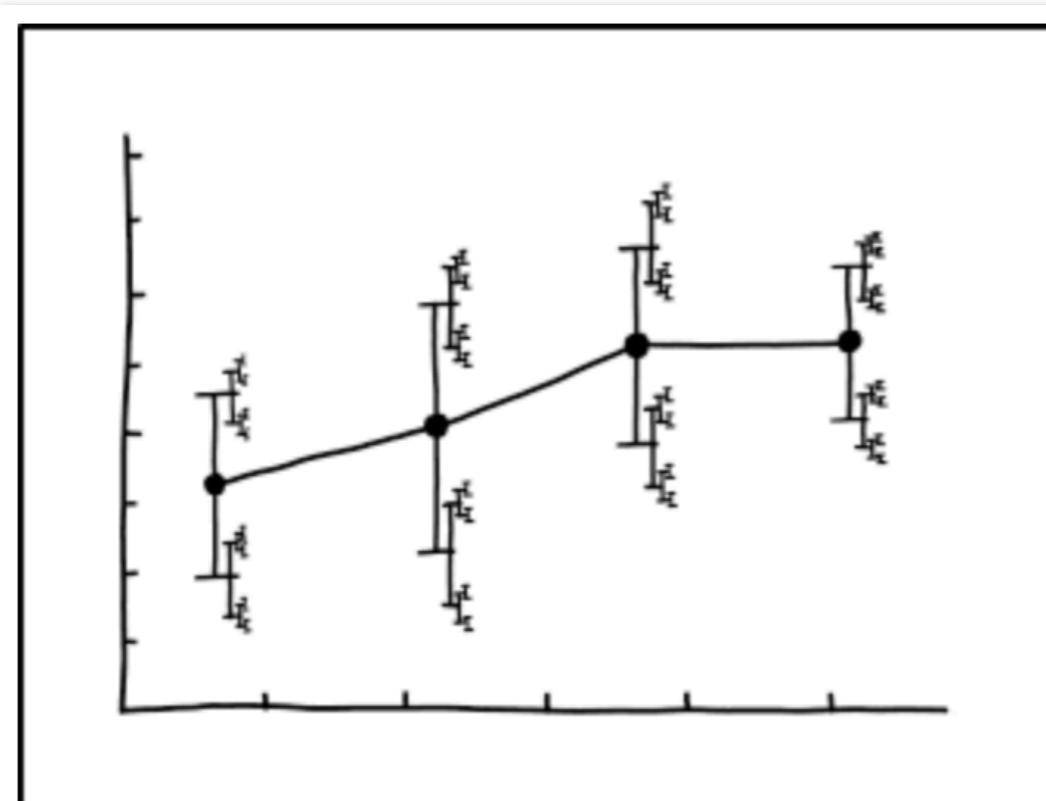


Linear mixed effects models 2



I DON'T KNOW HOW TO PROPAGATE
ERROR CORRECTLY, SO I JUST PUT
ERROR BARS ON ALL MY ERROR BARS.

Chat

If you could give a piece of advice to your younger self, what would it be?

To: Everyone

More

Type message here...

O COLLABORATIVE PLAYLIST

psych252

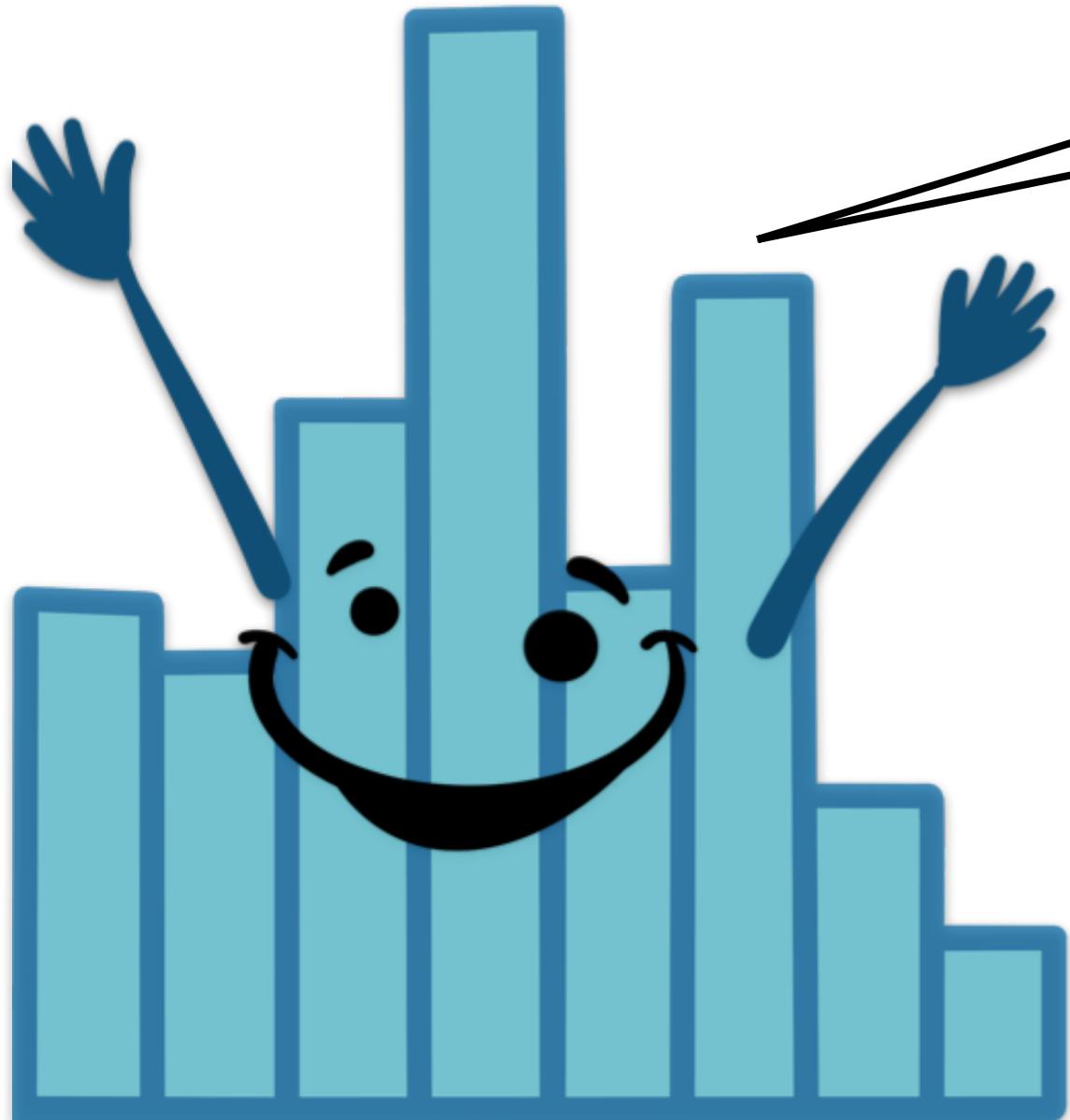
<https://tinyurl.com/psych252spotify21>

PLAY

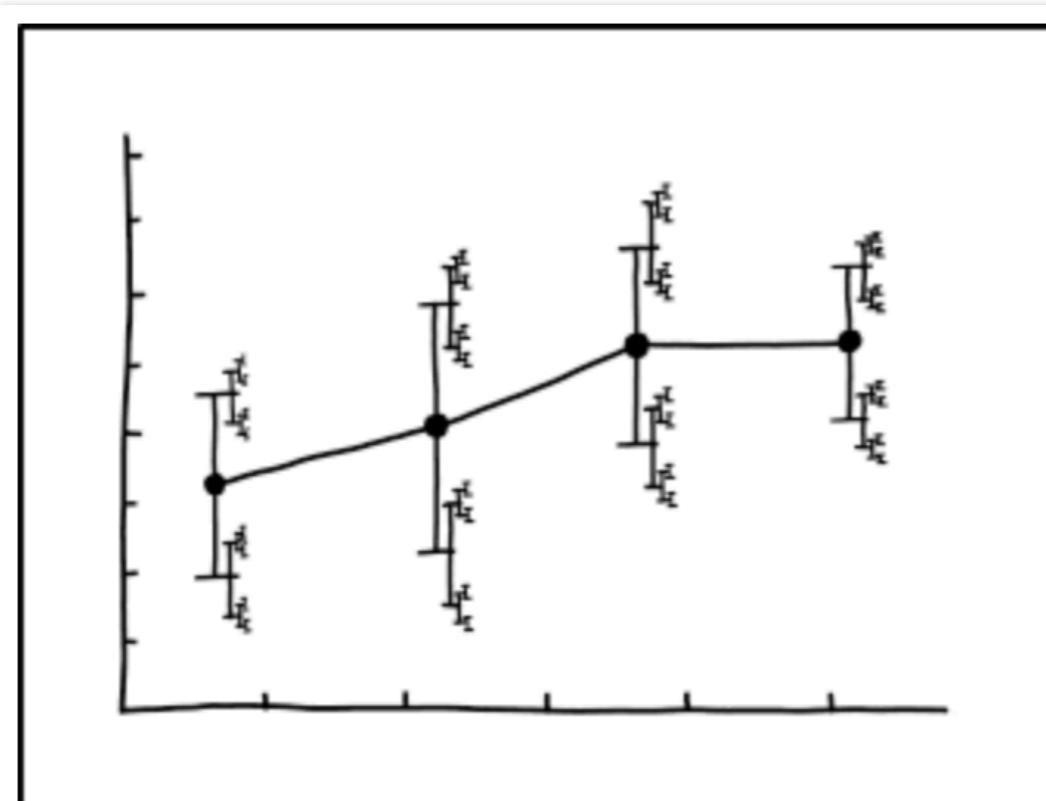
A Spotify collaborative playlist cover for "psych252". It features a dark background with light blue bars resembling a histogram or bar chart. A cartoon character with a smiling face and arms is positioned among the bars. The text "O COLLABORATIVE PLAYLIST" is at the top left, "psych252" is in large white letters in the center, and the URL "https://tinyurl.com/psych252spotify21" is below it. A green "PLAY" button is at the bottom left, and a three-dot menu icon is at the bottom right.

02/26/2021

Remember to
record the
lecture!



Linear mixed effects models 2



I DON'T KNOW HOW TO PROPAGATE
ERROR CORRECTLY, SO I JUST PUT
ERROR BARS ON ALL MY ERROR BARS.

Chat

If you could give a piece of advice to your younger self, what would it be?

To: Everyone

More

Type message here...

O COLLABORATIVE PLAYLIST

psych252

<https://tinyurl.com/psych252spotify21>

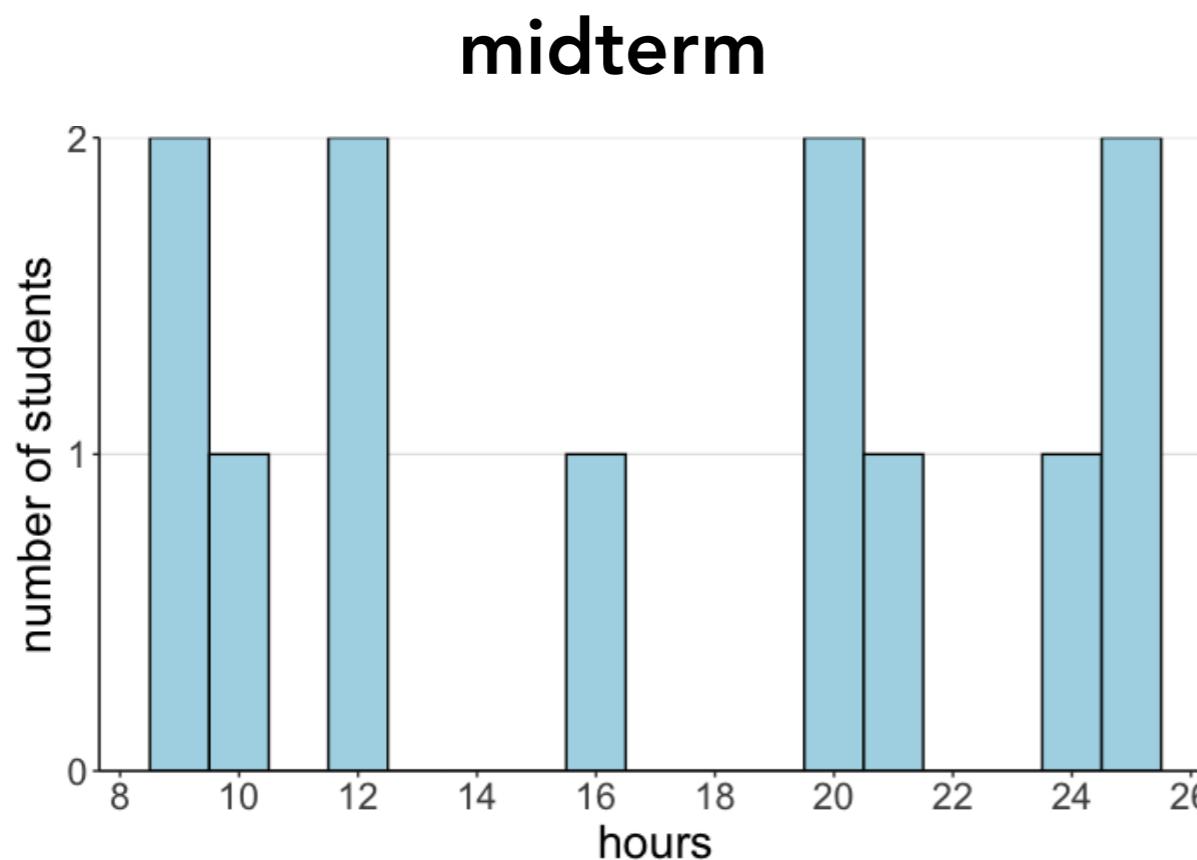
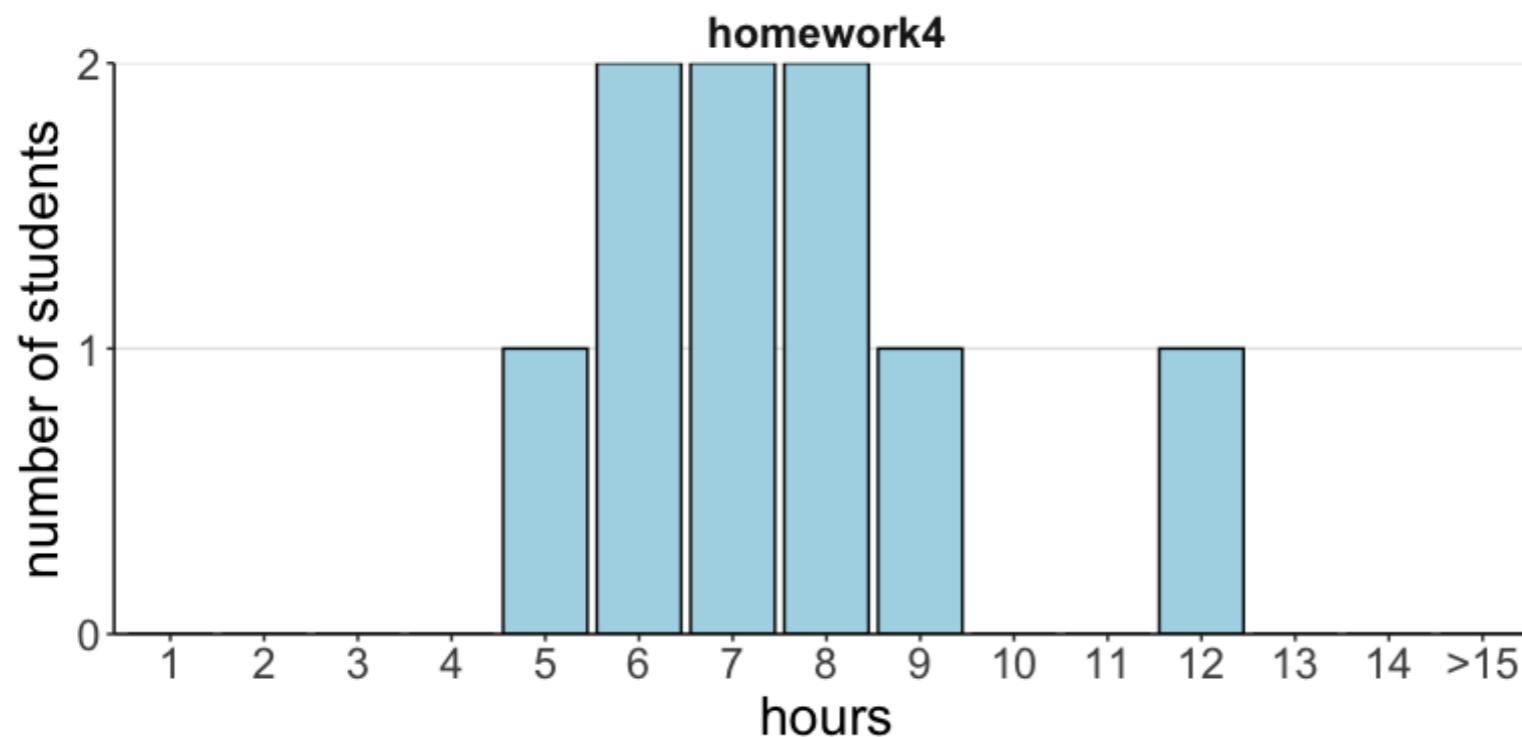
PLAY

A Spotify collaborative playlist cover for "psych252". It features a dark background with light blue bars resembling a histogram or bar chart. A cartoon character with a smiling face and arms is positioned among the bars. The text "O COLLABORATIVE PLAYLIST" is at the top left, "psych252" is in large white letters in the center, and the URL "https://tinyurl.com/psych252spotify21" is below it. A green "PLAY" button is at the bottom left, and a three-dot menu icon is at the bottom right.

02/26/2021

Feedback

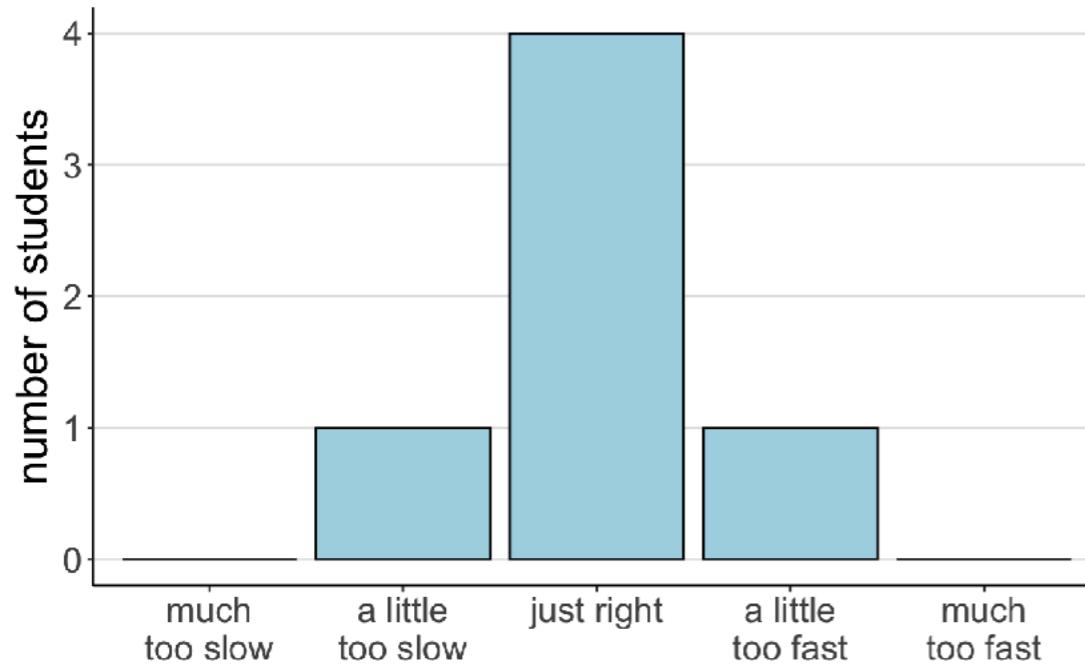
Feedback



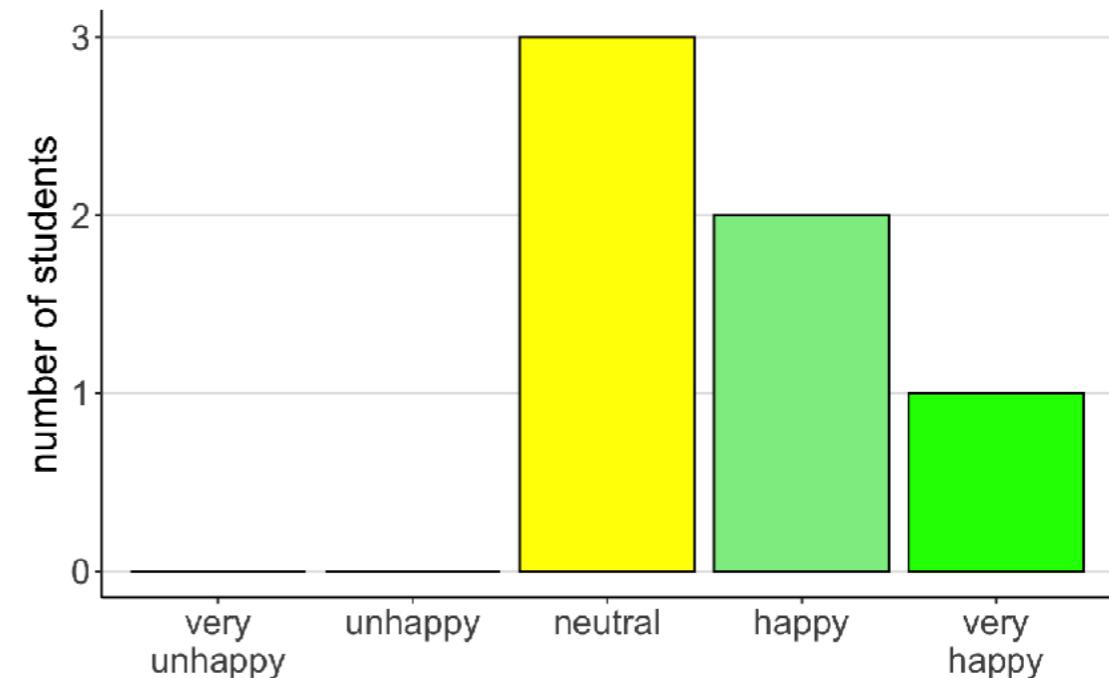
it was still too long (will shorten more in the future)

Feedback

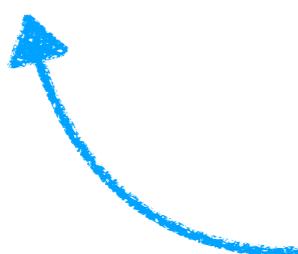
How was the pace of today's class?



How happy were you with today's class overall?



<https://tinyurl.com/psych252survey21>



so that students who can't be here
synchronously can give feedback, too

Feedback

i liked the review of the cross validation methods, and i think the linear mixed effects model was introduced well. i did feel like it was a bit fast and i'd like to talk a bit more about what fixed / random effects mean - i think i'm still a bit unclear as of now

Logistics

Homework 5

available after class today

Part 1: Causal graphs (2 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



Part 2: Mediation (3 points)

It's intuitive to believe that additional years of education would increase one's yearly earnings, but the relationship is more complex than it initially seems. For instance, a family which provided him with greater opportunities for his child to go to college may have prohibited him from pursuing further educational opportunities. In this section, we will explore the dataset of a seminal work that established a causal link between education and income through mediation. Here's the first paragraph from the paper.

Every developed country in the world has a compulsory school law. About the effect these laws have on educational attainment is an unusual natural experiment to estimate the impact of education on income in the United States. The experiment stems from the fact that children start school at different ages, while compulsory schooling begins in school until their sixteenth or seventeenth birthday. Requirements and compulsory schooling laws compel students to stay in school longer than students born in other months. Education is correlated with personal attributes other than age at birth, such as exogenous variation in education that can be used to estimate the effect of education on earnings.

