

# Multilevel Modeling (with R)

Princeton University

Jason Geller, PH.D.

4/20/23



# Overview

- What are multilevel models and why are they awesome?
- Important terminology
- Interpretation and estimation of multilevel models in R

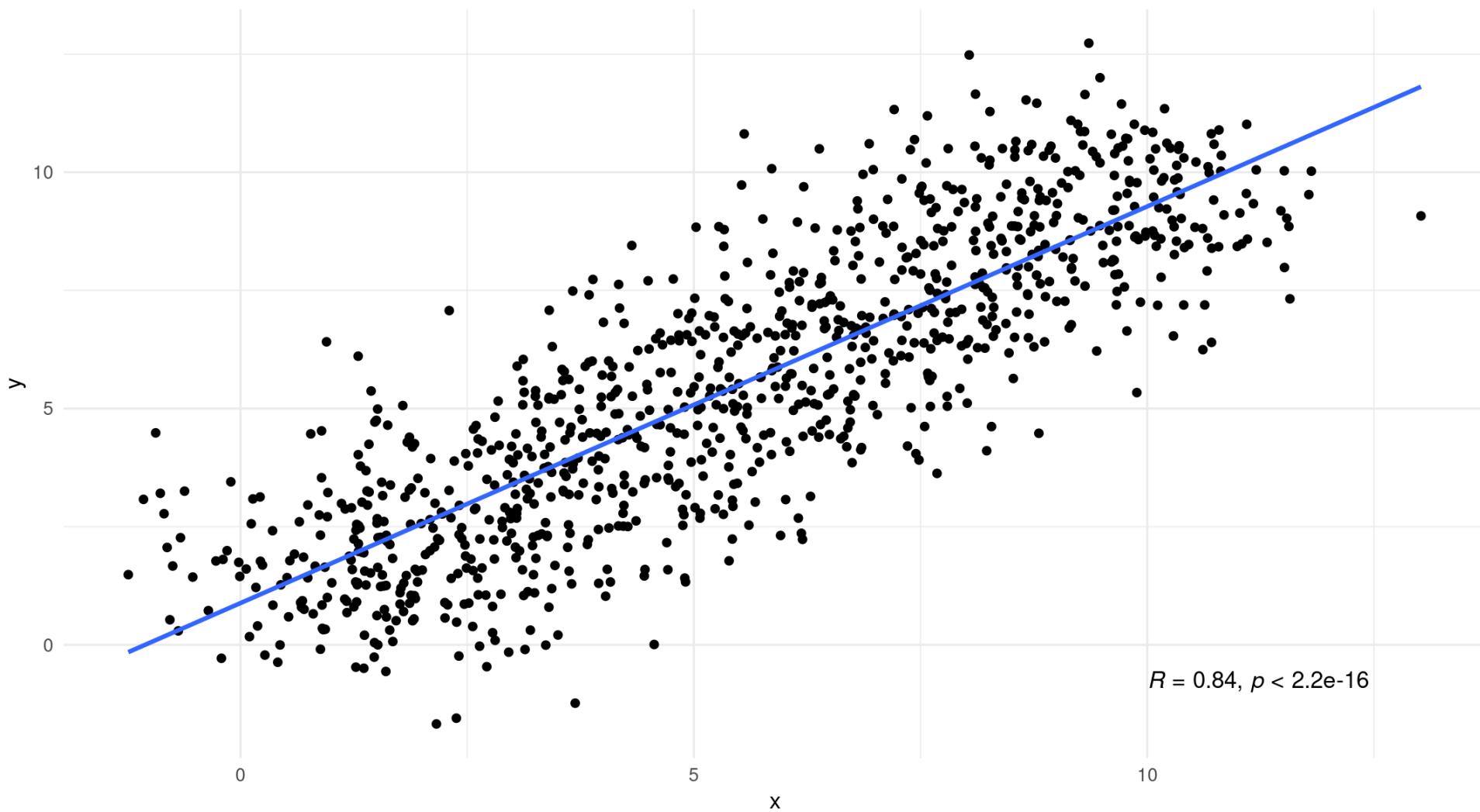
# What I wont be talking about

- Cross-classified models
- Partial-pooling/shrinkage
- Growth curve modeling
- Different types of degrees of freedom
- Grand-mean vs. group-mean centering

**If you have specific questions about these things ask me**

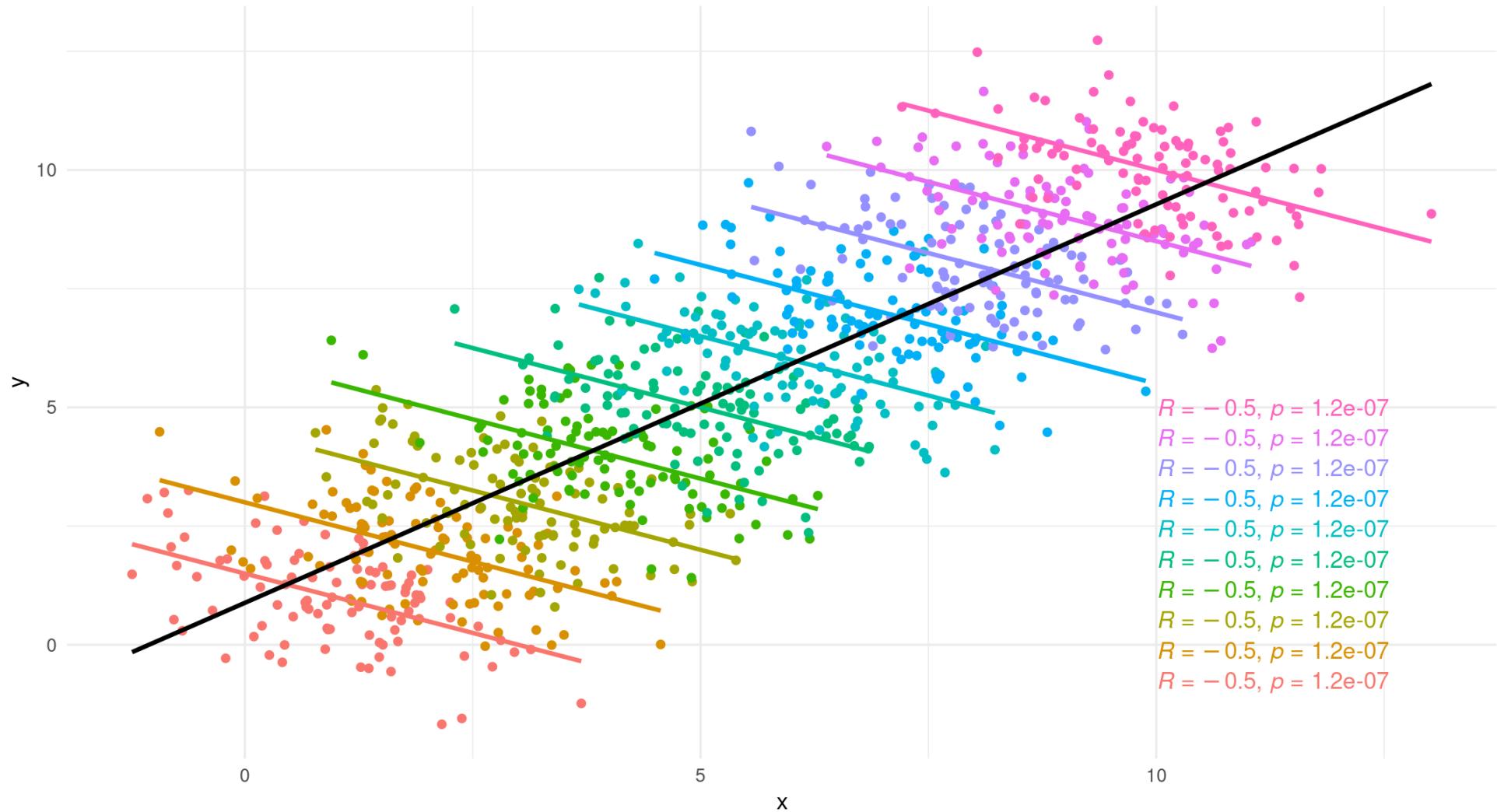
# Why multilevel modeling?

