4
Removing those cases from the analysis.Anova Table (Type 3 tests)

Response: frequency
           Effect   df     MSE          F  ges p.value
1          gender 1, 3 1729.42    17.22 * .851    .025
2        attitude 1, 3    3.65 309.71 *** .179   <.001
3 gender:attitude 1, 3    3.65    21.30 * .015    .019
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

ignores variation between scenarios, and just takes the mean

interaction effect

LMER

```
lmer(formula = frequency ~ 1 + attitude * gender +
                 (1 | subject) + (1 | scenario),
     data = df.politeness) %>%
  joint_tests()
```

```
model term      df1   df2 F.ratio p.value
 attitude         1 69.04  12.497  0.0007
 gender           1  4.00  26.578  0.0067
 attitude:gender  1 69.04   1.969  0.1650
```

no interaction effect

Summary

- Quick recap
- Understanding `lmer()` summary
- A worked example
- Reporting results
- Understanding `lmer()` syntax
- `lmer()` standard operating procedures
- Some more examples

Feedback

How was the pace of today's class?

How happy were you with today's class overall?

What did you like about today's class? What could be improved next time?

Thank you!