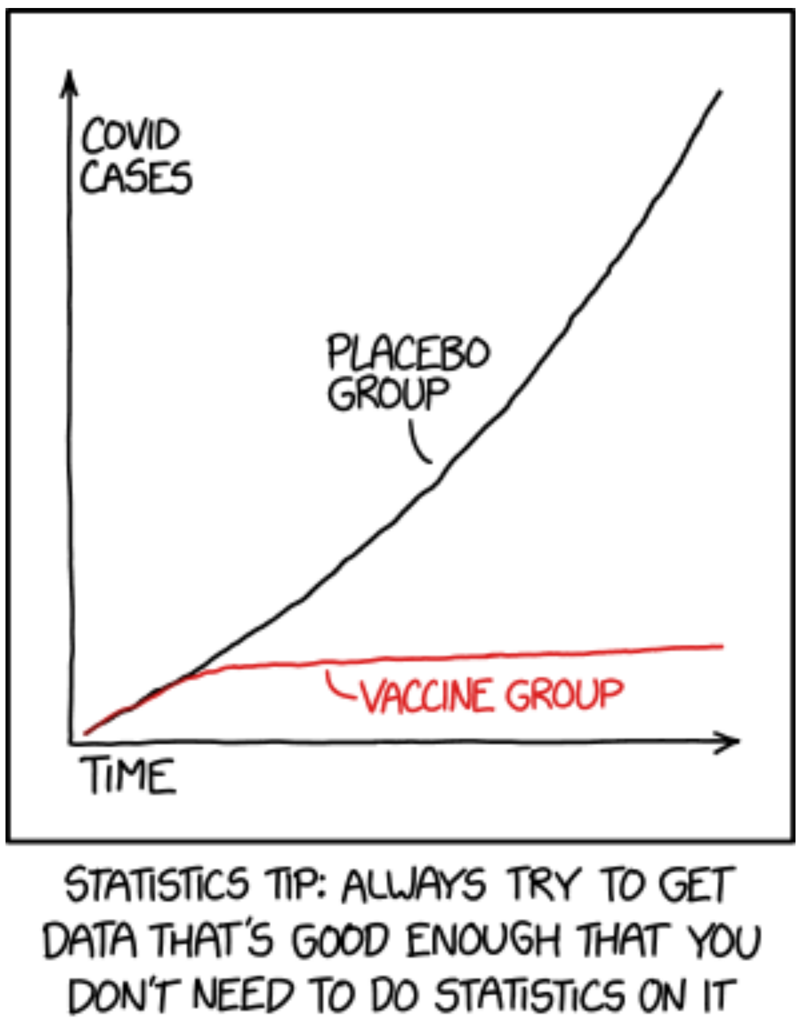
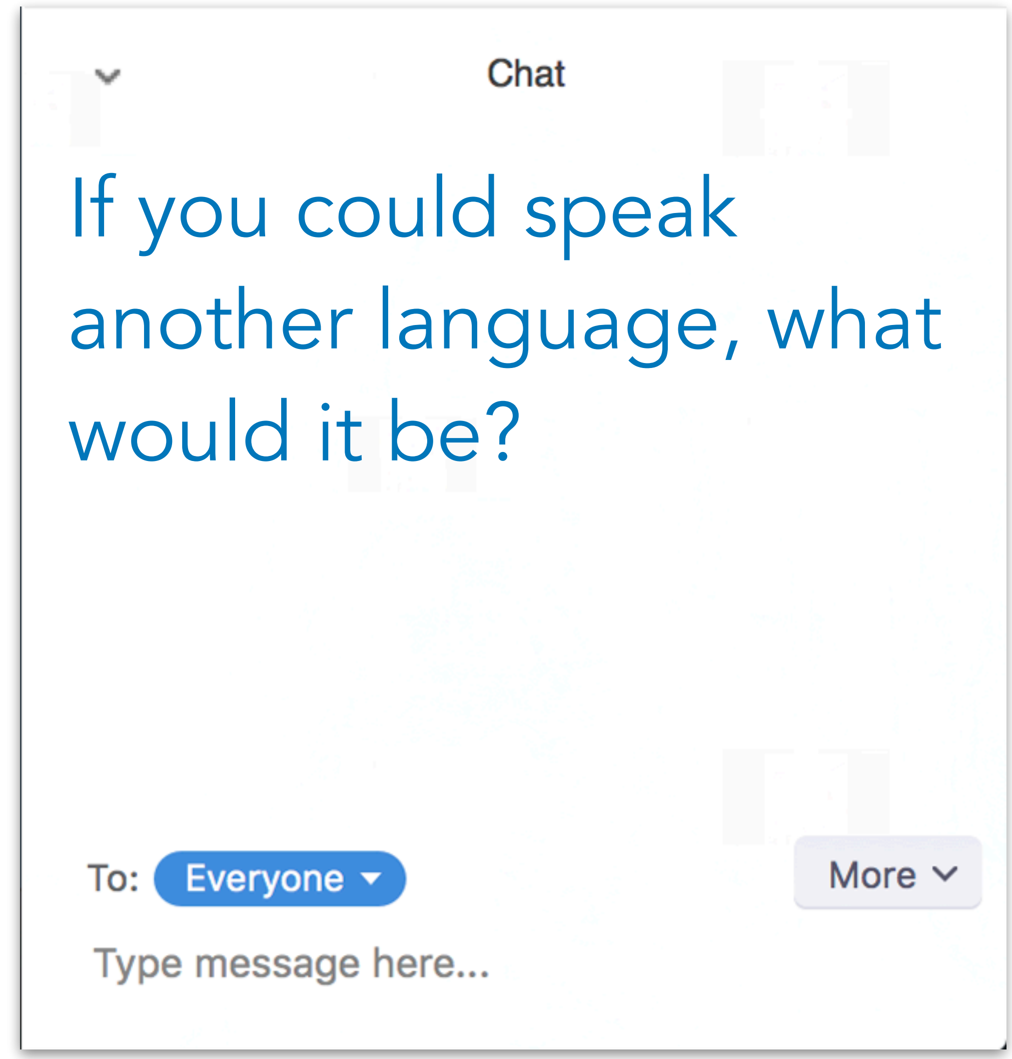
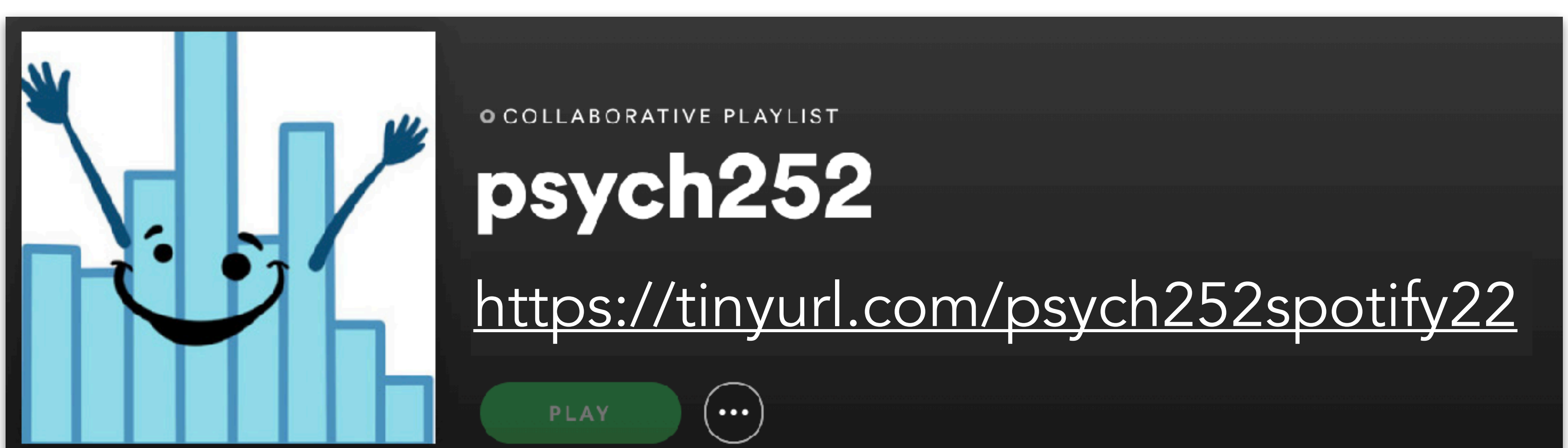
PSYCH 252: Statistical Methods for Behavioral and Social Sciences

Linear mixed effects models 4







We're listening to "Belo Horizonte" by "Cari Cari"

02/23/2022

Plan for today

- Quick Recap
- Linear mixed effects model
  - Some examples
  - `lmer()` standard operating procedures

