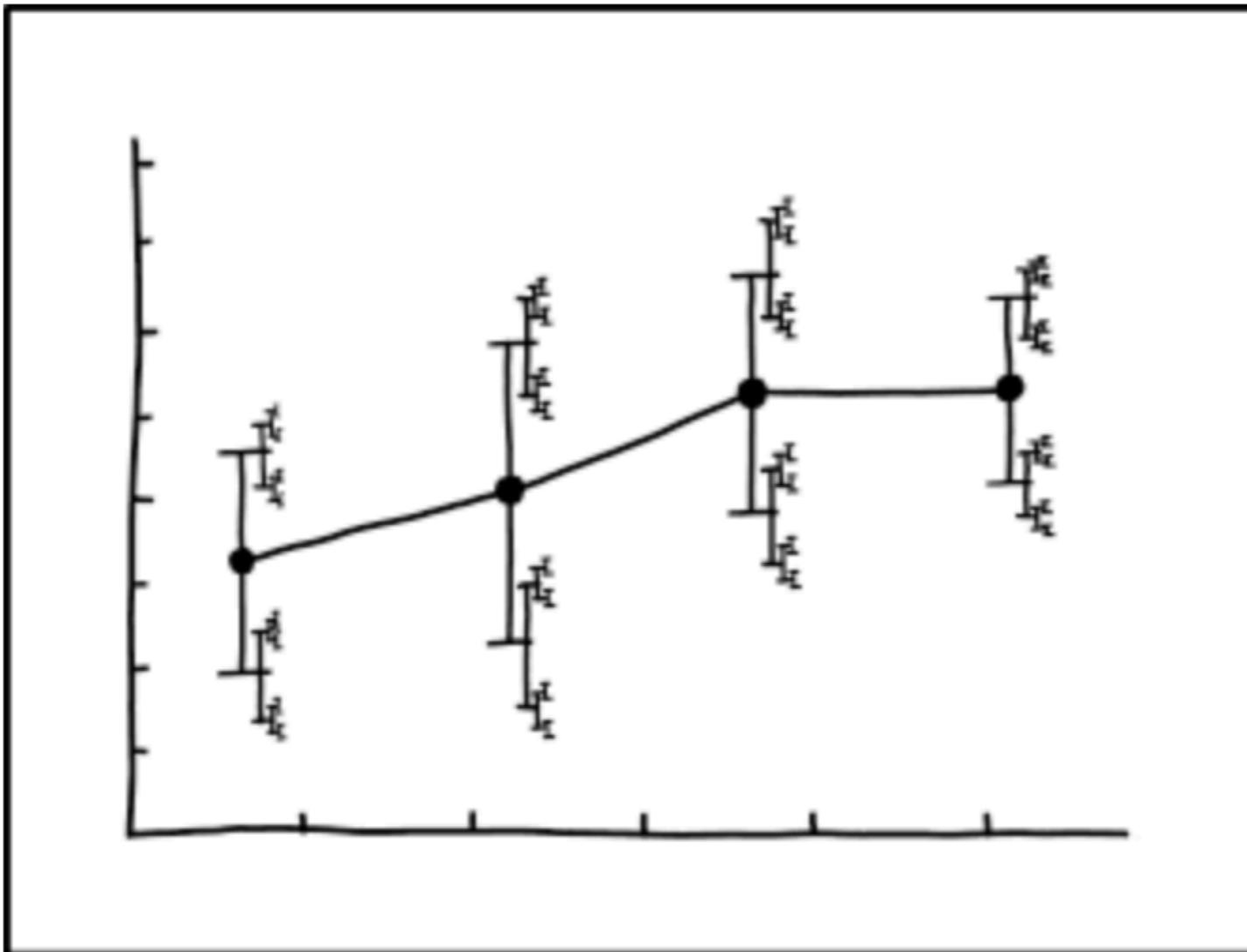


# Linear mixed effects models 2



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

# **Logistics**

# **Midterm**

# Midterm

- we will post results by **Wednesday 26th**
- the midterm accounts for only **20% of the final grade** (40% homeworks, and 40% final project)
- Friday 28th is the deadline for dropping the course
- but: we don't want for any of you to feel like you have to drop the course
- our goal is to help you learn the skills that will help you do good science!
- adopt a growth mindset! stats & R are difficult but worth it!

# Coding

# Coding style matters

- and it matters more and more, the more complicated the things are that you'd like to do (e.g. simulating a power analysis)
- we want to encourage you to use good coding style (this will greatly help your collaborators [including your future self])
- take a close look at how we do things in the homework/midterm solutions, and the course notes
  - Please leave the `\clearpage` commands where they are. This makes sure that each question is printed on a separate page in the pdf.
  - Some code chunks are set to `eval=F`, make sure to set these to `eval=T` before knitting the final version.
  - We note for each question how many points you can get. You can get up to 120 points in total.
  - Good coding style matters! We will add or subtract up to 5 points depending on style.

# **Homework 5**

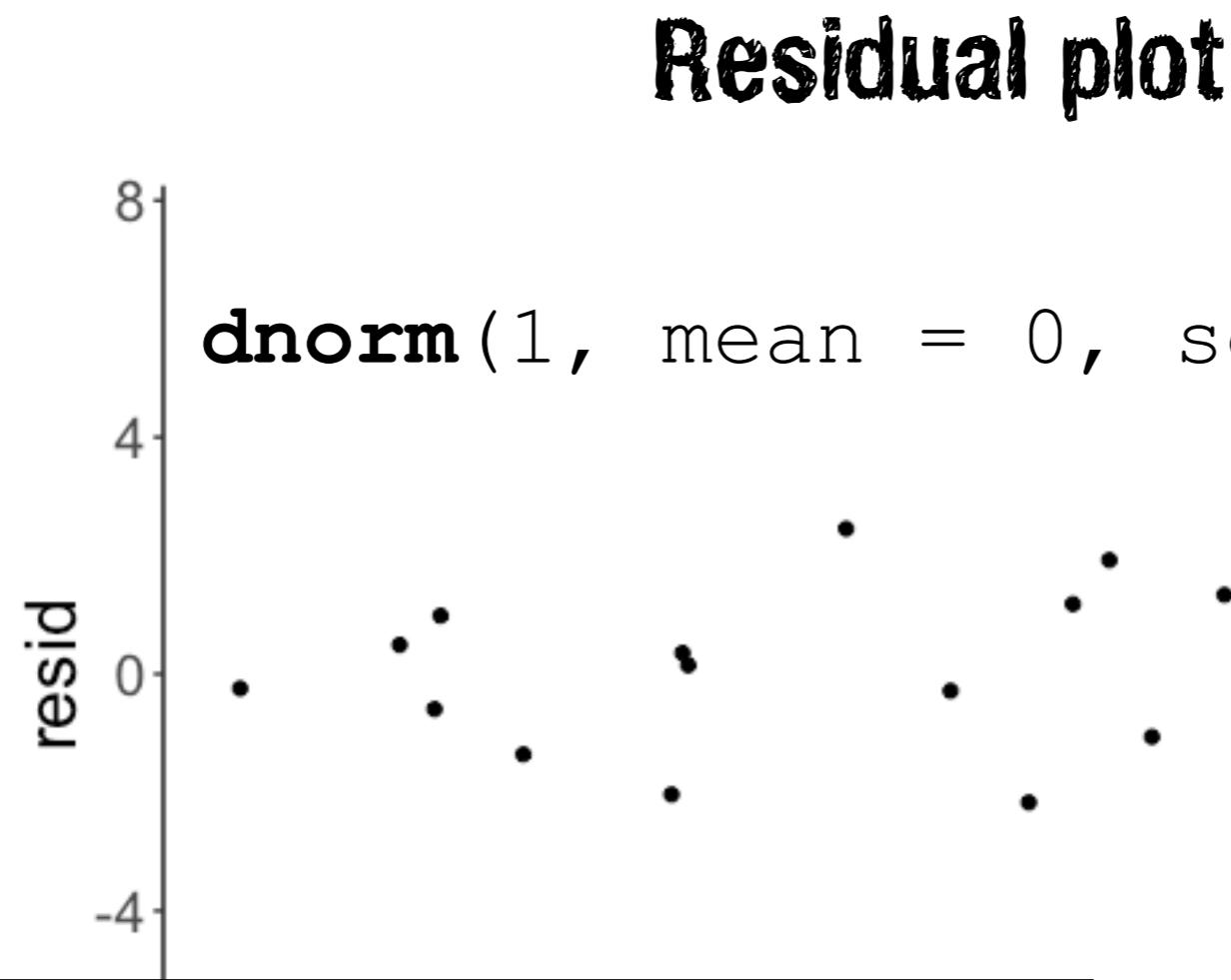
# Homework 5

Will be released on Saturday night.

Will be due **Friday 28th at 8pm**

# **Things that came up**

# AIC and BIC



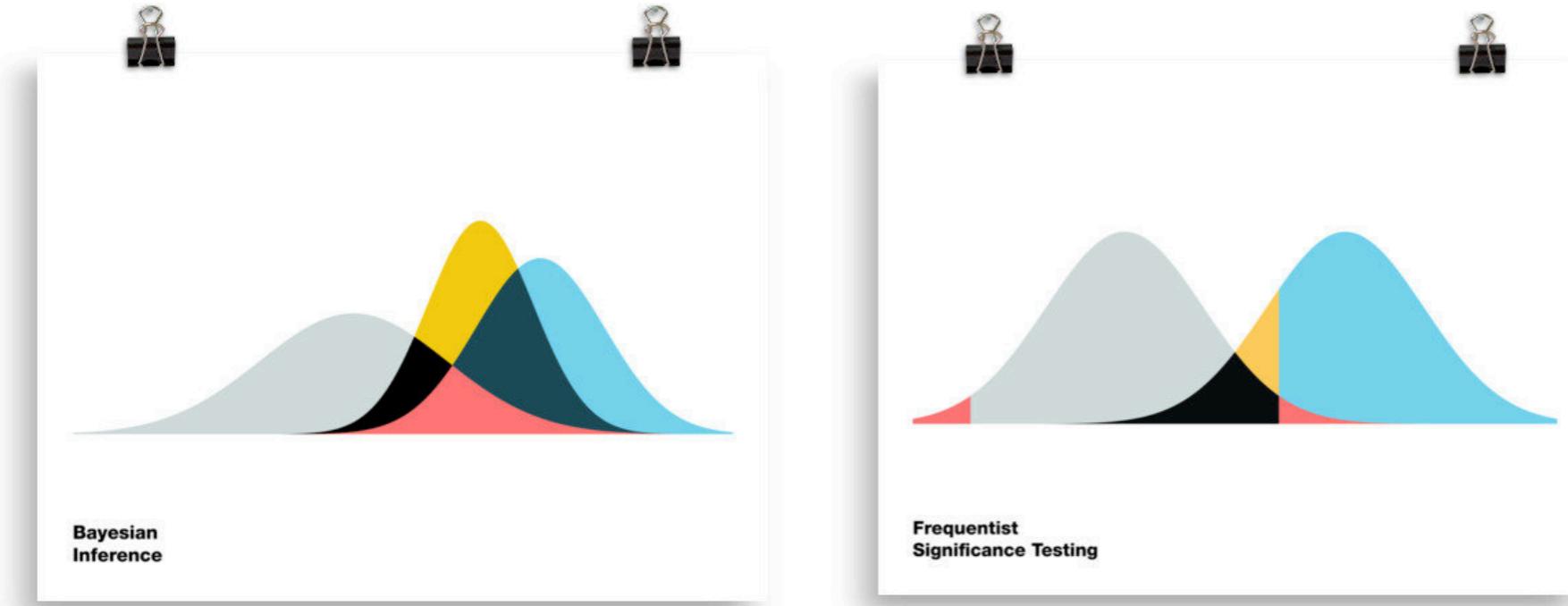
normal distribution  
over residuals

optimization

find the normal distribution  
that maximizes the  
likelihood of the data

# Visualizing likelihood

soon in my  
office



## Understanding Maximum Likelihood

An Interactive Visualization

Created by [Kristoffer Magnusson](#)



<https://rpsychologist.com/d3/likelihood/>

# Plan for today

- Linear mixed effects model
  - what are they?
  - when to use them?
  - understanding the `model summary()`
- Let's simulate some `lmer()`s
  - effect of outliers
  - different slopes
  - Simpson's paradox
- A worked example
  - pooling:
    - complete pooling
    - no pooling
    - partial pooling
  - shrinkage

# Plan for today

- **Linear mixed effects model**
  - what are they?
  - when to use them?
  - understanding the `model summary()`
- Let's simulate some `lmer()`s
  - effect of outliers
  - different slopes
  - Simpson's paradox
- A worked example
  - pooling:
    - complete pooling
    - no pooling
    - partial pooling
  - shrinkage

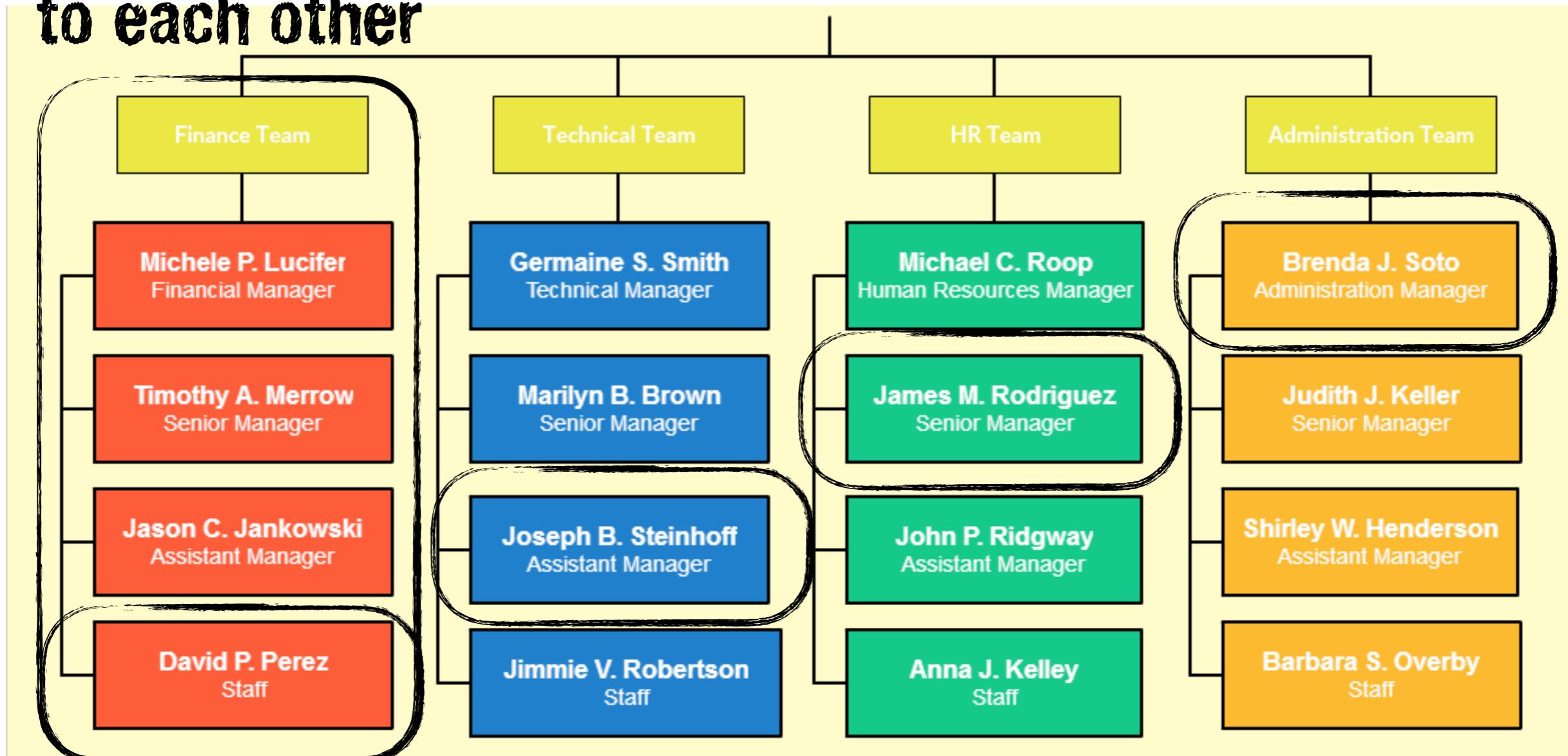
# **Linear mixed effects models**

# Dependence

- so far, all the models that we've discussed (linear model with different kinds of predictors and contrasts) make the assumption that the residuals are **iid** (independent, and identically distributed)
- often this assumption is violated
  - **psychology experiments**: many observations from the same participants
  - **survey data**: different populations between different states in the US
  - **time series**: distribution at  $t + 1$  depends on  $t$