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

due Thursday, March 4th, 8pm

3.1 (3 points)

Since there are so many different models we could make with all those variables, we would like to use some measure to compare how well they fit the data. Some methods are AIC/BIC and cross-validation.

Create a data frame with 4 rows and the following 5 columns: `model_name`, `rsquared`, `logLik`, `AIC`, `BIC`. Each row should represent 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`

To create the data frame, make sure to use `map()` for fitting the linear models, and you can use `glance()` from the `broom` package to get the model summaries in tidy format. For defining model 2, the `poly()` function is helpful.

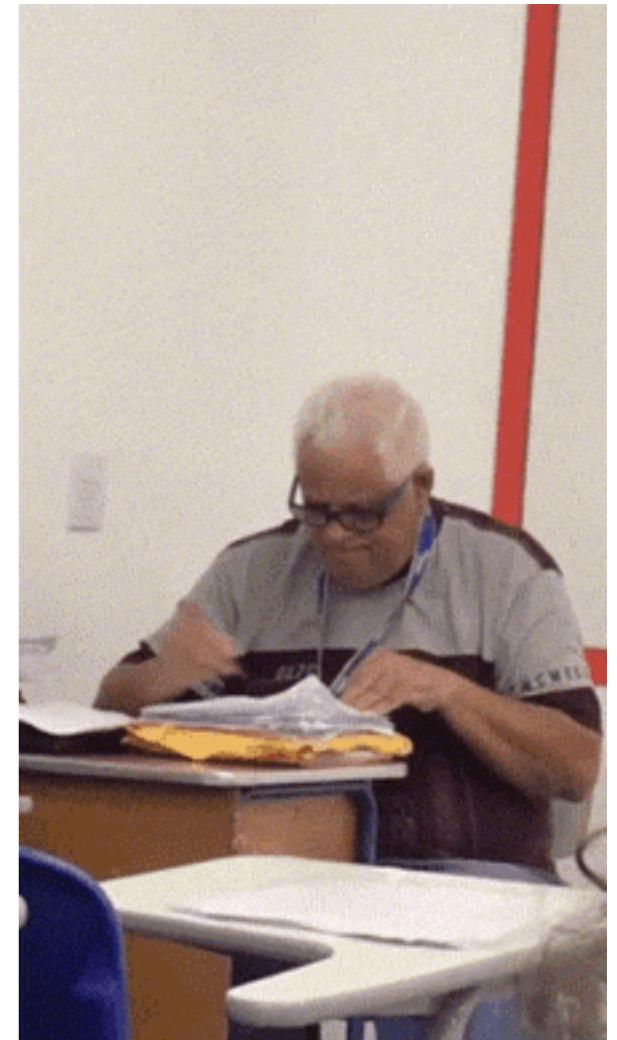
Which model is the best model using the different measures? Are the different model comparison measures consistent?

```
### YOUR CODE HERE
df.compare =
#####
df.compare
```

YOUR ANSWER HERE

Midterm

Grades will be released soon!



Plan for today

- Quick recap: Interpreting `summary()` output
- Linear mixed effects model
 - A worked example
 - Getting p-values
 - Reporting results
 - Let's simulate some `lmer()`s
 - Understanding `lmer()` syntax
 - `lmer()` standard operating procedures

Quick recap: Interpreting summary() output

lm() output

```
1 lm(formula = balance ~ income + student + income:student, data = df.credit) %>%
2   summary()
```

```
Call:
lm(formula = balance ~ income + student + income:student,
data = df.credit)
```

```
Residuals:
```

Min	1Q	Median	3Q	Max
-773.39	-325.70	-41.13	321.65	814.04

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	200.6232	33.6984	5.953	5.79e-09 ***
income	6.2182	0.5921	10.502	< 2e-16 ***
studentYes	476.6758	104.3512	4.568	6.59e-06 ***
income:studentYes	-1.9992	1.7313	-1.155	0.249

```
---
```

```
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
```

```
Residual standard error: 391.6 on 396 degrees of freedom
```

```
Multiple R-squared: 0.2799, Adjusted R-squared: 0.2744
```

```
F-statistic: 51.3 on 3 and 396 DF, p-value: < 2.2e-16
```



```
1 fit_c = lm(formula = balance ~ student + income:student, data = df.credit)
2 fit_a = lm(formula = balance ~ income + student + income:student, data = df.credit)
3
4 anova(fit_c, fit_a)
```

```
1 fit_c = lm(formula = balance ~ income + student, data = df.credit)
2 fit_a = lm(formula = balance ~ income + student + income:student, data = df.credit)
3
4 anova(fit_c, fit_a)
```

lm() output

```
1 lm(formula = balance ~ income + student + income:student, data = df.credit) %>%
2   summary()
```

```
Call:
lm(formula = balance ~ income + student + income:student,
data = df.credit)
```

Residuals:

Min	1Q	Median	3Q	Max
-773.39	-325.70	-41.13	321.65	814.04

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	200.6232	33.6984	5.953	5.79e-09 ***
income	6.2182	0.5921	10.502	< 2e-16 ***
studentYes	476.6758	104.3512	4.568	6.59e-06 ***
income:studentYes	-1.9992	1.7313	-1.155	0.249

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '

Residual standard error: 391.6 on 396 degrees of freedom
Multiple R-squared: 0.2799, Adjusted R-squared: 0.2744
F-statistic: 51.3 on 3 and 396 DF, p-value: < 2.2e-16

```
1 fit_c = lm(formula = balance ~ 1, data = df.credit)
2 fit_a = lm(formula = balance ~ income + student + income:student, data = df.credit)
3
4 anova(fit_c, fit_a)
```

Analysis of Variance Table

Model 1: balance ~ 1

Model 2: balance ~ 1 + income

	Res.Df	RSS	Df	Sum of Sq	F	Pr (>F)
1	399	84339912				
2	398	66208745	1	18131167	108.99	< 2.2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 . 0.1 ' ' 1						

anova () gives me F s
but lm () gives me t s ?

deterministic mapping
between t and F

$$t^2 = F$$

Call:
lm(formula = balance ~ 1 + income, data = df.credit)

Residuals:

Min	1Q	Median	3Q	Max
-803.64	-348.99	-54.42	331.75	1100.25

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	246.5148	33.1993	7.425	6.9e-13 ***
income	6.0484	0.5794	10.440	< 2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 407.9 on 398 degrees of freedom
Multiple R-squared: 0.215, Adjusted R-squared: 0.213
F-statistic: 109 on 1 and 398 DF, p-value: < 2.2e-16

Linear mixed effects model

highly recommended reading!

- 1 Understanding mixed effects models through data simulation

Lisa M. DeBruine¹ & Dale J. Barr¹

¹ Institute of Neuroscience and Psychology, University of Glasgow

In press at Advances in Methods and Practices in Psychological Science

Abstract

Experimental designs that sample both subjects and stimuli from a larger population need to account for random effects of both subjects and stimuli using mixed effects models. However, much of this research is analyzed using ANOVA on aggregated responses because researchers are not confident specifying and interpreting mixed effects models. The tutorial will explain how to simulate data with random effects structure and analyse the data using linear mixed effects regression (with the lme4 R package), with a focus on interpreting the output in light of the simulated parameters. Data simulation can not only enhance understanding of how these models work, but also enables researchers to perform power calculations for complex designs. All materials associated with this article can be accessed at <https://osf.io/3cz2e/>.

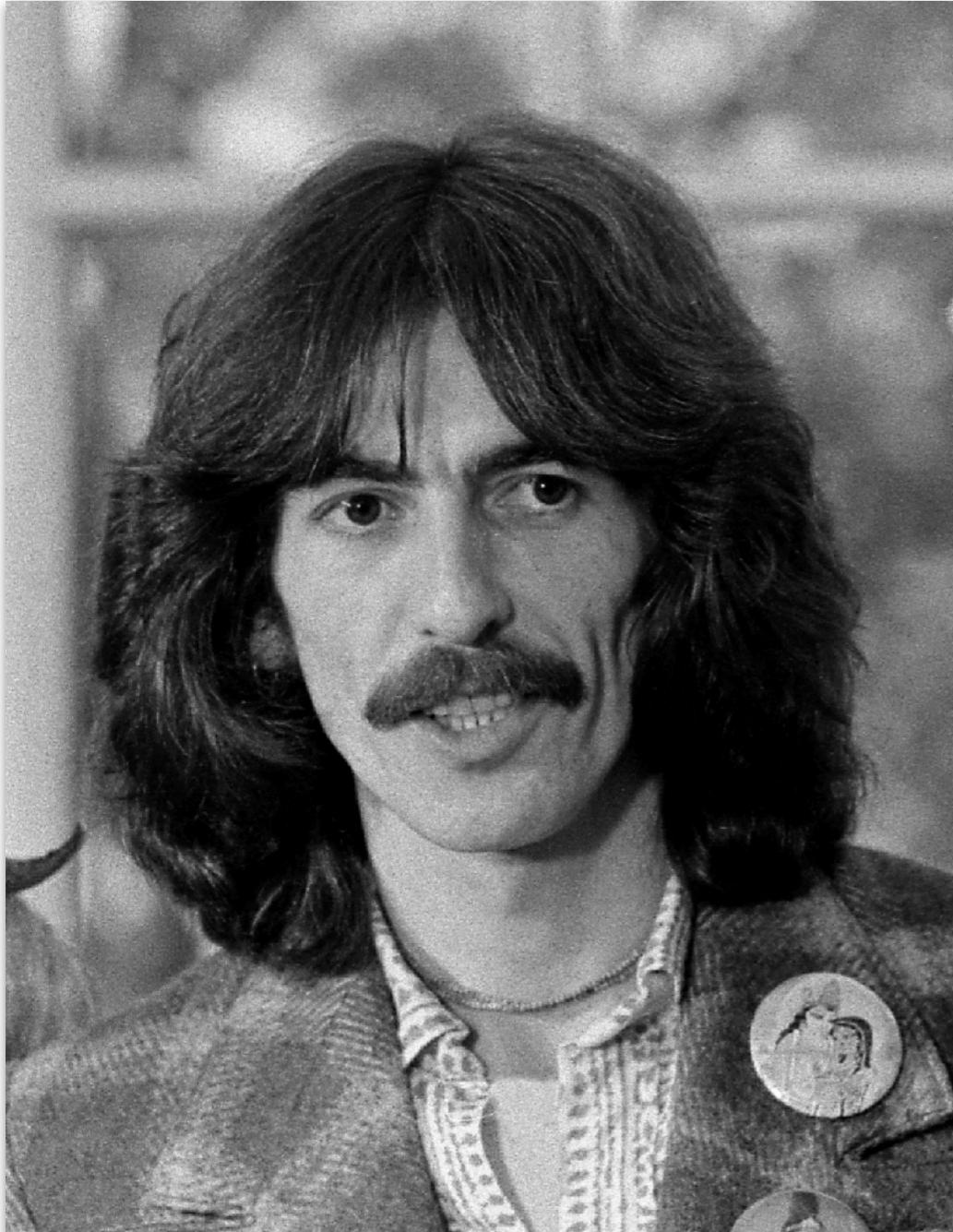
Keywords: simulation, mixed effects models, power, lme4, R

Word count: 7890



<https://psyarxiv.com/xp5cy/>

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



born on February 25th

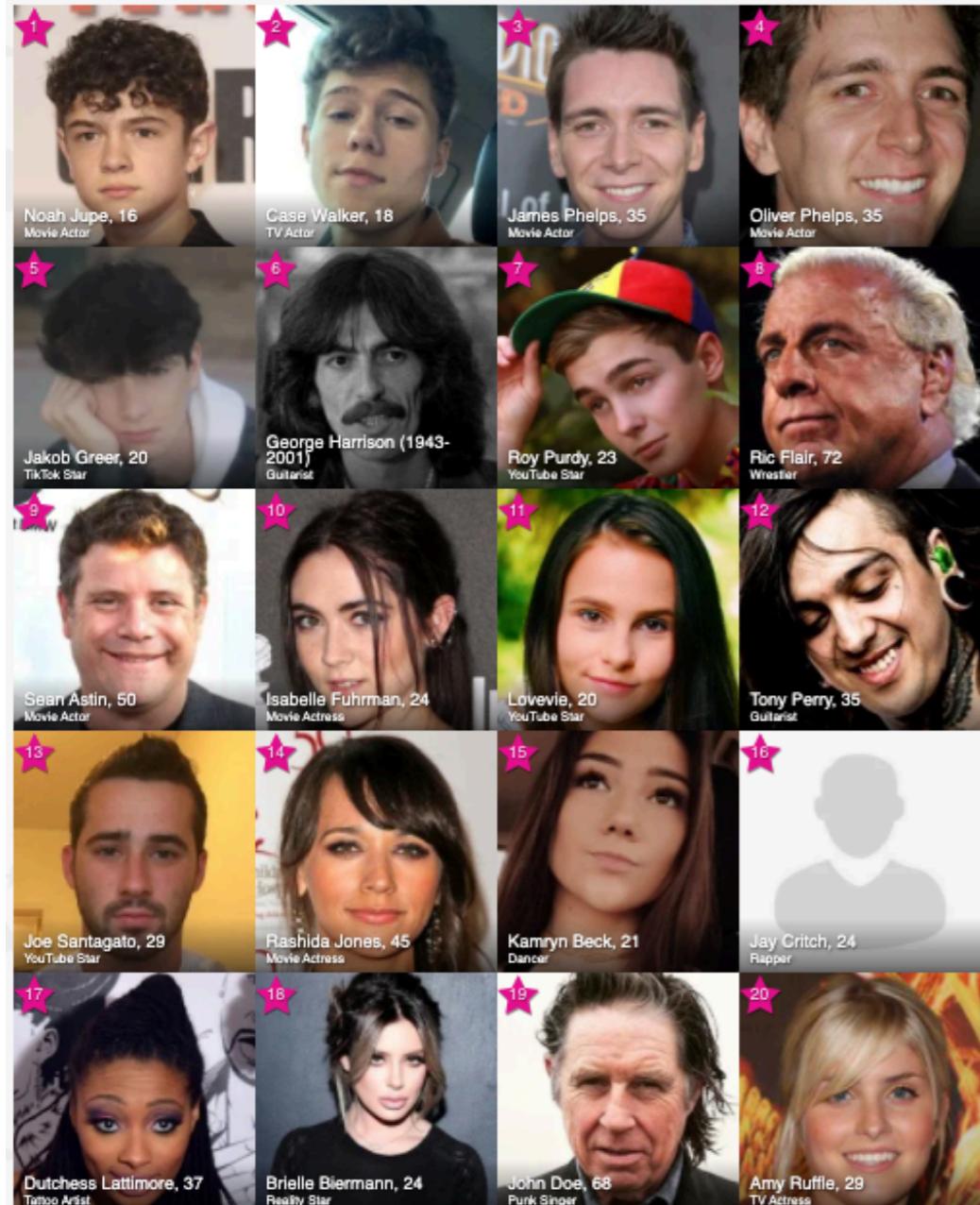
N = 100 participants



born on February 26th

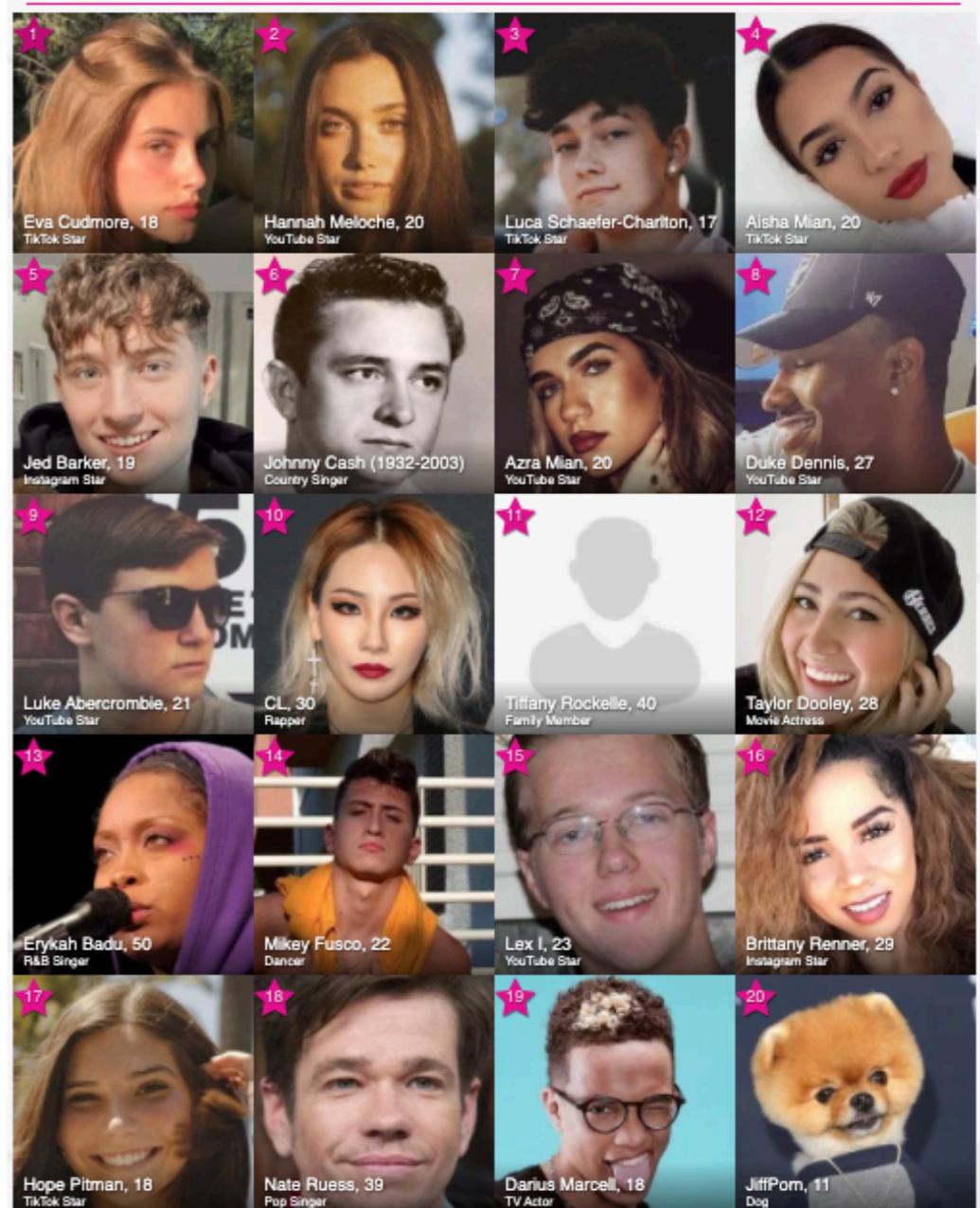
```
1 lm(formula = trustworthy ~ 1 + birthday,  
2   data = df.birthday)
```

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



born on February 25th

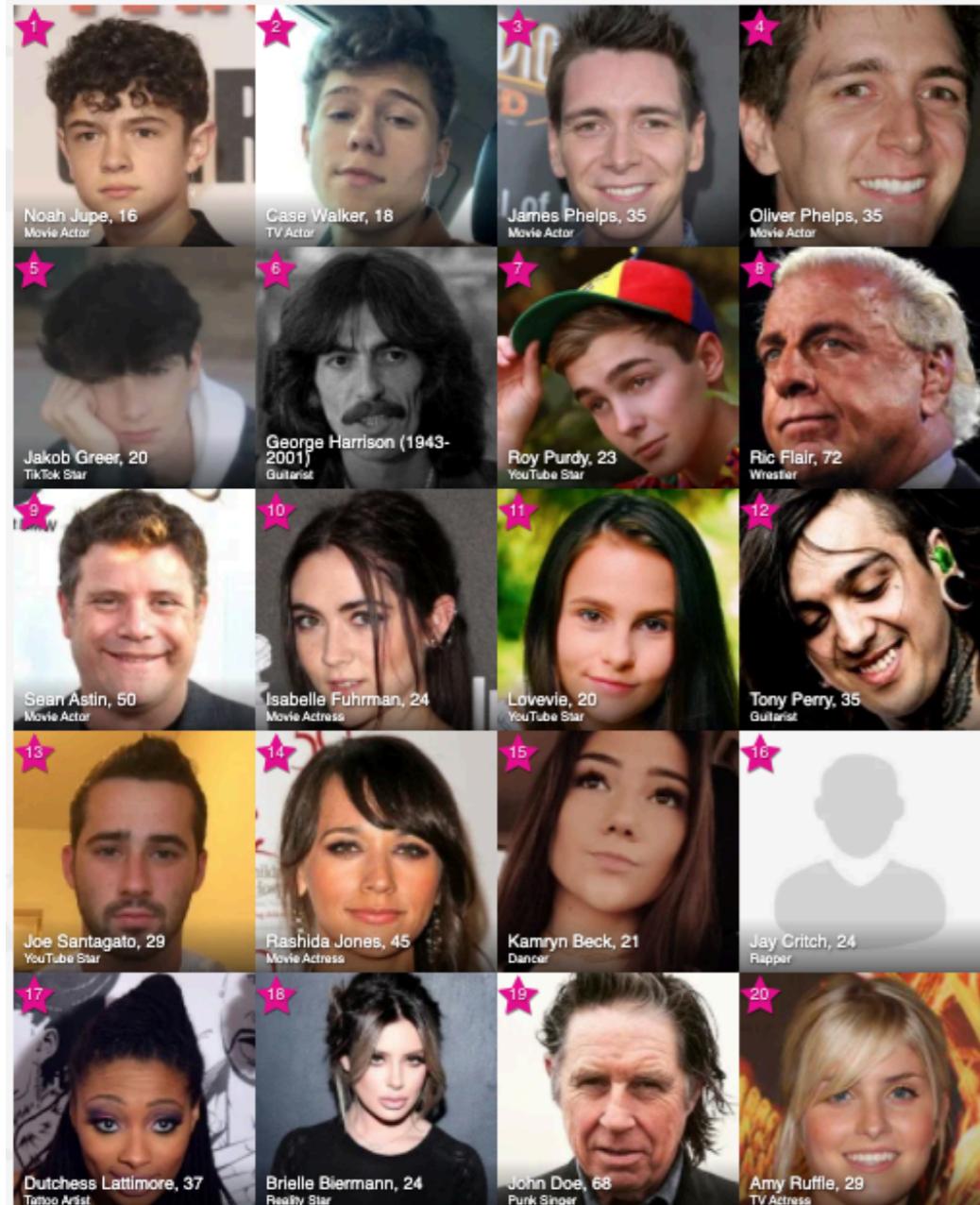
N = 2 participants



born on February 26th

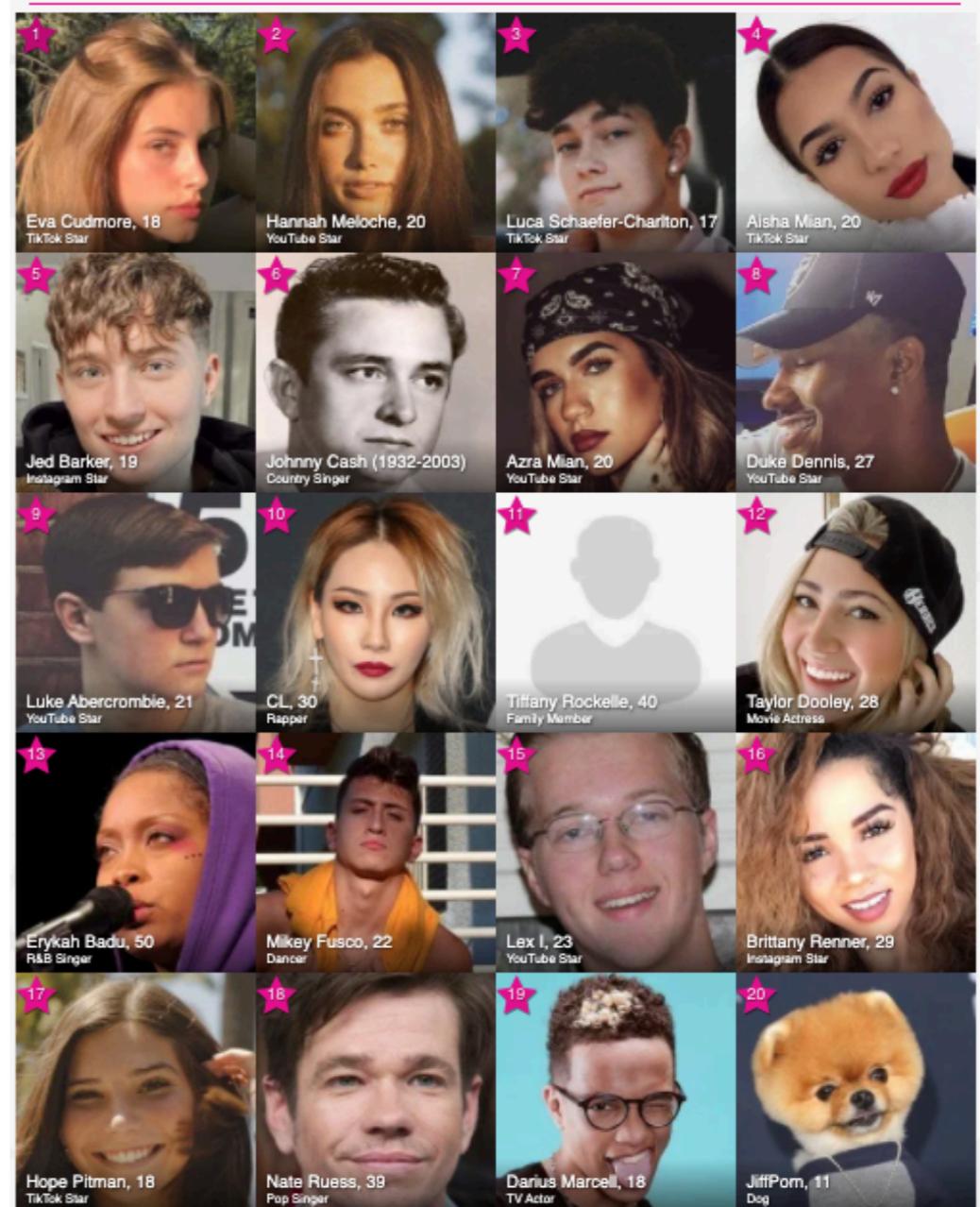
1 lm(formula = trustworthy ~ 1 + birthday,
2 data = df.birthday)