Quick recap

Quick recap: `lmer()` summary

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

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

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

REML criterion at convergence: 17.3

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

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

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

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

Fixed effects

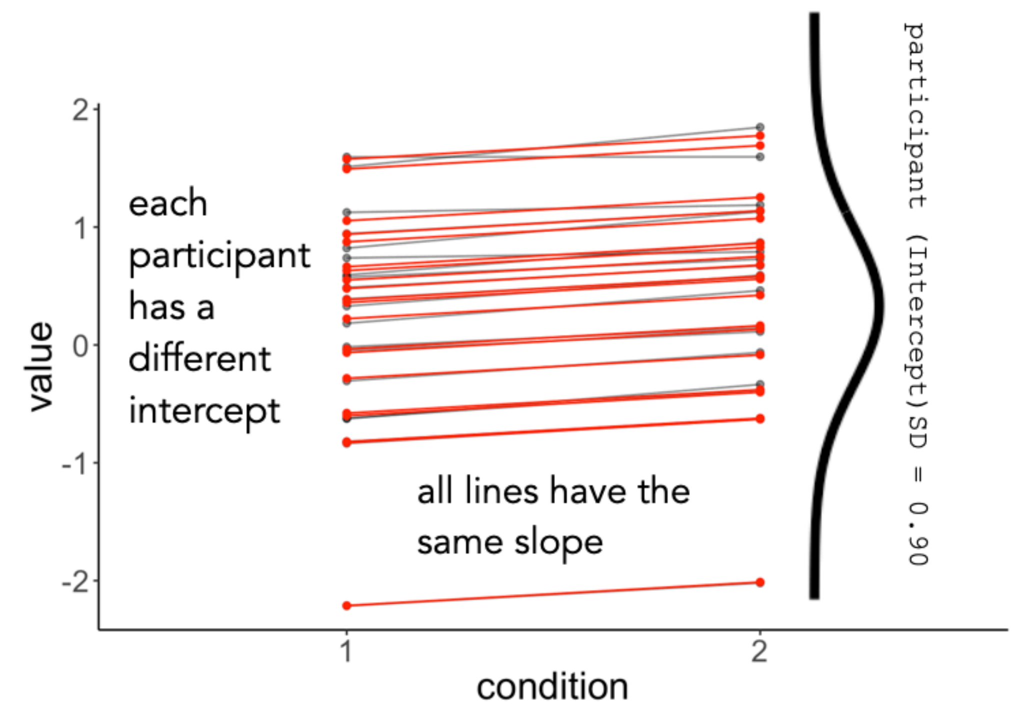
one parameter for the global intercept (value for the baseline condition)

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

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

18

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



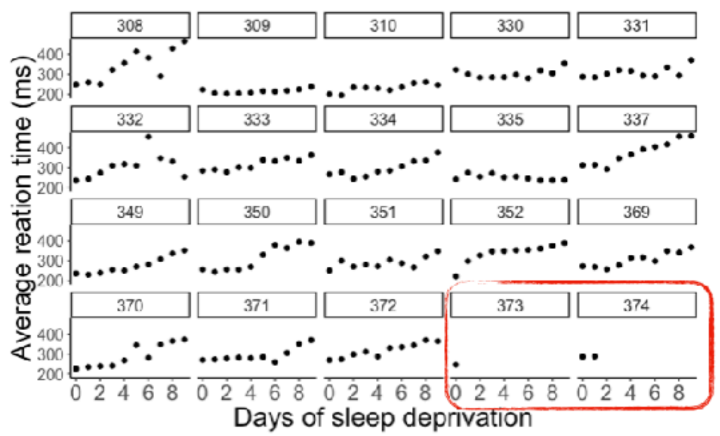
17

Quick recap: worked example

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

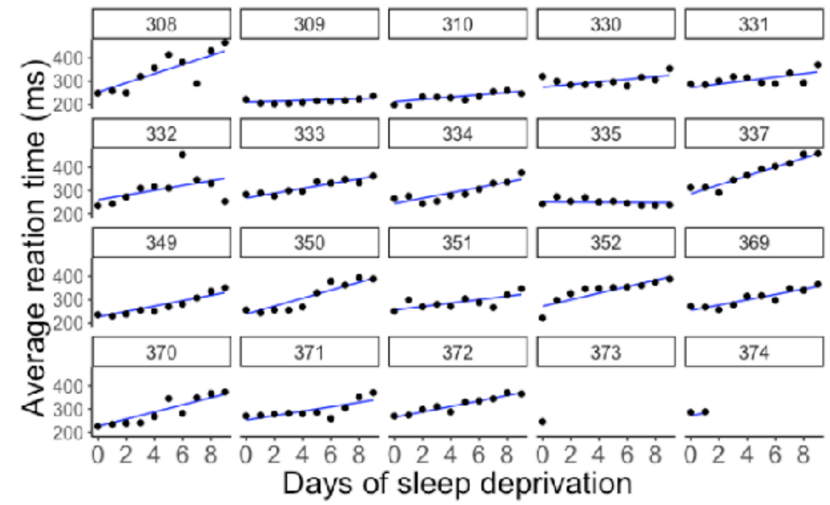
26

Partial pooling: Fit mixed effects model

intercepts and slopes differ between participants

random intercept random slope

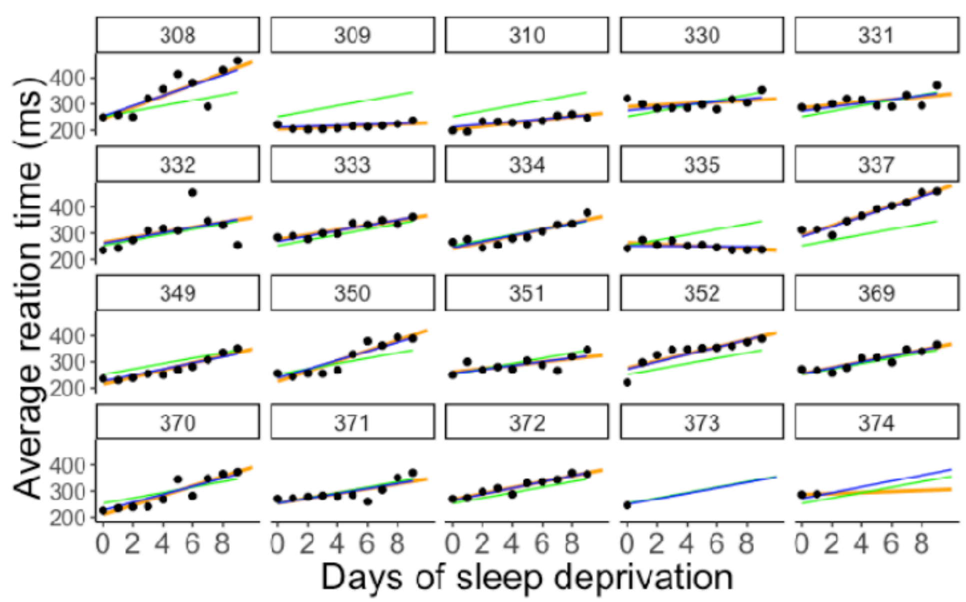
```
lmer(formula = reaction ~ 1 + days + (1 + days | subject),
     data = df.sleep)
```



33

Comparison

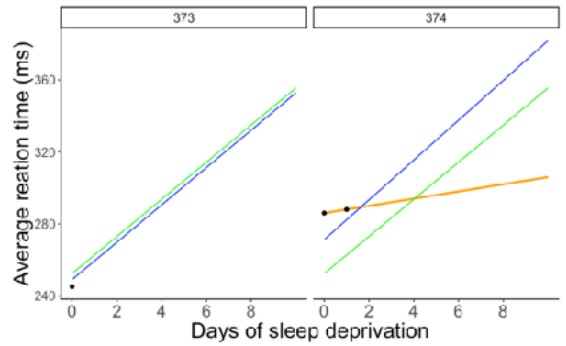
complete pooling
no pooling
partial pooling



37

Quick recap: shrinkage

Comparison



complete pooling
no pooling
partial pooling

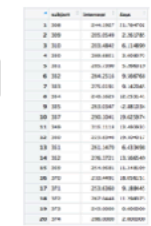
- **complete pooling**:
  - doesn't account for any individual variation
- **no pooling**:
  - doesn't yield predictions when we only have observation
  - doesn't consider the general effect of sleep deprivation when making predictions
- **partial pooling**:
  - draws on all the information in the data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

