

# The Art of Modeling

## An Introduction to Linear Mixed Effects Models with R

Klaus Frierer

Methods Consultant

Max Planck Institute for Empirical Aesthetics  
Frankfurt am Main

March 19, 2025

# Agenda

---

- Motivation: Who is Elmar?
- A Bit of Linear Algebra Basics
- A Bit of Theory
- The Art of Modeling
  - Model Building
  - Model Checking
  - Model Interpretation
  - Trouble Shooting
- Outlook: Generalized LMMs
- Hands on: Analyze your own data
- Wrap-up and Debrief

# Who is Elmar?

# Who is Elmar?

---

- In 2015, Elke Lange approached me with an idea for a cool study.
- Can perception of emotion expression in music be predicted by audio and other features?
- Setup: 20 audio engineers rated 60 musical excerpts ( $\approx 60$  s) on 22 variables (emotions, audio quality, modal analogues).
- Extracted 86 audio features with the MIRToolbox.

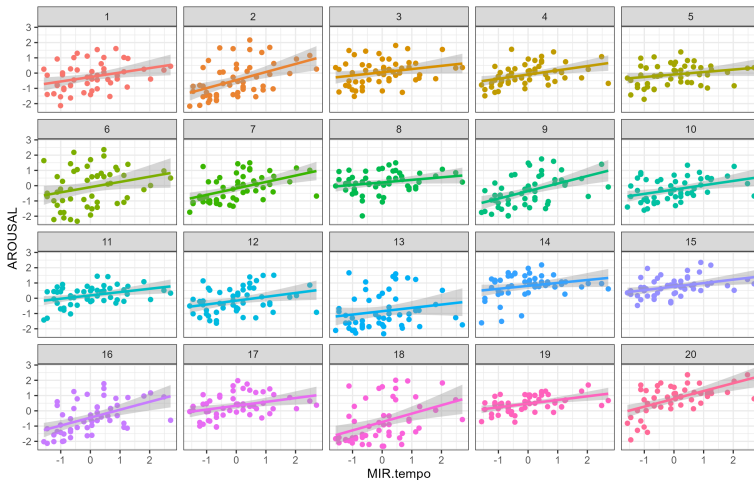
# Who is Elmar?

---

- We reduced the six emotion variables to two variables AROUSAL and VALENCE using factor analysis.
- We found that tempo correlates moderately with AROUSAL,  $r = .31$ .
- But is this true for all participants?
- Let's see!

# Who is Elmar?

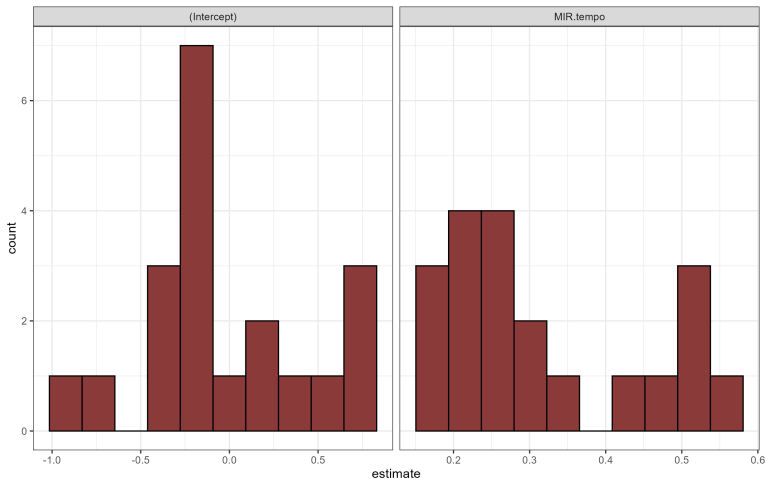
The correlations by participants range from  $r = .19$  to  $r = .54$  (mean  $\bar{r} = .37$ ).



# Who is Elmar?

---

Our first random effects!



# Why Linear Mixed Effect Models?

---

- As the example shows, on many occasions, a constant effect of something on an individual seems not reasonable.
- Fixed effects are average effects.
- Are individual differences often more interesting?
- Specifics of experiments, such as stimuli, often simply a nuisance, from which one wants to abstract to achieve a better generalization.
- LMMs allow to model all this.



# Why Linear Mixed Effect Models?

---

- The first reason to use LMMs was to deal with repeated measurement.
- When using a common (fully) crossed design, repeated measurement ANOVA is not the best option.
- Averaging loses power.
- LMMs allow modeling arbitrary complex models, when the independence assumption is violated due to repetitions or hierarchical nesting.
- Further benefits: Deals smoothly with NA, provide more detailed information.
- LMM is a very general model, unifying, t-tests, ANOVAS, rmANOVAS, ANCOVAS, and multiple linear regressions (and even  $\chi^2$  test and non-parametric tests based on ranks).

