Linear mixed models in R
Day 4

JONAS WALTHER

Error messages during model reduction

When reducing a maximal model (or running models in general) you can encounter the following error message:

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
control$checkConv, :

## Model failed to converge with max|grad| = 0.609847 (tol = 0.002,
component 1)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
control$checkConv, : Model is nearly unidentifiable: very large
eigenvalue

##  - Rescale variables?
```

This might indicate a problem with your data set, usually a variable that is not normally distribut.

Common transformations

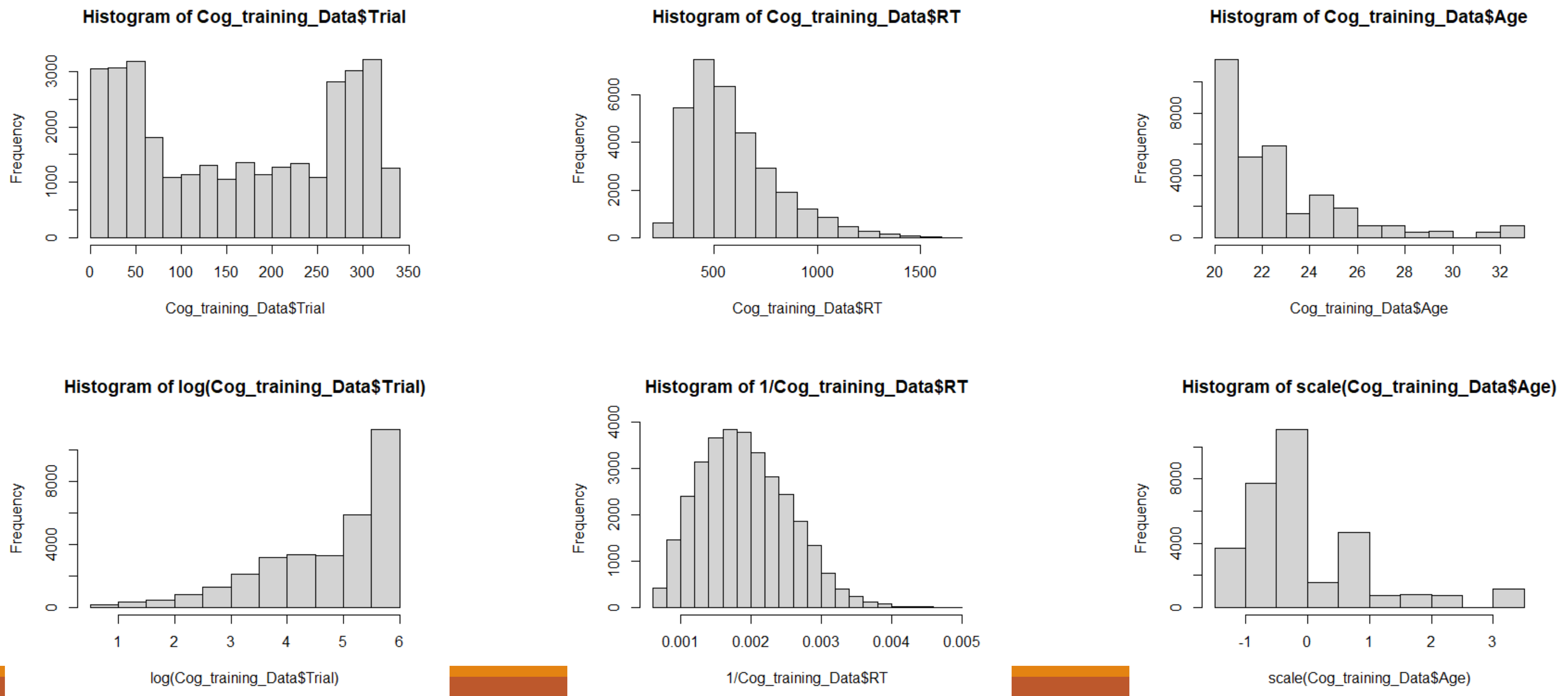
Often independent variables are also transformed for use in LMEs

Common transformations for independent variables are:

- Lograthmic
- Reciprocal
- Scaling

BUT much more important is the normal distribution of residuals (errors) and therefore the dependent variable (see Day 2)!

Common transformations



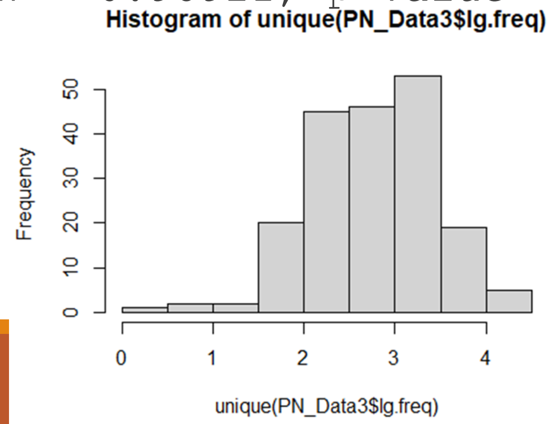
Shapiro-Wilk Test

Statistical test for checking if a variable is normally distributed

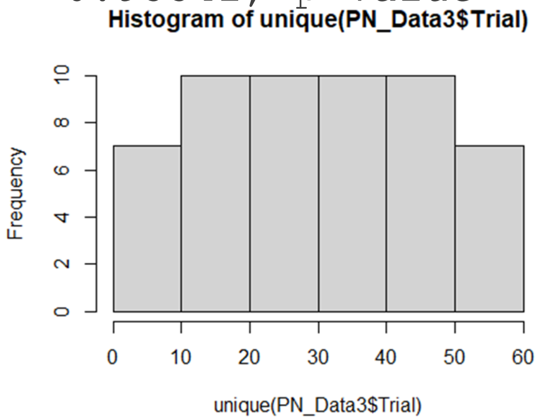
The sample size must be between 3 and 5,000

P-value below 0.05 indicates non-normally distributed data

