

# Linear mixed effects models 3,4



"Your recent Amazon purchases, Tweet score and location history makes you 23.5% welcome here."

COLLABORATIVE PLAYLIST  
**psych252**  
<https://tinyurl.com/psych252spotify25>

PLAY ...

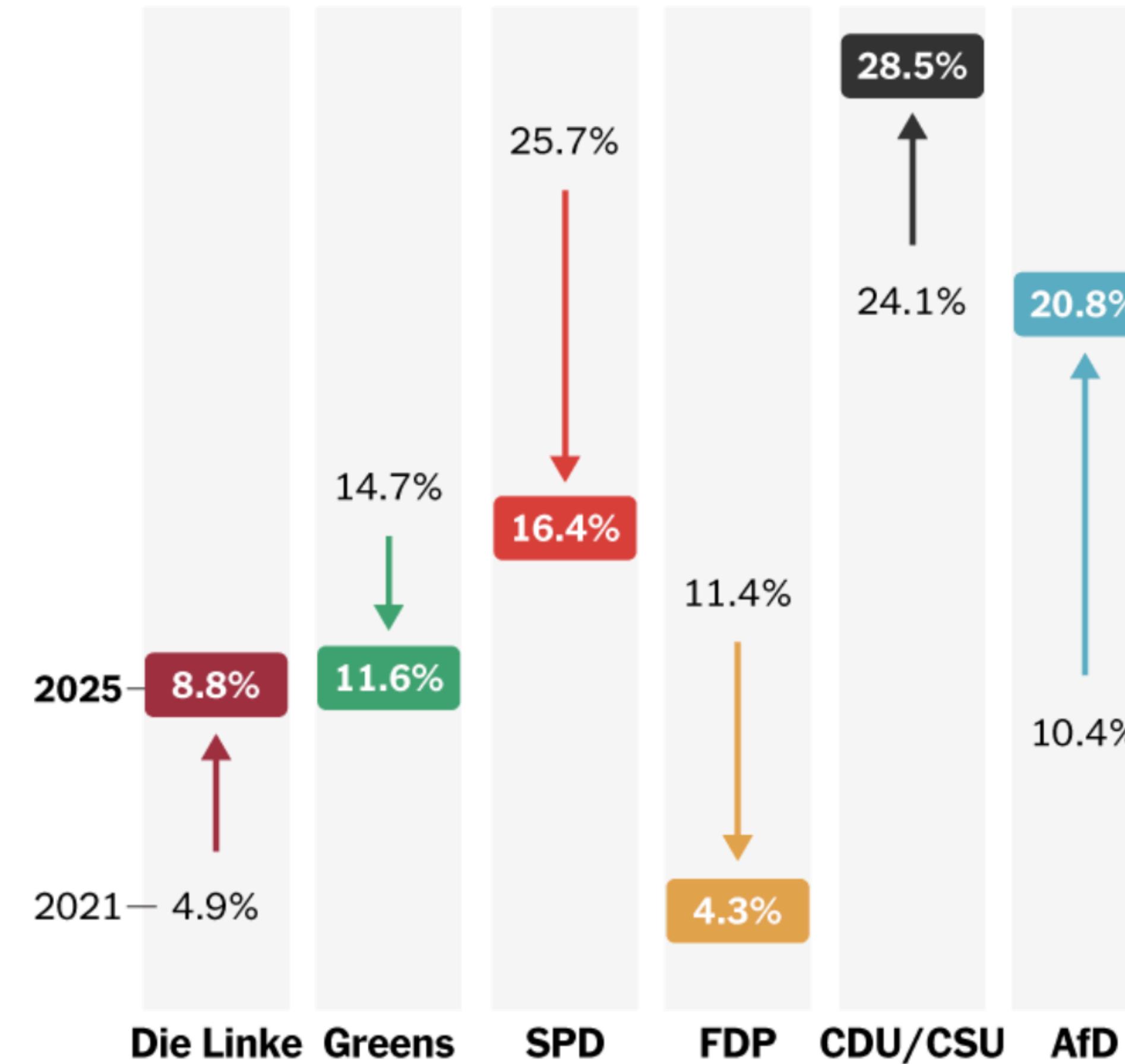
A screenshot of a Spotify collaborative playlist cover. The cover features a dark background with a stylized bar chart in light blue. A black smiley face with arms is superimposed on the bars. The title "psych252" is in large white font. Below it is the URL "https://tinyurl.com/psych252spotify25". At the bottom are two buttons: "PLAY" and "...".

# **Things that came up**

# Things that came up

## Maps: How Conservatives Regained Power

Change in vote share since 2021

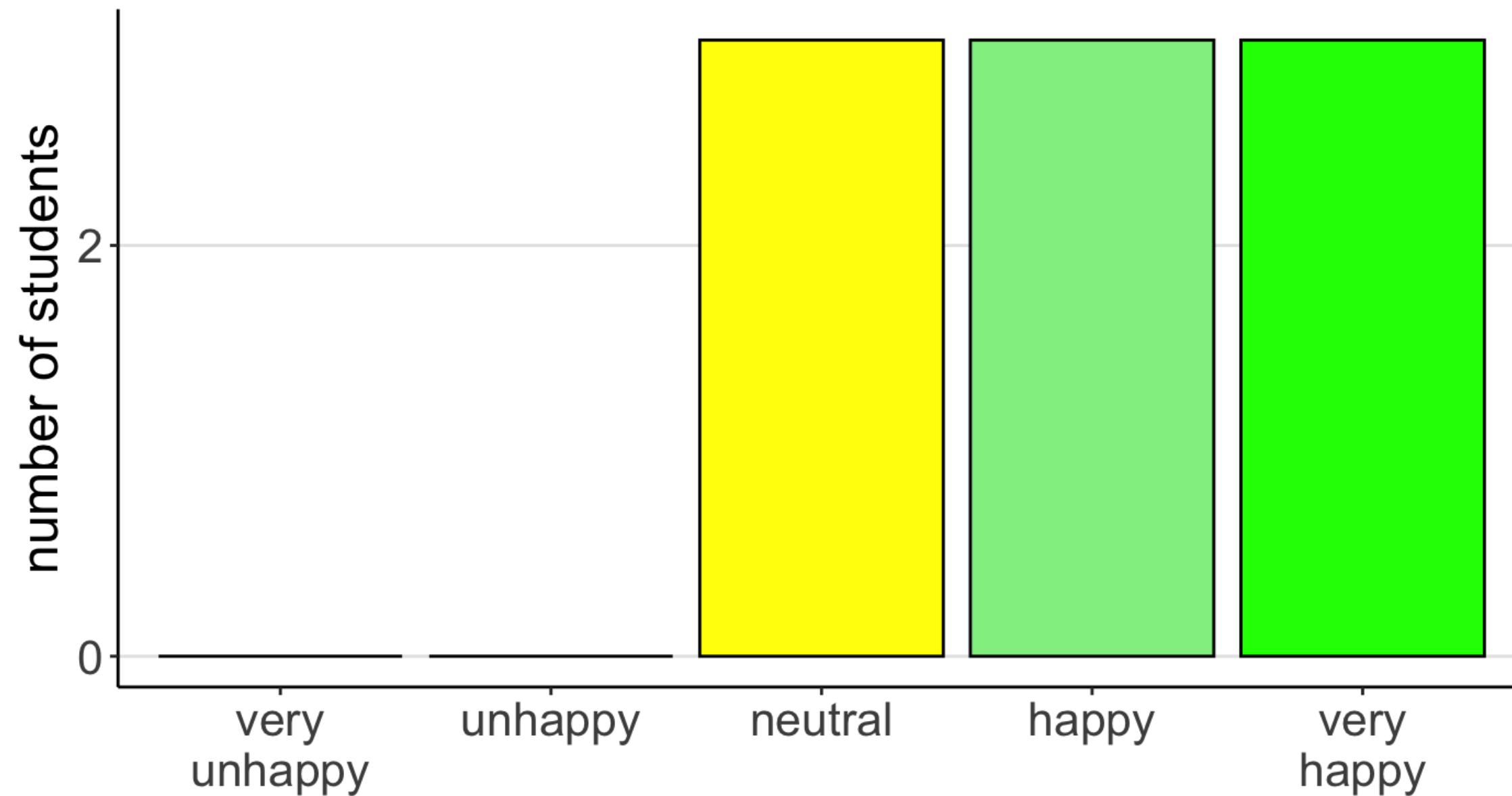


The New York Times

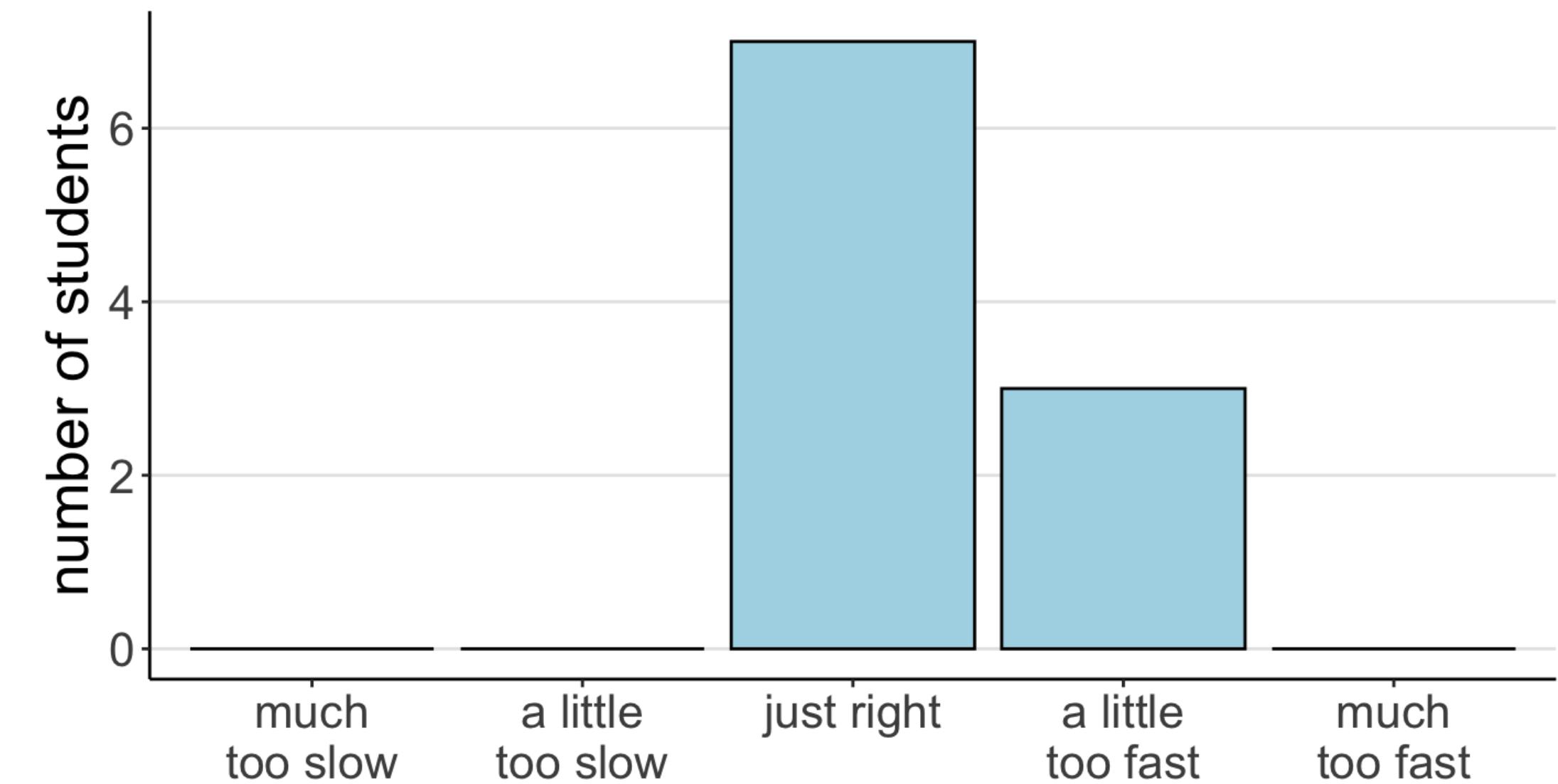
# **Feedback**

# Feedback

How happy were you with today's class overall?



How was the pace of today's class?



- I appreciate the enthusiasm! The topics are very challenging, so it would be nice if you could be a bit more precise. Thanks!!
- Would be great if we could include the interpretation of each parameters of the R results all in a separate document throughout the whole quarter
- It was very confusing, but the professor makes it very enjoyable
- I like the way you say zero

# Plan for today

- Quick recap
- Linear Mixed Model
  - Let's simulate some lmer()~~es~~
  - Some more examples
  - Different random effect structures
  - lmer() standard operating procedures

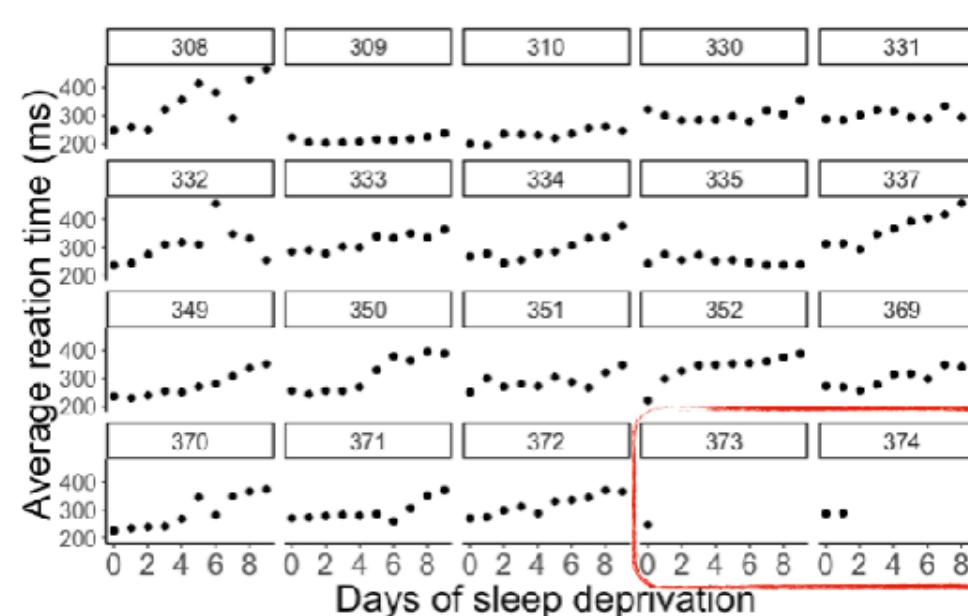
# Quick recap

# Quick recap: worked example

## Data set

### How does sleep deprivation affect reaction time?

subject	days	reaction
308	0	240.56
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	201.71
309	4	207.72



20 participants

2 with incomplete information

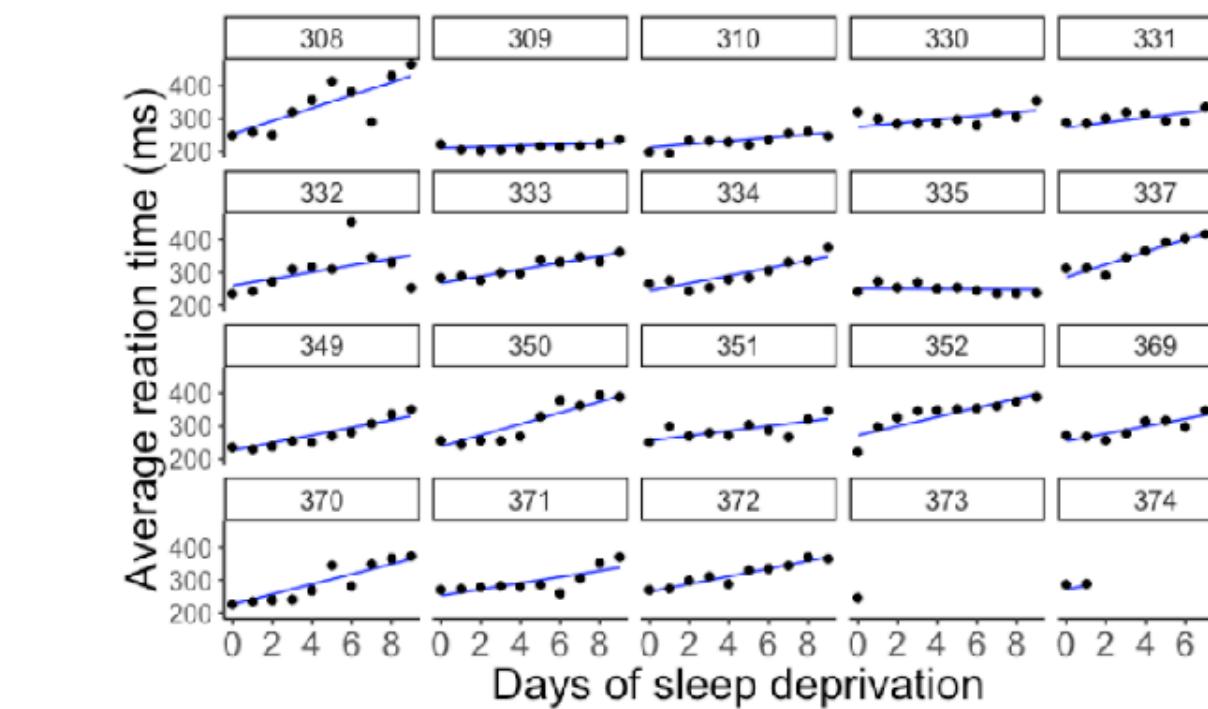
26

### Partial pooling: Fit mixed effects model

intercepts and slopes differ  
between participants

random intercept random slope

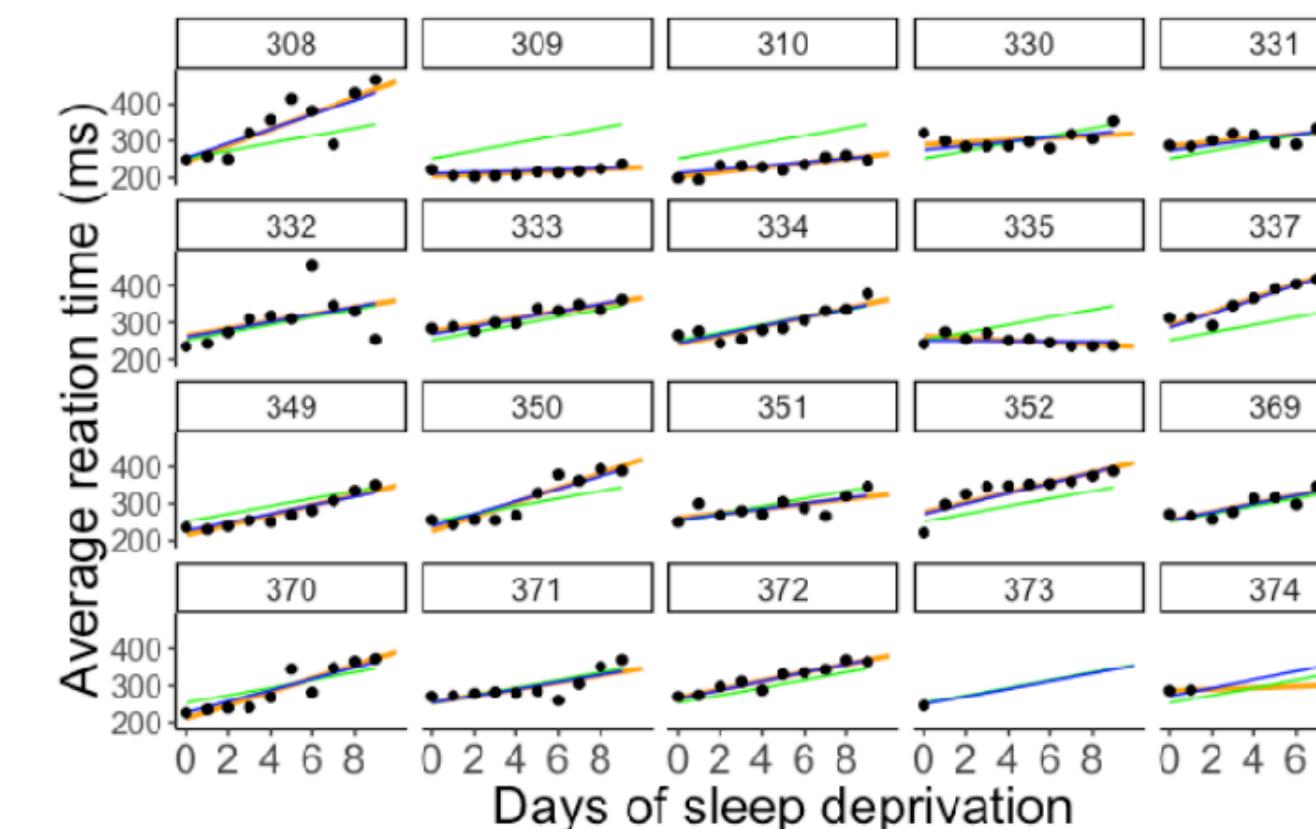
```
lmer(formula = reaction ~ 1 + days + (1 + days | subject),  
     data = df.sleep)
```



