

# Mixed Models

Jonathan Harrington & Ulrich Reubold

02.07.2019

```
# ggf.:  
#install.packages("lmerTest")  
#install.packages("emmeans")  
  
library(lmerTest)  
library(emmeans)  
library(ggplot2)  
library(dplyr)  
library(ez)  
  
# post-hoc-Funktionen Laden:  
source(file.path(pfadu, "phoc.txt"))  
  
# Daten Laden  
stimm = read.table(file.path(pfadu, "stimm.df.txt"))  
stimm2 = read.table(file.path(pfadu, "stimm2.df.txt"))  
soa = read.table(file.path(pfadu, "soa.txt"))  
int = read.table(file.path(pfadu, "dbwort.df.txt"))
```

## 1. Mixed Models: viele Vor-, wenige Nachteile

Mit einem Mixed Model (MM) (der deutschsprachige Begriff "lineare gemischte Modelle" wird sehr selten benutzt) wird geprüft, ob eine abhängige Variable (die kontinuierlich (`lmer()`) oder (wenn `glmer()` benutzt wird) kategorial sein kann) von einem oder mehreren unabhängigen Faktoren beeinflusst wird. Die unabhängigen Faktoren sind meistens kategorial, können aber auch numerisch sein. "Mixed" oder "gemischt" wird ein Mixed Model dadurch, dass es sowohl Fixed als auch Random Factors geben kann, also sowohl Faktoren, deren Einfluss auf die abhängige Variable den Forschenden interessieren (= Fixed Factor), als auch jene, von denen zu erwarten ist, dass sie zwar höchstwahrscheinlich einen Einfluss auf die abhängige Variable ausüben werden, dieser Einfluss aber für den Forschenden uninteressant ist (und möglicherweise das Zu-Tage-Treten den Effekt des interessierenden Faktors verdeckt).

Einen Random Effect gibt es zwar bei `ezANOVA()` auch (meistens in unserem Fall die unterschiedlichen Sprecher), aber es kann dort nicht mehr als einen Random Effect geben.

Mixed models (MM) haben eine breite Anwendung. Sie ersetzen:

- zum Teil die Varianzanalyse
- sie erweitern die lineare Regression
- sie erweitern auch die logistische Regression

## Vier Vorteile (A-D) von einem MM im Vergleich zur Varianzanalyse (und leider auch ein Nachteil (E))

### A. In einer ANOVA, aber nicht in einem Mixed Model muss über Wiederholungen gemittelt werden:

Z.B.: Inwiefern wird "vot" von der Artikulationsstelle ("K") und vom "Alter" beeinflusst?

```
head(stimm)

##   vot  K Vpn Alter
## 1  10 ba  A  Kind
## 2 -16 ba  B  Kind
## 3   9 ba  C  Kind
## 4 -21 ba  E  Erw
## 5  10 ba  F  Erw
## 6  -1 ba  G  Kind

# zuerst mitteln: ein Wert pro K-Vpn-Alter-Kombination
stimm.m = stimm %>%
  group_by(K, Vpn, Alter) %>%
  summarise(vot = mean(vot))

stimm.ez = ezANOVA(
  data = stimm.m,
  dv = .(vot),
  wid = .(Vpn),
  within = .(K)
)

stimm.ez

## $ANOVA
##   Effect DFn DFd      F      p p<.05      ges
## 2      K   1   7 41.05629 0.0003646654 * 0.02523055
```

Hier gibt es also keine Probleme. Aber wenn wir den Effekt des Alters ermitteln möchten, stellen wir fest, dass wir kein ausbalanciertes Design aufzuweisen haben:

### B. Kein Balanced-design in einem MM erforderlich:

```
# Balanced design: Die Anzahl der Vpn. muss pro Stufe des Between-Faktors
# gleich sein. Dies ist hier offensichtlich nicht gegeben für den
# Between-Faktor Alter.
with(stimm, table(Vpn, Alter))

##   Alter
## Vpn Erw Kind
##  A   0  16
##  B   0  16
##  C   0  16
```

```
## D 16 0
## E 16 0
## F 16 0
## G 0 16
## H 0 16
```

D.h.: es gibt 3 Erwachsene und 5 Kinder.