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



born on February 25th

N = 100 participants



born on February 26th

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

"Recently, psycholinguists have adopted linear mixed-effects modeling as the standard for statistical analysis, given numerous advantages over ANOVA, including the ability to **simultaneously model subject and stimulus variation**, to **gracefully deal with missing data or unbalanced designs**, and to **accommodate arbitrary types of continuous and categorical predictors or response variables.**"

"Models that account for subjects and stimuli (or other factors) as non-essential, exchangeable features of an experiment **will better characterize the uncertainty in the resulting estimates** and thus, **improve the generality of inferences we draw** from them (Yarkoni, 2019). That said, we **empathize with the frustration** — and sometimes, exasperation — expressed by many novices when they attempt to grapple with these models in their research. A profitable way to **build understanding and confidence is through data simulation.**"

A worked example

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

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

**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

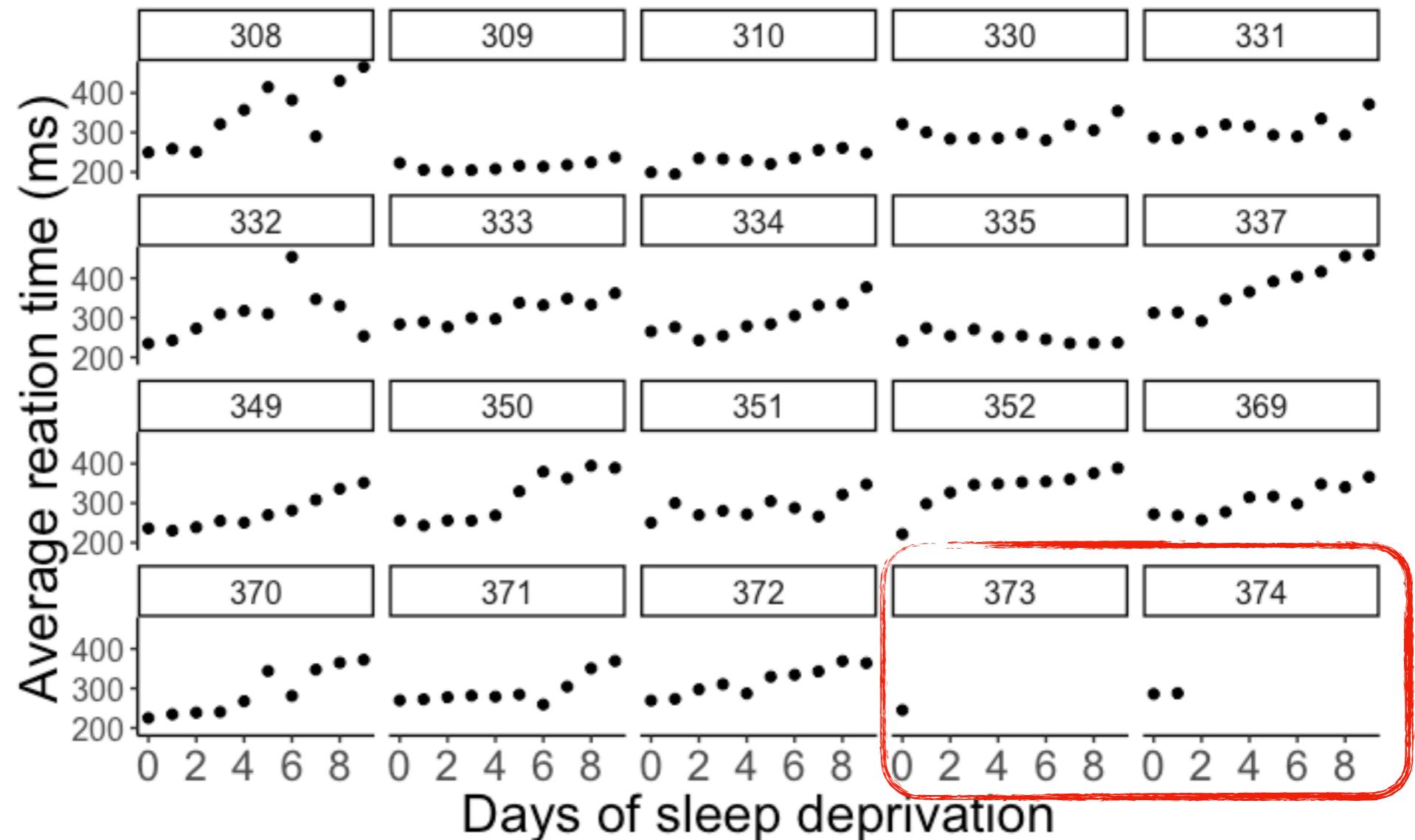
How does sleep deprivation affect reaction time?

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

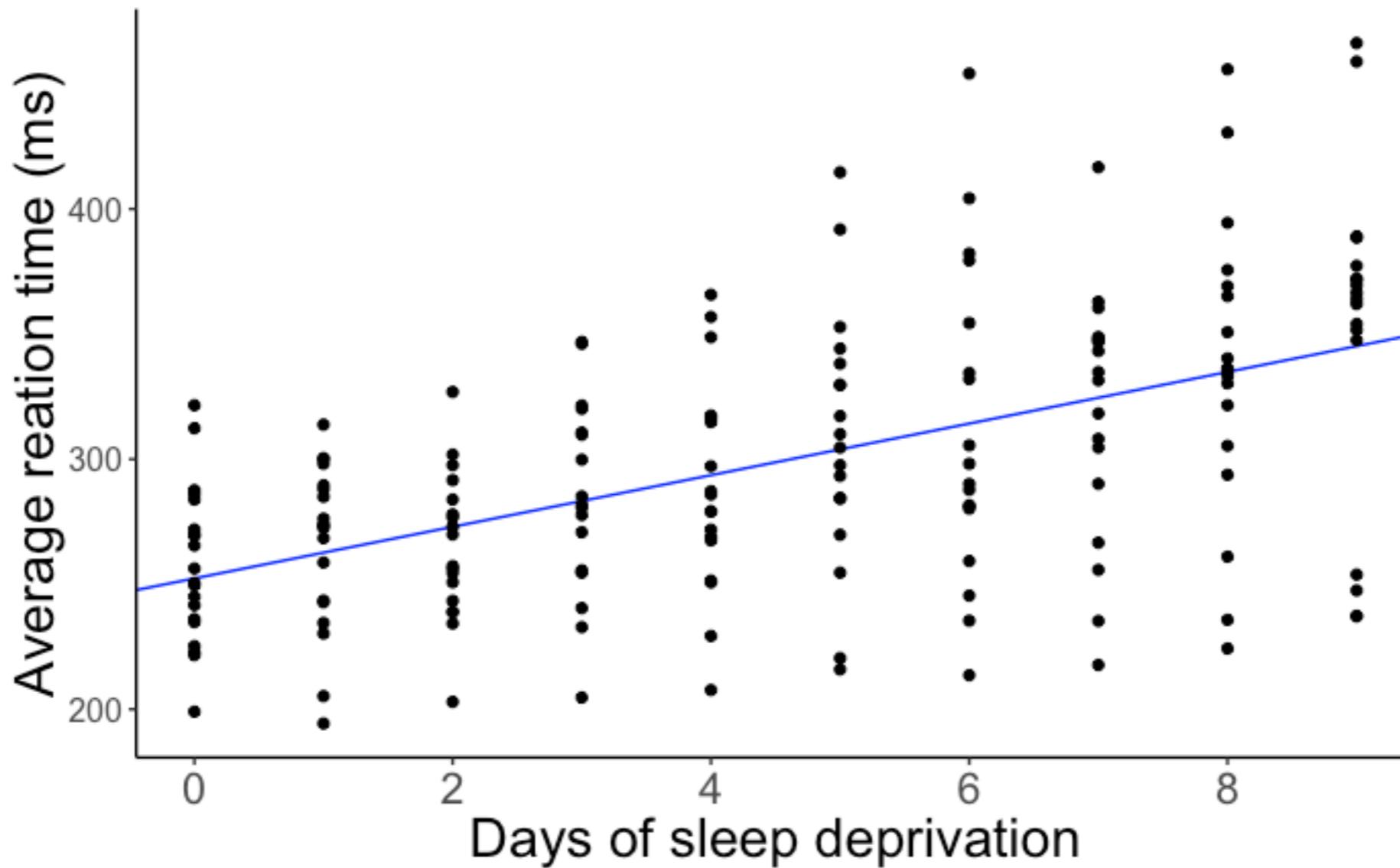
2 with incomplete information

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

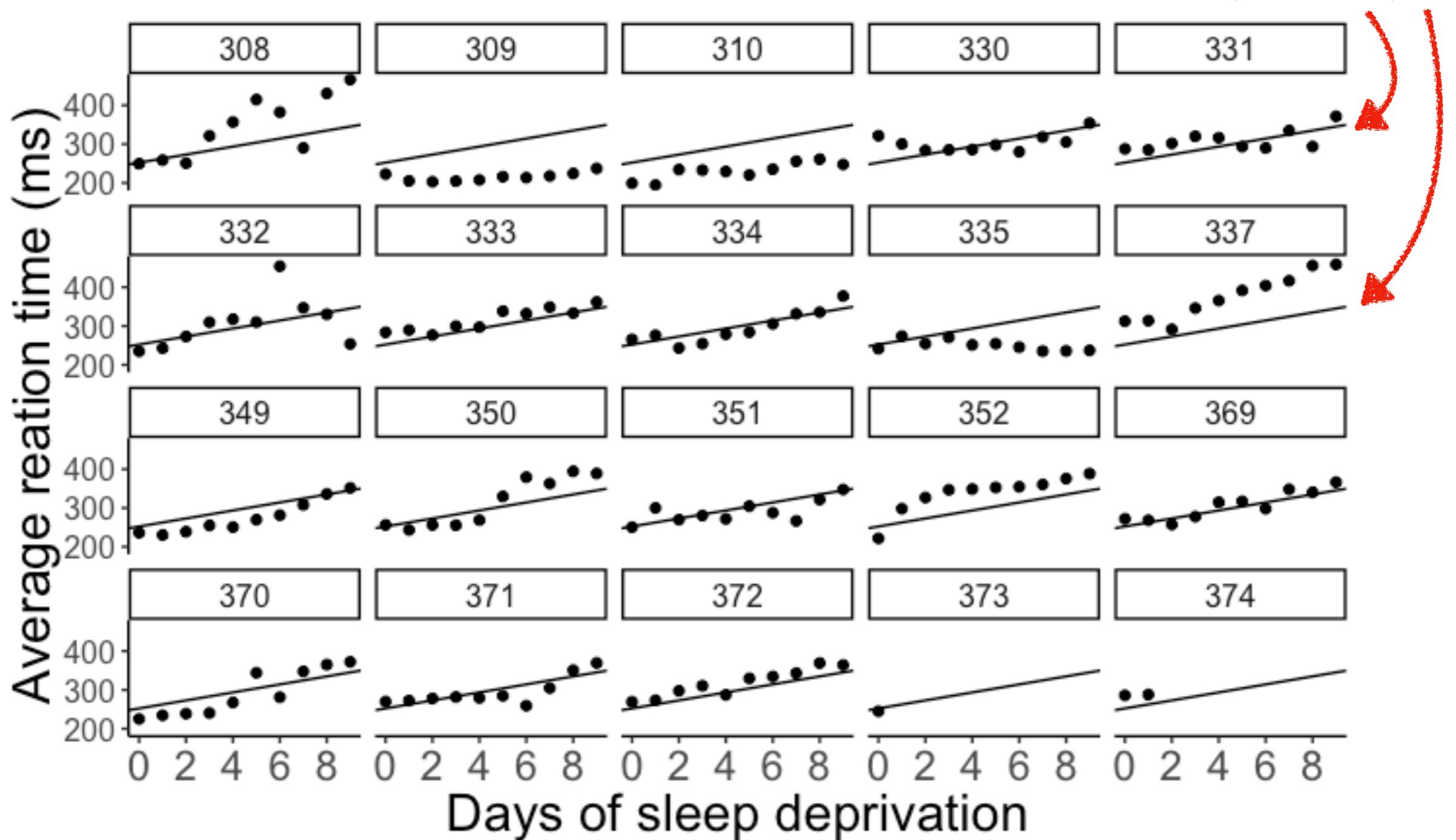
```
lm(formula = reaction ~ days,  
  data = df.sleep)
```