```
shapiro.test(unique(PN_Data3$lg.freq))
##
##   Shapiro-Wilk normality test
## data:   unique(PN_Data3$lg.freq)
## W = 0.98921, p-value = 0.1534
```




```
shapiro.test(unique(PN_Data3$Trial))
##
##   Shapiro-Wilk normality test
## data:   unique(PN_Data3$Trial)
## W = 0.95542, p-value = 0.04305
```



Lo and Andrews (2015) To transform or not to transform: using generalized linear mixed models to analyse reaction time data

Transforming your data can obscure or create effects in your model results

GLMs might be a better tool to account for non-normal distributed data

Understanding model output

```
summary(model_Large3)

## Fixed effects:

##                              Estimate Std. Error t value

## (Intercept)                  1020.739     23.344  43.726

## GroupExperimental             -76.150     29.565  -2.576

## ContextUK                     -18.660      8.471  -2.203

## GroupExperimental:ContextUK    53.089     11.706   4.535
```

	PL Context	UK Context
Experimental Group		
Control Group		

Emmeans-package

```
library(emmeans)

em1 <- emmeans(model_Large3, specs = pairwise ~ Group:Context)


em1$emmeans

##  Group        Context emmean   SE  df asymp.LCL asymp.UCL
##  Control      PL        1021 23.3 Inf       975      1066
##  Experimental PL         945 23.5 Inf       899       991
##  Control      UK        1002 23.3 Inf       956      1048
##  Experimental UK         979 23.4 Inf       933      1025
##
## Degrees-of-freedom method: asymptotic
## Confidence level used: 0.95
```

Computes estimated marginal means (EMMs) for specified factors or factor combinations in a linear model and comparisons or contrasts among them

EMMs are also known as least-squares means

Emmeans-package

```
em1$contrasts
##  contrast                               estimate    SE  df z.ratio p.value
##  Control PL - Experimental PL               76.1 29.57 Inf   2.576  0.0491
##  Control PL - Control UK                    18.7  8.47 Inf   2.203  0.1224
##  Control PL - Experimental UK               41.7 29.55 Inf   1.412  0.4919
##  Experimental PL - Control UK              -57.5 29.51 Inf  -1.948  0.2080
##  Experimental PL - Experimental UK         -34.4  8.04 Inf  -4.282  0.0001
##  Control UK - Experimental UK               23.1 29.51 Inf   0.782  0.8629
##
## Degrees-of-freedom method: asymptotic
## P value adjustment: tukey method for comparing a family of 4 estimates
```

Understanding model output

```
Em1$emmeans

##  Experimental PL          945 23.5 Inf       899       991

##  Experimental UK          979 23.4 Inf       933      1025

Em1$contrasts

##  Experimental PL - Experimental UK    -34.4  8.04 Inf  -4.282  0.0001
```

945ms – 979ms = -34ms

Why the weird variable names?