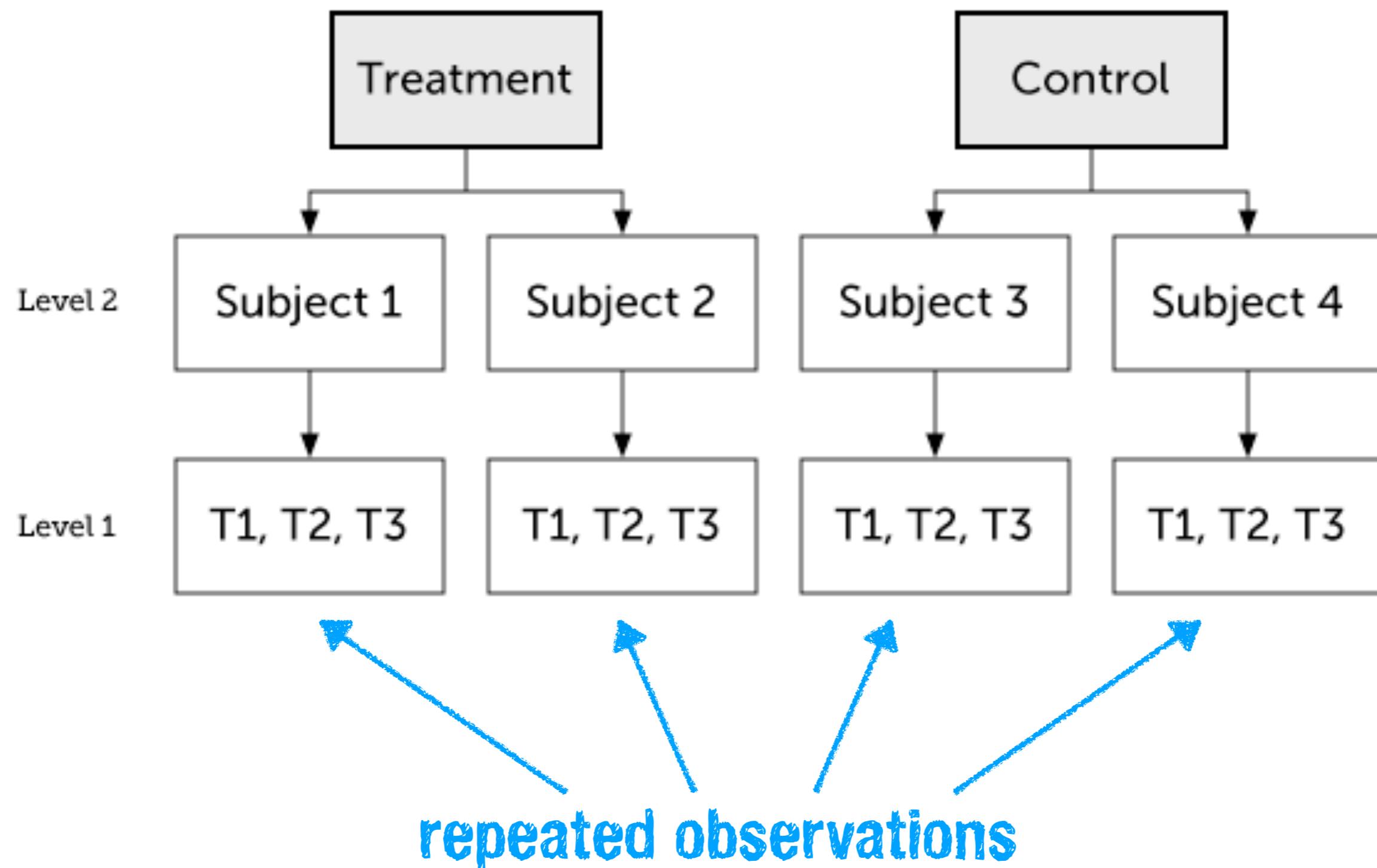
# Main use cases: Hierarchical models

more similar  
to each other



less similar to  
each other

# Main use cases: Longitudinal models



# general points about mixed effects models

- **fixed effects:**

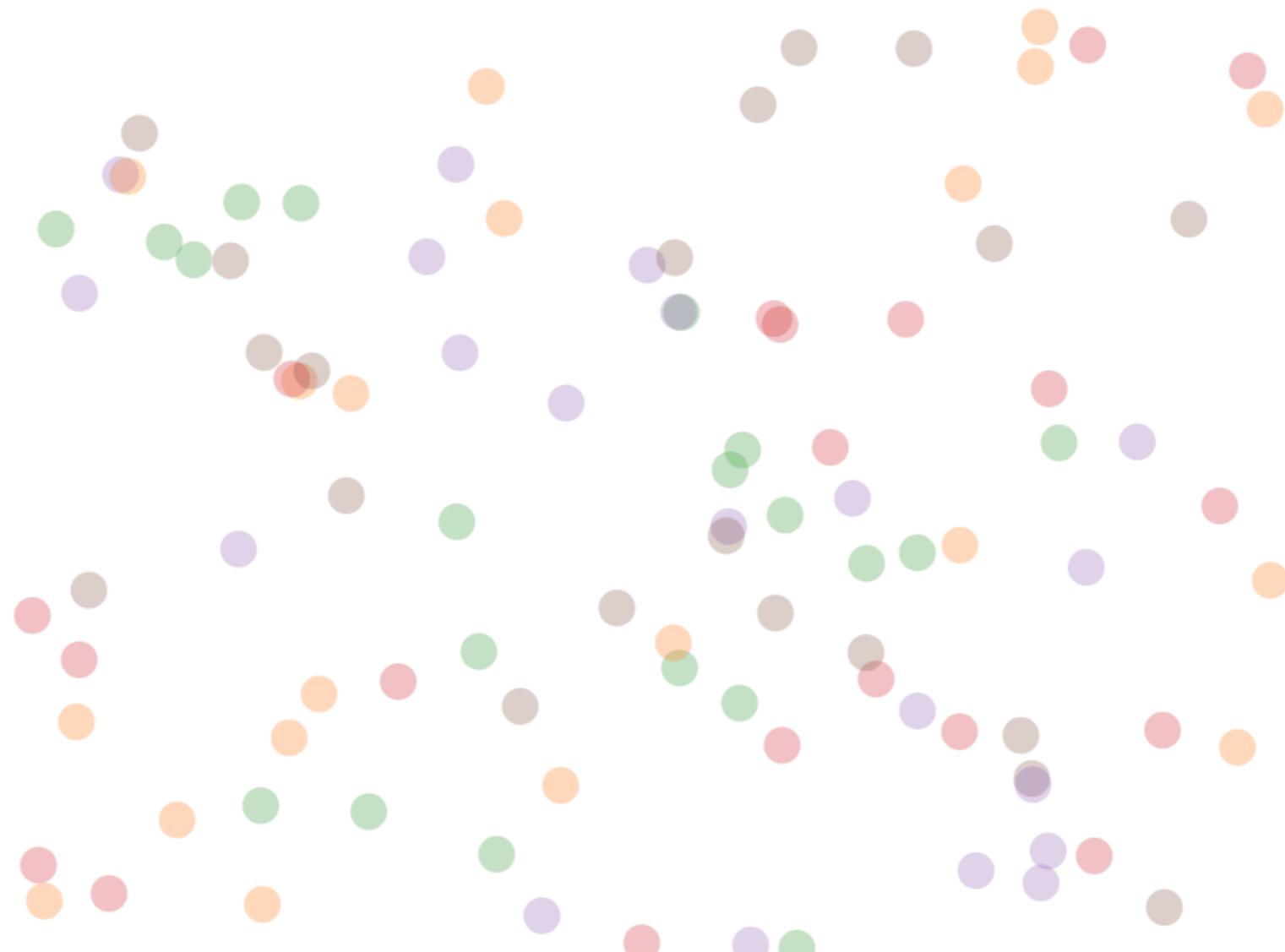
- often: factors that we manipulate experimentally
- parameters are estimated --> we are interested in characterizing the relationship between this variable and the outcome

- **random effects:**

- variation we want to control for
- often: differences between participants (or items) in our experiment
- sampling viewpoint: we explicitly model the variation in participants (or items)

# An Introduction to Hierarchical Modeling

This visual explanation introduces the statistical concept of **Hierarchical Modeling**, also known as *Mixed Effects Modeling* or by [these other terms](#). This is an approach for modeling **nested data**. Keep reading to learn how to translate an understanding of your data into a hierarchical model specification.

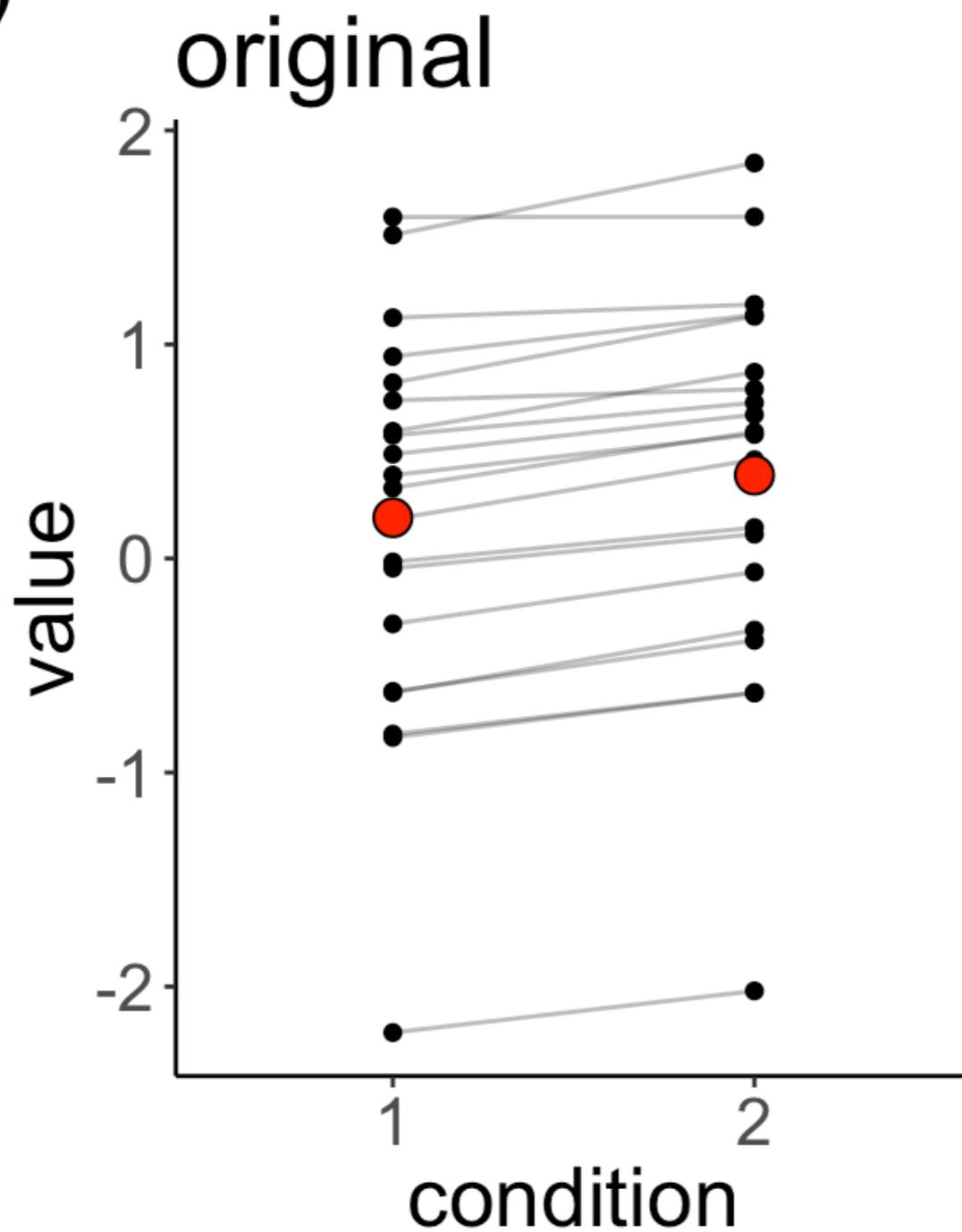


# Dependence

Does it really matter?

Is there a significant difference  
between conditions 1 and 2?

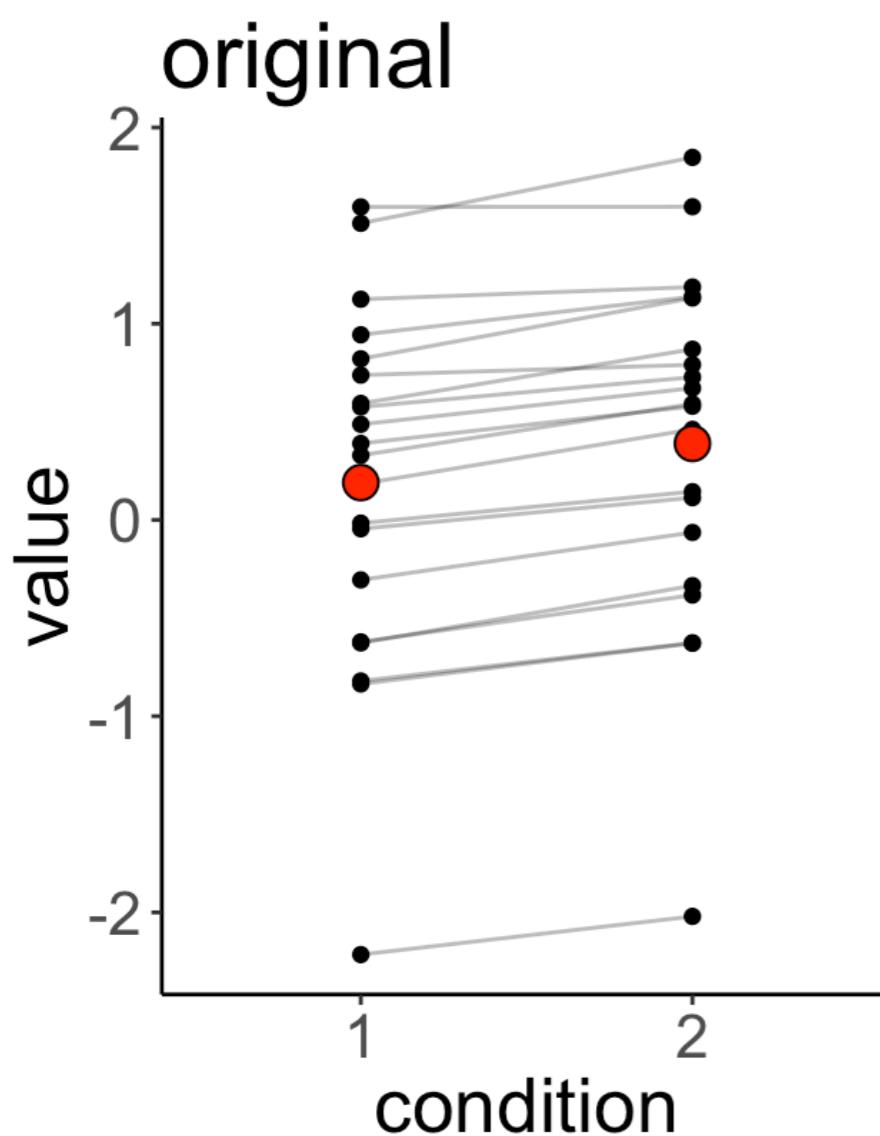
a)



# Dependence

assuming independence!

```
1 # linear model  
2 lm(formula = value ~ condition,  
3     data = df.original) %>%  
4 summary()
```



```
Call:  
lm(formula = value ~ condition, data = df.original)  
  
Residuals:  
    Min      1Q  Median      3Q     Max  
-2.4100 -0.5530  0.1945  0.5685  1.4578  
  
Coefficients:  
              Estimate Std. Error t value Pr(>|t|)  
(Intercept)  0.1905    0.2025   0.941  0.353  
condition2    0.1994    0.2864   0.696  0.491  
  
Residual standard error: 0.9058 on 38 degrees of freedom  
Multiple R-squared:  0.01259,    Adjusted R-squared: -0.0134  
F-statistic: 0.4843 on 1 and 38 DF,  p-value: 0.4907
```

- we ignore the fact that we have repeated observations from the same participants
- in the data it looks like there is a small but consistent effect of condition

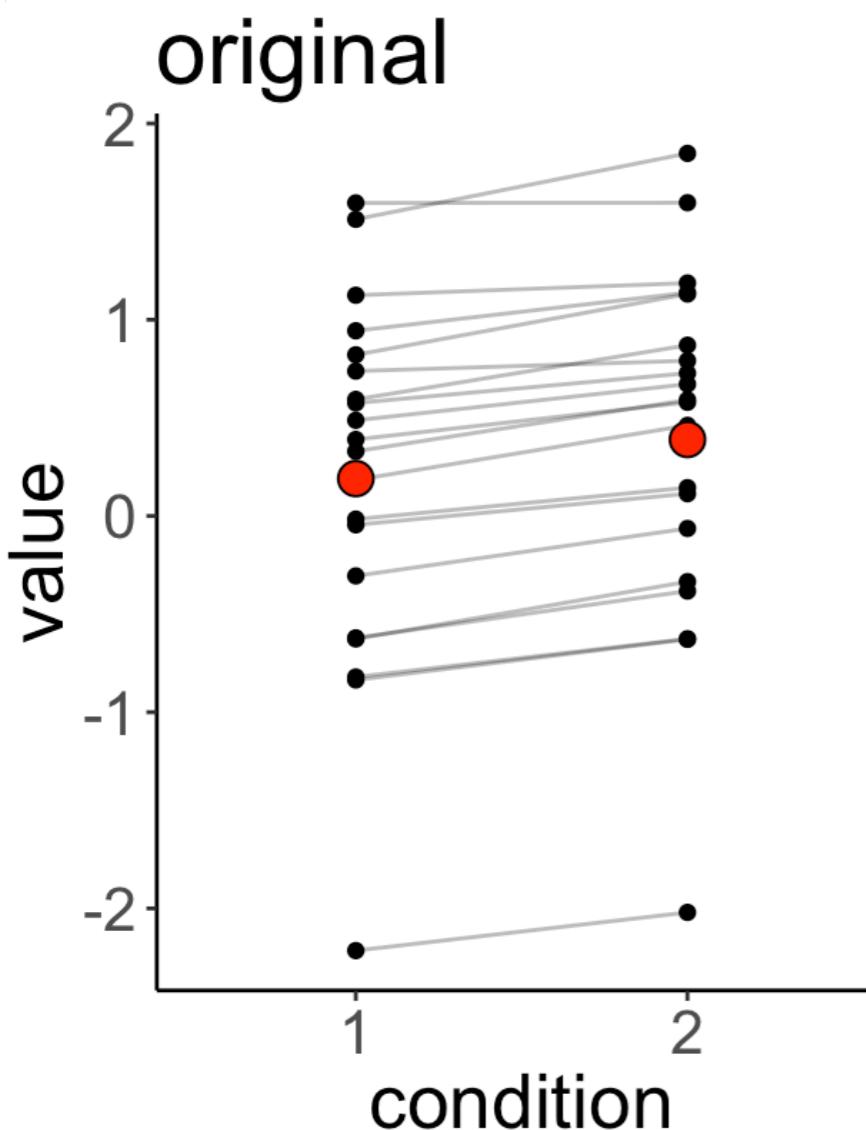
# meet lmer()



# Dependence

new syntax

```
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
```

no p-value!