- You might be used to your data looking like this: An independent variable (x) and a dependent variable (y)



# Why multilevel modeling?

- However, if we introduce grouping we tell a different story



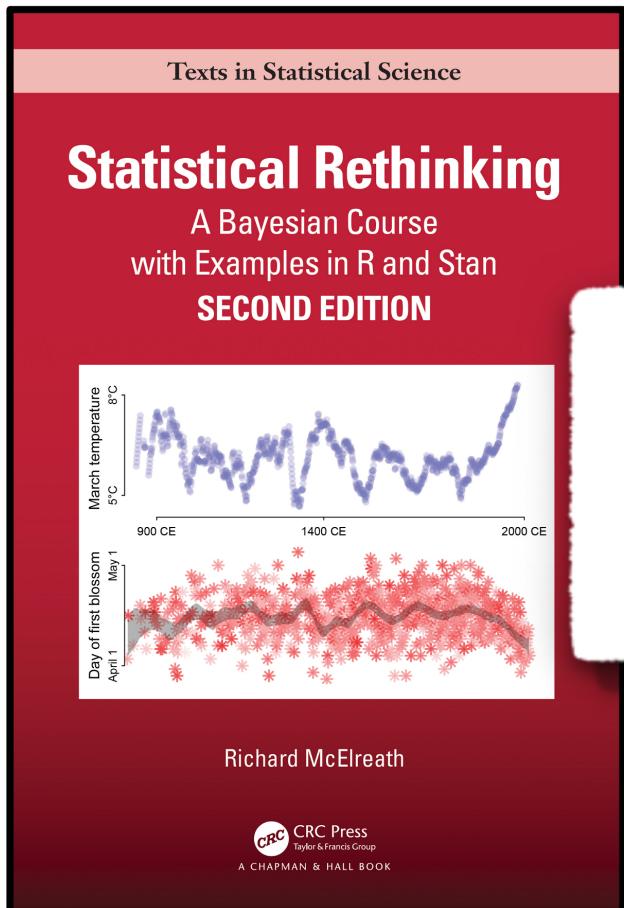


# What is multilevel modeling?

- Simpson's Paradox



# What is multilevel modeling?



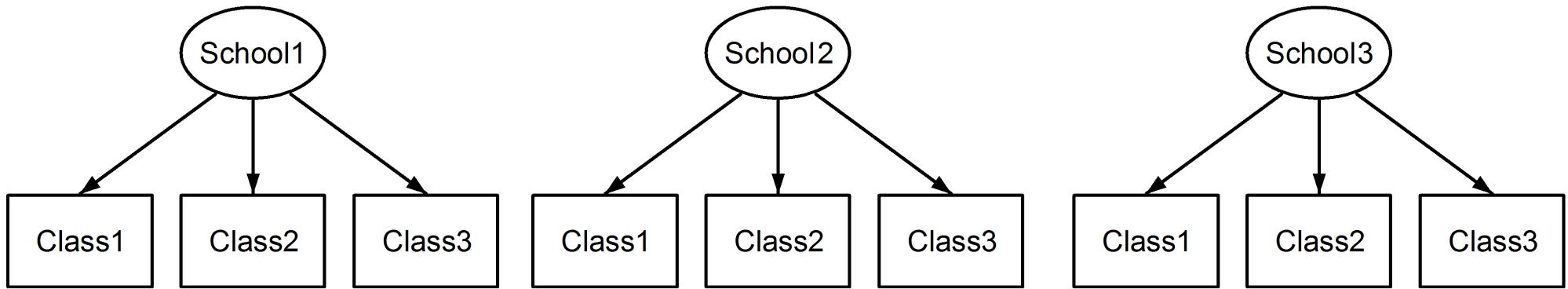
“When it comes to regression,  
multilevel regression deserves to be  
the default approach.”

# What is multilevel modeling?

- An elaboration on regression to deal with non-independence between data points (i.e., clustered data)



# Hierarchies



- For now we will focus on data with two levels:
  - Level one: Most basic level of observation
  - Level two: Groups formed from aggregated level-one observation

# Why multilevel models are awesome

- *Interdependence*
  - You can model the relationships between cases (regression for repeated observations)
- *Missing data*
  - Uses ML for missing data (partial pooling or shrinkage)
- *Power*
- Take into account within and between variance

# Multilevel models

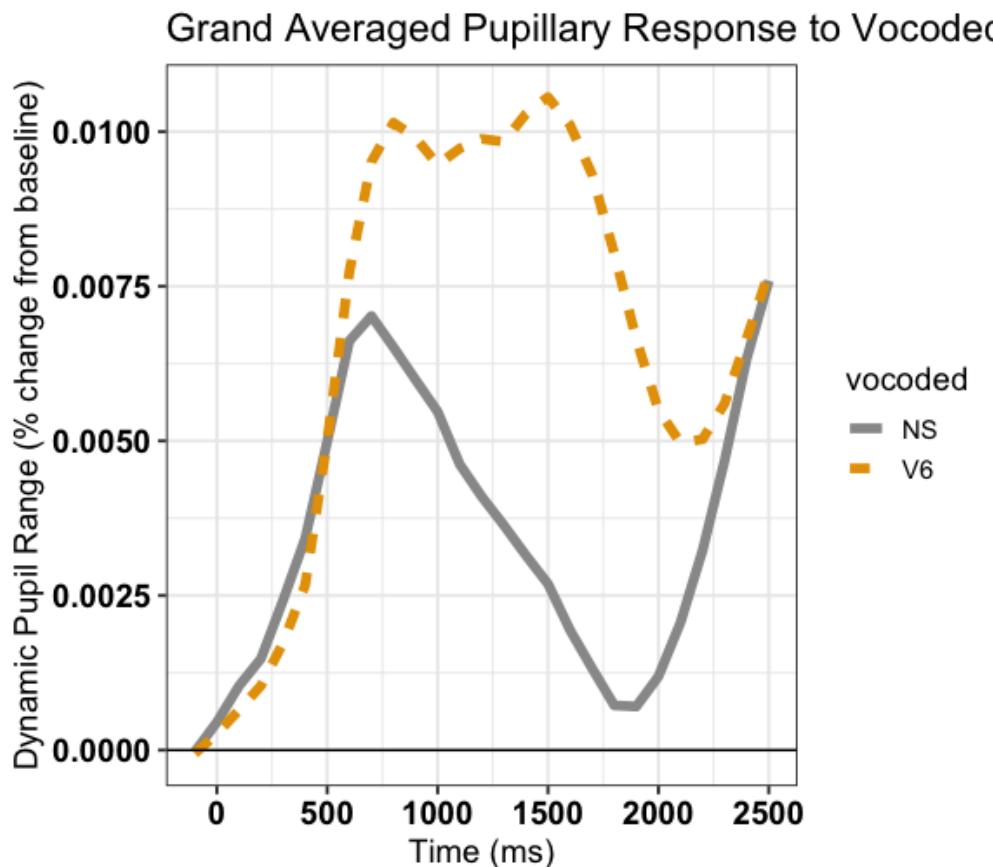
- When to use them:
  - Nested designs
  - Repeated measures
  - longitudinal data
- Why use them:
  - Captures variance occurring between groups and within groups
- What they are:
  - Linear model with extra residuals

# Jumping right in

- Words you hear constantly in MLM Land:
  - *Fixed effects*
  - *Random effects*
  - *Random intercepts*
  - *Random slopes*
- What do they all mean?

