Linear mixed models in R
Day 3

JONAS WALTHER

Check the available data

Dependent variable: RT

Grouping variables:

- Subject and Item

Experimental manipulations:

- Group
- Context

Background information:

- Age
- Age of L2 acquisition
- Trial
- Word frequency

RT ~ Context + (1 |Subject)

RT ~ Group*Context + (1 | Subject)

RT ~ Group* Context + (1 | Subject) + (1 |Item)

RT ~ Group* Context + Age + AoA + Trial + Freq + (1 | Subject) + (1 |Item)

RT ~ Group* Context + Age + AoA + Trial + Freq + (1 +Context +Trial+Freq| Subject) + (1 + Group + Age + AoA|Item)

How to design your own model?

Different researchers have suggested different methods of constructing your LMEs

- Data-driven approach
- Design driven approach

Data-driven model construction

Let your data „speak for itself“

1. Construct two versions of your model i.e. one with the fixed effect Age, one without
2. Check the fit of the models – which model explains the variation in the data better?
3. Select the model with the better fit
4. Repeat 1 to 3 for every variable/effect you want to add or remove from your model
5. You end with the best-fit model for your given data

Data-driven model construction

Use tools for determining the fit of a model

Akaike information criterion – AIC

- Gives information values of a model based on the amount of explained variation against the used parameters
- Rewards good fit to the data but punishes overfitting (too many parameters)

- Models with a lower AIC are a better fit than model with higher AIC

Data-driven model construction

```
model_General1 = lmer(RT ~ Group*Context + (1 | Subject) + (1 | ItemNr), data=PN_Data)
model_General2 = lmer(RT ~ Group*Context + (1 | ItemNr), data=PN_Data)

anova(model_General1, model_General2)
## refitting model(s) with ML (instead of REML)
## Data: PN_Data
## Models:
## model_General2: RT ~ Group * Context + (1 | ItemNr)
## model_General1: RT ~ Group * Context + (1 | Subject) + (1 | ItemNr)
##                npar    AIC    BIC logLik deviance  Chisq Df Pr(>Chisq)
## model_General2    6 102109 102150 -51048   102097
## model_General1    7 100790 100839 -50388   100776 1320.2  1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Data-driven model construction

```
model_General3 = lmer(RT ~ Group*Context + (1 | Subject) +  (1 | ItemNr), data=PN_Data)
model_General4 = lmer(RT ~ Group*Context + Age + (1 | Subject) + (1 | ItemNr), data=PN_Data)