# Why Linear Mixed Models?

---

## Hands On

- Team up with a partner, if you like.
- Start R, load the accompanying RStudio project, install missing packages.
- Git for project: [https://github.com/klausfrieler/lmm\\_workshop](https://github.com/klausfrieler/lmm_workshop).
- Source the file `setup.R`.
- Run `setup_workspace()`.
- As a warm-up, create a scatter plot like that before with `MIR.pulse_clarity` and `VALENCE`. The data frame is `mer` in the workspace. Bonus question: how many of the individual correlations are significant compared to expectation?

## **A bit of basic: Linear Algebra**

## A bit of basics: Linear Algebra

---

A (column) vector of dimension  $N$  is a set of  $N$  numbers:

$$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

Also written as boldface  $\mathbf{x}$ , or with components  $x_i$ .

A row vector is the transpose  $\mathbf{x}^T = (x_1, x_2, \dots, x_N)$ .

Vectors of same dimension can be added component-wise and multiplied by numbers:

$$\mathbf{z} = \alpha \mathbf{x} + \beta \mathbf{y},$$

with components

$$z_i = \alpha x_i + \beta y_i$$

## A bit of basics: Linear Algebra

---

The scalar product is defined as

$$\vec{x} \cdot \vec{y} = \mathbf{x}^T \mathbf{y} = \sum_{i=1}^N x_i y_i$$

The length of a vector is

$$\|\mathbf{x}\| = \sqrt{\sum_i x_i^2}$$

and the (Euclidian) distance between two vectors

$$d(x, y) = \|\mathbf{x} - \mathbf{y}\| = \sqrt{\sum_i (x_i - y_i)^2}$$

## A bit of basics: Linear Algebra

---

The (sample) standard deviation of a data vector is proportional to the distance to the vector of mean values.

$$\hat{\sigma}(\mathbf{x}) = \frac{1}{\sqrt{N-1}} d(\mathbf{x}, \bar{\mathbf{x}}) = \frac{1}{\sqrt{N-1}} \|\mathbf{x} - \bar{\mathbf{x}}\|$$

For two data vectors of dimension  $N$ ,  $\mathbf{x}$ ,  $\mathbf{y}$ , the Pearson correlation  $r_{xy}$  is defined by the scalar product of the z-transformed vectors as:

$$r_{xy} = \frac{1}{N-1} \mathbf{z}(\mathbf{x}) \cdot \mathbf{z}(\mathbf{y}) = \frac{\mathbf{x} - \bar{\mathbf{x}}}{\|\mathbf{x} - \bar{\mathbf{x}}\|} \cdot \frac{\mathbf{y} - \bar{\mathbf{y}}}{\|\mathbf{y} - \bar{\mathbf{y}}\|}$$

## A bit of basics: Linear Algebra

---

A **matrix**  $\mathbf{X}$  of dimension  $N \times M$  is a rectangular scheme of numbers, with  $N$  rows and  $M$  columns with elements  $x_{ij}$ .

$$\mathbf{X} = \begin{pmatrix} x_{11} & \dots & x_{1M} \\ \vdots & \ddots & \vdots \\ x_{N1} & \dots & x_{NM} \end{pmatrix}$$

A matrix can also be considered a  $N$ -dimensional row-vector of  $M$ -dimensional column vectors or vice versa.

Matrices of same dimensions can be added and multiplied with a number just like vectors. The transposed  $\mathbf{X}^T$  of a matrix  $\mathbf{X}$  switches row and columns, i. e.,  $x_{ji}^T = x_{ij}$ .

A  $N$ -dimensional vector  $\mathbf{x}$  can be regarded as a  $N \times 1$  matrix.  $\mathbf{x}^T$  is then a  $1 \times N$  vector.

## A bit of basics: Linear Algebra

---

Matrices **A** and **B** can be multiplied  $\mathbf{C} = \mathbf{AB}$ , if the number of columns of **A** is the same as the number of rows in **B**. The elements of the result are defined as

$$c_{ij} = \sum_{k=1}^M a_{ik} b_{kj}$$

The elements of **C** are the scalar products of the  $i$ th row vector of **A** with the  $j$ th column vector of **B**

$$c_{ij} = \mathbf{a}_i^T \mathbf{b}_j = \vec{a}_i \cdot \vec{b}_j$$

We have the following computation rules:

$$\mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C}, \quad \mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC}$$

Attention: matrix multiplication is, in general, not commutative :  $\mathbf{AB} \neq \mathbf{BA}$ .



## A bit of basics: Linear Algebra

---

The quadratic ( $N \times N$ ) matrices are special as they can have an inverse matrix, i.e., if a matrix  $\mathbf{A}$  there exists a matrix  $\mathbf{A}^{-1}$  such that

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}_N,$$

with the identity matrix  $\mathbf{I}_N$  which has 1's on the diagonal and 0's everywhere else.

The inverse exists, if  $A$  has full rank, i. e., no column or row vector is a linear combination of the other vectors or none of them is zero. This is equivalent to  $\det \mathbf{A} \neq 0$ .