NO P-VALUE



# Dependence

we can still do our good ol' model comparison trick

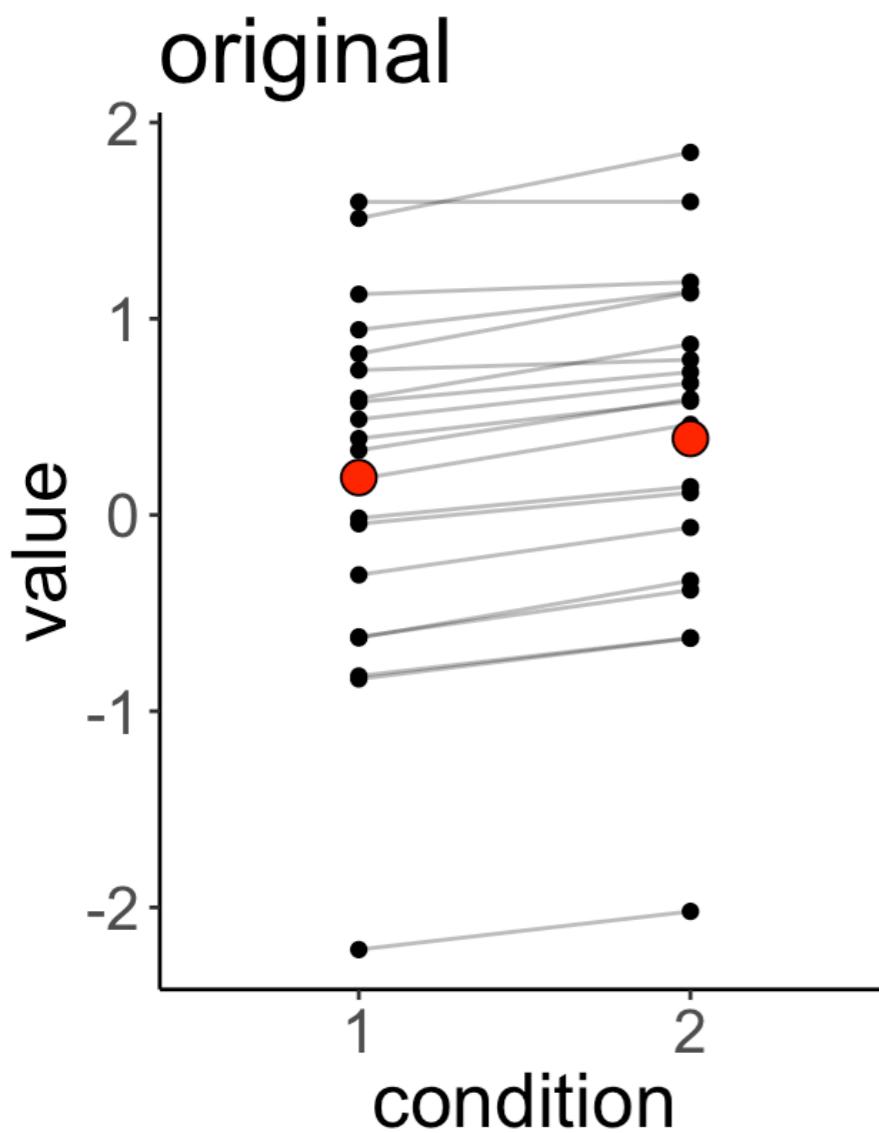
```
1 # fit models
2 fit.compact = lmer(formula = value ~ 1 + (1 | participant),
3                     data = df.original)

4 fit.augmented = lmer(formula = value ~ 1 + condition + (1 | participant),
5                     data = df.original)
6
7 # compare via Chisq-test
8 anova(fit.compact, fit.augmented)
```

```
refitting model(s) with ML (instead of REML)
Data: df.original
Models:
fit.compact: value ~ 1 + (1 | participant)
fit.augmented: value ~ 1 + condition + (1 | participant)
              Df     AIC     BIC   logLik deviance    Chisq Chi Df Pr(>Chisq)
fit.compact     3 53.315 58.382 -23.6575     47.315
fit.augmented   4 17.849 24.605  -4.9247      9.849 37.466          1 9.304e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# Dependence

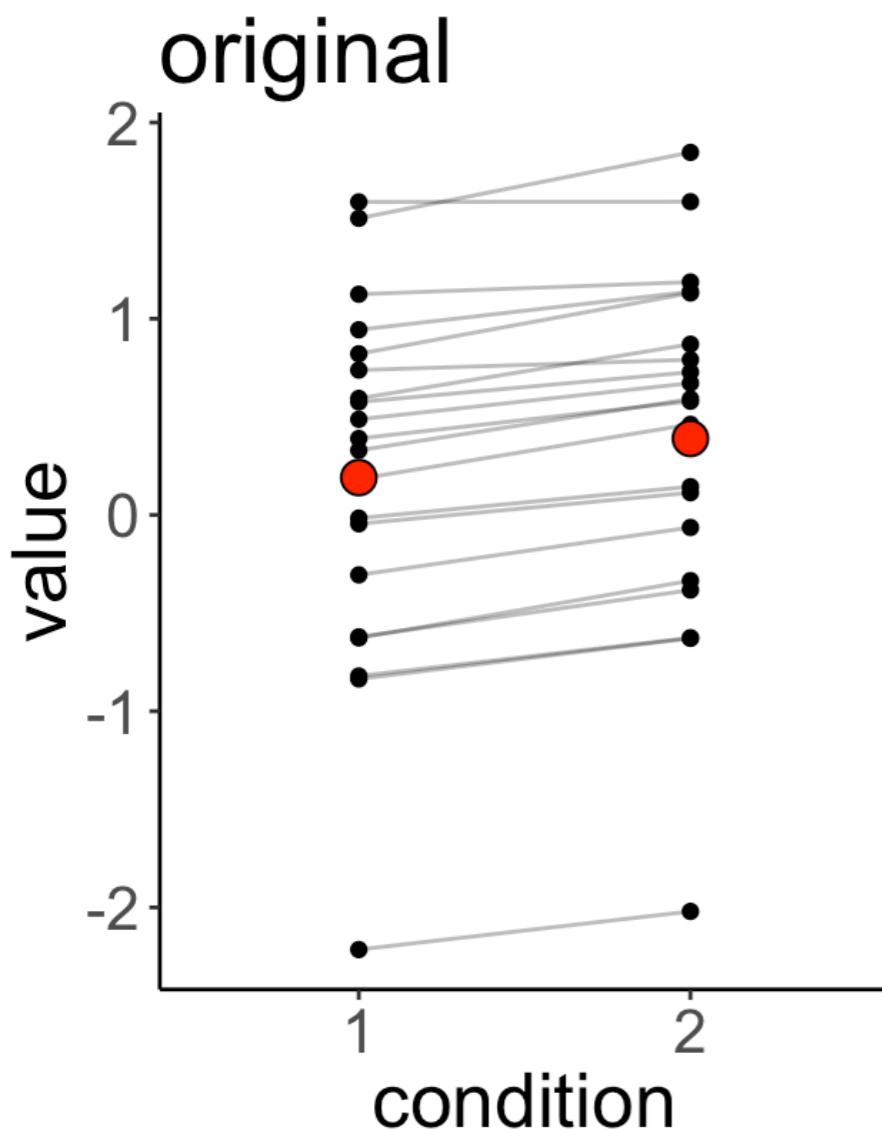
**Why is the effect of condition significant when we account for the dependence in the data?**



- there are large interindividual differences
- the variance explained by the effect of condition is (much) smaller than the interindividual variance
- **but:** the effect of condition is highly consistent

# Dependence

**Why is the effect of condition significant when we account for the dependence in the data?**



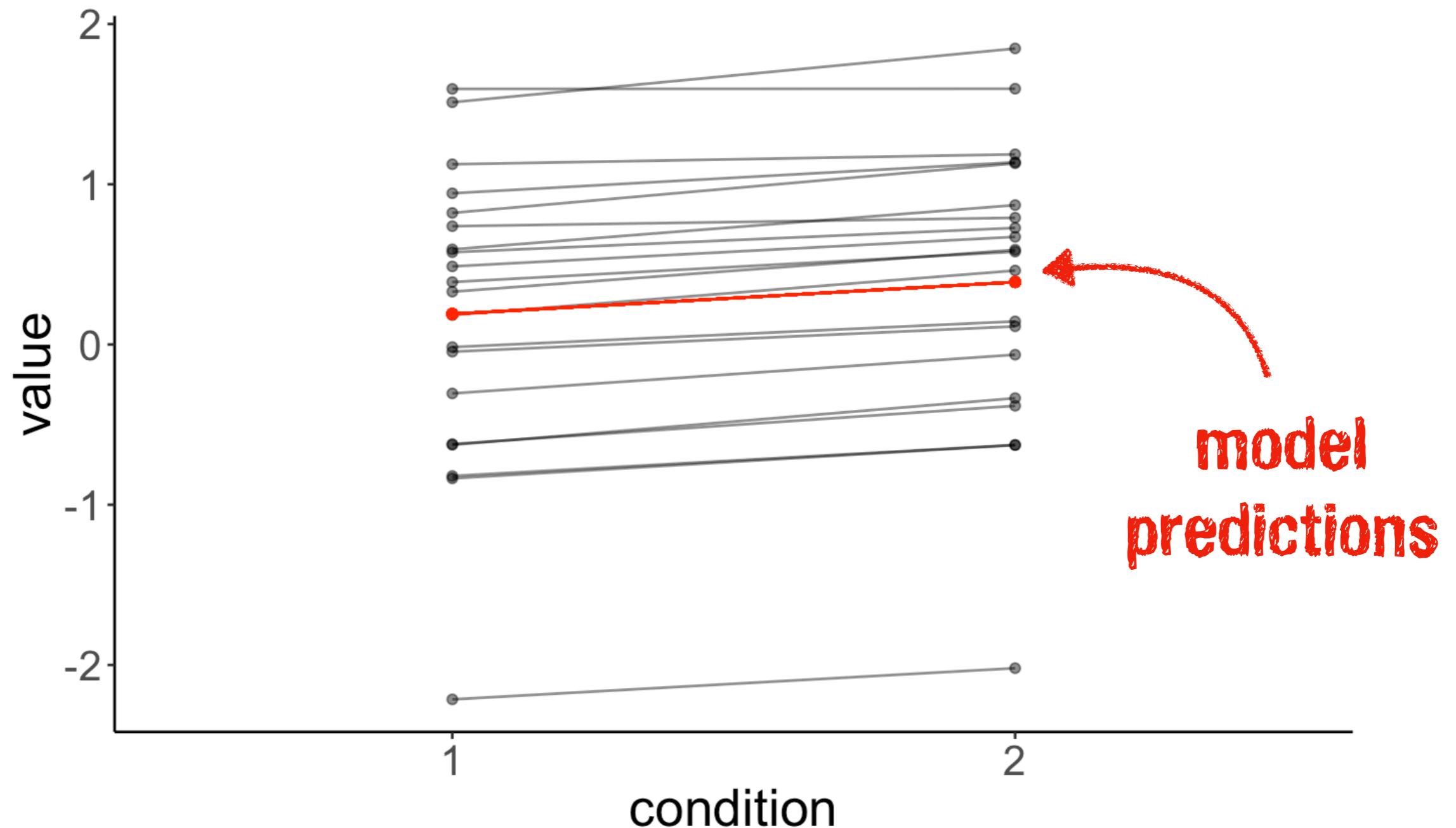
- by explicitly modeling the dependence in the data, we account for the interindividual differences

**let's visualize the model predictions!**

# Linear model (assuming independence)

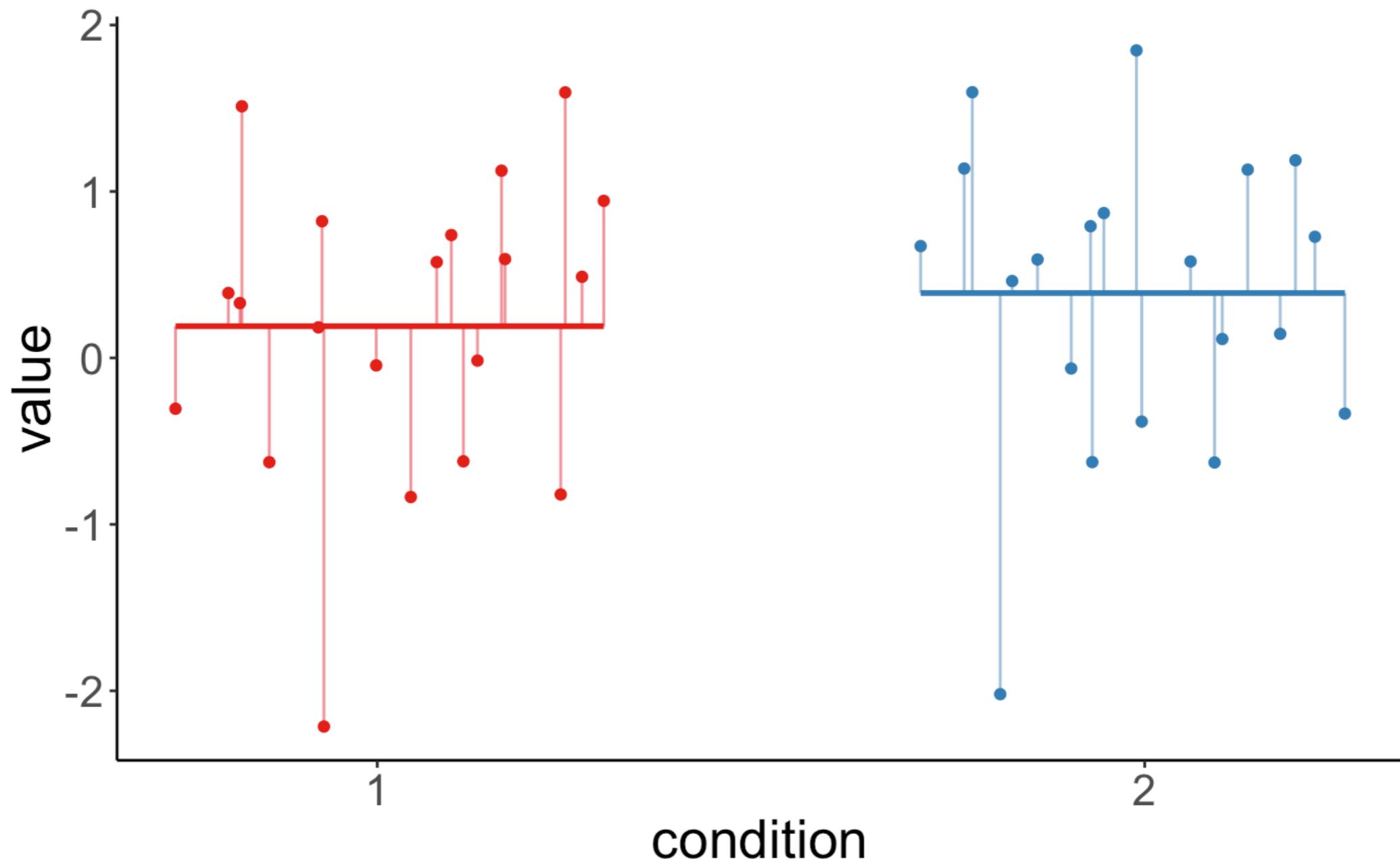
Predictions by the linear model which assumes independence

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



# Linear model (assuming independence)

## Residuals of the model

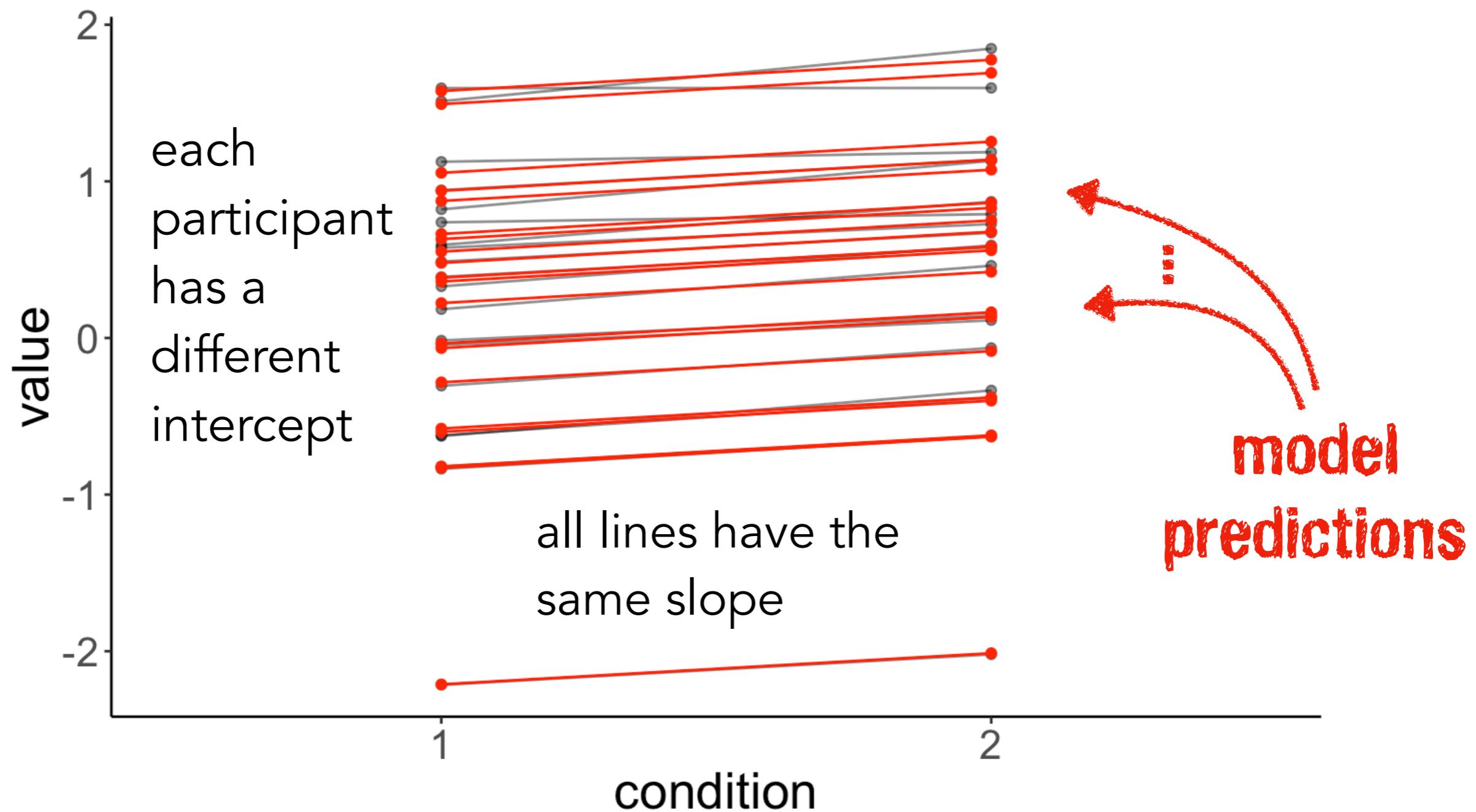


This is not much better than fitting a single line (point).

# Linear mixed effects model (accounting for dependence)

## Predictions by the linear mixed effects model

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

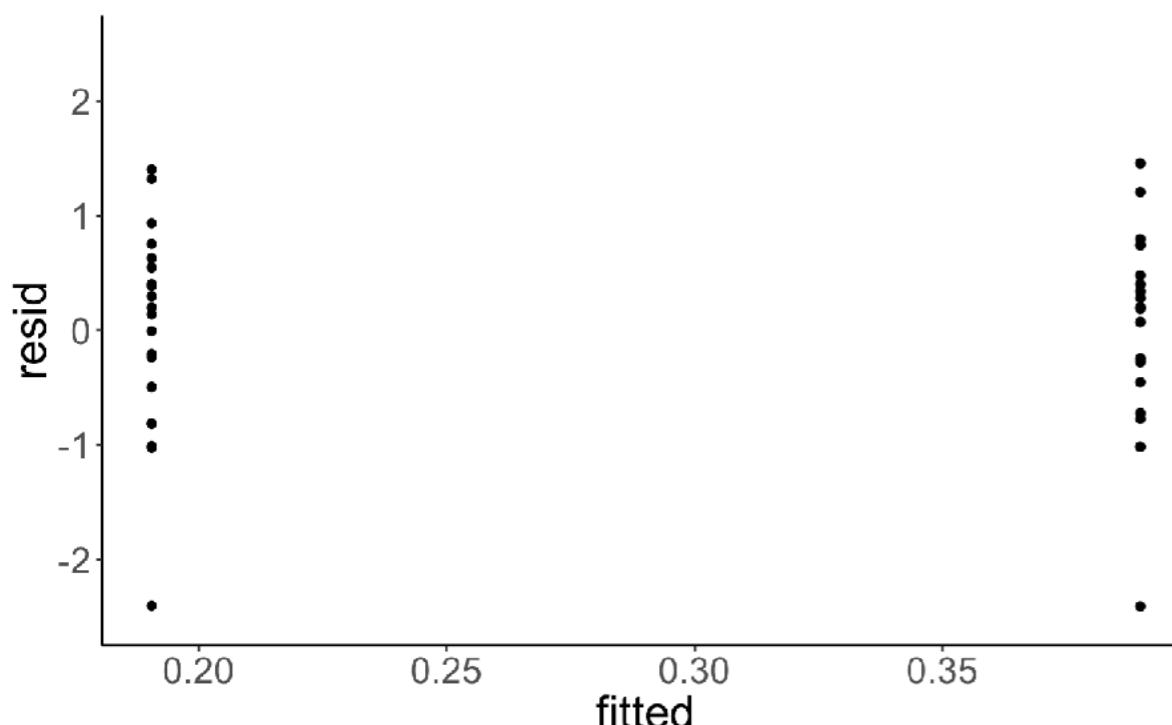


# Model comparison

## Residual plots

### linear model

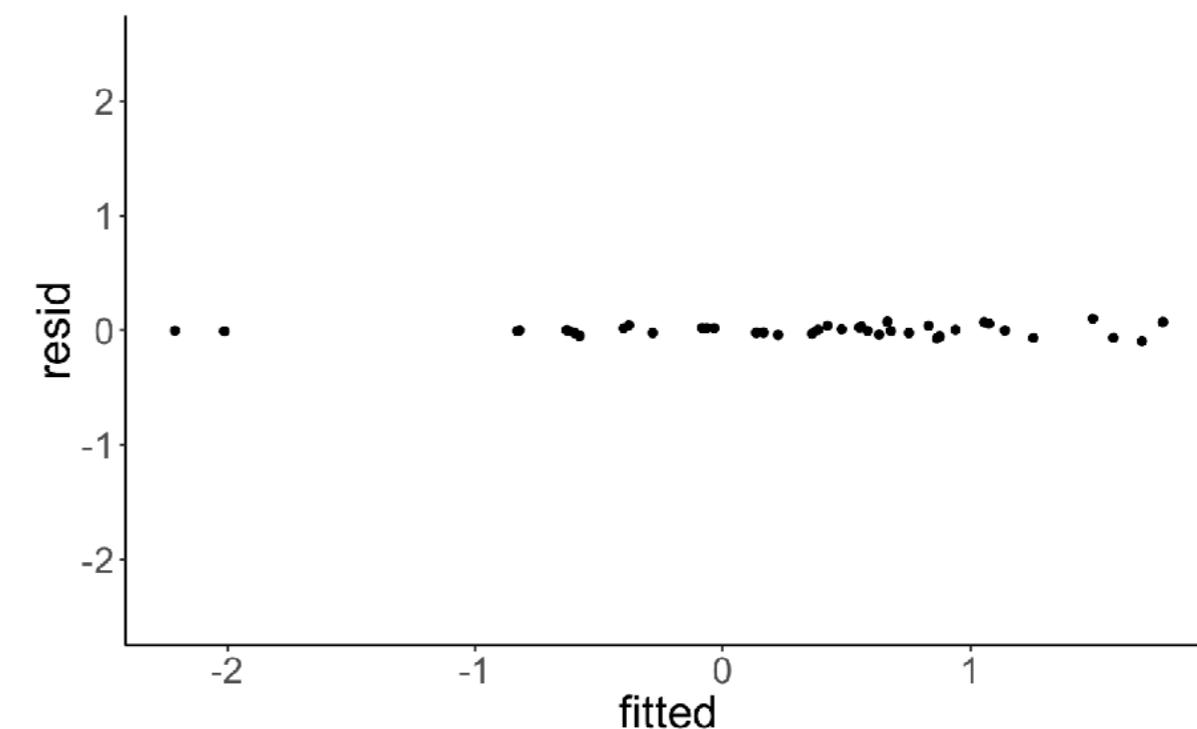
```
lm(formula = value ~ 1 + condition,  
  data = df.original)
```



**much variance left  
to be explained**

### linear mixed effects model

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



**almost all variance  
explained**

# Model comparison

# Hypothesis test

Is taking into account individual differences worth it?