anova(model_General3, model_General4)
## refitting model(s) with ML (instead of REML)
## Data: PN_Data
## Models:
## model_General3: RT ~ Group * Context + (1 | Subject) + (1 | ItemNr)
## model_General4: RT ~ Group * Context + Age + (1 | Subject) + (1 | ItemNr)
##                npar    AIC    BIC logLik deviance  Chisq Df Pr(>Chisq)
## model_General3    7 100790 100839 -50388   100776
## model_General4    8 100792 100847 -50388   100776 0.1057  1     0.7451
```

Barr, DJ, Levy, R, Scheepers, C and Tily, HJ (2013) Random effects structure for confirmatory hypothesis testing: Keep it maximal.

LMEs generalise best when researchers use the **maximal random structure**, justified by the design

Not including all available and sensible random effects inflates Type I errors

Within-subject designs can show very high Type I error rates, if only random intercepts are used but not random effects

But:

This might depend on your research

- Confirmatory hypothesis testing prefers design-driven approaches
- Exploratory science might prefer data-driven approaches

Barr, DJ, Levy, R, Scheepers, C and Tily, HJ (2013) Random effects structure for confirmatory hypothesis testing: Keep it maximal.

In order to correctly interpret treatment effects, you need to know which cluster variables were used and how they varied

<u>It is better to include all possible and sensible random slopes in your model!</u>

<u>Including only random intercepts might not be enough!</u>

Sometimes *underfitting the design* is worse than *overfitting the data*

Barr, DJ, Levy, R, Scheepers, C and Tily, HJ (2013) Random effects structure for confirmatory hypothesis testing: Keep it maximal.

(…) [T]he maximal random effect structure should be fitted to the data. This includes a variance component for subject-related and item-related intercepts, for every within-subject and within-item fixed effect, and in the ideal case even all possible correlations between these random effects. The random effect structure should be reduced if and only if the maximal model does not converge. (…)

Maximal model

RT ~ Group* Context + Age + AoA + Trial + Freq + (1 +Context +Trial+Freq| Subject) + (1 + Group + Age + AoA|Item)

Largest possible, sensible model design

Includes all relevant fixed effects for research questions and controls for all realistic covariates

Design-driven maximal model that accurately reflects the experimental design

- Fixed and random effects need to reflect the experiment/data structure

Barr, DJ, Levy, R, Scheepers, C and Tily, HJ (2013) Random effects structure for confirmatory hypothesis testing: Keep it maximal.

Sometimes your model does not converge i.e. the maximum likelihood estimation fails

- Chance of failure increases with complexity of the model, especially random effects
- -> It might be necessary to reduce the model
- -> You should follow established and strict rules, not do it randomly

Bates, D, Kliegl, R, Vasishth, S, Baayen, RH (2015) Parsimonious Mixed Models

Assumptions:

- Created maximal model
- Researcher is interested in fixed effects, not necessarily random effects or correlations

Procedure:

1. Remove components with the smallest variance
   - Switch from zero-correlation to correlation
2. Re-run model and summary() function
   - If it converges, use that model
   - If it doesn‘t converge, repeat from step 1

Add correlation between random slopes and intercepts

No correlations for Subject random effects:

RT ~ Group* Context + (1 + Context|| Subject)

Equates to:

RT ~ Group* Context + (0 + Context| Subject) + (1 | Subject)

Correlations between slopes and intercept

RT ~ Group* Context + (1 + Context| Subject)

Removing components with lowest variance

```
model_Large1 = lmer(RT ~ Group*Context + (1 | Subject) +
                         (1 + Group*Age| ItemNr), data=PN_Data)

## boundary (singular) fit: see help('isSingular')

summary(model_Large1)

## Random effects:

##  Groups   Name                  Variance  Std.Dev. Corr

##  ItemNr   (Intercept)           33530.038 183.112

##           GroupExperimental     17987.950 134.119  -0.88

##           Age                       3.565   1.888  -0.65  0.86

##           GroupExperimental:Age    11.617   3.408   0.92 -0.97 -0.72

##  Subject  (Intercept)           14923.867 122.163

##  Residual                       59035.597 242.972

## Number of obs: 7227, groups:  ItemNr, 210; Subject, 74
```

```
model_Large2 = lmer(RT ~ Group*Context + (1 | Subject) +
                         (1 + Group+Group:Age| ItemNr),
data=PN_Data)

## boundary (singular) fit: see help('isSingular')

summary(model_Large2)

## Random effects:

##  Groups   Name                  Variance  Std.Dev. Corr

##  ItemNr   (Intercept)           34100.018 184.662

##           GroupExperimental     16606.764 128.867  -0.94

##           GroupControl:Age          3.763   1.940  -0.66
0.88

##           GroupExperimental:Age     5.045   2.246   0.85 -
0.64 -0.19

##  Subject  (Intercept)           14501.643 120.423

##  Residual                       59070.801 243.045

## Number of obs: 7227, groups:  ItemNr, 210; Subject, 74
```

Removing components with lowest variance

```
model_Large2 = lmer(RT ~ Group*Context + (1 | Subject) +
                          (1 + Group+Group:Age| ItemNr), data=PN_Data)

## boundary (singular) fit: see help('isSingular')

summary(model_Large2)