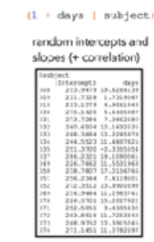
38

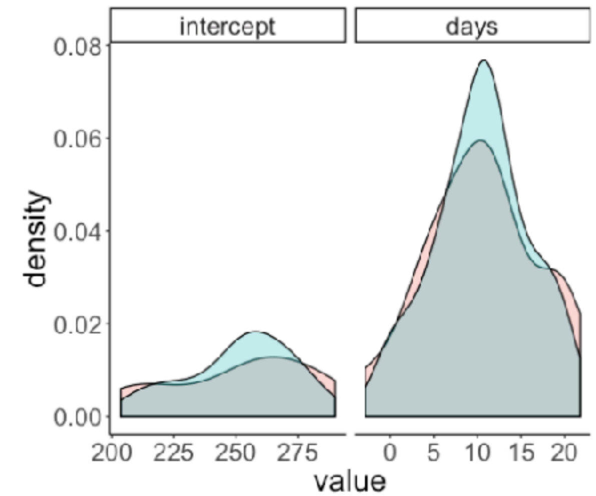
Shrinkage

separate regression for each participant

lmer()







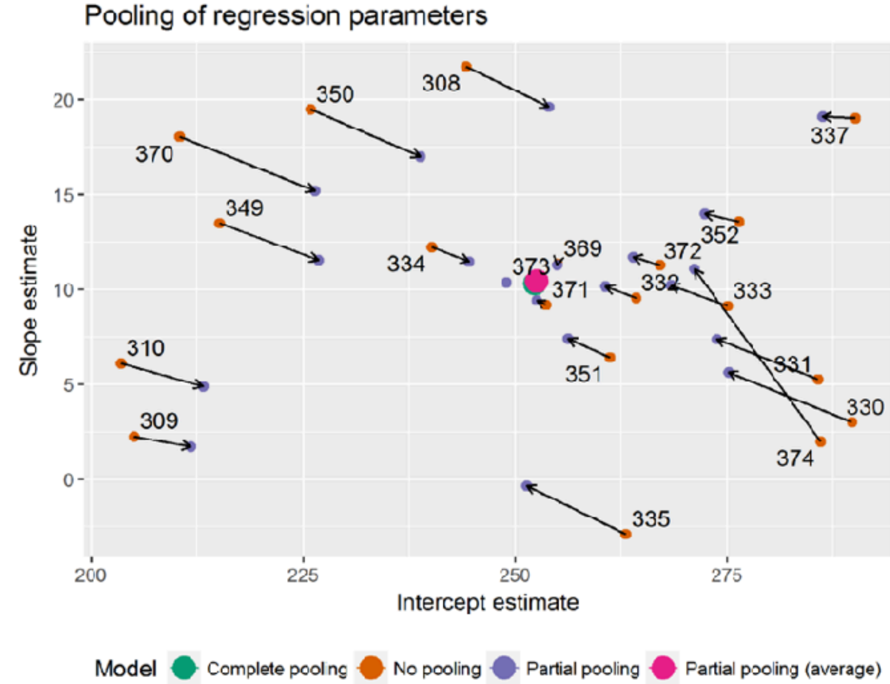
standard deviation

method	intercept	days
no pooling	28.95	6.56
partial pooling	21.59	5.48

variance "shrinks"

39

Shrinkage



40

Quick recap: Understanding the syntax

lmer() syntax summary

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

49

Coefficients

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

`(1 | subject)`

random intercepts

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

`(0 + days | subject)`

random slopes

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

`(1 + days | subject)`

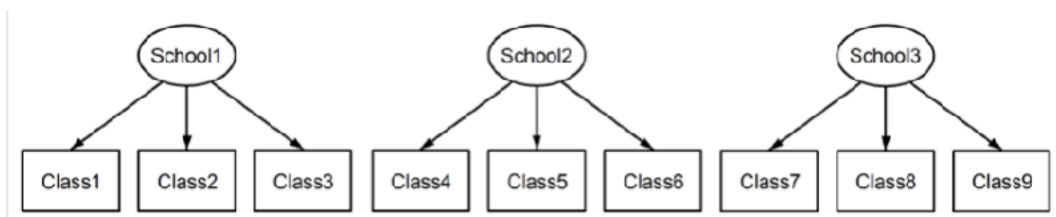
random intercepts and slopes (+ correlation)

```
$subject
     (Intercept)       days
308     253.9479 19.6264139
309     211.7328  1.7319567
310     213.1579  4.9061843
330     275.1425  5.6435987
331     273.7286  7.3862680
332     260.6504 10.1632535
333     268.3584 10.2245979
334     244.5523 11.4837825
335     251.3700 -0.3355554
337     286.2321 19.1090061
349     226.7562 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
```

50

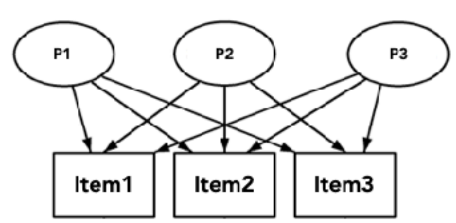
Multi-level models

nested (1|School/Class)



each class only appears within one school

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



each participant rates each item

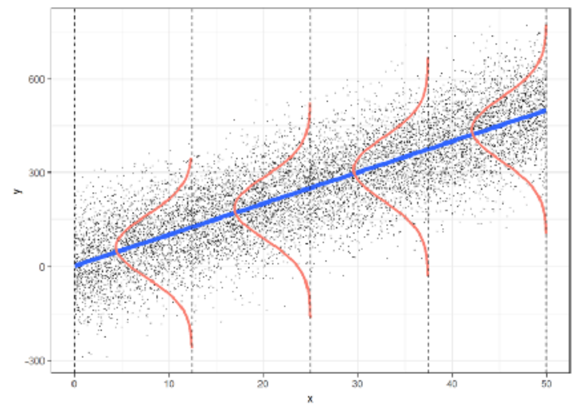
51

Centering predictors

Assumptions of **multiple** regression

- independent observations
- *Y* is continuous
- errors are normally distributed
- errors have constant variance
- error terms are uncorrelated
- **no multicollinearity**

predictors in the model should not be highly correlated with each other



18

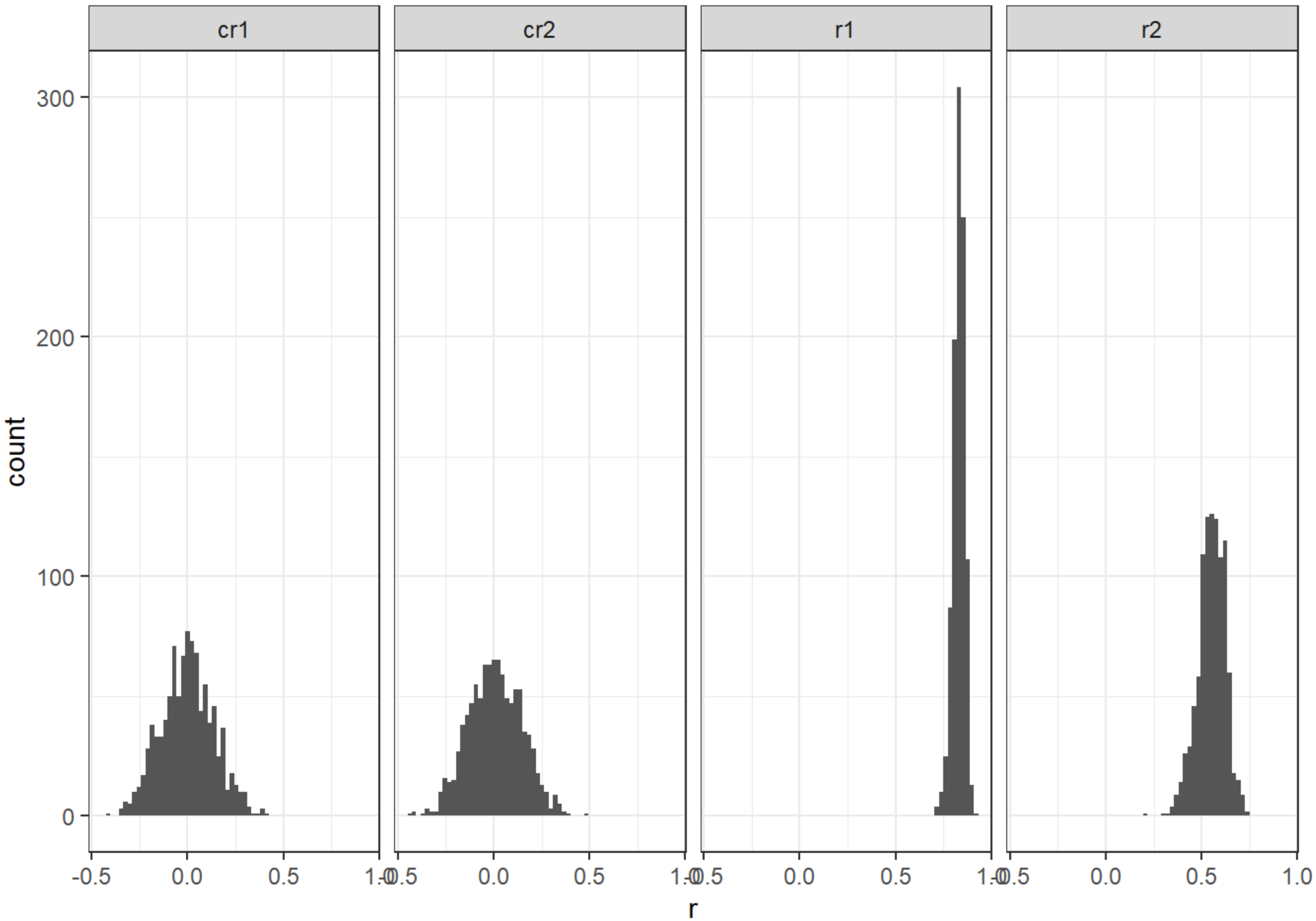
```
set.seed(1)

x1 = rnorm(100, 10, 1)
x2 = rnorm(100, 15, 1)
x1x2 = x1*x2

x1c = x1 - mean(x1)
x2c = x2 - mean(x2)
x1x2c = x1c * x2c
df.data = tibble(x1, x2, x1x2, x1c, x2c, x1x2c)
```