```
1 # fit models (without and with dependence)
2 fit.compact = lm(formula = value ~ 1 + condition,
3                   data = df.original)
4
5 fit.augmented = lmer(formula = value ~ 1 + condition + (1 | participant),
6                       data = df.original)
7
8 # compare models
9 # note: the lmer model has to be supplied first
10 anova(fit.augmented, fit.compact)
```

refitting model(s) with ML (instead of REML)

Data: df.original

Models:

fit.compact: value ~ 1 + condition

fit.augmented: value ~ 1 + condition + (1 | participant)

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
fit.compact	3	109.551	114.617	-51.775	103.551			
fit.augmented	4	17.849	24.605	-4.925	9.849	93.701	1	< 2.2e-16 ***
---								
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1								

## 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)
```

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

i = participant,  
j = time point

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

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

$b_0, b_1$  = 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

# 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
```

**REML** = restricted maximum likelihood method for fitting models with **random effects**

# 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	-0.048

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

**lmer()** complains when it didn't work

# 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	-0.048

summary information  
about residuals

# 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	10	Median	3Q	Max
-1.55996	-0.36309	-0.03341	0.34400	1.65823

```
Random effects:
```

Groups	Name	Variance	Std.Dev.
participant	(Intercept)	0.816722	0.90373
Residual		0.003796	0.06161

```
Number of obs: 40, groups: participant, 20
```

```
Fixed effects:
```

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

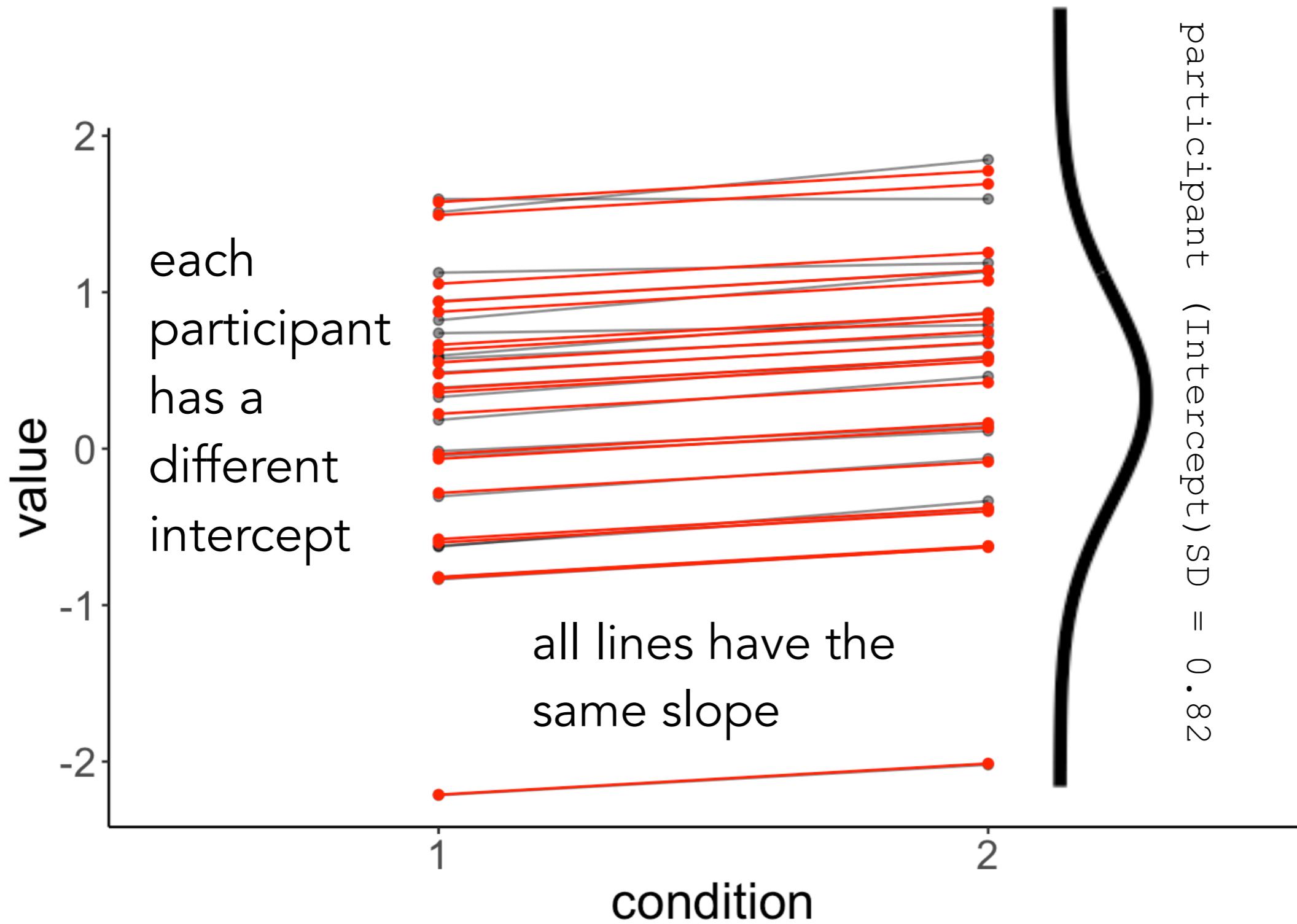
```
Correlation of Fixed Effects:
```

	(Intr)
condition2	-0.048

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

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

# Understanding the `lmer()` summary



# Understanding the **lmer()** summary

```
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	-0.048

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

```
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
```

correlation between intercept and condition2

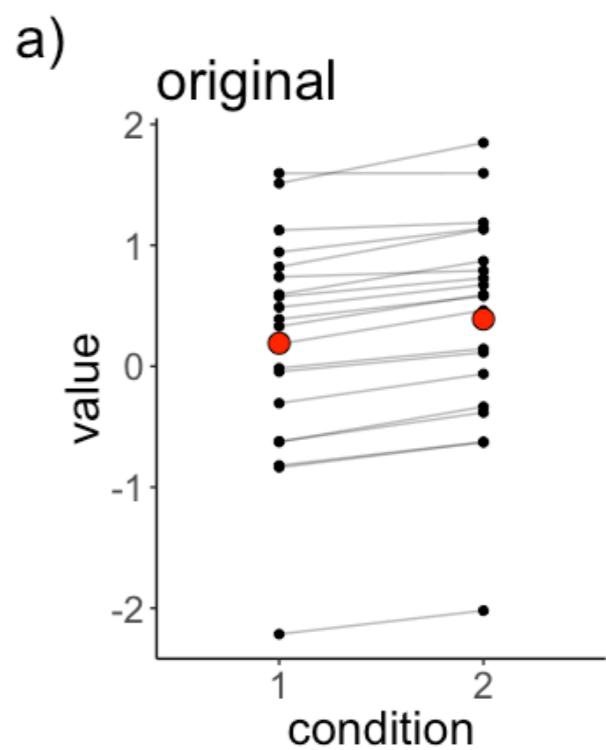
The "correlation of fixed effects" output doesn't have the intuitive meaning that most would ascribe to it. Specifically, is not about the correlation of the variables. It is in fact about the expected correlation of the regression coefficients.

# we just performed a paired t-test ...

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

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

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



# we just performed a paired t-test ...

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

- explicitly models the interindividual variation
- much more flexible ...

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

Paired t-test

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

```
Linear mixed model fit by REML ['lmerMod']  
Formula: value ~ condition + (1 | participant)  
Data: df.original  
  
REML criterion at convergence: 17.3  
  
Scaled residuals:  
    Min     1Q Median     3Q    Max  
-1.55996 -0.36399 -0.03341  0.34400  1.65823  
  
Random effects:  
Groups   Name        Variance Std.Dev.  
participant (Intercept) 0.816722 0.90373  
Residual           0.003796 0.06161  
Number of obs: 40, groups: participant, 20  
  
Fixed effects:  
            Estimate Std. Error t value  
(Intercept) 0.19052   0.20255  0.941  
condition2  0.19935   0.01948 10.231  
  
Correlation of Fixed Effects:  
          (Intr)  
condition2 -0.048
```

# Plan for today

- Linear mixed effects model
  - what are they?
  - when to use them?
  - understanding the `model summary()`
- **Let's simulate some `lmer()`s**
  - effect of outliers
  - different slopes
  - Simpson's paradox
- A worked example
  - pooling:
    - complete pooling
    - no pooling
    - partial pooling
  - shrinkage

**Let's stimulate some 1mer ()s**

# Let's simulate an `lmer()`

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

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

⋮

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

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

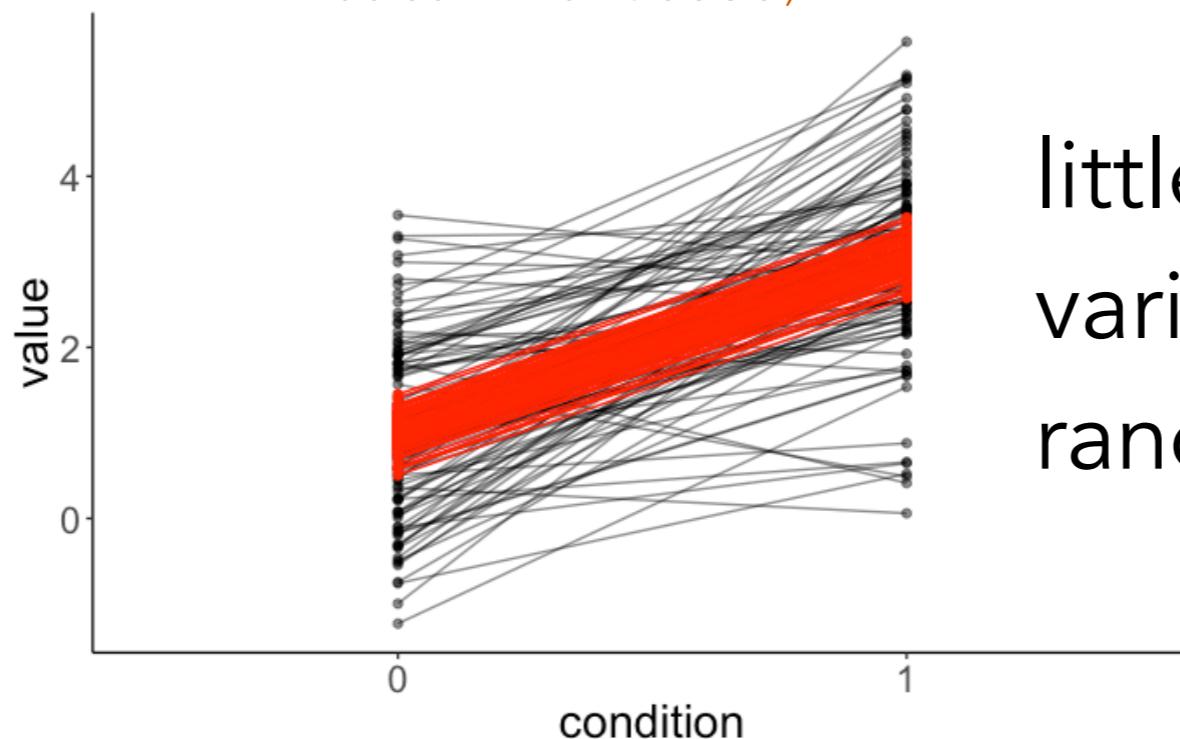
$$U_i \sim \mathcal{N}(\text{mean} = 0, \text{sd} = s_U)$$

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

# Let's simulate an `lmer()`

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

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



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

REML criterion at convergence: 606

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

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

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

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

little interindividual  
variation explained by  
random intercepts ...

# Model comparison

# Hypothesis test

Is taking into account individual differences worth it?

```
1 # fit models (without and with dependence)
2 fit.compact = lm(formula = value ~ 1 + condition,
3                   data = df.test)
4
5 fit.augmented = lmer(formula = value ~ 1 + condition + (1 | participant),
6                       data = df.test)
7
8 # compare models
9 # note: the lmer model has to be supplied first
10 anova(fit.augmented, fit.compact)
```

refitting model(s) with ML (instead of REML)

Data: df.test

Models:

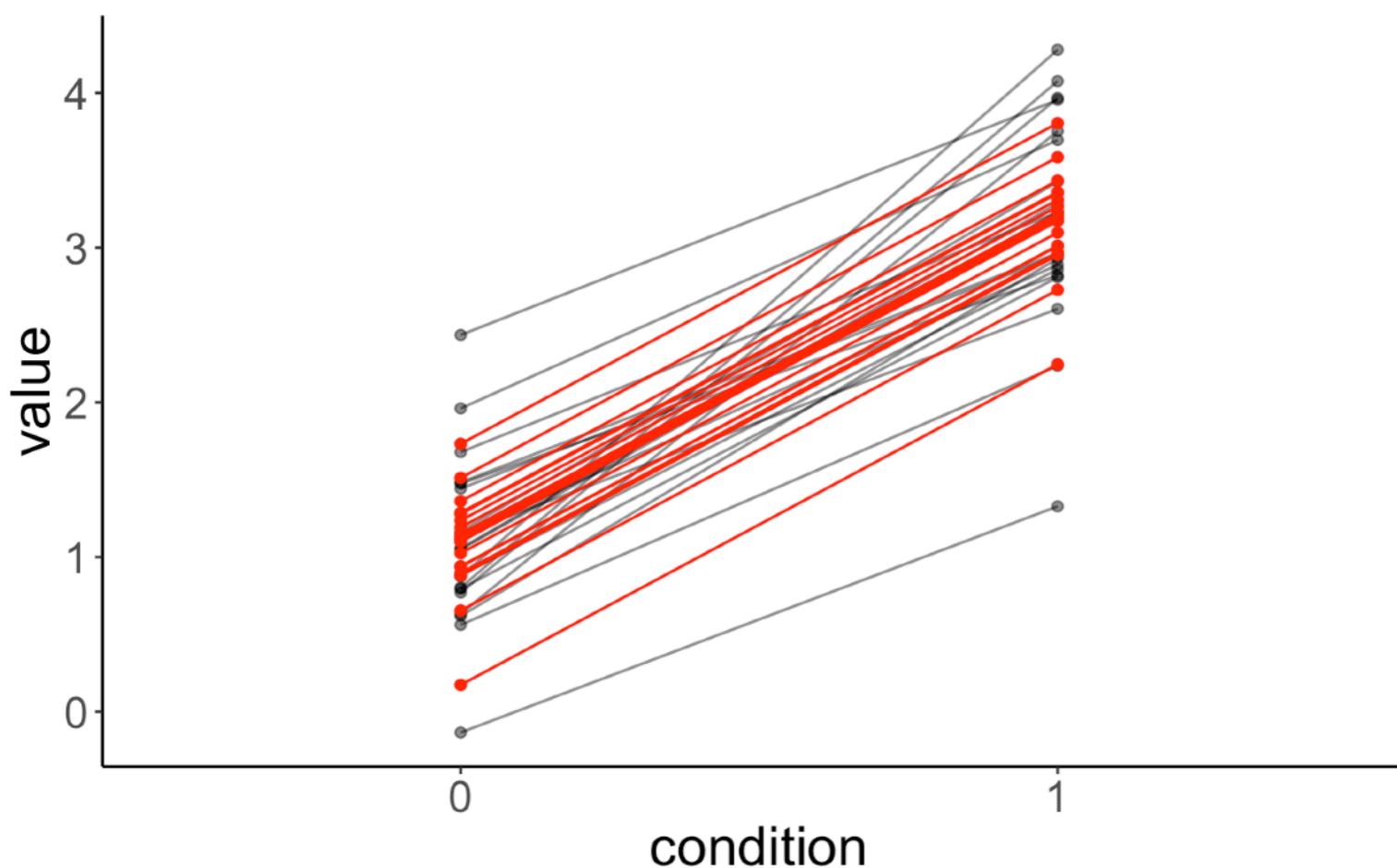
fit.lm: value ~ 1 + condition

fit.test: value ~ 1 + condition + (1 | participant)

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
fit.lm	3	608.6	618.49	-301.3	602.6			
fit.test	4	608.8	621.99	-300.4	600.8	1.7999	1	0.1797

not worth it

# Outliers



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

REML criterion at convergence: 74.9

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

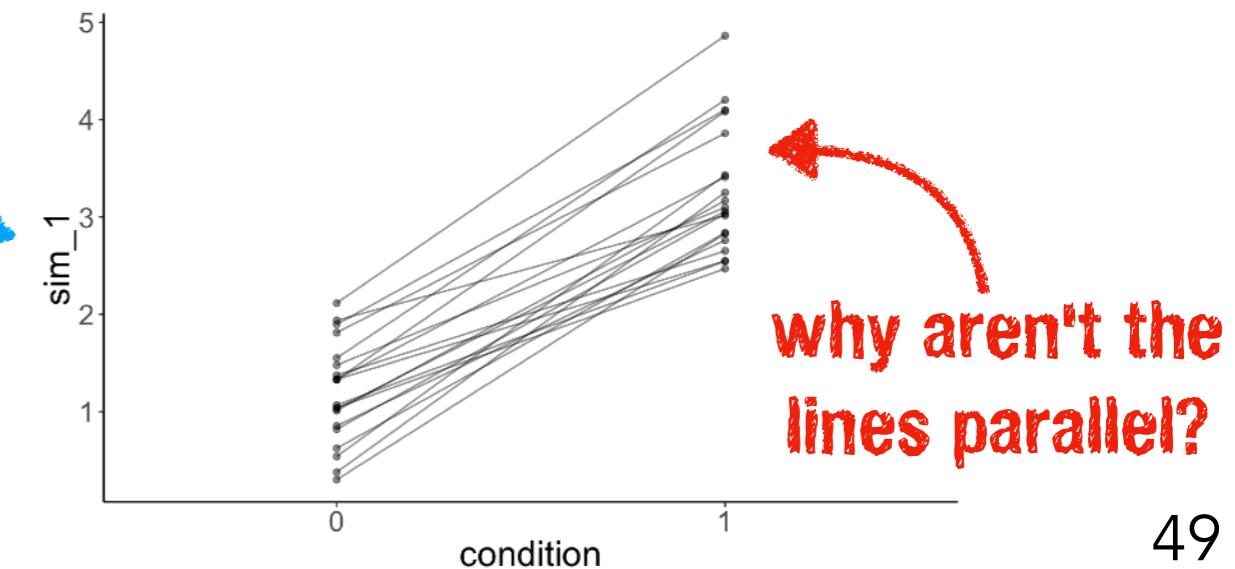
Random effects:
 Groups   Name        Variance Std.Dev. 
 participant (Intercept) 0.1702   0.4125  
 Residual           0.2270   0.4764  
Number of obs: 40, groups: participant, 20

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

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

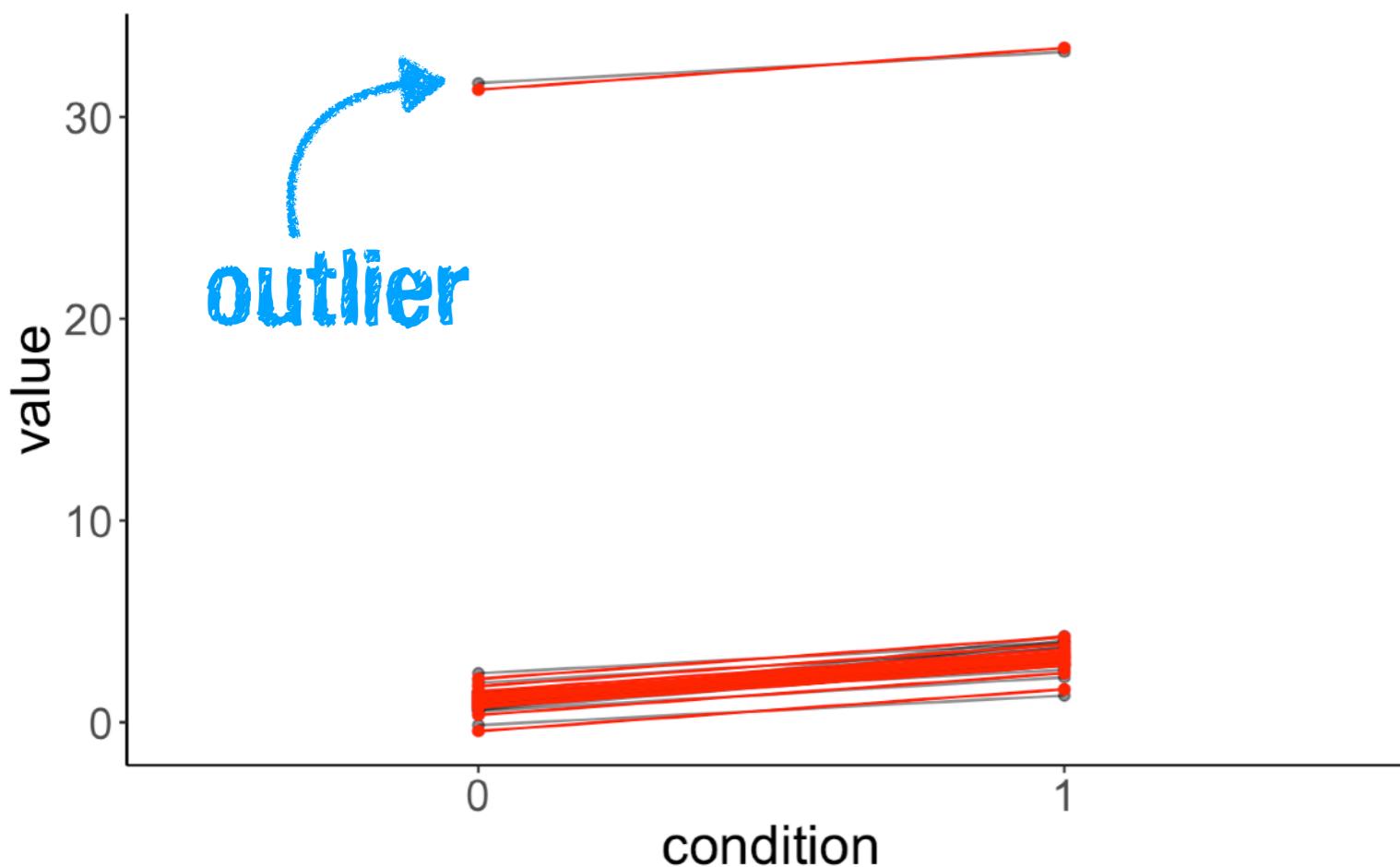
```
1 # fit model
2 fit.test = lmer(formula = value ~ 1 + condition + (1 | participant),
3                  data = df.test)
4
5 # simulate data
6 fit.test %>%
7   simulate()
```

simulated data



why aren't the lines parallel?

