

Linear mixed effects models 3



COLLABORATIVE PLAYLIST
psych252
<https://tinyurl.com/psych252spotify24>

PLAY ...

02/26/2024

Things that came up



Brenton Wiernik @bmwiernik

...

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

```
sandwich %>%  
  pivot_longer()
```



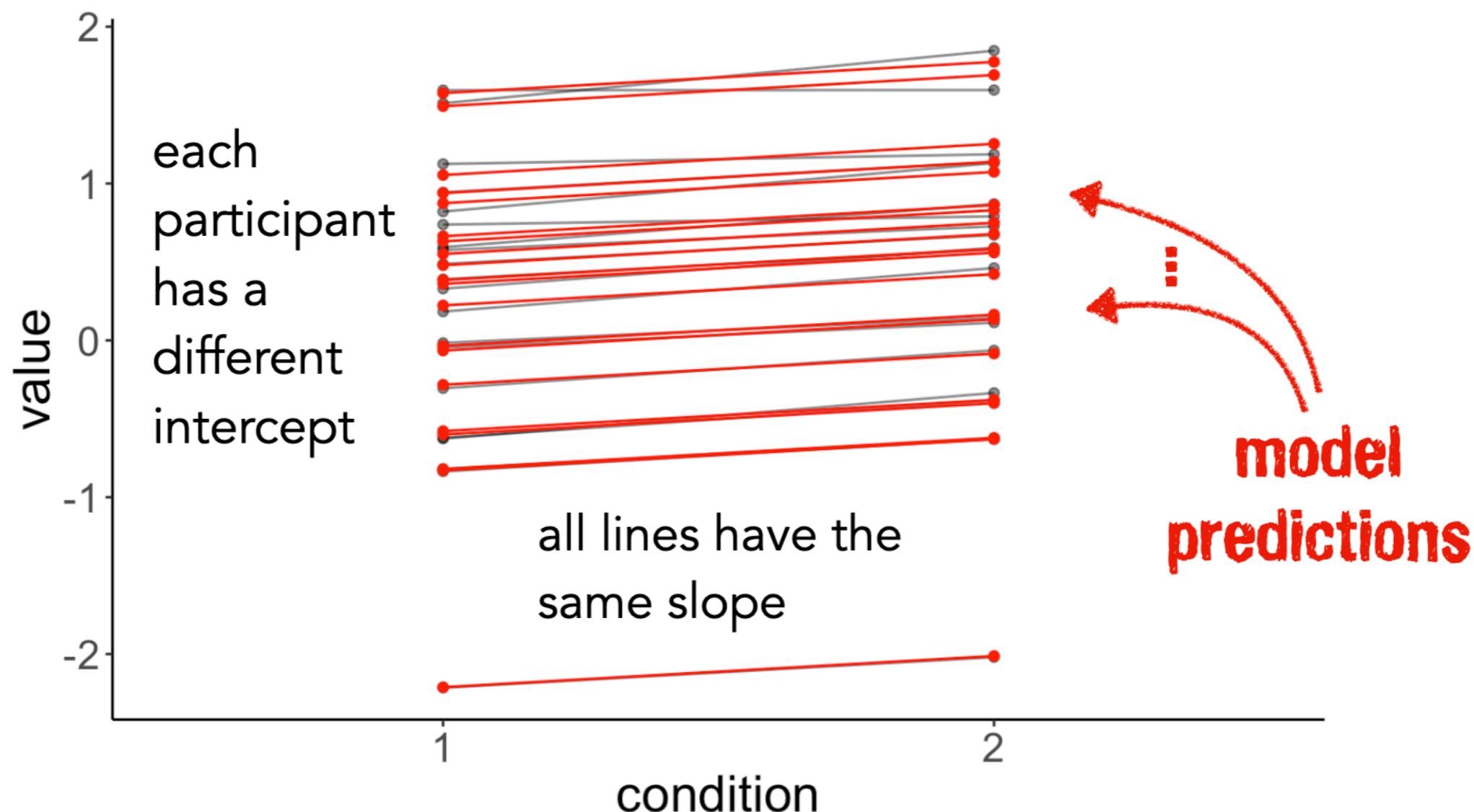
8:34 AM · Mar 1, 2021 · Twitter for iPhone

273 Retweets 29 Quote Tweets 2,012 Likes

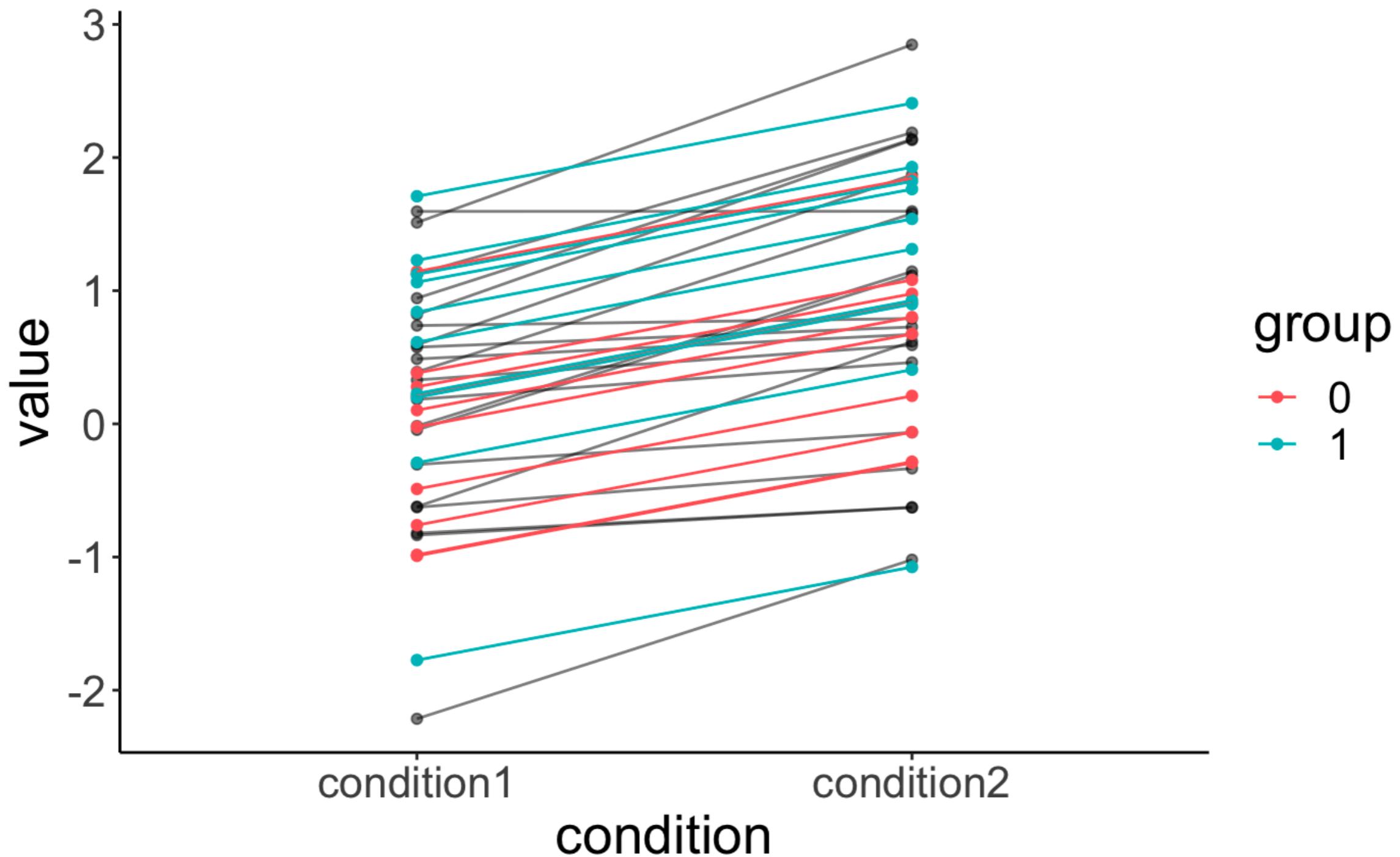
Linear mixed effects model (accounting for dependence)

Predictions by the linear mixed effects model

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



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



Plan for today

- Quick recap
- Simpsons paradox
- Understanding `lmer()` syntax
- Reporting results
- Let's simulate some `lmer()`s
- `lmer()` standard operating procedures
- Some more examples

Quick recap

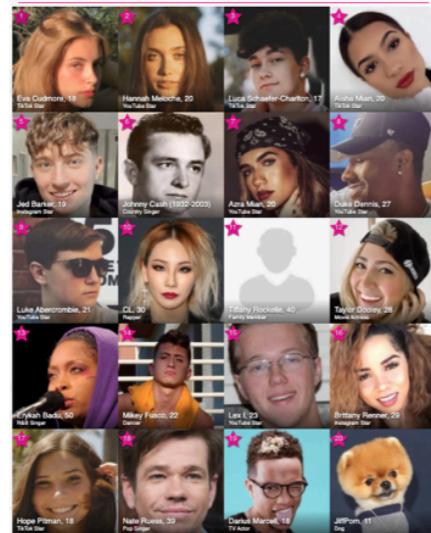
Quick recap: Linear mixed effects models

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



born on February 25th

N = 100 participants

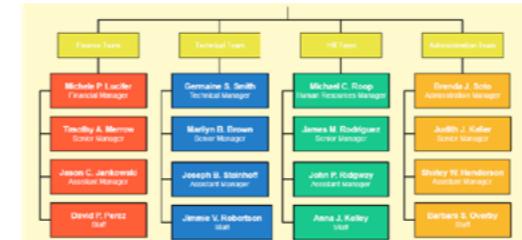


born on February 26th

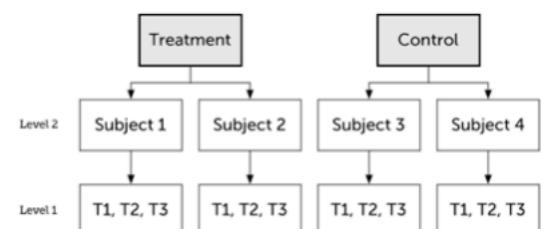
```
1 lmer(formula = trustworthy ~ 1 + birthday +
2   (1 | item),
      data = df.birthday)
```

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

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.

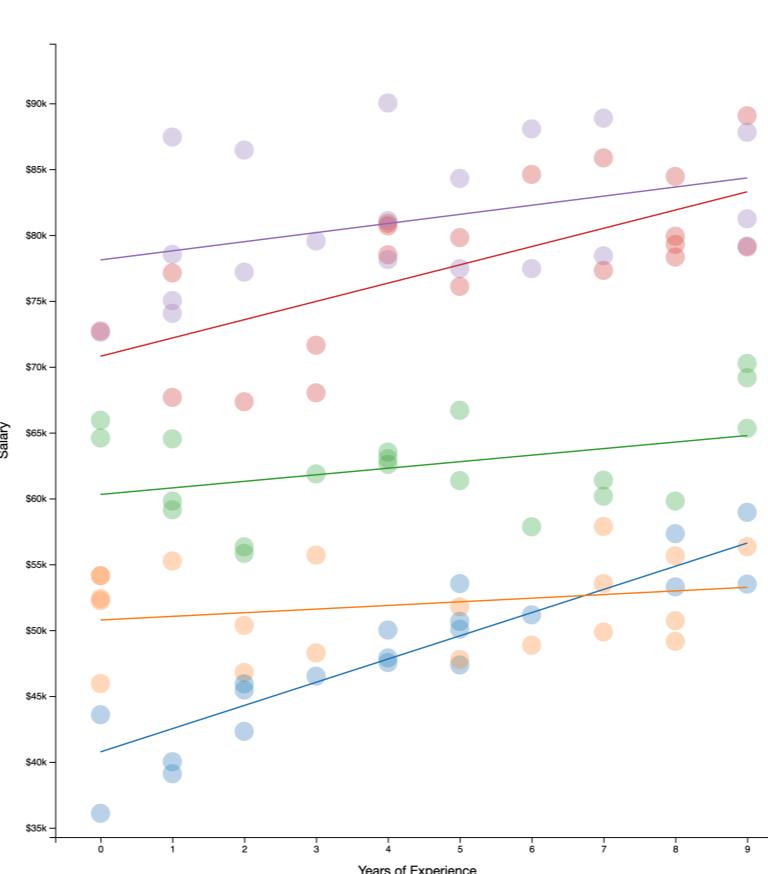
To incorporate both of these realities into our model, we want both the slope and the intercept to vary depending on the department of the faculty member. We can describe this with the following notation:

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

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

$$\text{salary}_i = \beta_{0j|i} + \beta_{1j|i} * \text{experience}_i$$

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).

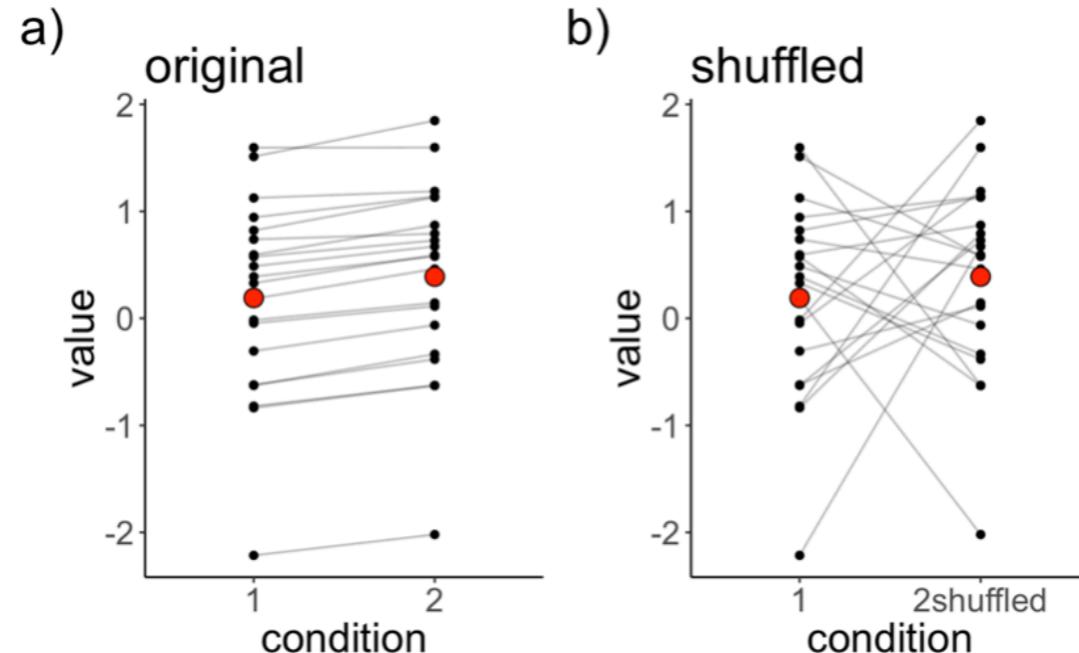


13

17

Quick recap: Dependence matters

Dependence
Does it really matter? Is there a significant difference
between conditions 1 and 2?