term	x1	x2	x1x2	x1c	x2c
x2	0.21				
x1x2	0.18	0.92			
x1c	1.00	0.21	0.18		
x2c	0.21	1.00	0.92	0.21	
x1x2c	-0.34	-0.40	-0.01	-0.34	-0.40

this is not great

this is much better

Centering predictors



correlation of each predictor with the interaction over many simulation runs

https://francish.netlify.app/post/centering-multicollinearity/

Centering predictors

Consider the basic equation for a correlation:

$$r_{(X,Y)} = \frac{cov(X,Y)}{\sqrt{(var(X) \cdot var(Y))}}$$

For the product score (X1X2) and X1:

$$r_{(X1X2,X1)} = \frac{cov(X1X2,X1)}{\sqrt{(var(X1X2) \cdot var(X1))}}$$

Focusing only on the *numerator* and using covariance algebra, the covariance of a product score (X1X2) with another variable (X1) can be written as:

$$\begin{aligned} cov(AB,C) &= \mathbb{E}(A) \cdot cov(B,C) + \mathbb{E}(B) \cdot cov(A,C) \\ &= \mathbb{E}(X1) \cdot cov(X2,X1) + \mathbb{E}(X2) \cdot cov(X1,X1) \\ &= \mathbb{E}(X1) \cdot cov(X2,X1) + \mathbb{E}(X2) \cdot var(X1) \end{aligned}$$

With **mean-centered** variables:

$$r_{((X1-\bar{X}1)(X2-\bar{X}2),(X1-\bar{X}1))} = \frac{cov((X1-\bar{X}1)(X2-\bar{X}2),(X1-\bar{X}1))}{\sqrt{var((X1-\bar{X}1)(X2-\bar{X}2)) \cdot var((X1-\bar{X}1))}}$$

Focusing only on the *numerator* again:

$$\begin{aligned} &= \mathbb{E}(X1-\bar{X}1) \cdot cov(X2-\bar{X}2, X1-\bar{X}1) + \mathbb{E}(X2-\bar{X}2) \cdot cov(X1-\bar{X}1, X1-\bar{X}1) \\ &= \mathbb{E}(X1-\bar{X}1) \cdot cov(X2-\bar{X}2, X1-\bar{X}1) + \mathbb{E}(X2-\bar{X}2) \cdot var(X1-\bar{X}1) \end{aligned}$$

The expected value though of a mean centered variable is zero. So if the numerator is zero, the whole equation reduces to zero (on average).

$$= 0 \cdot cov(X2-\bar{X}2, X1-\bar{X}1) + 0 \cdot var(X1-\bar{X}1)$$

https://francish.netlify.app/post/centering-multicollinearity/

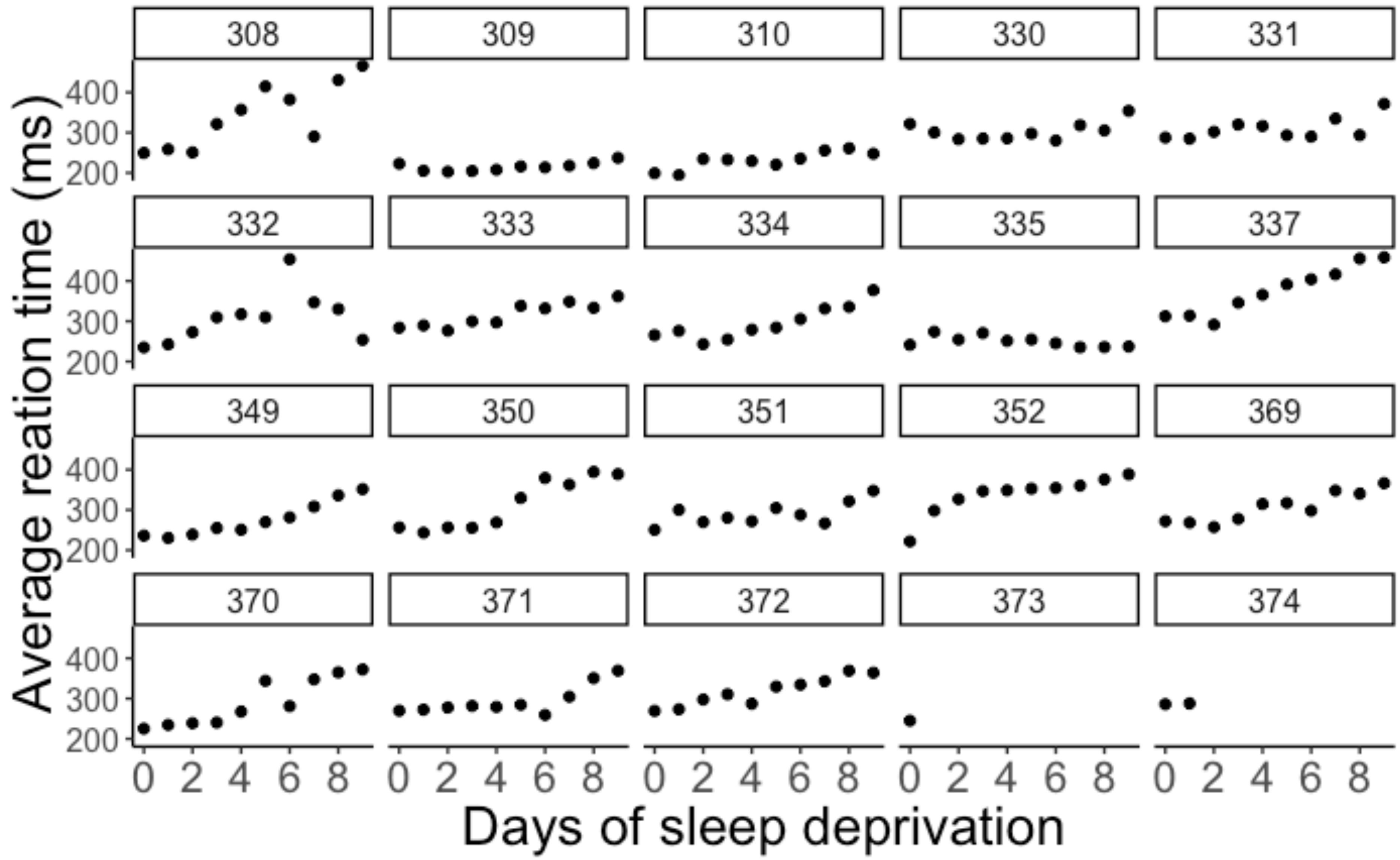
Some more examples

1. Sleep

Sleep data

How does sleep deprivation affect reaction time?