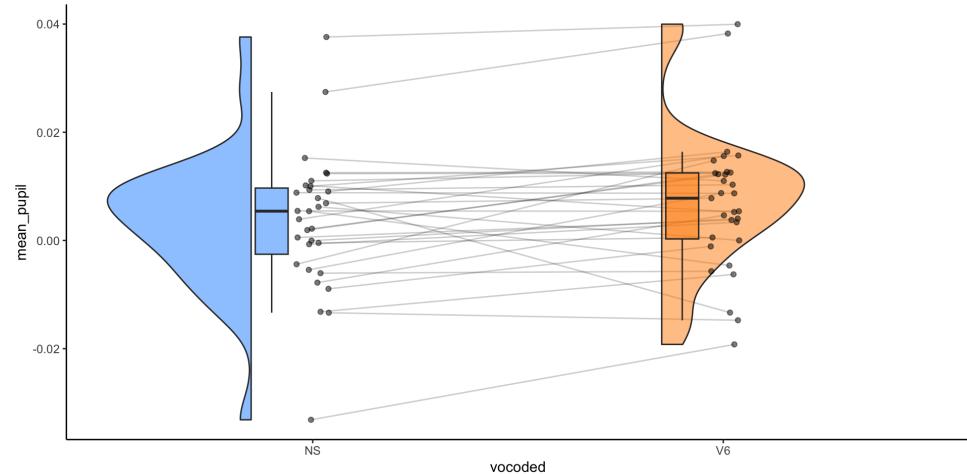
# Today's data

- What did you say?
  - Ps listened to clear or 6 channel vocoded speech ([https://www.mrc-cbu.cam.ac.uk/personal/matt.davis/vocode/a1\\_6.wav](https://www.mrc-cbu.cam.ac.uk/personal/matt.davis/vocode/a1_6.wav))



# Data

```
1 library(tidyverse)
2 library(lme4) # fit mixed models
3 library(broom.mixed) # tidy output of mixed models
4 library(afex) # fit mixed models
5 library(emmeans) # marginal means
6 library(ggeffects) # marginal means
7 eye <- read_csv(here::here("data", "eye.csv"))
```



# Fixed and random effects

- Fixed effect:
  - Assumed to be constant
  - Population-level (i.e., average) effects that should persist across experiments
  - Usually experimental manipulations
  - Can be continuous or categorical
- In our data: [vocoded](#)

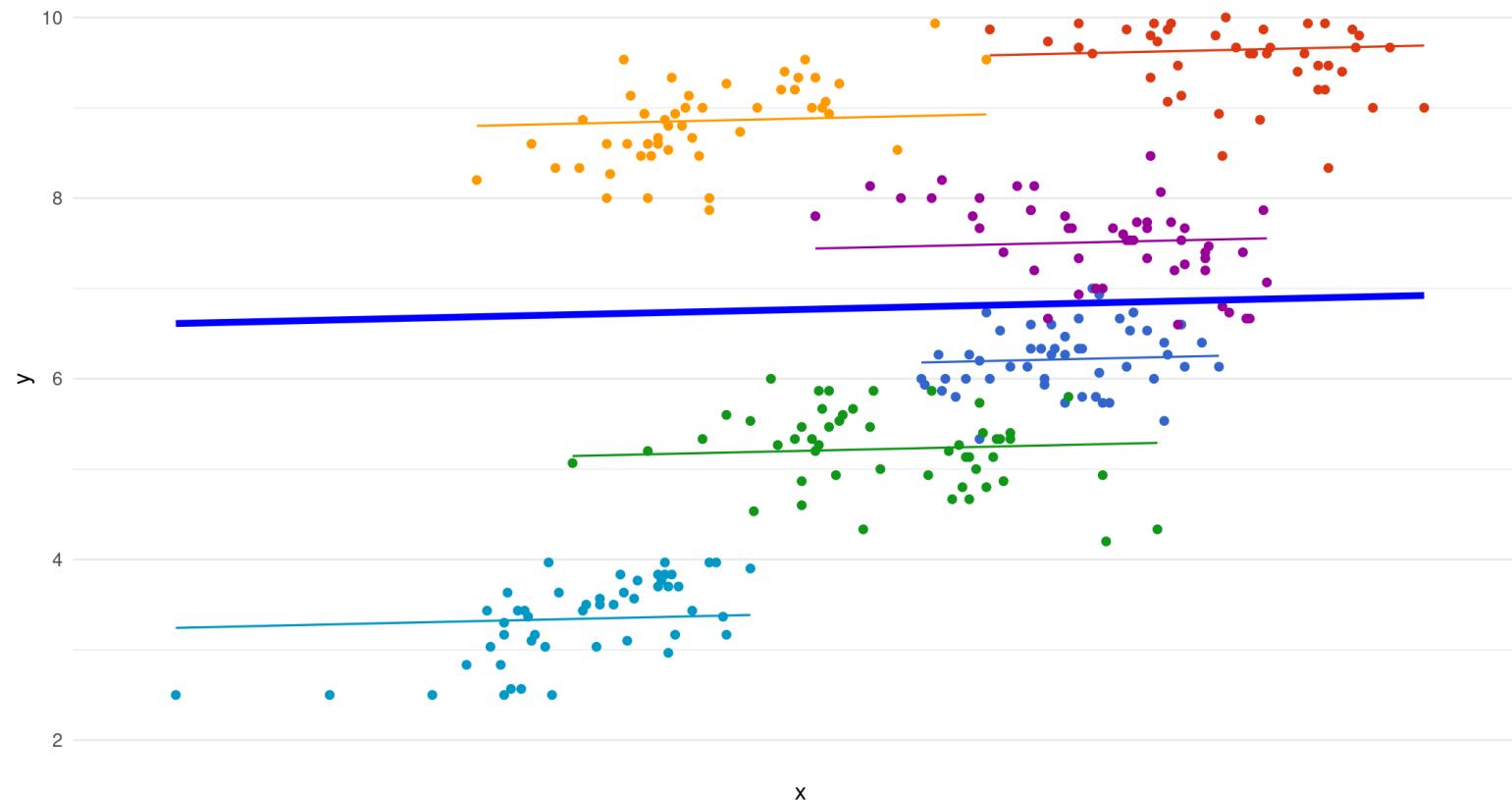
# Fixed and random effects

- Random effects:
  - Assumed to vary
    - Randomly sampled observations over which you plan to generalize
      - Participants
      - Schools
      - Words
      - Pictures
    - Can help account for individual variation
  - In our data: **subject**

# Random intercepts - Fixed slopes

- Varying starting point (intercept), same slope for each group
  - **(1|participant)**: random intercept for group