20

Linear model

```
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: $b_0, b_1, s_{\text{error}}$

Linear mixed effects model

```
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: $b_0, b_1, s_{\text{error}}, s_U$

Model coefficients

Linear model

```
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

```
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

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

Quick recap: `lmer()` summary

Understanding the `lmer()` summary

```
1 # fit a linear mixed effects model
2 lmer(formula = value ~ condition + (1 | participant),
3       data = df.original) %>%
4   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 
condition2 -0.048
```

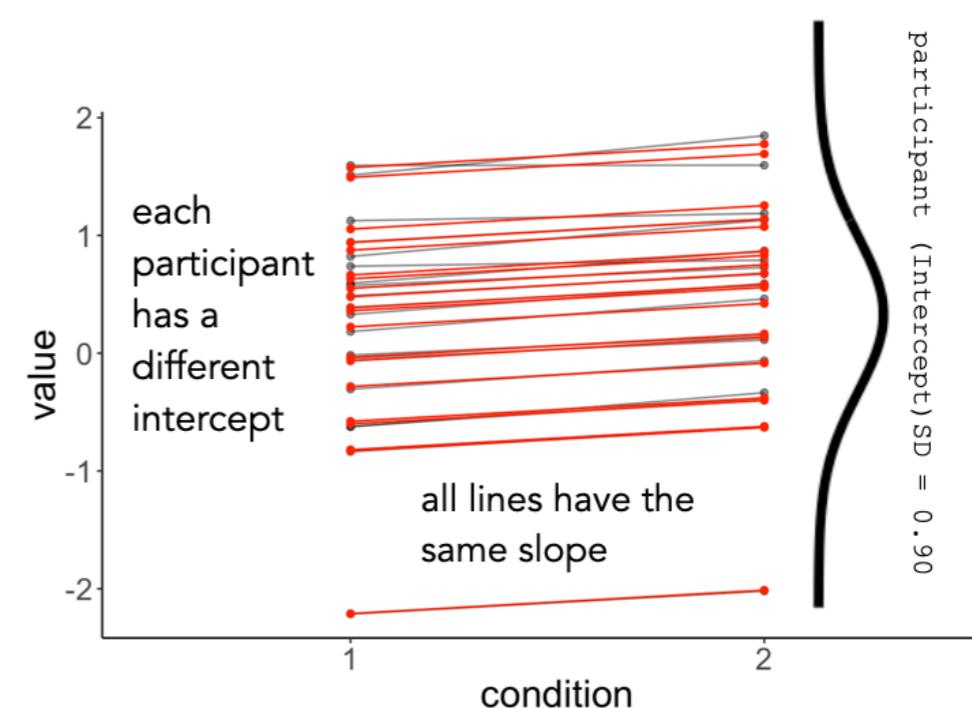
Fixed effects

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



18

17

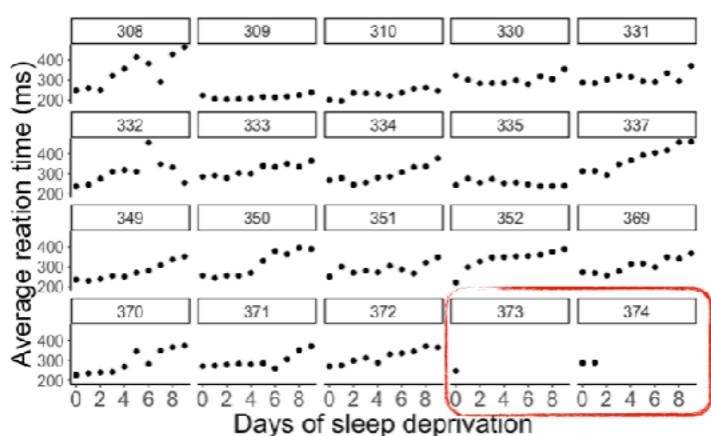
11

Quick recap: worked example

Data set

How does sleep deprivation affect reaction time?

subject	days	reaction
308	0	249.56
309	1	258.70
308	2	250.80
309	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

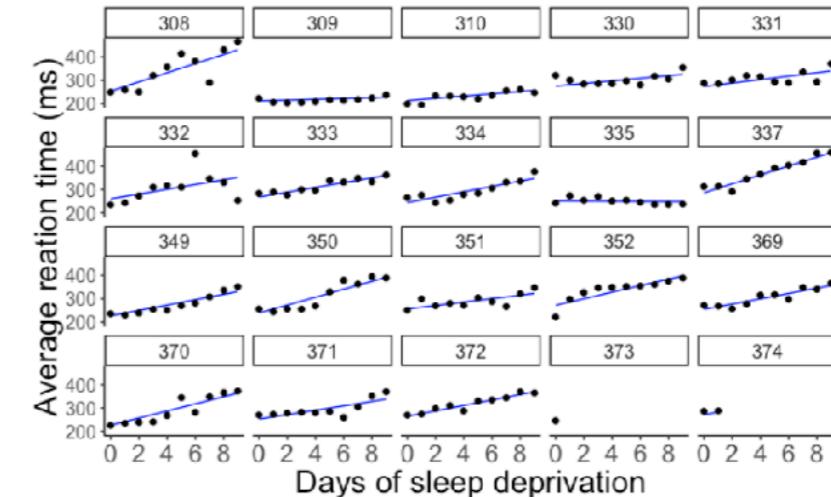
26

Partial pooling: Fit mixed effects model

Intercepts and slopes differ
between participants

random intercept random slope

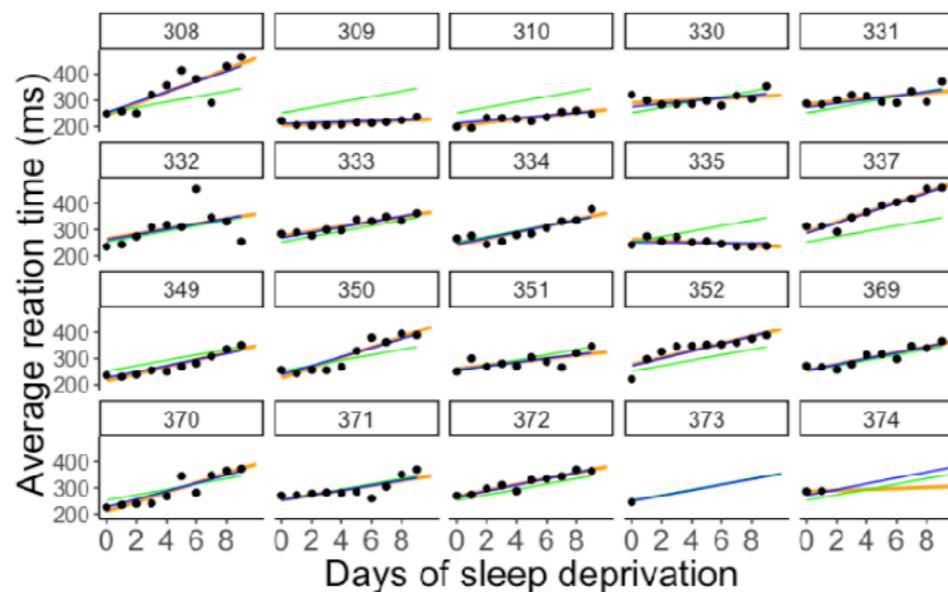
`lmer(formula = reaction ~ 1 + days + (1 + days | subject),
data = df.sleep)`



33

Comparison

complete pooling
no pooling
partial pooling

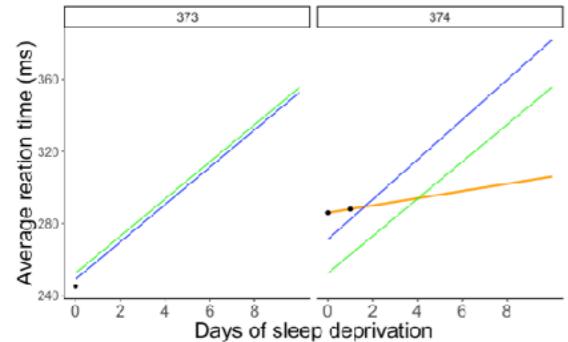


37

12

Quick recap: shrinkage

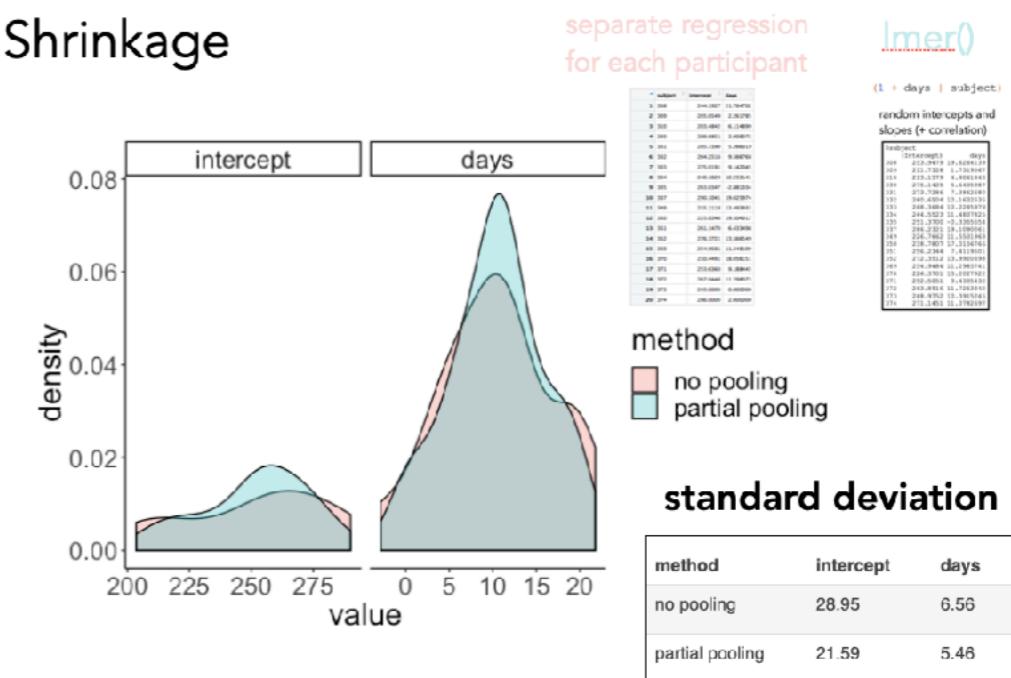
Comparison



- 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

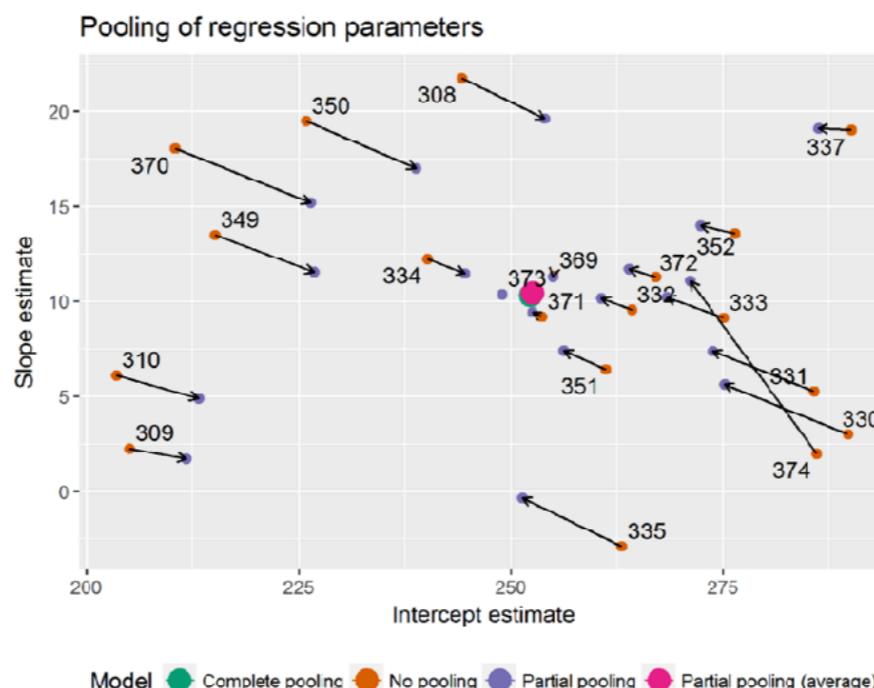
38

Shrinkage



39

Shrinkage

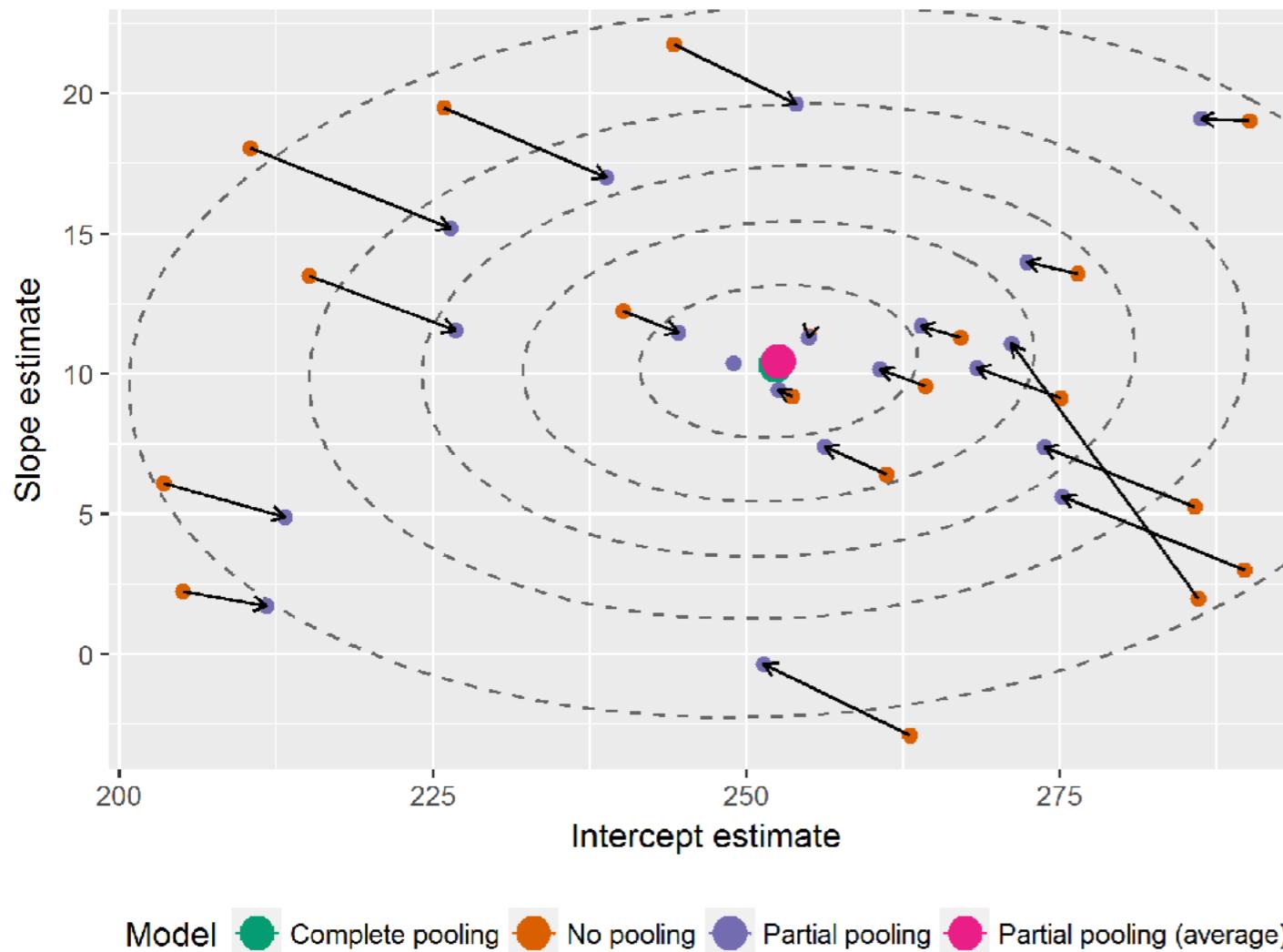


40

13

Shrinkage

Topographic map of regression parameters

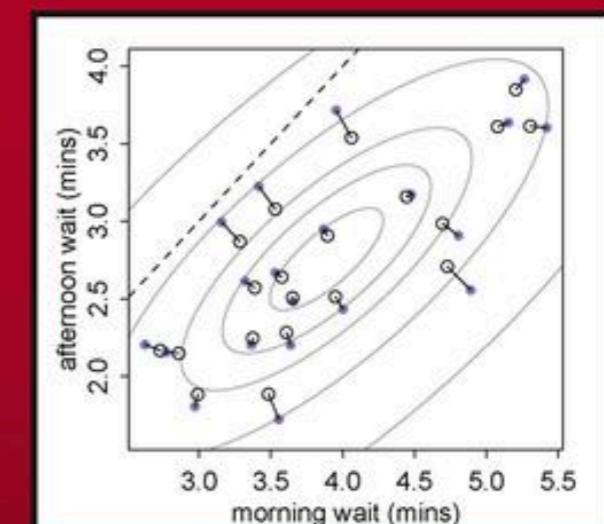


mixed effects model estimates a multi-variate Gaussian to account for (possible) correlations between intercepts and slopes