33

## Comparison

complete pooling  
no pooling  
partial pooling

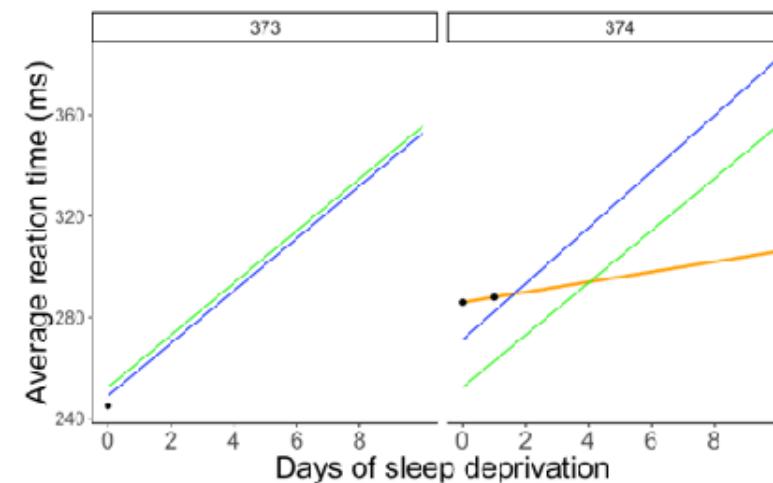


37

8

# Quick recap: shrinkage

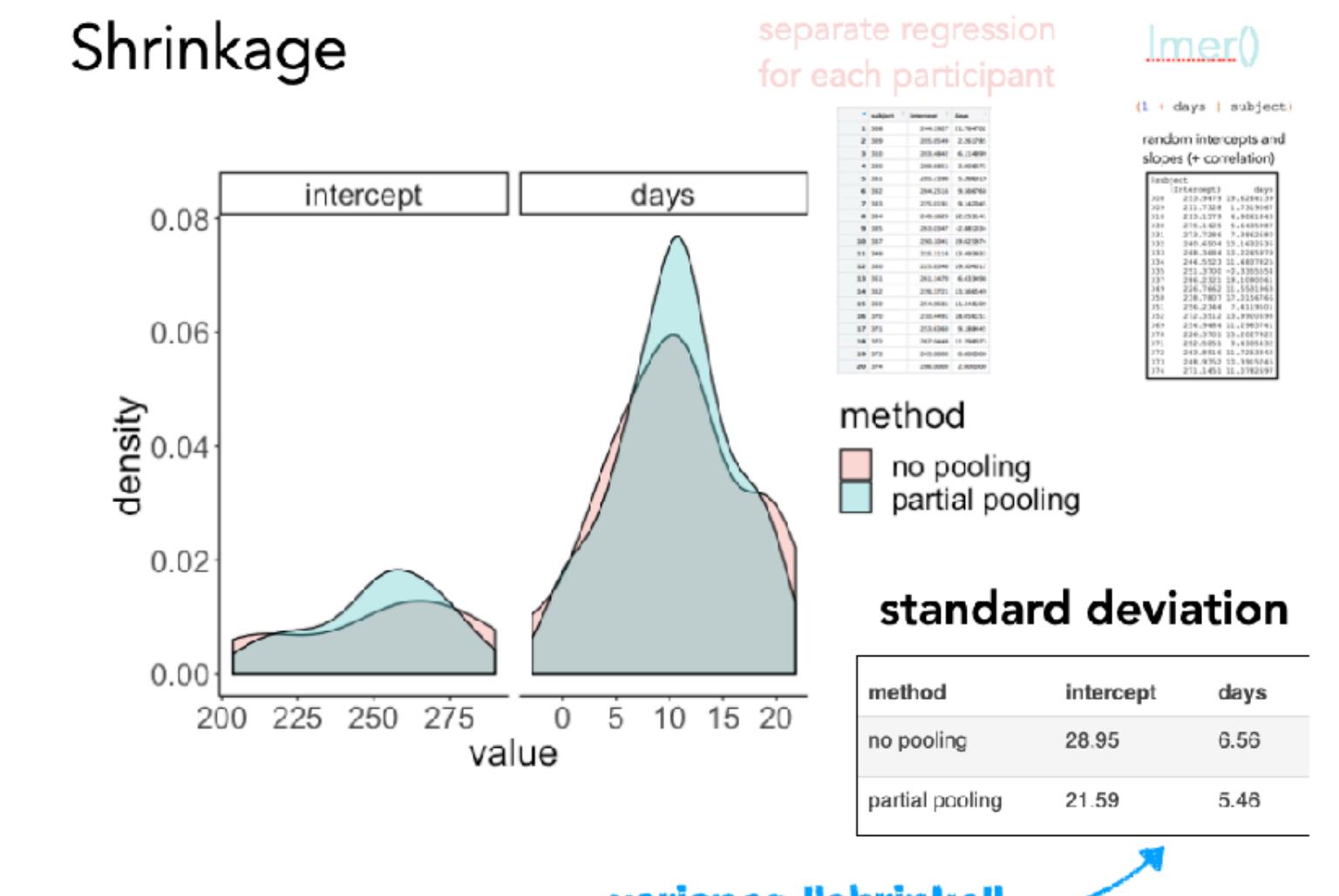
## Comparison



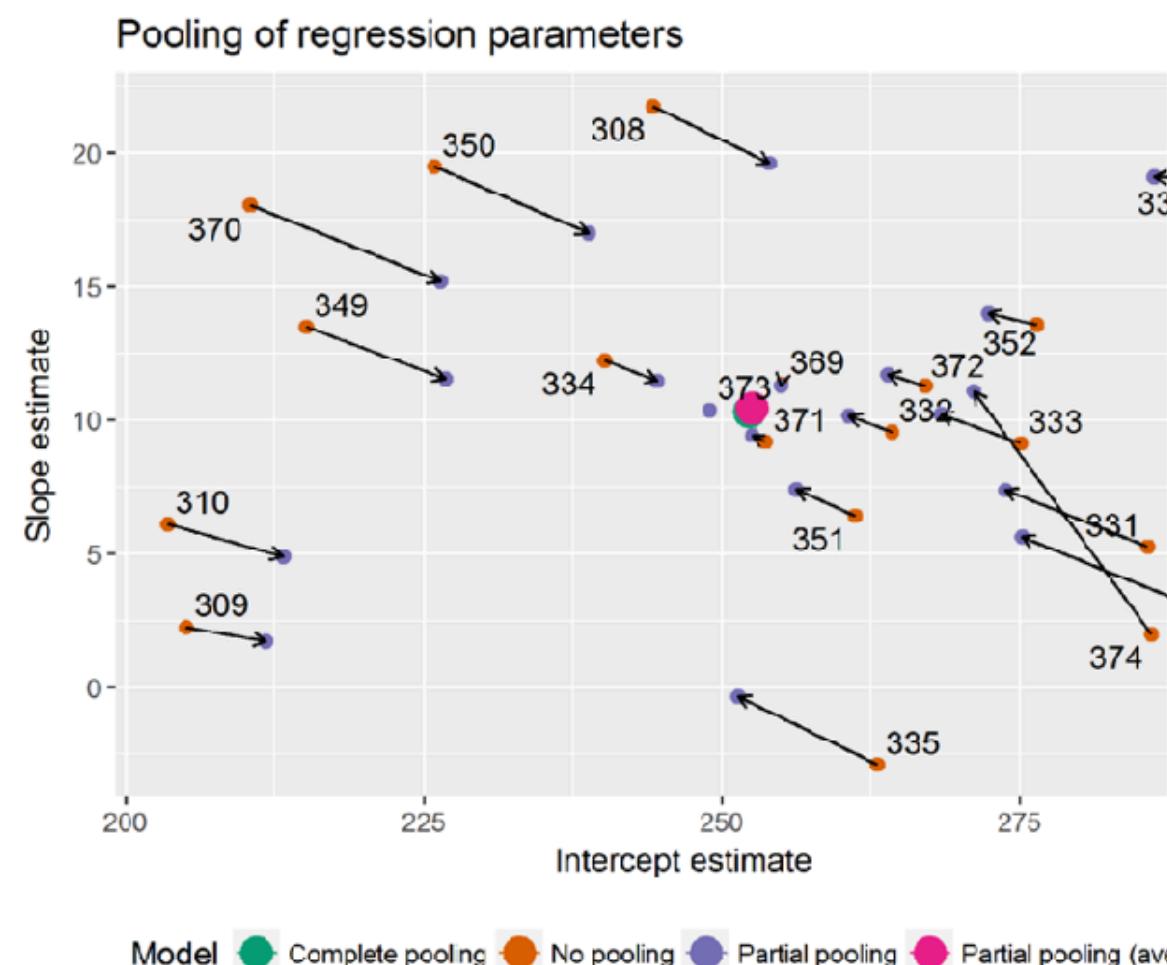
- complete pooling:**
  - doesn't account for any individual variation
- no pooling:**
  - doesn't yield predictions when we only have observation
  - doesn't consider the general effect of sleep deprivation when making predictions
- partial pooling:**
  - draws on all the information in the data
  -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

38

## Shrinkage



## Shrinkage



40

9

# Quick recap: Understanding lmer() syntax

## **lmer()** syntax summary

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

21

## Coefficients

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

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

### random intercepts

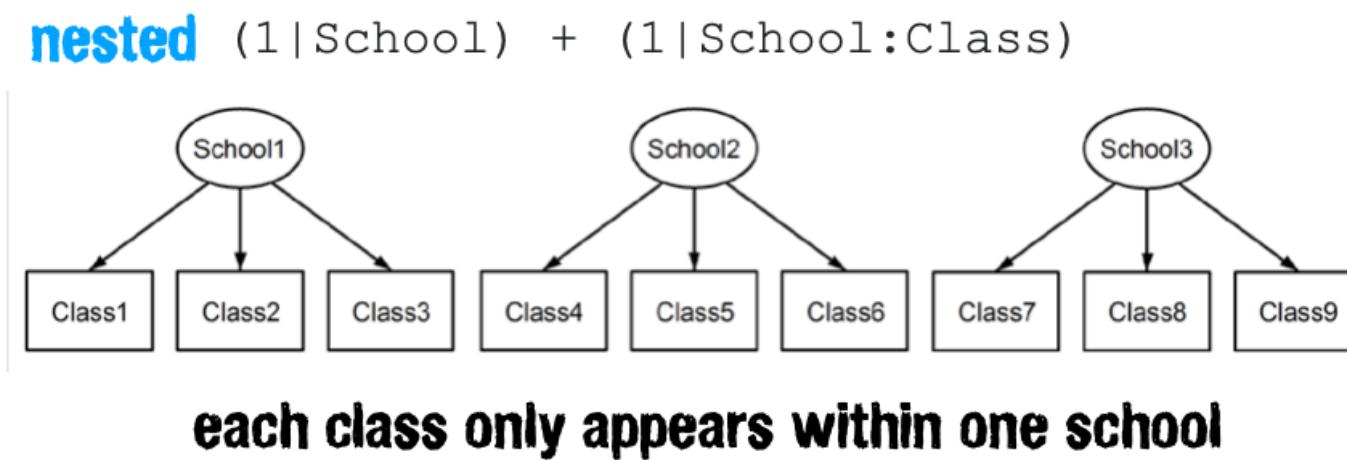
\$subject	(Intercept)	days
308	292.2749	10.43191
309	174.0559	10.43191
310	188.7454	10.43191
330	256.0247	10.43191
331	261.8141	10.43191
332	259.8262	10.43191
333	268.0765	10.43191
334	248.6471	10.43191
335	206.5096	10.43191
337	323.5643	10.43191
349	230.5114	10.43191
350	265.6957	10.43191
351	243.7988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

\$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	11.0907917

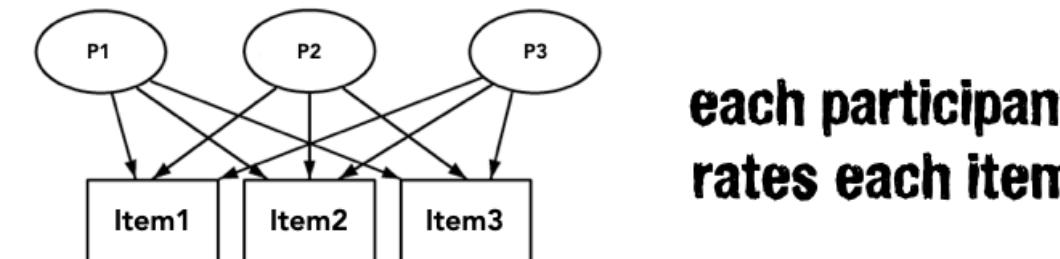
\$subject	(Intercept)	days
308	253.9479	19.6264139
309	211.7328	1.7319567
310	213.1579	4.9061843
330	275.1425	5.6435987
331	273.7286	7.3862680
332	260.6504	10.1632535
333	268.3684	10.2245979
334	244.5523	11.4837825
335	251.3700	-0.3355554
337	286.2321	19.1090061
349	226.7662	11.5531963
350	238.7807	17.0156766
351	256.2344	7.4119501
352	272.3512	13.9920698
369	254.9484	11.2985741
370	226.3701	15.2027922
371	252.5051	9.4353432
372	263.8916	11.7253342
373	248.9752	10.3915245
374	271.1451	11.0782697

22

## Multi-level models

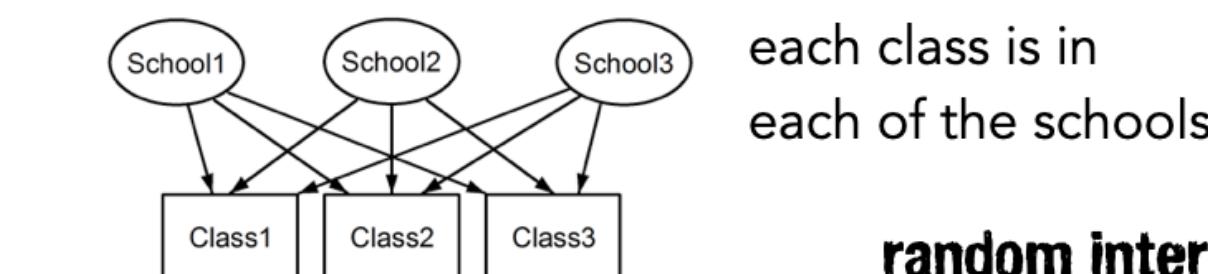


**crossed** `(1|participant) + (1|item)`



## Multi-level models

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



### 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()

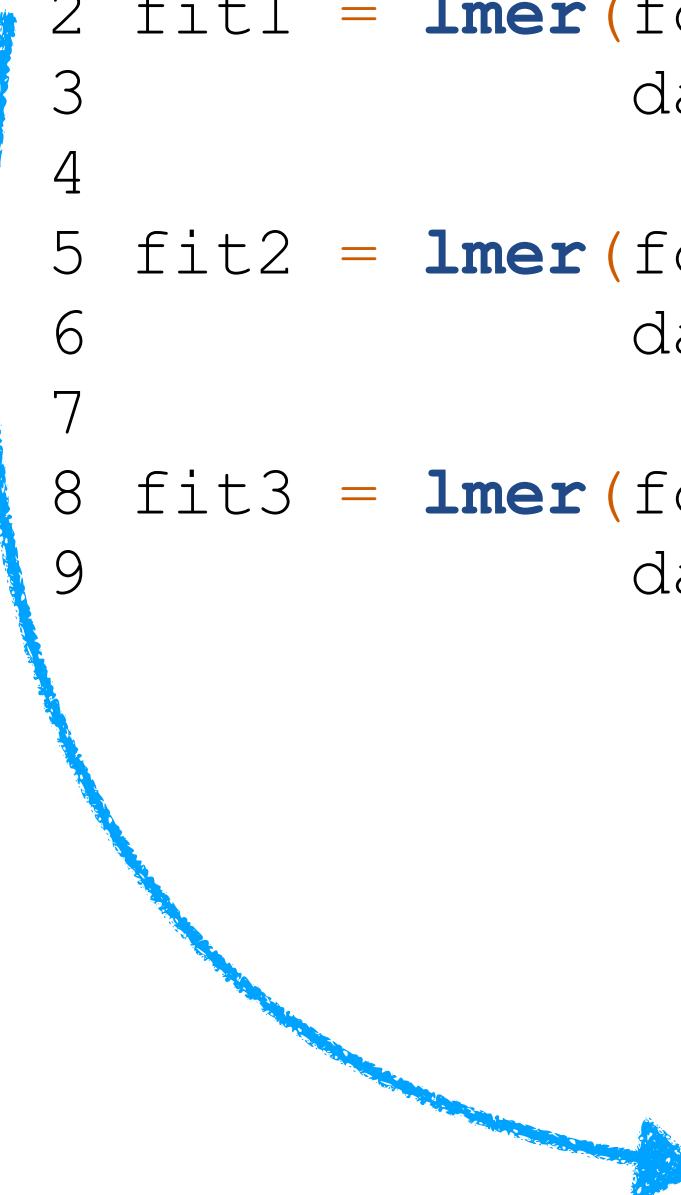
23

28

10

# Quick recap: Understanding lmer() syntax

```
1 # fit the model
2 fit1 = lmer(formula = value ~ 1 + condition + (1 | group) + (1 | group:participant),
3             data = df.dependence)
4
5 fit2 = lmer(formula = value ~ 1 + condition + (1 | group:participant),
6             data = df.dependence)
7
8 fit3 = lmer(formula = value ~ 1 + condition + (1 | participant),
9             data = df.dependence)
```



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

REML criterion at convergence: 89.9

Scaled residuals:
    Min      1Q  Median      3Q     Max 
-1.1247 -0.6901  0.1415  0.6592  1.0477 

Random effects:
 Groups           Name        Variance Std.Dev. 
group:participant (Intercept) 1.109e+00 1.053e+00
group            (Intercept) 6.235e-09 7.896e-05 
Residual          Residual   1.210e-01 3.479e-01 
Number of obs: 40, groups: group:participant, 20; group, 2

Fixed effects:
                Estimate Std. Error t value
(Intercept)     -0.1388    0.2480  -0.56 
conditioncondition2  0.6777    0.1100   6.16 

Correlation of Fixed Effects:
              (Intr) cndtn2 
cndtn1       -0.222
```

# Quick recap: Understanding lmer() syntax

```
1 # fit the model
2 fit1 = lmer(formula = value ~ 1 + condition + (1 | group) + (1 | group:participant),
3             data = df.dependence)
4
5 fit2 = lmer(formula = value ~ 1 + condition + (1 | group:participant),
6             data = df.dependence)
7
8 fit3 = lmer(formula = value ~ 1 + condition + (1 | participant),
9             data = df.dependence)
```

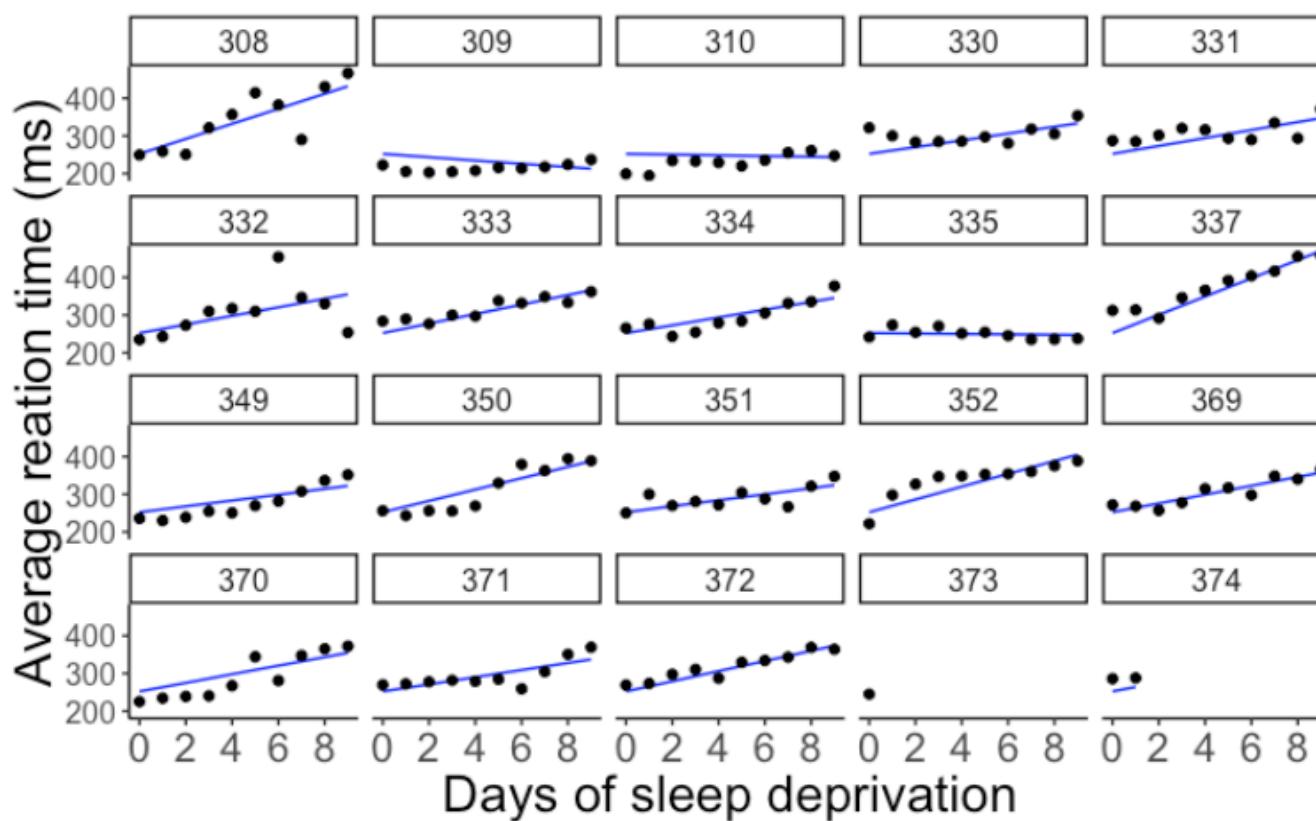
Linear mixed model fit by REML ['lmerMod']  
Formula: value ~ 1 + condition + (1 | group:participant)  
Data: df.dependence  
  
REML criterion at convergence: 89.9  
  
Scaled residuals:  
Min 1Q Median 3Q Max  
-1.1247 -0.6901 0.1415 0.6592 1.0477  
  
Random effects:  
Groups Name Variance Std.Dev.  
group:participant (Intercept) 1.109 1.0531  
Residual 0.121 0.3479  
Number of obs: 40, groups: group:participant, 20  
  
Fixed effects:  
Estimate Std. Error t value  
(Intercept) -0.1388 0.2480 -0.56  
conditioncondition2 0.6777 0.1100 6.16  
  
Correlation of Fixed Effects:  
(Intr) cndtn2 -0.222

Linear mixed model fit by REML ['lmerMod']  
Formula: value ~ 1 + condition + (1 | participant)  
Data: df.dependence  
  
REML criterion at convergence: 89.9  
  
Scaled residuals:  
Min 1Q Median 3Q Max  
-1.1247 -0.6901 0.1415 0.6592 1.0477  
  
Random effects:  
Groups Name Variance Std.Dev.  
participant (Intercept) 1.109 1.0531  
Residual 0.121 0.3479  
Number of obs: 40, groups: participant, 20  
  
Fixed effects:  
Estimate Std. Error t value  
(Intercept) -0.1388 0.2480 -0.56  
conditioncondition2 0.6777 0.1100 6.16  
  
Correlation of Fixed Effects:  
(Intr) cndtn2 -0.222

these two models are the same

# Quick recap: Reporting results

## Visualization

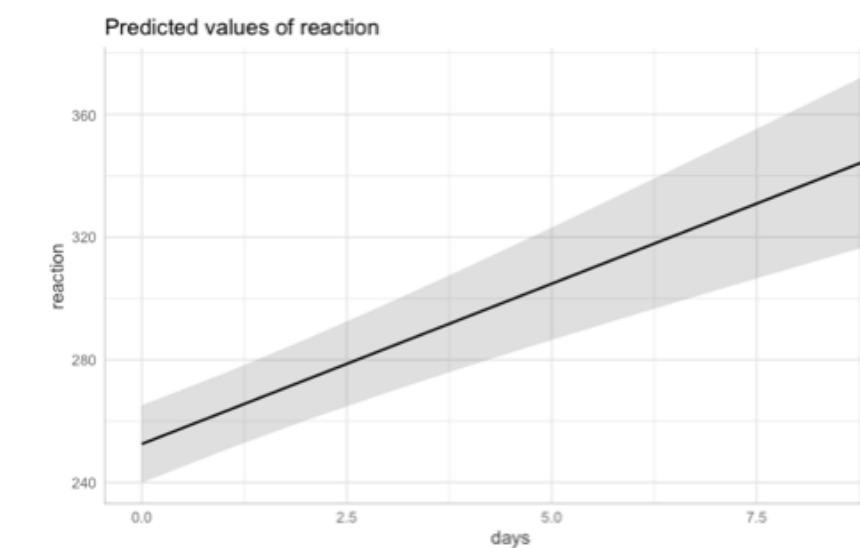


show the data together with the model predictions

30

## 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

31

## 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

[Meteyard, L. & Davies, R. A. I. \(2020\). Best practice guidance for linear mixed-effects models in psychological science. \*Journal of Memory and Language\*.](#) 32

## 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

<sup>7</sup>All 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.

13

**Let's simulate some lmer()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_intercept = 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(participant_intercepts = rnorm(n = 1, sd = sd_participant_intercept)) %>%
16 ungroup() %>%
17 mutate(value = b0 + b1 * condition + participant_intercept + rnorm(n()), sd = sd_residual)) %>%
18 arrange(participant, condition)
```

participant	condition	participant_intercept	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_intercept = 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(participant_intercept = rnorm(n = 1, sd = sd_participant_intercept)) %>%
16   ungroup() %>%
17   mutate(value = b0 + b1 * condition + participant_intercept + 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