Daher meckert die ANOVA:

```
stimm.ez = ezANOVA(
  data = stimm.m,
  dv = .(vot),
  wid = .(Vpn),
  within = .(K),
  between = .(Alter)
)
```

```
## Warning: Data is unbalanced (unequal N per group). Make sure you specified
## a well-considered value for the type argument to ezANOVA().
```

Das ist zwar “nur” eine Warnmeldung, aber wir können nie sicher sein, ob es wirklich “richtig” ist, was wir hier tun; “unequal N per group” ist aber kein Problem für MMs.

## C. Fehlende Werte sind in einem MM erlaubt:

In einer ANOVA muss jede Stufe des within-Faktors belegt sein. Das ist bei einem MM nicht der Fall:

```
with(stimm2, table(Vpn, K))
```

```
##      K
## Vpn ba pa
## A  0  8
## B  8  8
## C  8  8
## D  8  0
## E  8  8
## F  8  8
## G  8  8
## H  8  8
```

```
# Vpn ba pa
```

```
# A 0 8
```

```
# B 8 8
```

```
# C 8 8
```

```
# D 8 0
```

```
# E 8 8
```

```
# F 8 8
```

```
# G 8 8
```

```
# H 8 8
```

```
# daher kann ANOVA nicht durchgefuehrt werden.
```

```
# Mitteln:
```

```
stimm2.m = stimm2 %>%  
  group_by(K, Vpn, Alter) %>%  
  summarise(vot = mean(vot))
```

```
ezANOVA(  
  data = stimm2.m,  
  dv = .(vot),  
  wid = .(Vpn),  
  within = .(K),  
  between = .(Alter)  
)
```

```
## Error in ezANOVA_main(data = data, dv = dv, wid = wid, within = within, :  
One or more cells is missing data. Try using ezDesign() to check your data.
```

Fehlende Daten sind aber kein Problem für ein MM.

## D. Die Variabilität von mehr als nur einem einzigen Faktor ausklammern:

In einer ANOVA kann die Variabilität von höchstens einem Faktor (z.B. *Vpn* oder *Wort*) ausgeklammert werden. Solche Beschränkung gibt es für ein MM nicht (siehe 2. unten).

---

## E. Ein Nachteil: Ein MM ist bei einer kleinen Anzahl von Beobachtungen nicht stabil

Bei weniger als ca. 60 Beobachtungen (bei jeweils einem Fixed und einem Random Factor) konvergiert das MM oft nicht (obwohl das auch von der Anzahl der Faktoren abhängt - je mehr Faktoren, desto mehr Beobachtungen sind für eine "Konvergenz" (hier in etwa: "ein stabiles Modell") notwendig).

---

## 2. Fixed und Random Factors

In einem MM werden die Faktoren in 'fixed' und 'random' aufgeteilt:

- **Fixed Factor:** ein oder mehrere Faktoren, die geprüft werden sollen
- **Random Factor:** ein oder mehrere Faktoren, von denen die Variabilität ausgeklammert/reduziert werden soll.

In der Phonetik sind oft *Versuchsperson* und/oder *Item* (Materialien wie unterschiedliche Wörter) Random Factors.

Beispiel:

- 20 Sprecher produzierten verschiedene Wörter **phrasenfinal oder phraseninitial** und die Vokaldauer wurde gemessen. **Inwiefern wird die Vokaldauer von der Phrasenposition beeinflusst?**
  - Abhängige Variable: *Dauer*.
  - Fixed Factor: *Phrasenposition* (initial vs. final).
  - Random Factors: *Versuchspersonen, Wort* (da die Variabilität, die wegen Versuchspersonen und unterschiedlicher Wörter entsteht, entfernt werden soll).

### 3. Was in einem MM berechnet wird

Ein MM ist in einigen Hinsichten ähnlich wie die lineare Regression

- Die Regression:

$$\hat{y} = mx + k$$

- $\hat{y}$  sind die eingeschätzten Werte,
- $m$  und  $k$  sind die Steigung bzw. das Intercept
- Mixed Model (MM): In einem MM werden  $(m, k)$  in Random und Fixed aufgeteilt:

$$\hat{y} = (\text{Random}.m + \text{Fixed}.m)x + (\text{Random}.k + \text{Fixed}.k)$$

Die 4 Variablen werden auf eine solche Weise berechnet, sodass die Abstände zwischen den tatsächlichen Werten ( $y$ ) und den eingeschätzten Werten ( $\hat{y}$ ) minimiert werden (also ganz analog zur linearen Regression).