Texts in Statistical Science

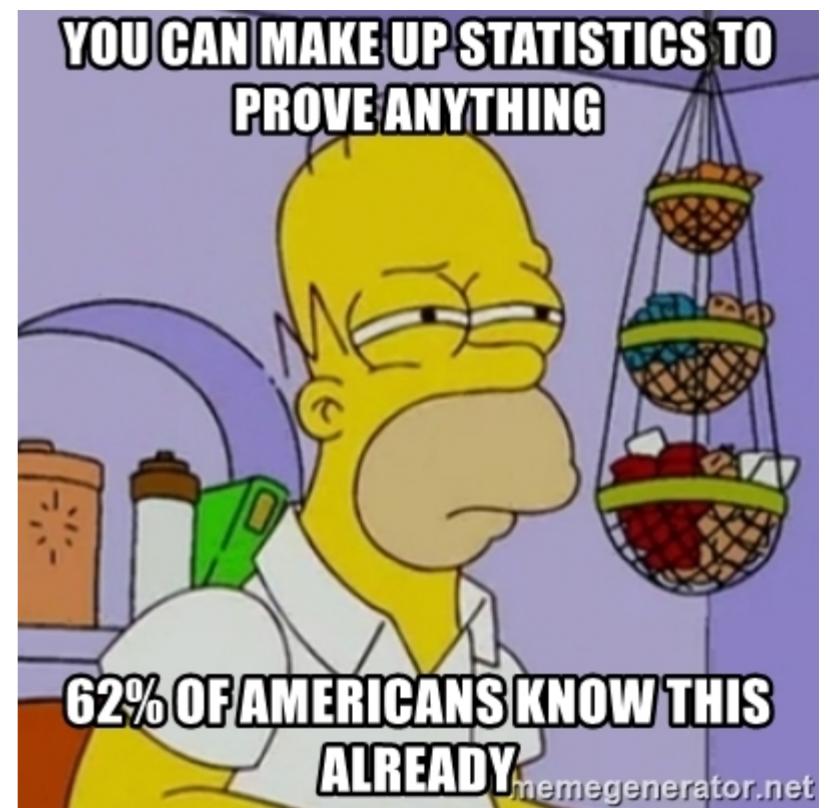
Statistical Rethinking

A Bayesian Course with Examples in R and Stan

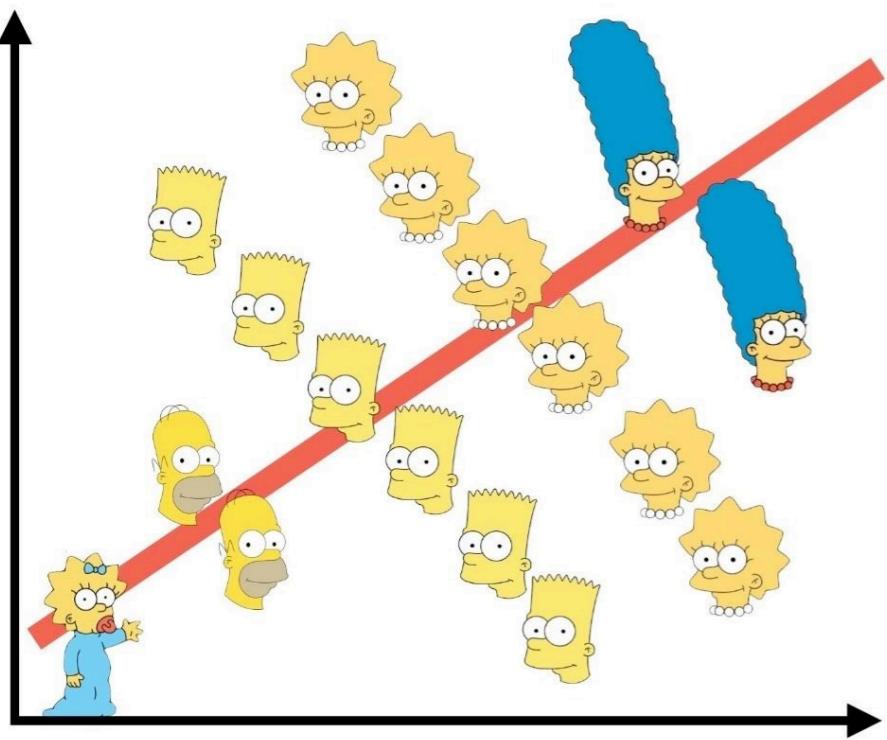


Richard McElreath

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

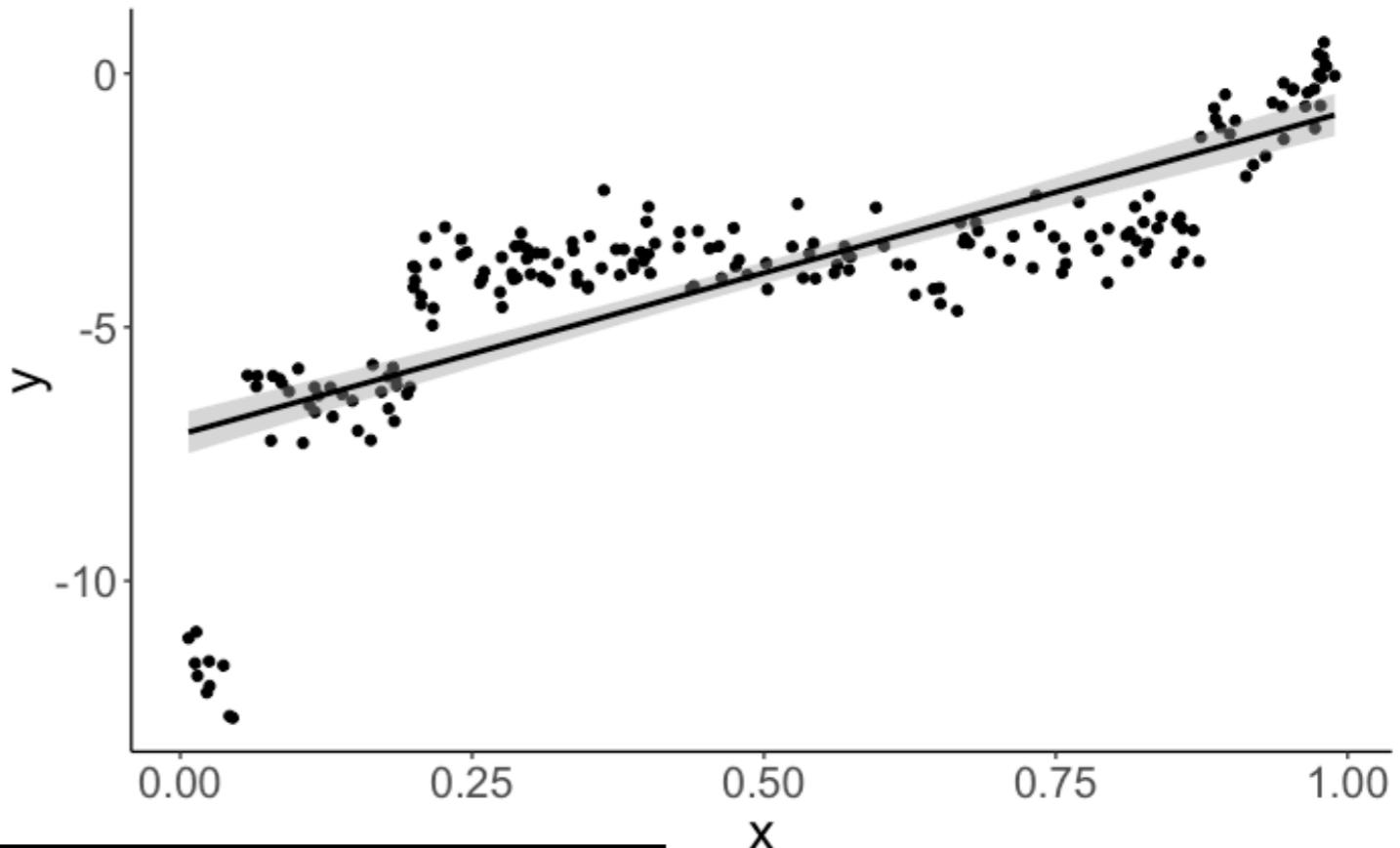


Simpsons Paradox



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

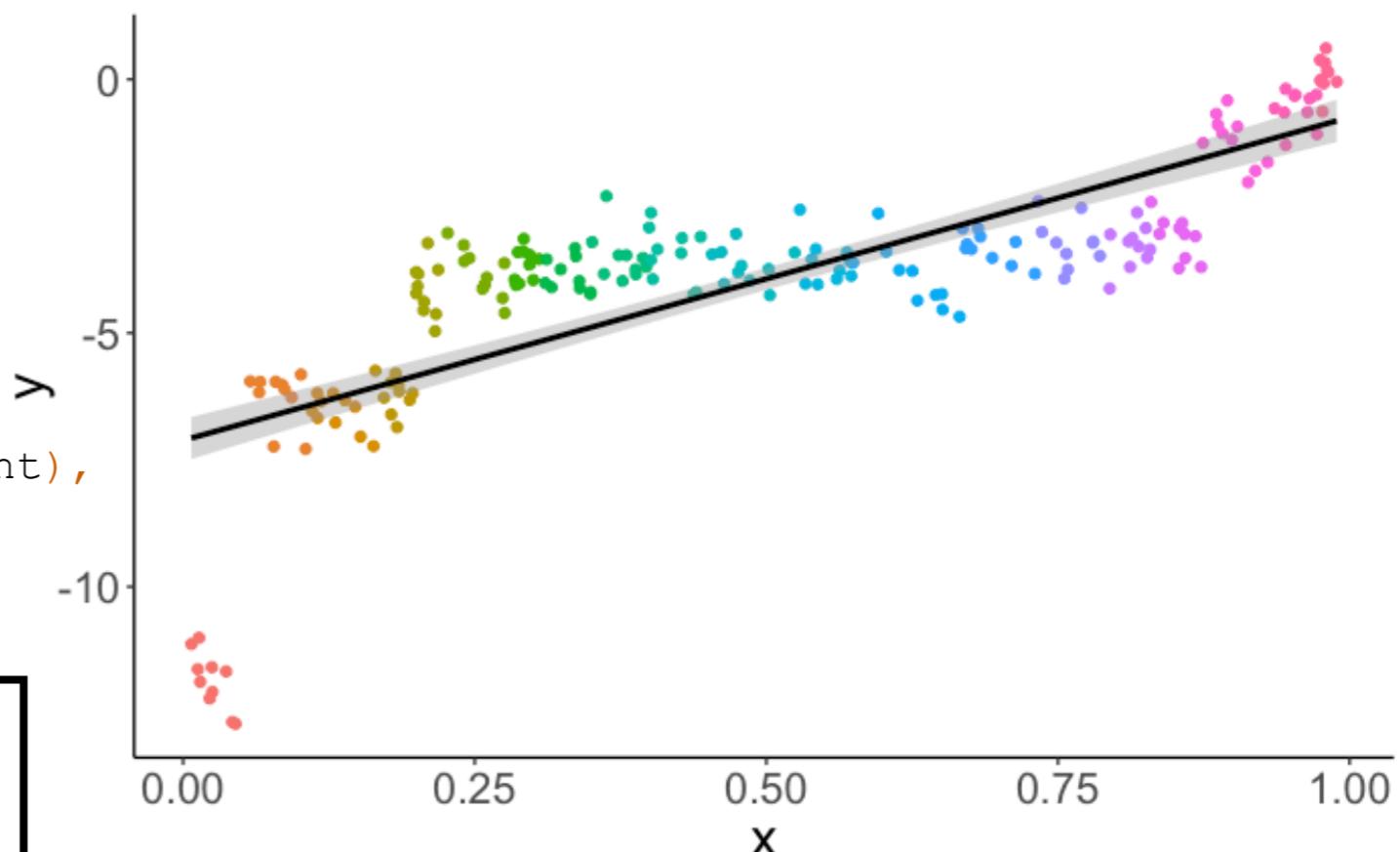
```
Number of obs: 200, groups: participant, 20
```

```
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

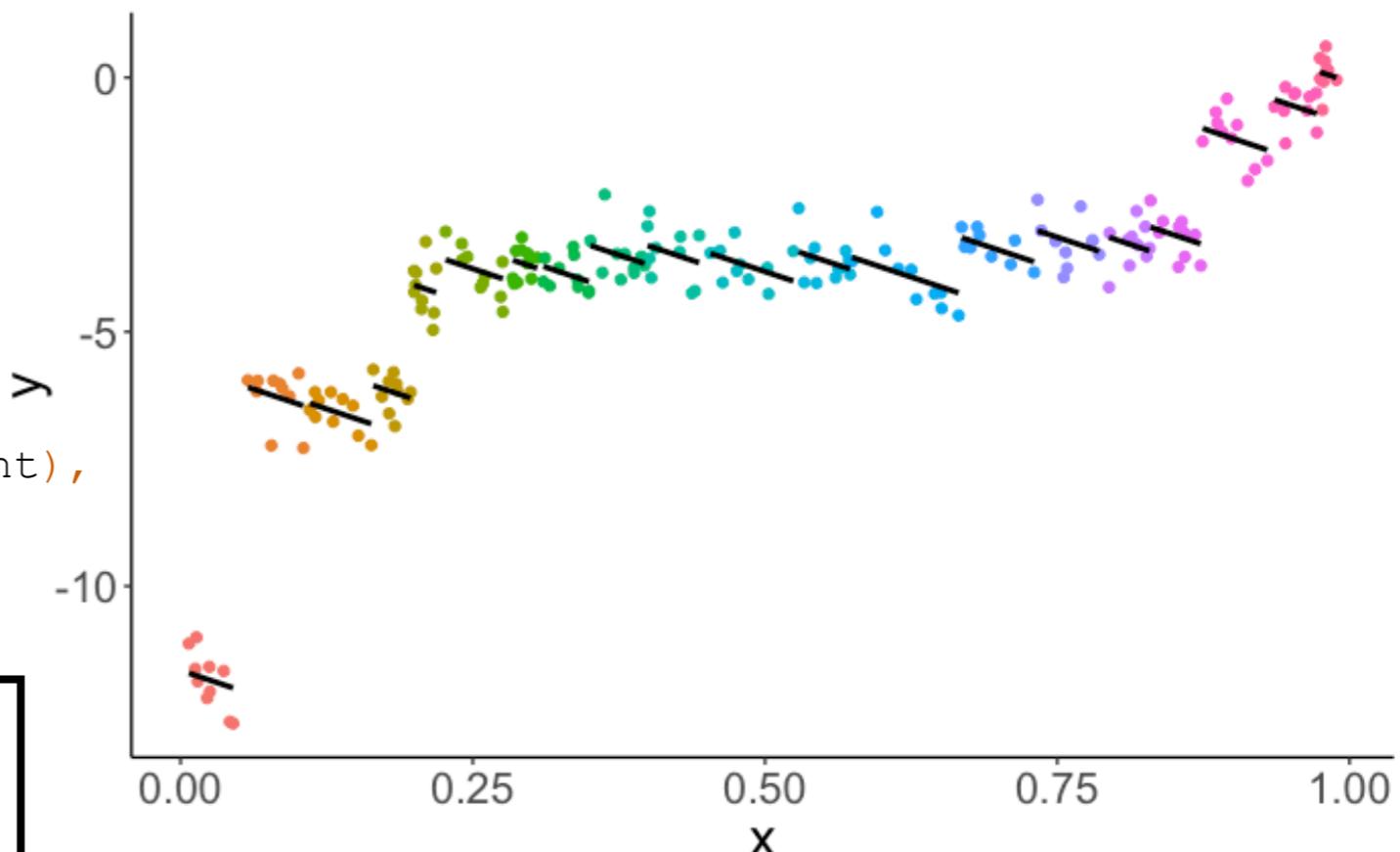
```
Number of obs: 200, groups: participant, 20
```

```
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

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

`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) + (1 school:class)</code>	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)
```

$(1 \mid \text{subject})$

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

$(0 + \text{days} \mid \text{subject})$

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

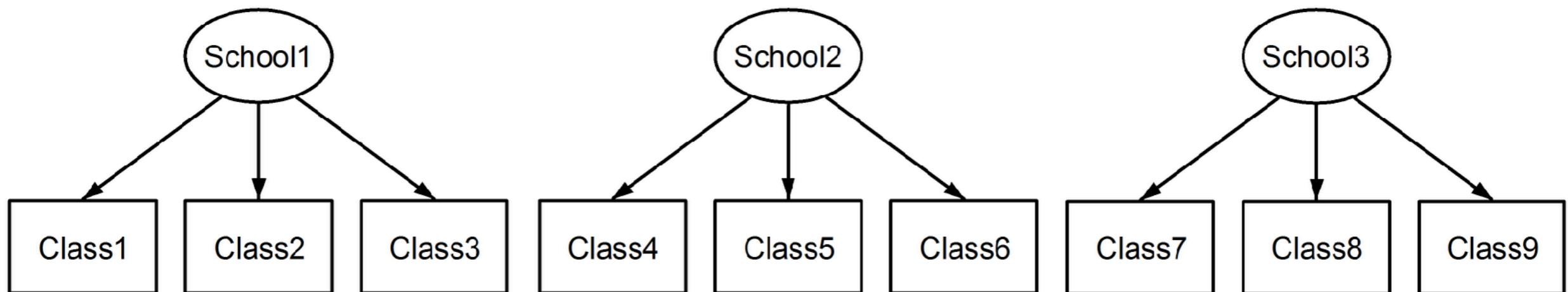
\dots
 $(1 + \text{days} \mid \text{subject})$

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

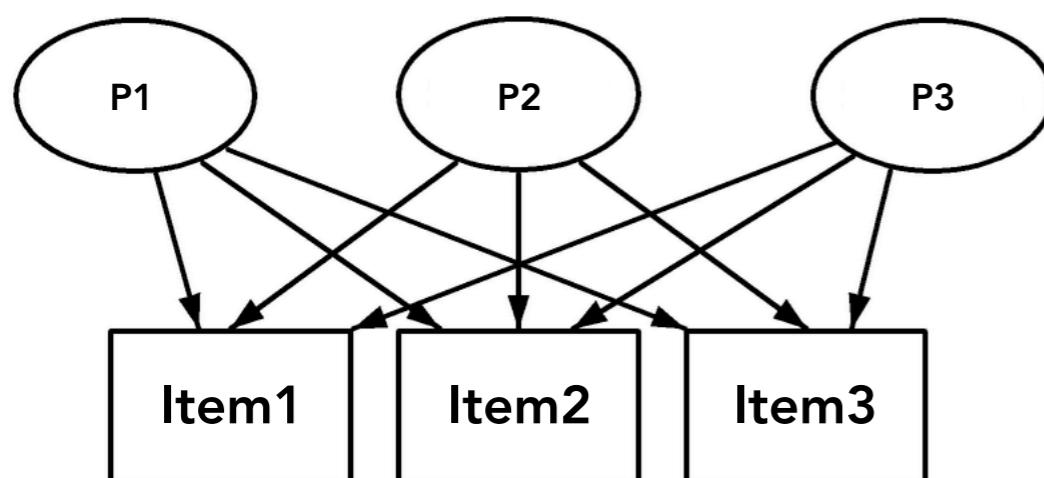
Multi-level models

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



each class only appears within one school

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

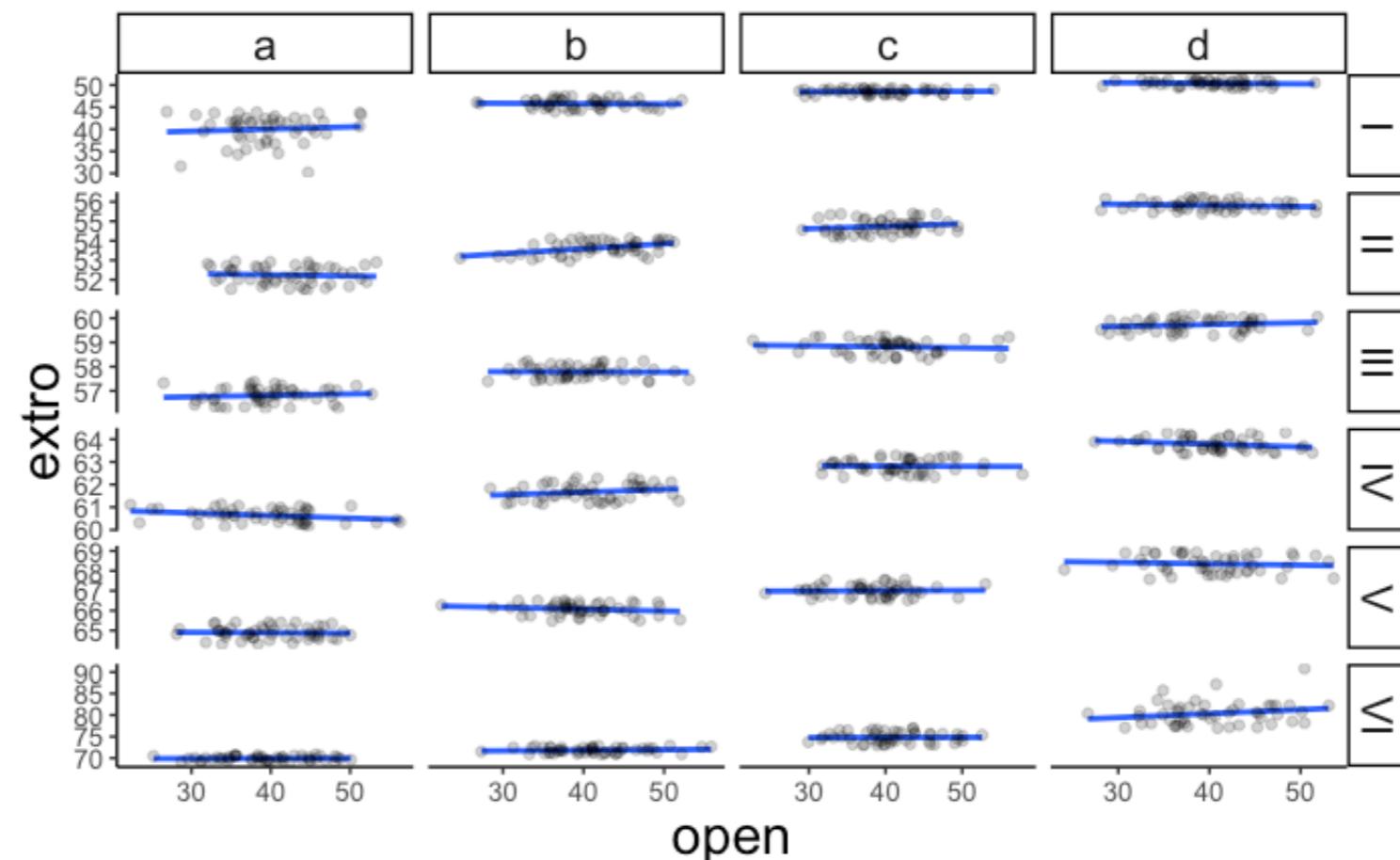


**each participant
rates each item**

Multi-level models

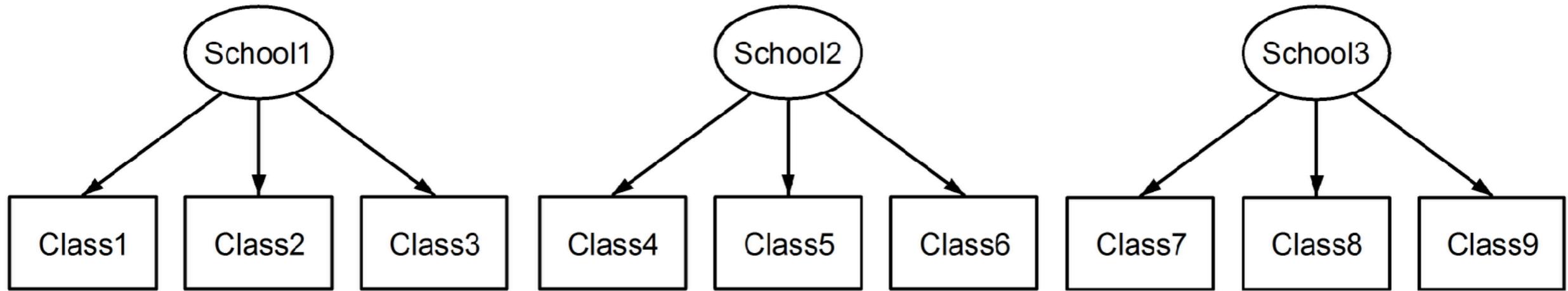
	id	extro	open	agree	social	class	school
1	1	63.69356	43.43306	38.02668	75.05811	d	IV
2	2	69.48244	46.86979	31.48957	98.12560	a	VI
3	3	79.74006	32.27013	40.20866	116.33897	d	VI
4	4	62.96674	44.40790	30.50866	90.46888	c	IV
5	5	64.24582	36.86337	37.43949	98.51873	d	IV
6	6	50.97107	46.25627	38.83196	75.21992	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 $(1 | \text{School}) + (1 | \text{School} : \text{Class})$



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