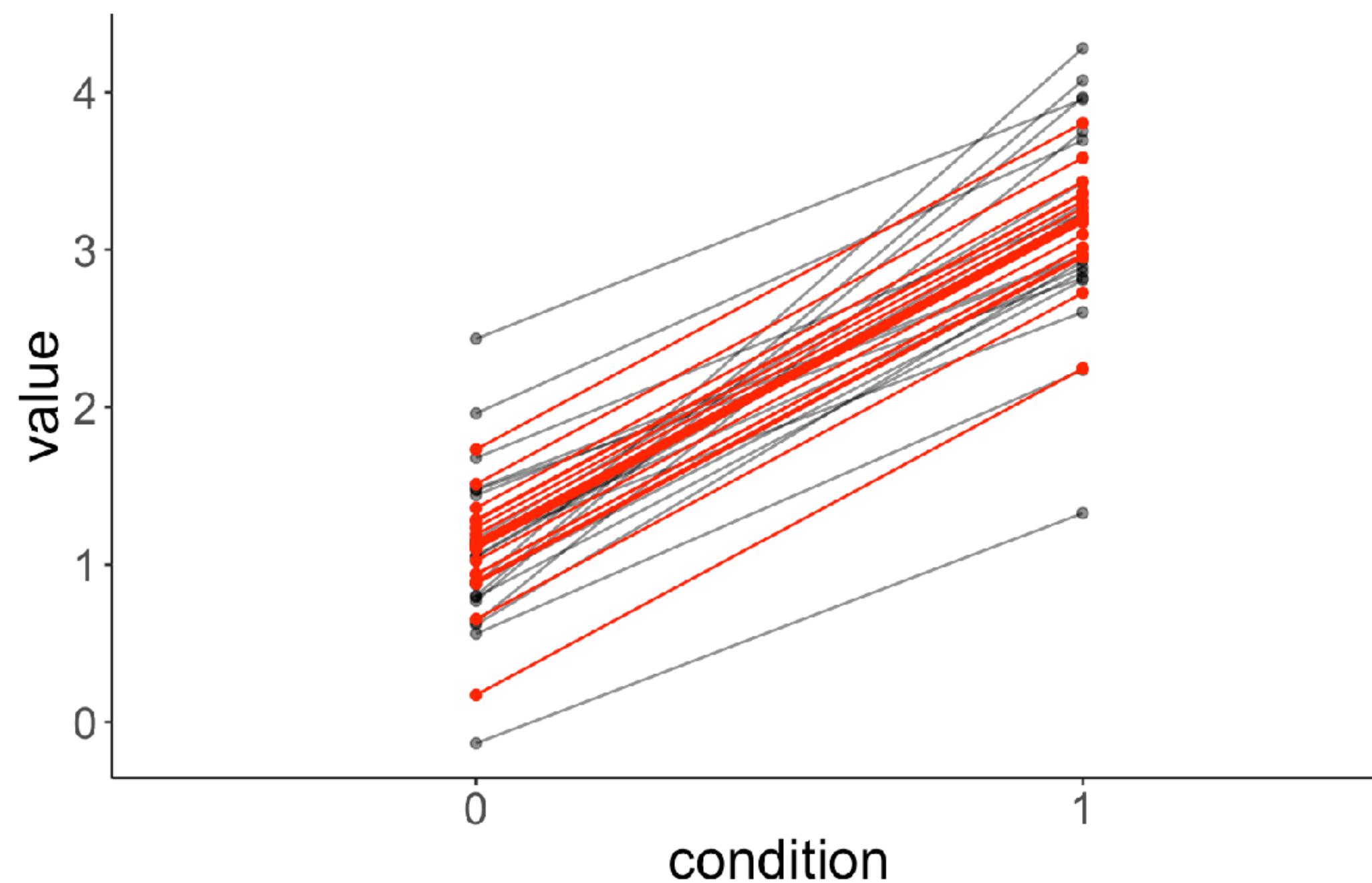
Scaled residuals:
    Min      1Q  Median      3Q     Max 
-2.53710 -0.62295 -0.04364  0.67035  2.19899 

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 

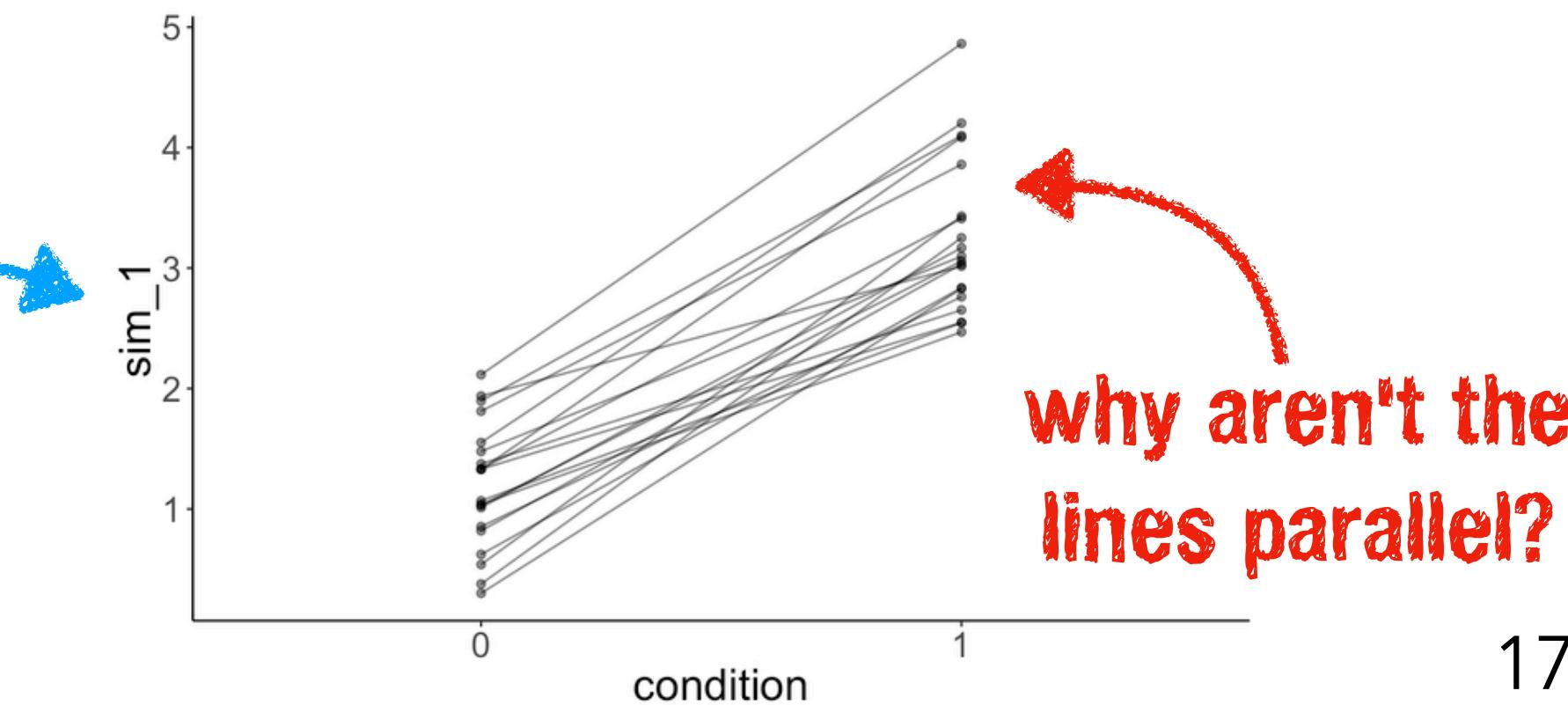
Random effects:
 Groups      Name        Variance Std.Dev. 
participant (Intercept) 0.1702   0.4125  
Residual           0.2270   0.4764  
Number of obs: 40, groups: participant, 20

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

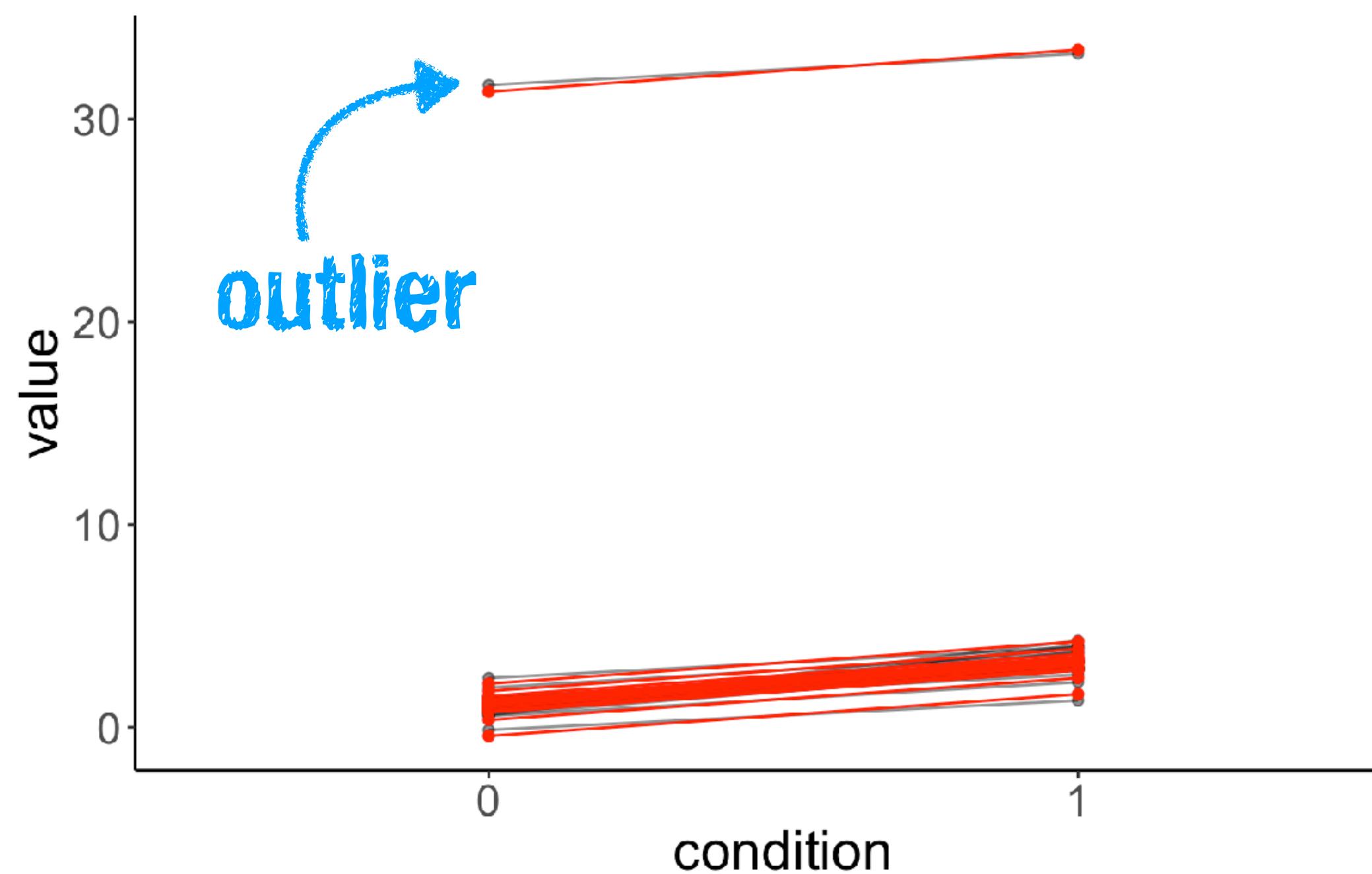
Correlation of Fixed Effects:
          (Intr) condition1 
condition1 -0.535
```

```
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



# 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 

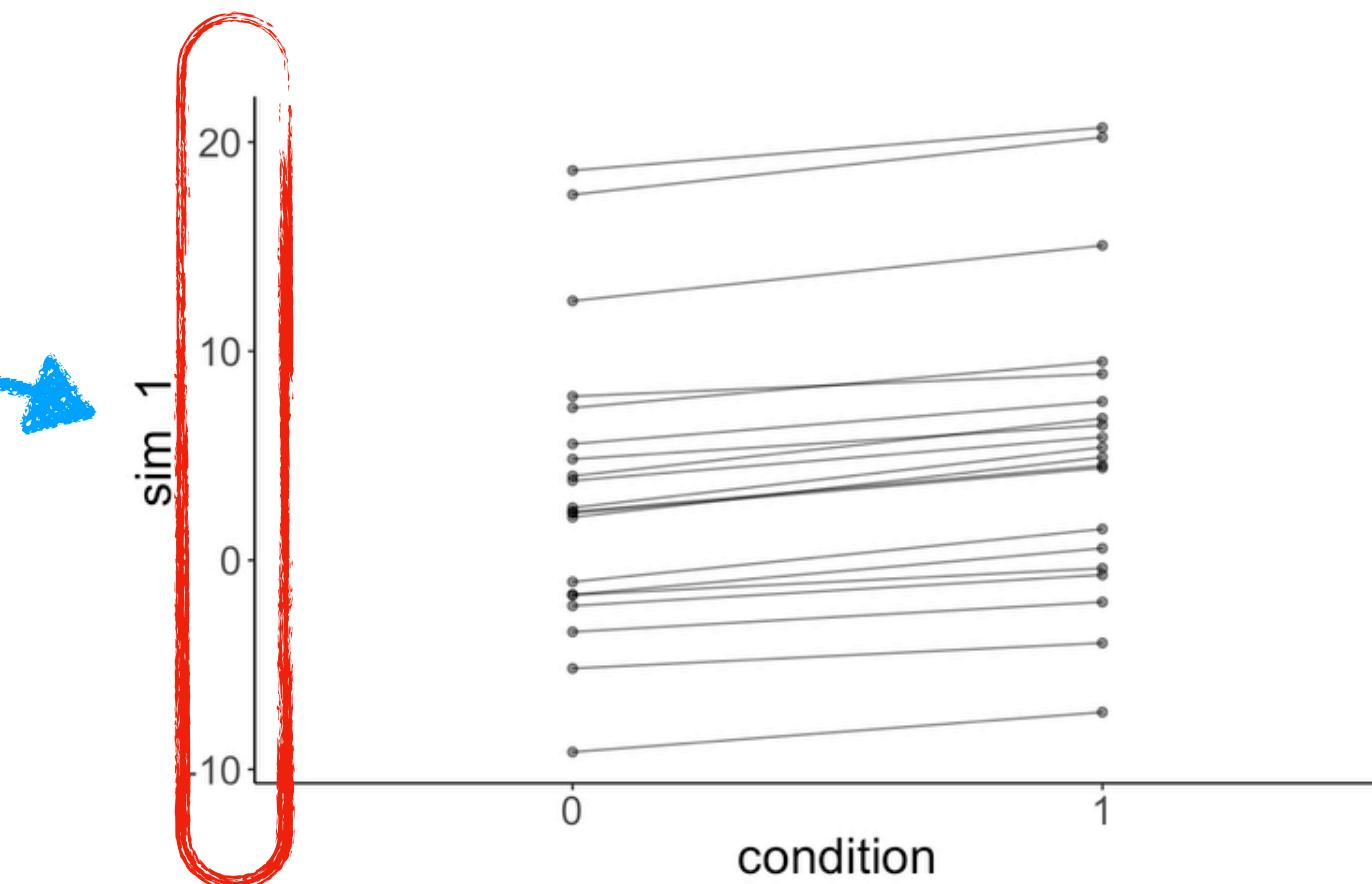
Random effects:
 Groups   Name        Variance Std.Dev. 
participant (Intercept) 46.198   6.7969 
Residual             0.227   0.4764 
Number of obs: 40, groups: participant, 20

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

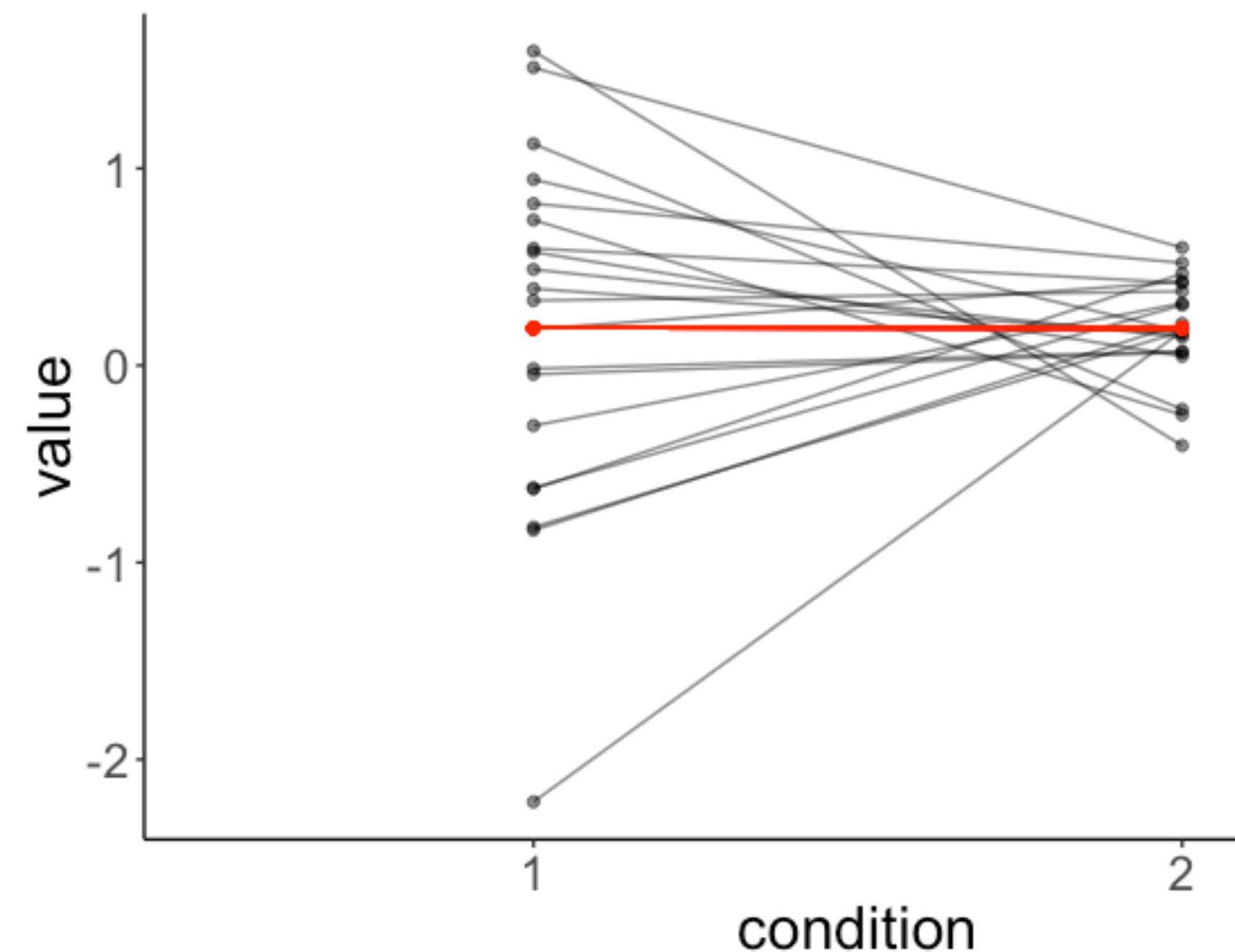
Correlation of Fixed Effects:
              (Intr) condition1 
condition1 -0.049
```

```
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 

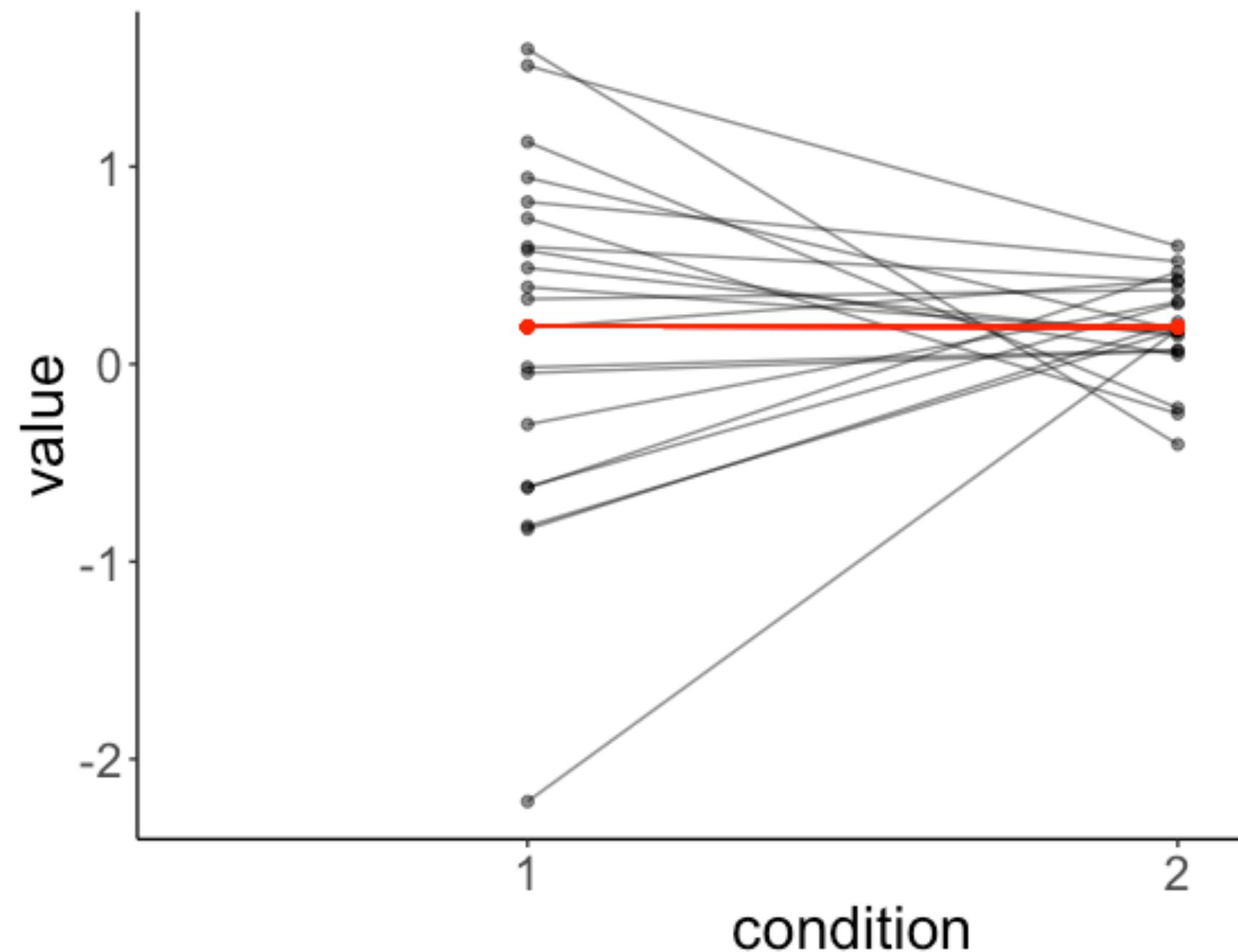
Random effects:
Groups      Name        Variance Std.Dev. 
participant (Intercept) 0.0000  0.0000  
Residual            0.4512  0.6717  
Number of obs: 40, groups: participant, 20

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?!