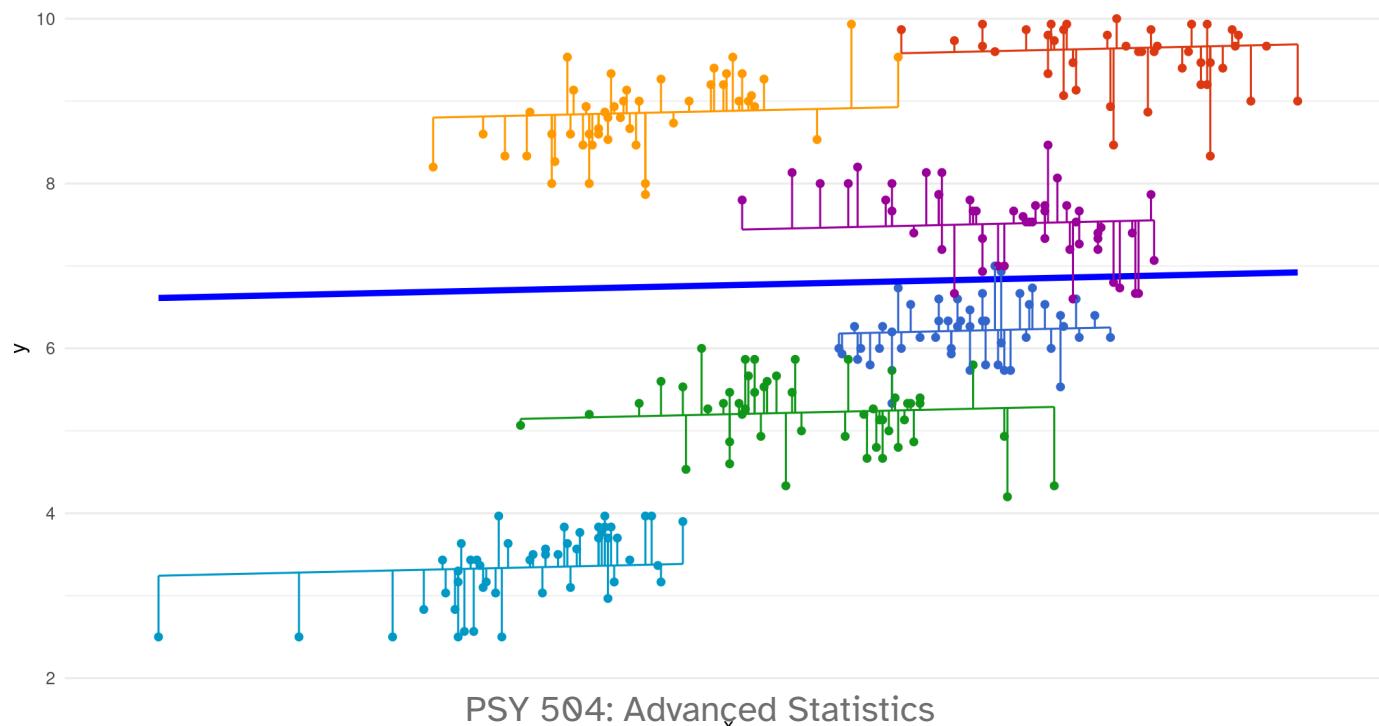
```
1 lmer(DV ~ (1 | participant))
```



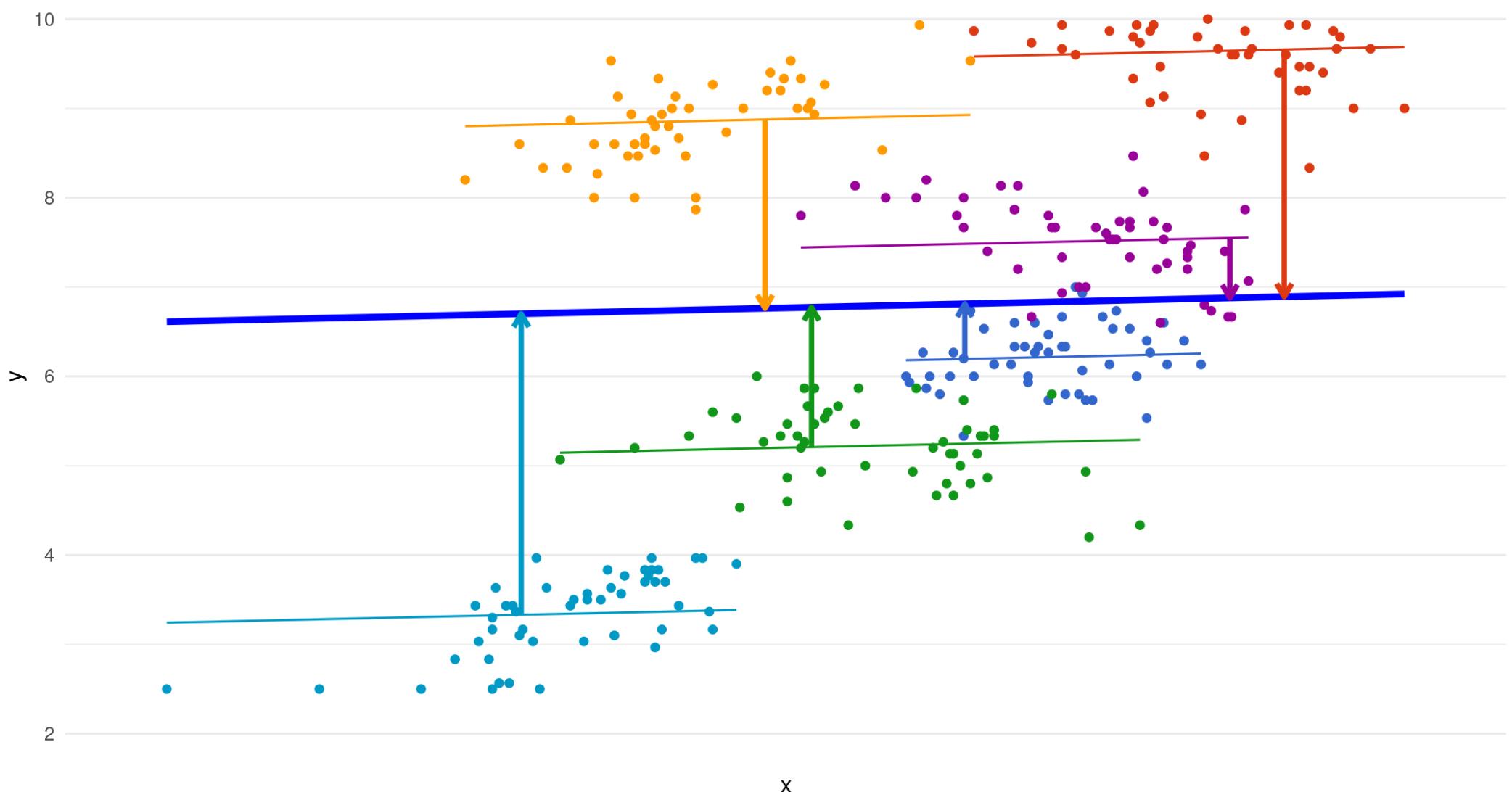
# Random intercepts - fixed slopes

- In a multilevel model, error terms for individual data points are estimated by group
  - Person-specific deviation from group's predicted outcome

$$y_{ij} = (\beta_0 + u_{0j})$$



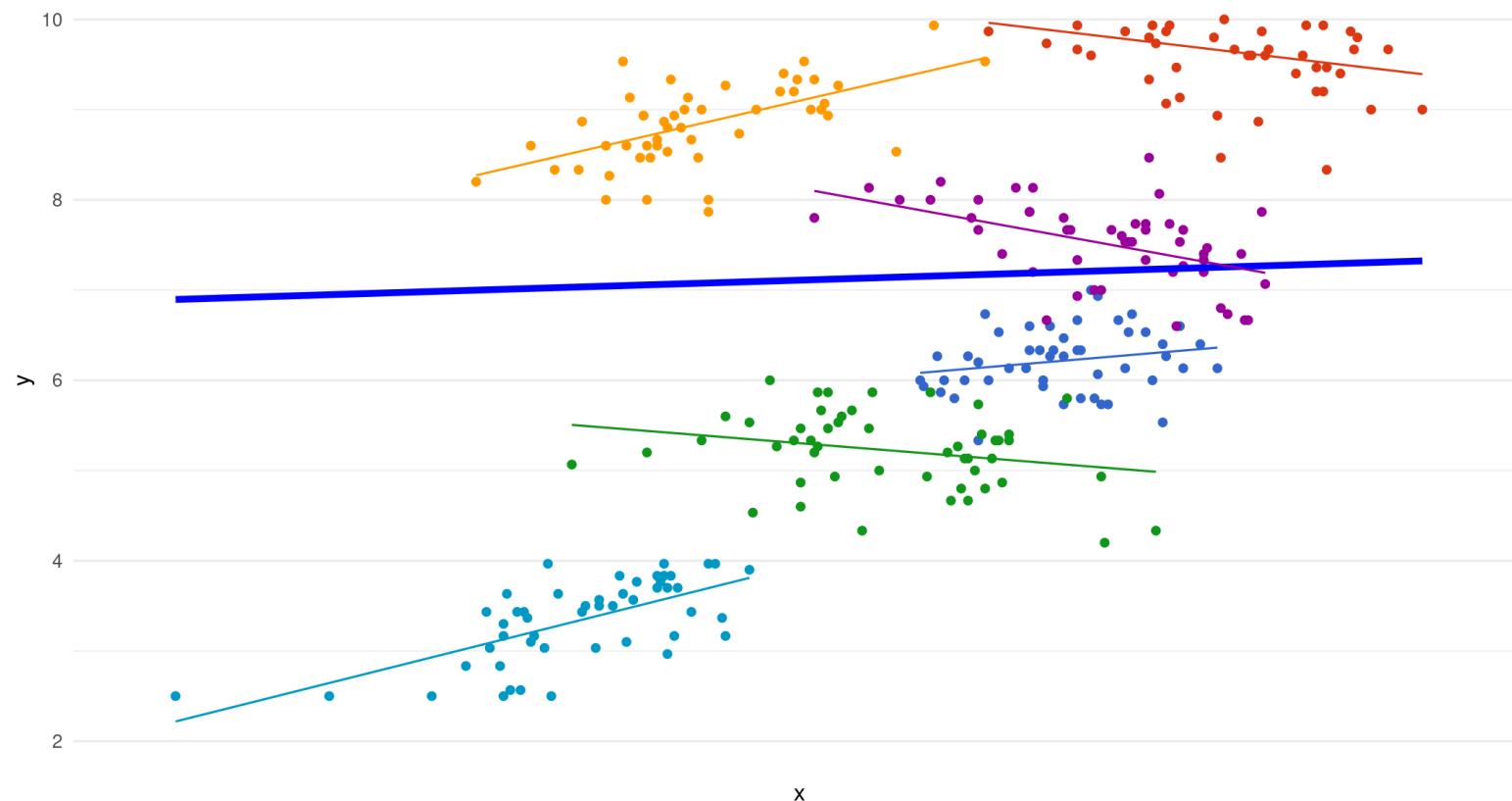
# Random Intercepts - Fixed Slopes



# Random intercepts - random slopes

- Varying starting point (intercept) and slope per group
- **(1+vocoded|group)**: random intercept and slopes per group