# Outliers



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

simulated data

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

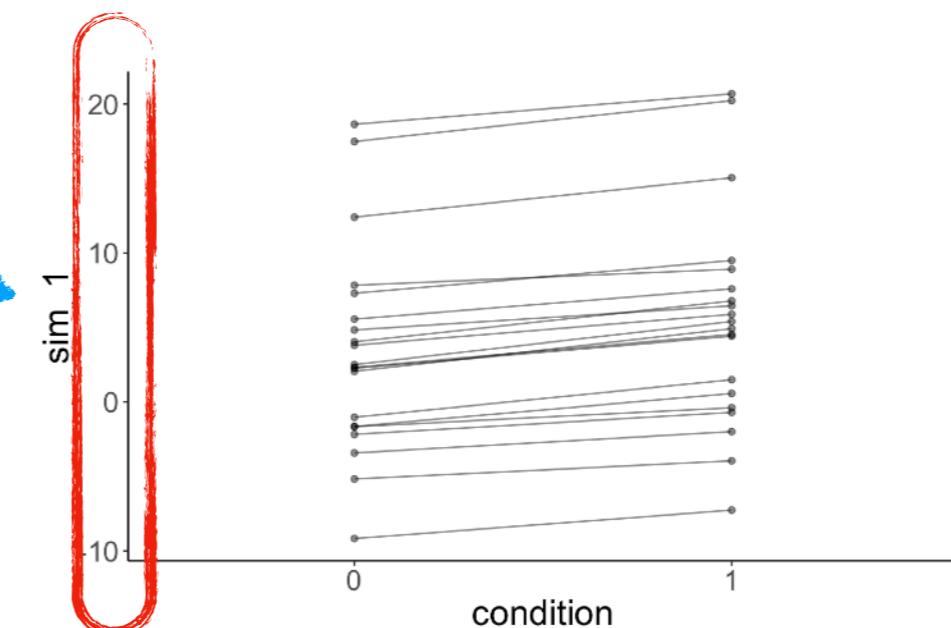
REML criterion at convergence: 171.7

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

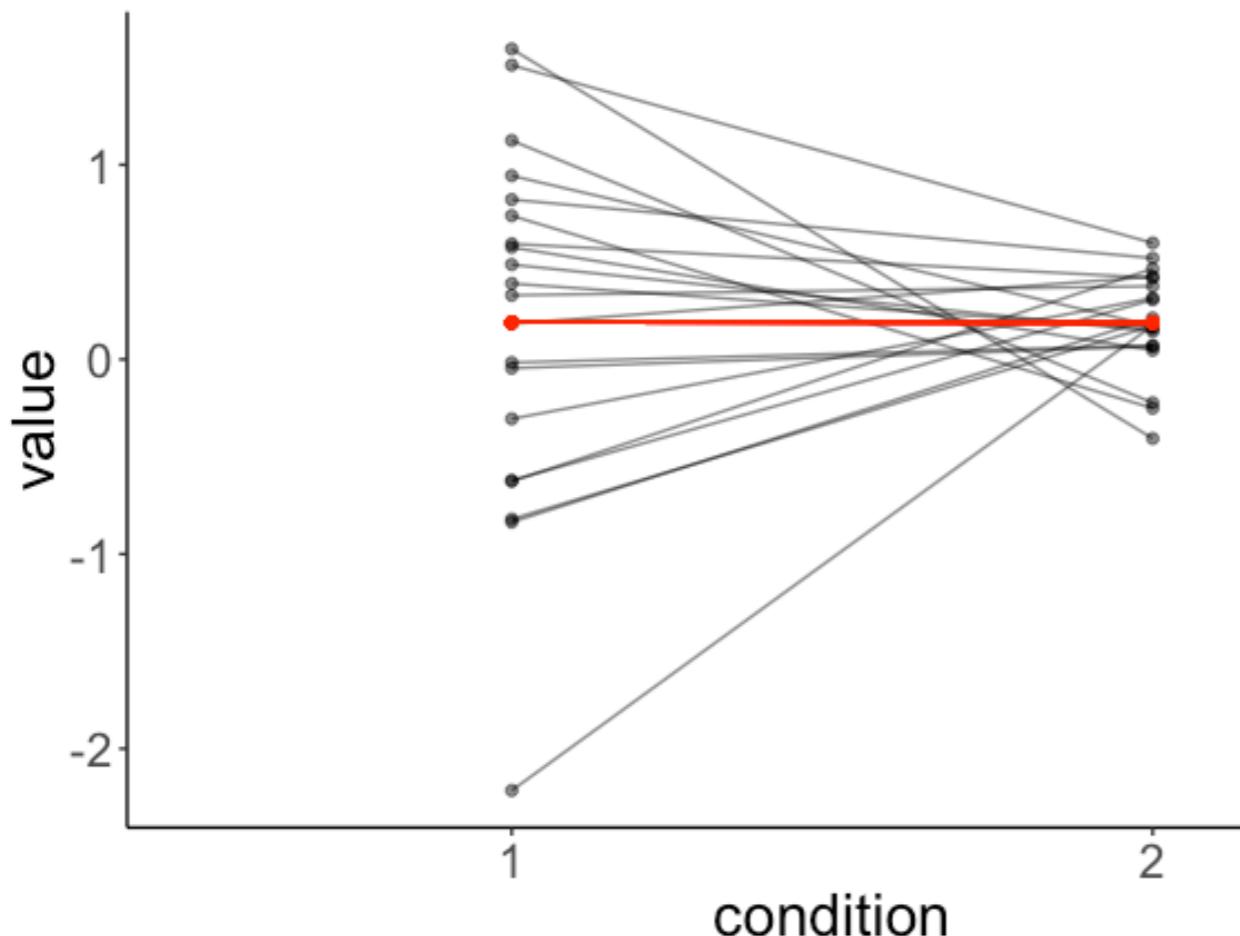
Random effects:
 Groups   Name        Variance Std.Dev. 
 participant (Intercept) 46.198   6.7969 
 Residual           0.227   0.4764 
Number of obs: 40, groups: participant, 20

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

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



# Different slopes



singular fit

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

REML criterion at convergence: 83.6

Scaled residuals:

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

Random effects:

Groups	Name	Variance	Std.Dev.
participant	(Intercept)	0.0000	0.0000
	Residual	0.4512	0.6717

Number of obs: 40, groups: participant, 20

Fixed effects:

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

Correlation of Fixed Effects:

(Intr)

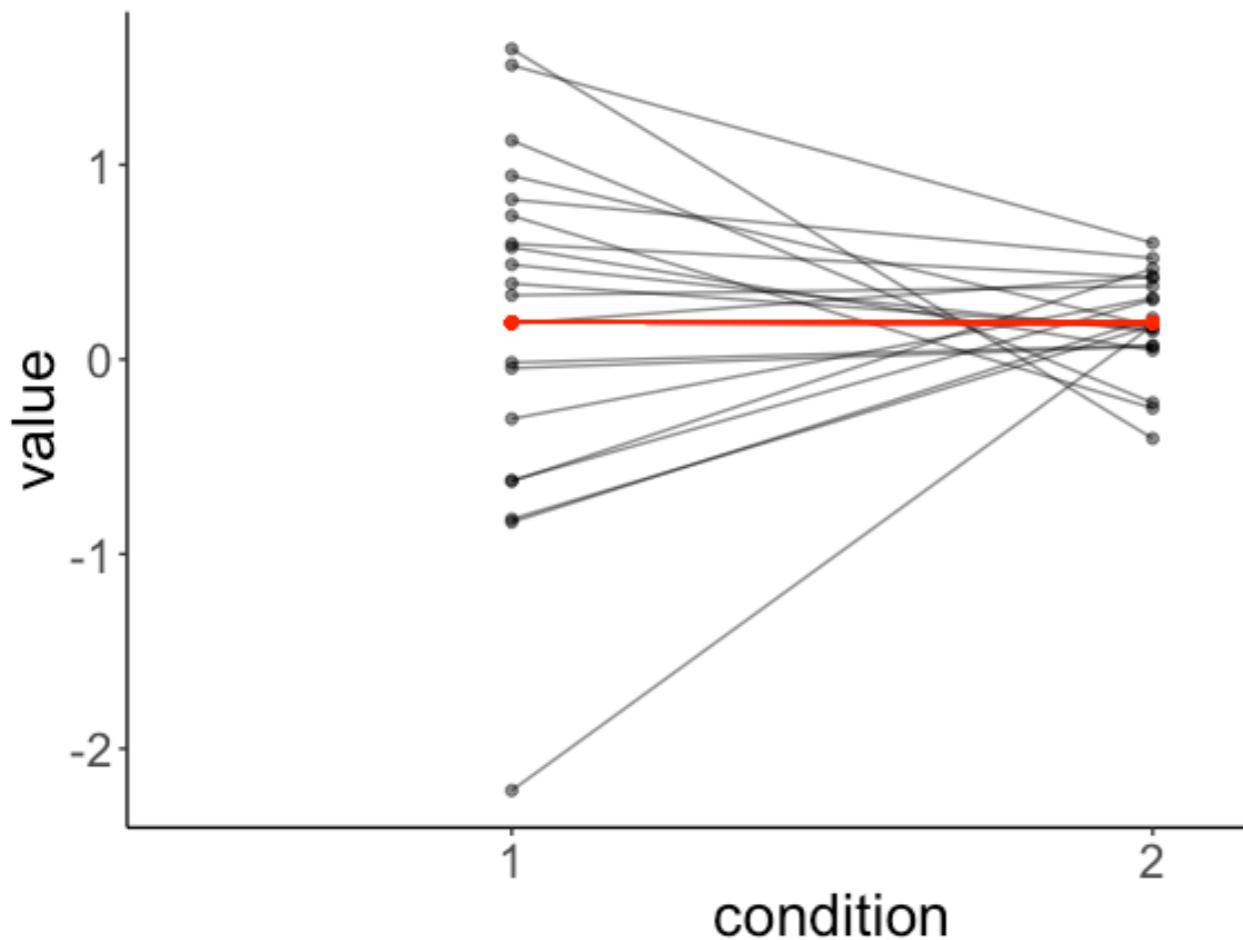
condition2 -0.707

convergence code: 0

singular fit

clearly there are interindividual differences though!?

# Different slopes



more parameters  
than data points

the "model" would  
just reproduce the  
data

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

random intercept

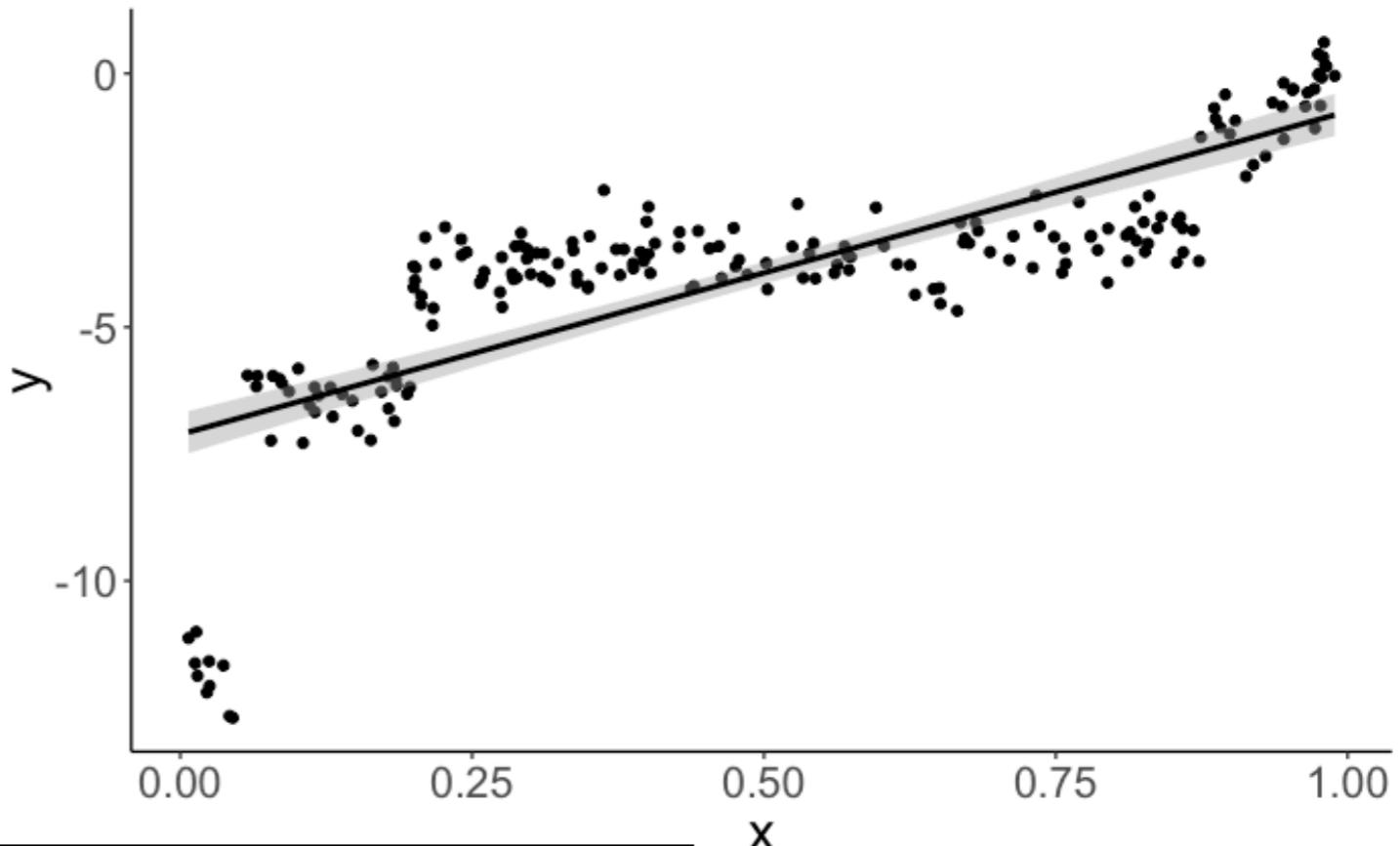


random slope

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
```

# Simpson's paradox



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

```
Call:  
lm(formula = y ~ x, data = df.simpson)  
  
Residuals:  
    Min      1Q  Median      3Q     Max  
-5.8731 -0.6362  0.2272  1.0051  2.6410  
  
Coefficients:  
            Estimate Std. Error t value Pr(>|t|)  
(Intercept) -7.1151    0.2107 -33.76 <2e-16 ***  
x             6.3671    0.3631  17.54 <2e-16 ***  
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 1.55 on 198 degrees of freedom  
Multiple R-squared:  0.6083, Adjusted R-squared:  0.6064  
F-statistic: 307.5 on 1 and 198 DF,  p-value: < 2.2e-16
```

positive relationship  
between x and y

# Simpson's paradox

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

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

```
REML criterion at convergence: 345.1
```

```
Scaled residuals:
```

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

```
Random effects:
```

Groups	Name	Variance	Std.Dev.
participant	(Intercept)	21.4898	4.6357
Residual		0.1661	0.4075

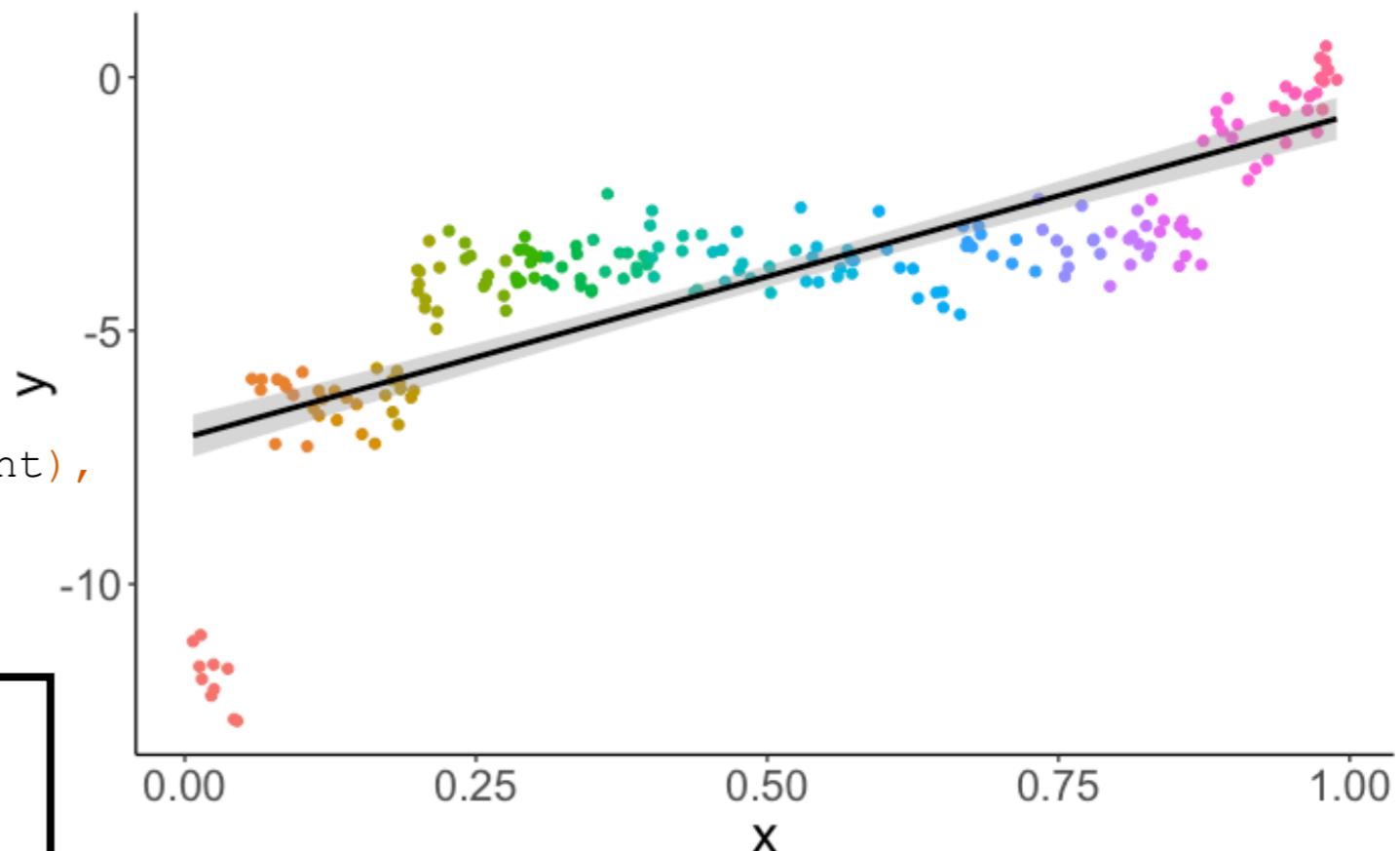
```
Number of obs: 200, groups: participant, 20
```

```
Fixed effects:
```

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

```
Correlation of Fixed Effects:
```

(Intr)	x
-0.621	



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

# Simpson's paradox

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

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