Complete pooling: Fit one global regression

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

same line for
each participant



No pooling: Fit separate regressions

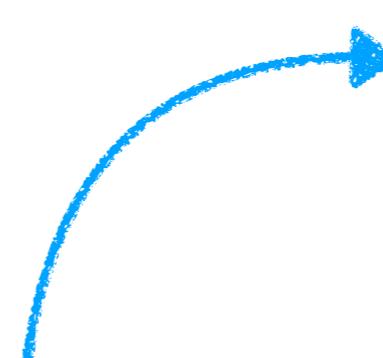
```
1 df.no_pooling = df.sleep %>%
2   group_by(subject) %>%
3   nest(data = c(days, reaction)) %>%
4   mutate(fit = map(data, ~ lm(reaction ~ days, data = .)),
5         params = map(fit, tidy)) %>%
6   unnest(c(params)) %>%
7   select(subject, term, estimate) %>%
8   complete(subject, term, fill = list(estimate = 0)) %>%
9   pivot_wider(names_from = term, values_from = estimate) %>%
10  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

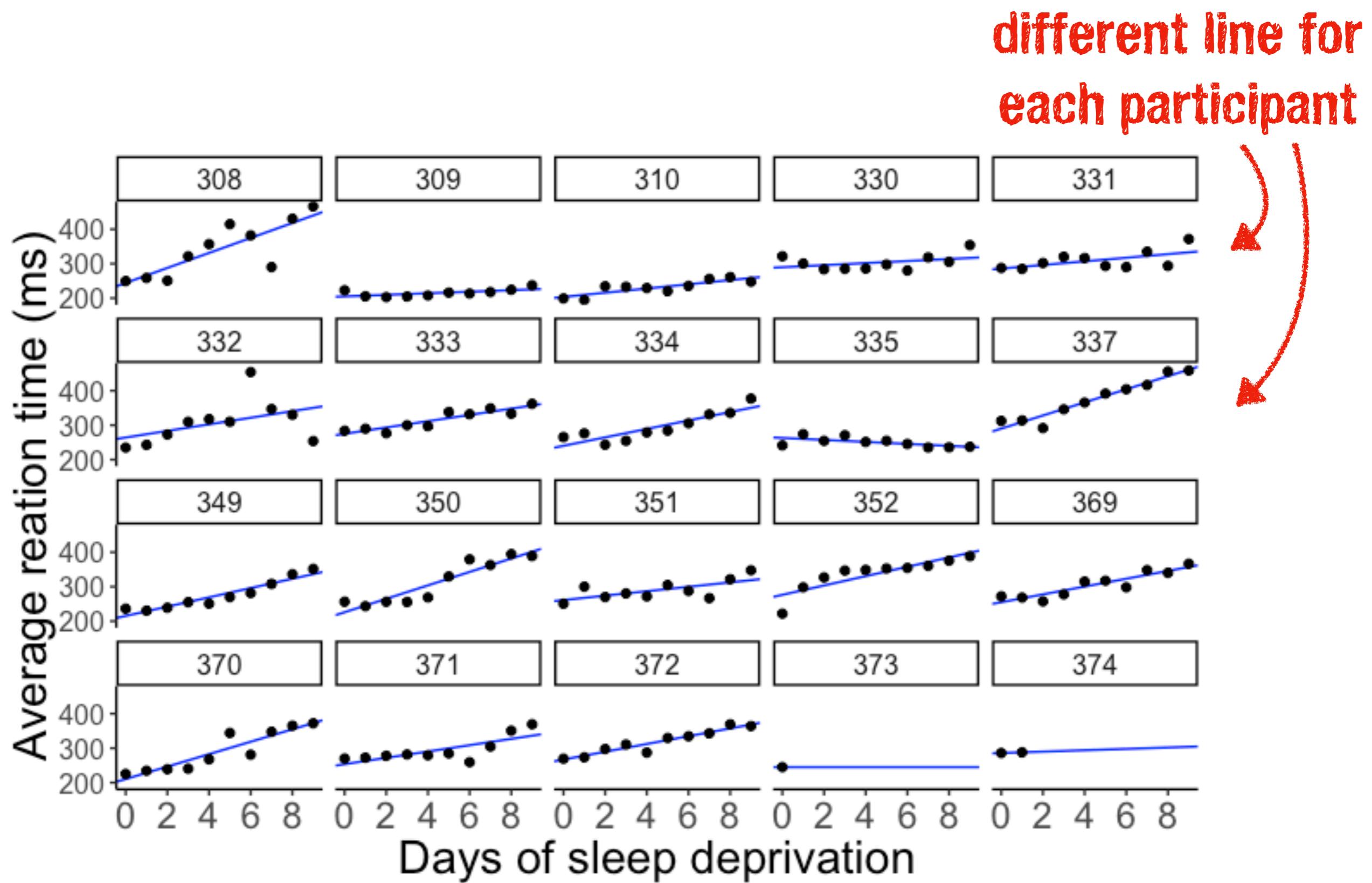
```
1 df.no_pooling = df.sleep %>%
2   group_by(subject) %>%
3   nest(data = c(days, reaction)) %>%
4   mutate(fit = map(data, ~ lm(reaction ~ days, data = .)),
5         params = map(fit, tidy)) %>%
6   unnest(c(params)) %>%
7   select(subject, term, estimate) %>%
8   complete(subject, term, fill = list(estimate = 0)) %>%
9   pivot_wider(names_from = term, values_from = estimate) %>%
10  clean_names()
```



separate intercept and
slope 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

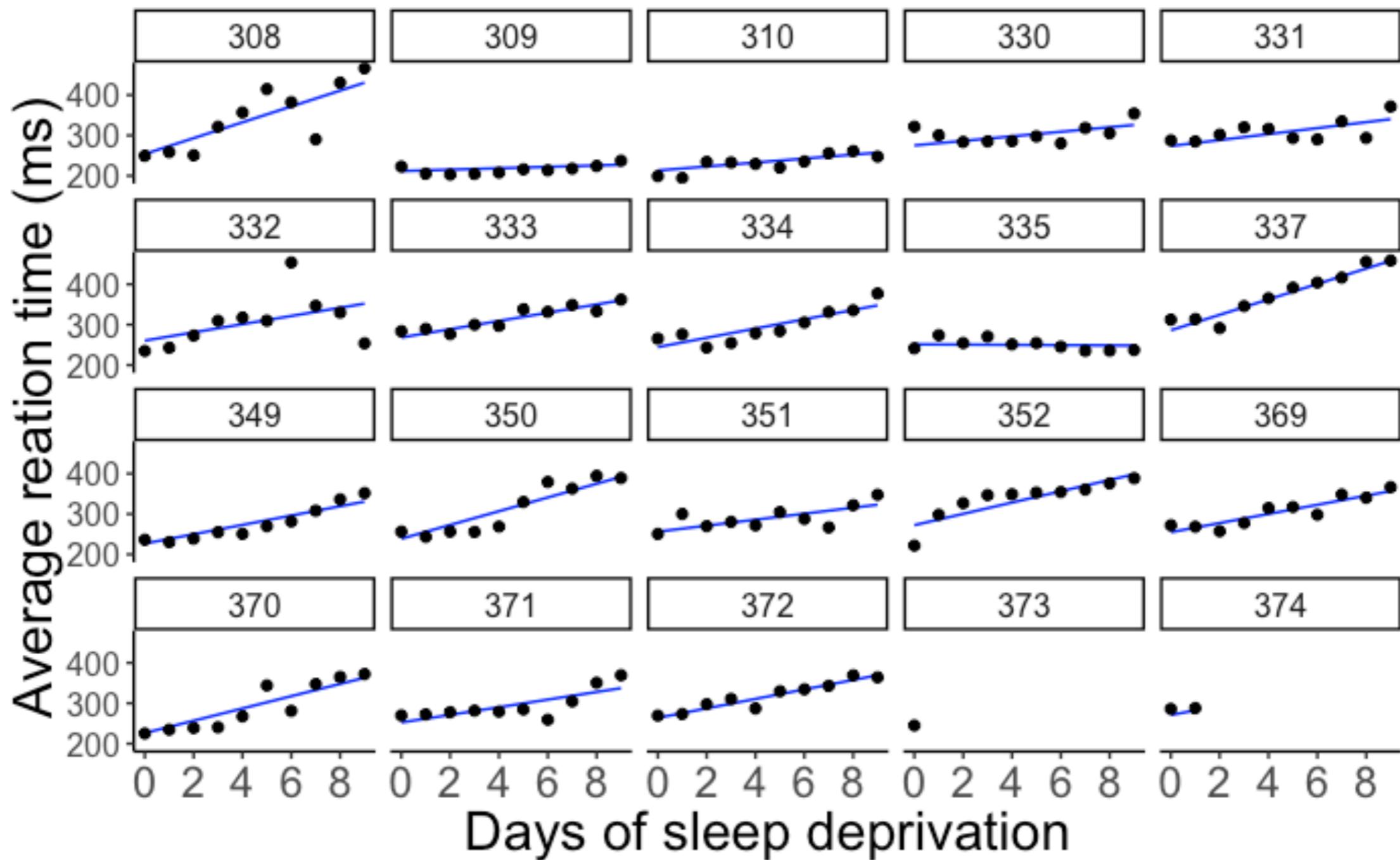
No pooling: Fit separate regressions



Partial pooling: Fit mixed effects model

intercepts and slopes differ
between participants

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

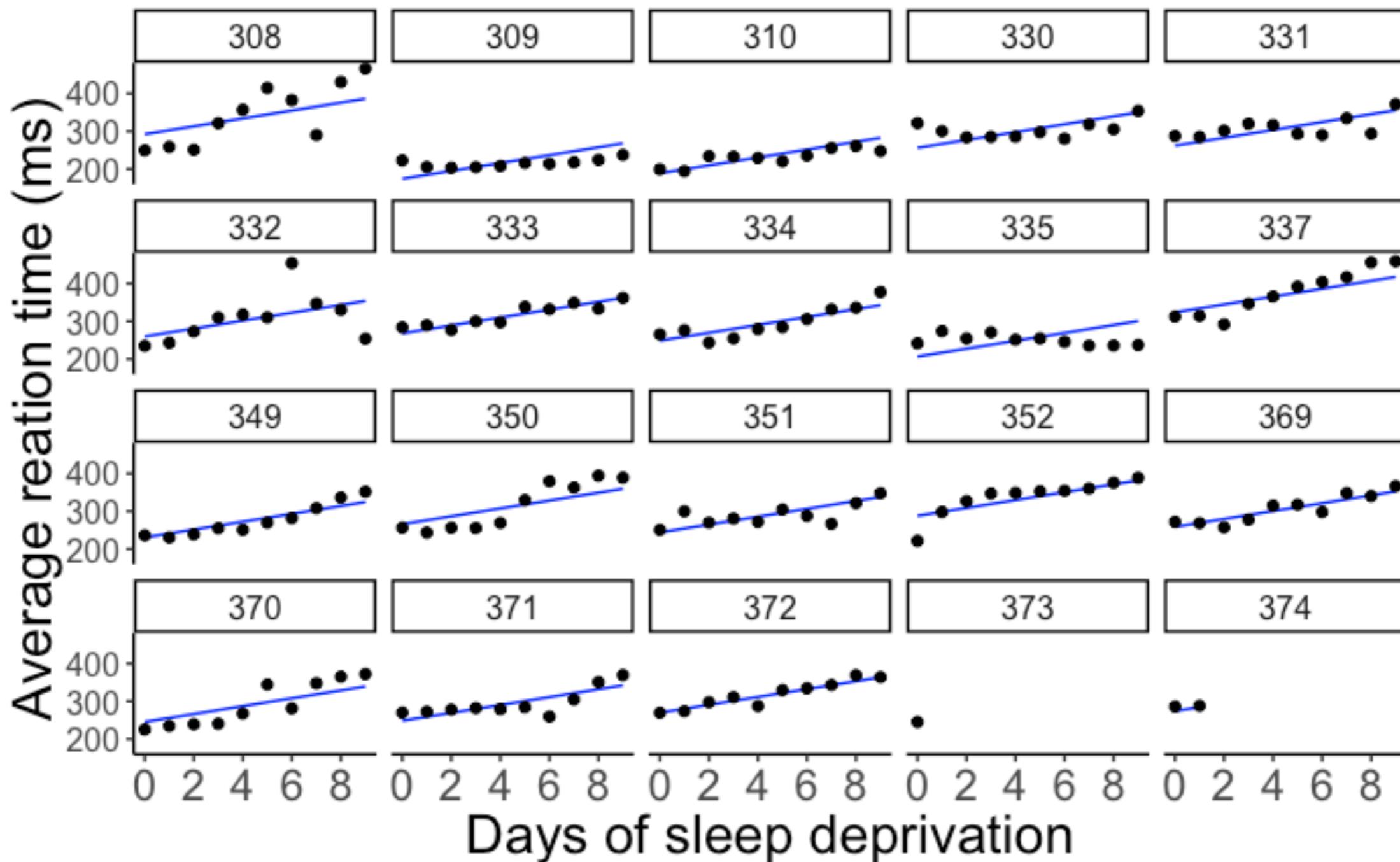


Partial pooling: Fit mixed effects model

only intercepts differ
between participants

random intercept

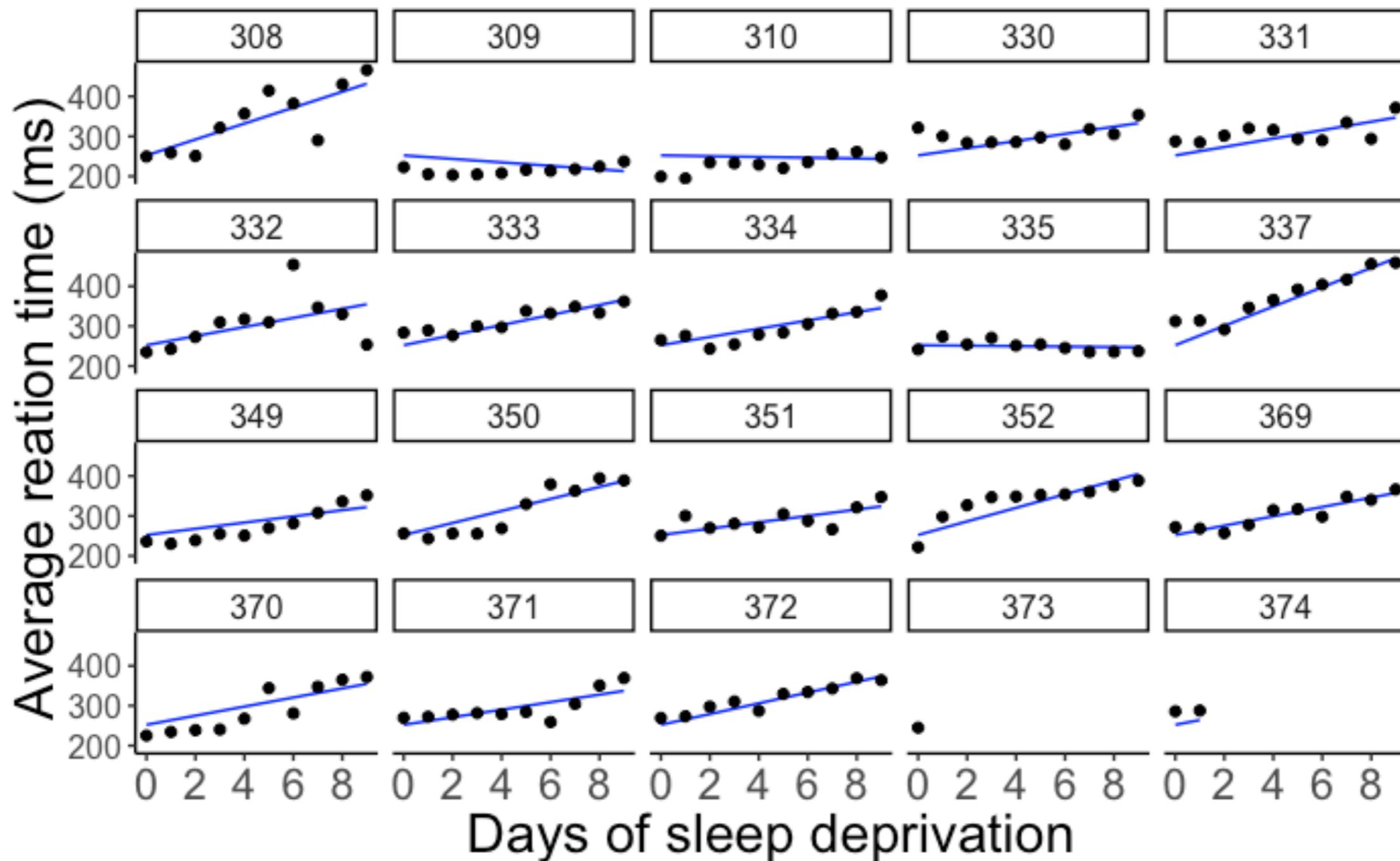
```
lmer (formula = reaction ~ 1 + days + (1 | subject),  
      data = df.sleep)
```



Partial pooling: Fit mixed effects model

only slopes differ between participants

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



Coefficients

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

`(1 | 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

`(0 + days | subject)`

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

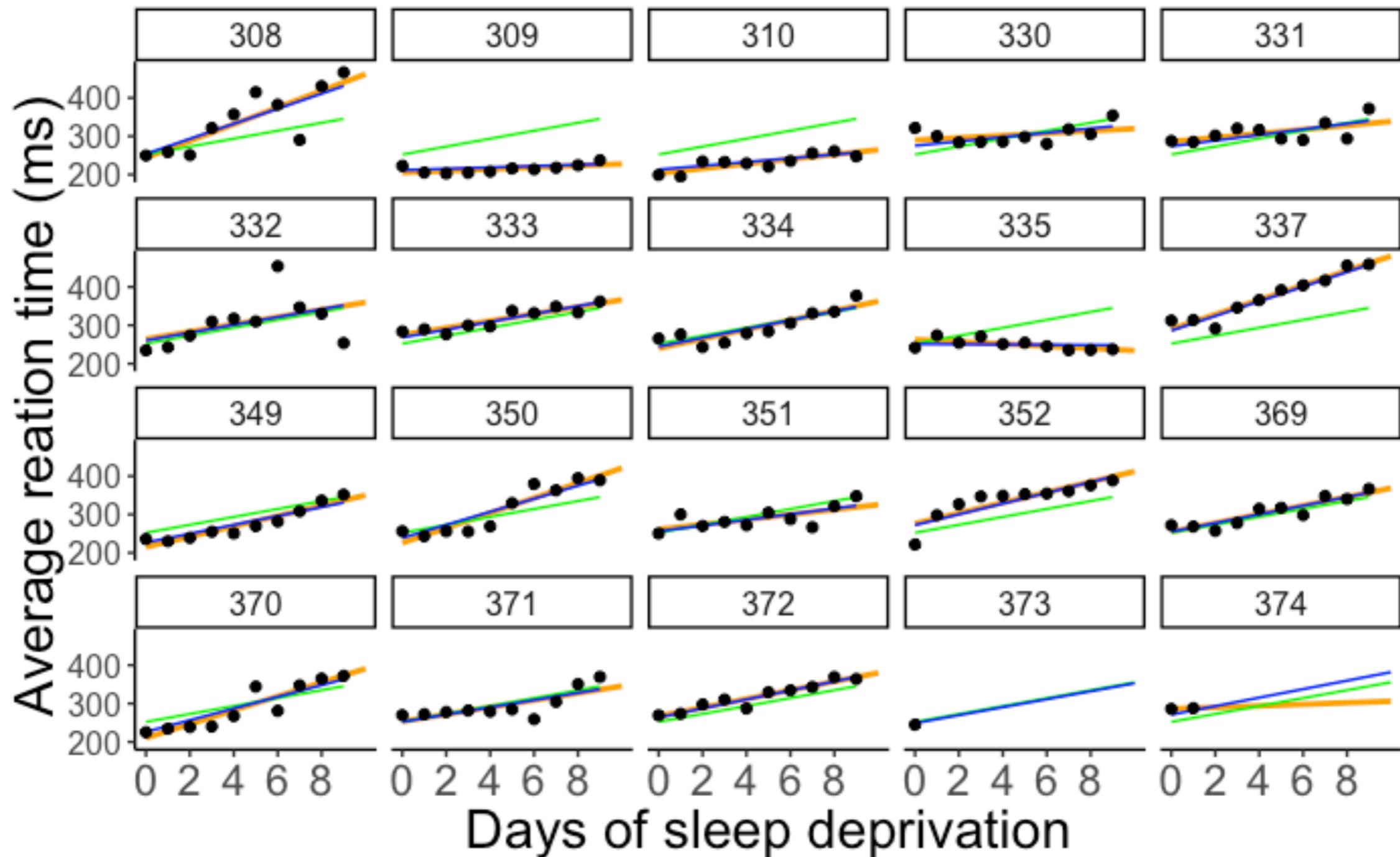
`... , (1 + days | subject)`

random intercepts and
slopes (+ correlation)

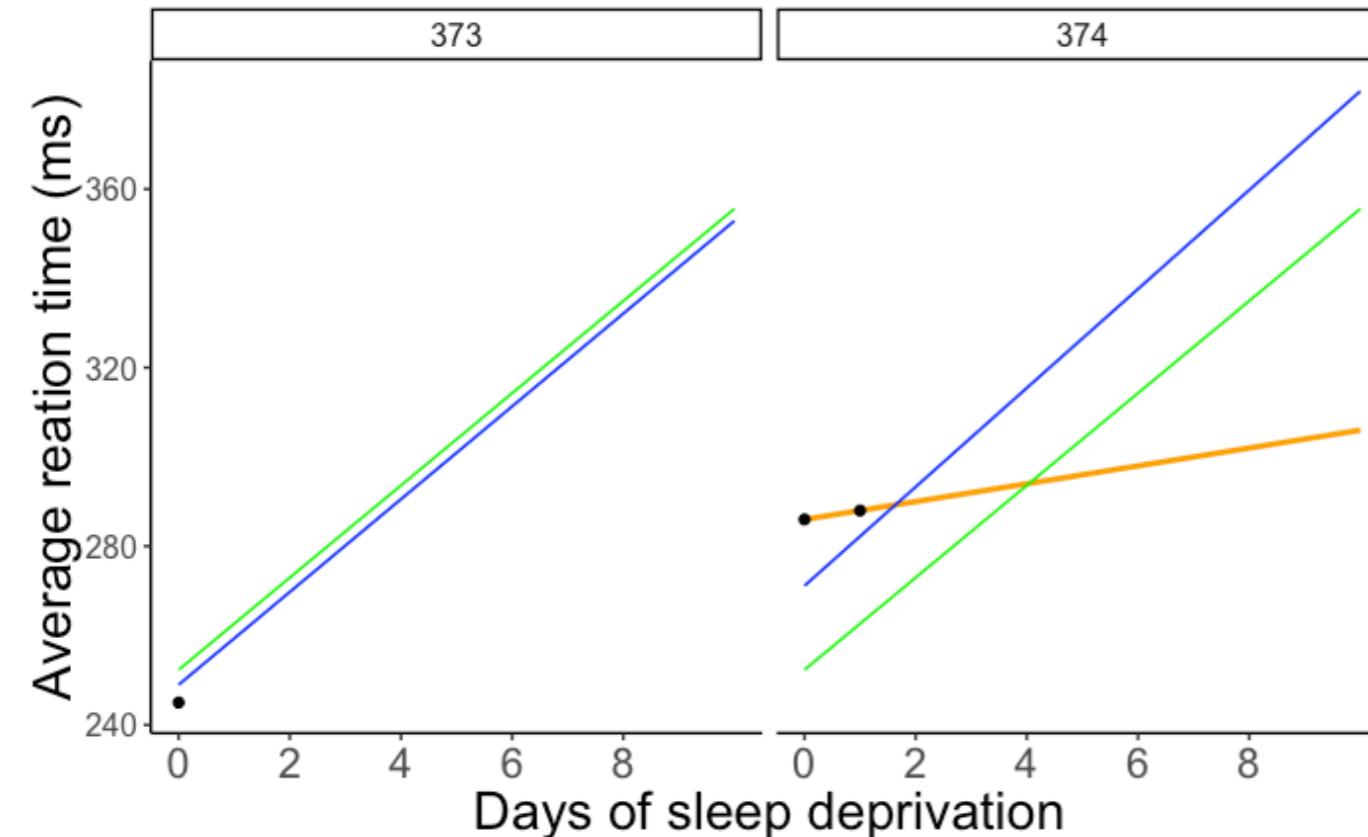
\$subject	(Intercept)	days
308	253.9479	19.6264139
309	211.7328	1.7319567
310	213.1579	4.9061843
330	275.1425	5.6435987
331	273.7286	7.3862680
332	260.6504	10.1632535
333	268.3684	10.2245979
334	244.5523	11.4837825
335	251.3700	-0.3355554
337	286.2321	19.1090061
349	226.7662	11.5531963
350	238.7807	17.0156766
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

Comparison

complete pooling
no pooling
partial pooling



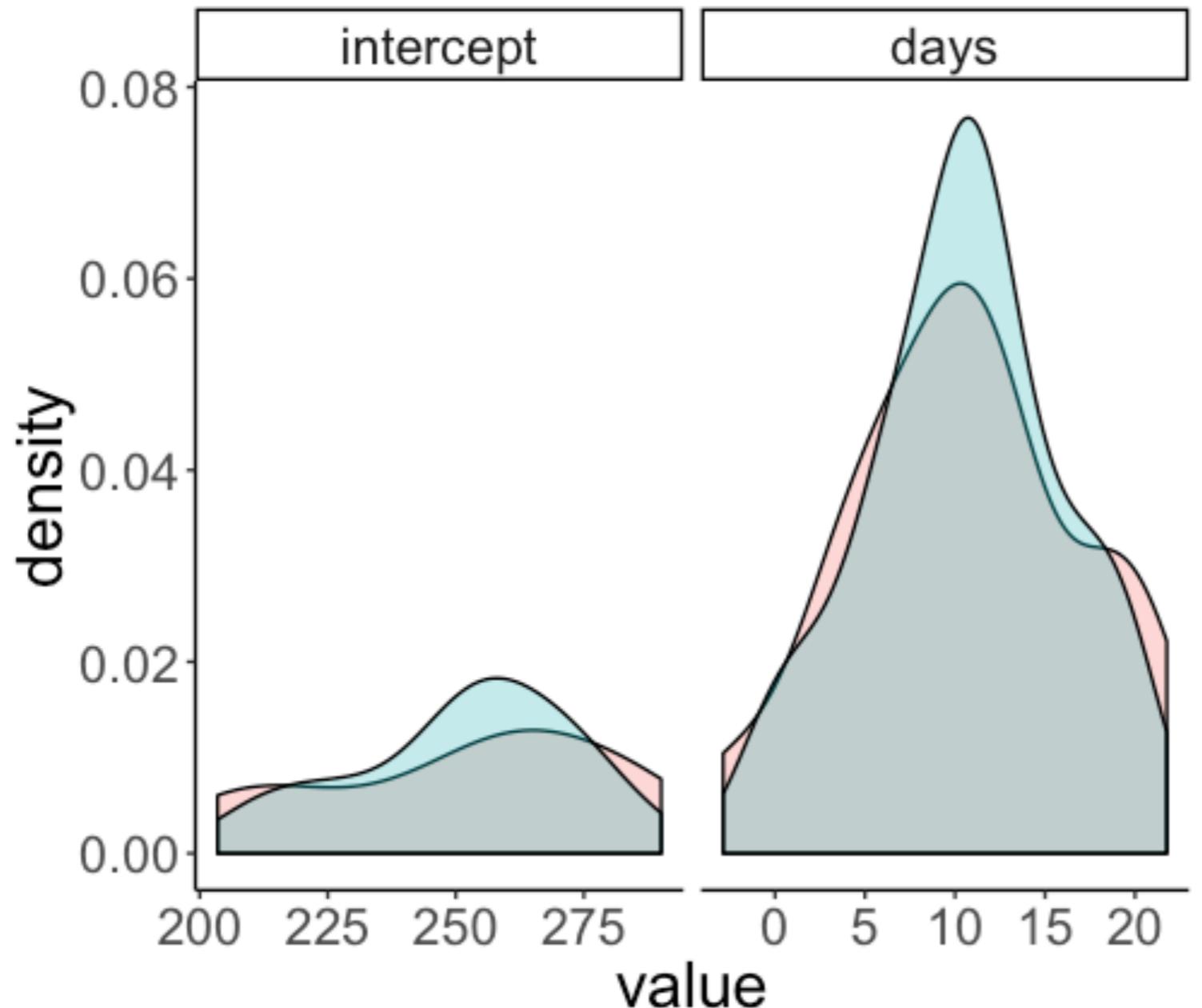
Comparison



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



method
no pooling
partial pooling

standard deviation

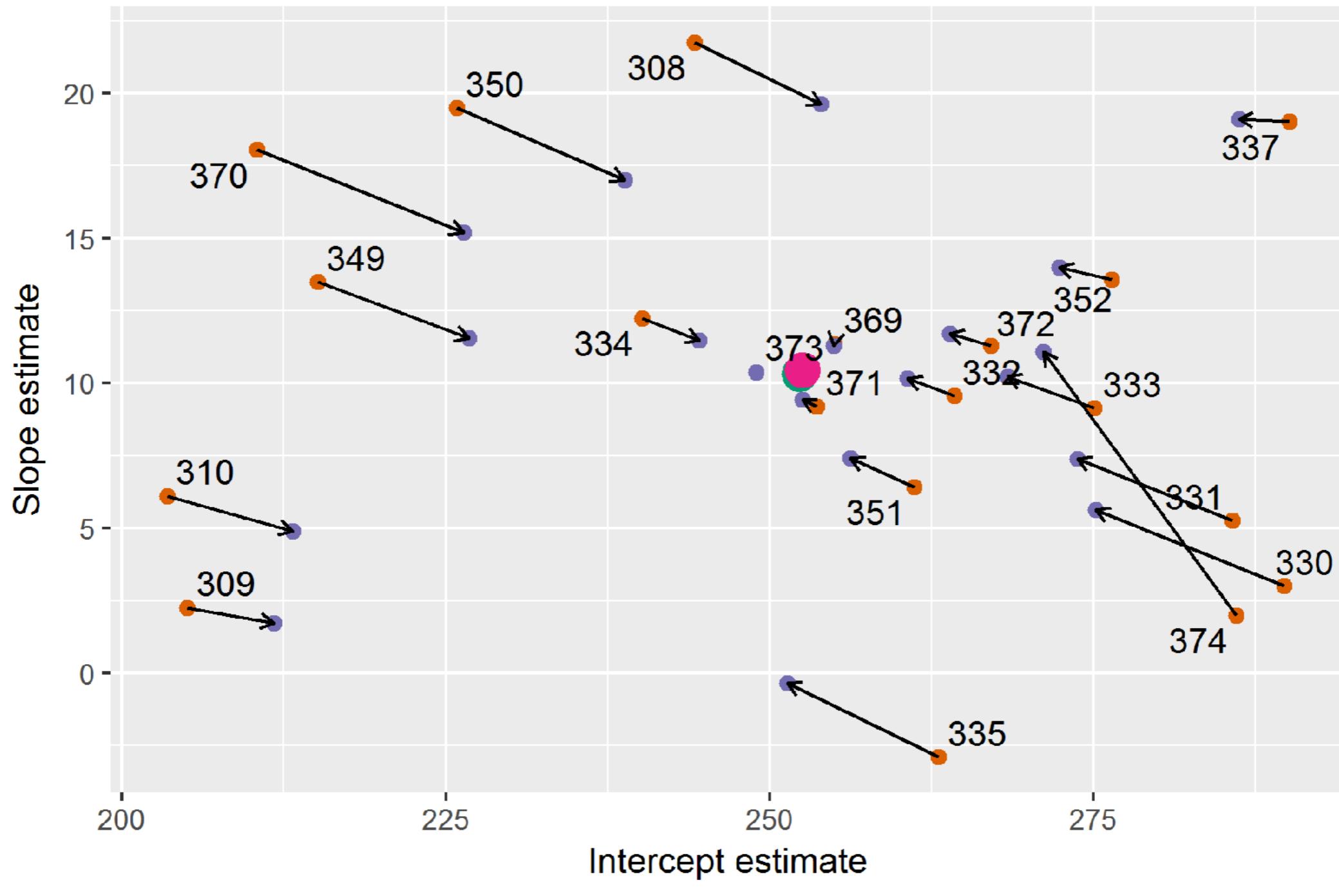
method	intercept	days
no pooling	28.95	6.56
partial pooling	21.59	5.46

variance "shrinks"



Shrinkage

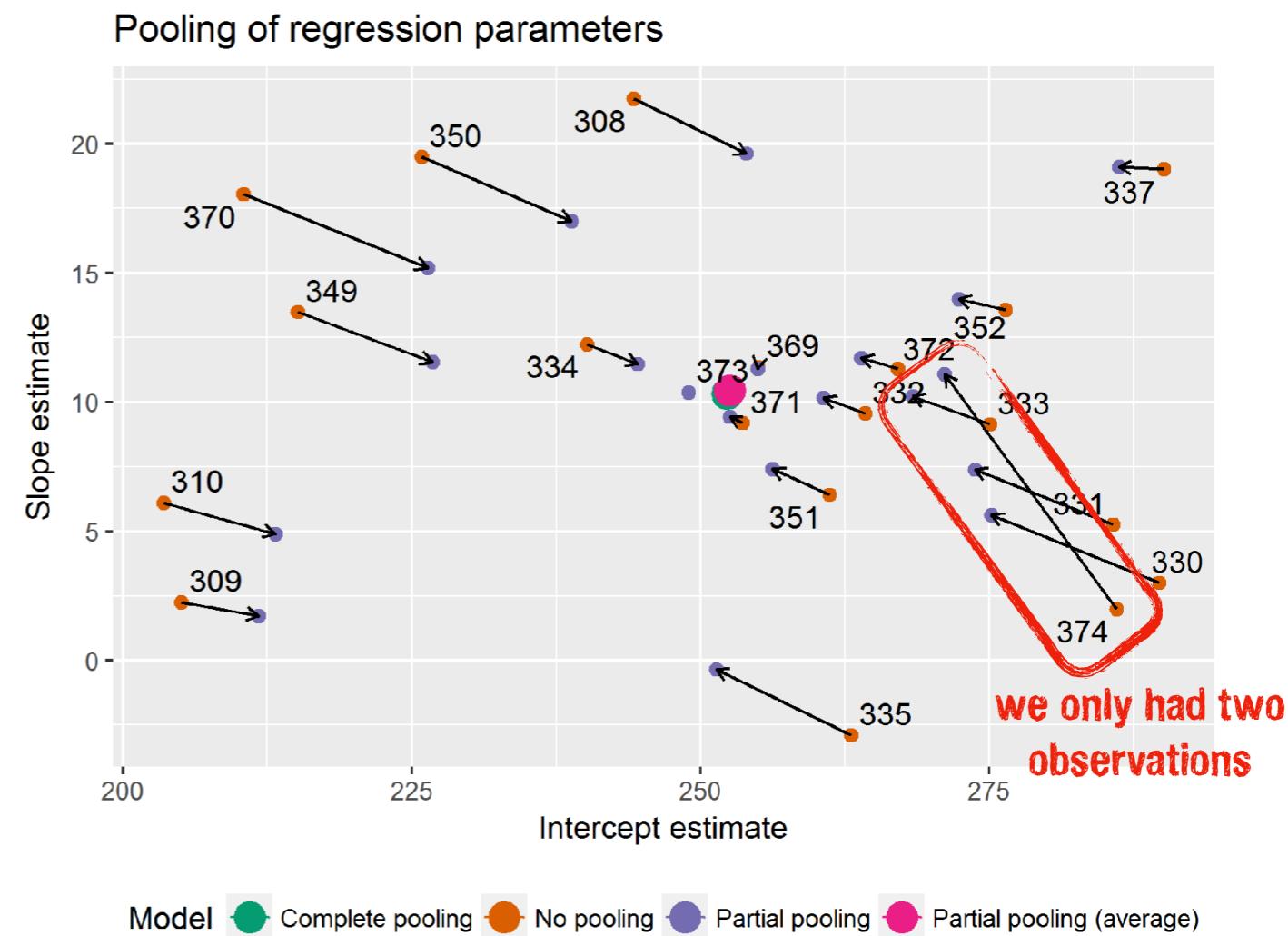
Pooling of regression parameters



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



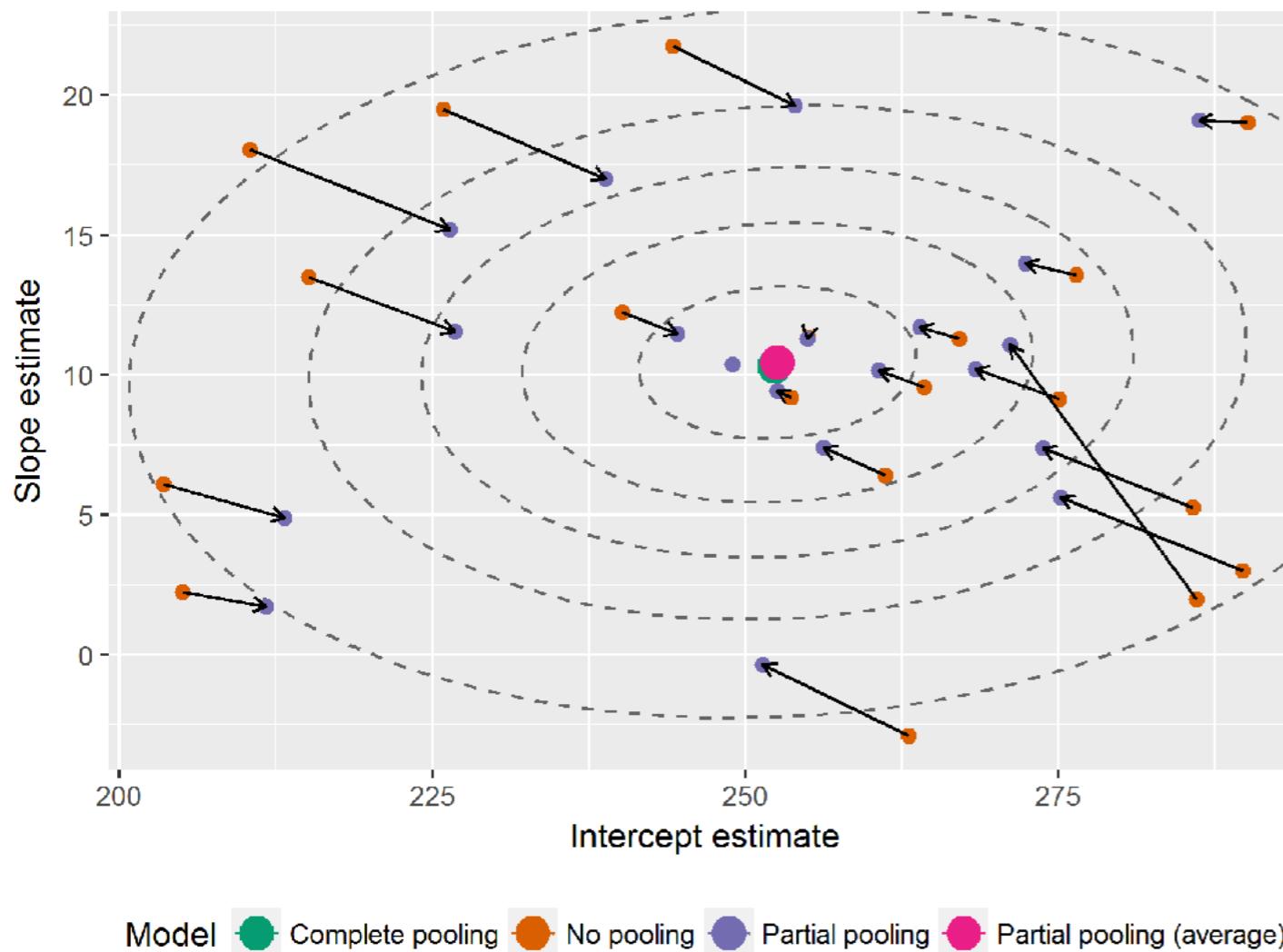
In [the lme4 book](#), Douglas Bates provides an alternative to *shrinkage*:

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

Topographic map of regression parameters

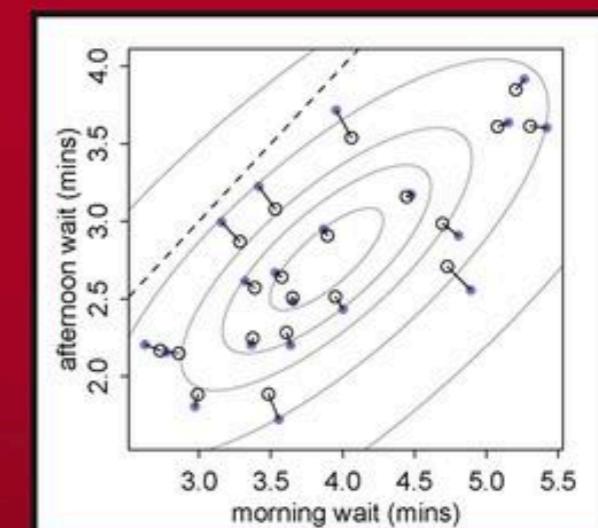


mixed effects model estimates a multi-variate Gaussian to account for (possible) correlations between intercepts and slopes

Texts in Statistical Science

Statistical Rethinking

A Bayesian Course with Examples in R and Stan

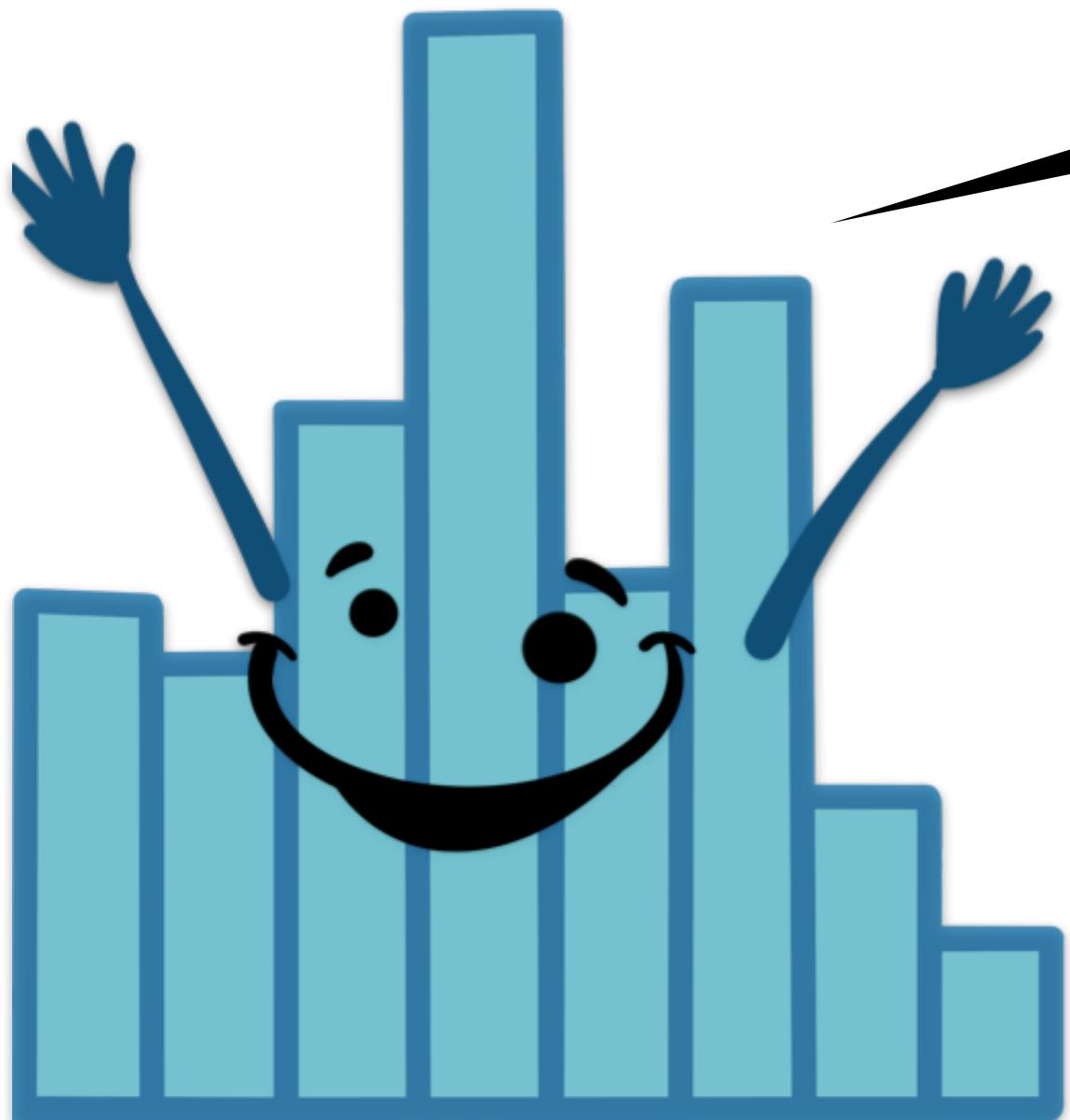


Richard McElreath

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

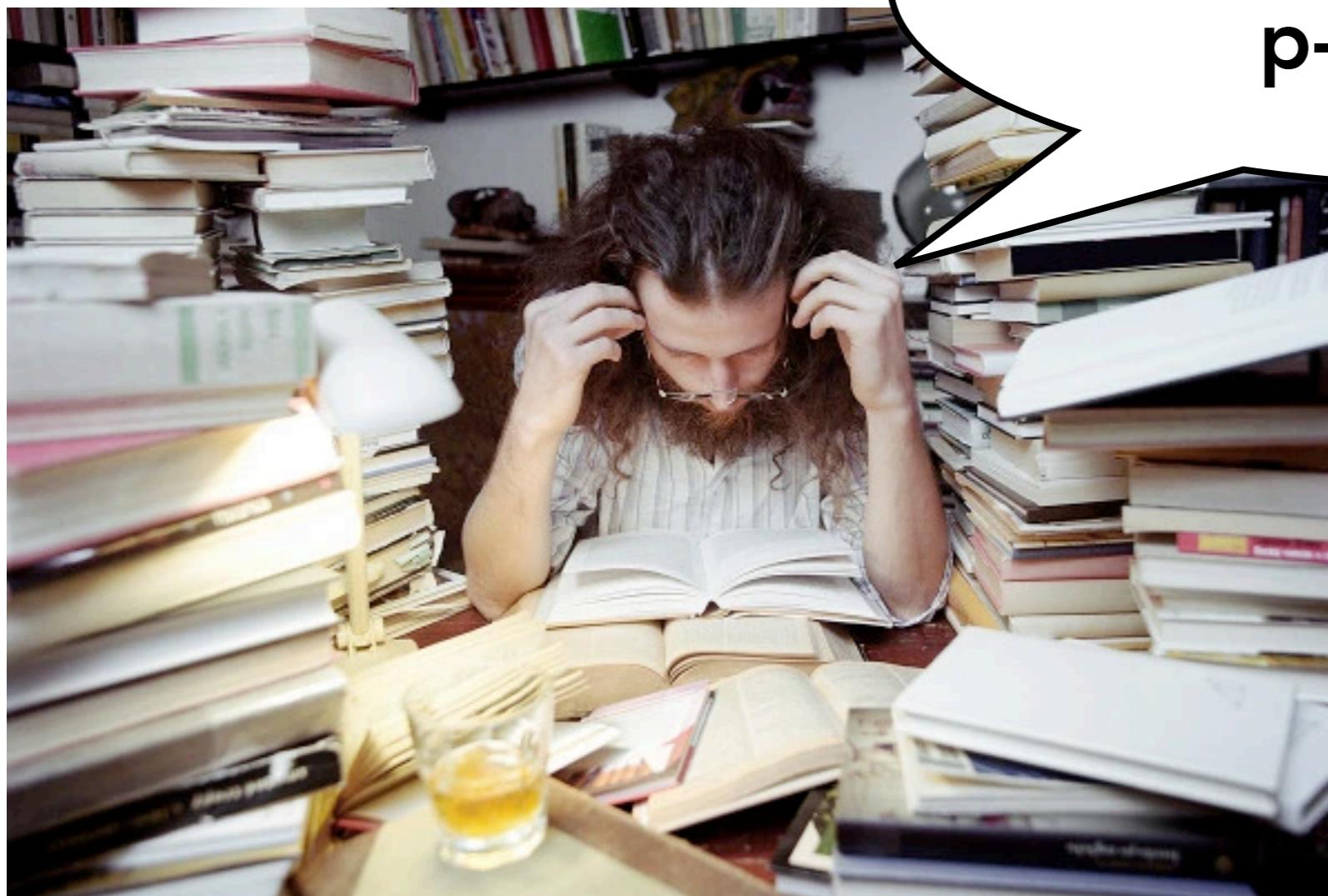
01:00

stretch break!



Getting p-values

Reviewer #2



I can't seem to find any
p-values ...

"afex" package