```
summary(model_Large3)

## Fixed effects:

##                             Estimate Std. Error t value

## (Intercept)                 1020.739     23.344  43.726

## GroupExperimental            -76.150     29.565  -2.576

## ContextUK                    -18.660      8.471  -2.203

## GroupExperimental:ContextUK   53.089     11.706   4.535
```

Lme4 uses contrast-coding or dummy-coding for categorical variables

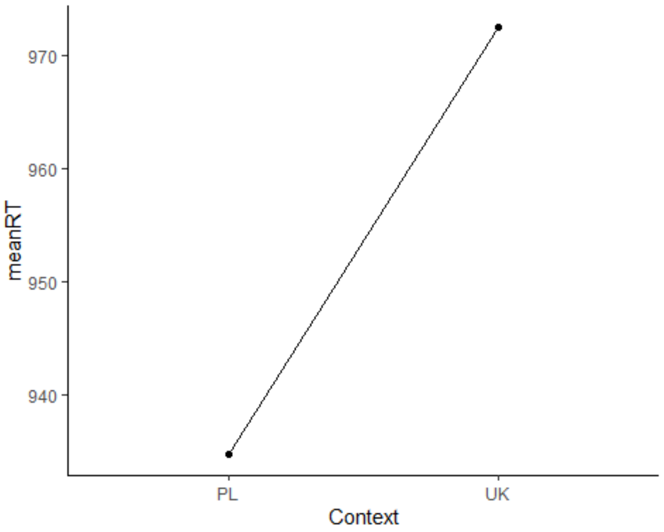
Contrasts have two functions:

- Control estimate calculation
- Handling multilevel categories

The explanatory variables are related linearly to the response.

Linear mixed effects models are **linear** and additive

Categorical variables are treated as 2-level variables



Contrasts

Categorical variables are used as factorial predictors, with several categorical levels

To estimate their effects, we assign numbers to each of the levels

These number are used for calculating comparisons

Different contrasts express different hypotheses

The underlying model remains the same, only the parametrisation of the effects changes

-> The choice of contrasts has no statistical consequence

-> It only changes estimate and hypothesis interpretation

Schad, D. J., Vasishth, S., Hohenstein, S., & Kliegl, R. (2020). How to capitalize on a priori contrasts in linear (mixed) models: A tutorial.

Implementing specific hypotheses into your model

Contrasts *reparametrize* the model and changes interpretation of parameters

R by default orders factors alphabetically and uses the first level as the baseline

- Contrasts are the re-ordering of levels

Wide variety of contrast types

- Treatment contrasts
- Sum contrasts
- Repeated contrasts
- Polynomial contrasts

Treatment contrasts

Default setting of R

-> Categorical variables are sorted alphabetically and turned into treatment contrasts

-> Category levels are ‘dummy-coded’ with 0/1 values

-> The level coded as 0 is the ‘reference level’ or ‘baseline level’

Contrast implementation in R

```
contrasts(PN_Data$Context)

##     UK

## PL   0

## UK   1


PN_Data$Context <- factor(PN_Data$Context,levels = c("UK", "PL"))
contrasts(PN_Data$Context)

##     PL

## UK   0

## PL   1
```

Treatment contrasts

```
##    UK
## PL  0
## UK  1
model_Large8 = lmer(RT ~ Group*Context + (1 | Subject) +
                             (1| ItemNr), data=PN_Data)
summary(model_Large8)
## Fixed effects:
##                              Estimate Std. Error t value
## (Intercept)                   1020.81      23.42  43.589
## GroupExperimental              -76.20      29.45  -2.587
## ContextUK                      -18.74       8.49  -2.208
## GroupExperimental:ContextUK     53.00      11.73   4.517
```

```
##    PL
## UK  0
## PL  1
model_Large8 = lmer(RT ~ Group*Context + (1 | Subject) +
                             (1| ItemNr), data=PN_Data)
summary(model_Large8)
## Fixed effects:
##                              Estimate Std. Error t value
## (Intercept)                   1002.07      23.36  42.895
## GroupExperimental              -23.19      29.40  -0.789
## ContextPL                       18.74       8.49   2.208
## GroupExperimental:ContextPL    -53.00      11.73  -4.517
```

Treatment contrasts