Example of a non-invertible matrix:

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} a & b \\ 0 & 0 \end{pmatrix} \neq \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

## A bit of basics: Linear Algebra

---

### Exercise

Let  $\mathbf{A}$  be the following matrix:

$$\mathbf{A} = \begin{pmatrix} 0 & 2 & -1 \\ 2 & -1 & 2 \\ -1 & 0 & 0 \end{pmatrix}$$

and  $\mathbf{y}^T = (1, -1, -1)$ .

Solve

$$(\mathbf{A}^T \mathbf{A})\mathbf{x} = \mathbf{y}$$

for  $\mathbf{x}$ .

Use matrix operations in R, to create a matrix: `matrix()`, `t()` for transposition, `%*%` for matrix multiplication, and `solve()` for matrix inversion.

## **A bit of theory: Linear Regression**

## A bit of theory: Linear regression

---

A linear regression model for one variable and  $N$  observations can be written as

$$y_i = \gamma + \beta x_i + \epsilon_i$$

with

- $y_i$  the outcome for participant  $i$ ,  $1 \leq i \leq N$ .
- $\gamma$  the overall intercept
- $\beta$  the fixed effect coefficient,
- $x_i$  the value of the predictor variable  $x$ .
- $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$  the normal distributed residual error with mean 0 and variance  $\sigma^2$ .

## A bit of theory: Linear regression

---

This can be extended to  $p$  predictors, each having their own beta coefficient.

$$y_i = \sum_k^p \beta_k x_{ik} + \epsilon$$

and be compactly written with vectors and matrices:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \epsilon$$

where  $\mathbf{X}$  is the model-matrix, containing the measurements for each participant in rows, and variables are contained in columns.

The global intercept is represented here by adding a 0th-column of ones to  $\mathbf{X}$  and a 0th-component to  $\boldsymbol{\beta}$ , i. e.,  $\gamma = \beta_0$ .

Categorical predictors are encoded with numerical dummy variables, e. g., one-hot encoding ('contrast').  $K$  categories add  $K - 1$  dummy variables to  $\mathbf{X}$ . This covers ANOVA and ANCOVA.

## A bit of theory: Linear regression

---

The  $\beta$ -coefficients can be estimated using various methods, e. g., ordinary least squares or maximum likelihood estimation.

In ordinary least squares, one seeks to find estimate  $\hat{\mathbf{y}}$  that minimize the error to the observed  $y$ ,

$$\sum_{i=1}^N (y_i - \hat{y}_i)^2 = \sum_{i=1}^N \left( \sum_{k=1}^p x_{ik} \beta_k - y_i \right)^2 = f(\beta_0, \beta_2, \dots, \beta_p) = \min$$

This is a function of the unknown variables  $\beta_i$ , and a minimum can be found by setting partial derivations with respect to all variables to 0.

## A bit of theory: Linear regression

---

Calculating the partial derivative with respect to  $\beta_j$  yields

$$\begin{aligned}\frac{\partial}{\partial \beta_k} \sum_{i=1}^N \left( \sum_{k=1}^p x_{ik} \beta_k - y_i \right)^2 &= 0 \\ \sum_{i=1}^N 2 \left( \sum_{k=1}^p x_{ik} \beta_k - y_i \right) x_{ij} &= 0 \\ \sum_{i=1}^N \sum_{k=1}^p x_{ij} x_{ik} \beta_k - \sum_{i=1}^N y_i x_{ij} &= 0\end{aligned}$$

## A bit of theory: Linear regression

---

Using matrix notation this implies

$$\mathbf{X}^T \mathbf{X} \beta = \mathbf{X}^T \mathbf{y},$$

and, finally,

$$\beta = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

This solution only exists, if the model matrix  $\mathbf{X}$  is invertible, which means that it should have full rank. That's why multicollinearity is a problem.

The maximum likelihood method has the same result, as it uses the logarithm of a normal distribution, which has a similar quadratic term.



## A bit of theory: Linear regression

---

### Exercise

Use the data set `scenario2` and calculate the beta coefficients using the formula above and compare these those from the linear model

```
coef(lm(y ~x + z, data = scenario2))
```

## A bit of theory: Hierarchical Linear regression

---

Now we incorporate the idea of varying beta coefficient. We use hierarchical models, introducing a linear equation for the betas. Consider

$$y_{ij} = \gamma + x_{ij}\beta_j + \epsilon_{ij}$$

where introduced a clustering index  $1 \leq j \leq q$  for  $q$  clusters, indicating participants or stimuli etc.  $y_{ij}$  is the  $j$ th value in cluster  $j$ .