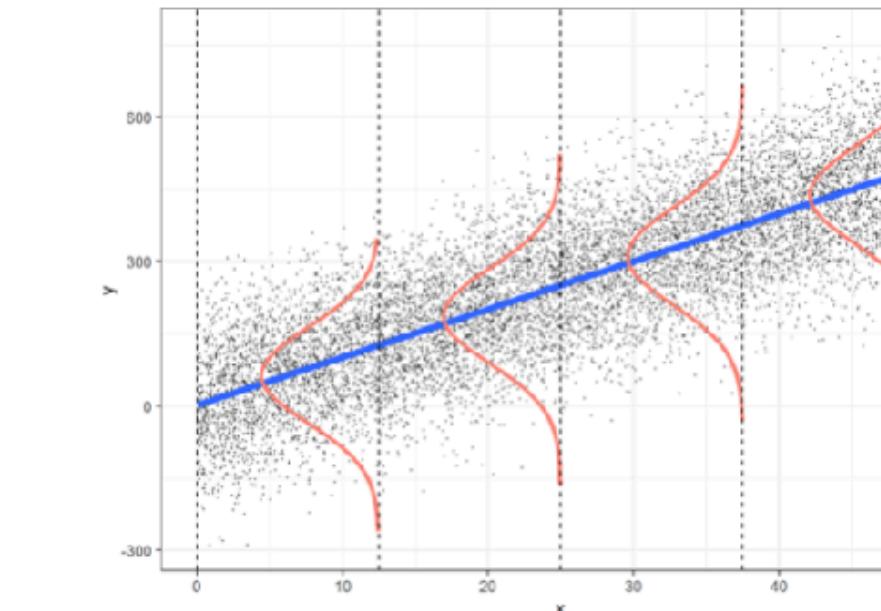
# Non-equal variance



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



random slope



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

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
```

# **Some more examples**

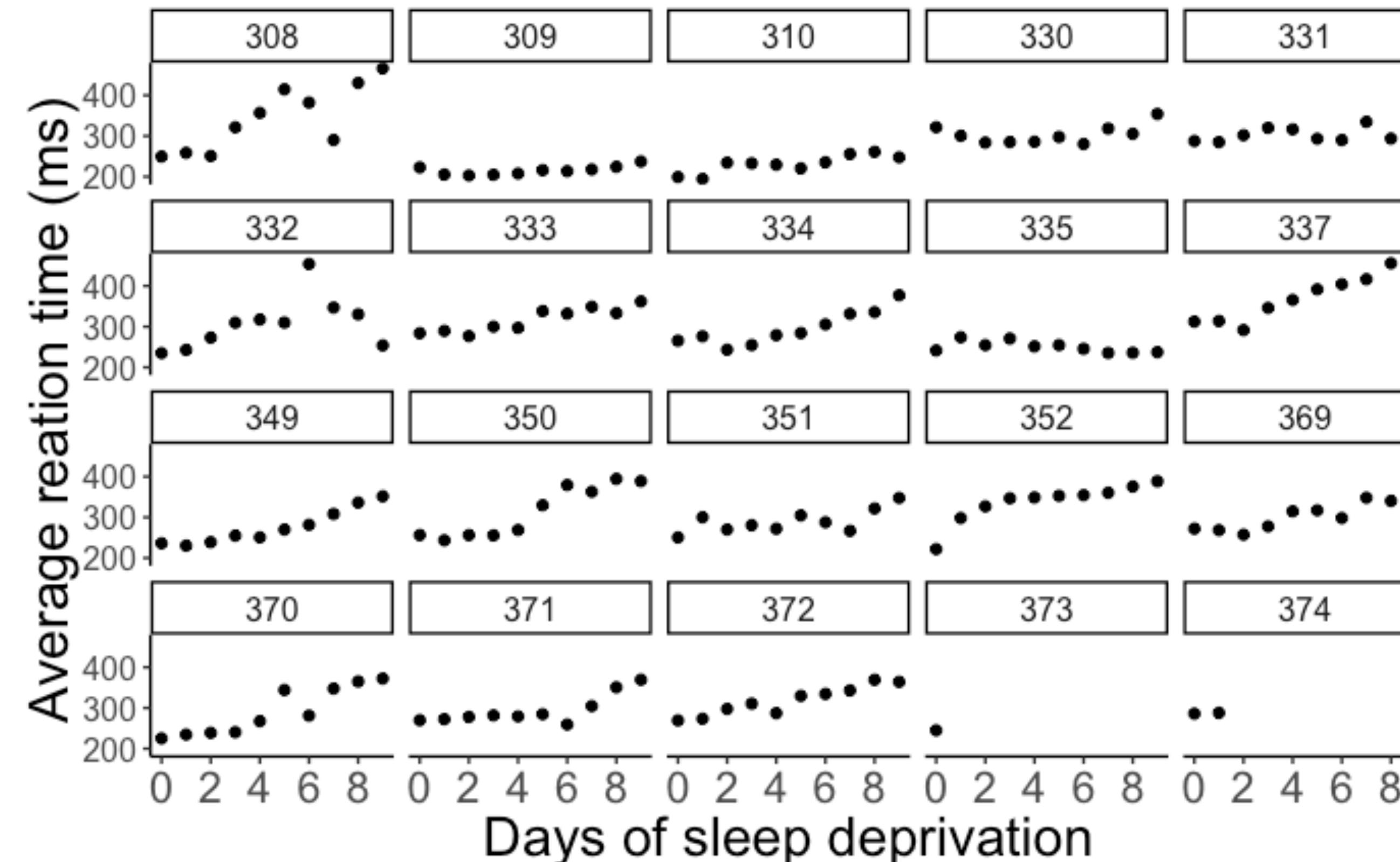
# 1. Sleep



# 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



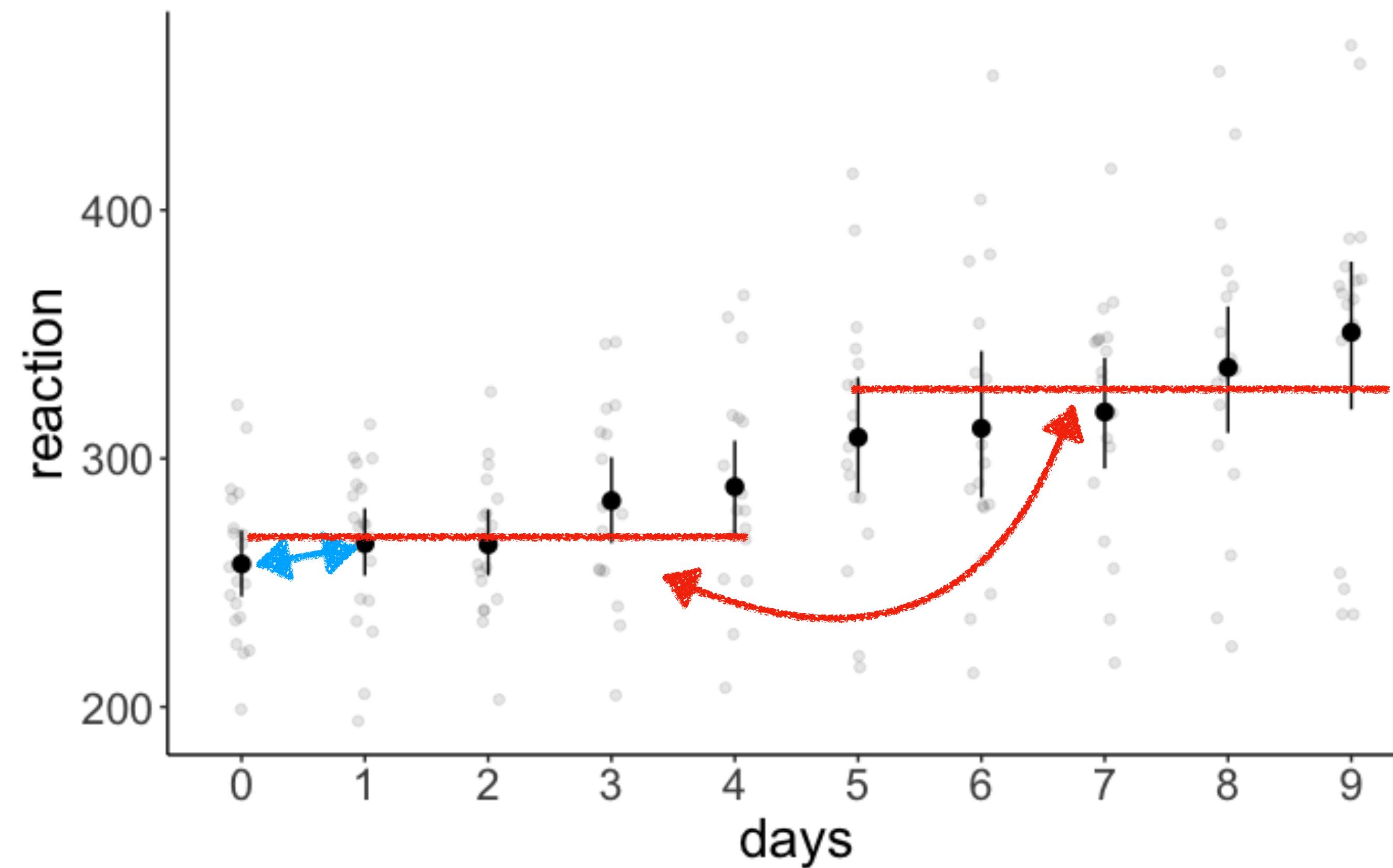
20 participants

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

<b>id</b>	<b>diet</b>	<b>exercises</b>	<b>timepoint</b>	<b>score</b>
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

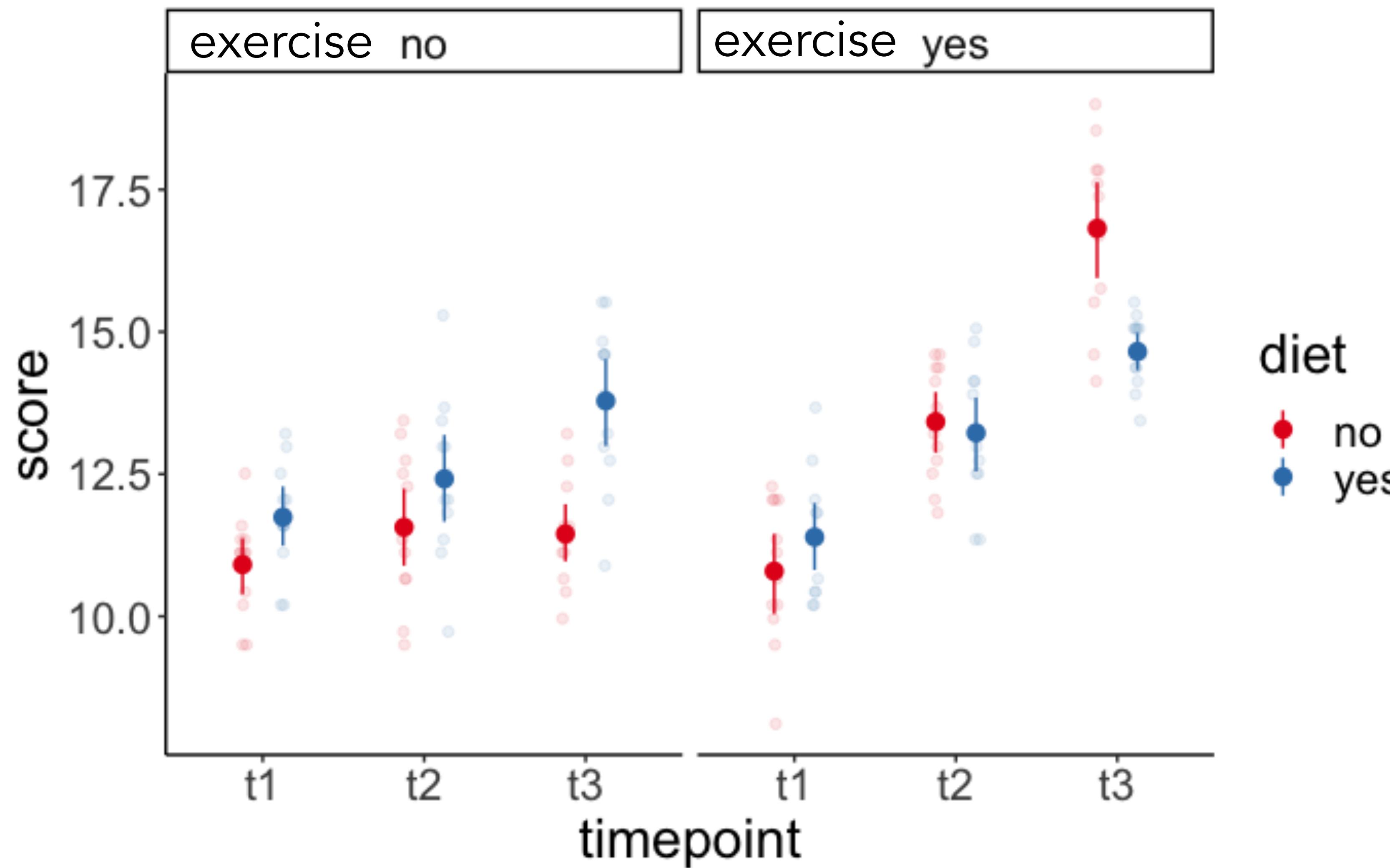
between participants: exercise yes/no

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

```
1 fit = aov_ez(id = "id",
2                 dv = "score",
3                 between = "exercises",
4                 within = c("diet", "timepoint"),
5                 data = df.weightloss)
```

df.weightloss

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

Anova Table (Type 3 tests)

Response: score

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

## main effects and interactions

# Weight loss data

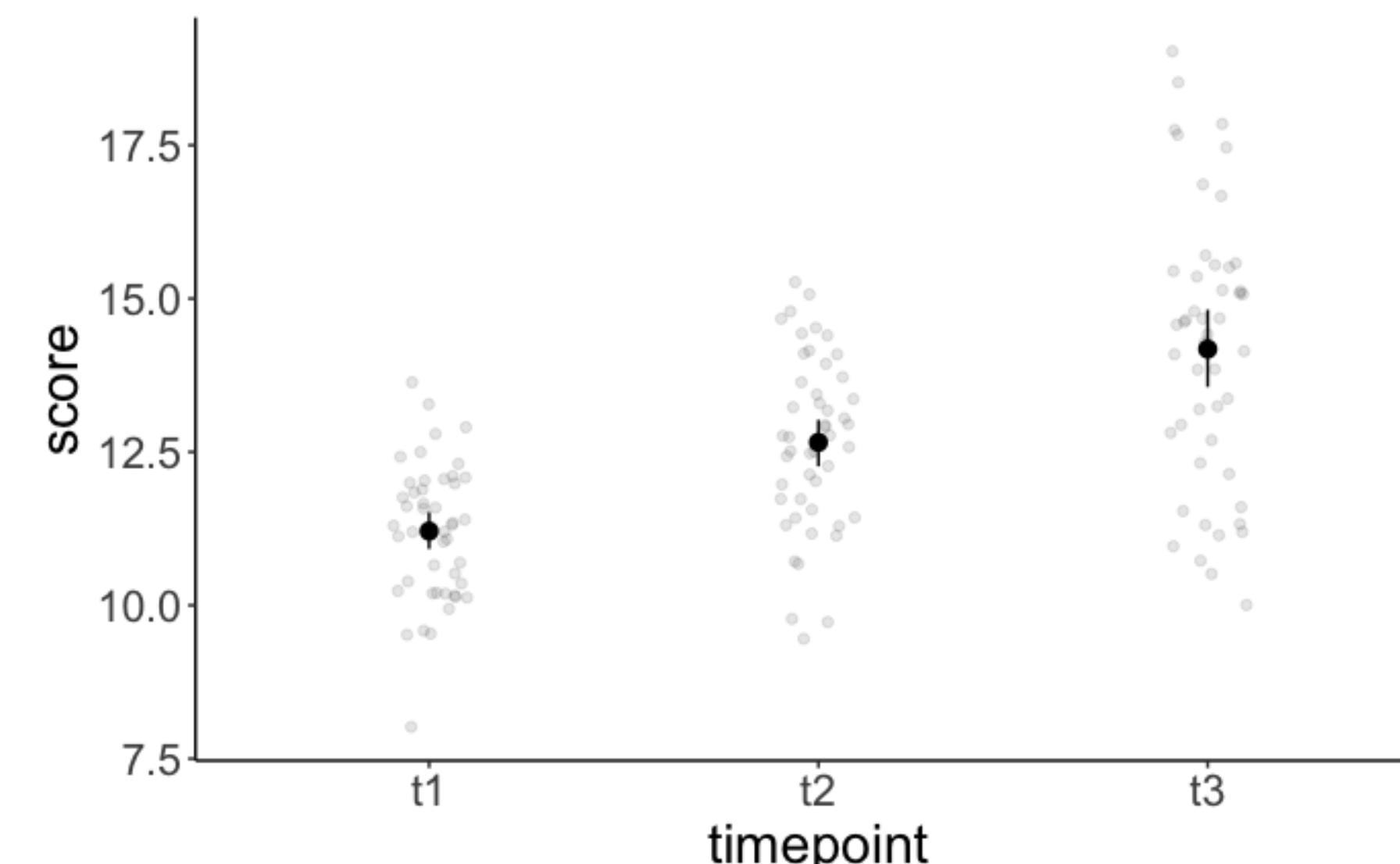
1. Is the score at the third time point different from the other two time points?
2. Is there a linear increase across time points?

```
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	44	11.194	<.0001
linear_increase	2.97	0.231	44	12.820	<.0001

df.weightloss

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



### 3. politeness



# Politeness

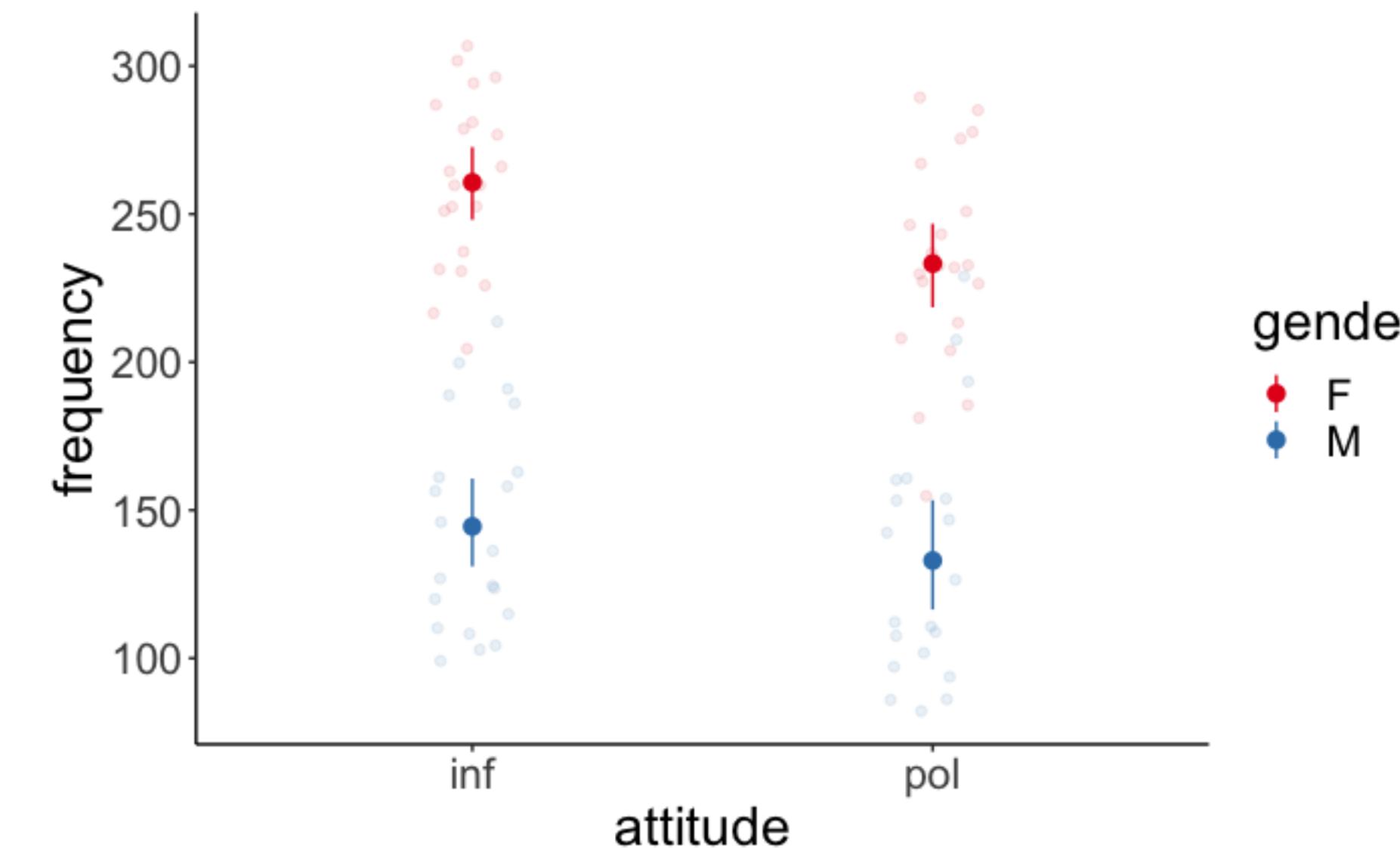
subject	gender	scenario	attitude	frequency
F1	F	1	pol	213.3
F1	F	1	inf	204.5
F1	F	2	pol	285.1
F1	F	2	inf	259.7
F1	F	3	pol	203.9
F1	F	3	inf	286.9
F1	F	4	pol	250.8
F1	F	4	inf	276.8
F1	F	5	pol	231.9
F1	F	5	inf	252.4
F1	F	6	pol	181.2
F1	F	6	inf	230.7
F1	F	7	inf	216.5
F1	F	7	pol	154.8
F3	F	1	pol	229.7