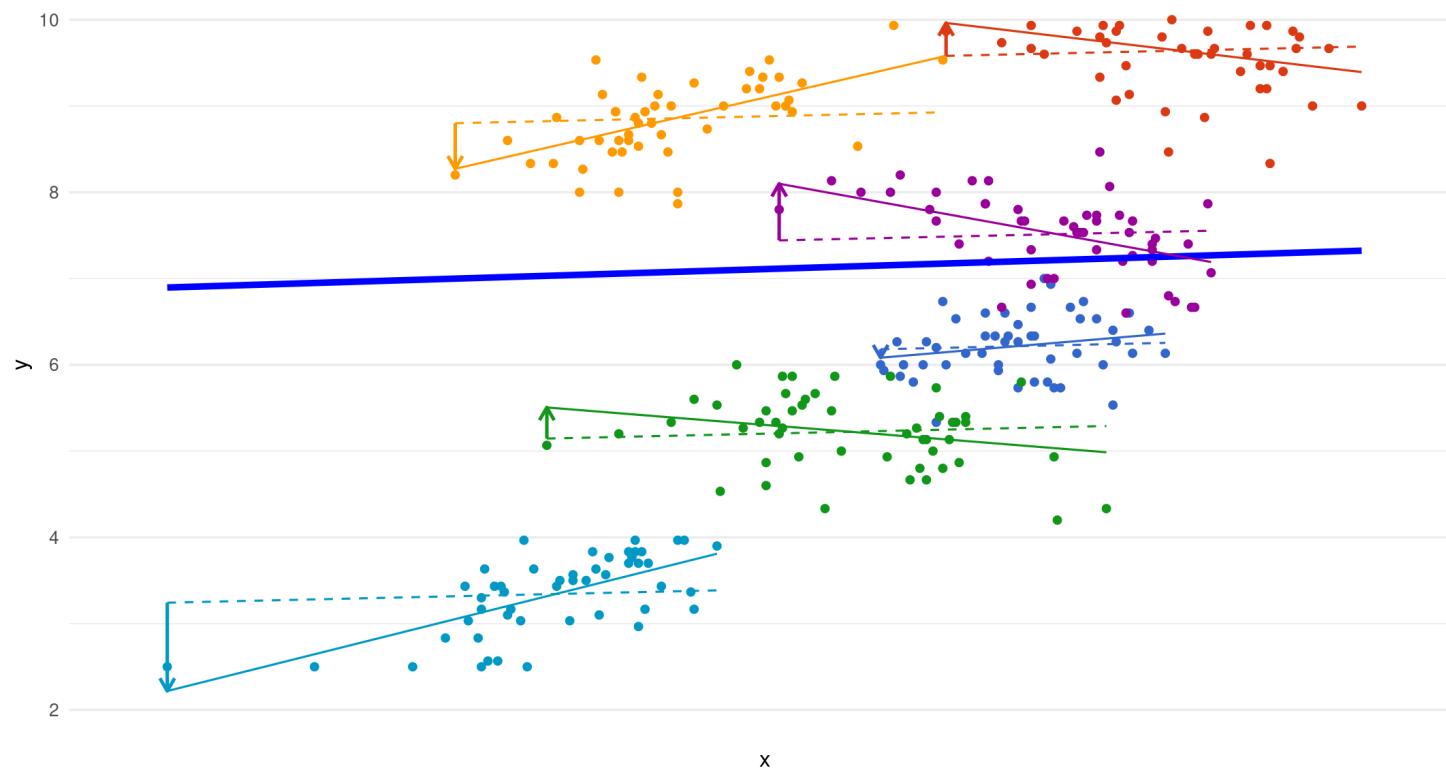
```
1 lmer(DV ~ vocoded + (1+vocoded|group))
```



# Random intercepts - random slope

- The dotted lines are fixed slopes. The arrows show the added error term for each random slope

$$(\beta_1 + u_{1j})$$

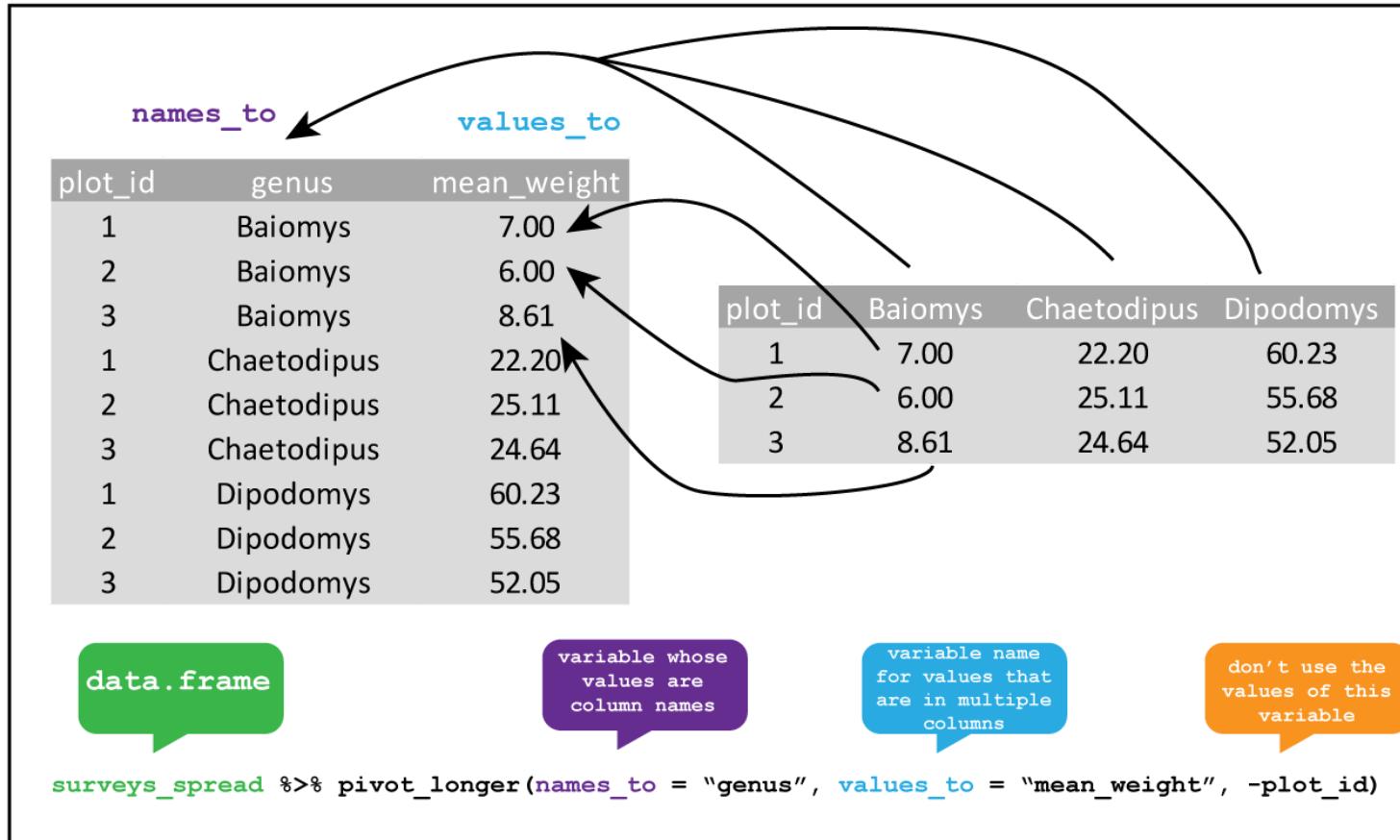


# Combined equation

$$y_{ic} = \beta_0 + \beta_1 \text{vocoded}_{ic} + u_{0j} + u_{1j} + \epsilon_{ic}$$