```
REML criterion at convergence: 345.1
```

```
Scaled residuals:
```

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

```
Random effects:
```

Groups	Name	Variance	Std.Dev.
participant	(Intercept)	21.4898	4.6357
Residual		0.1661	0.4075

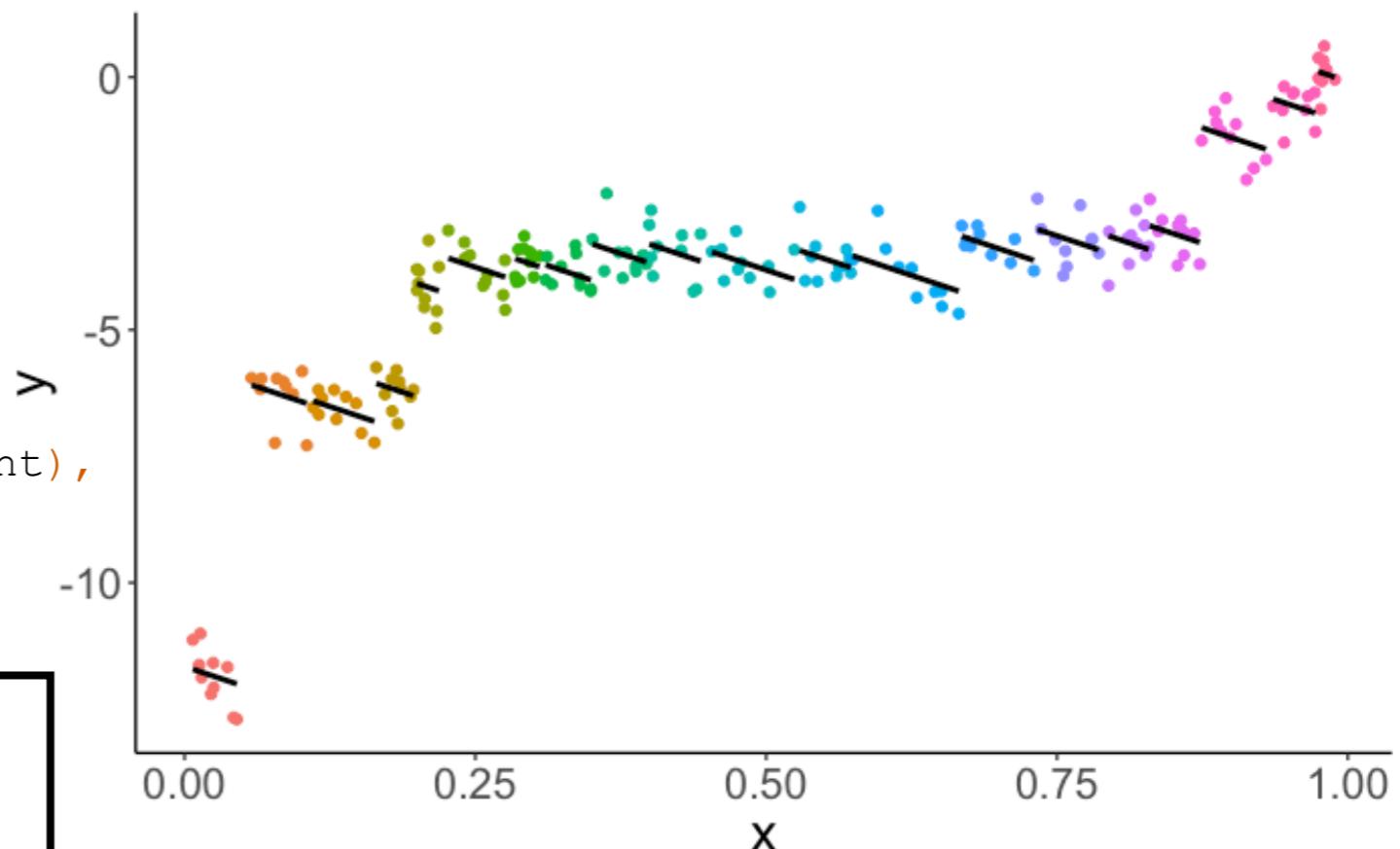
```
Number of obs: 200, groups: participant, 20
```

```
Fixed effects:
```

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

```
Correlation of Fixed Effects:
```

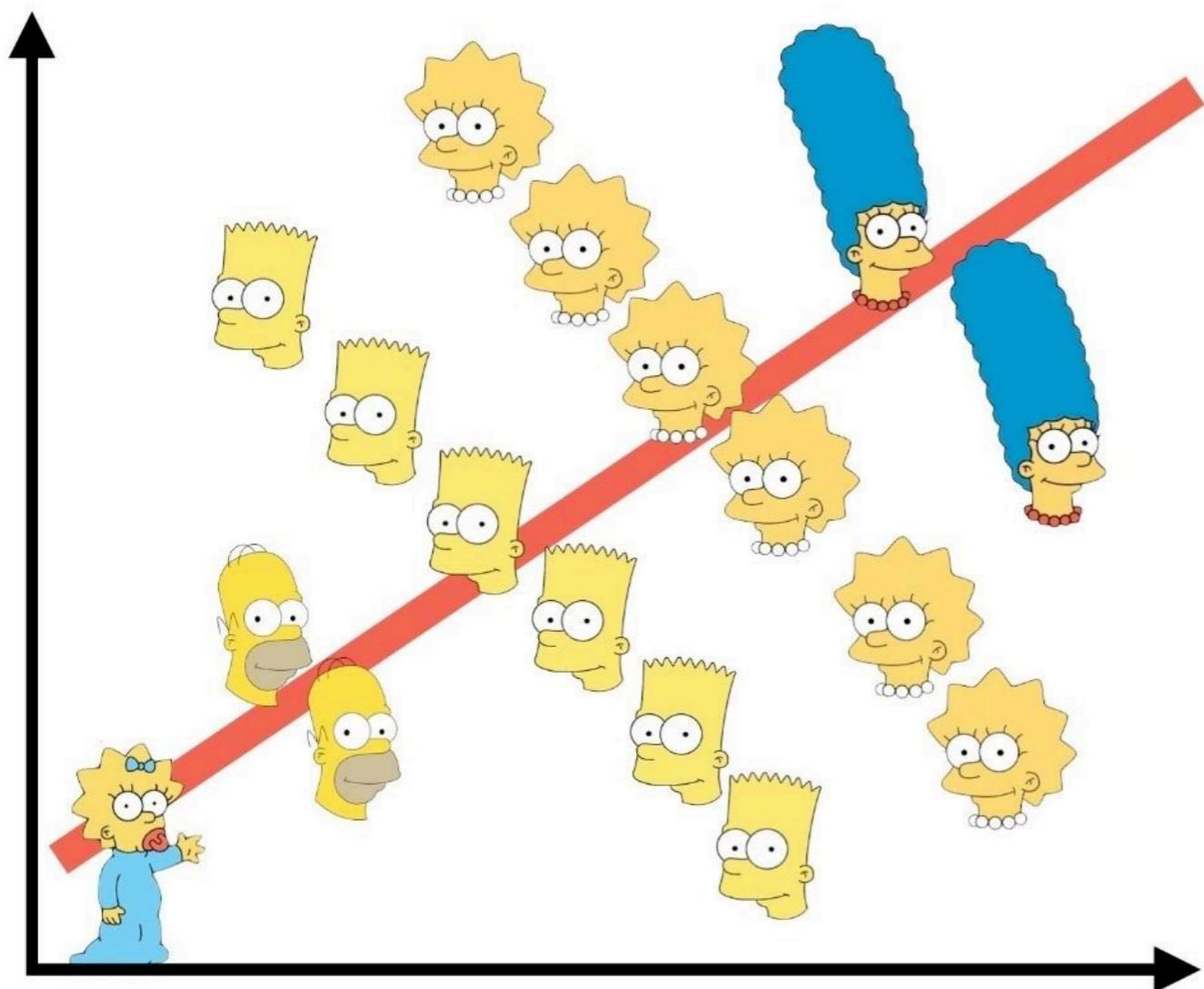
(Intr)	x
-0.621	



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

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

# Simpson's paradox



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

# Simpson's paradox

# UC Berkeley gender bias (1973)

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

overall men are more likely to be admitted

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

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

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

women applied to more competitive departments

# general points about `lmer()`

- Why don't we just run individual regressions?
  - overfitting ...
  - inflating type 1 error
  - larger uncertainty in parameter estimates because only few data points are used
- Why don't we just run a regression on the means?
  - we throw away a lot of information
- Mixed effects models:
  - make use of all available information

# Plan for today

- Linear mixed effects model
  - what are they?
  - when to use them?
  - understanding the `model summary()`
- Let's simulate some `lmer()`s
  - effect of outliers
  - different slopes
  - Simpson's paradox
- **A worked example**
  - pooling:
    - complete pooling
    - no pooling
    - partial pooling
  - shrinkage

# A worked example

# general points about `lmer()`

- Why don't we just run individual regressions?
  - overfitting ...
  - inflating type 1 error
  - larger uncertainty in parameter estimates because only few data points are used for each model
- Why don't we just run a regression on the means?
  - we throw away a lot of information
- Mixed effects models:
  - make use of all available information

let's take a look  
at an example

**Tristan Mahr**

Language and data scientist

 [Madison, WI](#) [Email](#) [Twitter](#) [GitHub](#) [Stackoverflow](#) [R Bloggers](#)

## Plotting partial pooling in mixed-effects models

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

### The sleepstudy dataset

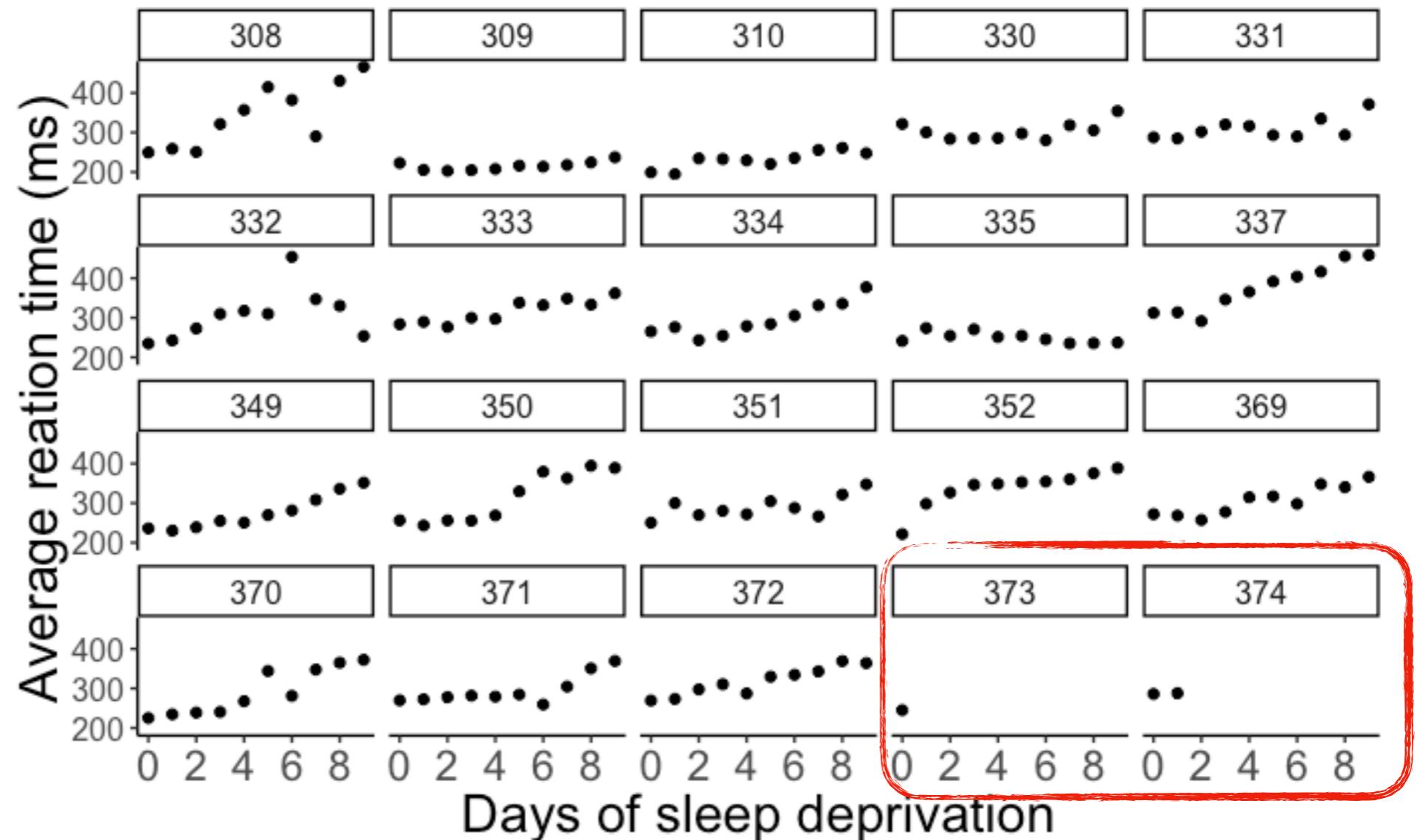
For these examples, I’m going to use the `sleepstudy` dataset from the `lme4` package. The outcome measure is reaction time, the predictor measure is days of sleep deprivation, and these measurements are nested within participants—we have 10 observations per participant. I am also going to add two fake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

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

# Data set

## How does sleep deprivation affect reaction time?

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



20 participants

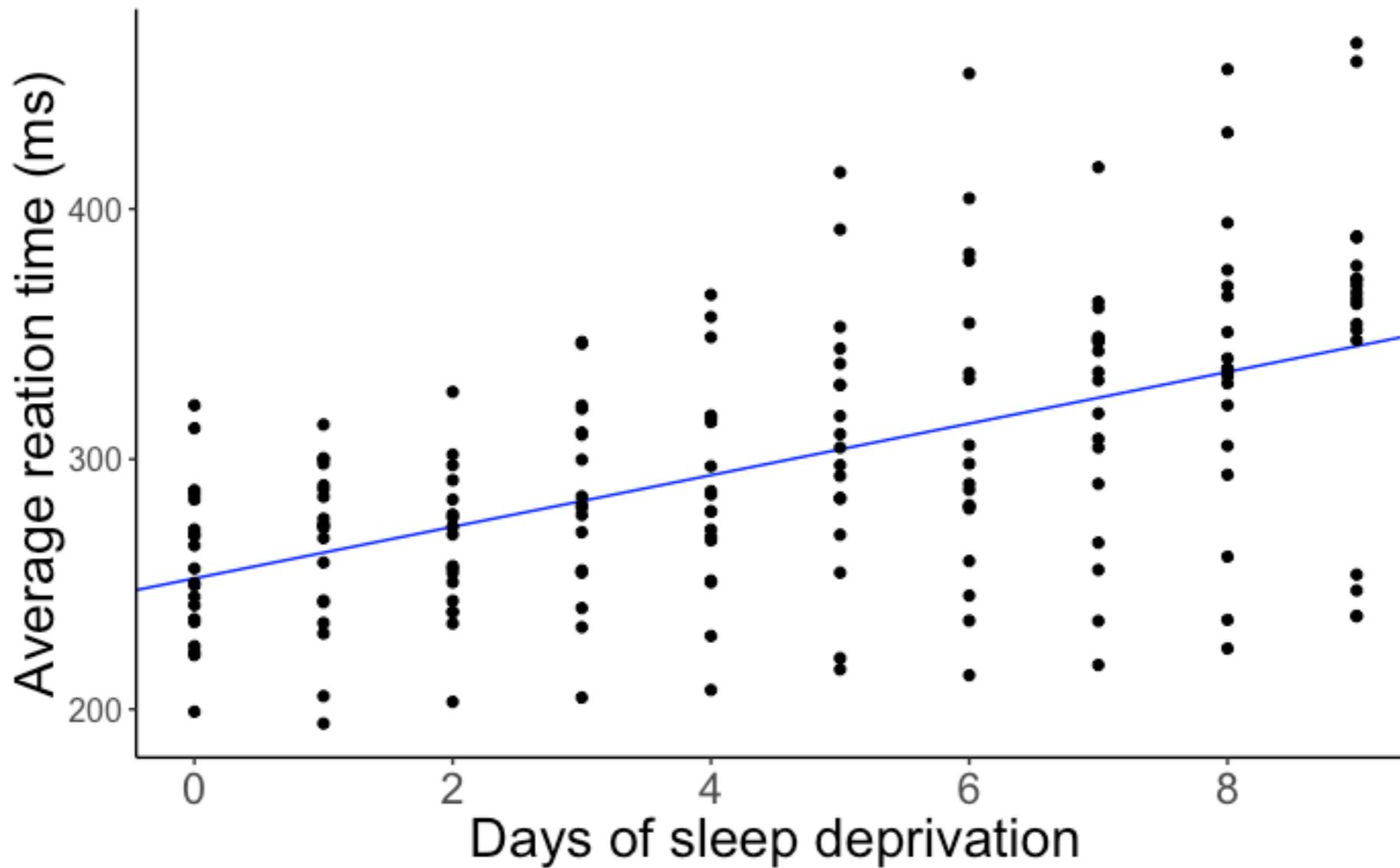
2 with incomplete information

# Pooling information

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

# Complete pooling: Fit one global regression

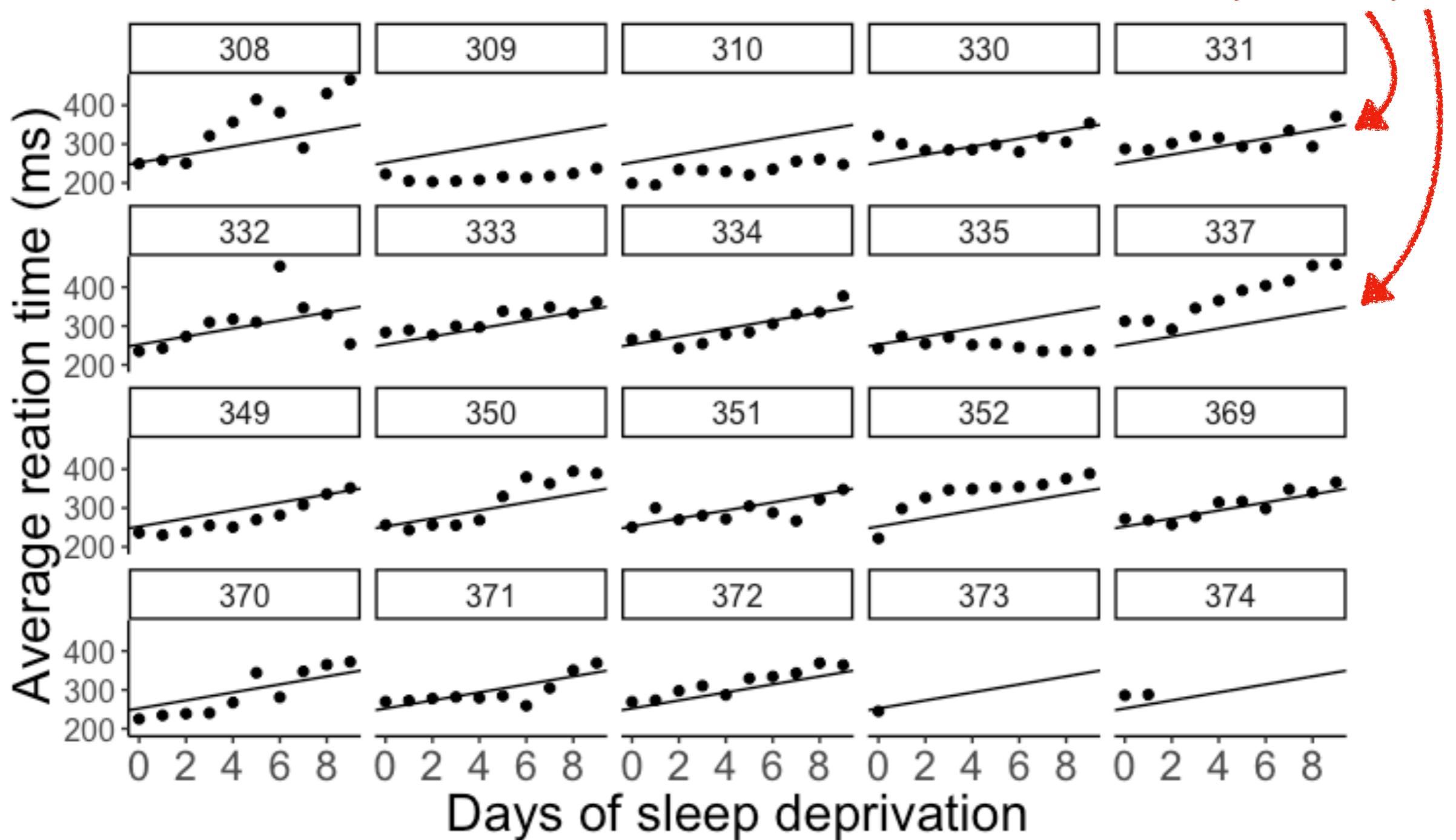
```
lm(formula = reaction ~ days,  
  data = df.sleep)
```



# Complete pooling: Fit one global regression

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

same line for  
each participant



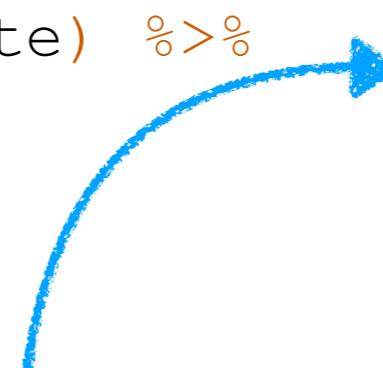
# No pooling: Fit separate regressions