gender: female, male

scenario: different text prompt

attitude: informal vs. polite

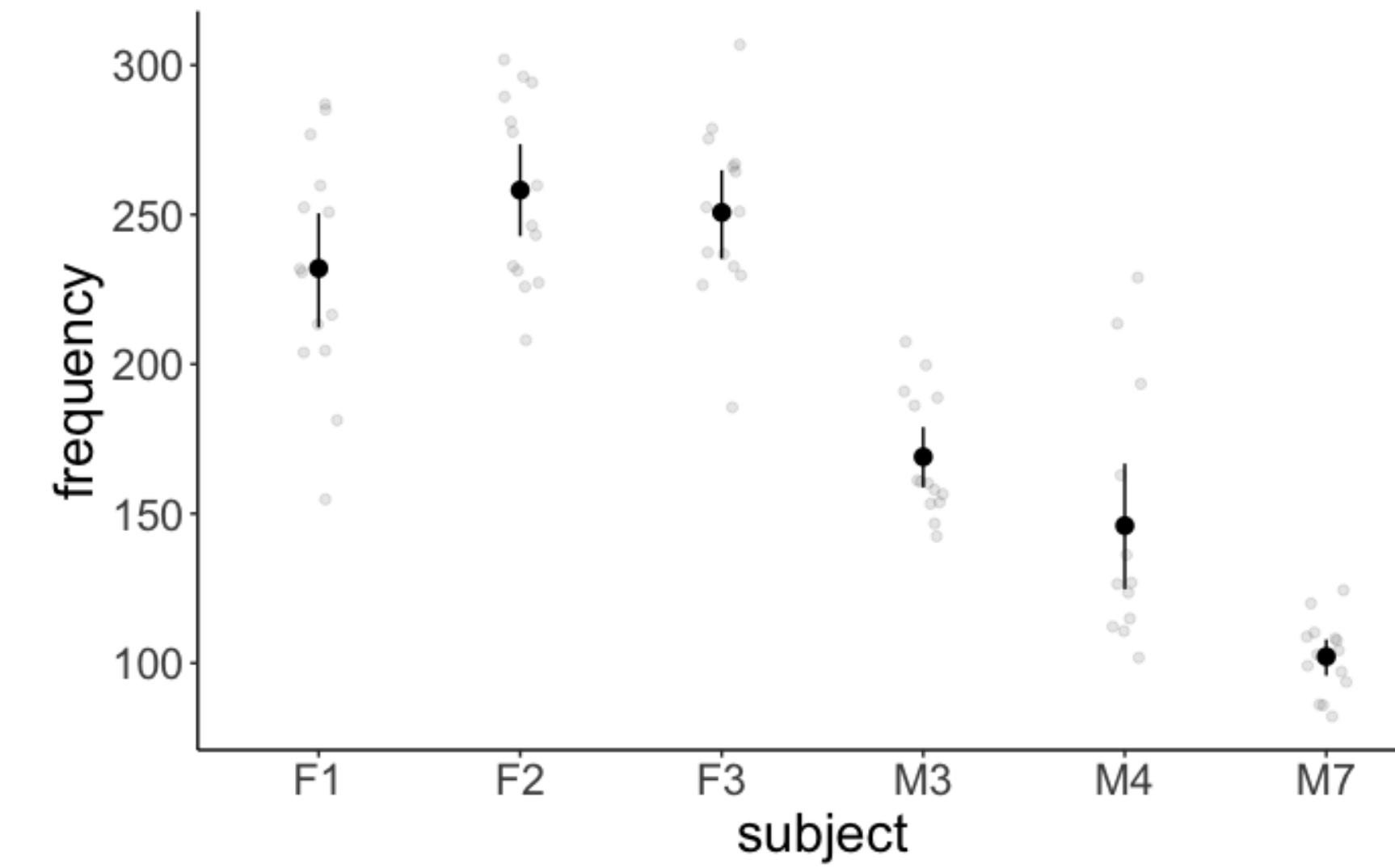
frequency: pitch of voice



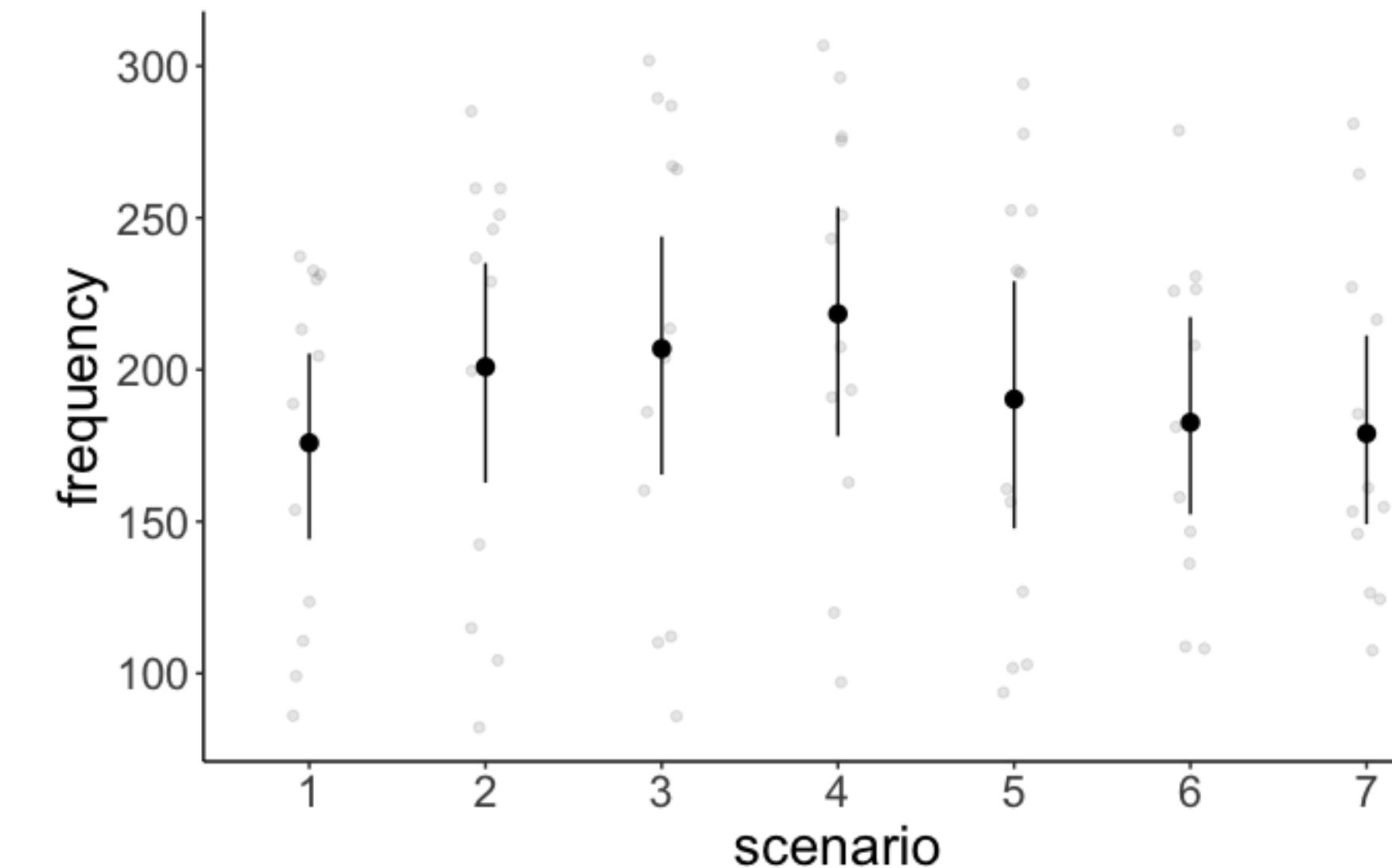
# Politeness

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

variation across subjects



variation across scenarios

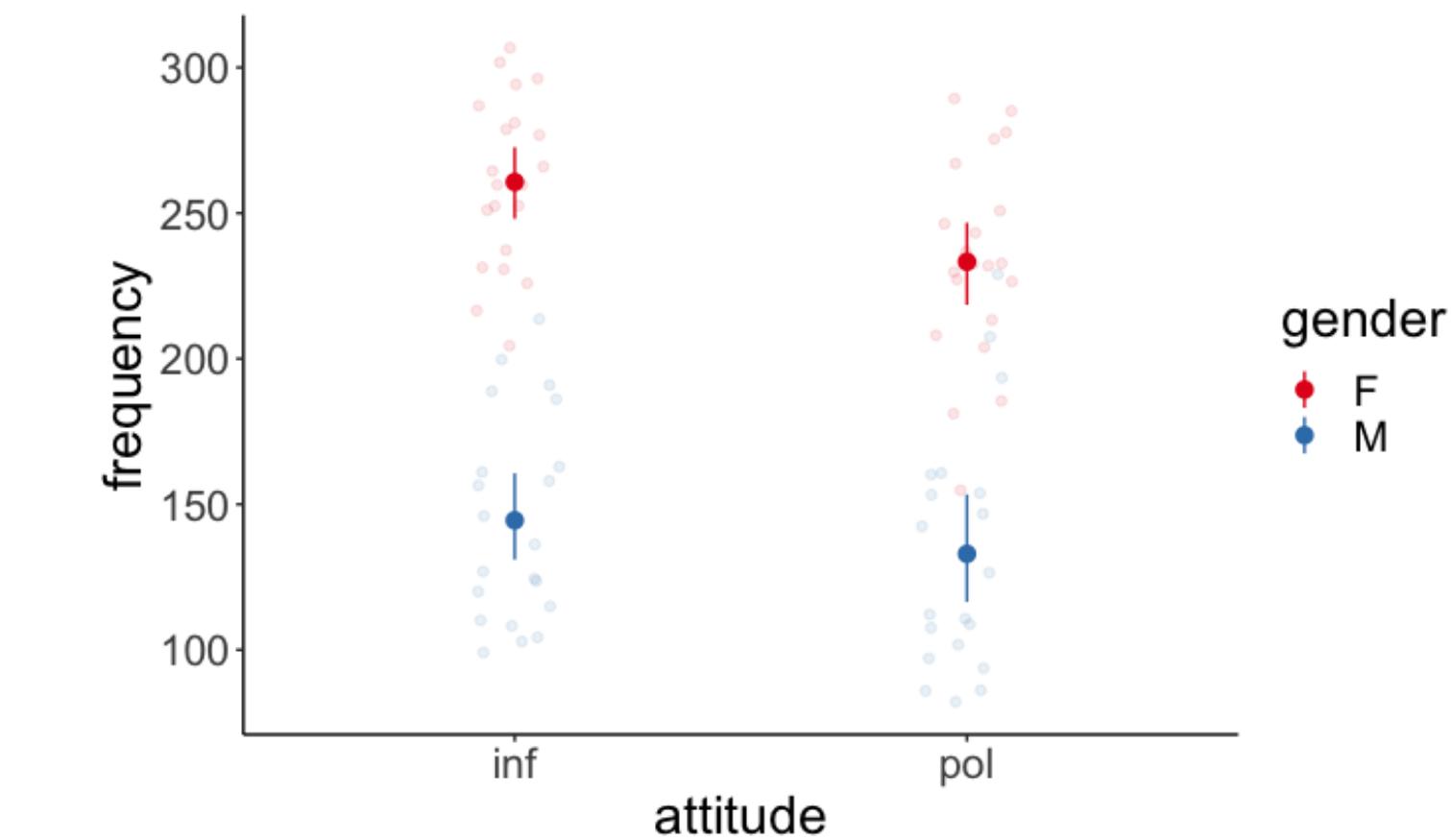


# Politeness

Was there an effect of gender and attitude on pitch?

```
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

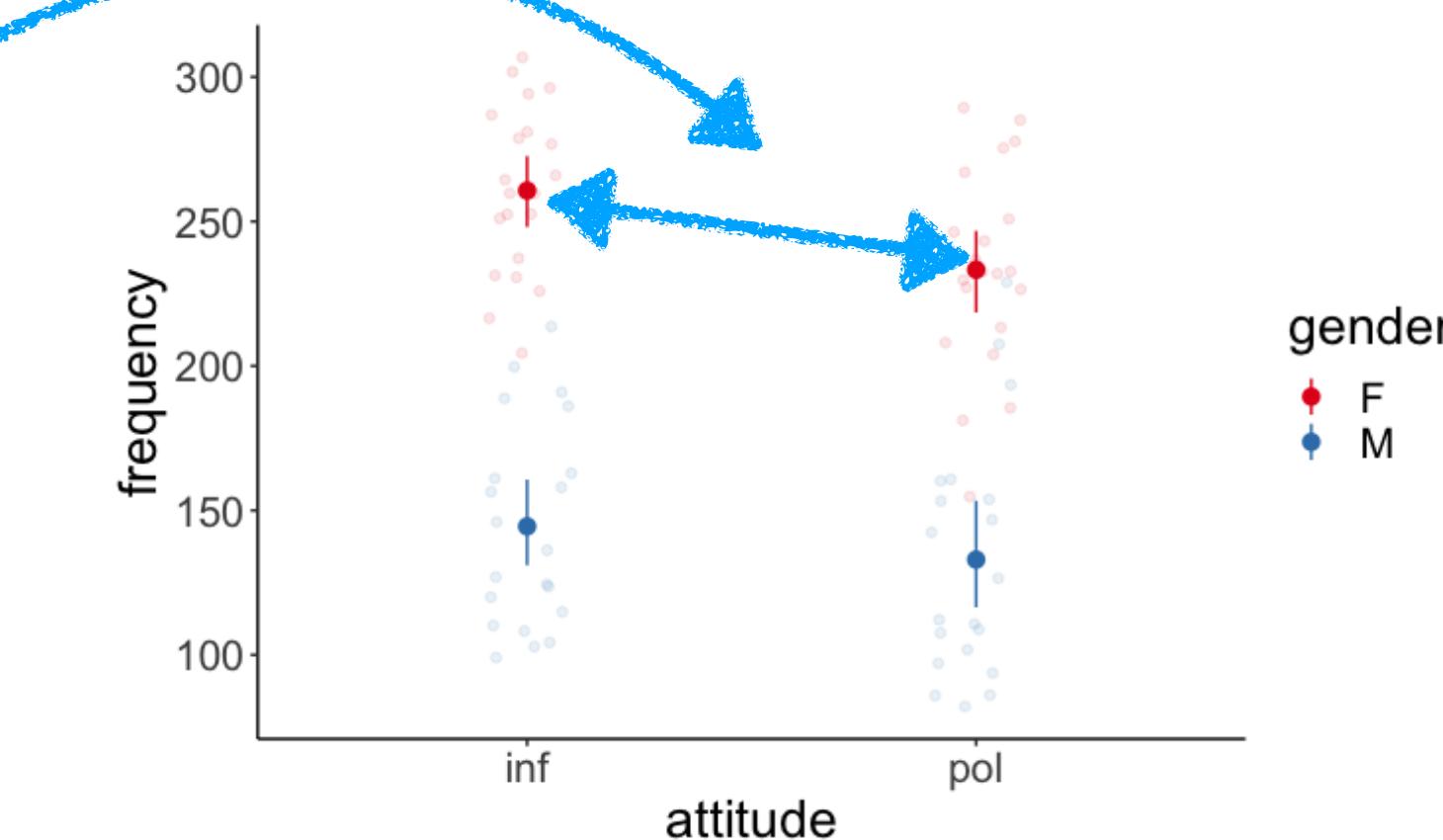
# Politeness

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
```

ignores variation between scenarios,  
and just takes the mean

interaction effect

## LMER

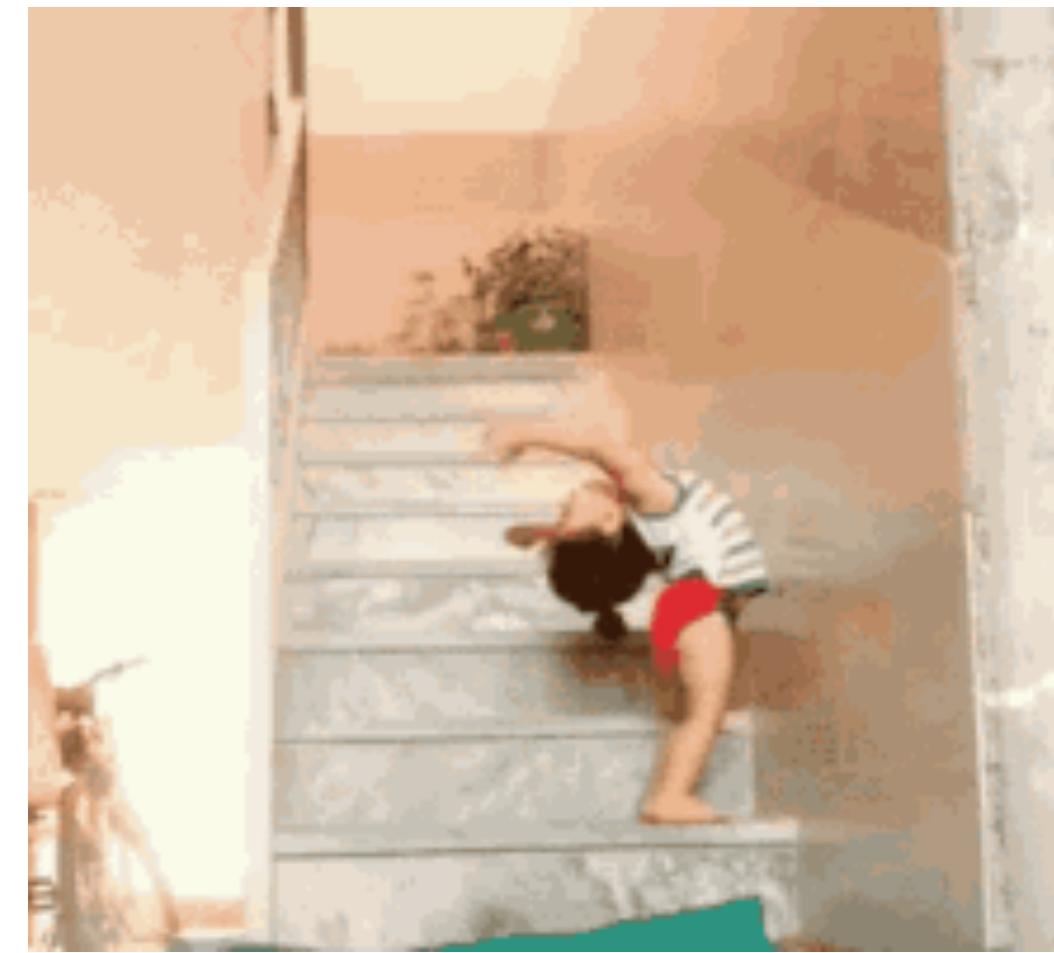
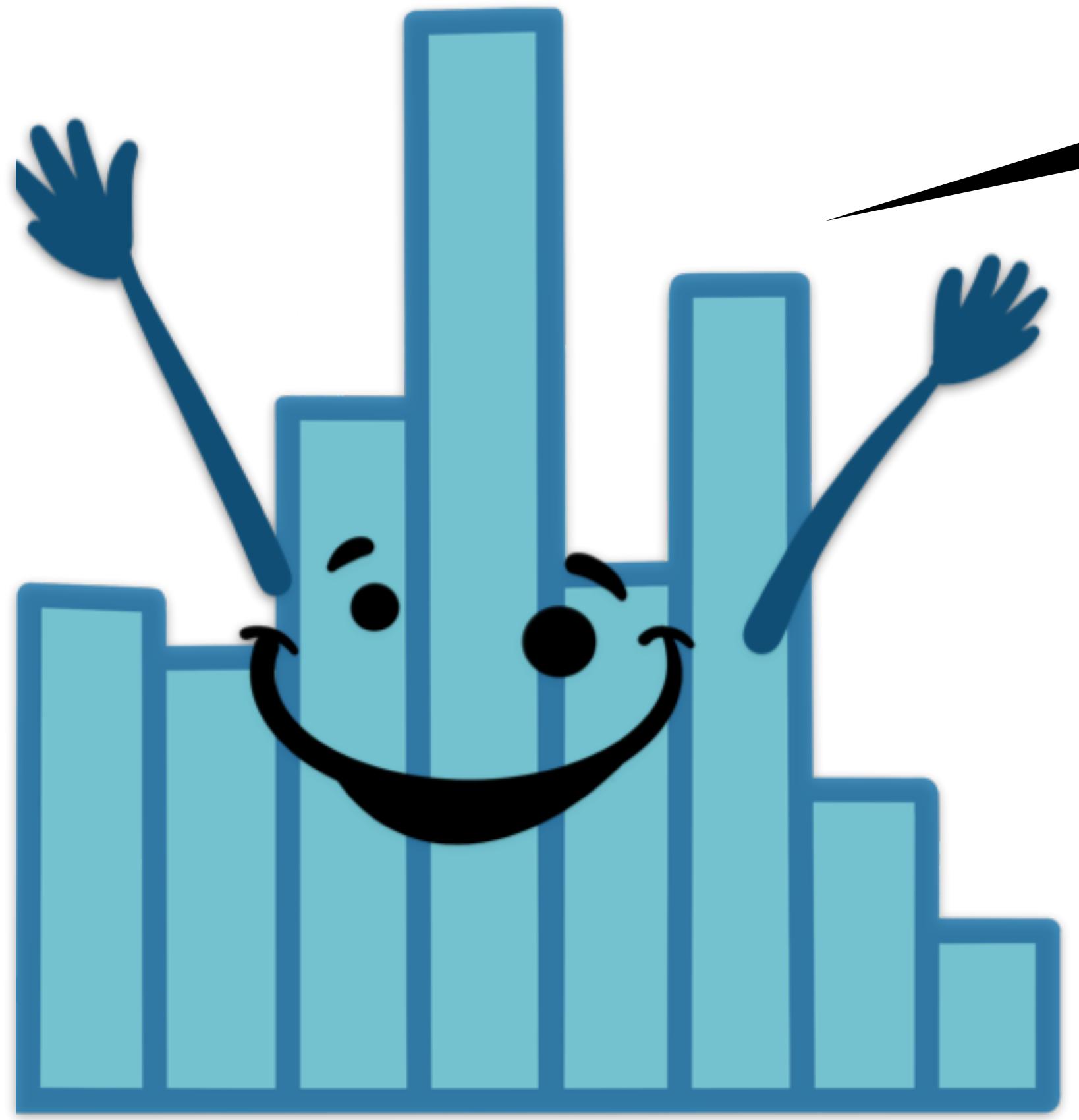
```
1 lmer(formula = frequency ~ 1 + attitude * gender +
       (1 + attitude | subject) + (1 | scenario),
2       data = df.politeness) %>%
3 joint_tests()
```

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

no interaction effect

02:00

stretch break!



# **Different random effect structures**

[Contents](#)[Data format](#)[Power analysis, and simulating these models](#)[Longitudinal two-level model](#)[Three-level models](#)[More on level 1 specification](#)[Hypothesis tests](#)[Book recommendations](#)[Suggestions, errors or typos](#)

# Using R and lme/lmer to fit different two- and three-level longitudinal models

April 21, 2015

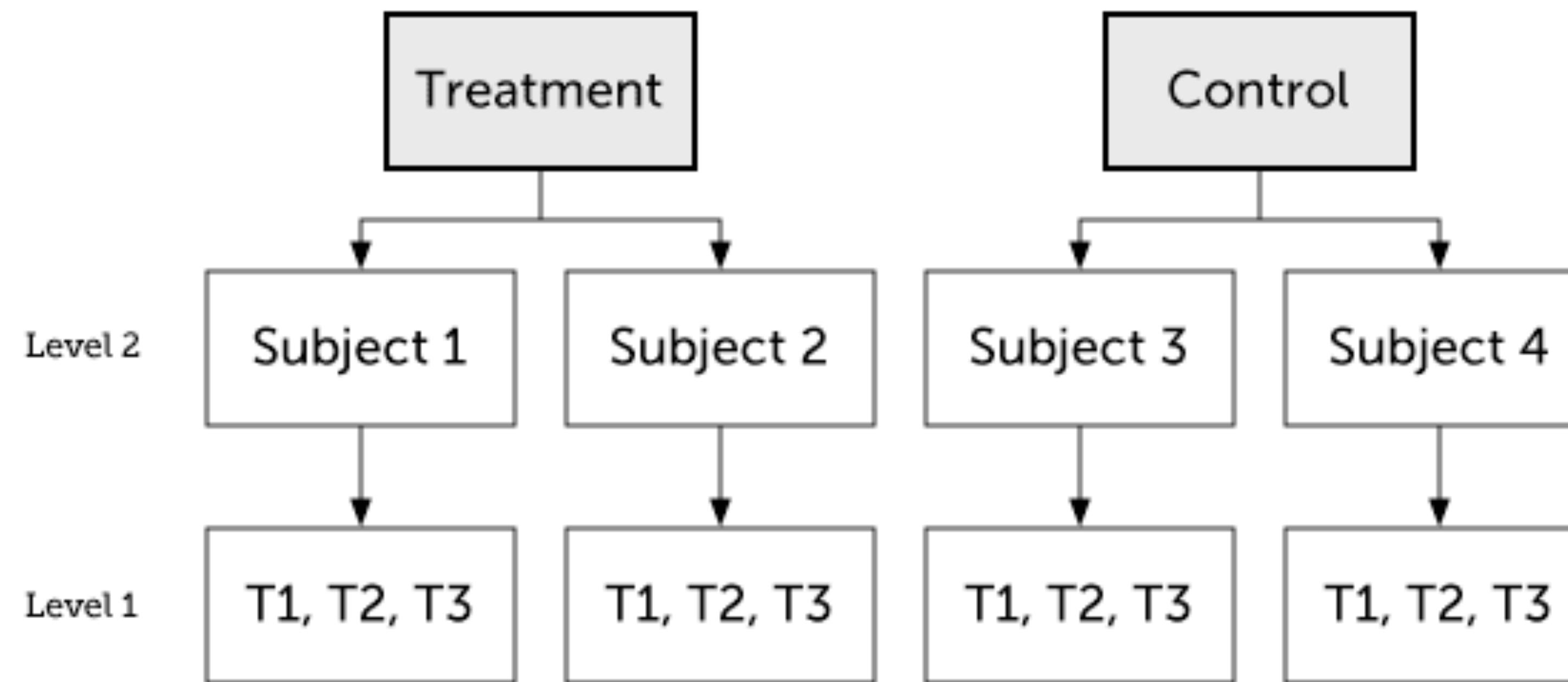


I often get asked how to fit different multilevel models (or individual growth models, hierarchical linear models or linear mixed-models, etc.) in R. In this guide I have compiled some of the more common and/or useful models (at least common in clinical psychology), and how to fit them using `nlme::lme()` and `lme4::lmer()`. I will cover the common two-level random intercept-slope model, and three-level models when subjects are clustered due to some higher level grouping (such as therapists), partially nested models where there are clustering in one group but not the other, and different level 1 residual covariances (such as AR(1)). The point of this post is to show how to fit these longitudinal models in R, not to cover the statistical theory behind them, or how to interpret them.