```
1 library("afex")
2 lmer(formula = reaction ~ 1 + days + (1 + days | subject),
3       data = df.sleep) %>%
4     summary()
```



Henrik Singmann

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
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)	582.72	24.140	
	days	35.03	5.919	0.07
Residual		649.36	25.483	

```
Number of obs: 183, groups: subject, 20
```

thanks Henrik!

```
Fixed effects:
```

	Estimate	Std. Error	df	t value	Pr(> t)						
(Intercept)	252.543	6.433	19.295	39.257	< 2e-16 ***						
days	10.452	1.542	17.163	6.778	3.06e-06 ***						

Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	'.'	0.1	' '	1

```
Correlation of Fixed Effects:
```

(Intr)	days
-0.137	

"afex" package

```
1 mixed(formula = reaction ~ 1 + days + (1 + days | subject),  
2       data = df.sleep,  
3       method = "LRT")  
4  
5 fit1 = lmer(formula = reaction ~ 1 + days + (1 + days | subject),  
6             data = df.sleep)  
7  
8 fit2 = lmer(formula = reaction ~ 1 + (1 + days | subject),  
9             data = df.sleep)  
10  
11 anova(fit1, fit2)
```



Henrik Singmann

Fitting 2 (g)lmer() models:
[...]
Mixed Model Anova Table (Type 3 tests, LRT-method)

Model: reaction ~ 1 + days + (1 + days | subject)
Data: df.sleep
Df full model: 6
Effect df Chisq p.value
1 days 1 23.60 *** <.001

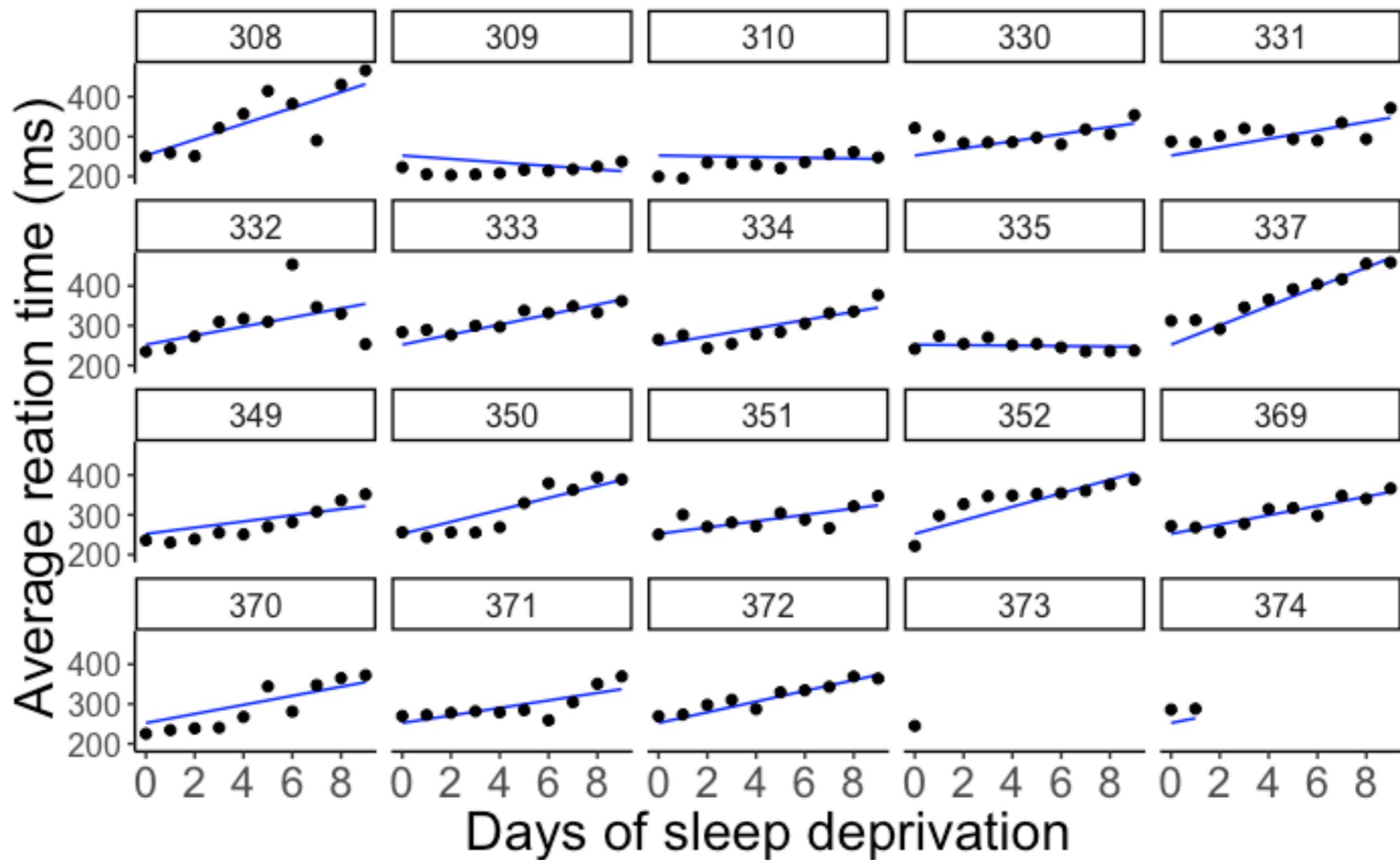
Signif. codes: 0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

refitting model(s) with `ml` (instead of REML)
Data: df.sleep
Models:
fit2: reaction ~ 1 + (1 + days | subject)
fit1: reaction ~ 1 + days + (1 + days | subject)
npar AIC BIC logLik deviance Chisq Df Pr(>Chisq)
fit2 5 1813.2 1829.3 -901.62 1803.2
fit1 6 1791.6 1810.9 -889.82 1779.6 23.602 1 1.184e-06 ***

Signif. codes: 0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Reporting results

Visualization

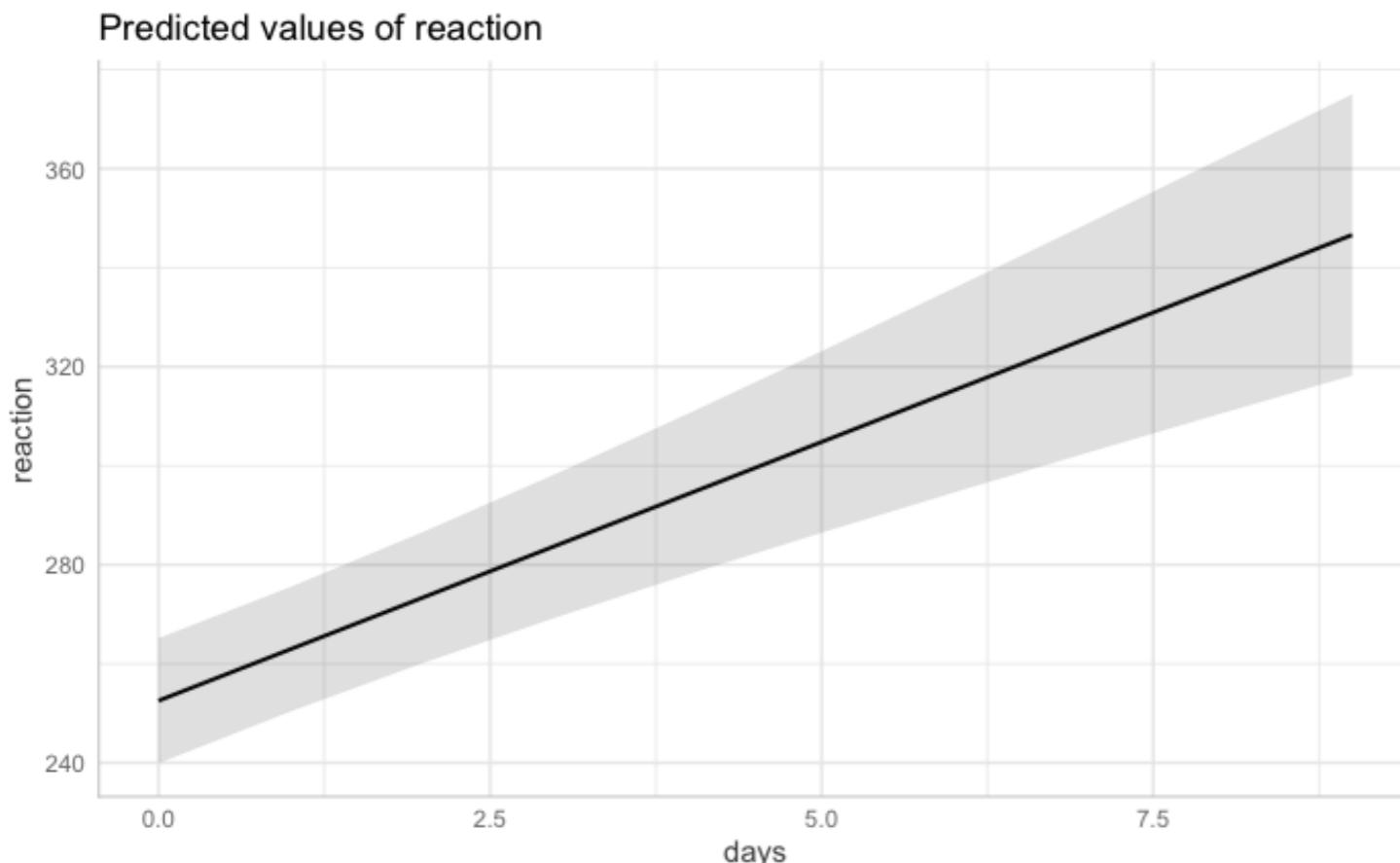


show the data together with the model predictions

Visualization