```
1 df.no_pooling = df.sleep %>%
2   group_by(subject) %>%
3   nest(data = c(days, reaction)) %>%
4   mutate(fit = map(data, ~ lm(reaction ~ days, data = .)),
5         params = map(fit, tidy)) %>%
```

	subject	data	regression fit	extracted parameters
1	308	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(2...	list(coefficients = c(`(Intercept)` = 244.1926690909...	list(term = c("(Intercept)", "days"), estimate = c(244.1...
2	309	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(2...	list(coefficients = c(`(Intercept)` = 205.0549454545...	list(term = c("(Intercept)", "days"), estimate = c(205.0...
3	310	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(1...	list(coefficients = c(`(Intercept)` = 203.4842254545...	list(term = c("(Intercept)", "days"), estimate = c(203.4...
4	330	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(3...	list(coefficients = c(`(Intercept)` = 289.6850927272...	list(term = c("(Intercept)", "days"), estimate = c(289.6...
5	331	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(2...	list(coefficients = c(`(Intercept)` = 285.7389654545...	list(term = c("(Intercept)", "days"), estimate = c(285.7...
6	332	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(2...	list(coefficients = c(`(Intercept)` = 264.2516145454...	list(term = c("(Intercept)", "days"), estimate = c(264.2...
7	333	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(2...	list(coefficients = c(`(Intercept)` = 275.0191054545...	list(term = c("(Intercept)", "days"), estimate = c(275.0...
8	334	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(2...	list(coefficients = c(`(Intercept)` = 240.1629145454...	list(term = c("(Intercept)", "days"), estimate = c(240.1...
9	335	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(2...	list(coefficients = c(`(Intercept)` = 263.0346927272...	list(term = c("(Intercept)", "days"), estimate = c(263.0...
10	337	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(3...	list(coefficients = c(`(Intercept)` = 290.1041272727...	list(term = c("(Intercept)", "days"), estimate = c(290.1...
11	349	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(2...	list(coefficients = c(`(Intercept)` = 215.1117727272...	list(term = c("(Intercept)", "days"), estimate = c(215.1...
12	350	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(2...	list(coefficients = c(`(Intercept)` = 225.8346036363...	list(term = c("(Intercept)", "days"), estimate = c(225.8...
13	351	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(2...	list(coefficients = c(`(Intercept)` = 261.1470109090...	list(term = c("(Intercept)", "days"), estimate = c(261.1...
14	352	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(2...	list(coefficients = c(`(Intercept)` = 276.3720690909...	list(term = c("(Intercept)", "days"), estimate = c(276.3...
15	369	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(2...	list(coefficients = c(`(Intercept)` = 254.9681490909...	list(term = c("(Intercept)", "days"), estimate = c(254.9...
16	370	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(2...	list(coefficients = c(`(Intercept)` = 210.4490909090...	list(term = c("(Intercept)", "days"), estimate = c(210.4...
17	371	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(2...	list(coefficients = c(`(Intercept)` = 253.6360381818...	list(term = c("(Intercept)", "days"), estimate = c(253.6...
18	372	list(days = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9), reaction = c(2...	list(coefficients = c(`(Intercept)` = 267.0448, days = ...	list(term = c("(Intercept)", "days"), estimate = c(267.0...
19	374	list(days = c(0, 1), reaction = c(286, 288))	list(coefficients = c(`(Intercept)` = 286, days = 2.000...	list(term = c("(Intercept)", "days"), estimate = c(286, 2...
20	373	list(days = 0, reaction = 245)	list(coefficients = c(`(Intercept)` = 245, days = NA), r...	list(term = "(Intercept)", estimate = 245, std.error = ...

# No pooling: Fit separate regressions

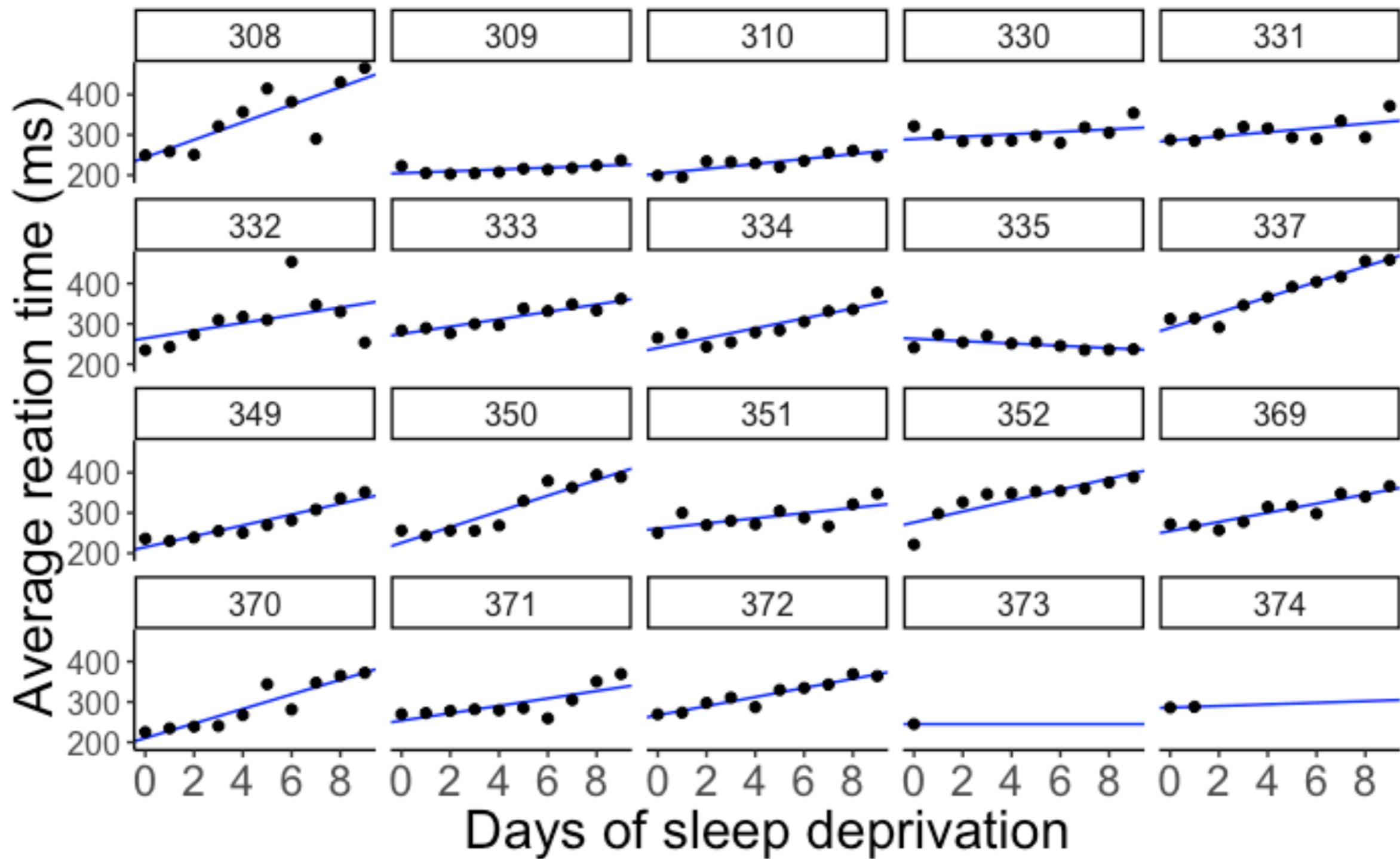
```
1 df.no_pooling = df.sleep %>%
2   group_by(subject) %>%
3   nest(data = c(days, reaction)) %>%
4   mutate(fit = map(data, ~ lm(reaction ~ days, data = .)),
5         params = map(fit, tidy)) %>%
6   ungroup() %>%
7   unnest(c(params)) %>%
8   select(subject, term, estimate) %>%
9   complete(subject,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



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

separate intercept and  
slope for each participant

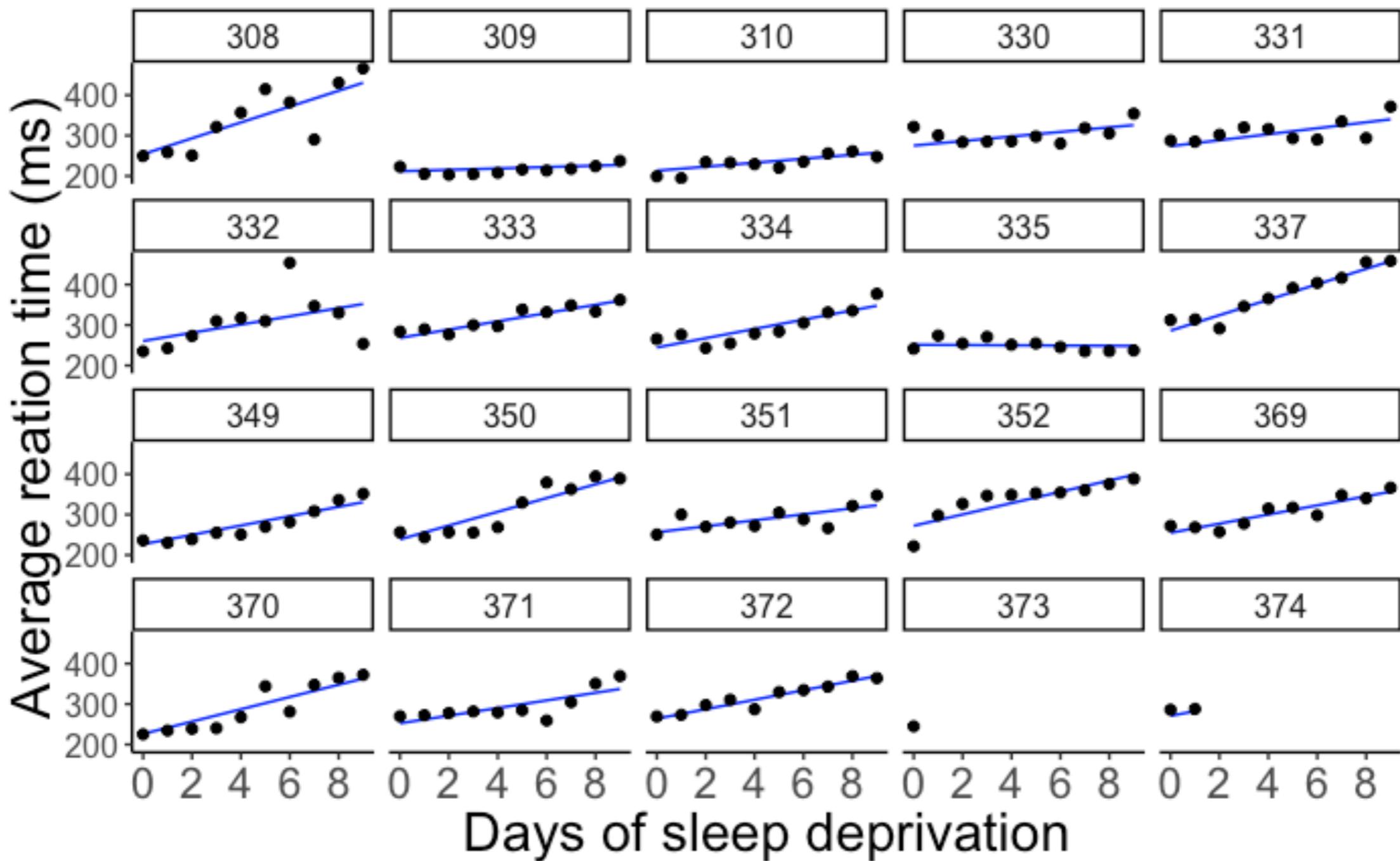
# No pooling: Fit separate regressions



# Partial pooling: Fit mixed effects model

intercepts and slopes differ  
between participants

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

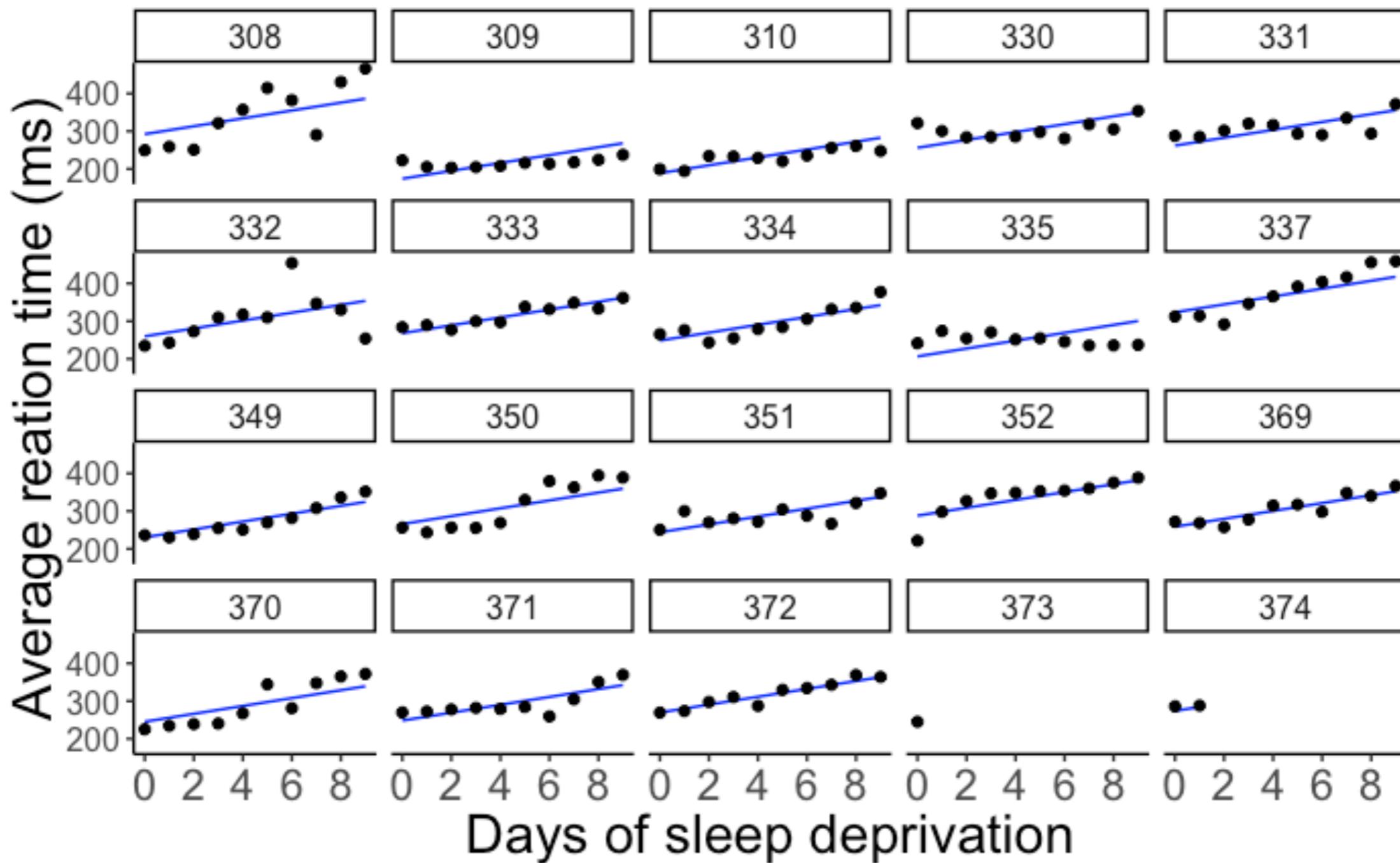


# Partial pooling: Fit mixed effects model

only intercepts differ  
between participants

random intercept

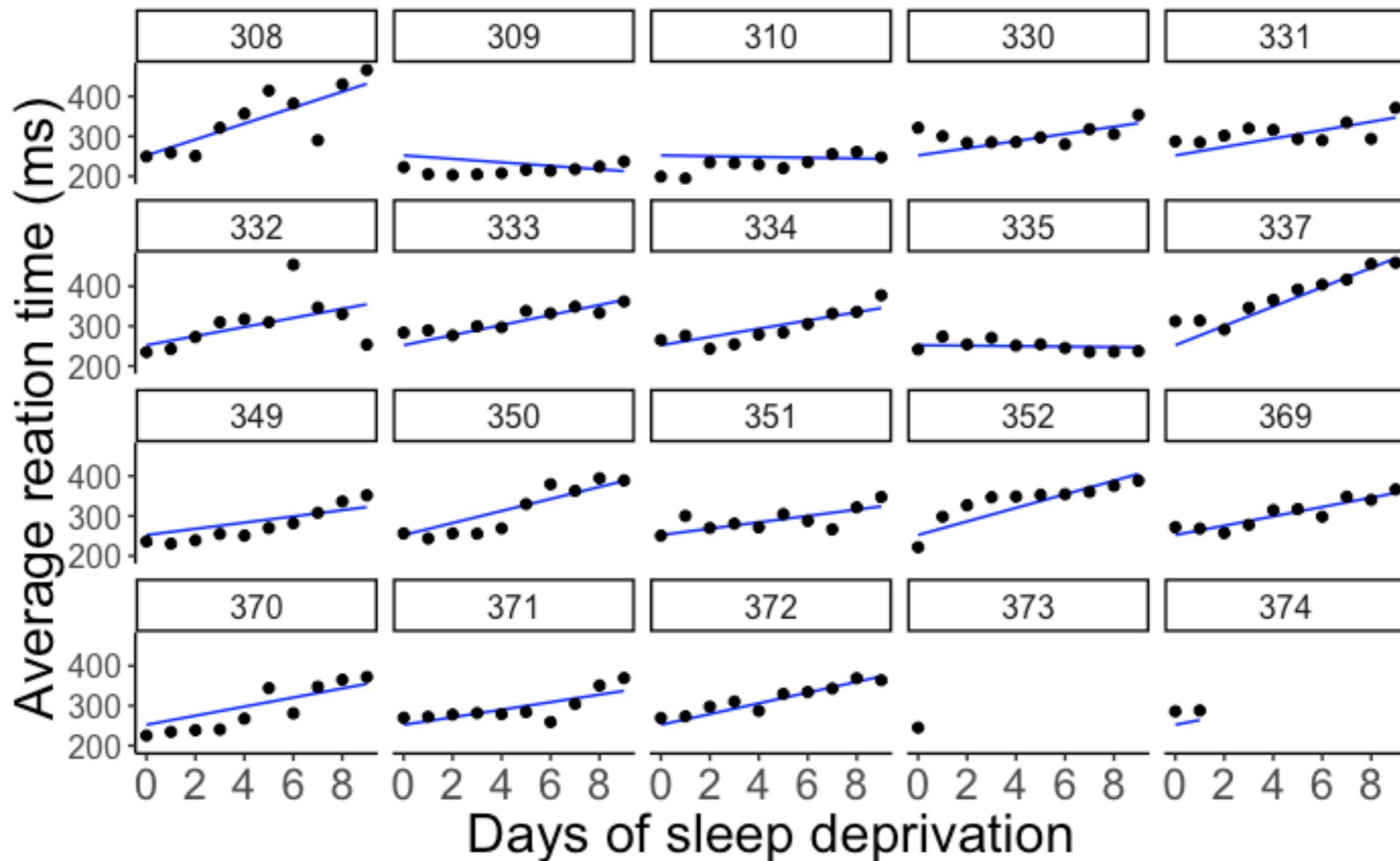
```
lmer (formula = reaction ~ 1 + days + (1 | subject),  
      data = df.sleep)
```