<https://rpsychologist.com/r-guide-longitudinal-lme-lmer#power-analysis-and-simulating-these-models>

# **Two-level model**

# Graphical representation

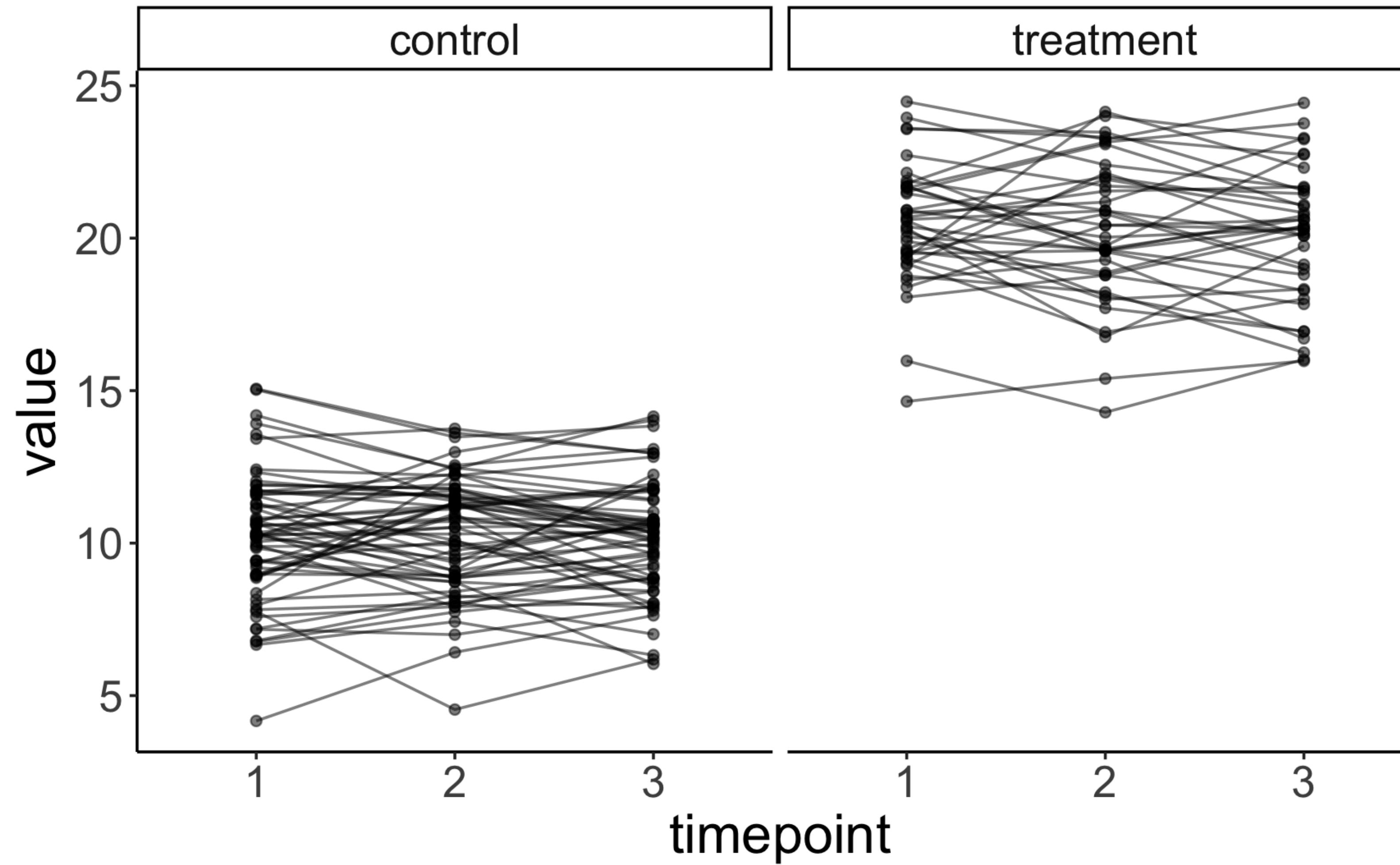


# Simulate data

```
1 set.seed(1)
2
3 n_participants = 100
4 n_timepoints = 3
5 n_conditions = 2
6 p_condition = 0.5
7 b0 = 10
8 b1 = 10
9 sd_participant = 2
10 sd_residual = 1
11
12 df.data = tibble(participant = rep(1:n_participants, each = n_timepoints),
13                    timepoint = rep(1:n_timepoints, times = n_participants),
14                    intercept_participant = rep(rnorm(n_participants, sd = sd_participant),
15                                         each = n_timepoints)) %>%
16  group_by(participant) %>%
17  mutate(condition = rbinom(n = 1, size = 1, prob = p_condition)) %>%
18  ungroup() %>%
19  mutate(value = b0 + b1 * condition + intercept_participant +
20         rnorm(n_participants * n_timepoints, sd = sd_residual))
```

	participant	timepoint	intercept_participant	condition	value
1	1	1	-1.25290762	0	9.197279
2	1	2	-1.25290762	0	8.728533
3	1	3	-1.25290762	0	8.429024
4	2	1	0.36728665	0	9.437925
5	2	2	0.36728665	0	8.879826
6	2	3	0.36728665	0	9.292094
7	3	1	-1.67125722	1	19.328772
8	3	2	-1.67125722	1	17.707476
9	3	3	-1.67125722	1	16.944316
10	4	1	3.19056160	0	15.059852
11	4	2	3.19056160	0	13.615662
12	4	3	3.19056160	0	12.951915

# Plot data



# Fit the model

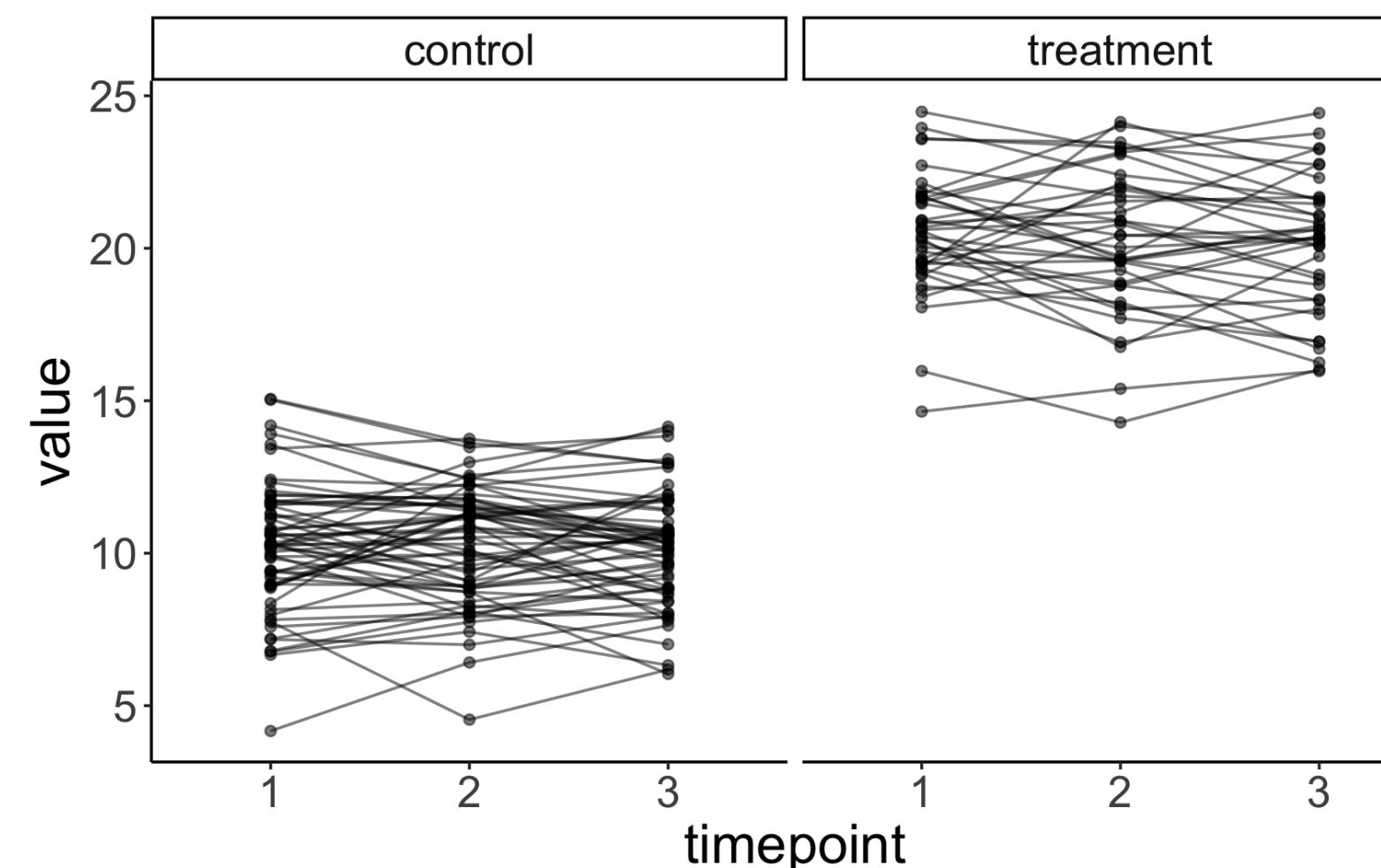
```
1 fit = lmer(formula = value ~ 1 + condition + (1 | participant),  
2             data = df.data)
```

```
Linear mixed model fit by REML ['lmerMod']  
Formula: value ~ 1 + condition + (1 | participant)  
Data: df.data  
  
REML criterion at convergence: 1102  
  
Scaled residuals:  
    Min     1Q   Median     3Q    Max  
-2.30522 -0.57146  0.03152  0.56826  2.28135  
  
Random effects:  
 Groups      Name        Variance Std.Dev.  
 participant (Intercept) 3.106    1.762  
 Residual            1.087    1.043  
Number of obs: 300, groups: participant, 100  
  
Fixed effects:  
            Estimate Std. Error t value  
(Intercept) 10.2199    0.2365  43.21  
condition    10.0461    0.3837  26.18  
  
Correlation of Fixed Effects:  
            (Intr)  
condition -0.616
```

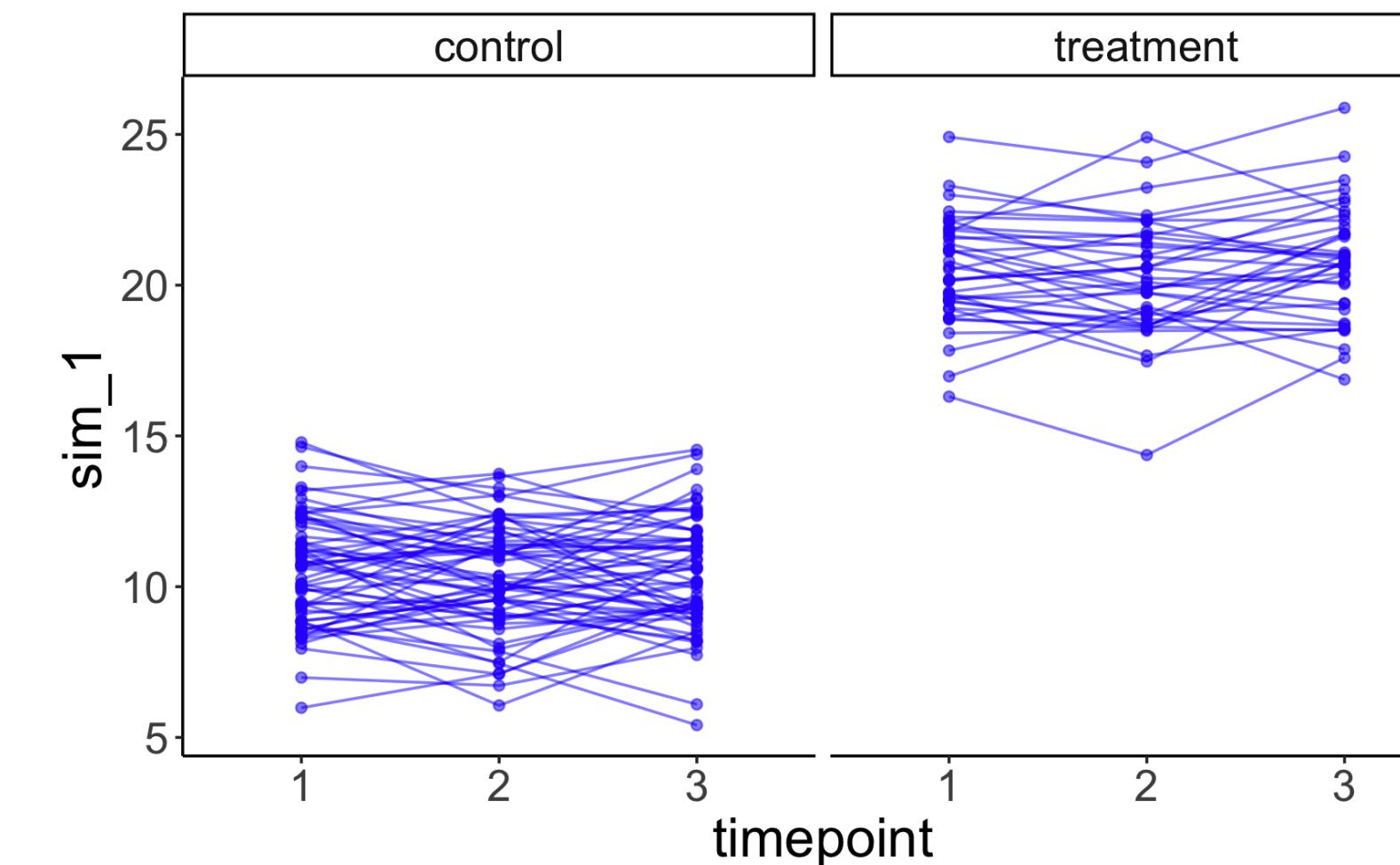
# Simulate new data from the model

```
1 set.seed(1)
2
3 fit %>%
4   simulate() %>%
5   bind_cols(df.data) %>%
6   mutate(condition = factor(condition,
7                             levels = c(0, 1),
8                             labels = c("control", "treatment")),
9          timepoint = as.factor(timepoint)) %>%
10 ggplot(data = .,
11         mapping = aes(x = timepoint,
12                         y = sim_1,
13                         group = participant)) +
14   geom_point(alpha = 0.5,
15             color = "blue") +
16   geom_line(alpha = 0.5,
17             color = "blue") +
18   facet_grid(~ condition) +
19   labs(x = "timepoint")
```

original data

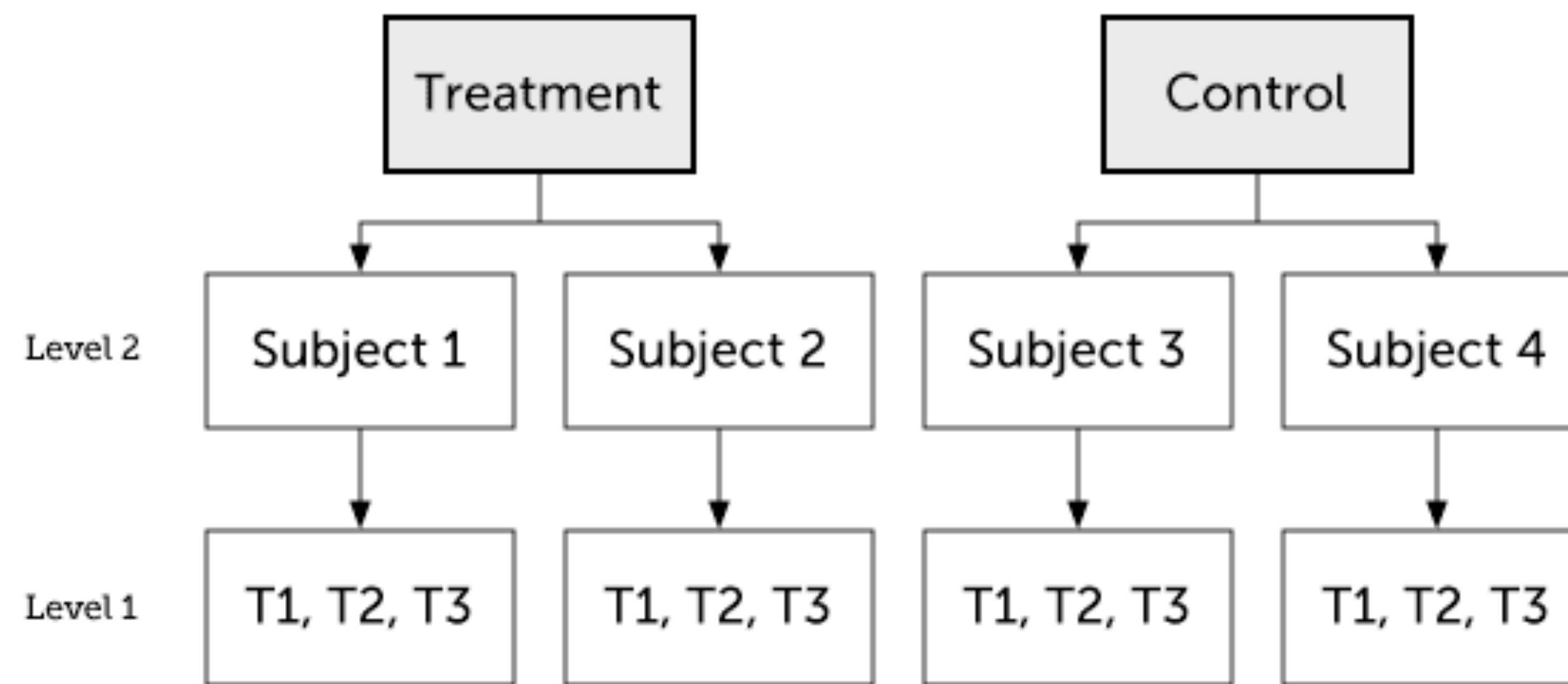


simulated data



# **Two-level growth model**

# Graphical representation



**assume a linear effect of time**

# Simulate data

```

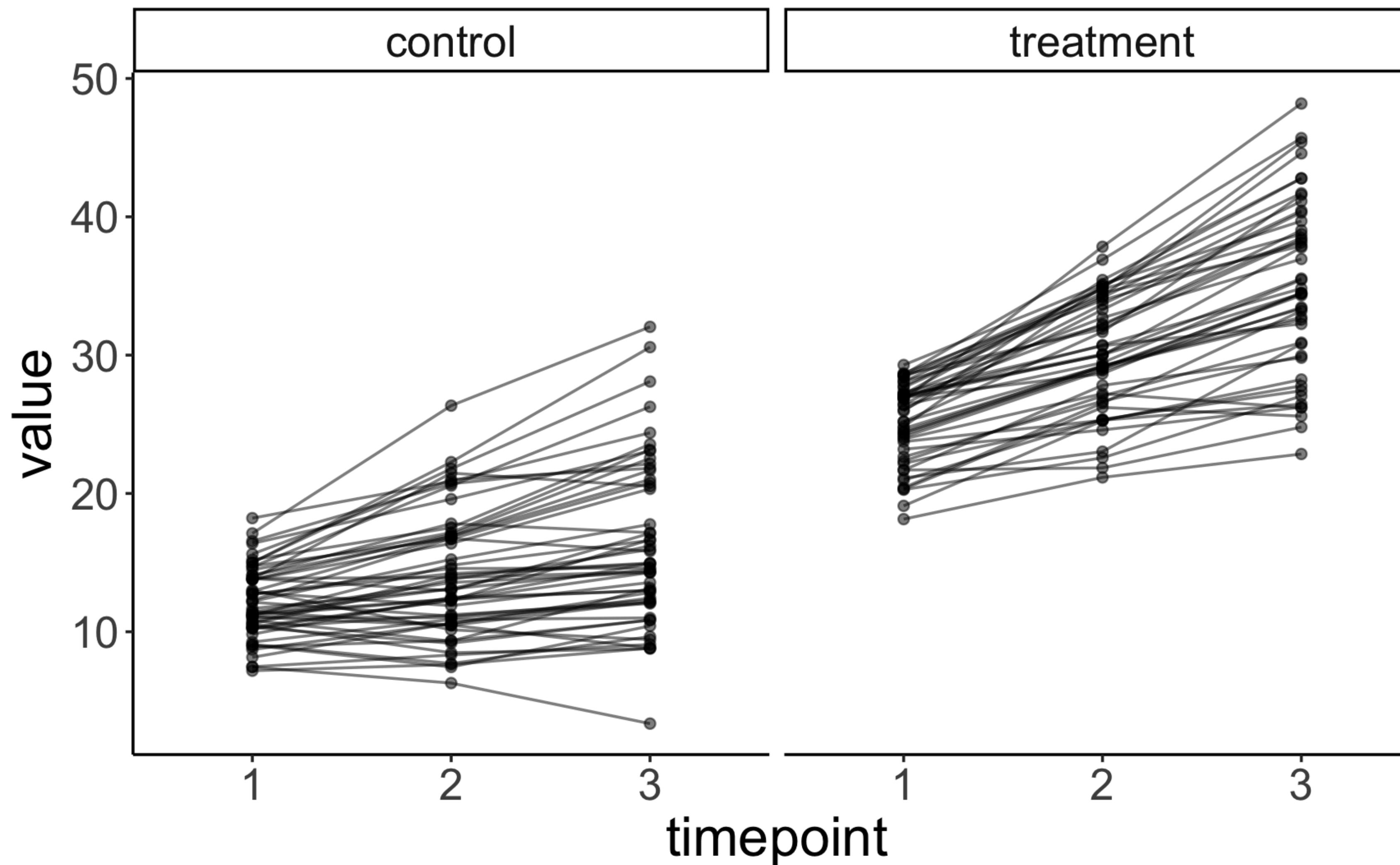
1 set.seed(1)
2
3 n_participants = 100
4 n_timepoints = 3
5 n_conditions = 2
6 p_condition = 0.5
7 b0 = 10 # intercept
8 b1 = 10 # condition
9 b2 = 2 # time
10 b3 = 3 # interaction
11 sd_intercept_participant = 2
12 sd_time_participant = 2
13 sd_residual = 1
14
15 df.data = tibble(participant = rep(1:n_participants, each = n_timepoints),
16                   timepoint = rep(1:n_timepoints, times = n_participants),
17                   intercept_participant = rep(rnorm(n_participants, sd = sd_intercept_participant),
18                                     each = n_timepoints),
19                   time_participant = rep(rnorm(n_participants, sd = sd_time_participant),
20                                     each = n_timepoints)) %>%
21   group_by(participant) %>%
22   mutate(condition = rbinom(n = 1, size = 1, prob = p_condition)) %>%
23   ungroup() %>%
24   mutate(value = b0 + intercept_participant +
25         b1 * condition +
26         (b2 + time_participant) * timepoint +
27         b3 * condition * timepoint +
28         rnorm(n_participants * n_timepoints, sd = sd_residual))