```
3
4 # print model summary
5 fit.nested %>% summary()
6
7 # model coefficients
8 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

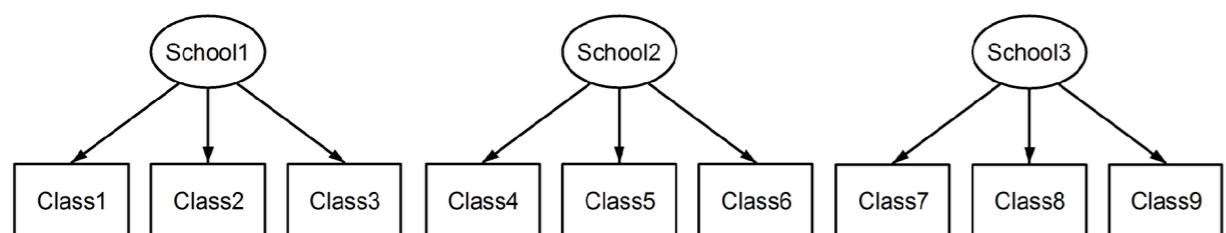
Scaled residuals:
    Min      1Q  Median      3Q     Max 
-9.9949 -0.3348  0.0057  0.3394 10.6476 

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

Multi-level models

nested

$(1 | \text{School}) + (1 | \text{School} : \text{Class})$



random intercepts
of class within
each school

random intercepts of
schools

random intercepts

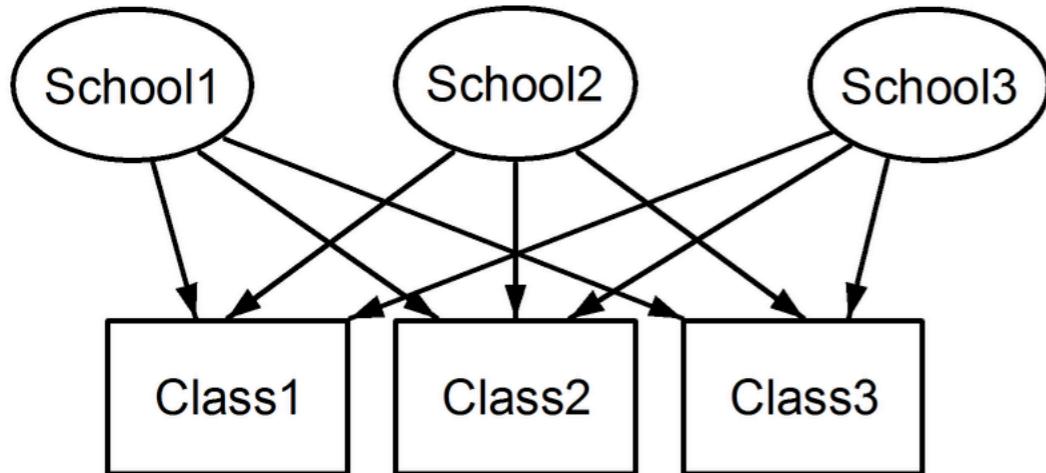
	\$`class:school`	(Intercept)	open	agree	social
a:I	53.77106	0.006106514	-0.007665927	0.0005404069	
a:II	58.23842	0.006106514	-0.007665927	0.0005404069	
a:III	58.71819	0.006106514	-0.007665927	0.0005404069	
a:IV	58.68813	0.006106514	-0.007665927	0.0005404069	
a:V	58.68268	0.006106514	-0.007665927	0.0005404069	
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 $(1 | \text{School}) + (1 | \text{Class})$



each class
appears in each of
the schools

```
1 # fit crossed model
2 fit.crossed = lmer(extro ~ open + agree + social + (1 | school) + (1 | class), data = df.school)
3
```

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

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

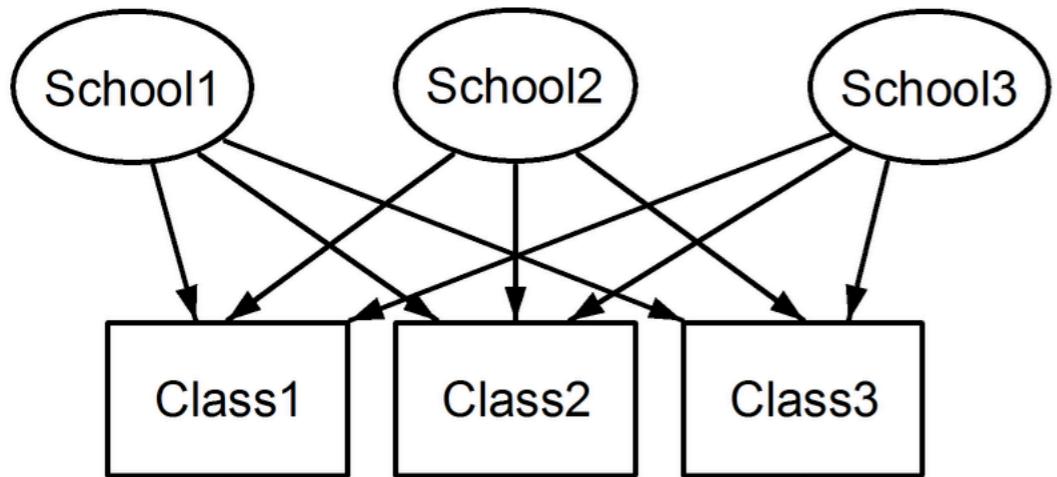
REML criterion at convergence: 4723.9

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 $(1 | \text{School}) + (1 | \text{Class})$



each class is in
each of the schools

random intercepts

**random intercepts
of school**

**random intercepts of
class**

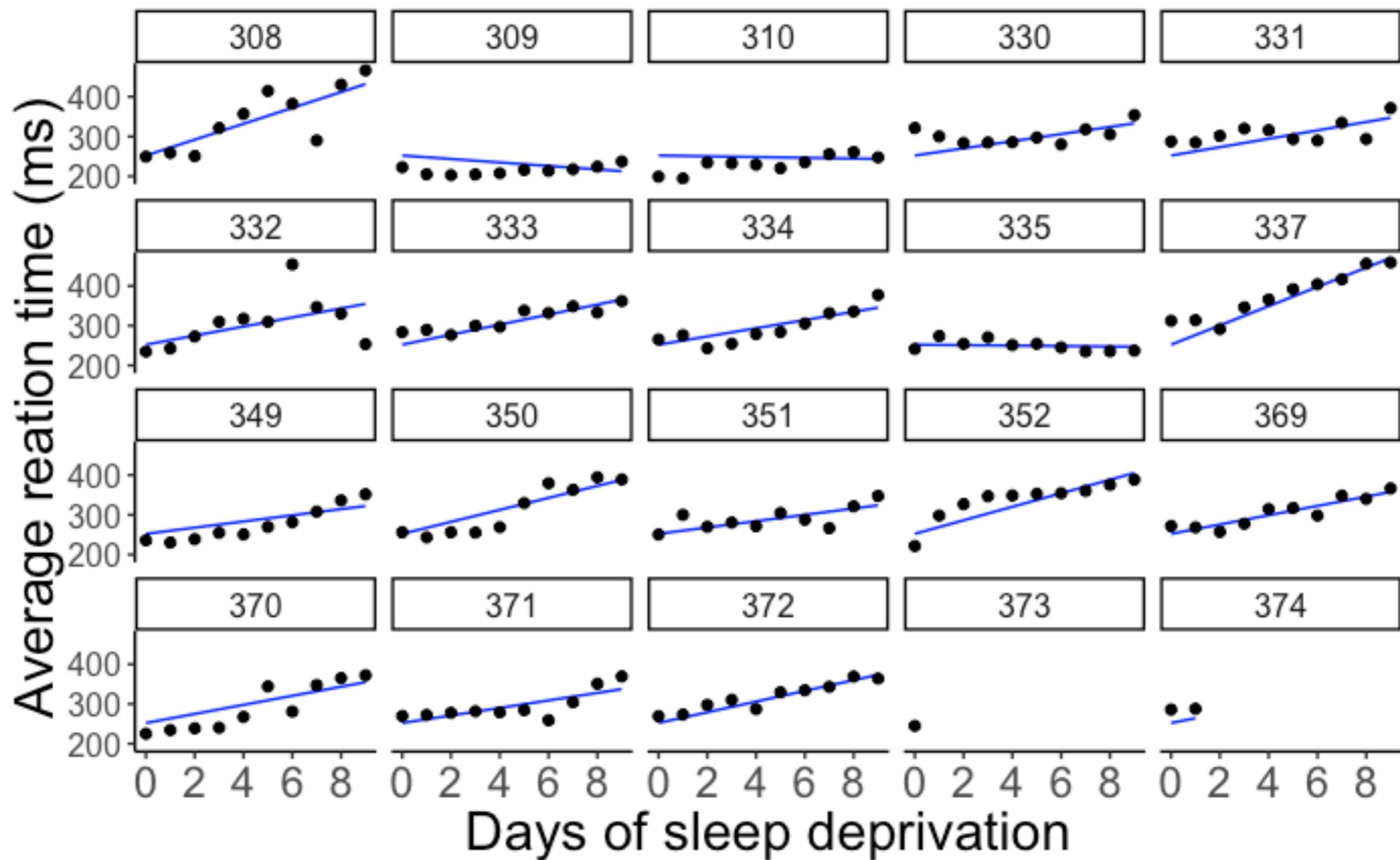
	\$school	(Intercept)	open	agree	social
I		46.10663	0.01083374	-0.005420032	-0.001761963
II		54.02956	0.01083374	-0.005420032	-0.001761963
III		58.22277	0.01083374	-0.005420032	-0.001761963
IV		62.15508	0.01083374	-0.005420032	-0.001761963
V		66.51062	0.01083374	-0.005420032	-0.001761963
VI		74.16838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

Reporting results

Visualization

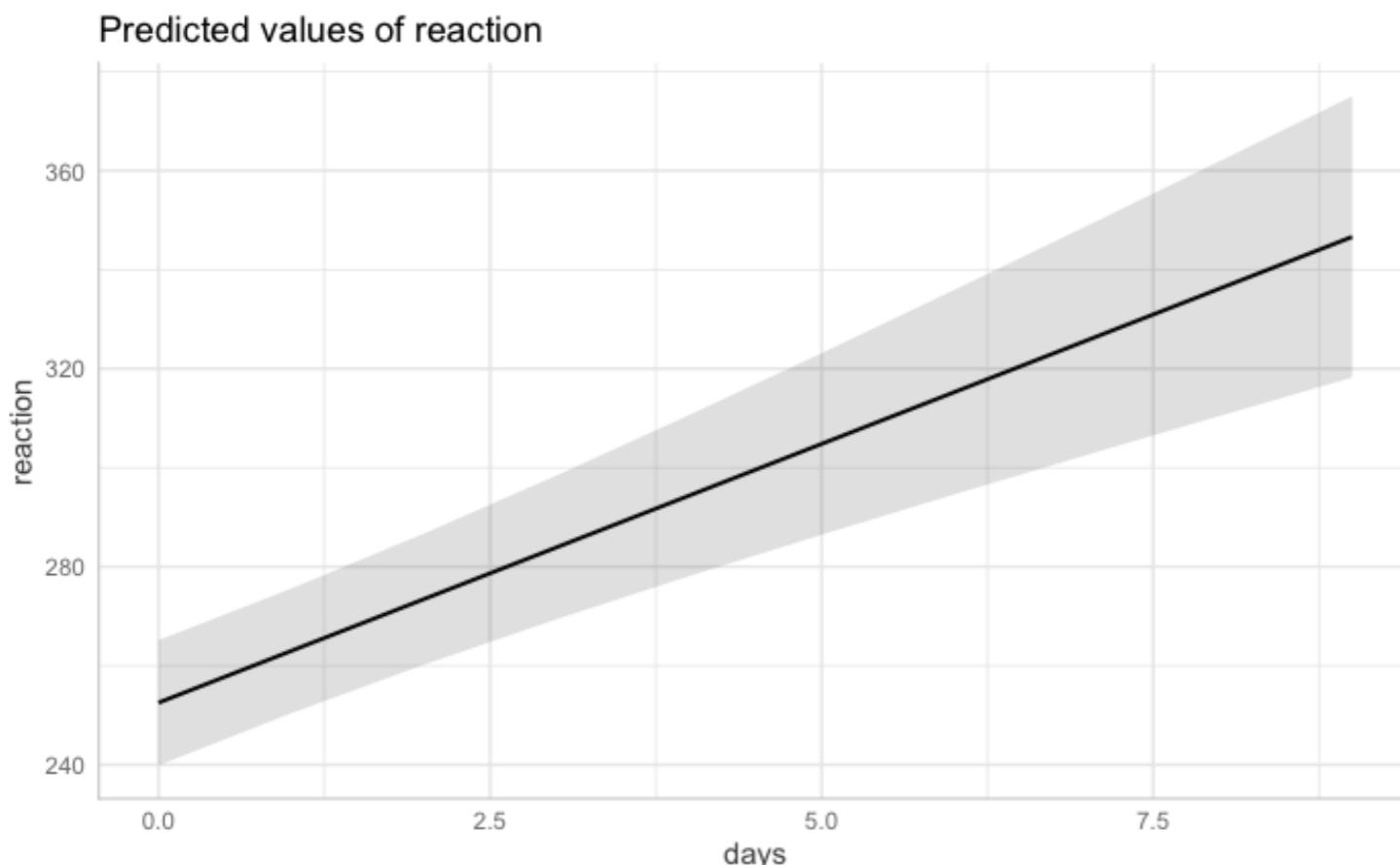


show the data together with the model predictions

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.