The random effects assumption is that the intercepts and slopes are a normal distribution over the clusters, with mean values  $\gamma$  and  $\beta$  of the fixed effects. Hence, we can write for each cluster

$$\begin{aligned}\gamma_j &= \gamma + u_j^\gamma \\ \beta_j &= \beta + u_j^\beta\end{aligned}$$

## A bit of theory: Hierarchical Linear regression

---

Inserting this into the original equation yields

$$\begin{aligned}y_{ij} &= \gamma + \gamma_k + x_{ij}(\beta + \beta_j) + \epsilon_{ij} \\ &= (\gamma + x_{ij}\beta) + (\gamma_j + x_{ij}\beta_j) + \epsilon_{ij}\end{aligned}$$

Again using compact matrix notation, this yields to most commonly used formulation for a random effects model

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{Z}u + \epsilon$$

## A bit of theory: Hierarchical Linear Regression

---

- In contrast to simple linear regression, no closed solution is available.
- Iterative method based maximum likelihood estimation (ML) or restricted maximum likelihood estimation (REML, default in `lmer`) needed.
- Algorithm might not find always a solution (see Trouble Shooting).
- The fixed coefficients are the same as when using simple linear regression.
- **It is all about the standard errors!** (= Type I errors).
- In a way, there are only random effects, where the mean of the random effects are the fixed effects.
- Model only identifiable under certain conditions, measurements need to allow estimation of the RE.
- Sometimes “boundary fits” are produced, which often indicates misspecification (see Trouble Shooting).

# Simulation is the way to wisdom

---

As a preparation for the following, let's create some artificial data.

## Exercise

- Scenario:  $N$  participants rate a set of  $M$  musical stimuli in two conditions/Variants, 'audio-only' (AO) and 'audio-visual' (AV) with respect to liking on a 5-point Likert scale (fully crossed design).
- **Task:** Write a function with parameters `n_raters`, `n_stimuli`, `error` (residual error) and `beta` (effect of condition) that generates liking ratings. (Approximate the 5-point Likert scale with a normal distribution with mean 3.)
- Tips: `expand_grid` or `crossing` create all combinations, `rnorm` produces normal-distributed random numbers for a given mean and standard deviation.
- **Bonus Task:** Set a parametrizable portion of values to NA. Add participant or stimulus-dependent covariates that correlate with liking to a certain (parametrizable) extent  $r$ .

# **The Art of Modeling**

# Roadmap

---

1. Data exploration
2. Model building
  - 2.1 Identifying variable types, random effects, fixed effects, covariates.
  - 2.2 Model optimization by way of model comparison
  - 2.3 Fixing converge and boundary fit issues
3. Model checking: are assumptions met?
4. Model interpretation: significance testing, marginal effects

# Data exploration

---

- Know you data!
- First step should be always data exploration.
- Particularly with complex data.
- Histograms and scatterplots of all variables (e. g., `psych::pairs.panels()`, `GGally::ggpairs().`)
- Data sanity: Missing values, outliers, 'pathological' distributions, unbalanced cluster/cell sizes etc.
- Altogether now: Know you data!



## Model building: `lmer`

---

- We will use `lmer` from the `lme4` package.
- In fact, we use the wrapper package `lmerTest`, which adds p-values and provides other useful functions.
- There are other options like `nlme` (old) or `glmmTB` (if you need special stuff).
- `lmer` uses (extended) formula syntax or R.
- Sample call `lmerTest::lmer(liking ~ condition + (1 + condition|rater), data = simulate_lmm())`

## Model building: `lmer` Formulas

---

A `lmer()` formula has the form

$$DV \sim FE + RE$$

The dependent variable is left of the tilde, then fixed and random effects.

Fixed effects contain variables, interactions, and intercepts.

<code>1 + IV1 + IV2 + ...</code>	Intercept + independent variable(s)
<code>0 + IV + ..., -1 + IV + ...</code>	Remove Intercept
<code>IV1 : IV2</code>	Interaction of two variables
<code>IV1 * IV2</code>	Same as <code>IV1 + IV2 + IV1 : IV2</code>

The variables correspond to column names in the `data.frame`.

## Model building: `lmer` Formulas

---

A `lmer()` formula has the form Random effects have the form

`(FE | CV)` or `(FE || CV)`

FE is a fixed-effect type formula, defined over a cluster variable CV.

A double bar `||` forces the random effects to be uncorrelated instead of being correlated (default, single bar `|`).

Random intercept only is written as `(1|CV)`.

The cluster variables can have nesting using the interaction operator `:` (colon).

The combination `(1|school) + (1|school:class)` can be abbreviated to `(1|school/class)`.

# Model building

---

- Generally, all units with repeated measurement need to be included as random effects.
- Variables of Interest (VOI) should be identified, i. e., fixed effects as well as additional covariates.
- Fixed effects should be included as random slopes for the random effects if they apply (i. e., if they vary across the cluster).
- Test necessity of random effects with the function `ranova()` from the `lmerTest` package.
- Model comparison is based on likelihood ratio tests
- For nested models, the difference of deviances ( $= -2\log Lik$ ) is approximately  $\chi^2$ -distributed.
- AIC (and BIC) lower for better fitting models (no sig-test).

# Model building

---

## Exercise

- Simulate data with 10 raters or 10 items (strong/weak random effect with/without slope) using `simulate_lmm()`
- Test if inclusion of random effects (over `rater` and `item`) improves model fit significantly
- Tip: Establish that the simplest random effects is better using `lmerTest::ranova()`, then compare different models with `anova()`
- Bonus: repeated for different effect sizes and `rater` and `item` numbers, what is the difference?