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



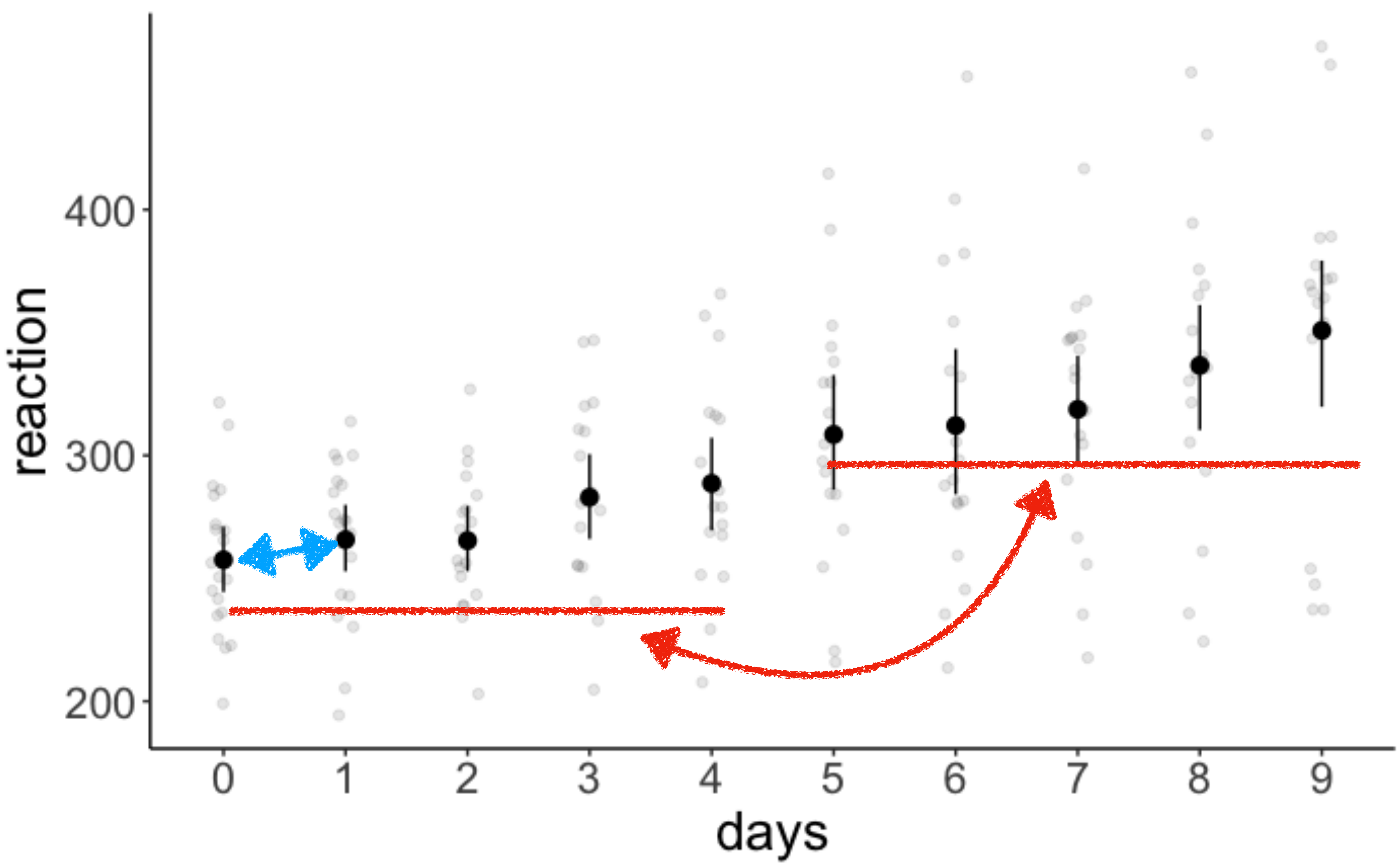
20 participants 2 with incomplete information