Für das obige Beispiel:

- Abhängige Variable: *vot* (= die tatsächlichen Werte)
- Fixed Factor: *K* (2 Stufen: *ba, pa*)
- Random Factor: *Vpn.* Wird in einem MM als  $(K|Vpn)$  oder  $(1|Vpn)$  verschlüsselt (mehr dazu in 4 unten).
  - $(K|Vpn)$  hat die Bedeutung: eine Steigung (*Random.m*) und ein Intercept (*Random.k*) sollen pro *Vpn.* berechnet werden.
  - $(1|Vpn)$  hat die Bedeutung: ein Intercept (*Random.k*) soll pro *Vpn.* berechnet werden, aber *keine* Steigung (mehr dazu in 4 unten).

Funktion: `lmer()` (für *linear mixed effects regression model*) in `library(lmerTest)`:

```
stimm.lmer = lmer(vot ~ K + (K | Vpn), data = stimm)
```

```
# Fixed.k, Fixed.m
fixef(stimm.lmer)
```

```

## (Intercept)          Kpa
## -2.432357          3.900090

# (Intercept)          Kpa
# -2.432357          3.900090

# Random.k, Random.m
# N.B. es gibt einen (k, m) pro Versuchsperson
ranef(stimm.lmer)

## $Vpn
## (Intercept)          Kpa
## A  13.438573  1.5463422
## B -16.549989  1.1125375
## C  10.581010 -1.5465398
## D  -6.152667  0.3295788
## E -19.812560 -0.1910081
## F  13.400605 -0.9500177
## G   1.524247  0.6061936
## H   3.570781 -0.9070865
##
## with conditional variances for "Vpn"

#
# (Intercept)          Kpa
# A  13.438581  1.5463280
# B -16.549963  1.1124886
# C  10.580984 -1.5464902
# D  -6.152658  0.3295624
# E -19.812548 -0.1910332
# F  13.400583 -0.9499770
# G   1.524253  0.6061825
# H   3.570768 -0.9070612

#
# Die ersten 12 eingeschätzten Werte (yhut)
fitted(stimm.lmer)[1:12]

##          1          2          3          4          5          6
## 11.0062161 -18.9823455  8.1486535 -22.2449172  10.9682478 -0.9081092
##          7          8          9         10         11         12
##  1.1384244 -13.9697176 10.5022041 -4.3553544 -18.5358349 13.9183205

#
# Berechnung der eingeschätzten Werte
# yhut = (Random.m + Fixed.m) x + (Random.k + Fixed.k)
#
# x ist hier /ba/ oder /pa/ und wird als 0 und 1 umgesetzt
contrasts(stimm$K)

```

```

##      pa
## ba  0
## pa  1

# ba  0
# pa  1
# d.h. 0 für ba, 1 für pa.
#
# z.B. yhut der 8en Beobachtung
#
stimm[8, ]

##      vot K Vpn Alter
## 8 -13 pa   B  Kind

#      vot K Vpn Alter
# 8 -13 pa   B  Kind
#
# Sprecher ist B, K = pa (daher x = 1)
# eingeschätzter Wert:
# yhut = (Random.m + Fixed.m) x + (Random.k + Fixed.k)
yhut = (1.1124886 + 3.900090) * 1 + (-16.549963 - 2.432357)
yhut

## [1] -13.96974

# [1] -13.96974
# das gleiche:
fitted(stimm.lmer)[8]

##          8
## -13.96972

# -13.96974
# der tatsächliche Wert -13 (siehe
stimm[8,]

##      vot K Vpn Alter
## 8 -13 pa   B  Kind

# )

```

---

#### 4. Mehr zu Random factors und der Wahl zwischen (FF|RF) (*within-RF factor*) und (1|RF) (*between-RF factor*)

Es gibt zwei Möglichkeiten, um einen Random factor zu deklarieren: (FF|RF) oder (1|RF).

```
head(soa)
```

```
##   Vpn   W   Pos   F1
## 1  s1  Bart  final 493.5
## 2  s1  Pfad  final 532.5
## 3  s1  Start final 514.5
## 4  s1  Bart  medial 482.0
## 5  s1  Pfad  medial 501.0
## 6  s1  Start medial 497.0
```