# Modeling

# Data organization



- Data Structure
  - Long vs. wide
  - MLM analysis requires data in long format

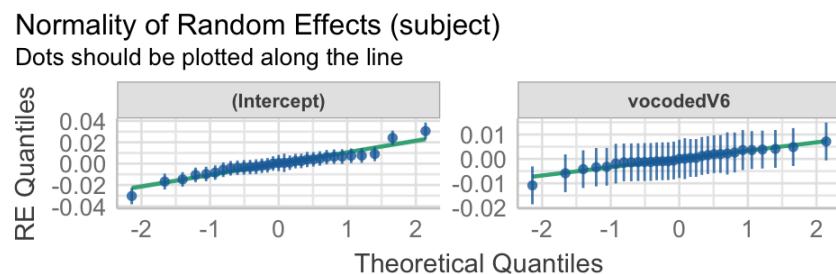
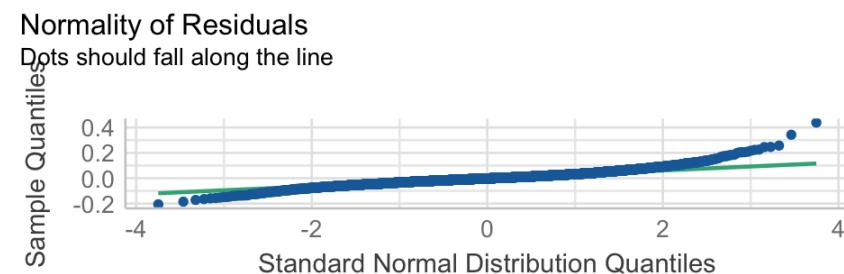
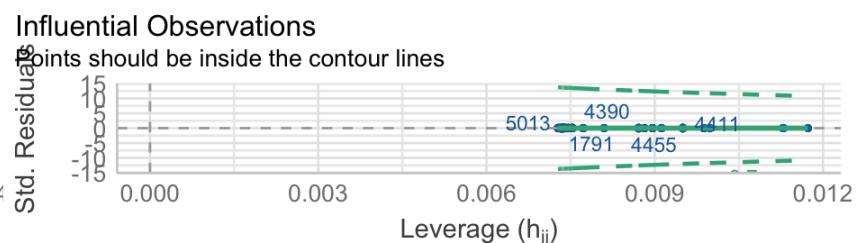
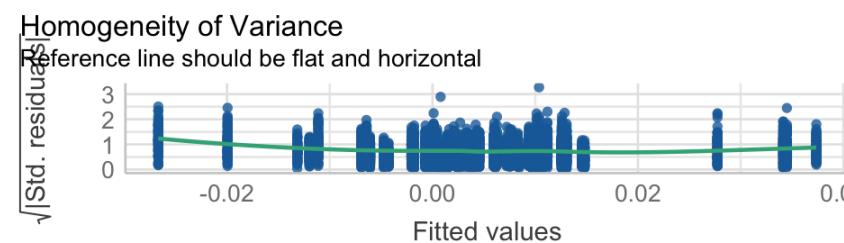
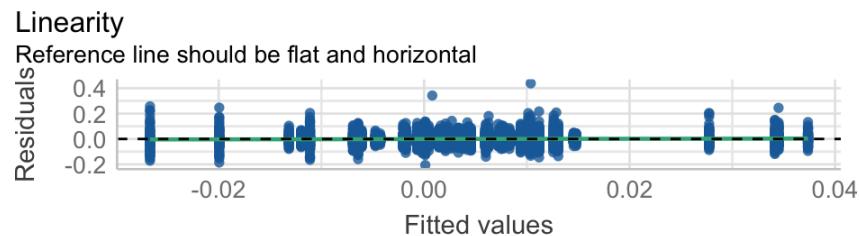
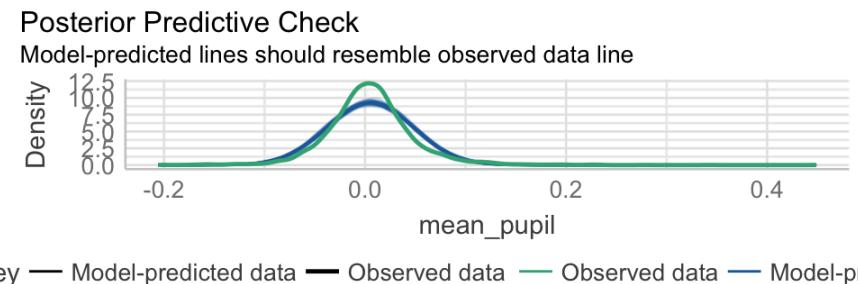
# Fitting and Interpretation of MLMs

# Modeling procedure

- Forward or backward
  - Null model (ICC)
  - Full model
    - if non-convergence (pay attention to warning messages):
      - First, remove correlation between random effects
      - If still not fixed, remove random effects
        - Start with random slopes

# Check assumptions

- Linearity
- Normality
- Homoscedasticity
- Collinearity
- Outliers



# Null model (unconditional means)

```
1 library(lme4) # pop linear modeling package  
2  
3 null_model <- lmer(mean_pupil ~ (1|subject), data = eye)  
4  
5 summary(null_model)
```

Linear mixed model fit by REML. t-tests use Satterthwaite's method [  
lmerModLmerTest]  
Formula: mean\_pupil ~ (1 | subject)  
Data: eye

REML criterion at convergence: -19811.6

Scaled residuals:

Min	1Q	Median	3Q	Max
-5.1411	-0.5530	-0.0463	0.4822	10.8130

Random effects:

Groups	Name	Variance	Std.Dev.
subject	(Intercept)	0.0001303	0.01142
Residual		0.0016840	0.04104

Number of obs: 5609, groups: subject, 31

# Random Effects

```
1 broom.mixed::tidy(null_model) %>% filter(effect == "ran_pars")
```

effect	group	term	estimate	std.error	statistic	df	p.value
ran_pars	subject	sd_(Intercept)	0.0114				
ran_pars	Residual	sd_Observation	0.041				

# Interclass correlation (ICC)

- ICC is a standardized way of expressing how much variance is due to clustering/group
  - Ranges from 0-1
- Also, can be interpreted as correlation among observations within cluster/group
- If ICC is sufficiently low (i.e.,  $\rho < .1$ ), then you don't have to use MLM! *BUT YOU PROBABLY SHOULD* 😊

# Calculating ICC

- Run baseline (null) model
- Get intercept variance and residual variance

$$ICC = \frac{\text{between-group variability}}{\text{between-group variability} + \text{within-group variability}}$$

$$ICC = \frac{\text{Var}(u_{0j})}{\text{Var}(u_{0j}) + \text{Var}(r_{ij})} = \frac{\tau_{00}}{\tau_{00} + \sigma^2}$$

# Fixed Effects

```
1 inter_model <- lmer(mean_pupil ~vocoded+(1|subject), data = eye)
2
3 broom.mixed:::tidy(inter_model) %>% filter(effect == "fixed")
```