```
1 library("ggeffects")
2
3 ggpredict(model = fit.random_intercept_slope,
4            terms = "days",
5            type = "fe") %>%
6 plot()
```



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

show the (marginalized) model prediction

Reporting results

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

Reporting results

Table 2

Experiment 1 – Normality inference: Estimates of the posterior mean and 95% highest density intervals (HDIs) for the different predictors in the Bayesian regression model. Note: For the dependent variable (normality rating), 100 = abnormal and 0 = normal.

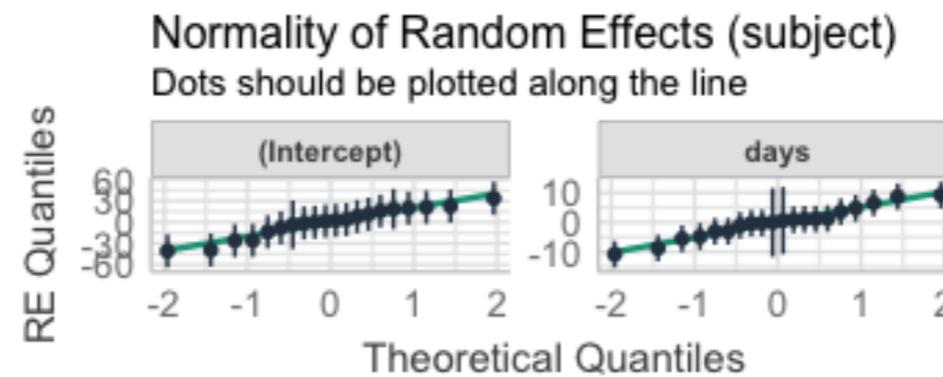
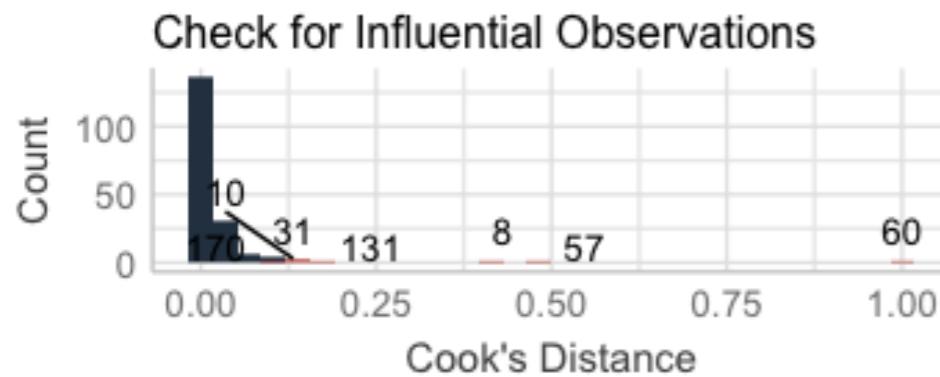
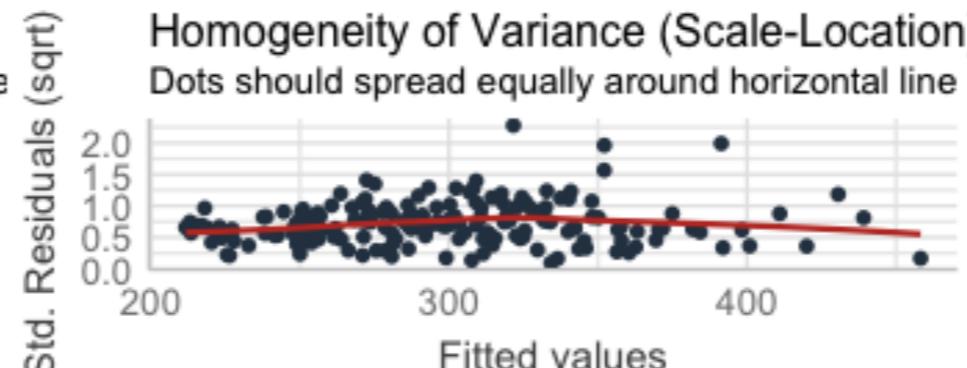
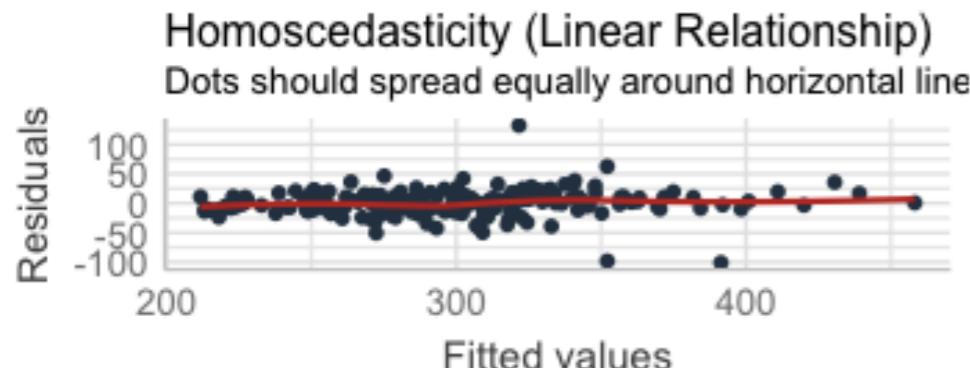
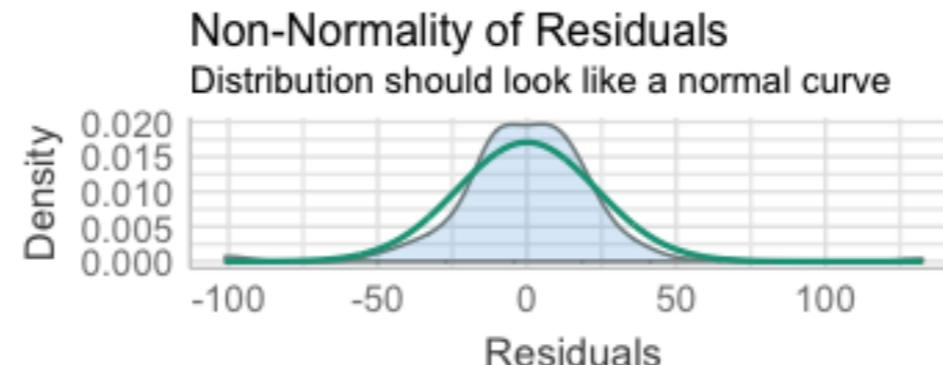
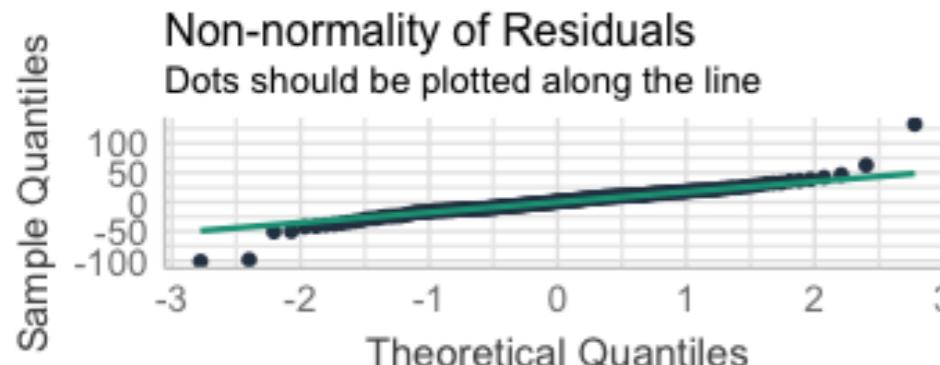
model specification: `normality rating ~ 1 + structure * norm`

term	estimate	lower 95% CI	upper 95% CI
intercept	62.83	57.57	68.11
structure	21.22	16.27	26.35
norm	-1.47	-6.37	3.83
structure:norm	-1.46	-6.41	3.53

⁷All categorical predictors were coded using sum contrasts. We adopt the convention of calling something an effect if the 95% highest density interval (HDI) of the estimated parameter in the Bayesian model excludes 0.

Useful package

```
1 lmer(formula = reaction ~ 1 + days + (1 + days | subject),  
2       data = df.sleep) %>%  
3 check_model()
```



Let's stimulate some 1mer ()s

Let's simulate an `lmer()`

```
1 # parameters
2 sample_size = 100
3 b0 = 1
4 b1 = 2
5 sd_residual = 1
6 sd_participant = 0.5
7
8 # randomly draw intercepts for each participant
9 intercepts = rnorm(sample_size, sd = sd_participant)
10
11 # generate the data
12 df.test = tibble(
13   condition = rep(0:1, each = sample_size),
14   participant = rep(1:sample_size, 2)) %>%
15   group_by(condition) %>%
16   mutate(value = b0 + b1 * condition + intercepts + rnorm(n(), sd = sd_residual))
```

participant	condition	value
1	0	0.07
1	1	3.10
2	0	1.13
2	1	4.78
3	0	-0.33
3	1	4.17
4	0	1.96
4	1	3.47
		⋮
		⋮

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

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

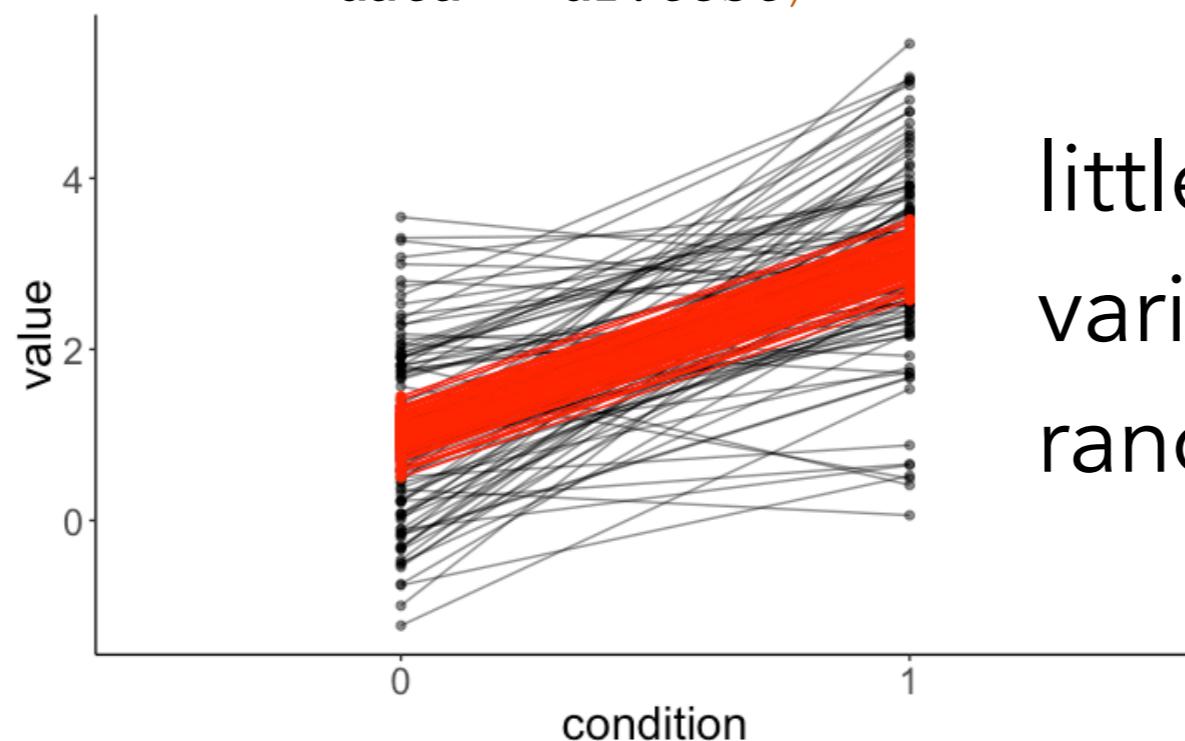
$$U_i \sim \mathcal{N}(\text{mean} = 0, \text{sd} = s_U)$$

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()`

```
1 # parameters
2 sample_size = 100
3 b0 = 1
4 b1 = 2
5 sd_residual = 1
6 sd_participant = 0.5
7
8 # randomly draw intercepts for each participant
9 intercepts = rnorm(sample_size, sd = sd_participant)
10
11 # generate the data
12 df.test = tibble(
13   condition = rep(0:1, each = sample_size),
14   participant = rep(1:sample_size, 2)) %>%
15   group_by(condition) %>%
16   mutate(value = b0 + b1 * condition + intercepts + rnorm(n(), sd = sd_residual))

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



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

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

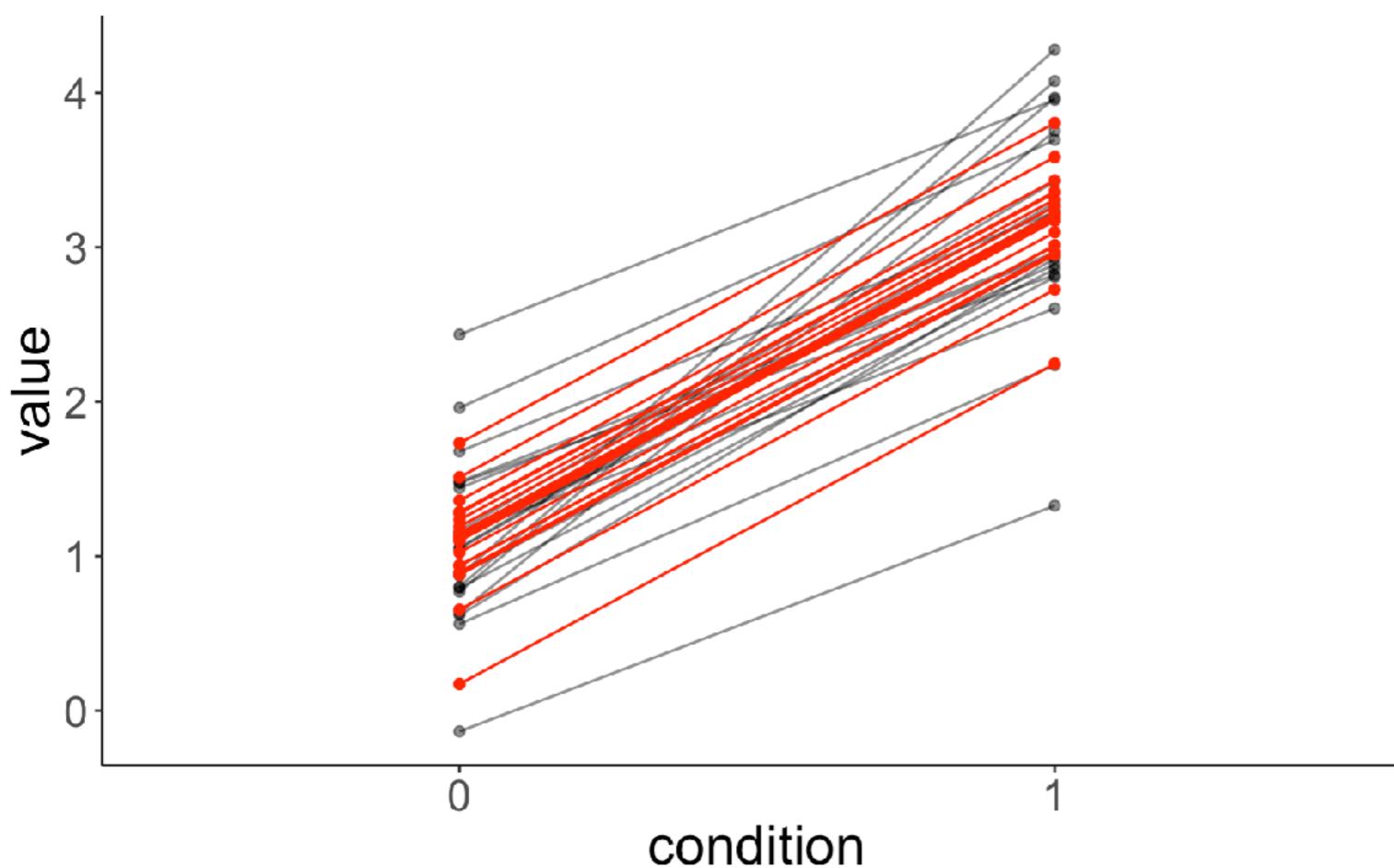
Fixed effects:
            Estimate Std. Error t value
(Intercept)  1.0166    0.1097  9.267 
condition1   2.0675    0.1444 14.317 

Correlation of Fixed Effects:
              (Intr) condition1 
condition1   -0.658
```

little interindividual
variation explained by
random intercepts ...

The effect of outliers

Outliers



```
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 

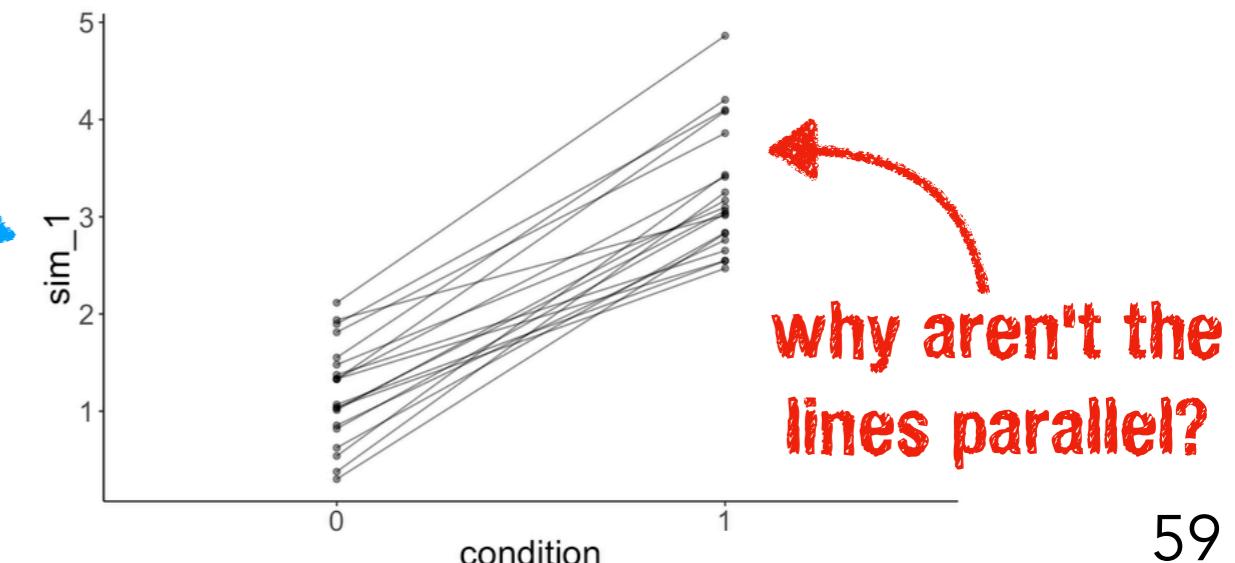
Random effects:
 Groups   Name        Variance Std.Dev. 
 participant (Intercept) 0.1702   0.4125  
 Residual           0.2270   0.4764  
Number of obs: 40, groups: participant, 20

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

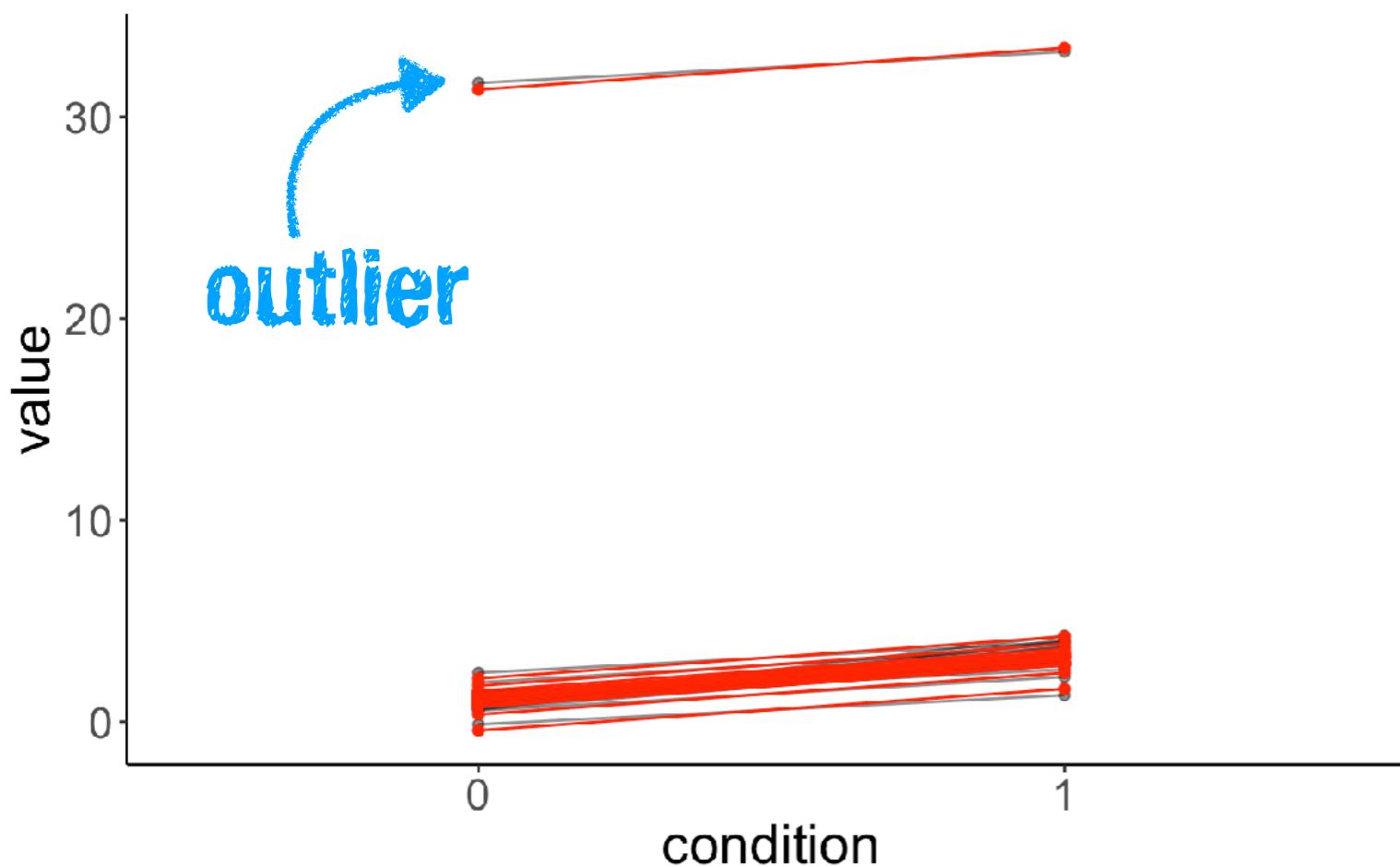
Correlation of Fixed Effects:
          (Intr) condition1 
condition1 -0.535
```