Drei Sprecher produzierten “Bad”, “Pfad”, “Start”, entweder in phrasenmedialer oder -finaler Position. **Inwiefern wird F1 im Vokal von der Phrasenposition beeinflusst?**

- Abhängige Variable: F1
- Fixed Factor: Pos (medial/final)
- Random Factors: Vpn, W (Versuchsperson, Wort)

Daher 4 Möglichkeiten:

1.

```
soa.lmer = lmer(F1 ~ Pos + (Pos | Vpn) + (Pos | W), data = soa)
```

2.

```
soa.lmer = lmer(F1 ~ Pos + (1 | Vpn) + (Pos | W), data = soa)
```

3.

```
soa.lmer = lmer(F1 ~ Pos + (Pos | Vpn) + (1 | W), data = soa)
```

4.

```
soa.lmer = lmer(F1 ~ Pos + (1 | Vpn) + (1 | W), data = soa)
```

Grundsätzlich (FF|RF) verwenden (daher 1.), es sei denn FF ist eindeutig ‘between’ im Bezug zu RF. für das obige Beispiel ist FF (Pos) ‘within’ im Bezug zum RF Vpn:

```
with(soa, table(Vpn, Pos))
```

```
##      Pos
## Vpn  final medial
##  s1      3      3
##  s2      3      3
##  s3      3      3
```

FF ist auch ‘within’ im Bezug zu Wort

```
with(soa, table(W, Pos))
```

```
##      Pos
## W      final medial
##  Bart      3      3
```

```
## Pfad      3      3
## Start    3      3

# also hier:
soa.lmer = lmer(F1 ~ Pos + (Pos | Vpn) + (Pos | W), data = soa)

## boundary (singular) fit: see ?isSingular

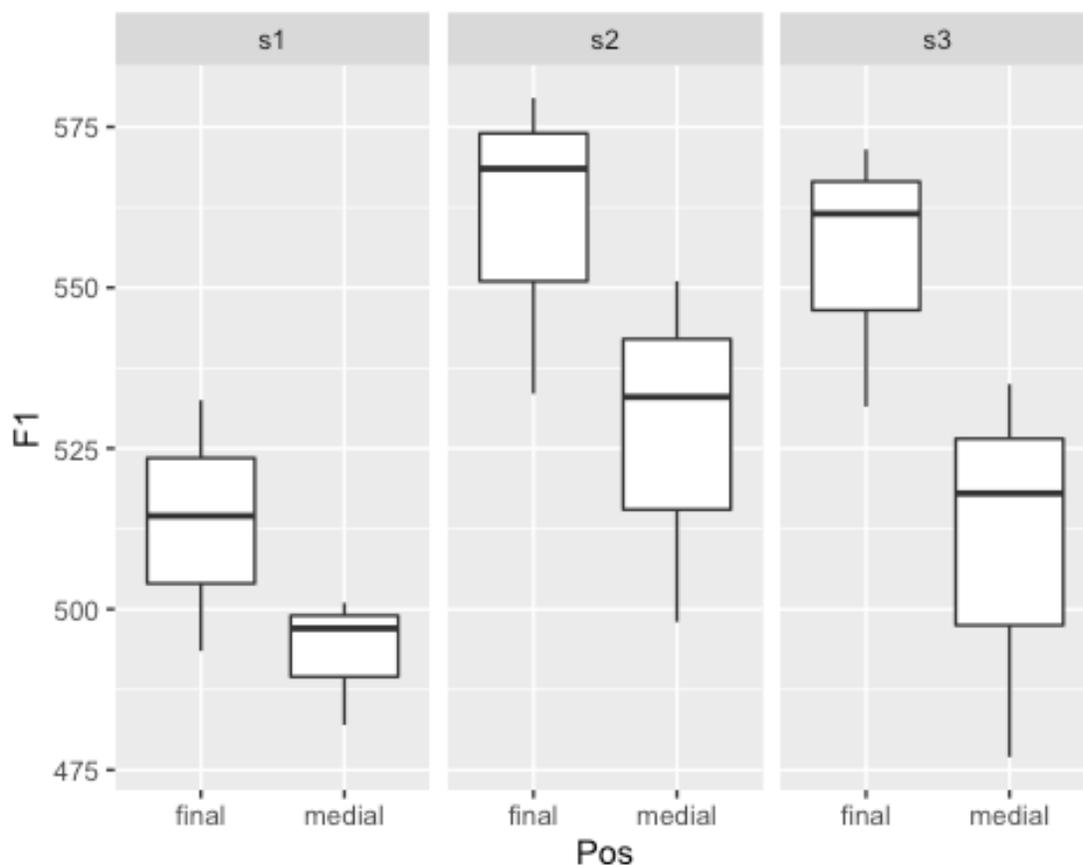
## Warning: Model failed to converge with 1 negative eigenvalue: -2.3e-01
```