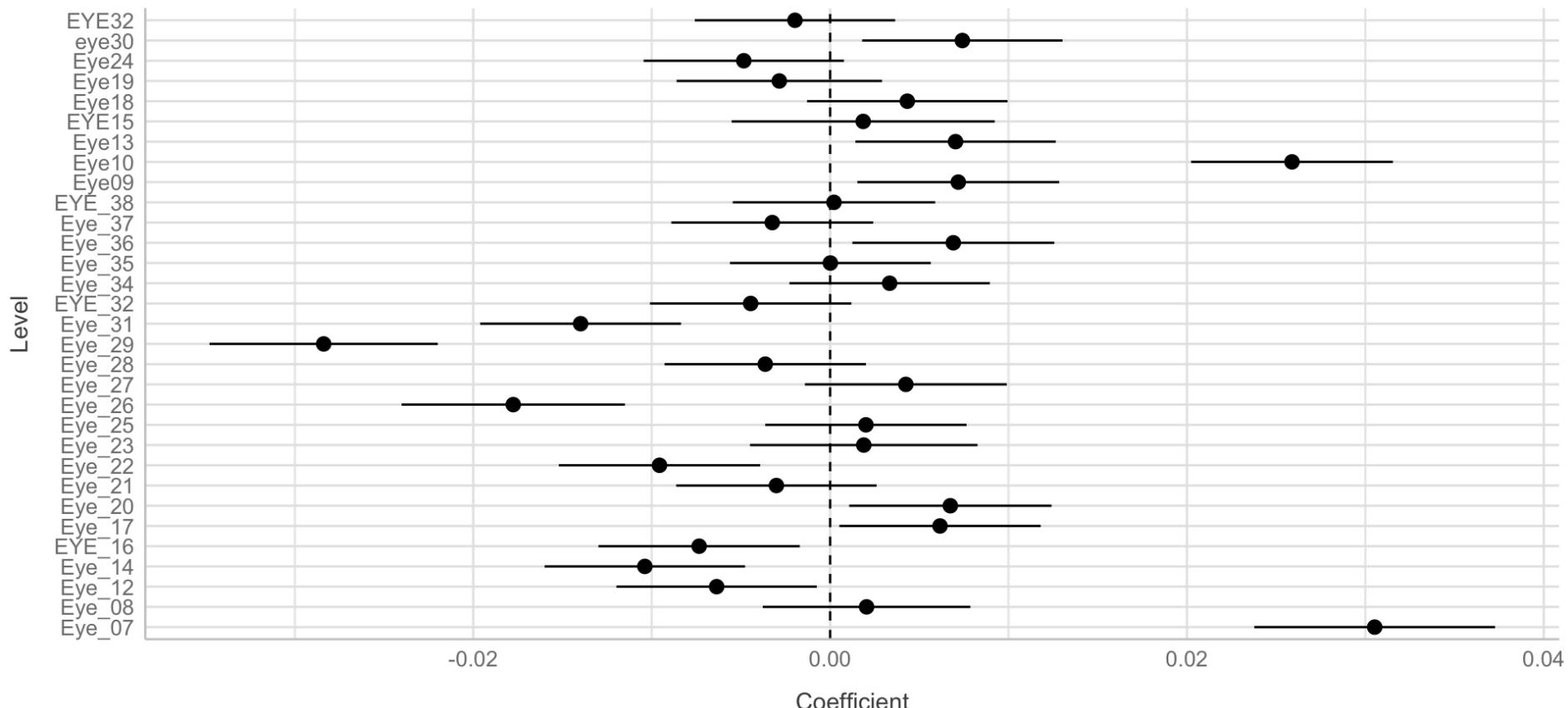
effect	group	term	estimate	std.error	statistic	df	p.value
fixed		(Intercept)	0.00369	0.0022	1.68	33.6	0.102
fixed		vocodedV6	0.00306	0.0011	2.79	5.58e+03	0.00526

- Default behavior is leave out \*p\*-values (Doug Bates doesn't like them)
  - Install `lmerTest` to include \*p\*-values

# Random effects

```
1 random <- estimate_grouplevel(inter_model)
2
3 plot(random) +
4 theme_lucid()
```

Group-level Scores



# Maximal model: Fixed effect random intercepts (subject) and slopes (vocoded) model

```
1 rand_model <- lmer(mean_pupil ~vocoded +(1+vocoded|subject), data = eye)
2
3 model_parameters(rand_model)
```

Parameter	Coefficient	SE	CI	CI_low	CI_high	t
(Intercept)	0.00364	0.00223	0.95	-0.000738	0.00802	1.63
vocodedV6	0.00312	0.00145	0.95	0.000275	0.00597	2.15
SD (Intercept)	0.0117		0.95			
SD (vocodedV6)	0.00531		0.95			
Cor	-0.195		0.95			
(Intercept~vocodedV6)						
SD (Observations)	0.0409		0.95			

# Maximal models

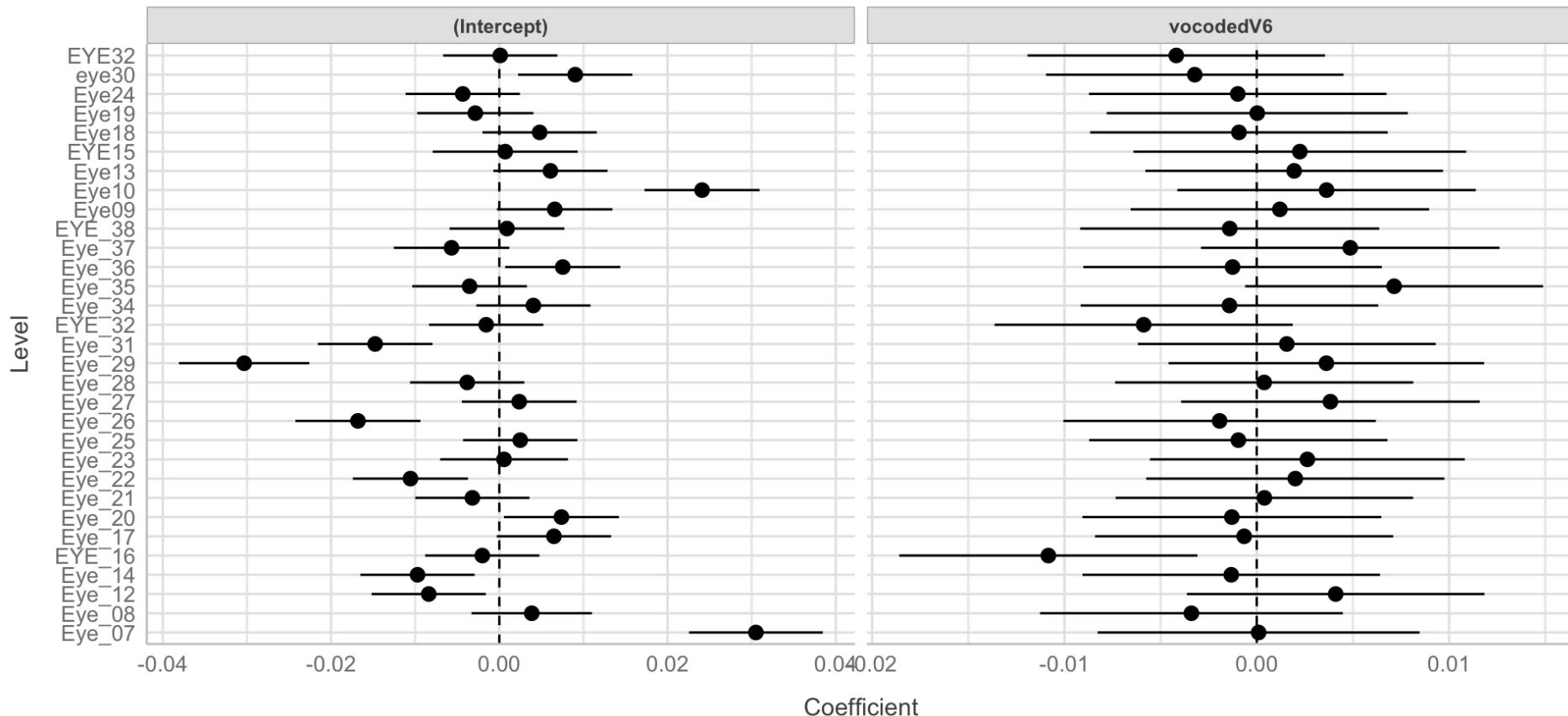
- Keep it maximal<sup>1</sup>
  - Whatever can vary, should vary
    - include random slopes only if it is a within cluster manipulation
  - Only when there is convergence issues should you remove terms
- Decreases Type 1 error

1. 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*, 61(3), 10.1016/j.jml.2012.11.001.  
<https://doi.org/10.1016/j.jml.2012.11.001>

# Random effects

```
1 random <- estimate_grouplevel(rand_model)
2
3 plot(random) +
4 theme_lucid()
```

Group-level Scores



# LRT

- For more complex models, use LRT chi-square (drop-in deviance test)
  - Can be interpreted as main effects and interactions

```
1 anova(rand_model) %>%
2   knitr::kable(., digits=3)
```