# Model checking

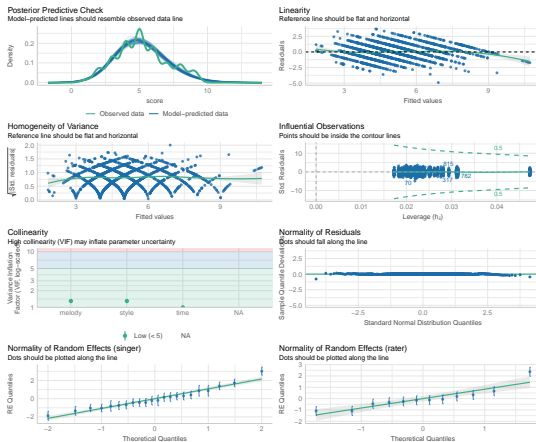
---

Check the common assumptions using `performance::check_model()`

- Normality of residuals
- Linearity
- Homoscedasticity
- Influential observations
- Variance Inflation
- Normality of random effects

# Model checking

Model check with `performance::check_model()` for the Covox data set with intercepts-only random effects of singer and rater.



# Model interpretation

## lmerTest::lmer() Output

```
Linear mixed model fit by REML.  
t-tests use Satterthwaite's method Formula:  
liking ~ condition + (1 + condition | rater) + (1 + condition | item)
```

```
REML criterion at convergence: 379.9
```

```
Scaled residuals:
```

Min	1Q	Median	3Q	Max
-1.99790	-0.56716	-0.02668	0.58813	2.95504

```
Random effects:
```

Groups	Name	Variance	Std.Dev.	Corr
rater	(Intercept)	0.1038	0.3222	
	conditionAV	0.2291	0.4787	0.57
item	(Intercept)	0.0919	0.3032	
	conditionAV	0.3157	0.5619	0.01
Residual		0.2651	0.5149	

```
Number of obs: 200, groups: rater, 10; item, 10
```

```
Fixed effects:
```

	Estimate	Std. Error	df	t value	Pr(> t )
(Intercept)	2.5529	0.1491	14.3158	17.13	6.16e-11 ***
conditionAV	0.7701	0.2445	14.9130	3.15	0.00665 **

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



# Model interpretation

---

## lmerTest::lmer() Output General Info

Linear mixed model fit by REML.

t-tests use Satterthwaite's method Formula:

liking ~ condition + (1 + condition | rater) + (1 + condition | item)

REML criterion at convergence: 379.9

Scaled residuals:

Min	1Q	Median	3Q	Max
-1.99790	-0.56716	-0.02668	0.58813	2.95504

## Model interpretation

### lmerTest::lmer() Output Random Effects

Random effects:

Groups	Name	Variance	Std.Dev.	Corr
rater	(Intercept)	0.1038	0.3222	
	conditionAV	0.2291	0.4787	0.57
item	(Intercept)	0.0919	0.3032	
	conditionAV	0.3157	0.5619	0.01
Residual		0.2651	0.5149	

Number of obs: 200, groups: rater, 10; item, 10

True values are .5 for all intercepts and slopes standard deviations and for intercept- slope correlations. Residual error is  $\epsilon = .5$ .

# Model interpretation

## lmerTest::lmer() Fixed Effects

Fixed effects:

	Estimate	Std. Error	df	t value	Pr(> t )	
(Intercept)	2.5529	0.1491	14.3158	17.13	6.16e-11	***
condition2	0.7701	0.2445	14.9130	3.15	0.00665	**

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:

	(Intr)
conditionAV	0.177

True values are Intercept = 3.0,  $\beta_{\text{cond}} = 1.0$ .  $\Rightarrow$  Very small sample size!

# Model interpretation

Test for overall significance using `anova()`. Compute model fit as marginal  $R_m^2$ , variance explained by fixed effects, and conditional  $R_c^2$ , variance explained by full model. Usually  $R_c^2 \gg R_m^2$ .

## `lmerTest::lmer()` Further metrics

```
anova(mod_r10i10_sws)

Type III Analysis of Variance Table with Satterthwaite's method
      Sum Sq Mean Sq NumDF  DenDF  F value    Pr(>F)
condition    2.63    2.63     1 14.913    9.92 0.006651 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

performance::r2(mod_r10i10_sws)

# R2 for Mixed Models

Conditional R2: 0.728
Marginal R2: 0.153
```

Sometimes `MuMIn::r.squaredGLMM()` is needed. Try also `sjPlot::tab_model()`.

## Model interpretation: A Note on Significances

---

- Even though the main reason to use LMMs is getting the standard errors 'right', there is a bit of a problem with p-values for coefficients.
- The `lme4` deliberately does not provide p-value estimates for the fixed effect coefficients. Problem is to get the degrees of freedom right.
- Alternative methods to establish significance are model comparison, parametric bootstrap, profiled confidence intervals, MCMC sampling, sandwich estimators (package `sandwich`, or simply looking at  $t > 2$ , if sample size is large enough.
- The `lmerTest` package adds p-values for betas using Satterthwaite's degrees of freedom method or the Kenward-Rogers method, both are not beyond discussion.