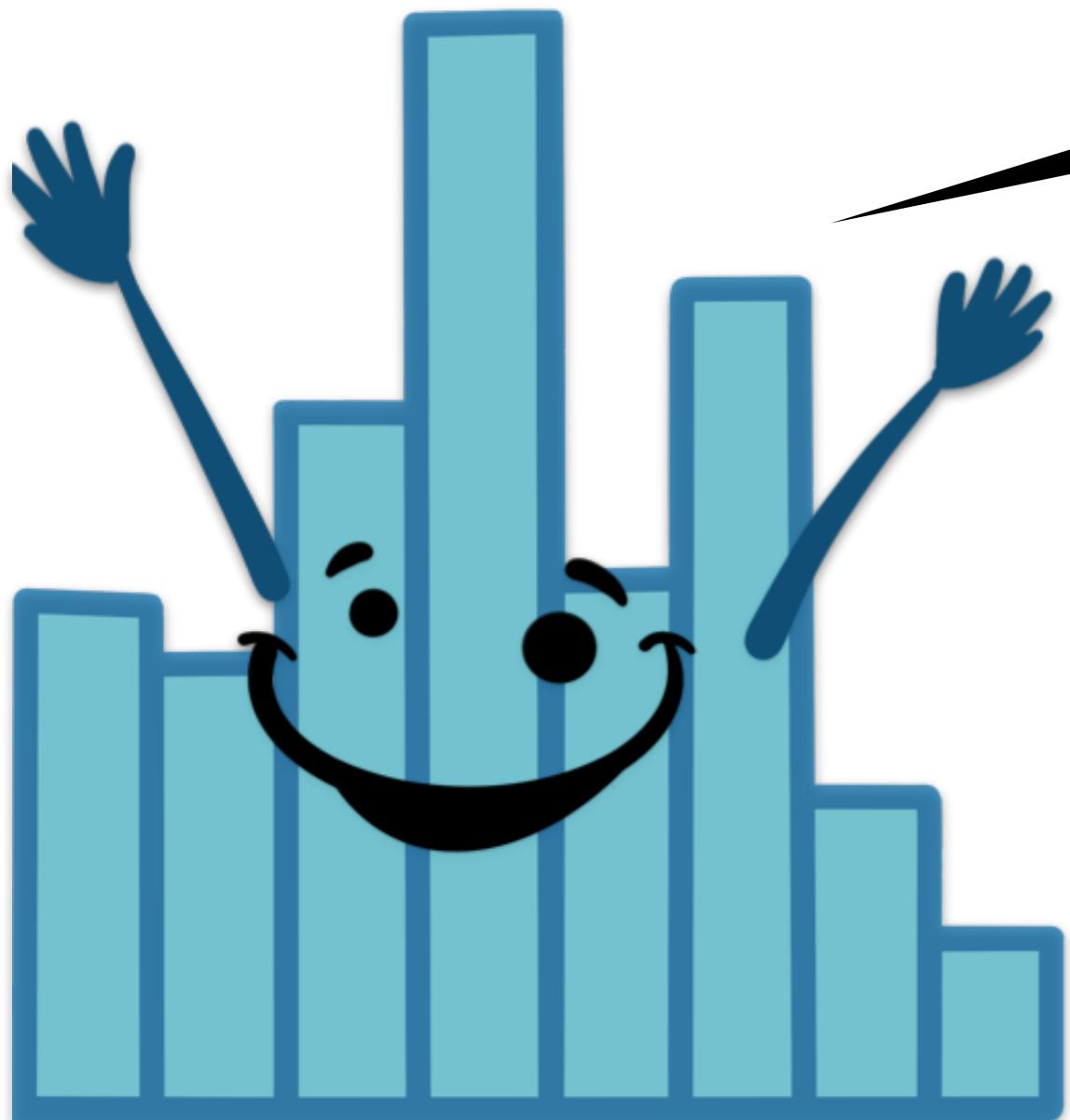
model specification: `normality rating ~ 1 + structure * norm`

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.

02:00

stretch break!



Let's simulate some 1mer()s

Let's simulate an `lmer()`

```
1 # make example reproducible
2 set.seed(1)
3
4 # parameters
5 sample_size = 100
6 b0 = 1
7 b1 = 2
8 sd_residual = 1
9 sd_participant = 0.5
10
11 # generate the data
12 df.mixed = tibble(participant = rep(1:sample_size, 2),
13 condition = rep(0:1, each = sample_size)) %>%
14 group_by(participant) %>%
15 mutate(intercepts = rnorm(n = 1, sd = sd_participant)) %>%
16 ungroup() %>%
17 mutate(value = b0 + b1 * condition + intercepts + rnorm(n(), sd = sd_residual)) %>%
18 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

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

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

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

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 # make example reproducible
2 set.seed(1)
3
4 # parameters
5 sample_size = 100
6 b0 = 1
7 b1 = 2
8 sd_residual = 1
9 sd_participant = 0.5
10
11 # generate the data
12 df.mixed = tibble(participant = rep(1:sample_size, 2),
13                     condition = rep(0:1, each = sample_size)) %>%
14   group_by(participant) %>%
15   mutate(intercepts = rnorm(n = 1, sd = sd_participant)) %>%
16   ungroup() %>%
17   mutate(value = b0 + b1 * condition + intercepts + rnorm(n(), sd = sd_residual)) %>%
18   arrange(participant, condition)
```

```
1 # fit model
2 fit.test = lmer(formula = value ~ 1 + condition + (1 | participant),
3                  data = df.mixed)
4
```

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

REML criterion at convergence: 606

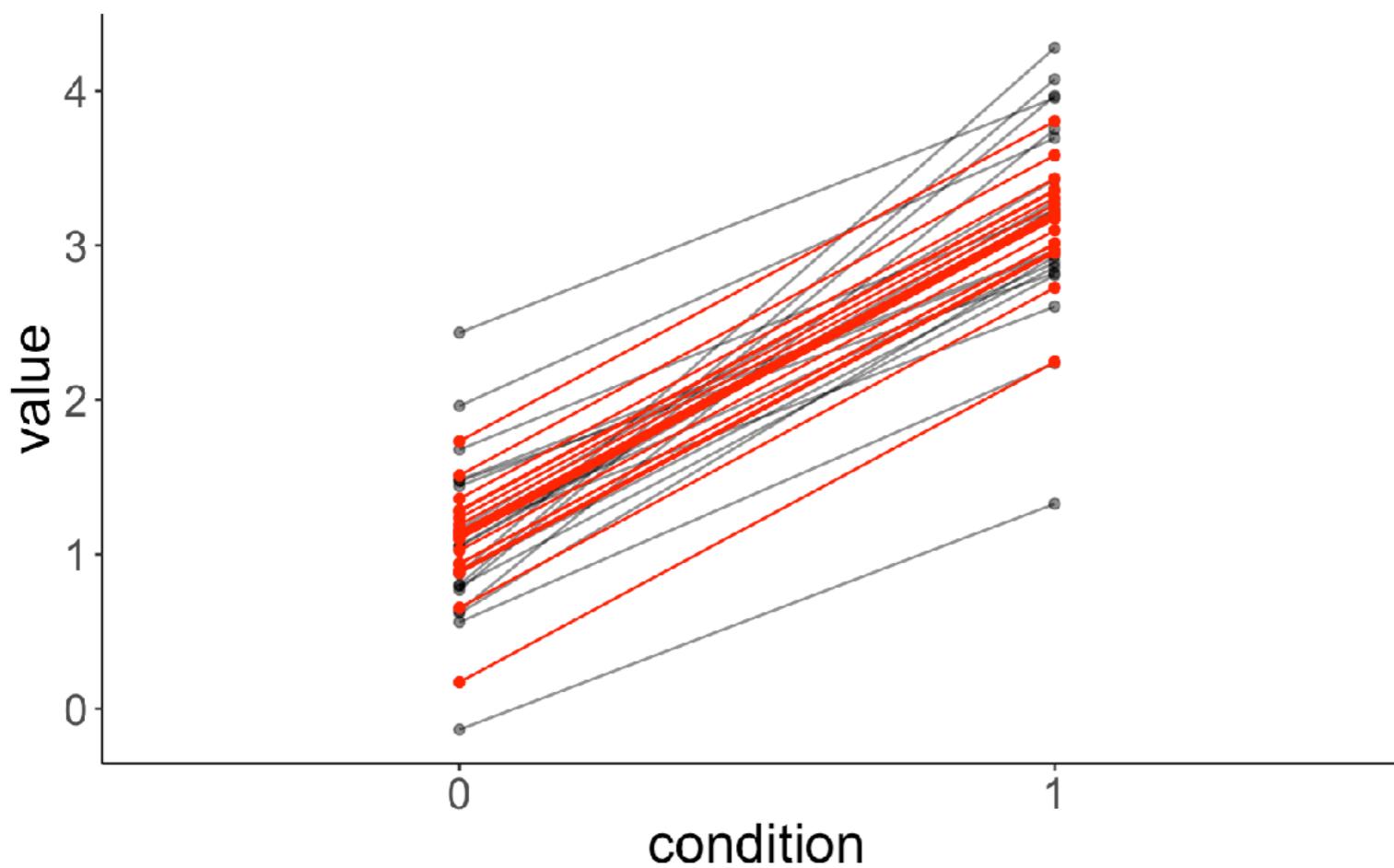
Scaled residuals:
    Min      1Q  Median      3Q     Max 
-2.53710 -0.62295 -0.04364  0.67035  2.19899 

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
```

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 

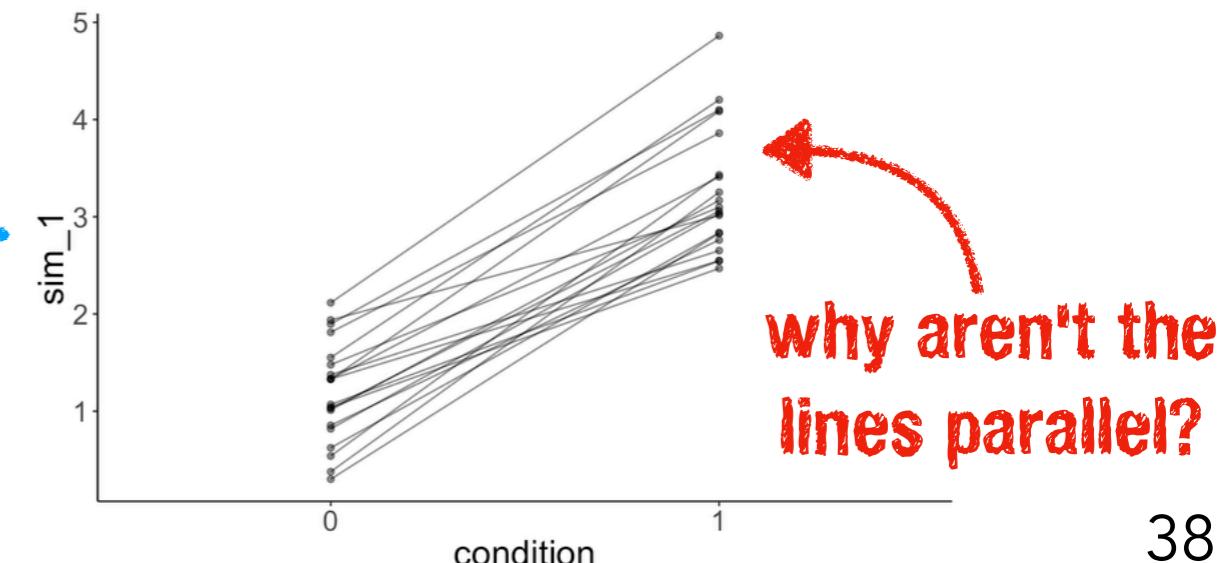
Random effects:
 Groups   Name        Variance Std.Dev. 
 participant (Intercept) 0.1702   0.4125  
 Residual           0.2270   0.4764  
Number of obs: 40, groups: participant, 20

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 
condition1   -0.535
```

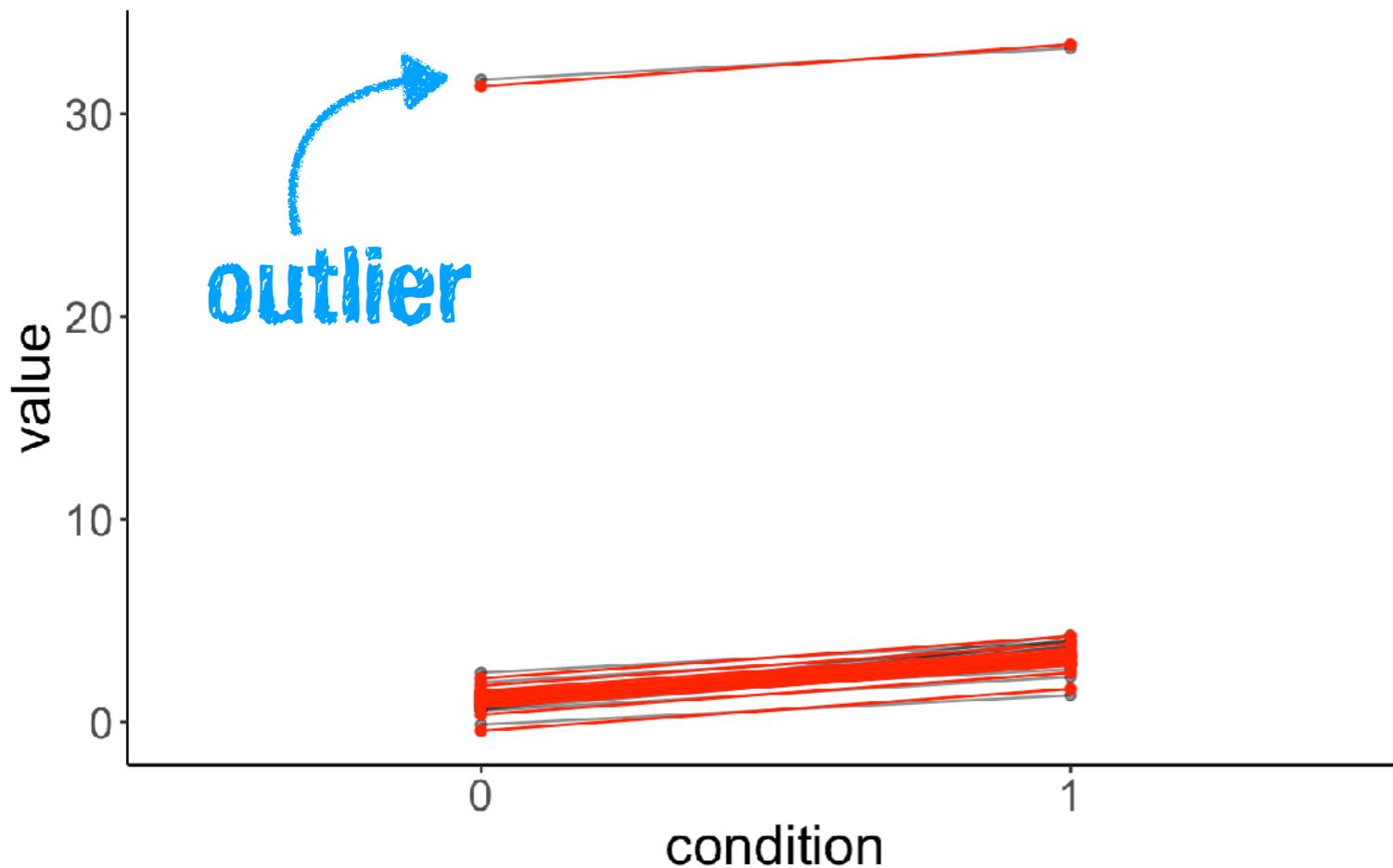
```
1 # fit model
2 fit.test = lmer(formula = value ~ 1 + condition + (1 | participant),
3                  data = df.test)
4
5 # simulate data
6 fit.test %>%
7   simulate()
```

simulated data



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 

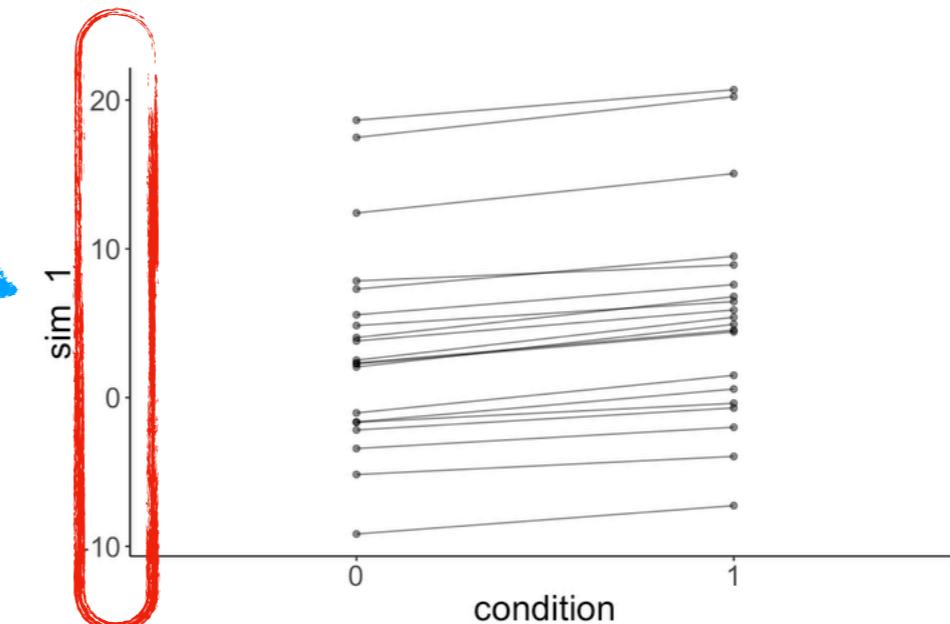
Random effects:
 Groups   Name        Variance Std.Dev. 
 participant (Intercept) 46.198   6.7969 
 Residual           0.227   0.4764 
Number of obs: 40, groups: participant, 20