## Die Bedeutung von (FF|RF)

1. Eine Steigung  $RF.m$  und Intercept  $RF.k$  werden berechnet
2. (Wichtiger): die Variabilität vom  $RF$  wird ausgeklammert, unter der Annahme dass  $RF$  und  $FF$  interagieren könnten.

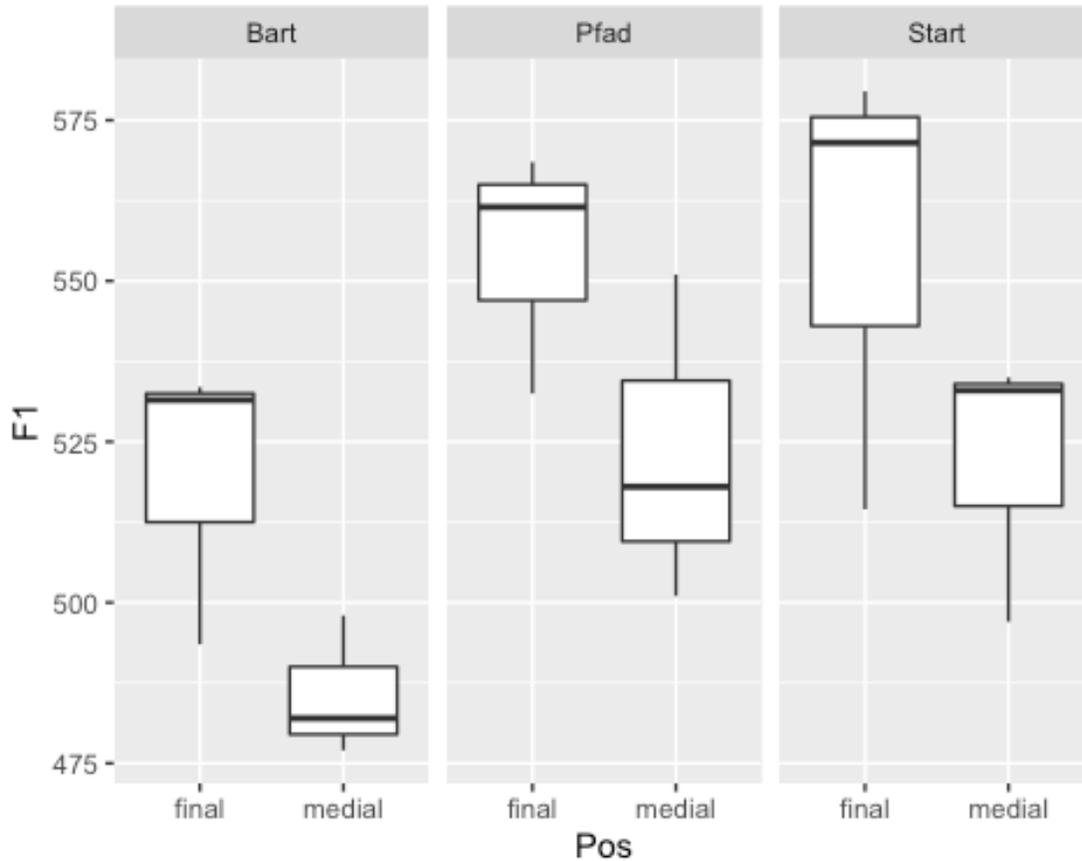
d.h. mit (Pos|Vpn) wird geprüfert, *inwiefern Pos und Vpn miteinander interagieren:*

```
ggplot(soa) +
  aes(y = F1, x = Pos) +
  geom_boxplot() +
  facet_wrap(~ Vpn)
```



Ebenfalls wird mit (Pos|W) geprüft, *inwiefern Pos und W miteinander interagieren:*

```
ggplot(soa) +
  aes(y = F1, x = Pos) +
  geom_boxplot() +
  facet_wrap(~ W)
```



### Die Bedeutung von (1|RF)

1. Ein Intercept  $RF.k$  (aber keine Steigung) wird berechnet.
2. (Wichtiger): die Variabilität vom  $RF$  wird ausgeklammert, ohne dass  $RF$  und  $FF$  interagieren.