Testing specific hypotheses with linear contrasts

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

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

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



Sleep data

fit the model

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

Sleep data

fit the model

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

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

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

define the contrasts

test the contrasts

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

Degrees-of-freedom method: kenward-roger

days	reaction
0	257.54
1	265.73

index	reaction
early	271.67
late	325.39

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

2. Weight loss

Weight loss data

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

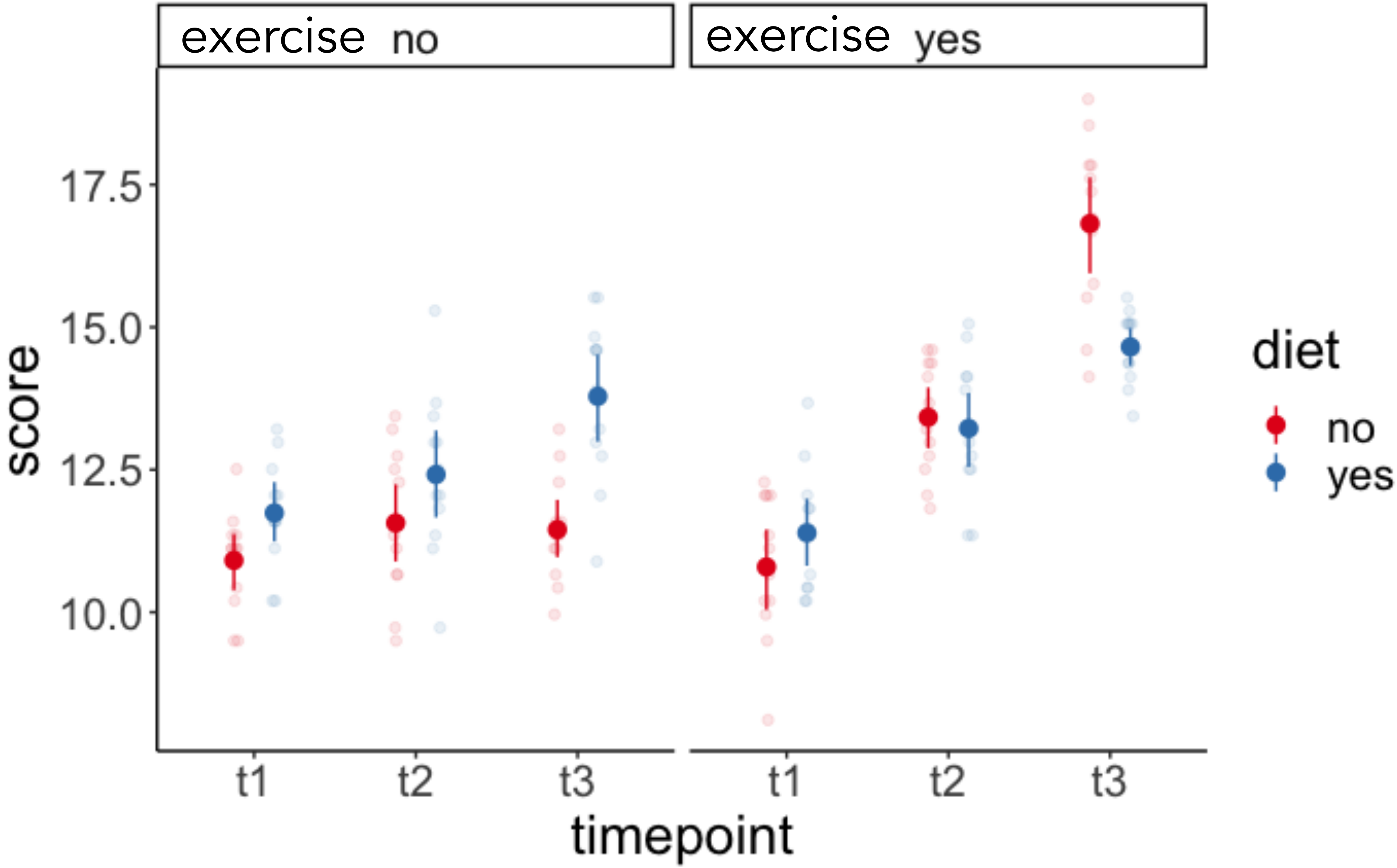
between participants: exercise yes/no

within participants: diet yes/no

within participants: time points

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

Weight loss data



Weight loss data

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

df.weightloss

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

```
Anova Table (Type 3 tests)

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

main effects and interactions

Weight loss data

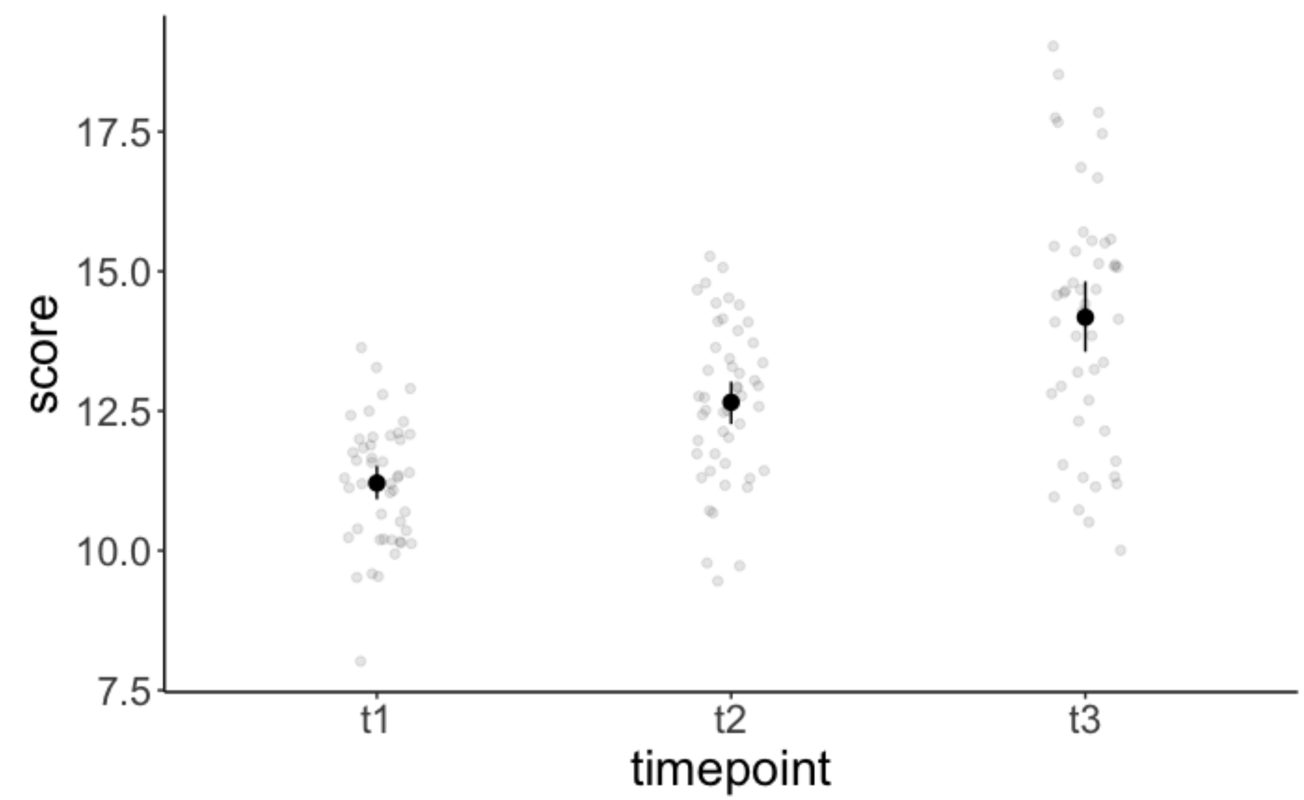
1. Is the score at the third time point different from the other two time points?
2. Is there a linear increase across time points?

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

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

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

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

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



3. politeness

Politeness

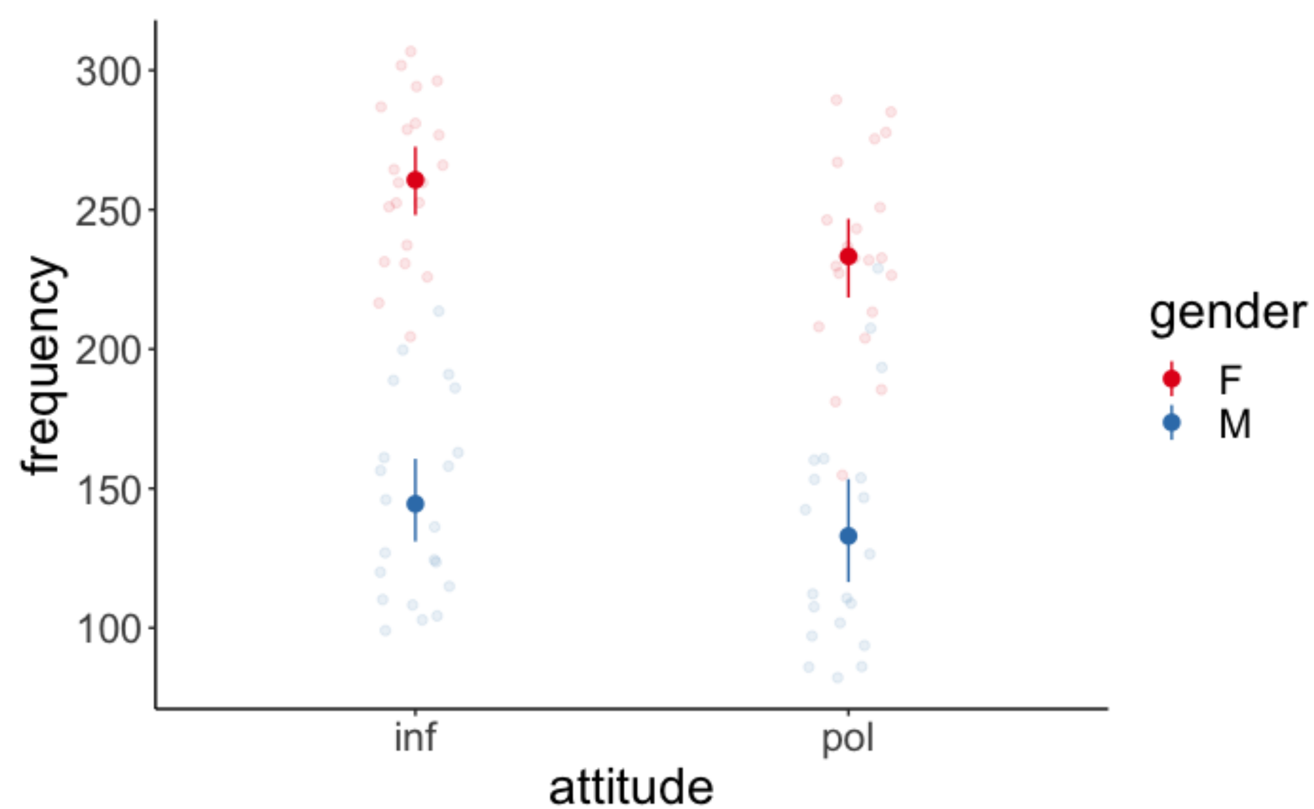
subject	gender	scenario	attitude	frequency
F1	F	1	pol	213.3
F1	F	1	inf	204.5
F1	F	2	pol	285.1
F1	F	2	inf	259.7
F1	F	3	pol	203.9
F1	F	3	inf	286.9
F1	F	4	pol	250.8
F1	F	4	inf	276.8
F1	F	5	pol	231.9
F1	F	5	inf	252.4
F1	F	6	pol	181.2
F1	F	6	inf	230.7
F1	F	7	inf	216.5
F1	F	7	pol	154.8
F3	F	1	pol	229.7

gender: female, male

scenario: different text prompt

attitude: informal vs. polite

frequency: pitch of voice

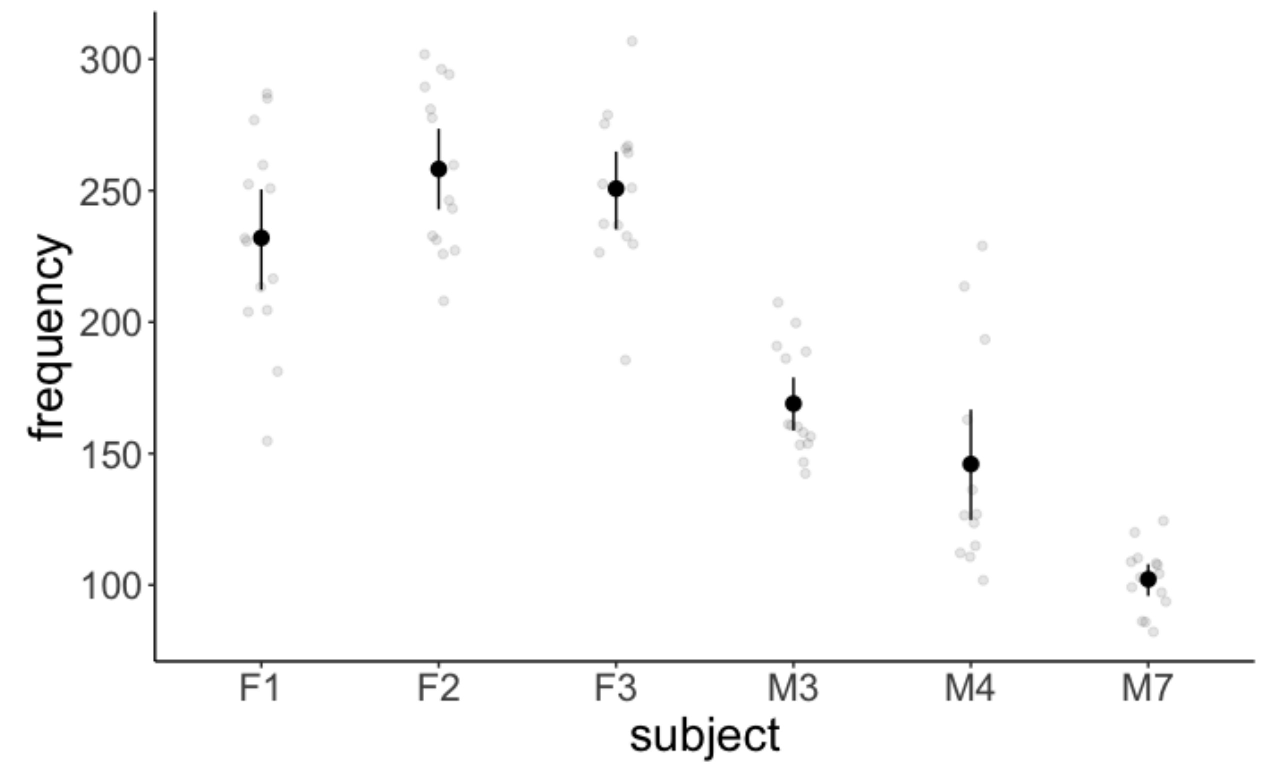


https://jontalle.web.engr.illinois.edu/MISC/lme4/bw_LME_tutorial.pdf

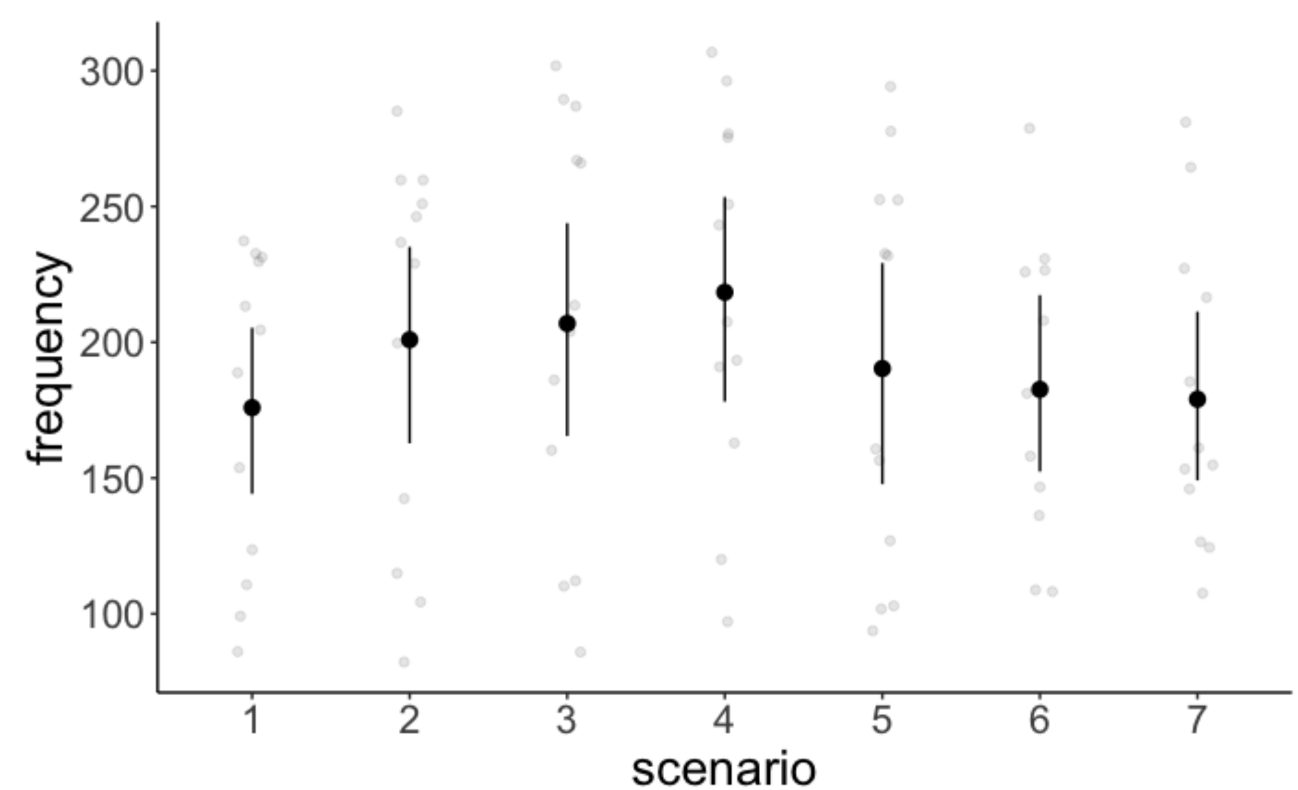
Politeness

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

variation across subjects



variation across scenarios

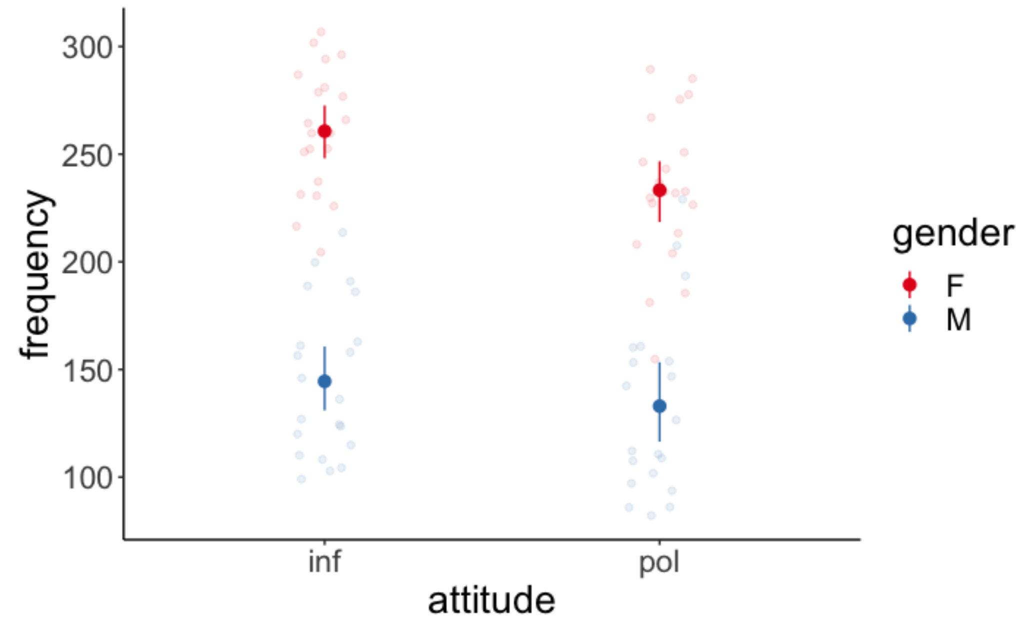


Politeness