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

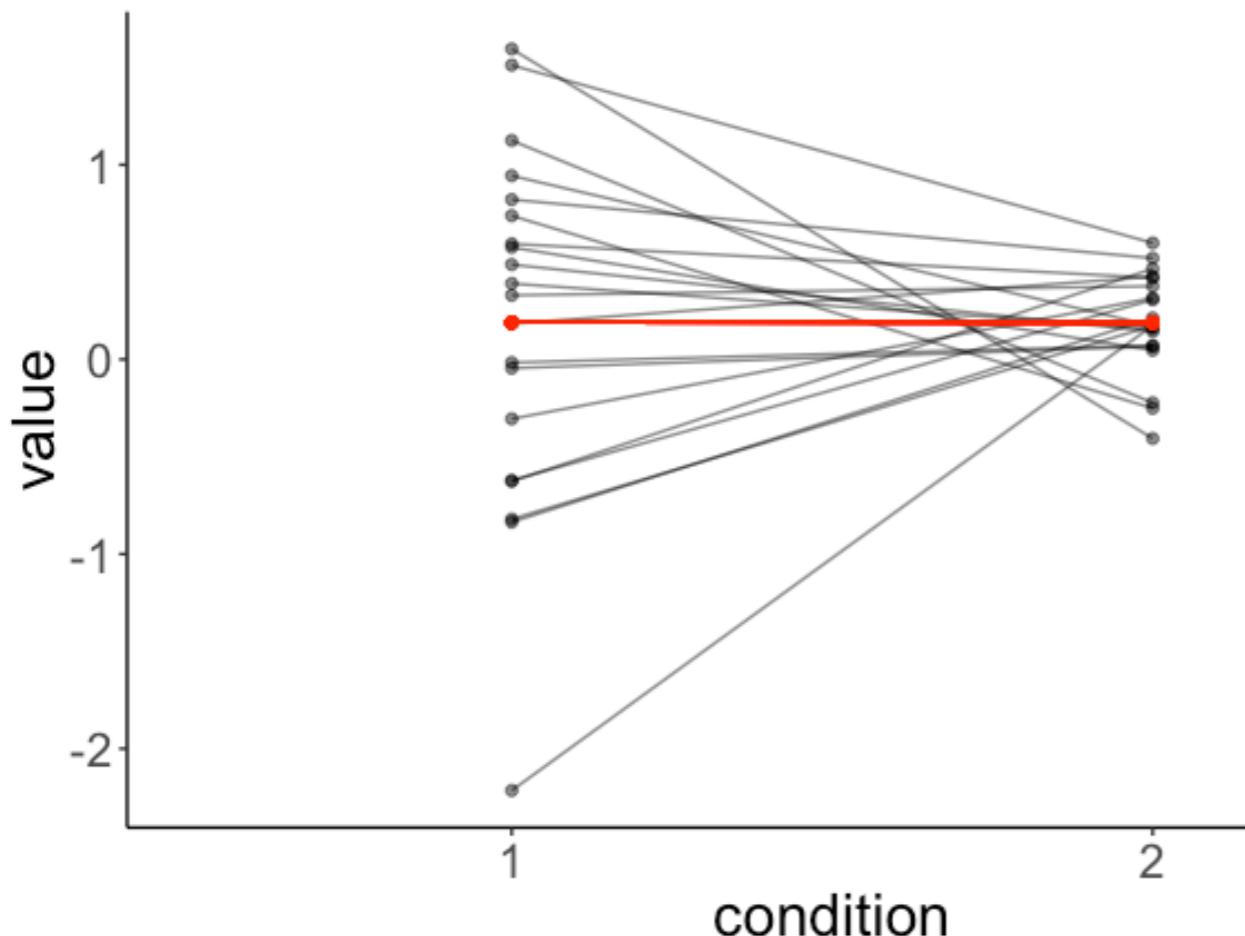
Correlation of Fixed Effects:
          (Intr) condition1 
condition1 -0.049
```

```
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



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 

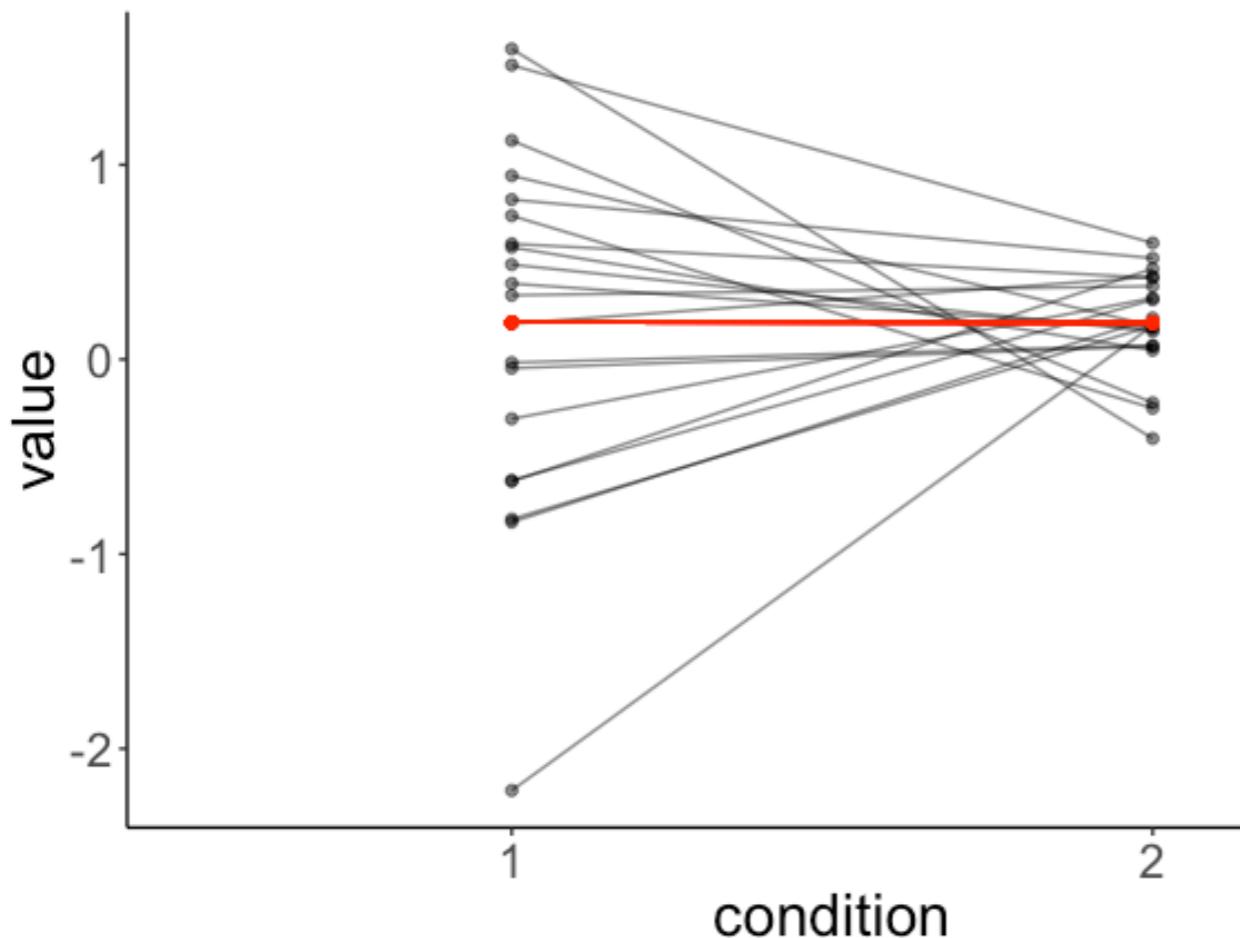
Random effects:
Groups      Name        Variance Std.Dev. 
participant (Intercept) 0.0000   0.0000  
Residual           0.4512   0.6717  
Number of obs: 40, groups: participant, 20

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

clearly there are interindividual differences though!?

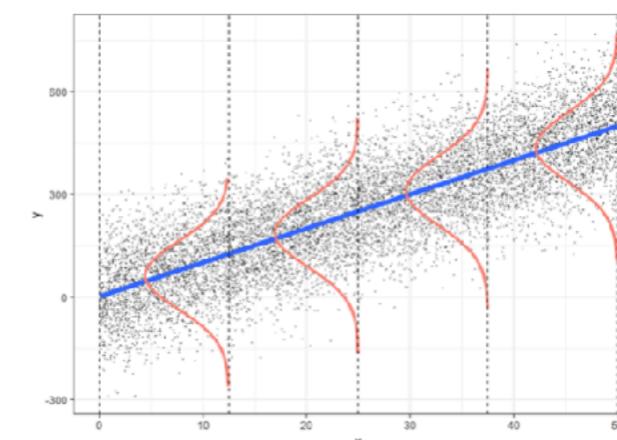
Non-equal variance



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



```
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
```

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

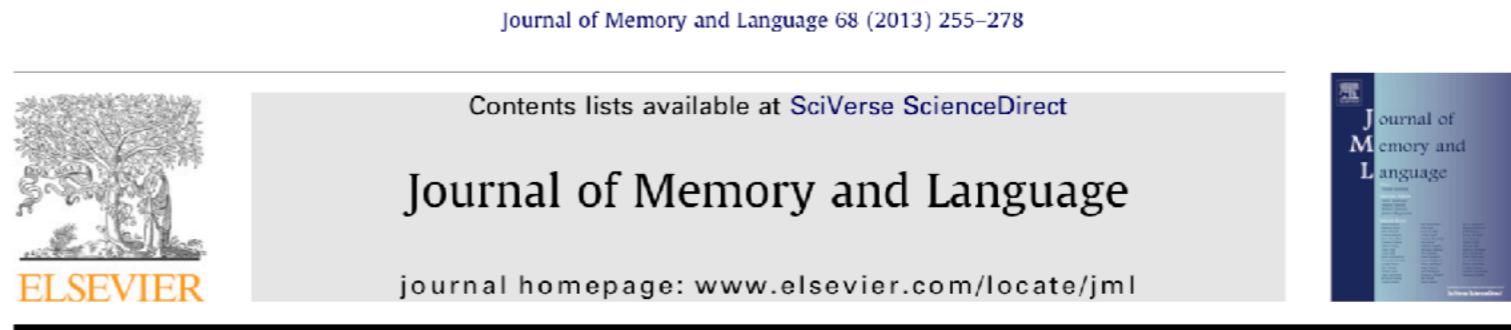
Content/Analysis Steps

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

[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)

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?

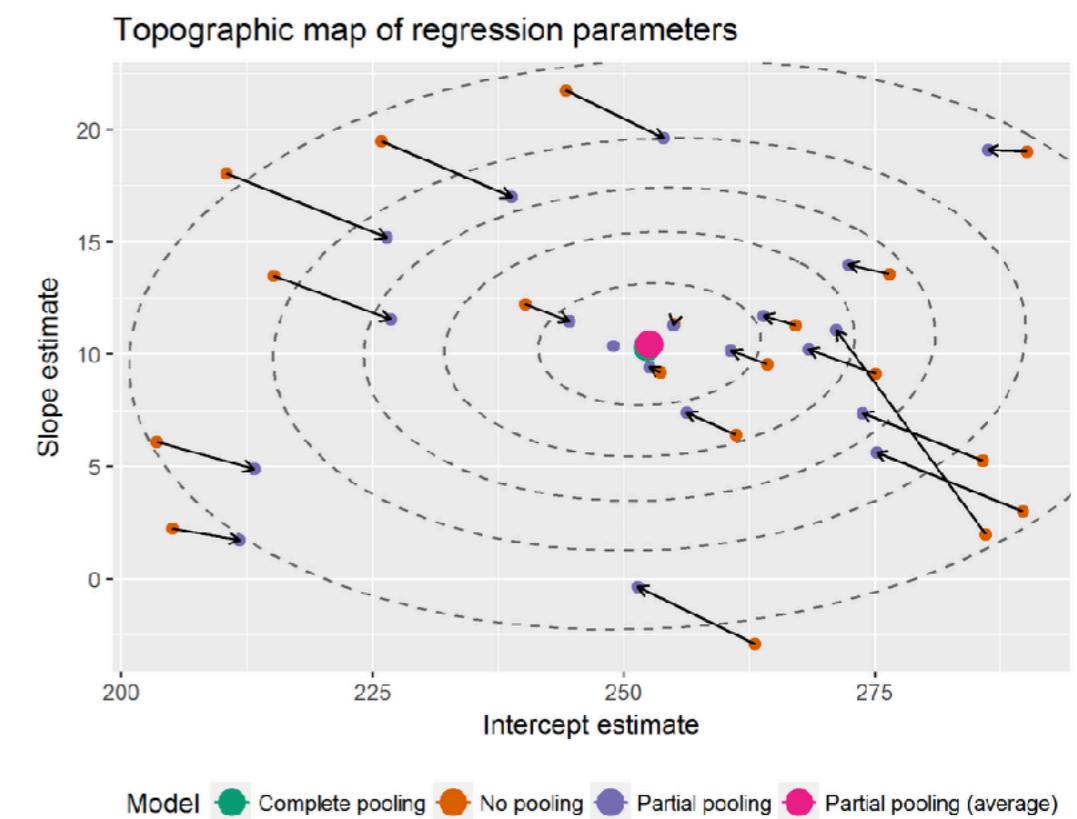
- general advice:
 - start maximal (as supported by the design)
 - random intercepts for different participants
 - random slopes when participants are tested multiple times
 - random intercepts for items
 - reduce complexity of the random effects structure step by step
 - remove the correlation between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

Remove the correlation component from your model

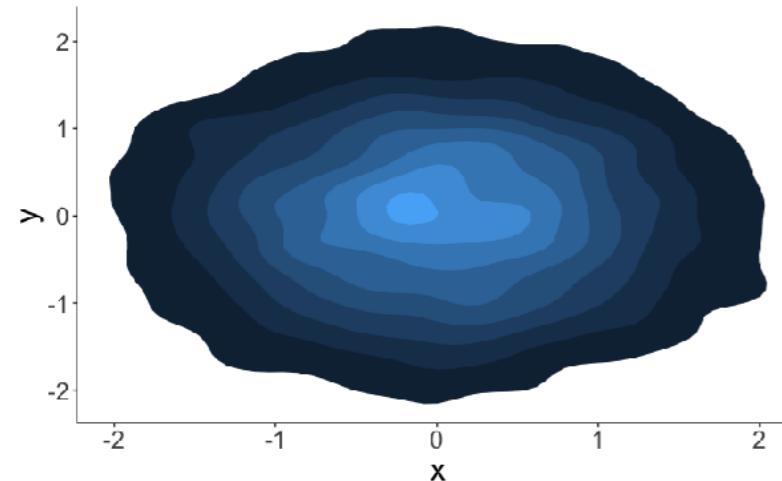
```
1 # fit the model  
2 fit.lmer = lmer(formula = reaction ~ 1 + days + (1 + days | subject),  
3                   data = df.sleep)  
4 # model summary  
5 fit.lmer %>%  
6   summary()
```

```
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
```

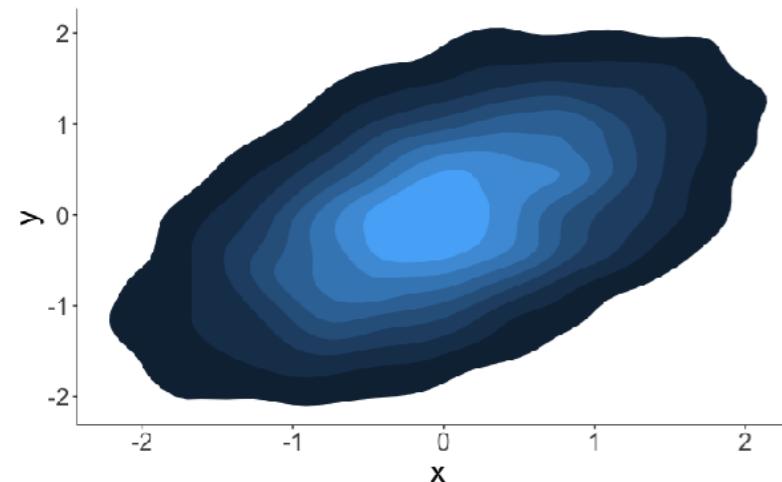
multivariate Gaussian



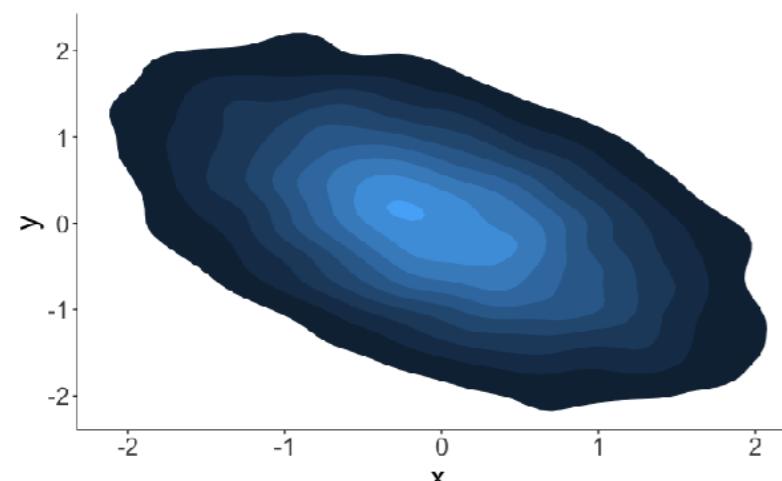
Remove the correlation component from your model



uncorrelated



positively correlated



negatively correlated

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 

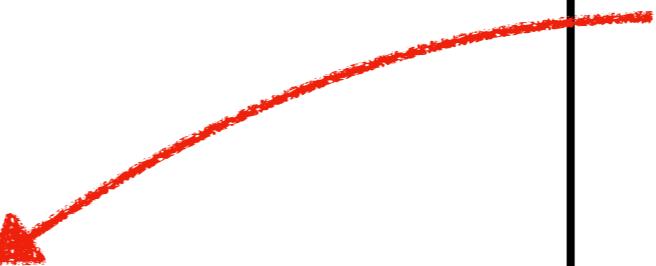
Random effects:
 Groups   Name        Variance Std.Dev.    
subject  days       35.88    5.99      
subject.1 (Intercept) 598.11   24.46    
Residual           647.90   25.45    
Number of obs: 183, groups: subject, 20

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
```

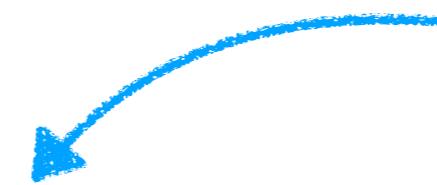
alternative syntax (doesn't model correlation between random effects)

independent Gaussians

What if lmer() fails to converge?