```
em1$contrasts
##  contrast                          estimate    SE  df z.ratio p.value
##  Control PL - Experimental PL          76.1 29.57 Inf   2.576  0.0491
##  Control UK - Experimental UK          23.1 29.51 Inf   0.782  0.8629
em1$emmeans
##  Group        Context emmean   SE  df asymp.LCL asymp.UCL
##  Control      PL        1021 23.3 Inf       975      1066
##  Control      UK        1002 23.3 Inf       956      1048
```

```
## Fixed effects:
##                              Estimate Std. Error t value
## (Intercept)                   1020.81      23.42  43.589
## GroupExperimental              -76.20      29.45  -2.587
## ContextUK                      -18.74       8.49  -2.208
## GroupExperimental:ContextUK     53.00      11.73   4.517
```

```
## Fixed effects:
##                              Estimate Std. Error t value
## (Intercept)                   1002.07      23.36  42.895
## GroupExperimental              -23.19      29.40  -0.789
## ContextPL                       18.74       8.49   2.208
## GroupExperimental:ContextPL    -53.00      11.73  -4.517
```

Using different contrasts

Intercepts estimate the dependent variable when *all* predictors are at 0

If the ‘distance’ between the levels equals 1, then the slope estimates their difference

->estimates measure the difference *per unit* of the predictor

Sum contrasts

If all predictors have their mean at 0, then the intercept estimates the grand-mean (across levels)

-> You are centering your predictor on the mean

-> Sum contrasts use -0.5/05; -1/1; -2/2 etc for centering

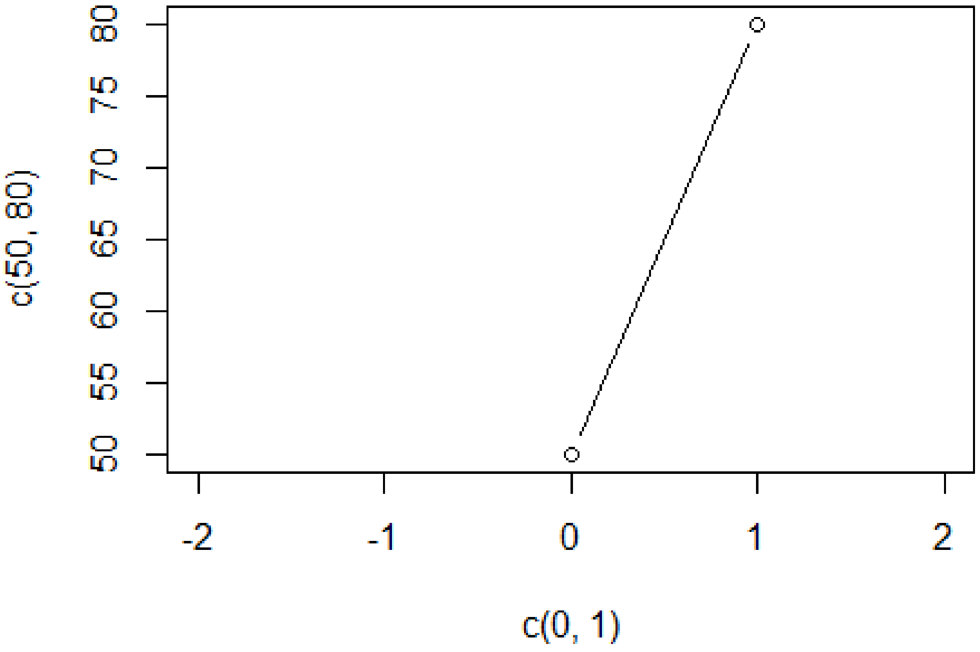
-0.5/0.5 also maintains distance between levels at 1

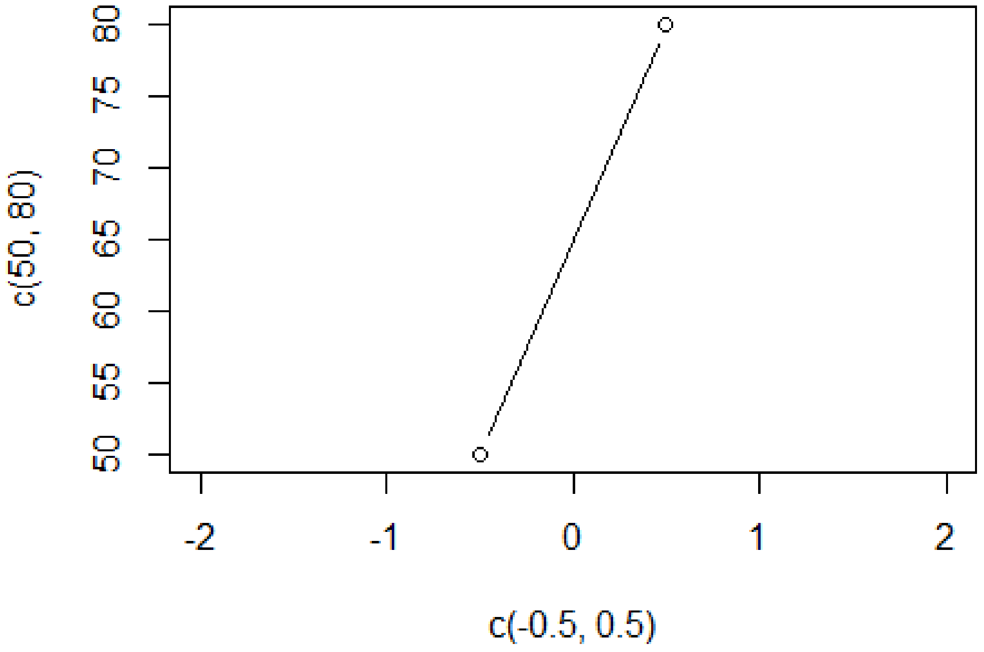
Sum contrasts