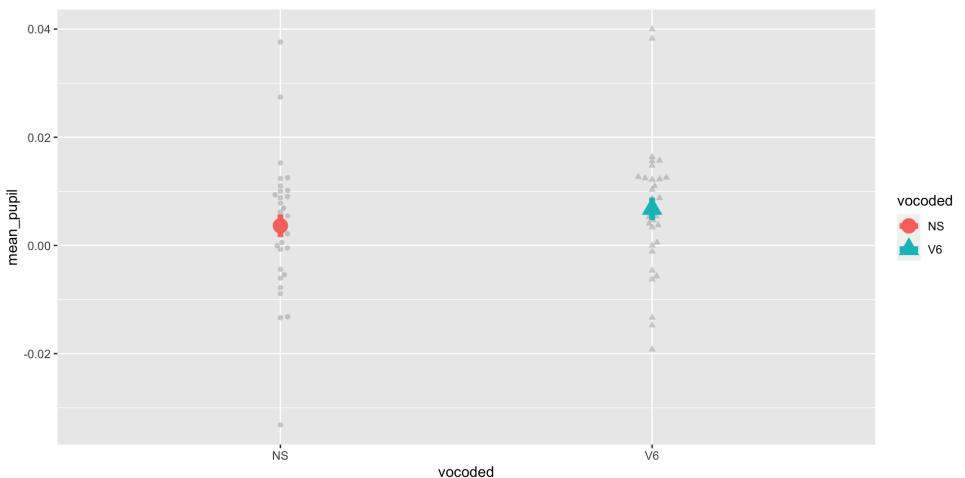
	Sum Sq	Mean Sq	NumDF	DenDF	F value	Pr(>F)
vocoded	0.008	0.008	1	30.472	4.622	0.04

# Afex

```
1 library(afex)
2
3 m <- mixed(mean_pupil ~ 1 + vocoded
4
5 nice(m)
```

Effect	df	Chisq	p.value
vocoded	1	4.47 *	.034

```
1 #| fig-align: "center"
2 #
3 p2 <- afex_plot(m, x = "vocoded",
4                   mapping = c("color",
5                   error = "within",
6                   point_arg = list(size =
7                   error_arg = list(lin
```



# Using emmeans

- Get factor means and contrasts

```
1 library(emmeans)
2
3 emmeans(rand_model, specs = "vocoded") # grabs means for each level of modality
```

	vocoded	emmean	SE	df	asymp.LCL	asymp.UCL
NS	0.00364	0.00223	Inf	-0.000737	0.00802	
V6	0.00677	0.00226	Inf	0.002334	0.01120	

Degrees-of-freedom method: asymptotic  
Confidence level used: 0.95

```
1 emmeans(rand_model, specs = "vocoded") %>%
2   pairs() # use this to get pairwise comparisons between levels of factors
```

	contrast	estimate	SE	df	z.ratio	p.value
NS - V6	-0.00312	0.00145	Inf	-2.150	0.0316	

Degrees-of-freedom method: asymptotic

# Effect Size

- Highly debated
  - Report Pseudo- $R^2$  for marginal (fixed) and conditional model (random) parts
  - Transform f to  $\eta_p^2$  (*when using afex::mixed*)

```
1 #easystats
2 r2(rand_model)
3 #Generate R-squared measures for that model:
4
5 model_performance(rand_model)
```

# Write-up

```
1 report::report(rand_model)
```

We fitted a linear mixed model (estimated using REML and nloptwrap optimizer) to predict mean\_pupil with vocoded (formula: mean\_pupil ~ vocoded). The model included vocoded as random effects (formula: ~1 + vocoded | subject). The model's total explanatory power is weak (conditional R<sup>2</sup> = 0.08) and the part related to the fixed effects alone (marginal R<sup>2</sup>) is of 1.34e-03. The model's intercept, corresponding to vocoded = NS, is at 3.64e-03 (95% CI [-7.38e-04, 8.02e-03], t(5603) = 1.63, p = 0.103). Within this model:

- The effect of vocoded [V6] is statistically significant and positive (beta = 3.12e-03, 95% CI [2.75e-04, 5.97e-03], t(5603) = 2.15, p = 0.032; Std. beta = 0.07, 95% CI [6.49e-03, 0.14])

Standardized parameters were obtained by fitting the model on a standardized version of the dataset. 95% Confidence Intervals (CIs) and p-values were computed using a Wald t-distribution approximation.

# Model fitting: ML or REML?

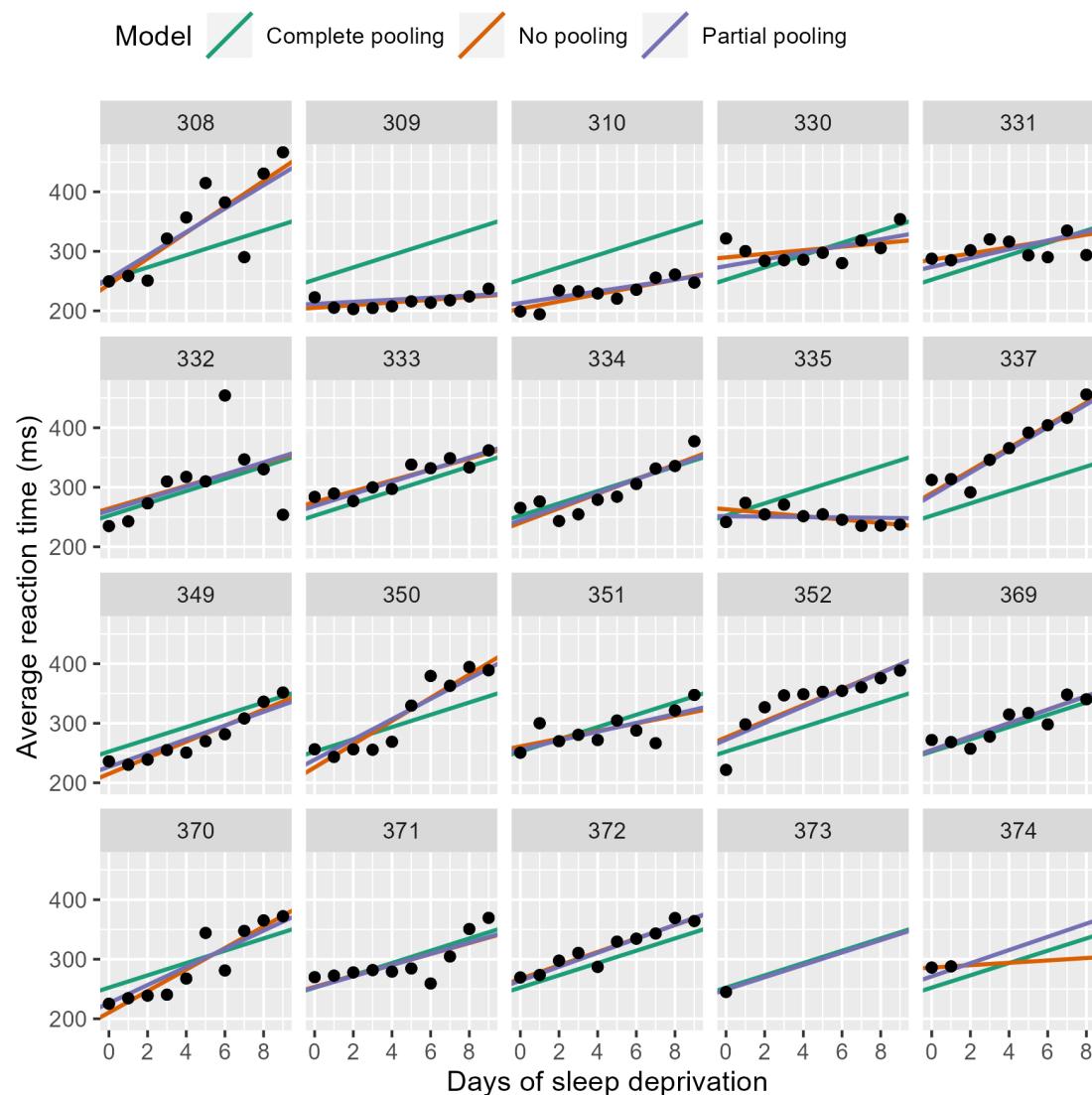
- Research has not determined one method absolutely superior to the other
- **REML** (`REML = TRUE`; default in `lmer`) is preferable when:
  - The number of parameters is large or primary, or
  - 
  - Primary objective is to obtain estimates of the model parameters
- **ML** (`REML = FALSE`) must be used if you want to compare nested fixed effects models using a likelihood ratio test (e.g., a drop-in-deviance test)
- For REML, the goodness-of-fit and likelihood ratio tests can only be used to draw conclusions about variance components

# Generalized linear mixed models

- We can fit most of the models we talked about this semester as a multilevel model

```
1 # poisson  
2 # binomial  
3 # negative binomial  
4  
5 glmer(family="binomial")  
6 glmer(family="poisson")  
7 glmer.nb()
```

# Shrinkage (if there is time)



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

# Centering (if there is time)