```
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

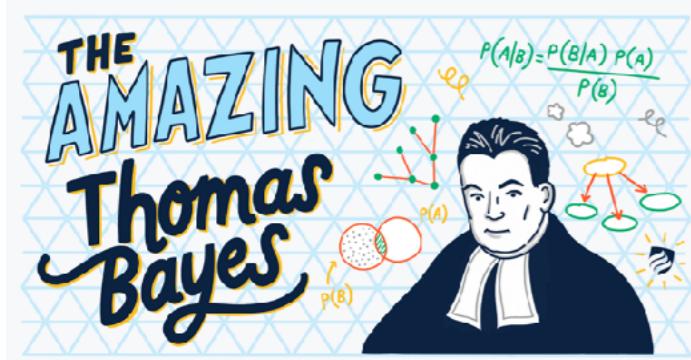
\$llik	bobyqa	Nelder_Mead	nlminbwrap
	-885.7239	-885.7239	-885.7239
	nloptwrap.NLOPT_LN_NELDERMEAD	nloptwrap.NLOPT_LN_BOBYQA	

\$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

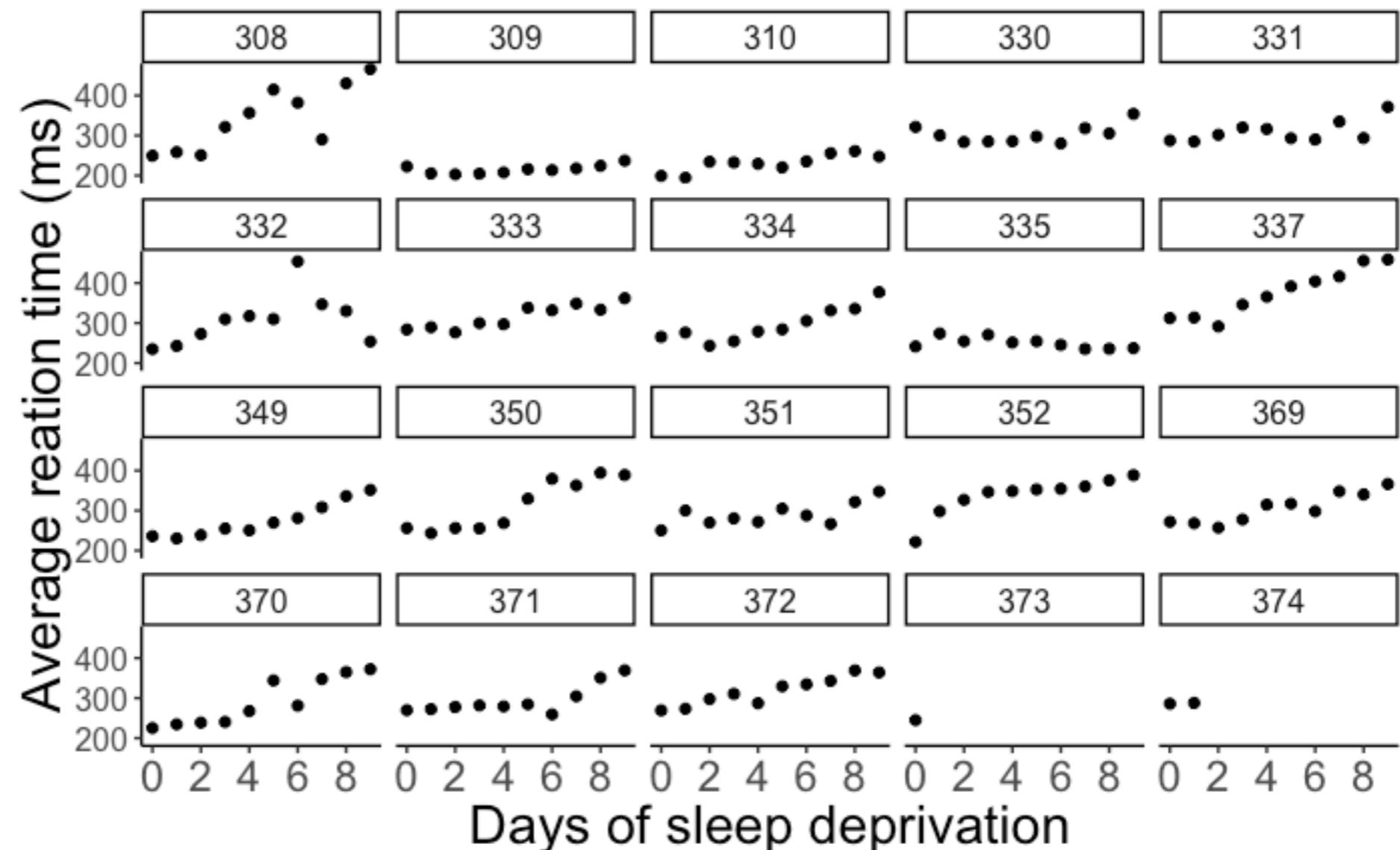
1. Sleep



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



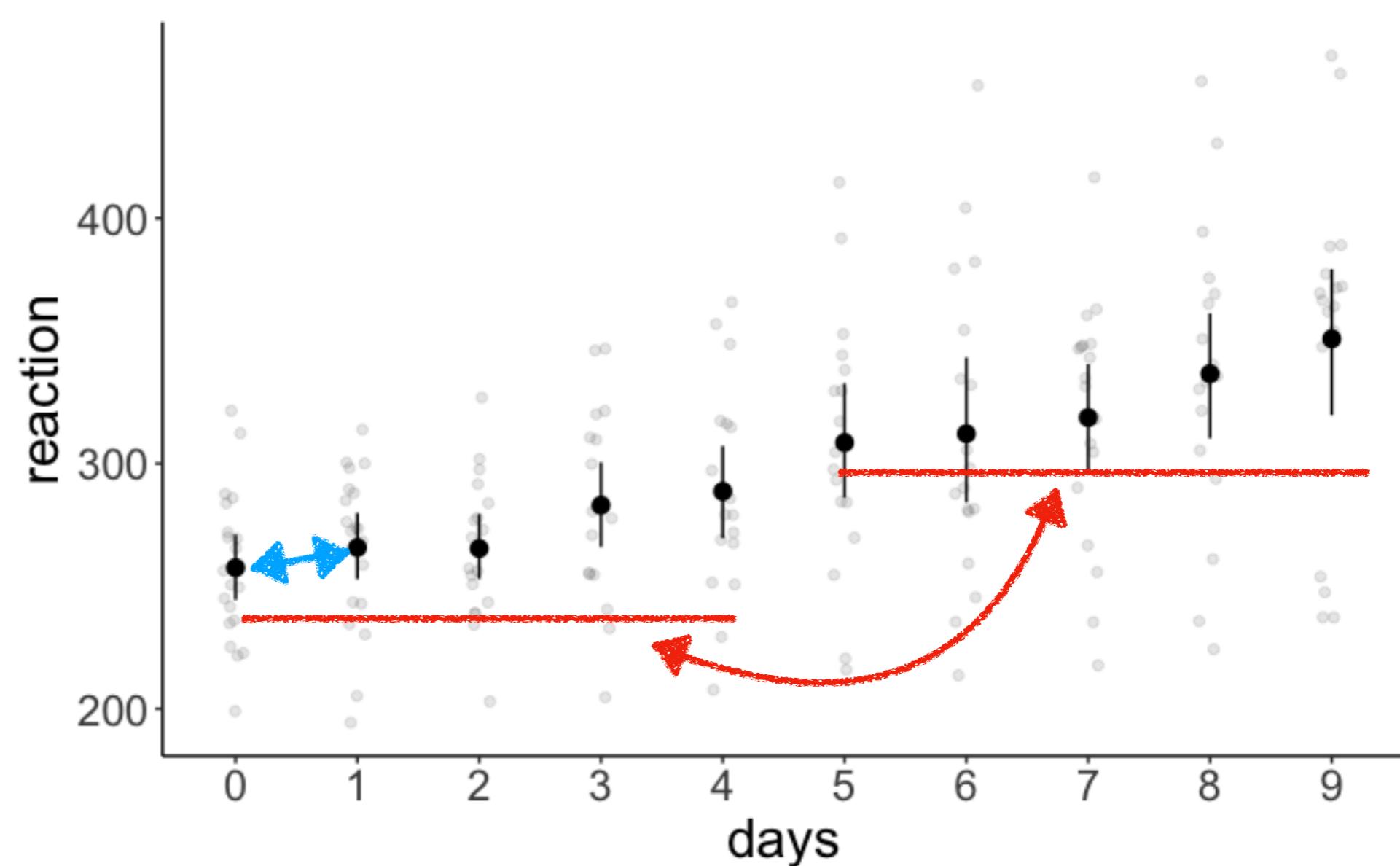
20 participants

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

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



Sleep data

fit the model

```
1 fit = lmer(formula = reaction ~ 1 + days + (1 | subject),  
2             data = df.sleep %>%  
3             mutate(days = as.factor(days)))  
4  
5 contrast = list(first_vs_second = c(-1, 1, rep(0, 8)),  
6                   early_vs_late = c(rep(-1, 5)/5, rep(1, 5)/5))  
7  
8 fit %>%  
9   emmeans(specs = "days",  
10            contr = contrast) %>%  
11  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

days	reaction
0	257.54
1	265.73

Degrees-of-freedom method: kenward-roger

index	reaction
early	271.67
late	325.39

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

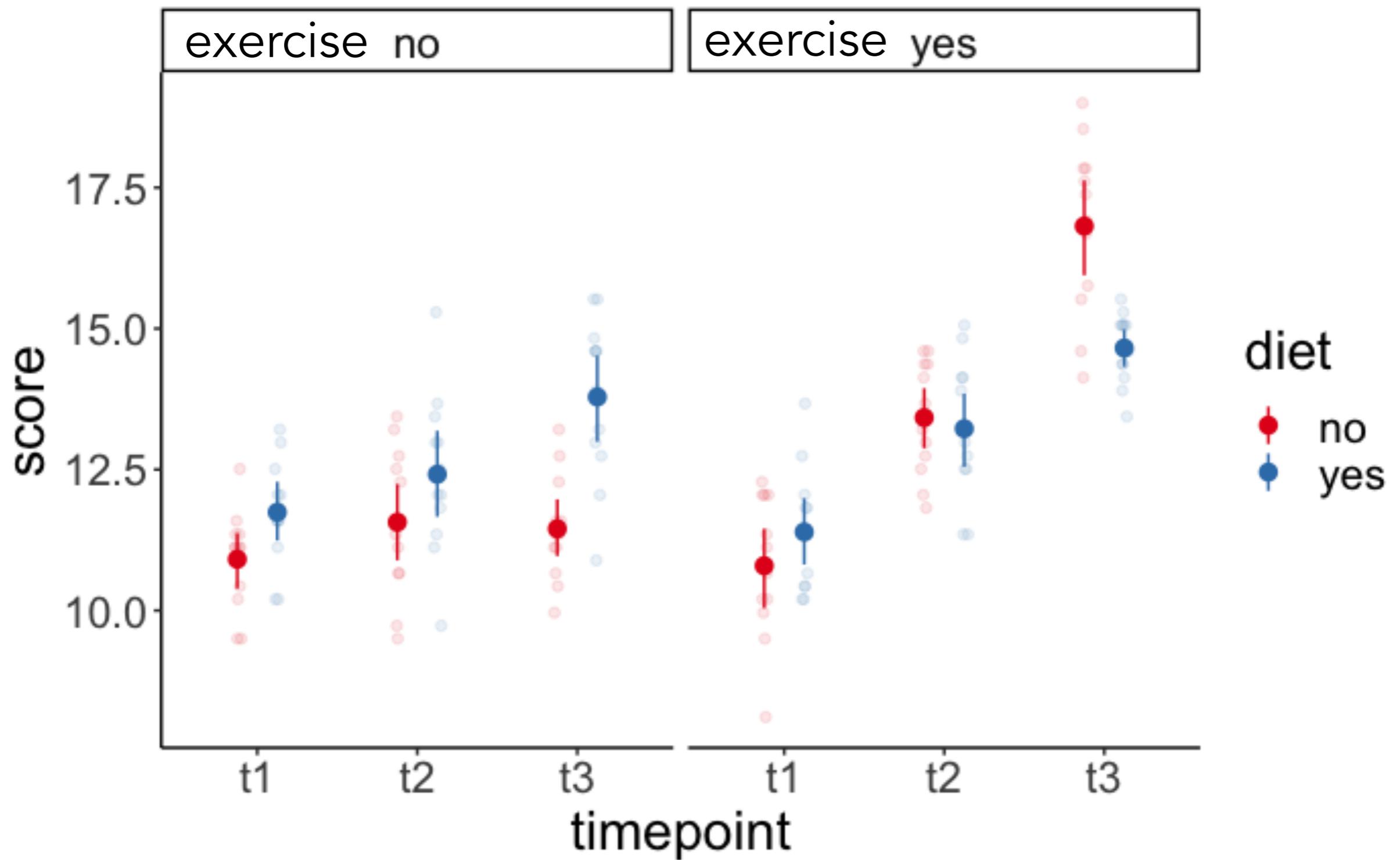
between participants: exercise yes/no

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

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	.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

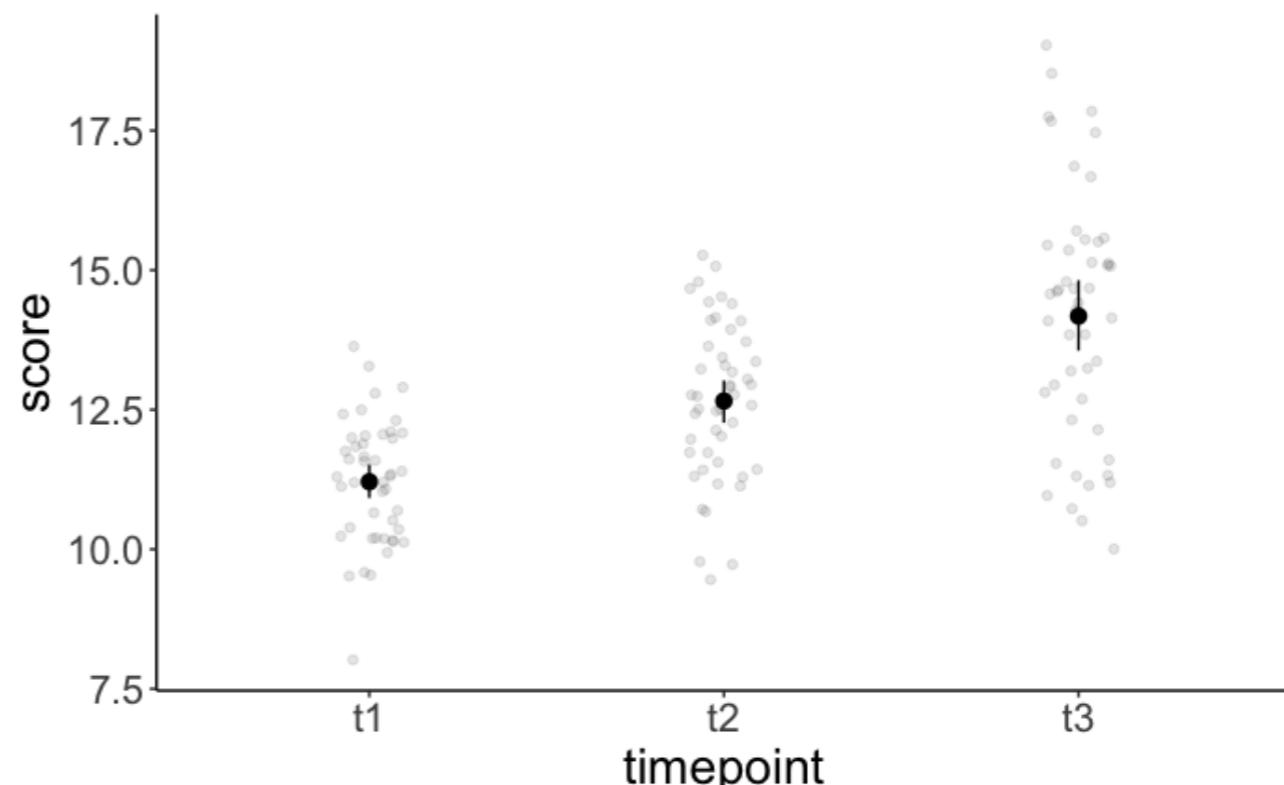
Weight loss data

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?