# Partial pooling: Fit mixed effects model

only slopes differ between participants

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



# Coefficients

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

`(1 | subject)`

random intercepts

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

`(0 + days | subject)`

random slopes

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

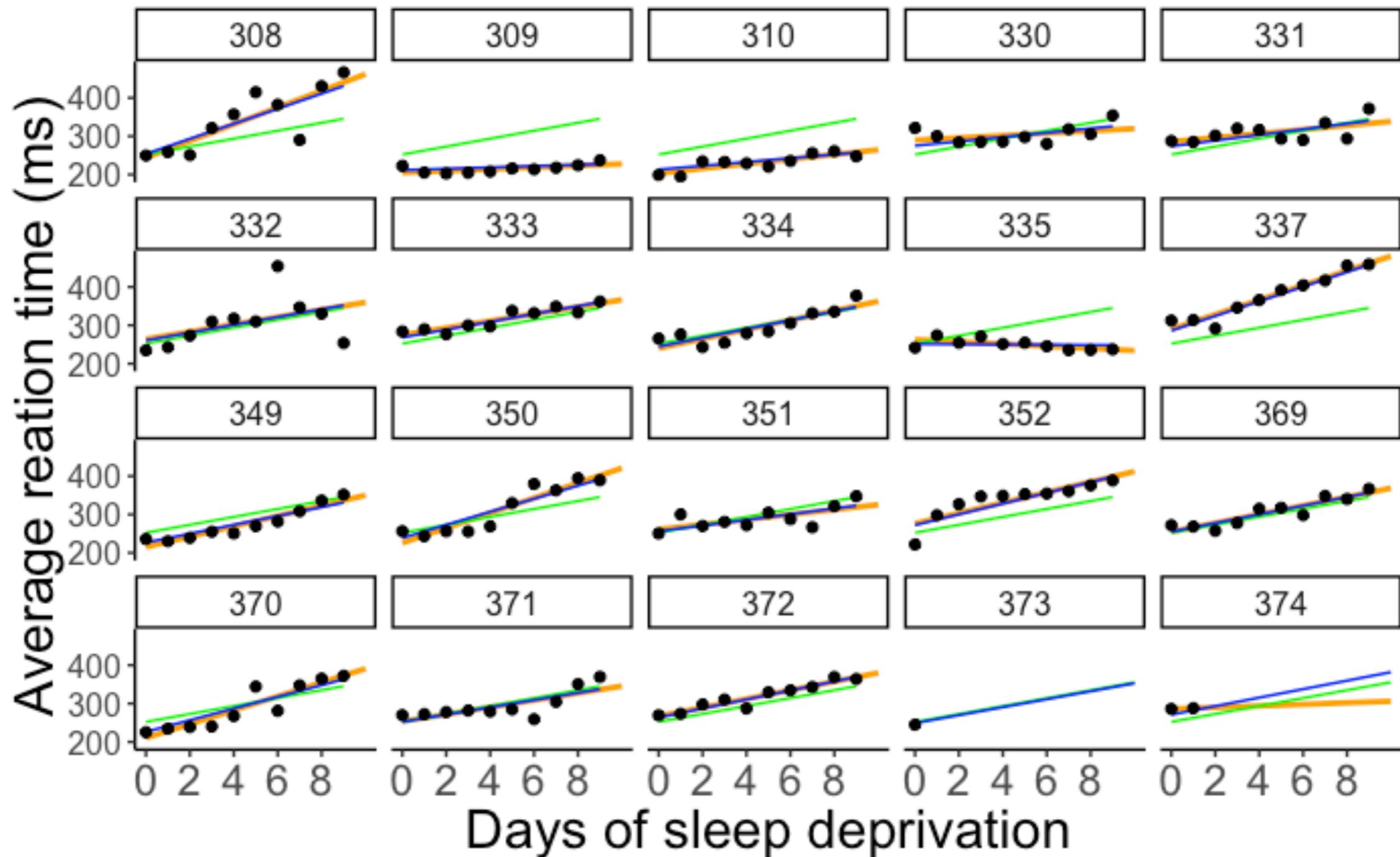
`... + (1 + days | subject)`

random intercepts  
and slopes

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

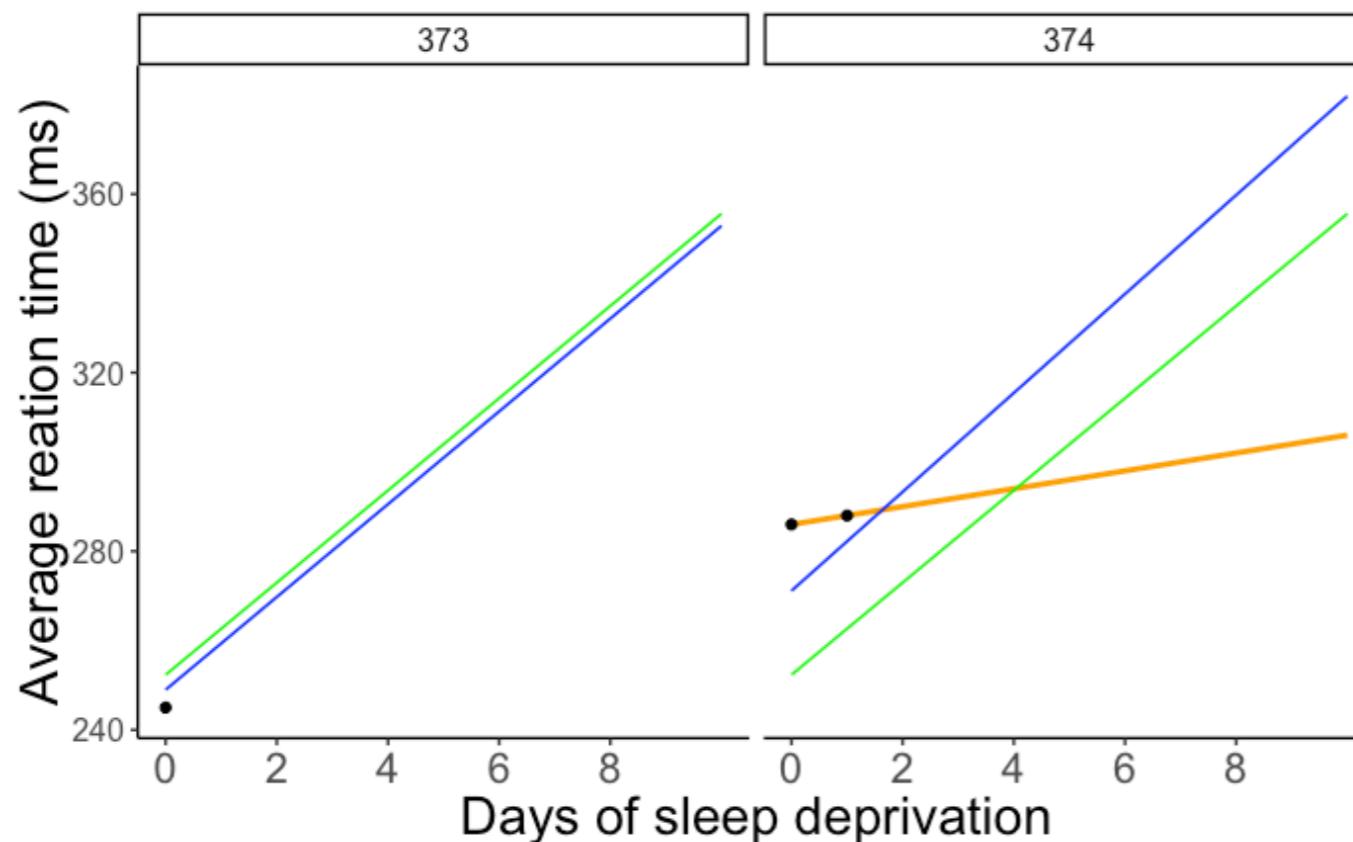
# Comparison

complete pooling  
no pooling  
partial pooling



# Comparison

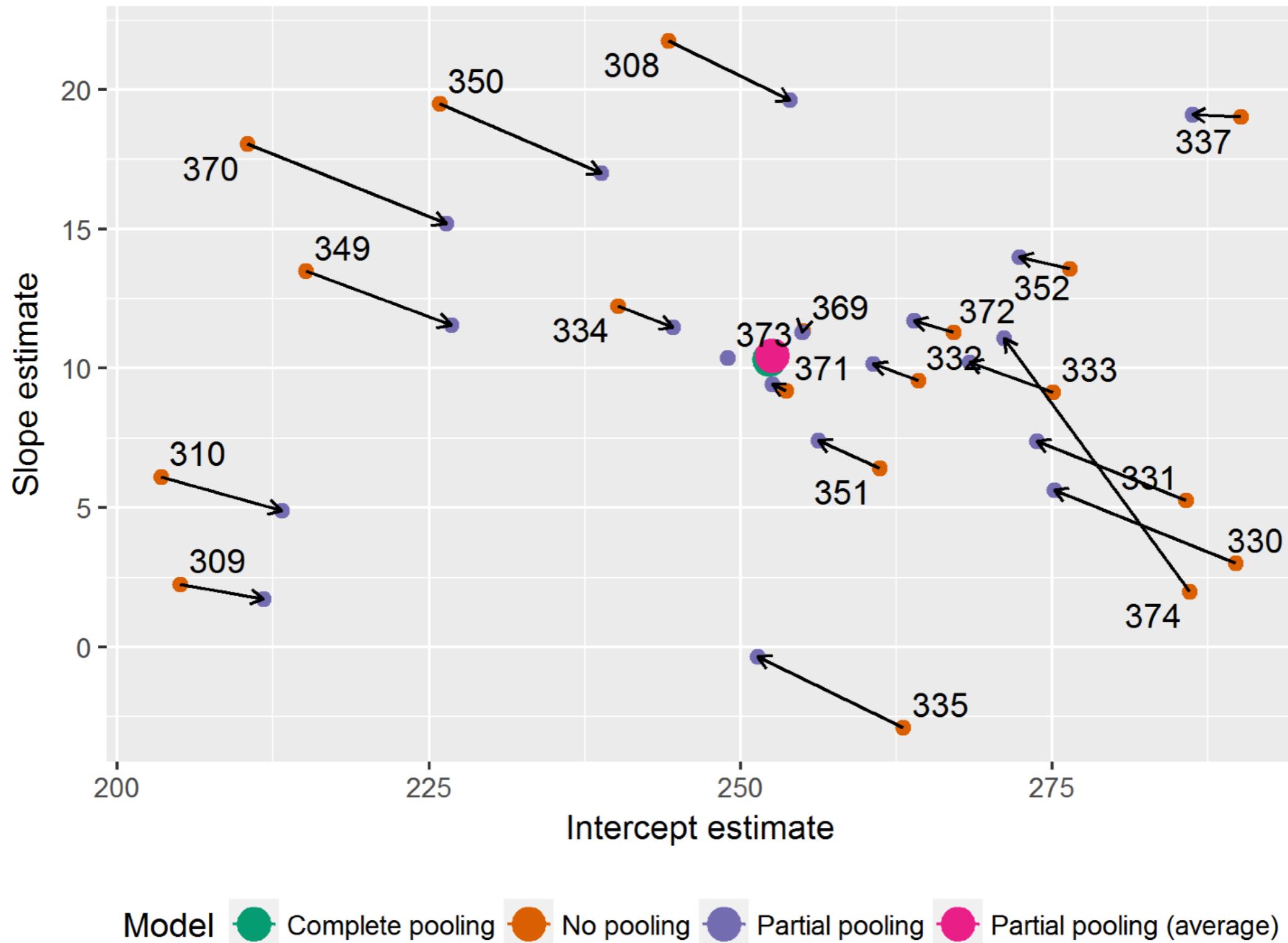
complete pooling  
no pooling  
partial pooling



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

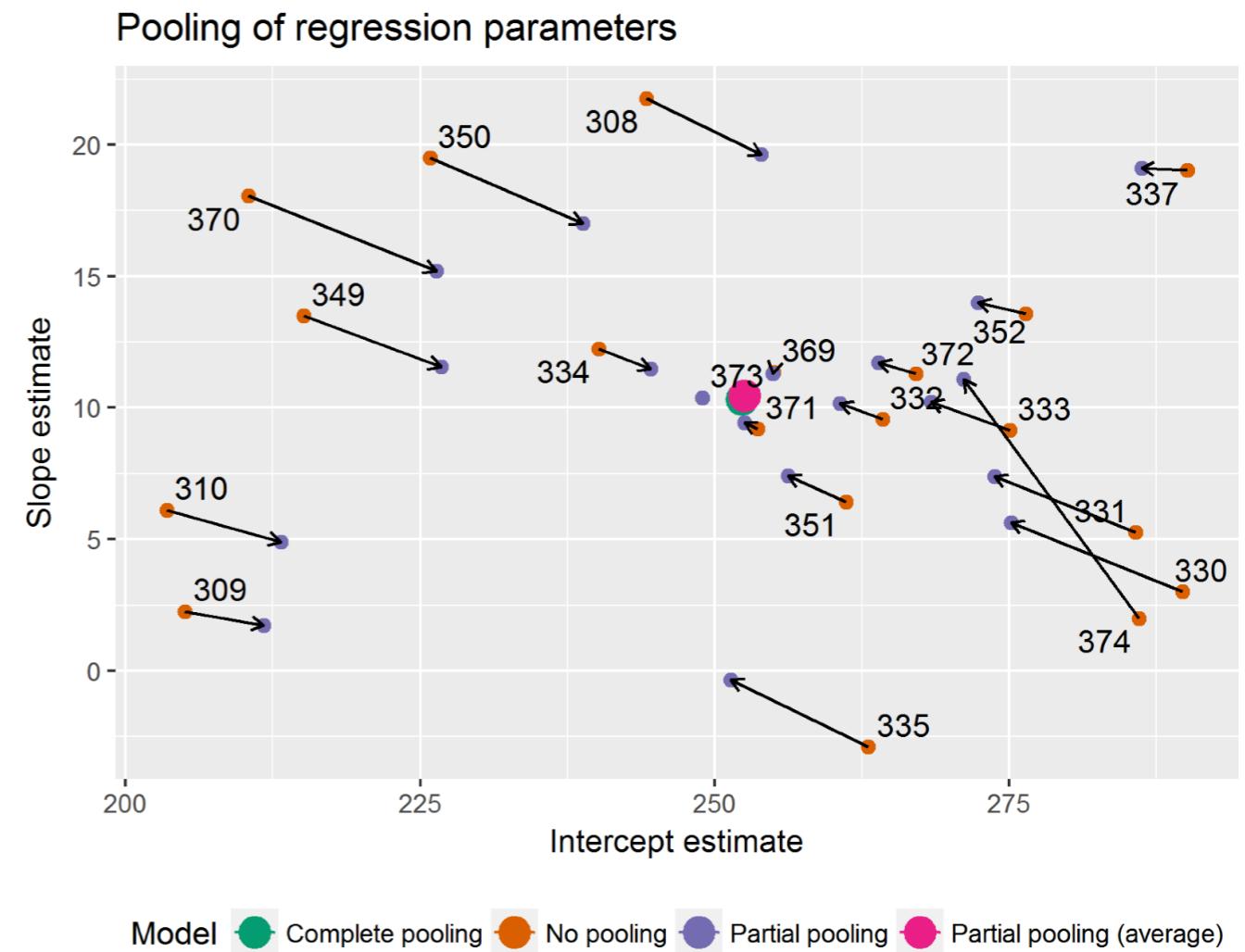
# Shrinkage

## Pooling of regression parameters



# Shrinkage

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



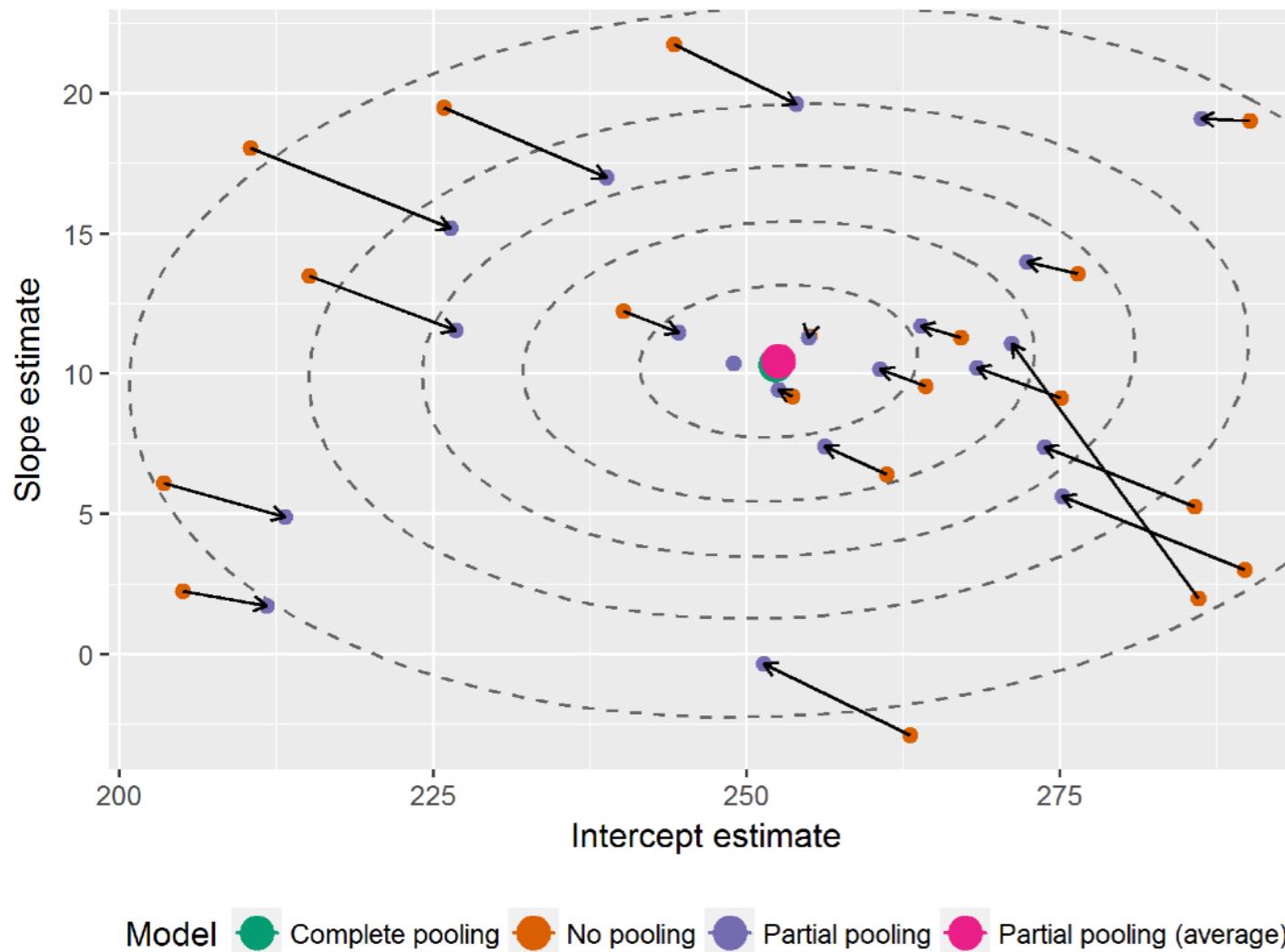
In [the lme4 book](#), Douglas Bates provides an alternative to *shrinkage*:

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

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

# Shrinkage

Topographic map of regression parameters

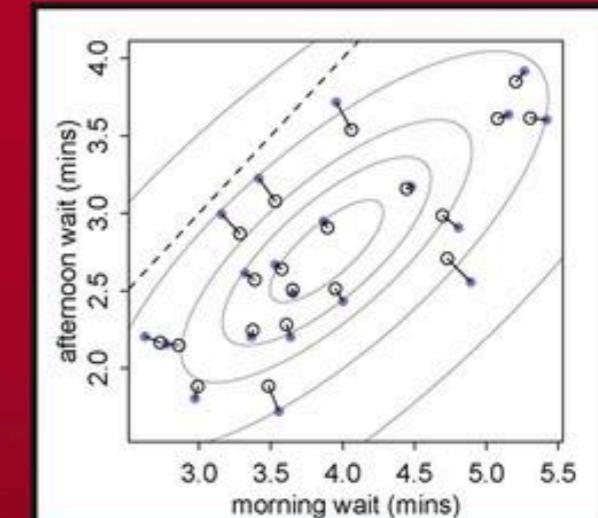


mixed effects model estimates a multi-variate Gaussian to account for (possible) correlations between intercepts and slopes

Texts in Statistical Science

## Statistical Rethinking

A Bayesian Course with Examples in R and Stan



Richard McElreath

CRC Press  
Taylor & Francis Group  
A CHAPMAN & HALL BOOK

# Summary

- Linear mixed effects model
  - understanding the model summary
  - simulating data for an **lmer()**
  - effect of outliers
  - different slopes
  - Simpson's paradox
- A worked example
  - pooling:
    - complete pooling
    - no pooling
    - partial pooling
  - shrinkage

# **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!**