## Random effects:

##  Groups   Name                   Variance  Std.Dev. Corr

##  ItemNr   (Intercept)            34100.018 184.662

##           GroupExperimental      16606.764 128.867  -0.94

##           GroupControl:Age           3.763   1.940  -0.66  0.88

##           GroupExperimental:Age      5.045   2.246   0.85 -0.64 -0.19

##  Subject  (Intercept)            14501.643 120.423

##  Residual                        59070.801 243.045

## Number of obs: 7227, groups:  ItemNr, 210; Subject, 74
```

```
model_Large3 = lmer(RT ~ Group*Context + (1 | Subject) +
                          (1 + Group| ItemNr), data=PN_Data)

summary(model_Large3)

## Random effects:

##  Groups   Name              Variance Std.Dev. Corr

##  ItemNr   (Intercept)       22974    151.57

##           GroupExperimental  1641     40.51   0.03

##  Subject  (Intercept)       14736    121.39

##  Residual                   59279    243.47

## Number of obs: 7227, groups:  ItemNr, 210; Subject, 74
```

Matuschek, H, Kliegl, R, Vasishth, S, Baayen, H, Bates, D (2017) Balancing Type I error and power in linear mixed models

Maximal model drecreases rate of Type 1 errors succesfully

- But this can come at the cost of power

A good compromise might be parsimonious models based on the maximal model and best-fit criteria

An alternative might be the collection of more data to have more power to use more complex maximal models

Lme4 vs nlme

Lme4
- More modern
- Handles large number of random effects better (implemented in C)
- Handles crossed random effects better
- Allows easier implementation of supplementary packages

But:
- Nlme gives more freedom in the covariance structures for random effects
- Nlme provides p-values

Lme4 Addons

lmerTest

Emmeans

ggeffects

simr

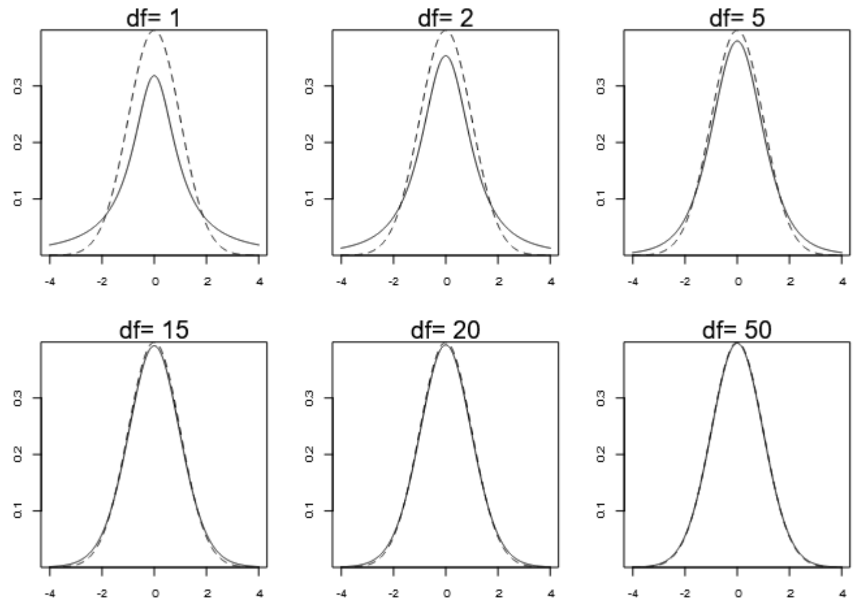
Summary-Output of an LME

- Lme4 does not produce p-values
  - But designed to be modular with other packages
- Multiple options for p-value calculation/significance test

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: RT ~ Group + Context + Group:Context + Age + AoA + Trial + (1 +
##     Context | Subject) + (1 + Context | ItemNr)
##    Data: PN_Data
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  ItemNr   (Intercept) 24021    154.99
##           ContextUK    3449     58.73   -0.15
##  Subject  (Intercept) 19522    139.72
##           ContextUK   16426    128.16   -0.41
##  Residual             55688    235.98
## Number of obs: 7227, groups:  ItemNr, 210; Subject, 74
##
## Fixed effects:
##                              Estimate Std. Error t value
## (Intercept)                 1019.6555    79.0877  12.893
## GroupExperimental            -59.6657    37.2576  -1.601
## ContextUK                    -11.2579    23.2922  -0.483
## Age                           -1.9501     2.4100  -0.809
## AoA                            3.2835     3.9845   0.824
## Trial                          0.4629     0.2875   1.610
## GroupExperimental:ContextUK   45.7348    32.1411   1.423
##
## Correlation of Fixed Effects:
##             (Intr) GrpExp CntxUK Age    AoA    Trial
## GrpExprmntl  0.196
## ContextUK   -0.127  0.290
## Age         -0.810 -0.309 -0.012
## AoA         -0.269 -0.217  0.001 -0.234
## Trial       -0.110 -0.001 -0.006  0.000  0.001
## GrpExpr:CUK  0.088 -0.404 -0.704  0.010 -0.001  0.000
```