Was there an effect of gender and attitude on pitch?

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

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



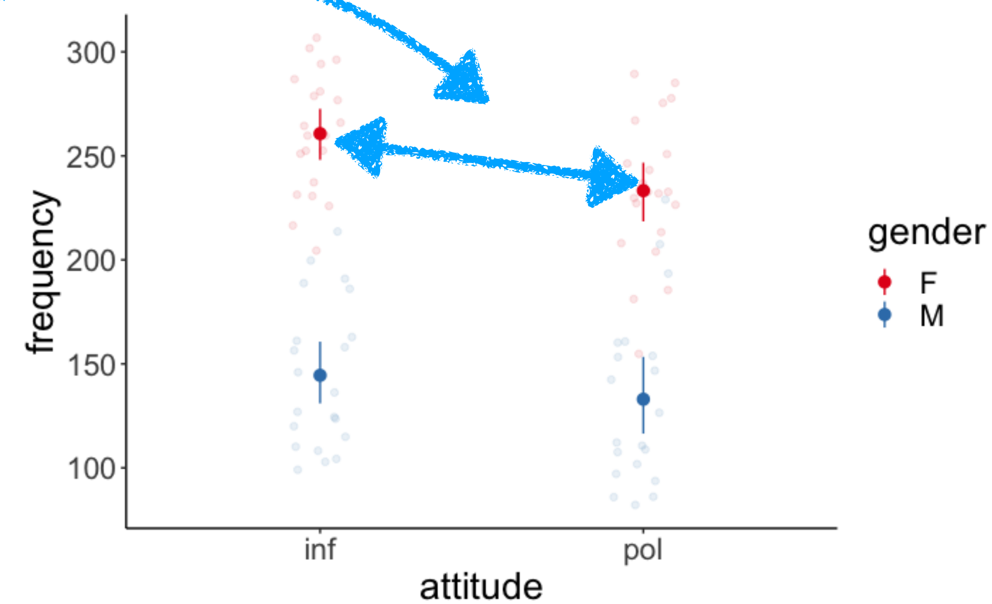
main effect of attitude, main effect of gender, no significant interaction effect

Politeness

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

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

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

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

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



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

Politeness

Was there an effect of gender and attitude on pitch?

ANOVA

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

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

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

ignores variation between scenarios, and just takes the mean

interaction effect

LMER

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

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

no interaction effect

`lmer()` standard operating procedures

Standard Operating Procedures For Using Mixed-Effects Models

Standard Operating Procedures For Using Mixed-Effects Models

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

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

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

Content/Analysis Steps

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

What shall I include as random effects?

- mixed opinions on the topic
- go maximal!

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

Contents lists available at SciVerse ScienceDirect

Journal of Memory and Language

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

Random effects structure for confirmatory hypothesis testing: Keep it maximal

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

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

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

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

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

What shall I include as random effects?

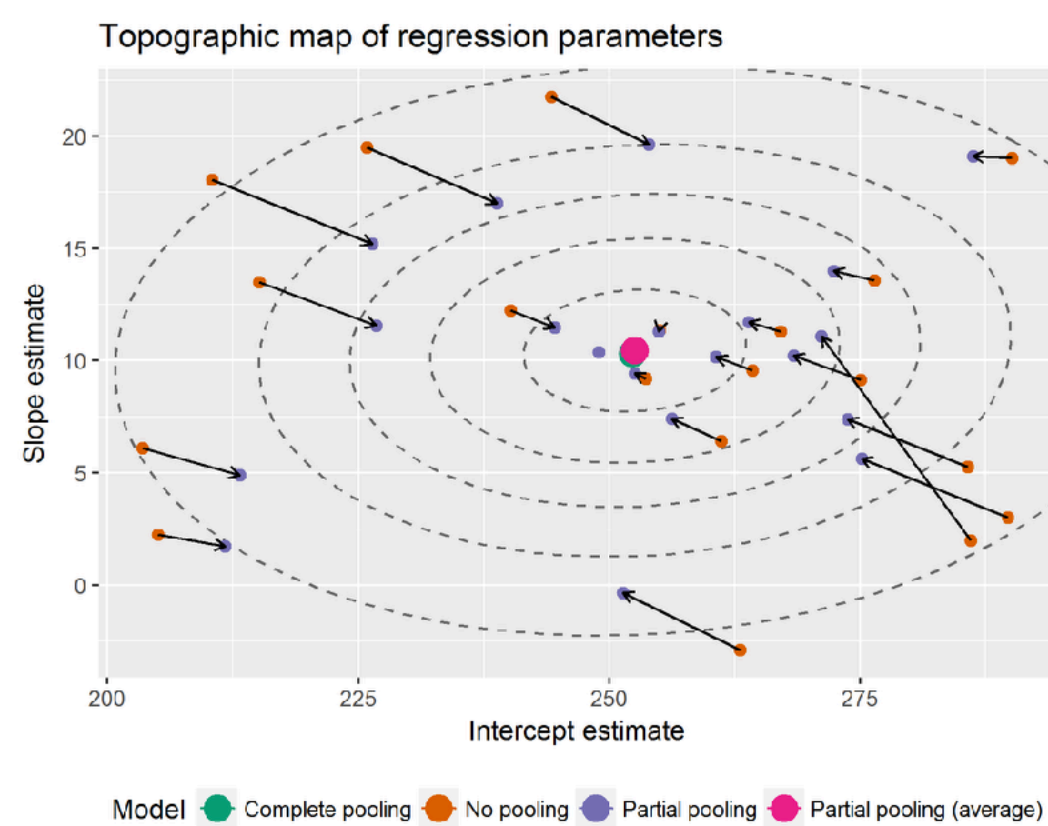
- general advice:
  - start maximal (as supported by the design)
    - random intercepts for different participants
    - random slopes when participants are tested multiple times
    - random intercepts for items
  - reduce complexity of the random effects structure step by step
    - remove the correlation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

Remove the correlation component from your model