(1|RF) soll verwendet werden, wenn der  $FF$  'between' ist im Bezug zum  $RF$ . Z.B.:

```
head(stimm)
```

```
##  vot K Vpn Alter
## 1  10 ba  A  Kind
## 2 -16 ba  B  Kind
## 3   9 ba  C  Kind
## 4 -21 ba  E  Erw
## 5  10 ba  F  Erw
## 6  -1 ba  G  Kind
```

```
# Inwiefern wird vot vom Alter beeinflusst?
# Fixed factor: Alter
# Random factor: Vpn
```

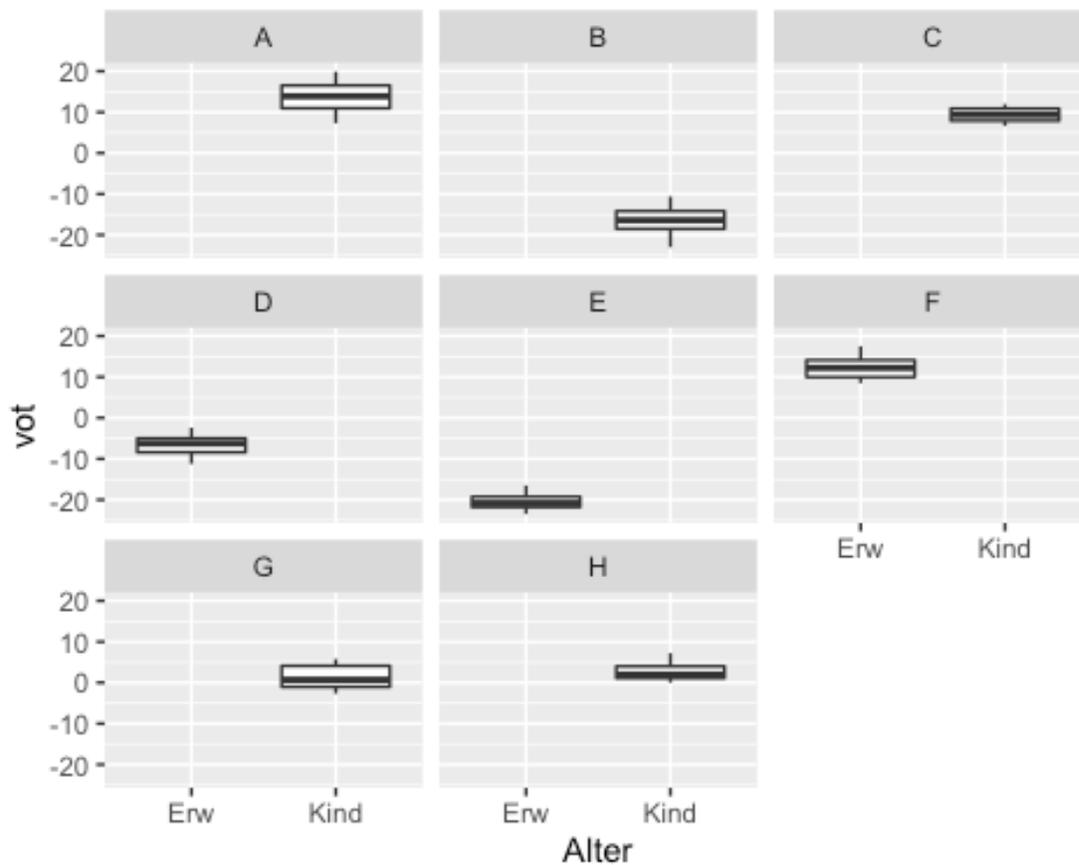
```

# Alter ist between im Bezug zu Vpn:
with(stimm, table(Vpn, Alter))

##      Alter
## Vpn Erw Kind
##  A   0  16
##  B   0  16
##  C   0  16
##  D  16   0
##  E  16   0
##  F  16   0
##  G   0  16
##  H   0  16

#
# Alter und Vpn können nicht interagieren, da jede Vpn
# entweder Kind oder Erw ist (wie folgender,
# an sich nicht uebermäßig sinnvoller
# Plot zeigt):
ggplot(stimm) +
  aes(y = vot, x = Alter) +
  geom_boxplot() +
  facet_wrap( ~ Vpn)