Why doesn't lme4 give p-values?

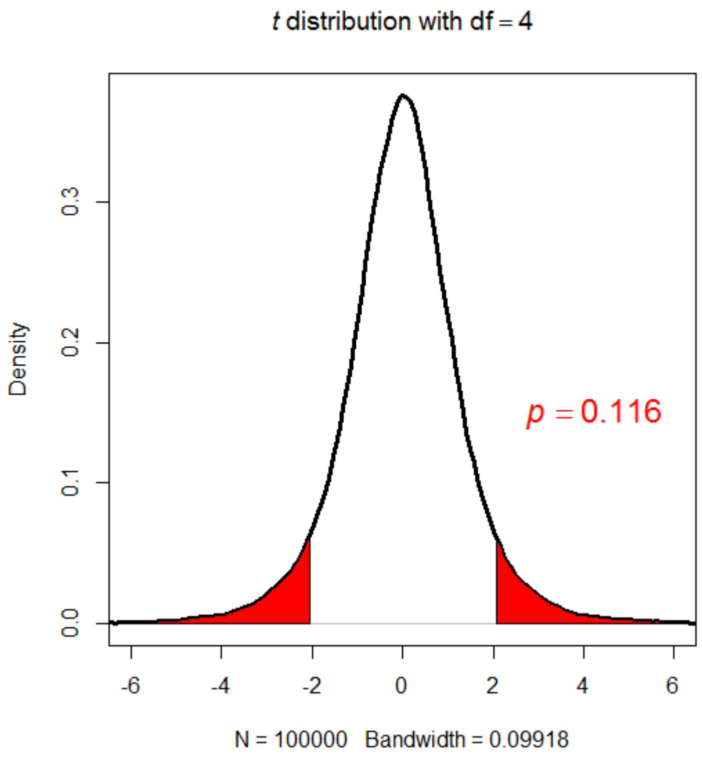
The shape of the *t* distribution is based on degrees of freedom