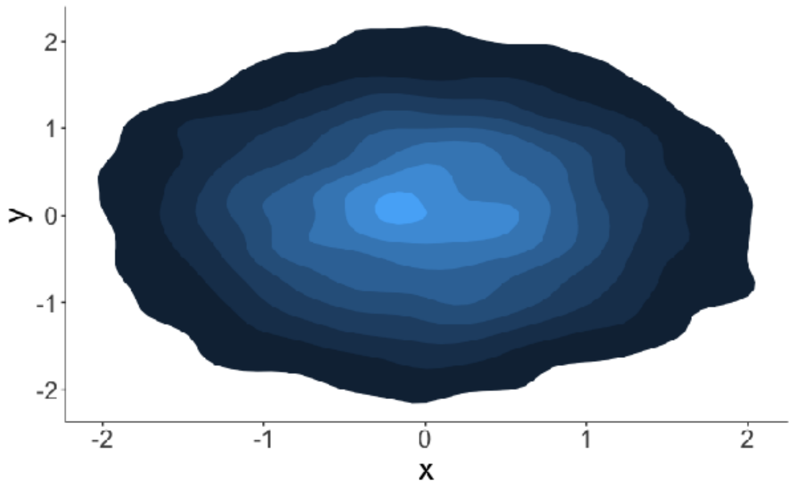
```r
# fit the model
fit.lmer = lmer(formula = reaction ~ 1 + days + (1 + days | subject),
                data = df.sleep)
# model summary
fit.lmer %>%
  summary()
```

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

REML criterion at convergence: 1771.4

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

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

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

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

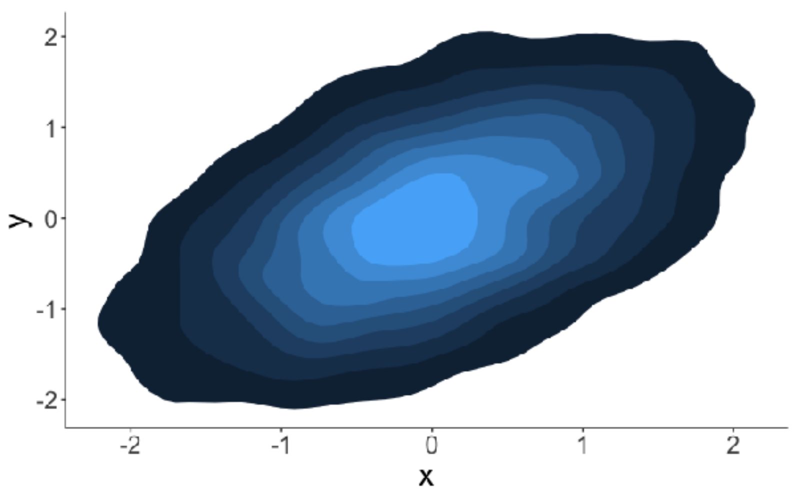
multivariate Gaussian



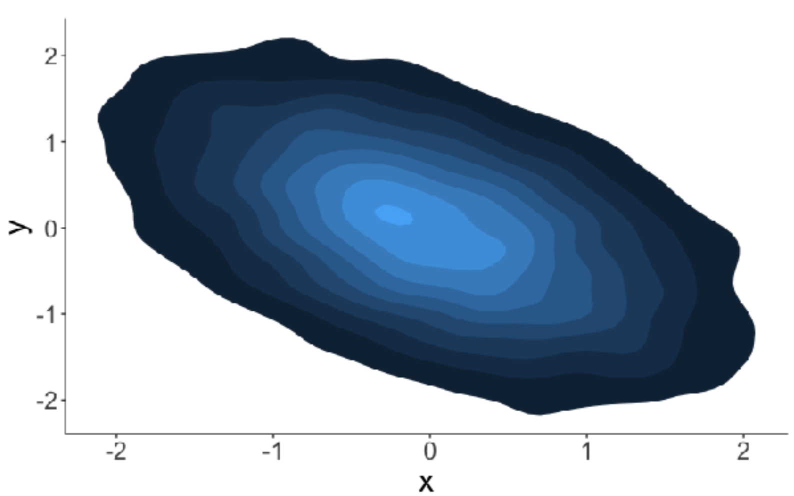
Remove the correlation component from your model



uncorrelated



positively correlated



negatively correlated

Remove the correlation component from your model

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

random slopes

random intercepts

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

REML criterion at convergence: 1771.5

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

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

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

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

independent Gaussians

Remove the correlation component from your model

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

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

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

REML criterion at convergence: 1771.5

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

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

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

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

independent Gaussians

What if `lmer()` fails to converge?

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

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

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

comparison of the different optimization algorithms

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

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

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

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

What if `lmer()` fails to converge?

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

3.2.2. Or we choose a Bayesian approach

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

Plan for today

- Quick Recap
- Linear mixed effects model
  - Some examples
  - `lmer()` standard operating procedures

Feedback

How was the pace of today's class?

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