## Model interpretation: Extracting stuff

---

- `coef()`: Retrieves all coefficients as list of data frames per random effect. Includes fixed effects. Data frames of same size of data, except excluded cases.
- `fixef()`: Retrieves fixed effect coefficients as vector.
- `lme4::ranef()`: Retrieves random effects as data frame.
- `AIC()`, `BIC()`, `logLik()`, `deviance(REML = FALSE)` for global model fit parameters
- `vcov()/VarCorr()` for fixed/random effects variance-covariance matrices
- `predict()`, `residuals()` Get predictions/residuals.
- `confint()`, get confidence intervals for all random and fixed effects coefficients..
- `broom.mixed::tidy()`, `broom.mixed::glance()` retrieves coefficients / model fit as tibble, useful for aggregation

# Model interpretation

---

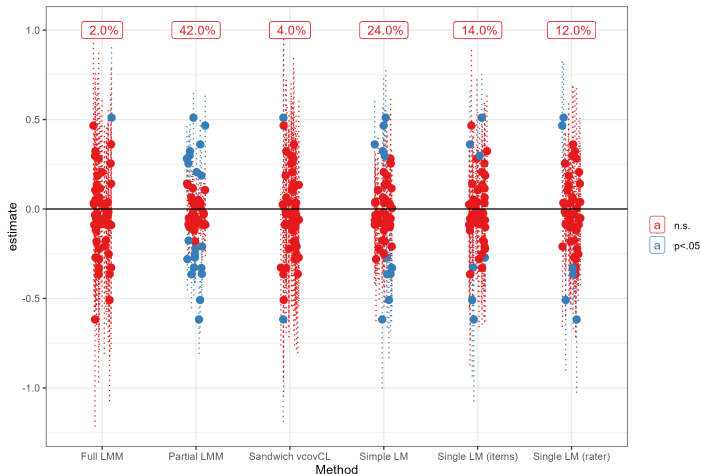
Work on one or more of these problems.

## Exercise

- Simulate  $N$  data sets with  $M$  raters and  $K$  items using `simulate_lmm()`, aggregate the fixed effects coefficients and calculate mean and sd, plot histograms.
- Simulate a data set of you liking with `simulate_lmm()`. Fit a simple model `fit <- lmer(liking ~ condition + (1|rater))`. Compare the standard deviations of individual random effect estimates with those given by the summary (`summary(fit)$varc` or `VarCorr(fit)`)? What is going on?
- Simulate a data set with zero fixed effect `simulate_lmm(fixef = 0)`. Fit three models with `condition` as fixed effects and either intercept + slope, intercept-only, simple linear regression. Compare the results (using, e.g., `sjPlot::plot_models()`).

# Model interpretation: Type I and Type II errors

Estimates and Type I errors for null effect, small sample size.

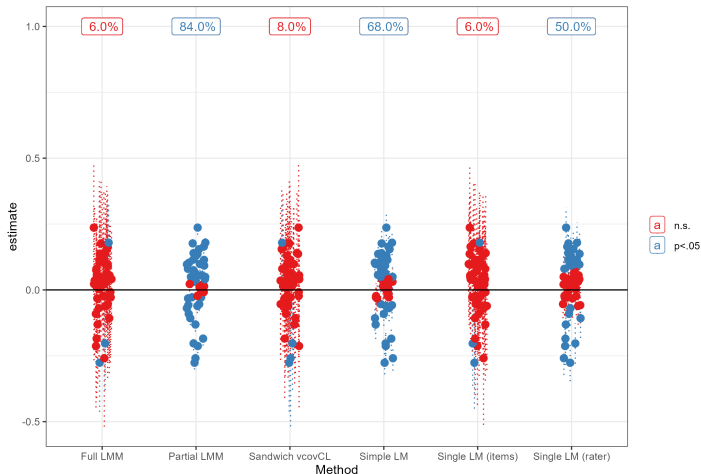


$$N_{\text{sim}} = 50, N_{\text{raters}} = 10, N_{\text{items}} = 10, \beta_{\text{cond}} = 0.0$$



# Model interpretation: Type I and Type II errors

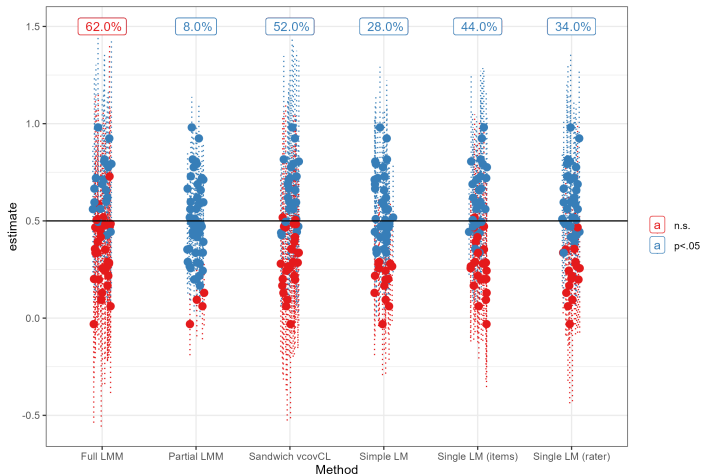
Estimates and Type I errors for null effect, large sample size.