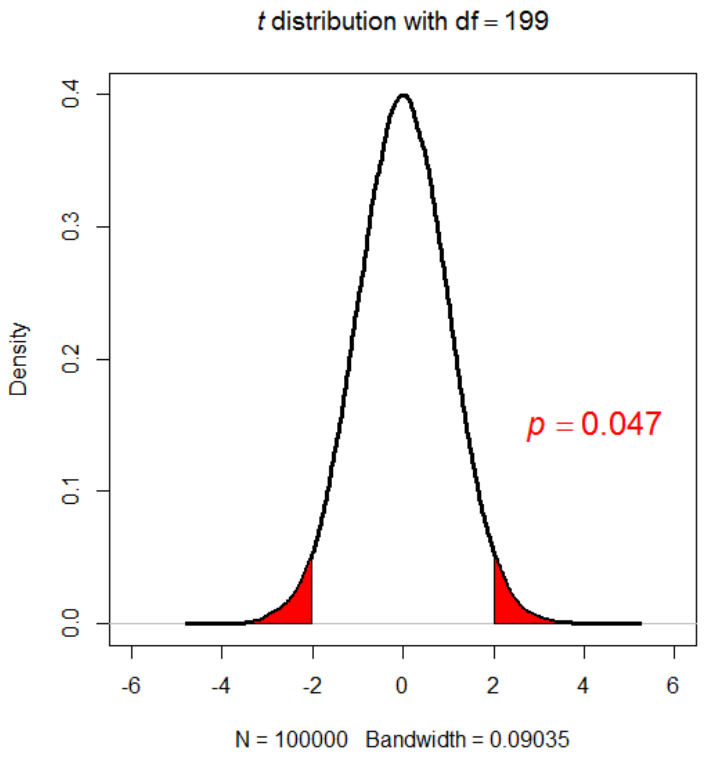


Why doesn't lme4 give p-values?

The more degrees of freedom, the skinnier the distribution (and thus the less likely effects near the tails are)

For mixed-effects models, no one is sure how to count the degrees of freedom





Assuming high dfs

With high *df*s, the critical *t*-value for two-tailed α=.05 approaches **1.96**

You can use t= 2 as an approximation of significance

Only appropriate for hundreds of trials

lmerTest-package

Satterthwaite method estimates the appropriate *df*s for *t*-values of mixed-effect model coefficients

After you load the `{lmerTest}` package, these will be shown automatically in model summaries

lmerTest modifies functions of lme4

Loading the package changes functions from other packages

In order to load and unload lmerTest you can use:

```
library(lmerTest)

model_Large3 = lmer(RT ~ Group*Context + (1 | Subject) + (1 + Group| ItemNr), data=PN_Data)

summary(model_Large3)

unloadNamespace("lmerTest")
```

LmerTest: before and after

```
Before:
## Random effects:
##  Groups   Name               Variance Std.Dev. Corr
##  ItemNr   (Intercept)        22974    151.57
##           GroupExperimental   1641     40.51   0.03
##  Subject  (Intercept)        14736    121.39
##  Residual                    59279    243.47
## Number of obs: 7227, groups:  ItemNr, 210; Subject, 74
## 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
```

```
After:
## Random effects:
##  Groups   Name               Variance Std.Dev. Corr
##  ItemNr   (Intercept)        22974    151.57
##           GroupExperimental   1641     40.51   0.03
##  Subject  (Intercept)        14736    121.39
##  Residual                    59279    243.47
## Number of obs: 7227, groups:  ItemNr, 210; Subject, 74
## Fixed effects:
##                              Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)                  1020.739     23.344  120.000  43.726  < 2e-16 ***
## GroupExperimental             -76.150     29.565   79.696  -2.576   0.0119 *
## ContextUK                     -18.660      8.471 6940.635  -2.203   0.0276 *
## GroupExperimental:ContextUK    53.089     11.706 6963.765   4.535 5.85e-06 ***
```

Notes on lmerTest

Lme4 intentionally did not include p-values, as people disagree on correct calculation

- Lme4 was designed to be modular with other packages

Using lmerTest is more a personal choice

lmerTest modifies lmer()-object, summary() and anova()

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
```

$$y_{iCond} = \beta_0 + u_{0i} + \binom{0}{1}\delta + \epsilon_{Cond}$$

$$RT_{PL} = 1020 + -18 \times 0 + \epsilon = 1020$$

$$RT_{UK} = 1020 + -18 \times 1 + \epsilon = 1002$$

```
model_Large3 %>%
    ggpredict(c("Context"))
```

```
## # Predicted values of RT

## Context | Predicted |          95% CI

## ----------------------------------------

## PL      |   1020.74 | 974.98, 1066.50

## UK      |   1002.08 | 956.43, 1047.72
```

Thank you for your attention!