```
contrasts(PN_Data$Context) <-  contr.sum(2)/2
contrasts(PN_Data$Context)

##    [,1]

## UK  0.5

## PL -0.5


## Fixed effects:

##                             Estimate Std. Error t value

## (Intercept)                  1011.44      23.00  43.973

## GroupExperimental              -49.70      28.84  -1.723

## Context1                       -18.74       8.49  -2.208

## GroupExperimental:Context1     53.00      11.73   4.517
```

```
##    UK

## PL  0

## UK  1


## Fixed effects:

##                             Estimate Std. Error t value

## (Intercept)                  1020.81      23.42  43.589

## GroupExperimental              -76.20      29.45  -2.587

## ContextUK                      -18.74       8.49  -2.208

## GroupExperimental:ContextUK    53.00      11.73   4.517
```

Treatment and sum contrasts





Contrasts for multilevel categories

Three level categories are more difficult

-> Contrasts include the creation of dummy variables for comparing all levels of a category with each other

```
## Fixed effects:
##                         Estimate Std. Error t value
## (Intercept)              1011.44      23.00  43.973
## Context.1                  -9.74       7.34  -1.784
## Context.2                  -2.54       8.75  -2.984
## Context.3                  -11.67      8.63  -2.648
```

-> R provides functions to automatically create different contrasts for multiple levels

-> part of the MASS package

```
library(MASS)
```

Multiple levels with Treatment contrasts

```
contrasts(PN_Data$Context) <-
contr.treatment(5)

contr.treatment(5)

##   2 3 4 5
## 1 0 0 0 0
## 2 1 0 0 0
## 3 0 1 0 0
## 4 0 0 1 0
## 5 0 0 0 1
```

Rows are group levels

Columns are tested comparisons

Control group against experimental groups

Default setting in R

First level is treated as control group

Evers other level is compared against it

Multiple levels with Sum contrasts

```
contr.sum(5)
##    [,1] [,2] [,3] [,4]
## 1     1    0    0    0
## 2     0    1    0    0
## 3     0    0    1    0
## 4     0    0    0    1
## 5    -1   -1   -1   -1
```

Rows are group levels

Columns are tested comparisons

Experimental groups against grand average

Level with 1 is always compared against all others

Level 5 is set to -1, as it is already implicitly compared to all others

Multiple levels with Sum contrasts

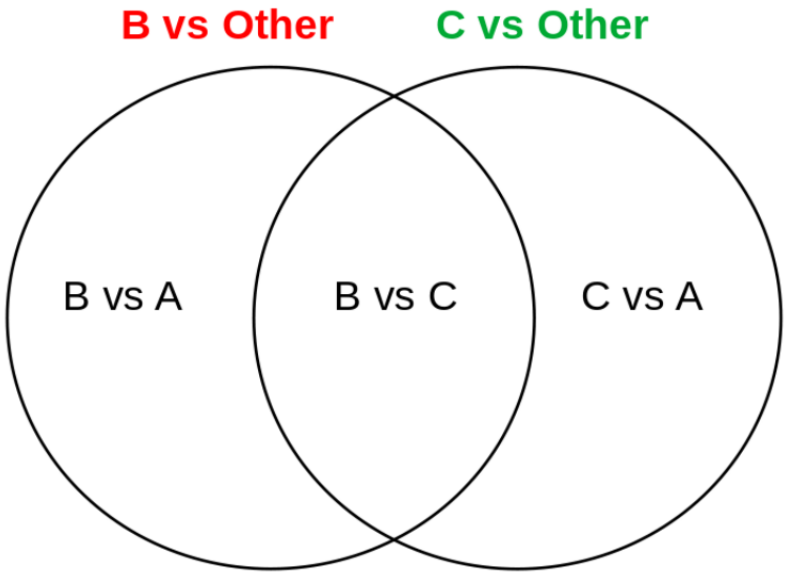
```
contr.sum(5)
```