$N_{\text{sim}} = 50$ ,  $N_{\text{raters}} = 200$ ,  $N_{\text{items}} = 20$ ,  $\beta_{\text{cond}} = 0.0$

# Model interpretation: Type I and Type II errors

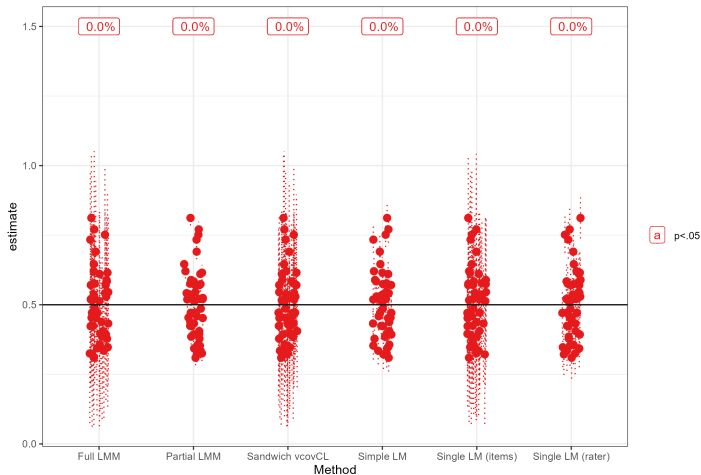
Estimates and Type II errors for weak effect, small sample size.



$$N_{\text{sim}} = 50, N_{\text{raters}} = 10, N_{\text{items}} = 10, \beta_{\text{cond}} = 0.5$$

# Model interpretation: Type I and Type II errors

Estimates and Type II errors for weak effect, large sample size.



$$N_{\text{sim}} = 50, N_{\text{raters}} = 200, N_{\text{items}} = 20, \beta_{\text{cond}} = 0.5$$

## Model interpretation: Marginal effects

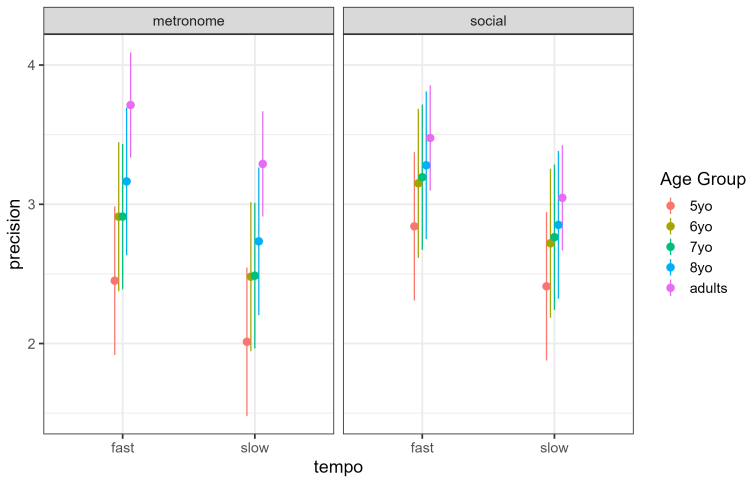
---

- For post-hoc testing and visualization you could use two packages: `emmeans` (older) and `marginalEffects` (modern and fancy).
- These allow to aggregate model predictions across conditions and mean covariates etc. taking care of standard errors etc.

### Example

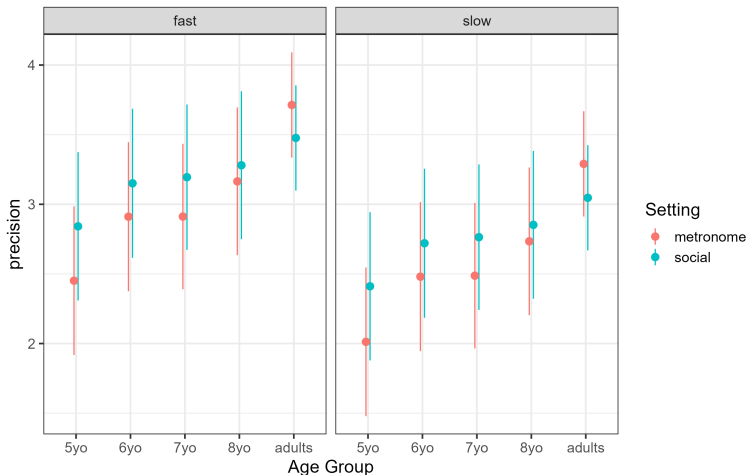
We use the `kid_beats` data set and fit the following model `precision ~ age_group * setting + tempo + beat_prod + beat_perc + 1 | experimenter) + (1 | p_id)`. Participants from five different `age_groups` tapped along to a metronome or a human (`setting`) in two different `tempo` conditions (fast: 400 ms, slow: 600 ms)) and the `precision` of the tapping was measured (logarith of standard deviation of IOIs). For each participant, there were also two beat-related covariates `beat_prod` and `beat_perc`.