```
1 fit = aov_ez(id = "id",
2                 dv = "score",
3                 between = "exercises",
4                 within = c("diet", "timepoint"),
5                 data = df.weightloss)
6
7 contrasts = list(first_two_vs_last = c(-0.5, -0.5, 1),
8                   linear_increase = c(-1, 0, 1))
9
10 fit %>%
11   emmeans(spec = "timepoint",
12             contr = contrasts)
```

contrast	estimate	SE	df	t.ratio	p.value
first_two_vs_last	2.24	0.200	4	11.194	<.0001
linear_increase	2.97	0.231	4	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

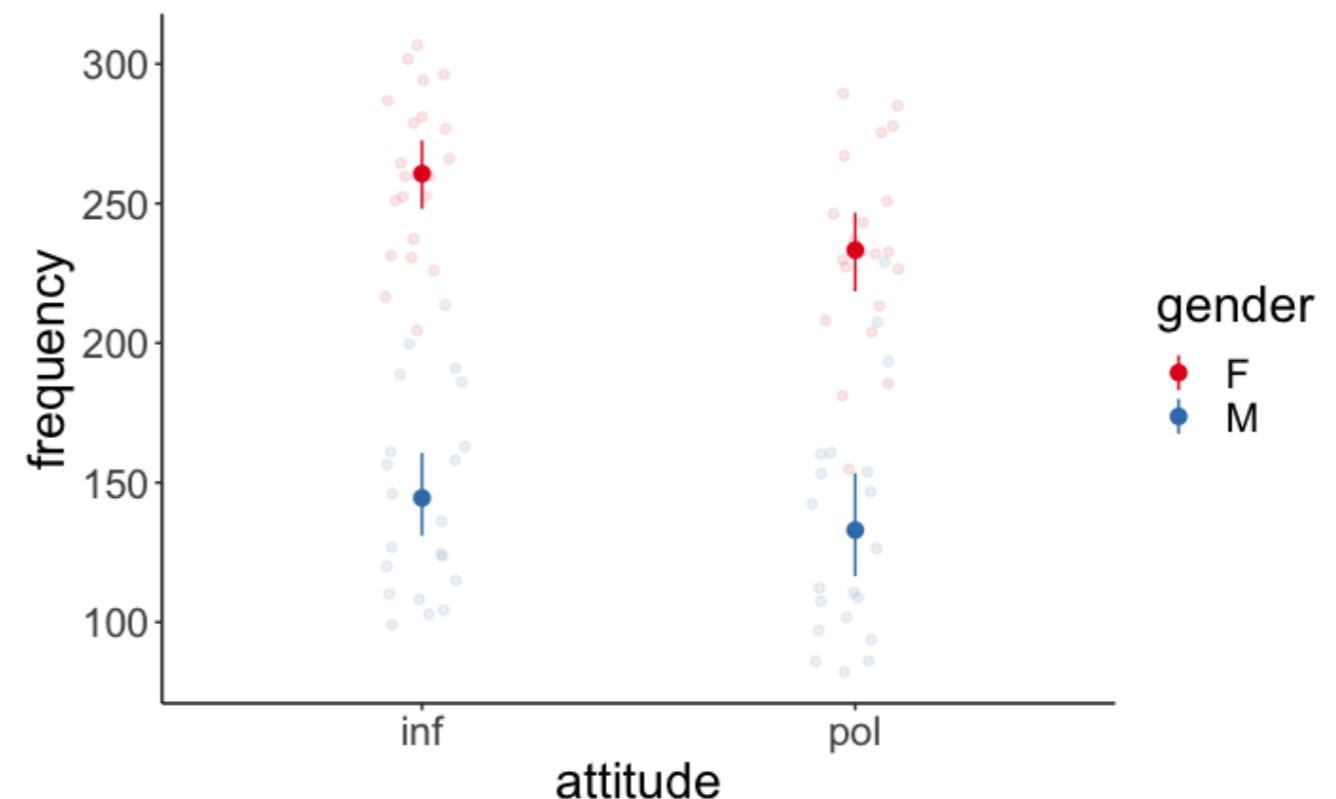
subject	gender	scenario	attitude	frequency
F1	F	1	pol	213.3
F1	F	1	inf	204.5
F1	F	2	pol	285.1
F1	F	2	inf	259.7
F1	F	3	pol	203.9
F1	F	3	inf	286.9
F1	F	4	pol	250.8
F1	F	4	inf	276.8
F1	F	5	pol	231.9
F1	F	5	inf	252.4
F1	F	6	pol	181.2
F1	F	6	inf	230.7
F1	F	7	inf	216.5
F1	F	7	pol	154.8
F3	F	1	pol	229.7

gender: female, male

scenario: different text prompt

attitude: informal vs. polite

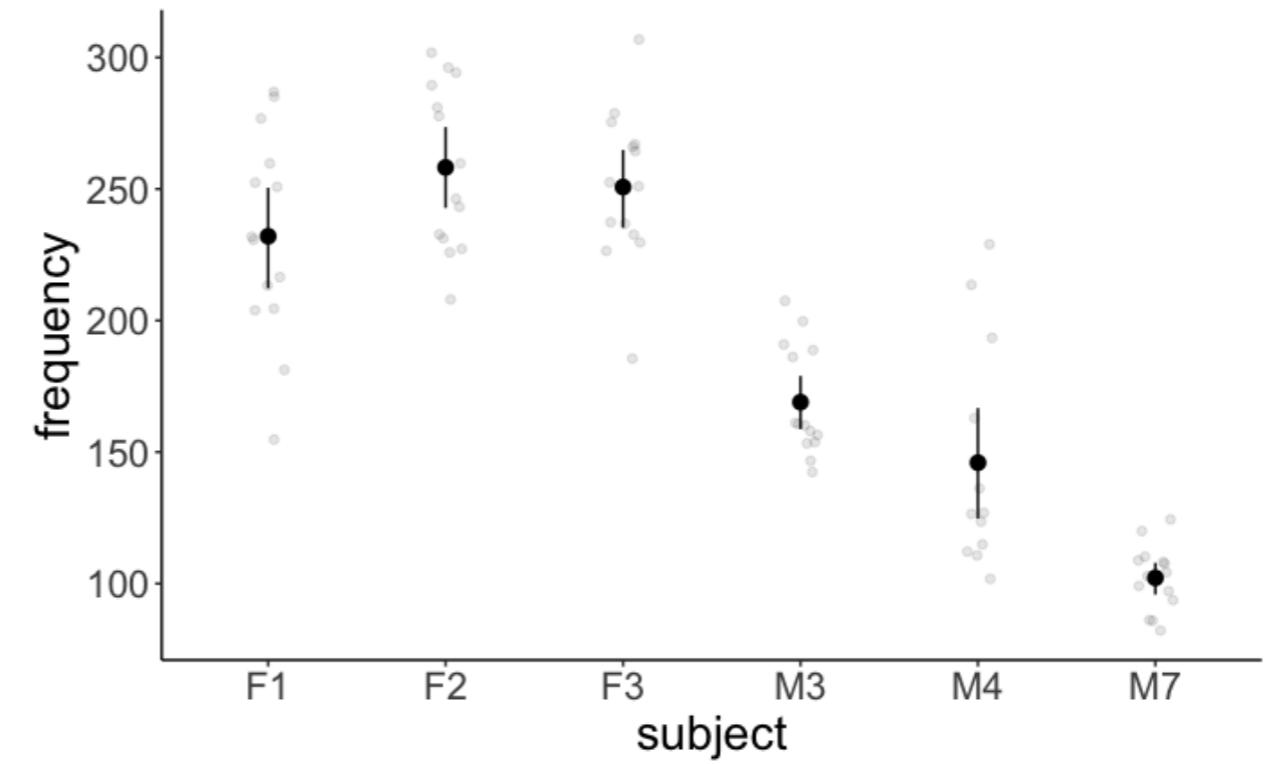
frequency: pitch of voice



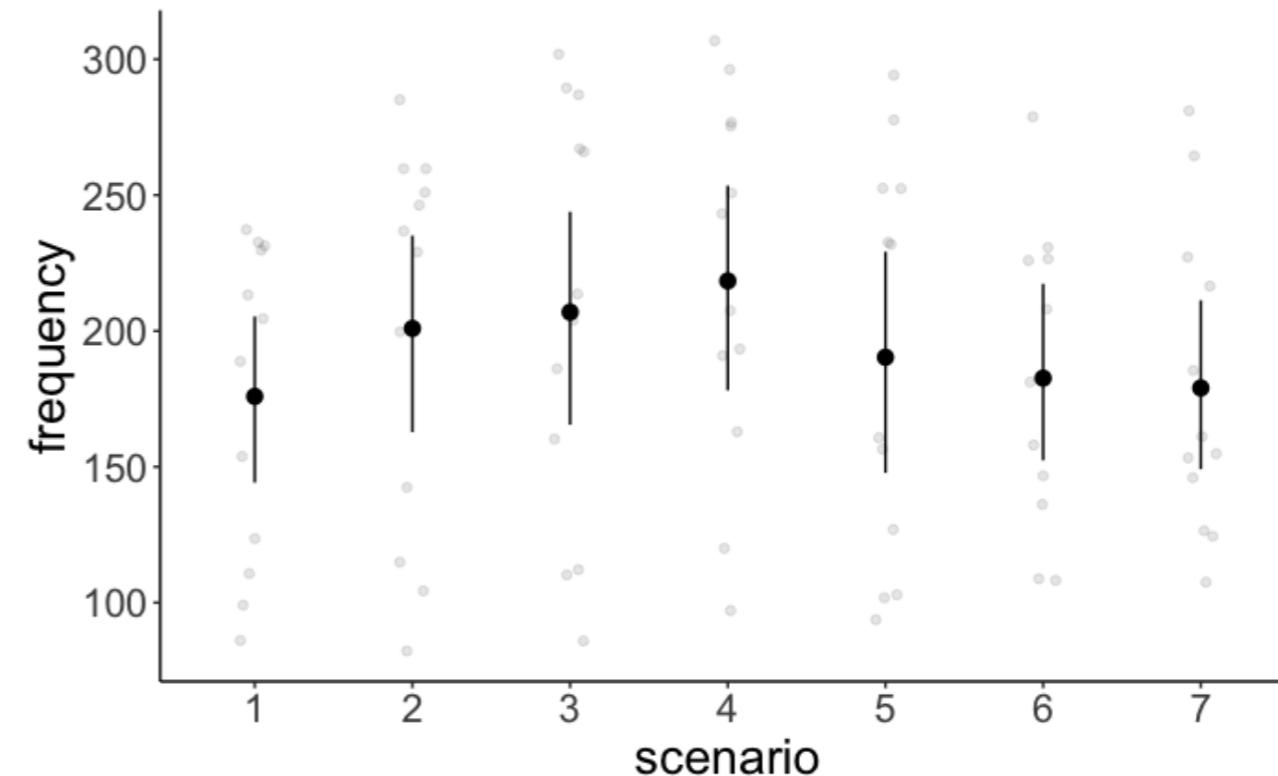
Politeness

variation across subjects

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 scenarios

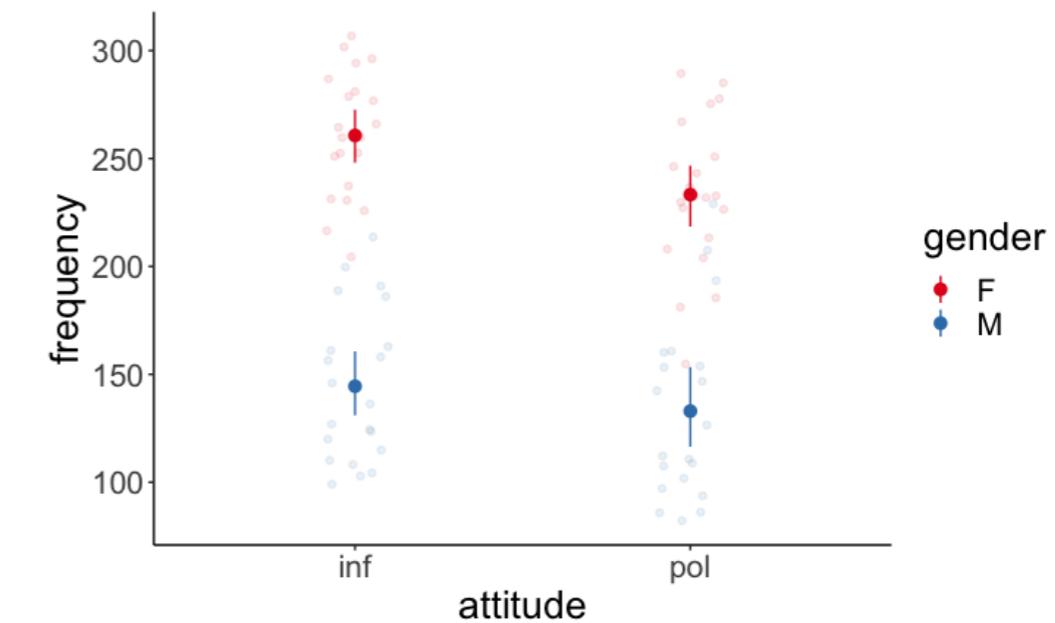


Politeness

Was there an effect of gender and attitude on pitch?

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

model term	df1	df2	F.ratio	p.value
attitude	1	3.99	12.411	0.0244
gender	1	4.00	26.570	0.0067
attitude:gender	1	3.99	1.959	0.2342



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

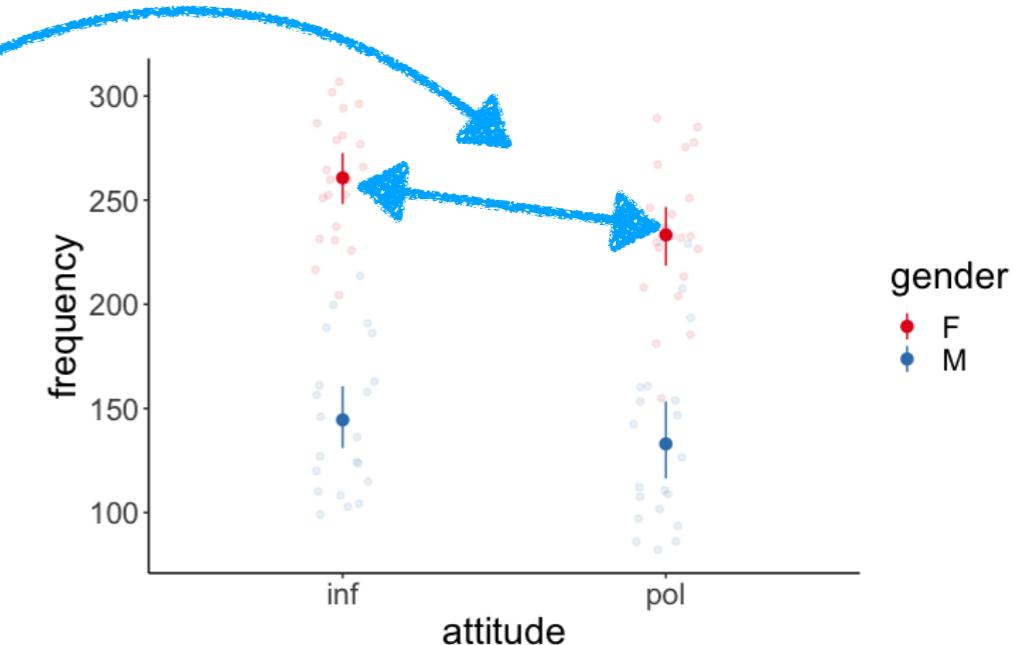
Politeness

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

```
1 fit = lmer(formula = frequency ~ 1 + attitude * gender + (1 | subject) + (1 | scenario),  
2             data = df.politeness)  
3  
4 fit %>%  
5   emmeans(specs = pairwise ~ attitude + gender,  
6             adjust = "none")
```

contrast	estimate	SE	df	t.ratio	p.value
inf F - pol F	27.4	7.81	3.89	3.508	0.0259
inf F - inf M	116.2	21.35	4.00	5.443	0.0055
inf F - pol M	128.0	21.78	4.57	5.879	0.0027
pol F - inf M	88.8	21.73	4.54	4.086	0.0116
pol F - pol M	100.6	22.16	4.00	4.541	0.0105
inf M - pol M	11.8	7.93	4.10	1.490	0.2088

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

```
1 aov_ez(id = "subject",
2         dv = "frequency",
3         between = "gender",
4         within = "attitude",
5         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
```

LMER

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

model term	df1	df2	F.ratio	p.value
attitude	1	3.99	12.411	0.0244
gender	1	4.00	26.570	0.0067
attitude:gender	1	3.99	1.959	0.2342

ignores variation between scenarios,
and just takes the mean

interaction effect

no interaction effect

Summary

- Quick recap
- Simpsons paradox
- Understanding `lmer()` syntax
- Reporting results
- Let's simulate some `lmer()`s
- `lmer()` standard operating procedures
- Some more examples

Feedback

How was the pace of today's class?

much a little just a little much
too too right too too
slow slow

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!