```
##    [,1] [,2] [,3] [,4]
## 1     1    0    0    0
## 2     0    1    0    0
## 3     0    0    1    0
## 4     0    0    0    1
## 5    -1   -1   -1   -1
```

Rows are group levels

Columns are tested comparisons

Three levels (A, B, C)



Multiple levels with repeated contrasts

```
contr.sdif(5)

##    2-1  3-2  4-3  5-4
## 1 -0.8 -0.6 -0.4 -0.2
## 2  0.2 -0.6 -0.4 -0.2
## 3  0.2  0.4 -0.4 -0.2
## 4  0.2  0.4  0.6 -0.2
## 5  0.2  0.4  0.6  0.8
```

Rows are group levels

Columns are tested comparisons

Comparisons are made between successive neighbouring levels

Requires ordered categories, but not evenly spaced ones

Centered around the grand average

Used for testing „increasing“ levels

Multiple levels with polynomial contrasts

```
contr.poly(5)

##                 .L         .Q            .C         ^4
## [1,] -6.324555e-01  0.5345225 -3.162278e-01  0.1195229
## [2,] -3.162278e-01 -0.2672612  6.324555e-01 -0.4780914
## [3,] -3.510833e-17 -0.5345225  1.755417e-16  0.7171372
## [4,]  3.162278e-01 -0.2672612 -6.324555e-01 -0.4780914
## [5,]  6.324555e-01  0.5345225  3.162278e-01  0.1195229
```

Rows are group levels

Columns are tested comparisons

Checking for linear, quadratic, cubical and quartic trends in categories

Required sorted and evenly spaced categories

Multiple levels with Helmert contrasts

`contr.helmert(5)`

```
##   [,1] [,2] [,3] [,4]
## 1   -1   -1   -1   -1
## 2    1   -1   -1   -1
## 3    0    2   -1   -1
## 4    0    0    3   -1
## 5    0    0    0    4
```

Rows are group levels

Columns are tested comparisons

Comparisons of the level with all previous levels

LME to-do list

1 Hypotheses

Generate hypotheses based on previous research and your specific research questions

Collect data that might answer those questions

Formalize your hypotheses into falsifiable predictions

$$H_0 \text{ and } H_1$$

2 Prepare your data

In order to use data in your models, you need to prepare it correctly:

- Use the correct variable format (numeric, factor)
- Apply transformations if necessary (log, scaling, reciprocal)
- Apply contrasts for answering your research questions

```
 load("PictureNaming.RData")

Model_Data <-
  PN_Data %>%
  mutate(Context = as.factor(Context)) %>%
  select(Subject, Context, RT, Trial) %>%
  mutate(Trial = as.numeric(Trial)) %>%
  mutate(trans_RT = 1/RT)

contrasts(Model_Data$Context) <-  contr.sum(2)/2
```

3 Build your model

Create the maximal model based on your data structure (clustering variables) and all available sensible data

Reduce the model according to Barr et al (2013)

Check your model assumptions

- Linearity
- Constant variance for residuals
- Normal distribution of residuals
- If necessary go back to step 2

4 Analyse your model

Use summary() to get your model output

- If desired with p-values using lmerTest

Understand and interpret your estimates of all fixed effects

Perform post-hoc test for (significant) interactions to understand the underlying driving effect

5 Report your results

Describe your data, your maximal and final model and your final statistical analyses

Create plots based on your model output

Questions and discussions

If you have anymore questions regarding previous lectures or your own research, please send me a mail to:

Jonas.walther@uni-tuebingen.de

We can discuss them during the course on Friday.

Evaluation

Please give your feedback for the course at:

https://evaluation.zeq.uni-tuebingen.de/evasys/public/online/

Thank you for your attention!