# Model interpretation: Marginal effects



```
marginalEffects::plotPredictions(kid.beats.model, by = c("tempo", "age_group", "setting"))
```

# Model interpretation: Marginal effects



```
marginalEffects::plotPredictions(kidBeatsModel, by = c("age-group", "setting", "tempo"))
```

# Trouble Shooting

---

- Frequently, `lmer` does not converge or shows scary warning messages.
- Singular fit is actually a feature not a bug. It mostly means that some random effects are estimated to 0 or could not estimated at all, which means the model is overspecified. **Solution:** Inspect the random effects results, possibly simplify the model or decide to ignore.
- Model does not converge. This also mostly means that the model is overspecified or the data do not support the identification of the model or are otherwise peculiar. **Solution:** Try another optimizer first, or increase tolerance and iteration steps. If that does not work: simplify model.
- An often helpful optimizer is "bobyqa". Usage: `lmer(..., control = lmerControl(optimizer = "bobyqa"))`
- **Warning:** for large datasets and complicated random effects (random slopes!), the estimation can take a VERY long time. Setting `control = lmerControl(calc.deriv = FALSE)` can speed up things.

# Trouble Shooting

---

- Frequently, `lmer` does not converge or shows cryptic warning messages.
- Singular fit is actually a feature not a bug. It mostly means that some random effects are estimated to 0 or could not estimated at all, which means the model is overspecified. **Solution:** Inspect the random effects results, possibly simplify the model or decide to ignore.)
- Model does not converge. This also mostly means that the model is overspecified or the data do not support the identification of the model or are otherwise peculiar. **Solution:** Try another optimizer first, or increase tolerance and iteration steps. If that does not work: simplify model.
- An often helpful optimizer is "bobyqa". Usage: `lmer(..., control = lmerControl(optimizer = "bobyqa"))`
- **Warning:** for large datasets and complicated random effects (random slopes!), the estimation can take a VERY long time. Setting `control = lmerControl(calc.deriv = FALSE)` can speed up things.



## **Outlook and Wrap-Up**

## Outlook: Generalized LMMs

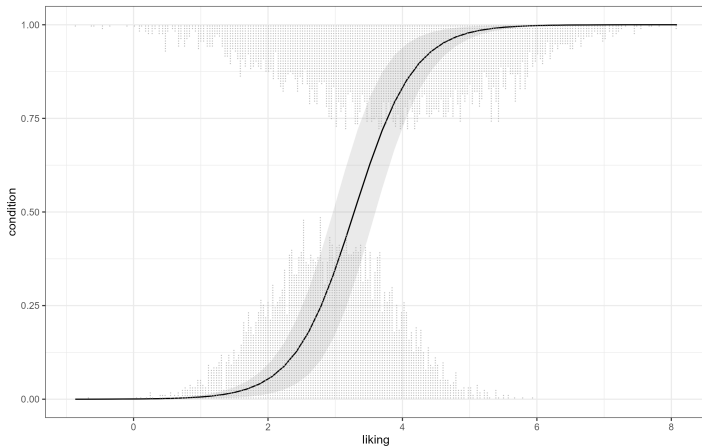
---

- Generalized Linear Models can be applied to binary (and other) data types, using the `lme4::glmer` function. Syntax is very similar but it needs specification of a link function.
- The link function transforms the outcome to something continuous that can be modeled using LMMs.
- Involves extra steps of transformation, prediction and interpretation might be a bit different, as well as model parameter
- Mixed effects logistic regression uses `family = binomial("logit")`, i.e.  
$$y_i \rightarrow \log \left( \frac{p_i}{1-p_i} \right).$$
- Mixed-effects ordinal regression with package `ordinal`, multinomial mixed regression with `mclogit::mblogit`.
- Bayesian mixed models (e.g. the `brms` package), also possible provides different numerical estimation method
- Generalized Additive Mixed Models (GAMMs) and much more ...

## Example: Logistic Regression

---

Predicting condition from liking with simulated data and strong random effects with slope (200 Rater, 20 Stimuli).



## Hands On: Your own models

---

### Try it yourself!

- Take some of your own data and run some models. Show us the results!
- Or explore the `mer`, `covox`, `kid_beats` or `scenario2` data sets in the repository.

## Wrap-up

---

- LMM are state-of-the-art statistic models, substituting most other methods.
- Need to be used in case of repeated measurements, particularly, the common fully crossed designs
- Mainly affects estimation of standard error, better inference.
- Comes with some complexities and subtleties, but this is basically true for all statistical models . . .
- Happy modeling!

# Debrief

---

- How did you like the workshop?
- Did you learn something?
- Are you eager to use linear mixed effect models now?
- What could be improved?
- Further comments and questions?

# The End