```

participant	timepoint	intercept_participant	time_participant	condition	value
1	1	-1.252907621	-0.310183339	1	20.573131
2	1	-1.252907621	-0.310183339	1	22.844077
3	1	-1.252907621	-0.310183339	1	24.367254
4	2	0.367286648	0.021057937	0	12.119622
5	2	0.367286648	0.021057937	0	15.062837
6	2	0.367286648	0.021057937	0	17.513515
7	3	-1.671257225	-0.455460824	1	17.470186
8	3	-1.671257225	-0.455460824	1	22.446022
9	3	-1.671257225	-0.455460824	1	24.248006
10	4	3.190561604	0.079014386	1	24.844308
11	4	3.190561604	0.079014386	1	28.220589
12	4	3.190561604	0.079014386	1	28.880339
13	5	0.659015544	-0.327292322	1	22.047393
14	5	0.659015544	-0.327292322	1	25.189133
15	5	0.659015544	-0.327292322	1	28.051351

random slopes

# Plot data

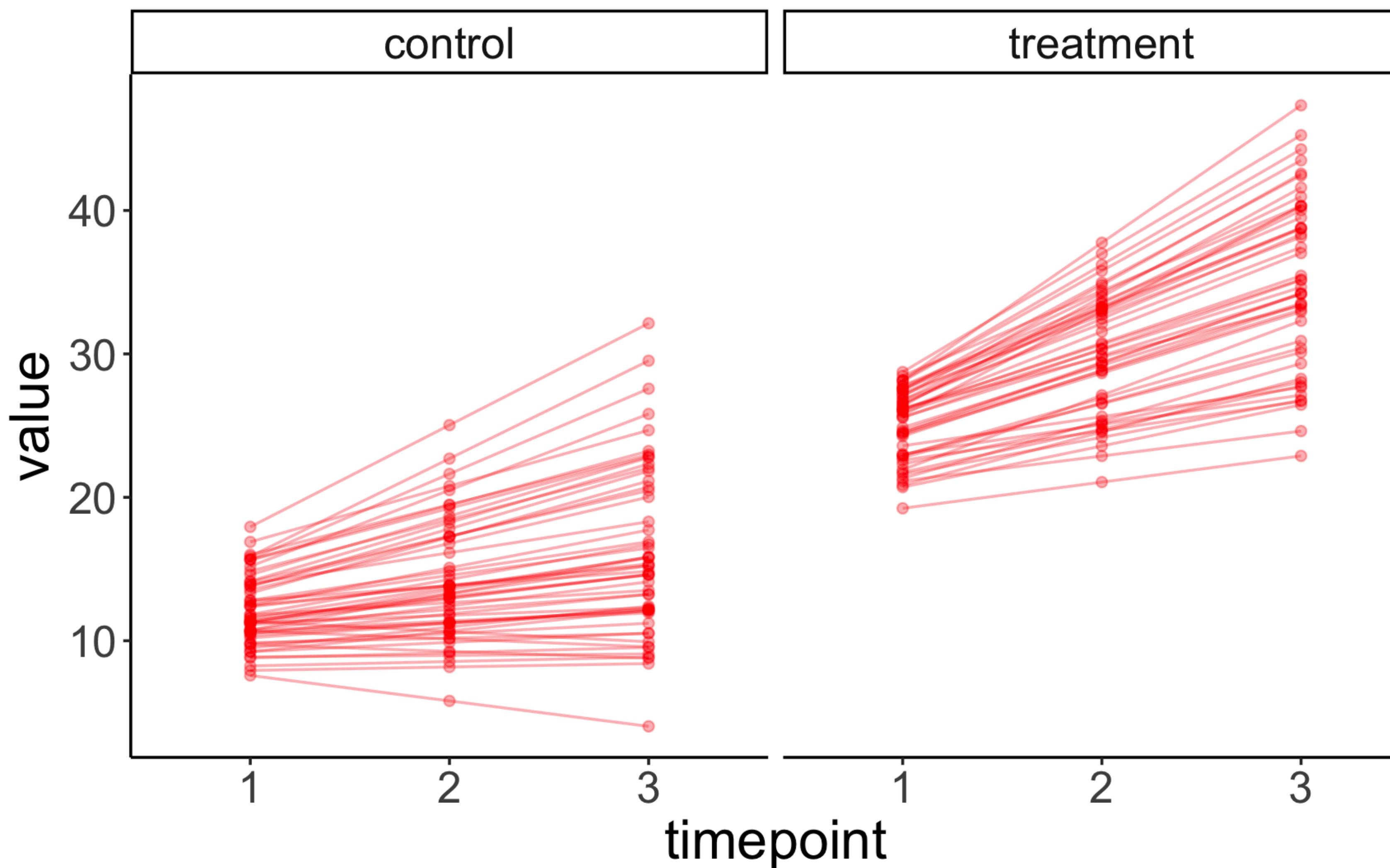


# Fit the model

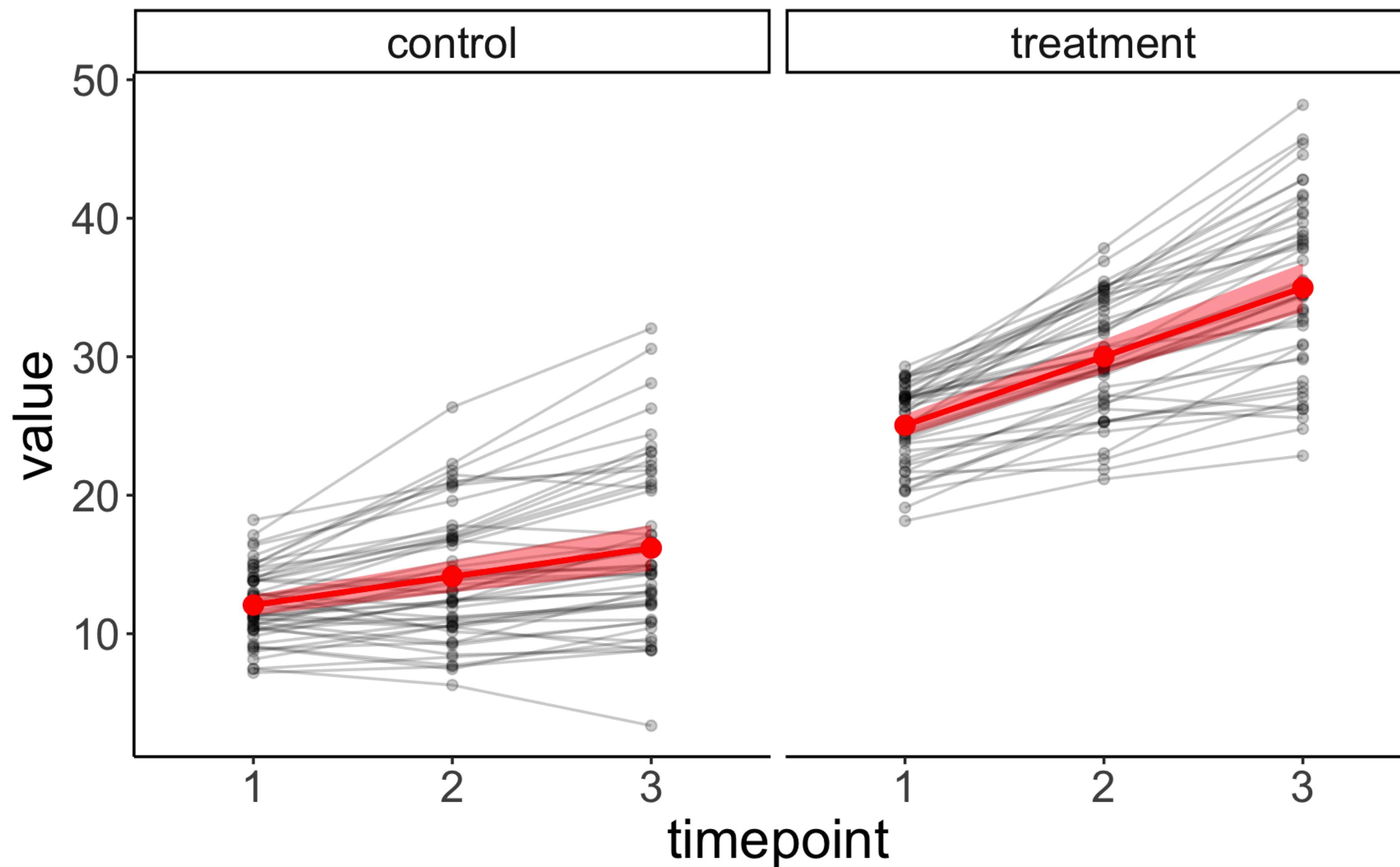
```
1 fit = lmer(formula = value ~ 1 + condition * timepoint + (1 + timepoint | participant),  
2             data = df.data)
```

```
Linear mixed model fit by REML ['lmerMod']  
Formula: value ~ 1 + condition * timepoint + (1 + timepoint | participant)  
Data: df.data  
  
REML criterion at convergence: 1360.3  
  
Scaled residuals:  
    Min     1Q   Median     3Q     Max  
-2.14633 -0.46360  0.03902  0.42302  2.82945  
  
Random effects:  
Groups      Name        Variance Std.Dev. Corr  
participant (Intercept) 3.190    1.786  
                timepoint  3.831    1.957  -0.06  
Residual            1.149    1.072  
Number of obs: 300, groups: participant, 100  
  
Fixed effects:  
              Estimate Std. Error t value  
(Intercept)  10.0101  0.3328  30.079  
condition     10.0684  0.4854  20.741  
timepoint     2.0595   0.2883   7.143  
condition:timepoint  2.9090  0.4205   6.917  
  
Correlation of Fixed Effects:  
          (Intr) condtn timpnt  
condition -0.686  
timepoint -0.266  0.182  
cndtn:tmpnt 0.182 -0.266 -0.686
```

# Visualize model predictions (individual)



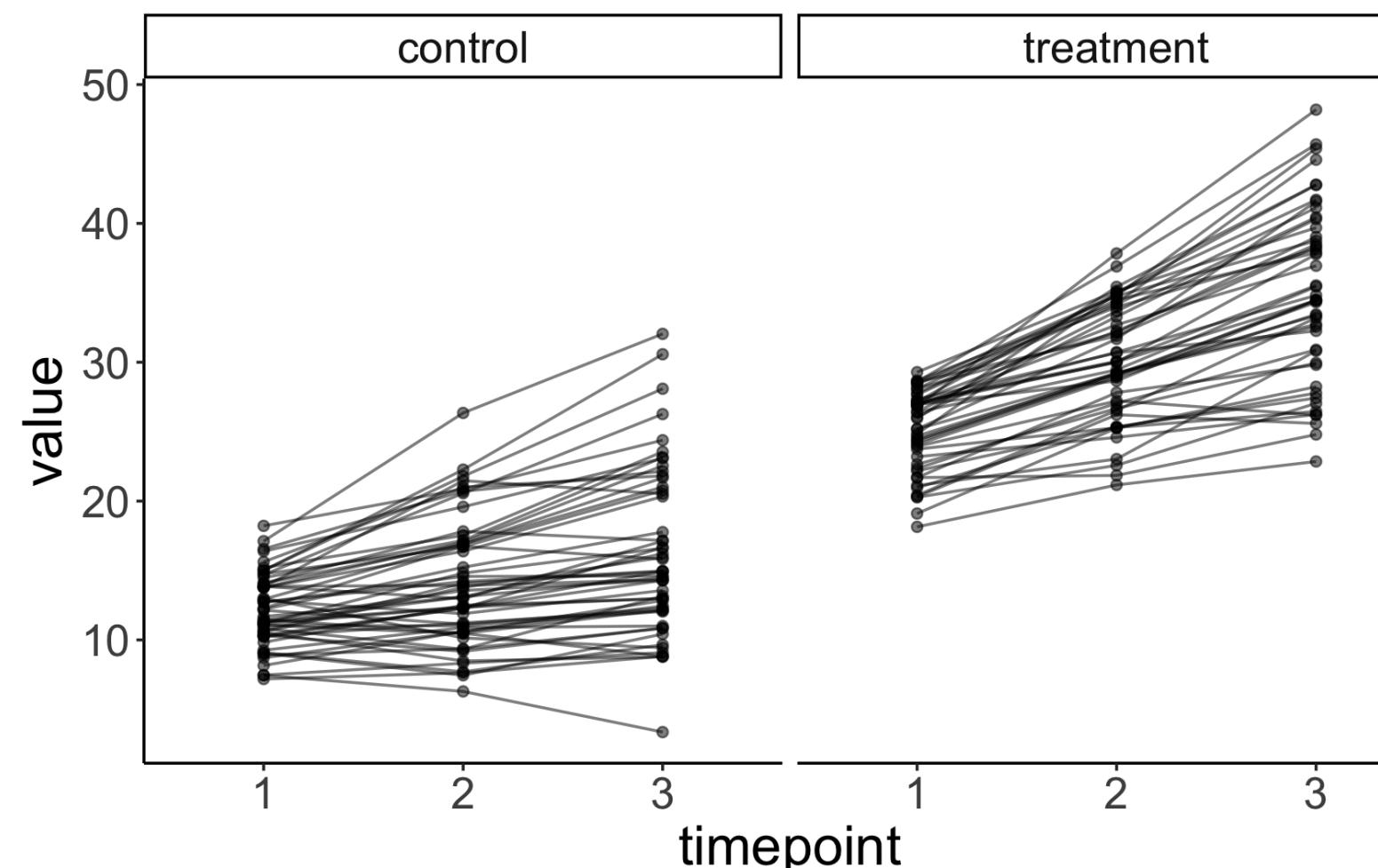
# Visualize model predictions (overall)



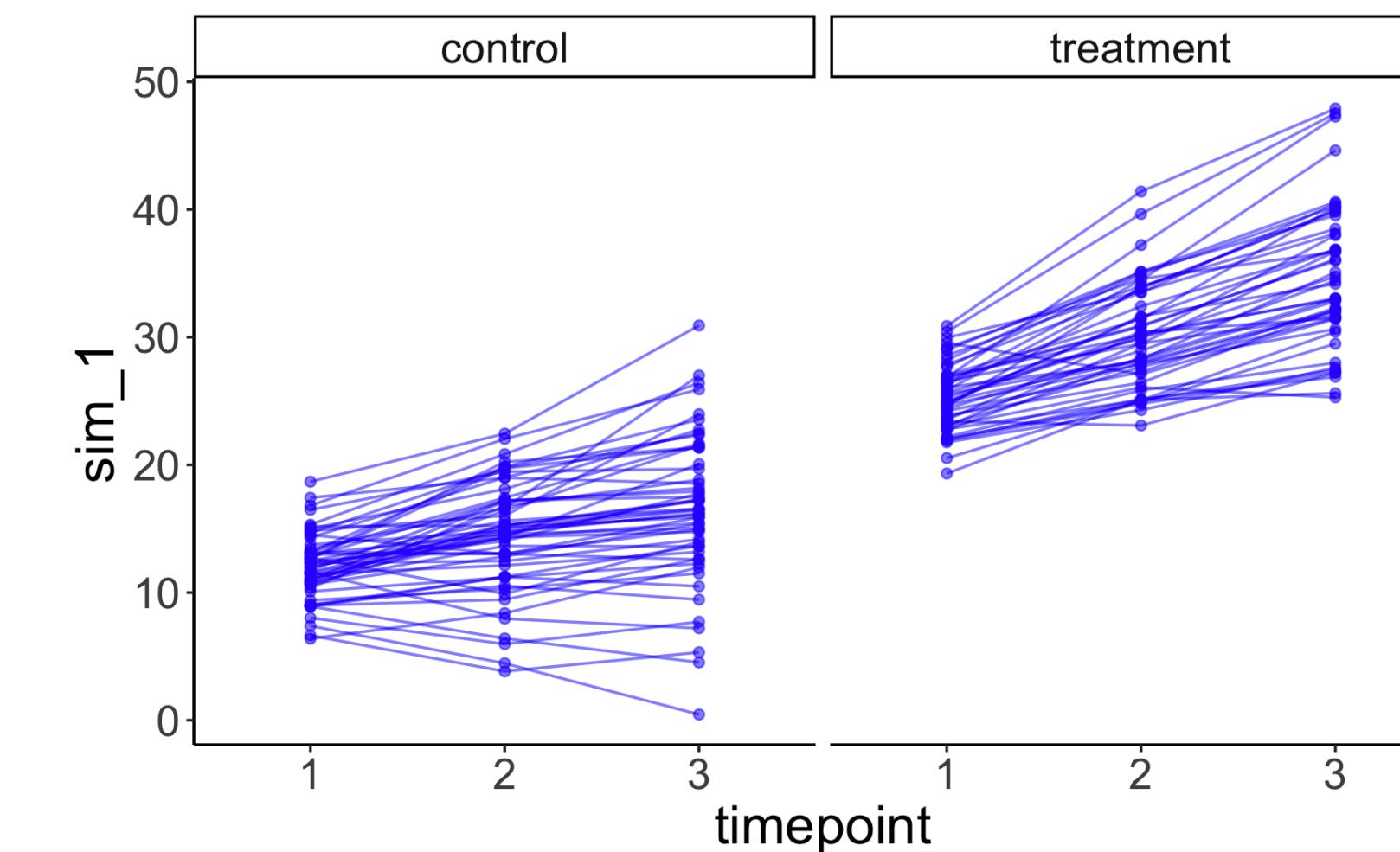
# Simulate new data from the model

```
1 set.seed(1)
2
3 fit %>%
4   simulate() %>%
5   bind_cols(df.data) %>%
6   mutate(condition = factor(condition,
7                             levels = c(0, 1),
8                             labels = c("control", "treatment")),
9          timepoint = as.factor(timepoint)) %>%
10 ggplot(data = .,
11         mapping = aes(x = timepoint,
12                         y = sim_1,
13                         group = participant)) +
14   geom_point(alpha = 0.5,
15             color = "blue") +
16   geom_line(alpha = 0.5,
17             color = "blue") +
18   facet_grid(~ condition) +
19   labs(x = "timepoint")
```

original data

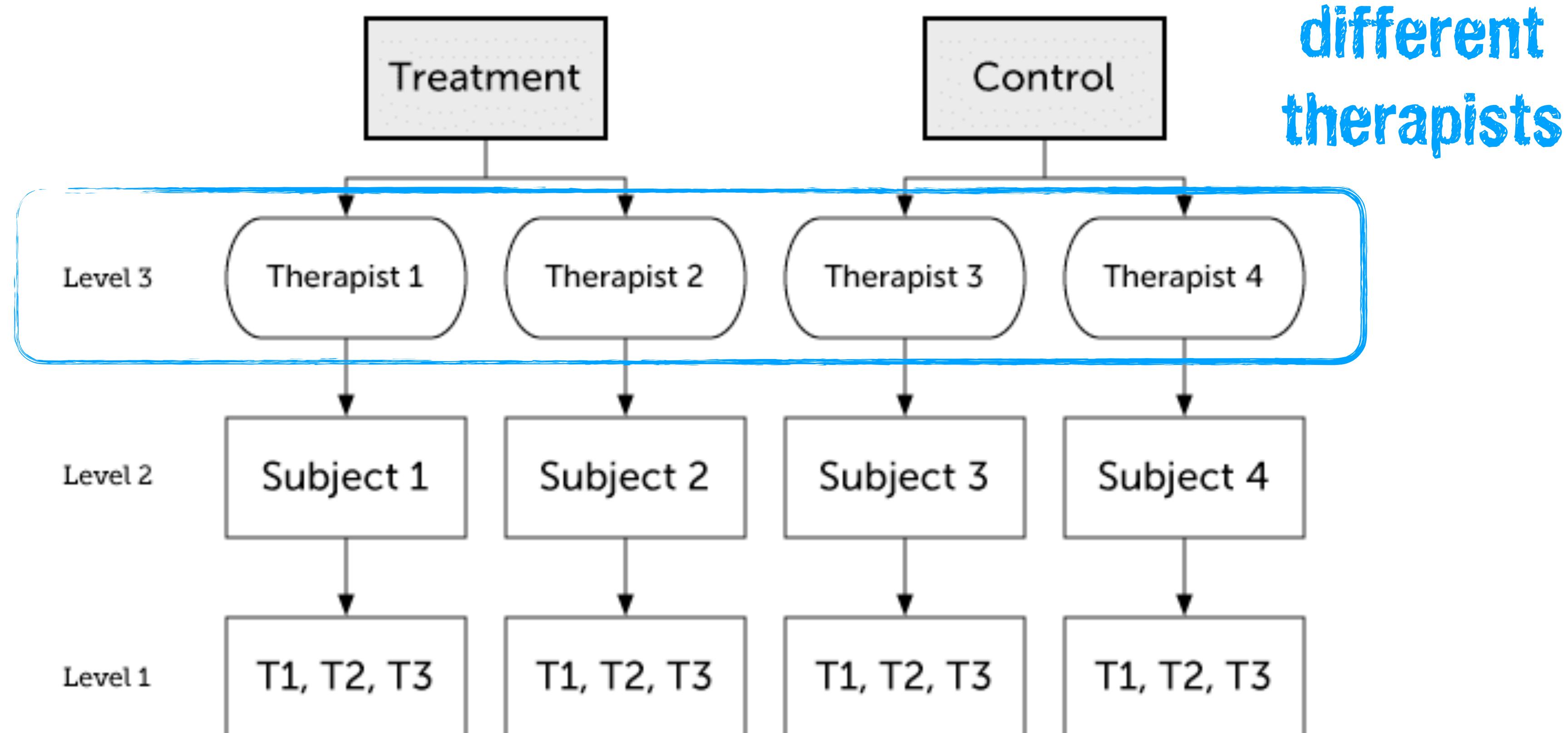


simulated data



# **Three-level model**

# Graphical representation



# Fit the model

```
1 fit = lmer(formula = value ~ 1 + condition * timepoint +
2           (1 + timepoint | therapist) +
3           (1 + timepoint | therapist:participant),
4           data = df.data)
```