```
1 # fit model
2 fit.test = lmer(formula = value ~ 1 + condition + (1 | participant),
3                  data = df.test)
4
5 # simulate data
6 fit.test %>%
7   simulate()
```

simulated data



Outliers



```
1 # fit model
2 fit.test = lmer(formula = value ~ 1 + condition + (1 | participant),
3                  data = df.test)
4
5 # simulate data
6 fit.test %>%
7   simulate()
```

simulated data

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

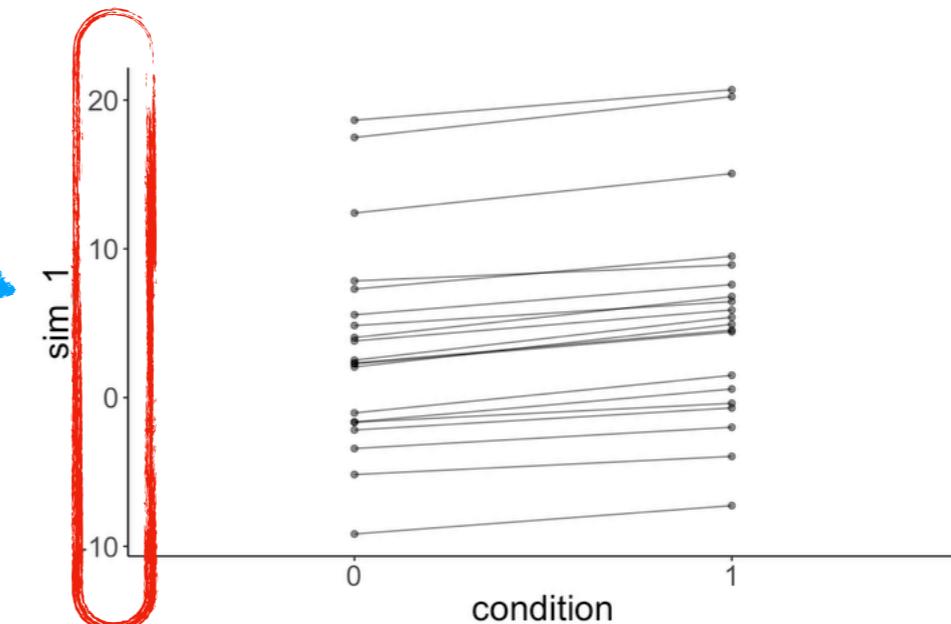
REML criterion at convergence: 171.7

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

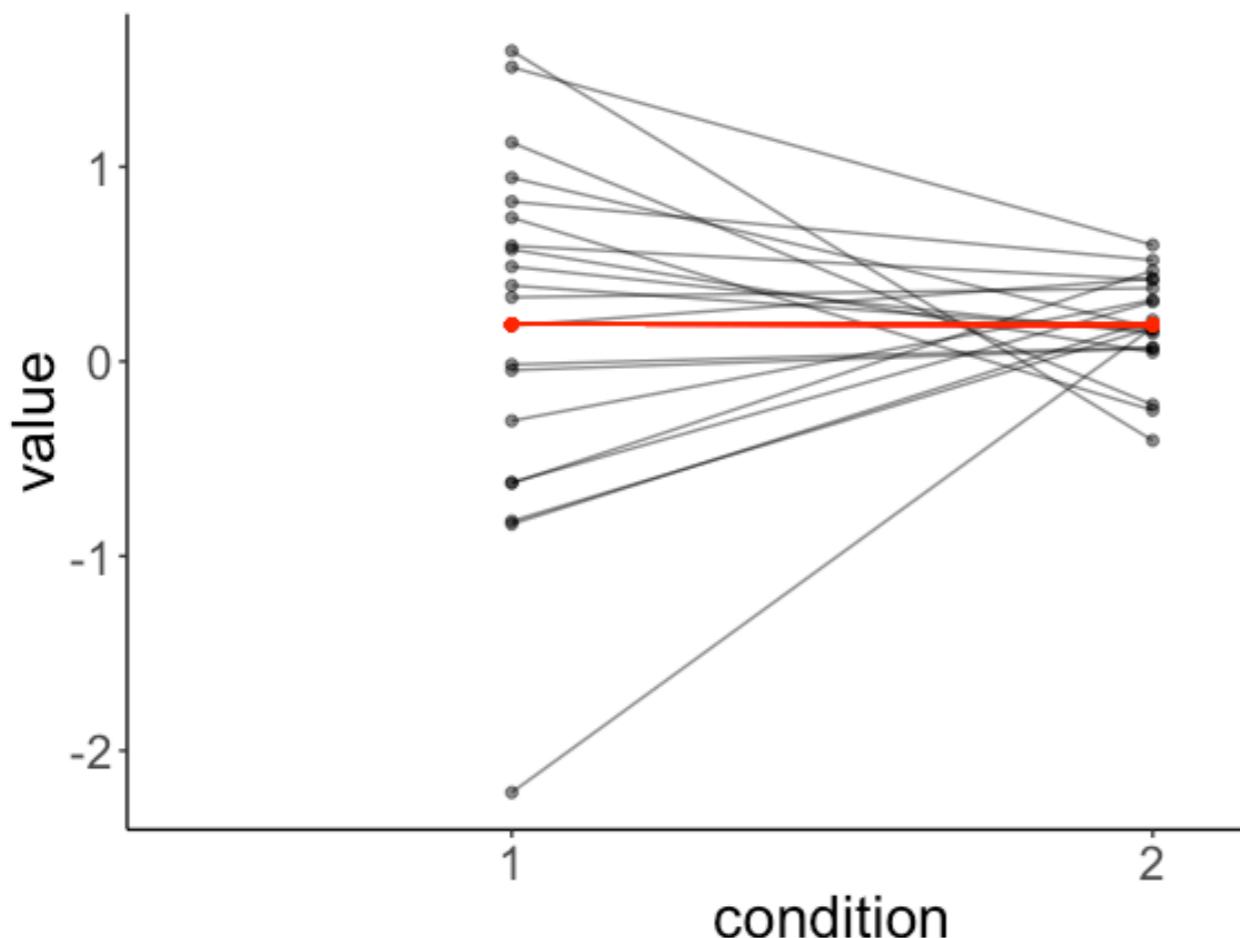
Fixed effects:
            Estimate Std. Error t value
(Intercept)  2.5920    1.5236  1.701
condition1   2.0726    0.1507 13.758

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



Heterogeneity of variance

Different slopes



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

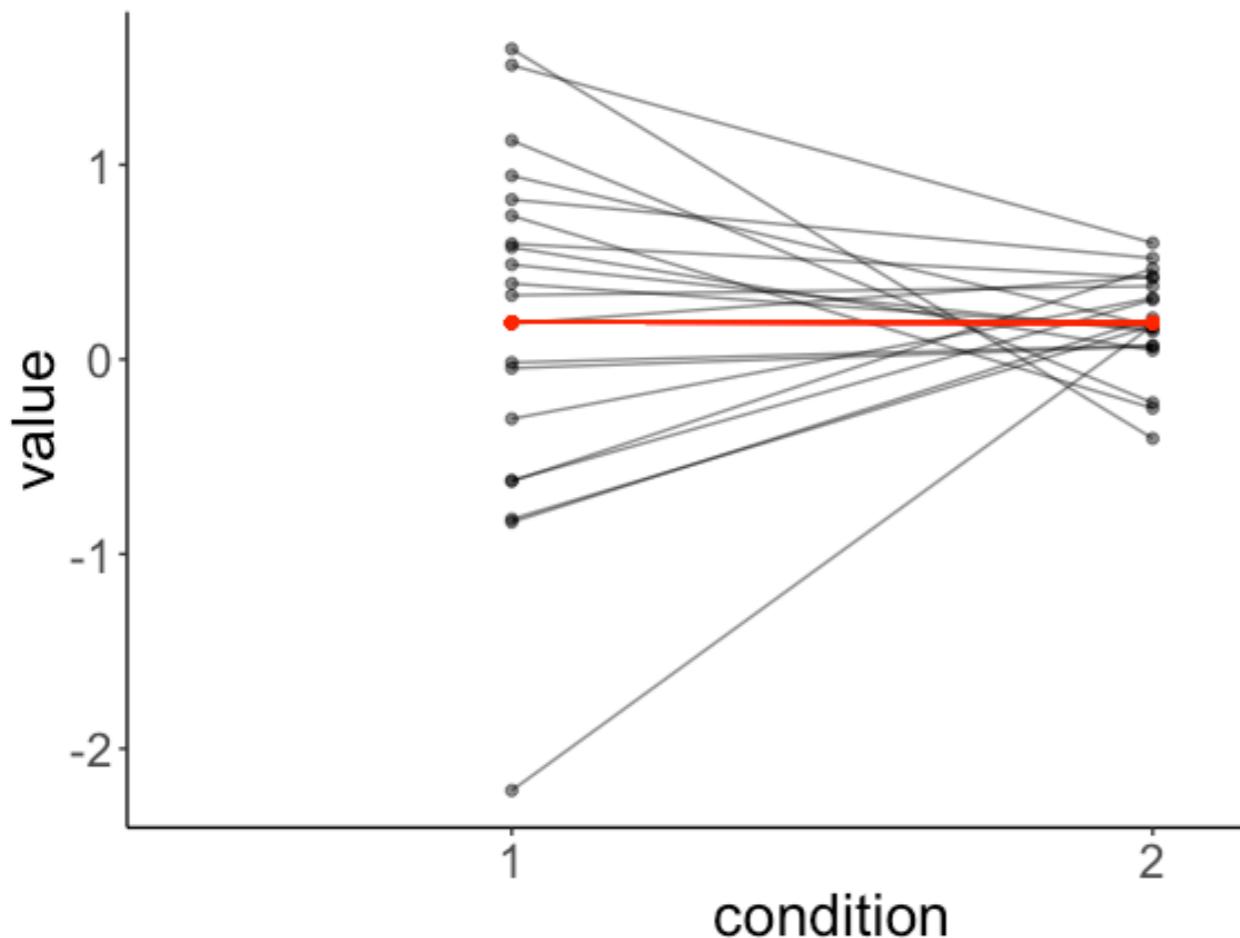
Random effects:
Groups      Name        Variance Std.Dev. 
participant (Intercept) 0.0000   0.0000  
Residual           0.4512   0.6717  
Number of obs: 40, groups: participant, 20

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 
condition2 -0.707 
convergence code: 0 
singular fit
```

clearly there are interindividual differences though!?

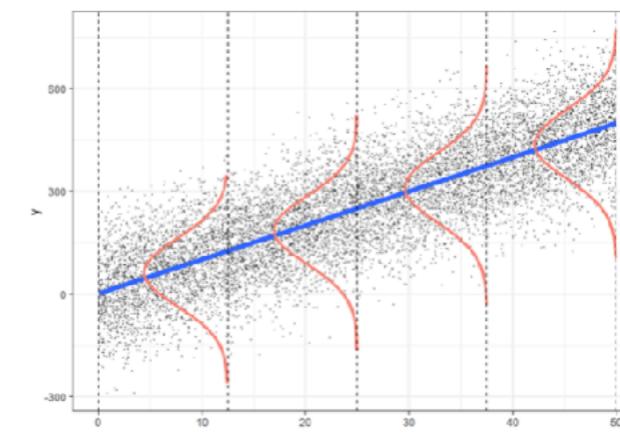
Different slopes



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



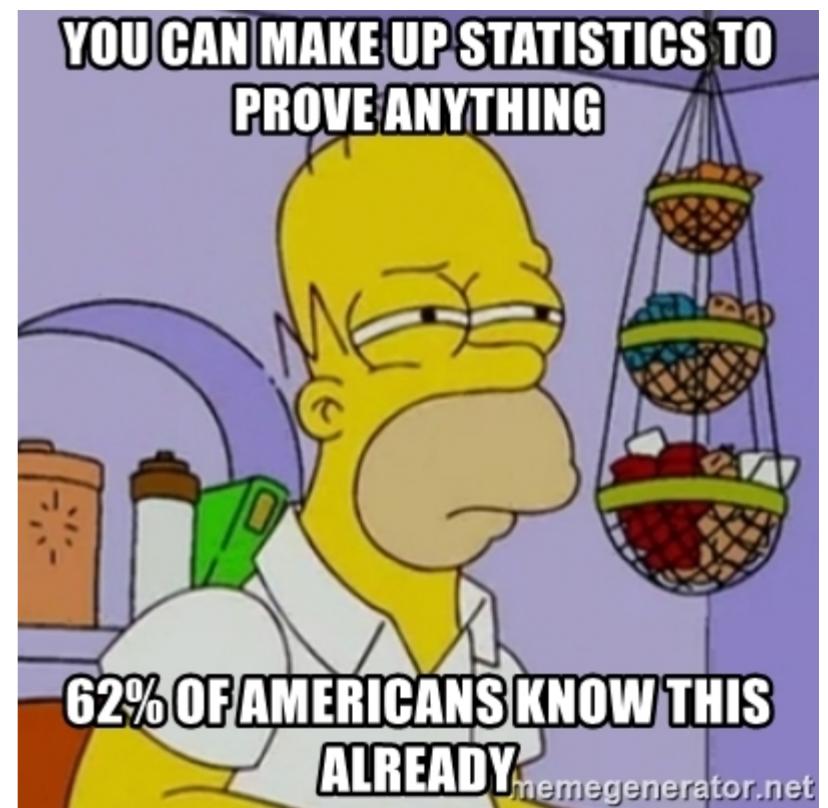
```
1 # fit model  
2 lmer(formula = value ~ 1 + condition + (1 + condition | participant),  
3       data = df.test)
```



random slope

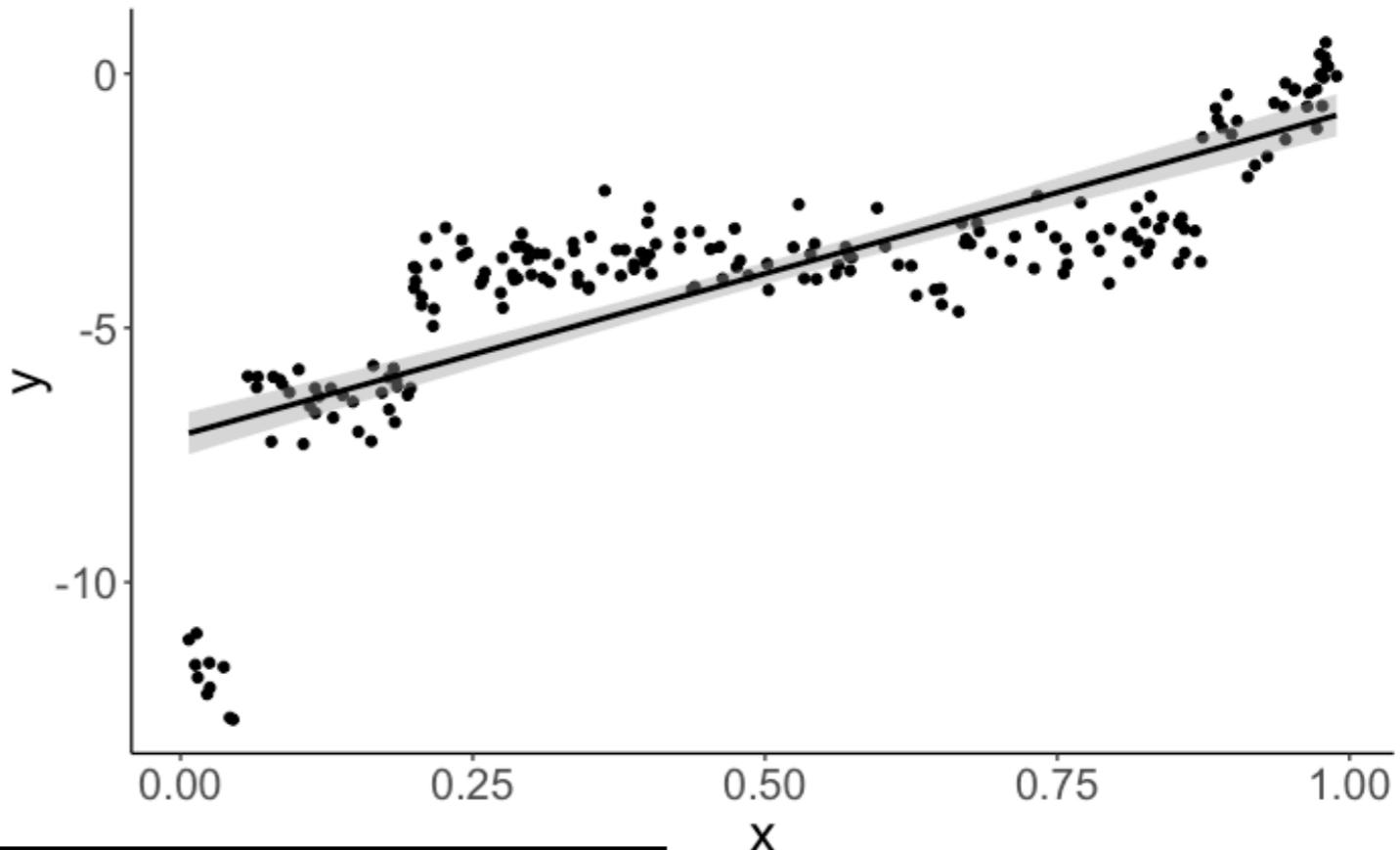
won't work

```
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
```



Simpsons Paradox

Simpson's paradox



```
1 lm(formula = y ~ x,  
2     data = df.simpson) %>%  
3     summary()
```

```
Call:  
lm(formula = y ~ x, data = df.simpson)
```

```
Residuals:
```

Min	1Q	Median	3Q	Max
-5.8731	-0.6362	0.2272	1.0051	2.6410

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-7.1151	0.2107	-33.76	<2e-16 ***
x	6.3671	0.3631	17.54	<2e-16 ***

```
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 1.55 on 198 degrees of freedom
```

```
Multiple R-squared: 0.6083, Adjusted R-squared: 0.6064
```

```
F-statistic: 307.5 on 1 and 198 DF, p-value: < 2.2e-16
```

positive relationship
between x and y

Simpson's paradox

```
1 lmer(formula = y ~ 1 + x + (1 | participant),  
2       data = df.simpson) %>%  
3   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

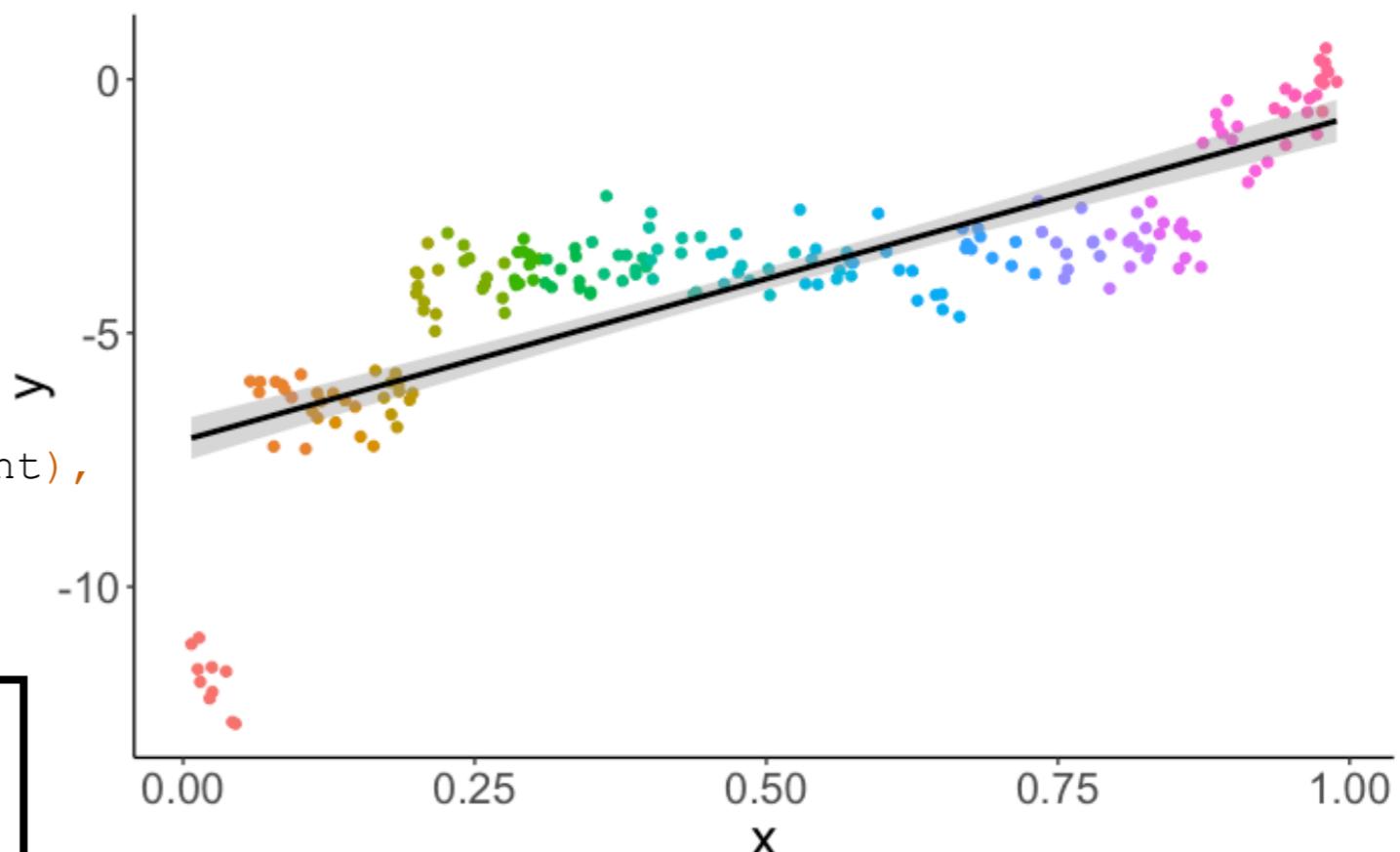
```
Number of obs: 200, groups: participant, 20
```

```
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**

Simpson's paradox

```
1 lmer(formula = y ~ 1 + x + (1 | participant),  
2       data = df.simpson) %>%  
3 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

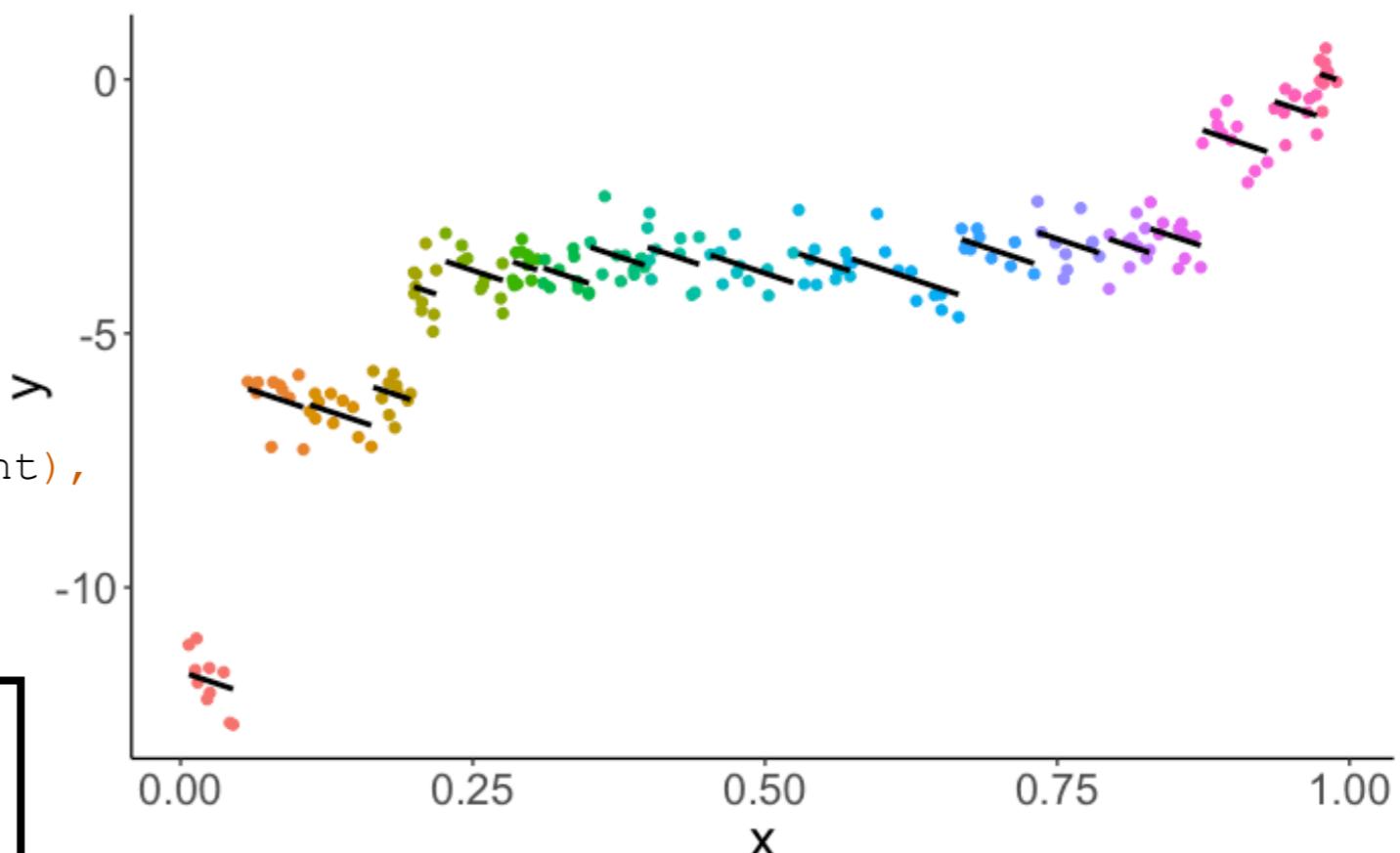
```
Number of obs: 200, groups: participant, 20
```

```
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:
```

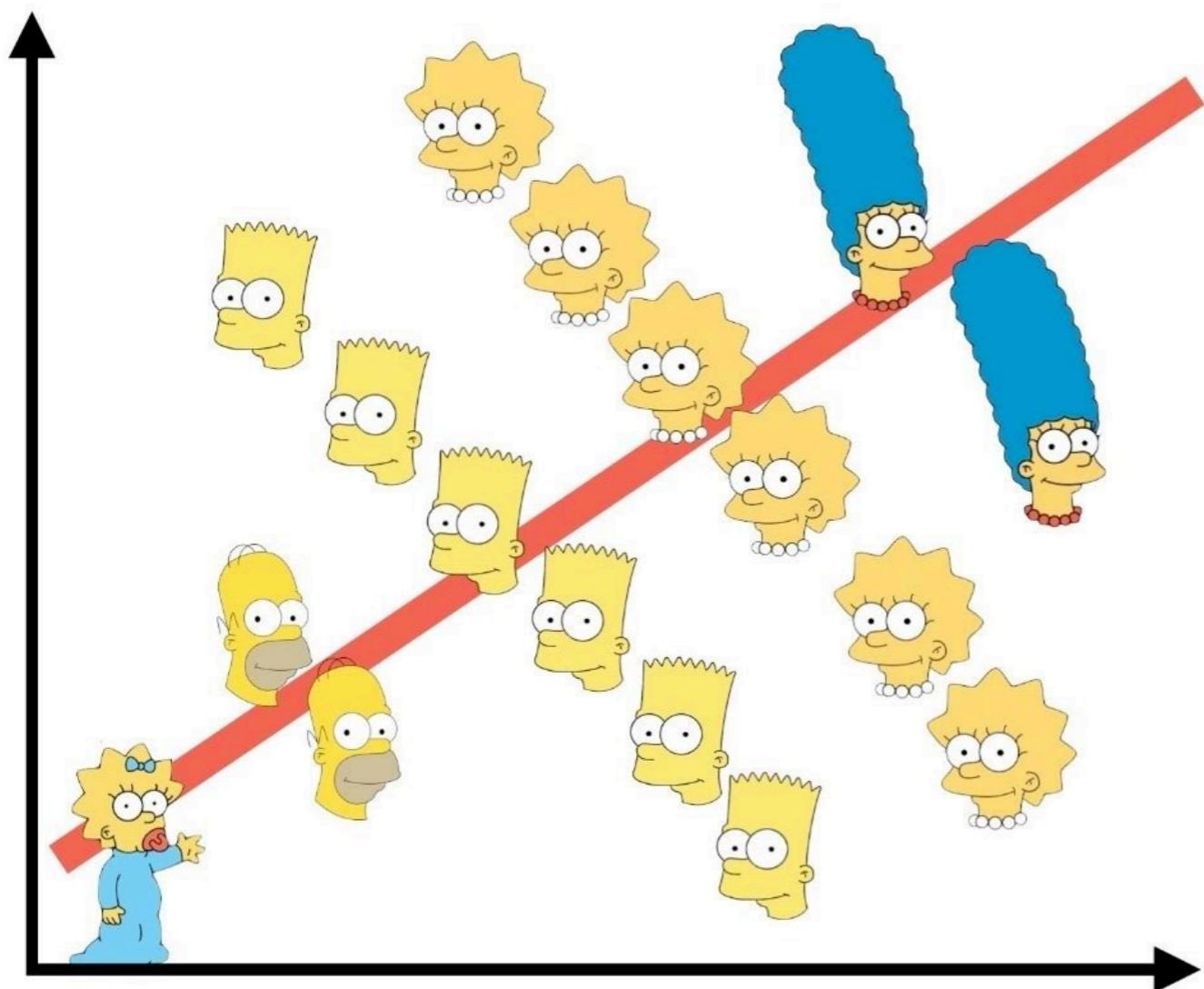
(Intr)	x
-0.621	



**negative (!)
relationship between
x and y**

**(once we take into
account individual
differences)**

Simpson's paradox



- when the relationship between two variables changes strongly after conditioning on a grouping variable (e.g. individual participants)
- interesting real world cases
- **google it!**

Simpson's paradox

UC Berkeley gender bias (1973)

	Men		Women	
	Applicants	Admitted	Applicants	Admitted
Total	8442	44%	4321	35%

overall men are more likely to be admitted

Department	Men		Women	
	Applicants	Admitted	Applicants	Admitted
A	825	62%	108	82%
B	560	63%	25	68%
C	325	37%	593	34%
D	417	33%	375	35%
E	191	28%	393	24%
F	373	6%	341	7%

men not more likely to be admitted when broken down by department

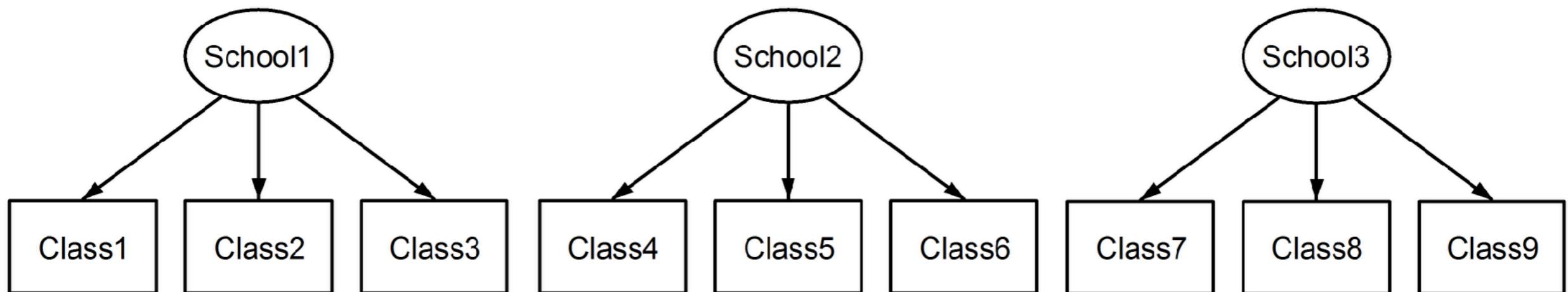
In fact, the pooled and corrected data showed a "small but statistically significant bias in favor of women."

women applied to more competitive departments

Understanding lmer() syntax

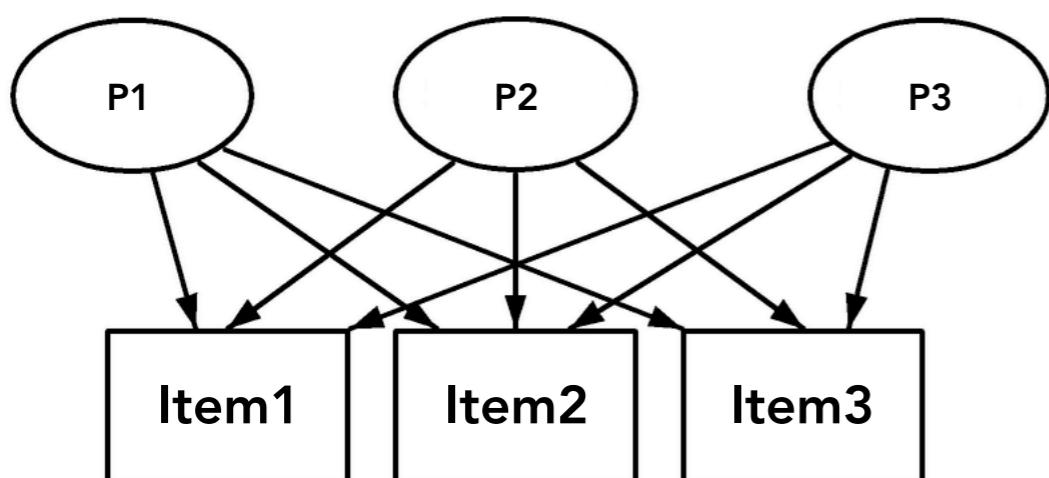
Multi-level models

nested $(1 | \text{School}/\text{Class})$



each class only appears within one school

crossed $(1 | \text{participant}) + (1 | \text{item})$



**each participant
rates each item**

`lmer()` syntax summary

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

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

Content/Analysis Steps	1
1. Before data collection:	
Power/ design/ planning/ sample size	3
1.1. Power analysis	3
1.2. Sensitivity analysis	4
1.3. Sequential sampling with stopping rules	5
1.4. More readings	5
2. Preparing data	6
2.1. Categorical variables	6
2.2. Continuous variables	6
3. Running the model	7
3.1. Model specification and random effects	7
3.2. Addressing convergence warnings	7
3.2.1. Convergence warnings in R's lme4	7
3.2.2. Or we choose the Bayesian approach	9
3.2.3. MixedModels in Julia	9

also available
in Files >
papers on
Canvas

[http://decision-lab.org/wp-content/uploads/2020/07/
SOP_Mixed_Models_D2P2_v1_0_0.pdf](http://decision-lab.org/wp-content/uploads/2020/07/SOP_Mixed_Models_D2P2_v1_0_0.pdf)

2. Preparing data

2.1. Categorical variables

We most commonly use sum-to-zero coding for categorical predictors (via the options (contrasts = c("contr.sum", "contr.poly")) for factors. We use this coding scheme because we are typically interested in main effects and main interactions rather than simple effects or simple interactions (see also [this blog post by Dale Barr](#)). One option is also to use the command `mixed()` from the package **afex**, as it will automatically set all contrasts to sum-to-zero.

Reasons to deviate might include the use of *custom contrasts* to test specific hypotheses.

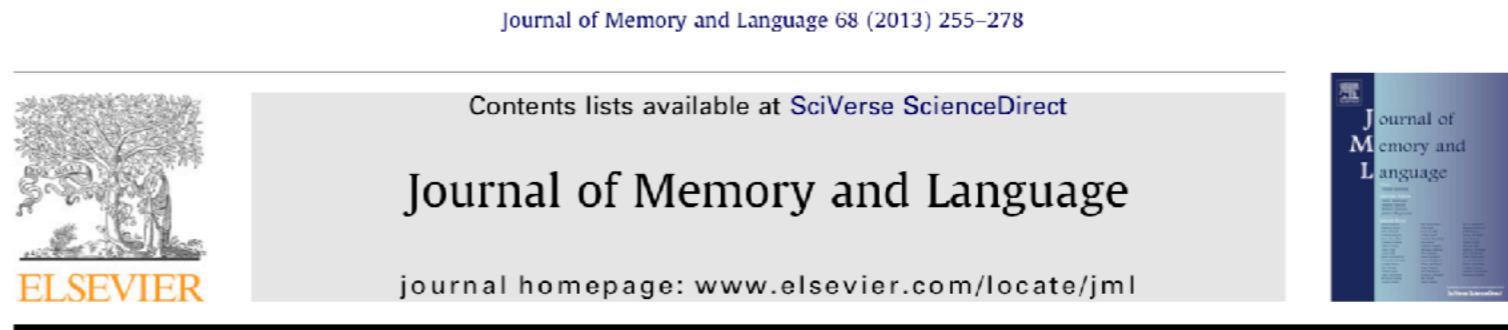
We usually will *follow-up* on significant effects involving factors with more than two levels by either restricting analyses to only two levels in the form of follow-up models (i.e. analyzing a subset of the data comprising only two levels of the given factor) or, alternatively, we use some post-hoc procedures, e.g., using the package **emmeans** (for more details on both, see the section on post-hocs and follow-ups below).

2.2. Continuous variables

As a default, we typically use z-standardization for the continuous predictors (to help with model estimation), unless there are specific reasons not to do so (e.g. if we want to interpret effects on the original scale; in these cases, we typically center the predictor(s)) Centering is essential to make interactions interpretable and avoid so-called nonessential multicollinearity ([Dunlap and Kemery, 1987](#); [Marquardt, 1980](#); also see [this blog post by Philipp Masur](#)).

What shall I include as random effects?

- mixed opinions on the topic
- go maximal!



Random effects structure for confirmatory hypothesis testing:
Keep it maximal



Dale J. Barr ^{a,*}, Roger Levy ^b, Christoph Scheepers ^a, Harry J. Tily ^c

^aInstitute of Neuroscience and Psychology, University of Glasgow, 58 Hillhead St., Glasgow G12 8QB, United Kingdom

^bDepartment of Linguistics, University of California at San Diego, La Jolla, CA 92093-0108, USA

^cDepartment of Brain and Cognitive Sciences, Massachusetts 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."

What shall I include as random effects?

- Failure to include maximal random-effect structures in LMEMs (when such random effects are present in the underlying populations) **inflates Type I error rates.**
- For designs including within-subjects (or within-items) manipulations, random-intercepts-only LMEMs **can have catastrophically high Type I error rates**, regardless of how p-values are computed from them.
- The performance of a data-driven approach to determining random effects (i.e., model selection) depends strongly on the specific algorithm, size of the sample, and criteria used; **moreover, the power advantage of this approach over maximal models is typically negligible.**
- In terms of power, maximal models perform surprisingly well even in a “worst case” scenario where they assume random slope variation that is actually not present in the population.

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

```
1 # fit the model
2 fit.lmer = lmer(formula = reaction ~ 1 + days + (1 + days | subject),
3                   data = df.sleep)
4 # model summary
5 fit.lmer %>%
6   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 

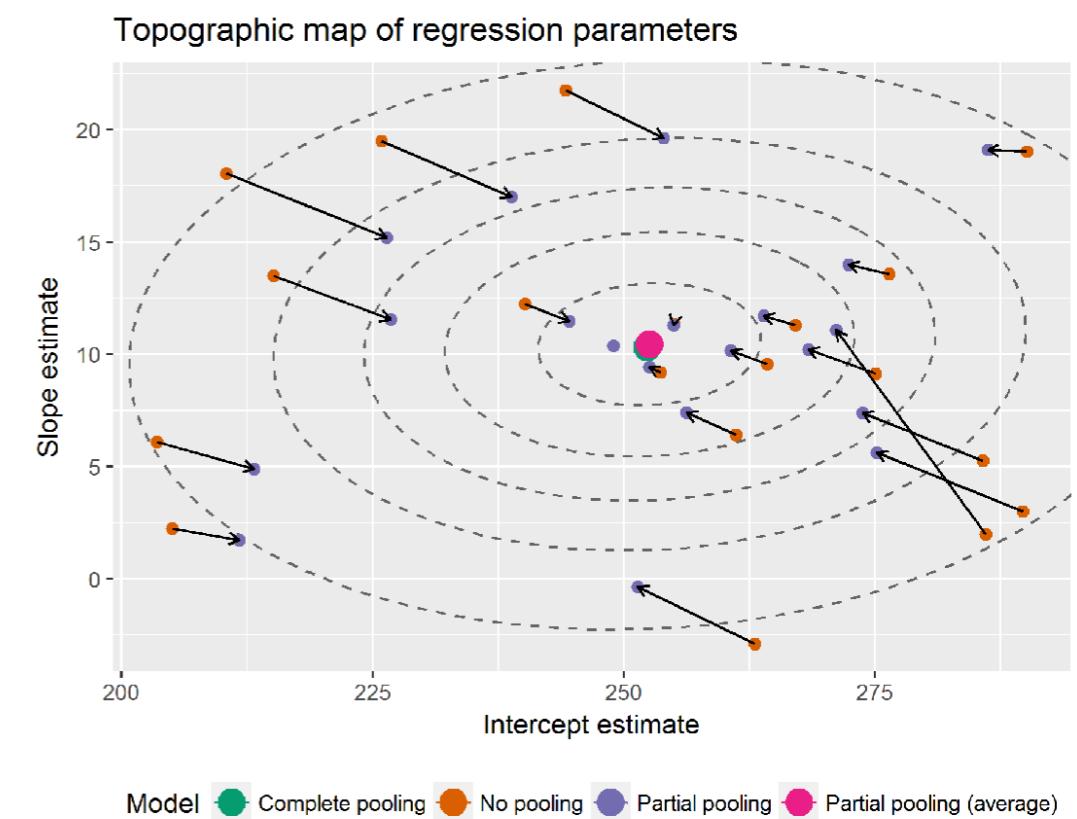
Random effects:
Groups   Name        Variance Std.Dev. Corr
subject (Intercept) 582.73   24.140
          days       35.03   5.919   0.07
Residual            649.36   25.483

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  
days -0.137
```

multivariate Gaussian



Remove the correlation component from your model

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

```
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  
days -0.184
```

↑
random slopes
↑
random intercepts

independent Gaussians

Remove the correlation component from your model

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

```
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  
days -0.184
```

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

independent Gaussians

What if lmer() fails to converge?

3.2.1. Convergence warnings in R's lme4

In case of convergence warnings, we attempt the listed approaches, typically in the order in which they are listed (these steps are based mainly on recommendations by Ben Bolker/the lme4 team and [Barr et al., 2013](#)). For each step, we check whether it resolves the convergence issues.

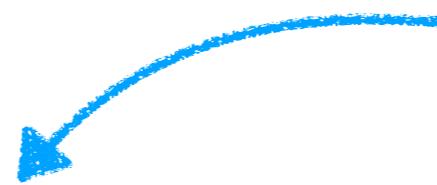
Before going through the listed steps, some of us would always set the optimizer to bobyqa as default via `optimizer = c("bobyqa")` (since it has been suggested that it might work better for the kind of data that we typically have in psychology) and/or switch off the calculation of the gradient and Hessian via `control = [g]lmerControl(calc.derivs = FALSE)`; these settings might already resolve the convergence issues .

1. We increase the number of iterations to the maximum.
2. We use the estimates from the previous (non-converged) fit as our new starting values.
3. We compare the estimates of different optimizers (e.g., using `allFit()`); if different optimizers give highly similar estimates (even if they give convergence warnings), the convergence warnings can be considered false positives.
4. We follow the steps suggested [in Ben Bolker's blog post](#):
 - a. Center independent (and dependent) variables instead of scaling; multiply the independent variables by 10 (or 100) to increase the variance
 - b. Robustness check: Check whether certain random correlations are close to +/-1 and/or certain random slope variances are close to 0. If yes, remove those; afterwards check whether the estimates are still the same
 - c. Double check gradient calculations: Check the (parallel) minimum of the absolute and relative gradients. If those gradients are > 0.001, gradient calculation is likely not a problem.

What if lmer() fails to converge?

```
1 # fit the model
2 fit.lmer = lmer(formula = reaction ~ 1 + days + (1 + days | subject),
3   data = df.sleep)
4
5 # explore different optimization algorithms
6 fit.all = allFit(fit.lmer)
7
8 # summarize result
9 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

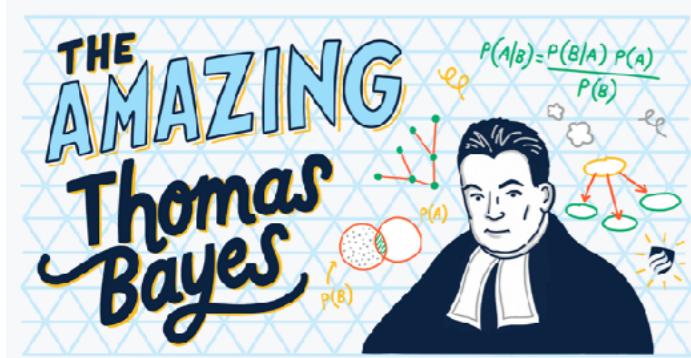
\$llik	bobyqa	Nelder_Mead	nlminbwrap
	-885.7239	-885.7239	-885.7239
	nloptwrap.NLOPT_LN_NELDERMEAD	nloptwrap.NLOPT_LN_BOBYQA	
	-885.7239	-885.7239	

\$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

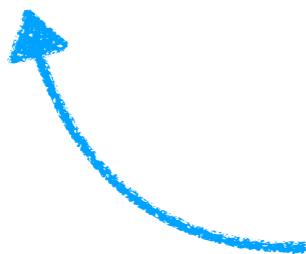
As an alternative to targeting convergence issues within **lme4**, we suggest fitting the same model with **brms** and comparing it to the **lme4** fit. We assume that both provide similar results when

Plan for today

- Quick recap: Interpreting `summary()` output
- Linear mixed effects model
 - A worked example
 - Getting p-values
 - Reporting results
 - Let's simulate some `lmer()`s
 - Understanding `lmer()` syntax
 - `lmer()` standard operating procedures

Feedback

<https://tinyurl.com/psych252survey21>



so that students who can't be here
synchronously can give feedback, too

Thank you!