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

REML criterion at convergence: 1237.9

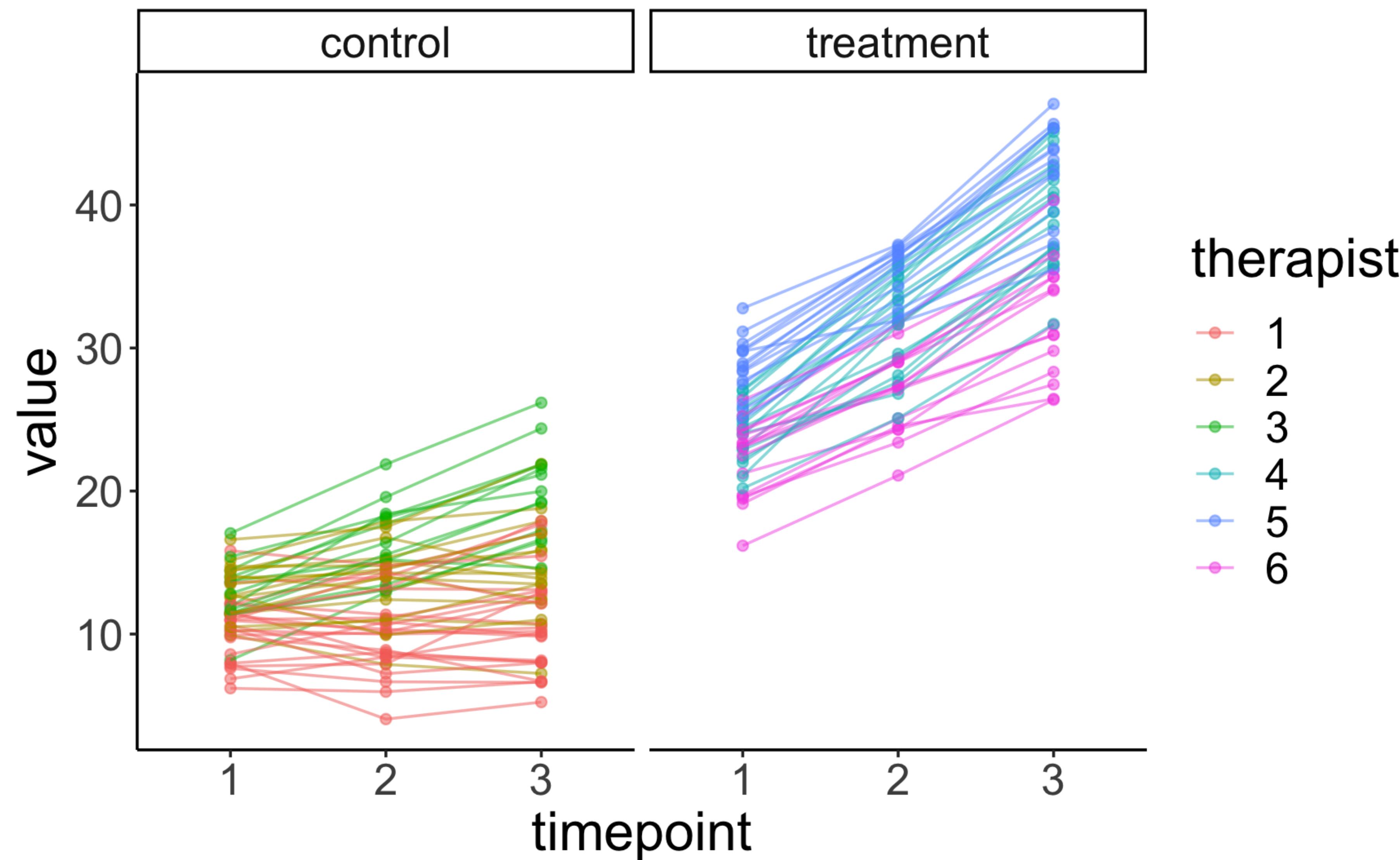
Scaled residuals:
    Min      1Q  Median      3Q     Max 
-2.02925 -0.51105  0.01576  0.48074  2.12182

Random effects:
Groups             Name        Variance Std.Dev. Corr
therapist:participant (Intercept) 2.1361   1.4615
                      timepoint   0.8205   0.9058   0.33
therapist            (Intercept) 5.6373   2.3743
                      timepoint   2.4171   1.5547   -0.21
Residual             1.0515   1.0254
Number of obs: 300, groups: therapist:participant, 100; therapist, 6

Fixed effects:
            Estimate Std. Error t value
(Intercept) 10.5078    1.4040  7.484
condition    7.7672    1.9870  3.909
timepoint   1.5160    0.9125  1.661
condition:timepoint 5.0489    1.2912  3.910

Correlation of Fixed Effects:
  (Intr) condtn timptn
condition -0.707
timepoint -0.208  0.147
condtn:tmpnt 0.147 -0.208 -0.707
```

# Plot data



# 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, Afreene Khalid, Felix Klaassen, Farnaz Mosannenzadeh, Julian Quandt**

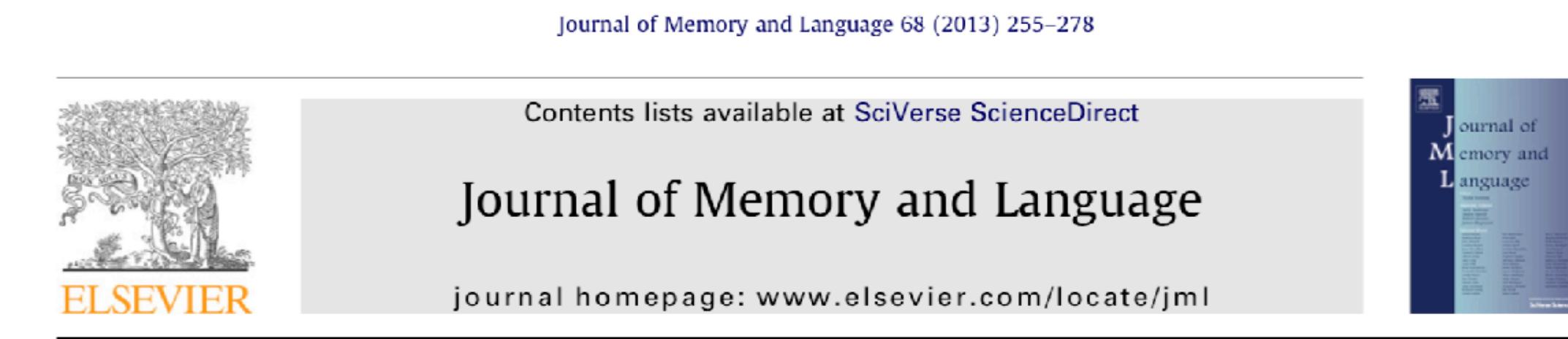
## Content/Analysis Steps

<b>Content/Analysis Steps</b>	<b>1</b>
<b>1. Before data collection:</b>	
<b>Power/ design/ planning/ sample size</b>	<b>3</b>
1.1. Power analysis	3
1.2. Sensitivity analysis	4
1.3. Sequential sampling with stopping rules	5
1.4. More readings	5
<b>2. Preparing data</b>	<b>6</b>
2.1. Categorical variables	6
2.2. Continuous variables	6
<b>3. Running the model</b>	<b>7</b>
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 <sup>a,\*</sup>, Roger Levy <sup>b</sup>, Christoph Scheepers <sup>a</sup>, Harry J. Tily <sup>c</sup>

<sup>a</sup>Institute of Neuroscience and Psychology, University of Glasgow, 58 Hillhead St., Glasgow G12 8QB, United Kingdom

<sup>b</sup>Department of Linguistics, University of California at San Diego, La Jolla, CA 92093-0108, USA

<sup>c</sup>Department 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 and slopes for participants
  - random intercepts for items
  - reduce complexity of the random effects structure step by step
  - remove the correlations 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 

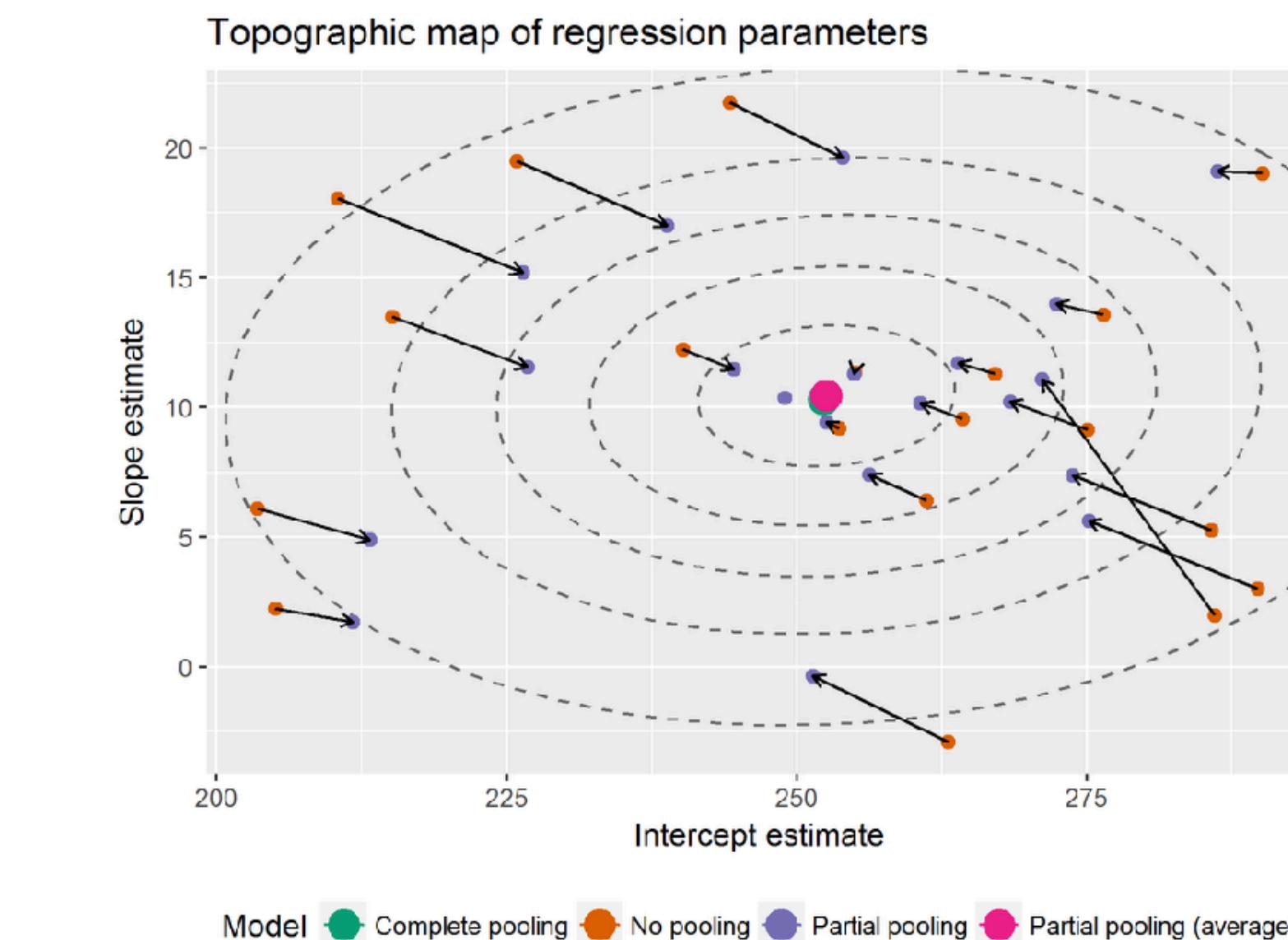
Random effects:
Groups      Name        Variance Std.Dev. Corr
subject (Intercept) 582.73   24.140
          days       35.03   5.919   0.07
Residual           649.36   25.483

Number of obs: 183, groups: subject, 20

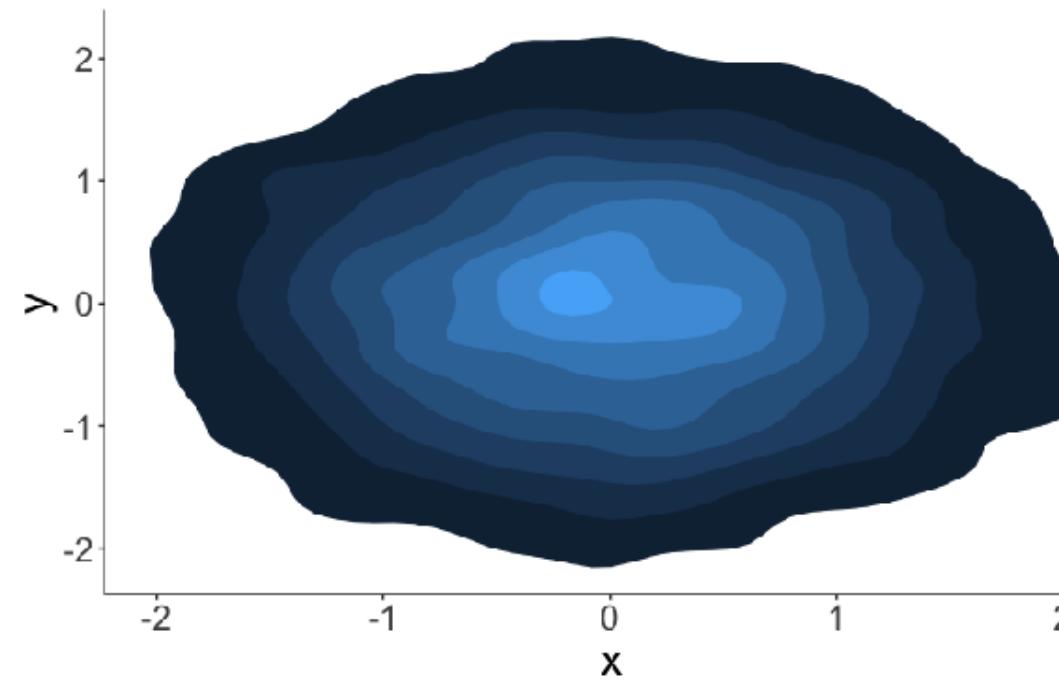
Fixed effects:
            Estimate Std. Error t value
(Intercept) 252.543    6.433 39.256
days         10.452    1.542  6.778

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

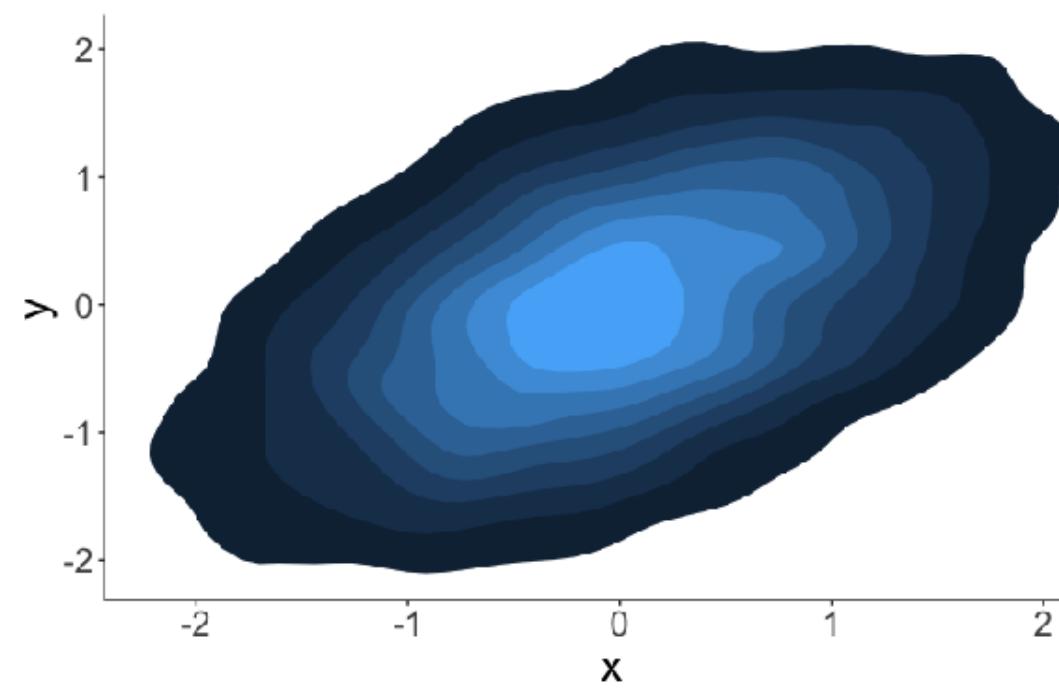
multivariate  
Gaussian



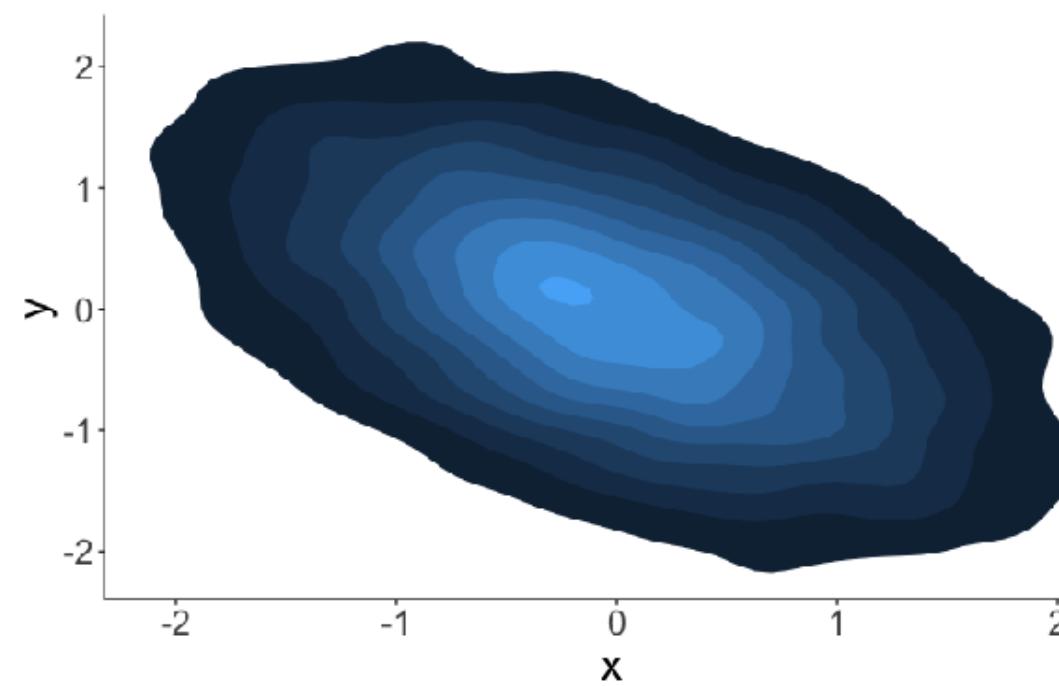
# Remove the correlation component from your model



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()
```

↑  
**random slopes**      ↑  
**random intercepts**

```
Linear mixed model fit by REML ['lmerMod']  
Formula: reaction ~ 1 + days + (0 + days | subject) + (1 | subject)  
Data: df.sleep  
  
REML criterion at convergence: 1771.5  
  
Scaled residuals:  
    Min      1Q  Median      3Q     Max  
-3.9805 -0.4673  0.0250  0.4589  5.2083  
  
Random effects:  
Groups   Name        Variance Std.Dev.  
subject  days       35.88    5.99  
subject.1 (Intercept) 598.11   24.46  
Residual           647.90   25.45  
Number of obs: 183, groups: subject, 20  
  
Fixed effects:  
            Estimate Std. Error t value  
(Intercept) 252.550     6.491  38.907  
days         10.439     1.556   6.708  
  
Correlation of Fixed Effects:  
  (Intr)  
days -0.184
```

independent Gaussians

# Remove the correlation component from your model

```
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()
```

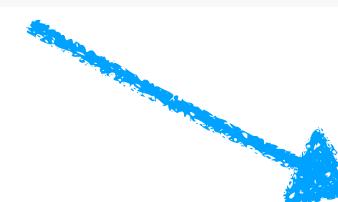
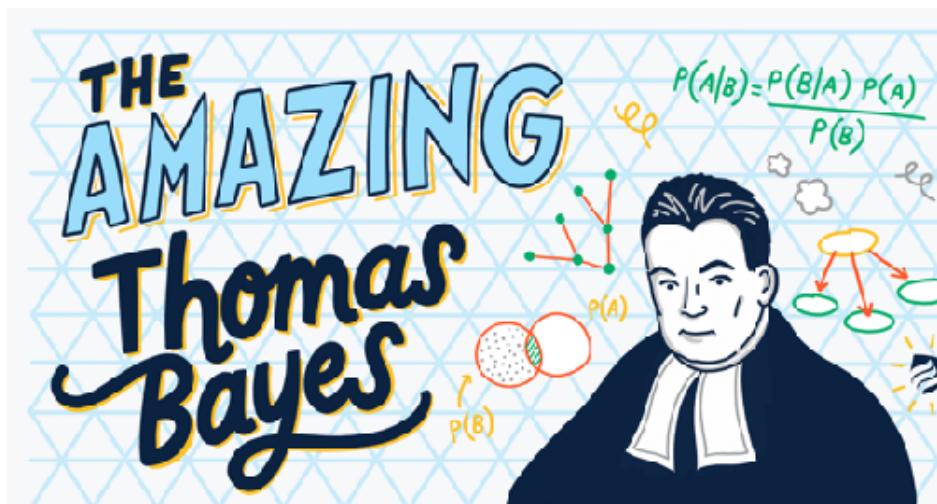
```
Linear mixed model fit by REML ['lmerMod']  
Formula: reaction ~ 1 + days + (0 + days | subject) + (1 | subject)  
Data: df.sleep  
  
REML criterion at convergence: 1771.5  
  
Scaled residuals:  
    Min     1Q Median     3Q    Max  
-3.9805 -0.4673  0.0250  0.4589  5.2083  
  
Random effects:  
 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 -0.184
```

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

independent  
Gaussians

# 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 *B/C* or *A/C*.



## 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

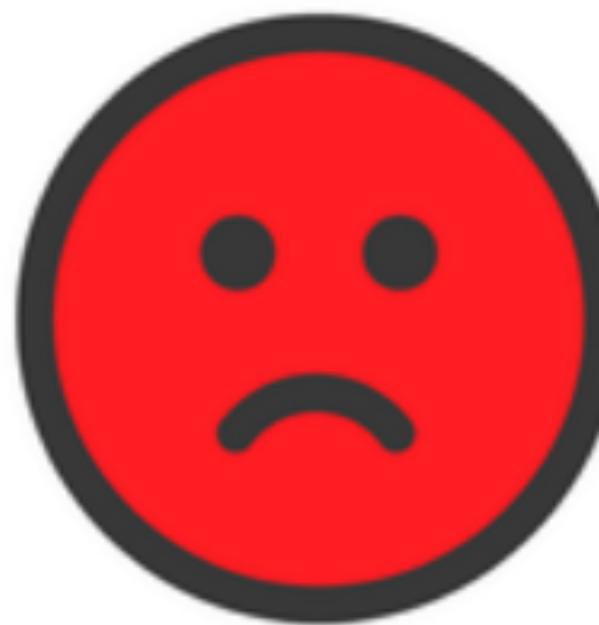
# Plan for today

- Quick recap
- Linear Mixed Model
  - Accommodating non-independence in data
  - Understanding lmer() syntax
  - A worked example
  - Simplicon's Paradox
  - Reporting results
  - Understanding lmer() syntax
  - Reporting results
- **Let's simulate some lmer()**s
- lmer() standard operating procedures
- Some more examples

# How was the pace of today's class?

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

# 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!