```



Daher wäre (Alter|Vpn) sinnlos - da dies voraussetzt, dass Alter und Vpn interagieren können, was nicht möglich ist, wenn *FF* 'between' ist in Bezug zum *RF*. Daher:

```
stimm3 = lmer(vot ~ Alter + (1 | Vpn), data = stimm)
# und nicht (Alter|Vpn)
```

Im Allgemeinen: **immer (FF|RF) verwenden, abgesehen von dem Fall, in dem *FF* wirklich 'between' ist in Bezug zu *RF*.**

## 5. Mehr als ein Fixed Factor

```
dim(stimm)
```

```
## [1] 128 4
```

VOT wurde in /ba, pa/ von Kindern und Erwachsenen produziert.

**Inwiefern wird "VOT" von der Artikulationsstelle (Faktor "K") und/oder von der Sprechergruppe (ob Kind oder Erwachsener: Faktor "Alter") beeinflusst?**

```
# K ist 'within' im Bezug zu Vpn: daher (K|Vpn)
with(stimm, table(Vpn, K))
```

```
##      K
## Vpn ba pa
##  A  8  8
##  B  8  8
##  C  8  8
##  D  8  8
##  E  8  8
##  F  8  8
##  G  8  8
##  H  8  8
```

```
# Alter ist 'between' im Bezug zu Vpn daher (1|Vpn).
with(stimm, table(Vpn, Alter))
```

```
##      Alter
## Vpn Erw Kind
##  A   0  16
##  B   0  16
##  C   0  16
##  D  16   0
##  E  16   0
##  F  16   0
##  G   0  16
##  H   0  16
```

```
# (K|Vpn) enthält aber bereits (1|Vpn)
# Weil (K|Vpn) bedeutet:
# random-slope und random-intercept berechnen;
# und (1|Vpn) bedeutet: nur random-intercept berechnen.
# Daher (K|Vpn) und NICHT (K|Vpn) + (1|Vpn)
```

```

# Zusätzlich für zwei fixed factors:
# K * Alter (bedeutet K, Alter, und deren Interaktion).
# K * Alter ist dasselbe wie: K + Alter + K:Alter
stimm.lmer = lmer(vot ~ K * Alter + (K | Vpn), data = stimm)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.0304774
## (tol = 0.002, component 1)

# Oder
stimm.lmer = lmer(vot ~ K + Alter + K:Alter + (K | Vpn), data = stimm)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.0304774
## (tol = 0.002, component 1)

# 2.
dim(int)

## [1] 120  5

```

10 Sprecher (Vpn), davon Männer und Frauen (Faktor G) produzierten zwei Vokale (Faktor V), und die Intensität wurde gemessen (db). Die Vokale kamen in verschiedenen Wörtern vor. **Inwiefern wird die Intensität vom Vokal und vom Geschlecht beeinflusst?**

```

# Fixed factors: V, G
# Random factors: Wort, Vpn
#
# V ist 'within' im Bezug zu Wort. Daher (V|Wort)
with(int, table(Wort, V))

##      V
## Wort a i
## w1 10 10
## w2 10 10
## w3 10 10
## w4 10 10
## w5 10 10
## w6 10 10

# V ist 'within' im Bezug zu Vpn. Daher (V|Vpn)
with(int, table(Vpn, V))

##      V
## Vpn  a i
## S1   6 6
## S10  6 6
## S2   6 6
## S3   6 6
## S4   6 6
## S5   6 6
## S6   6 6

```

```

##    S7  6  6
##    S8  6  6
##    S9  6  6

# G ist 'within' im Bezug zu Wort. Daher (G|Wort)
with(int, table(Wort, G))

##      G
## Wort  f  m
##   w1 10 10
##   w2 10 10
##   w3 10 10
##   w4 10 10
##   w5 10 10
##   w6 10 10

# G ist 'between' im Bezug zu Vpn. Daher (1|Vpn)
with(int, table(Vpn, G))

##      G
## Vpn   f  m
##   S1   0 12
##  S10 12  0
##   S2   0 12
##   S3   0 12
##   S4   0 12
##   S5   0 12
##   S6  12  0
##   S7  12  0
##   S8  12  0
##   S9  12  0

```

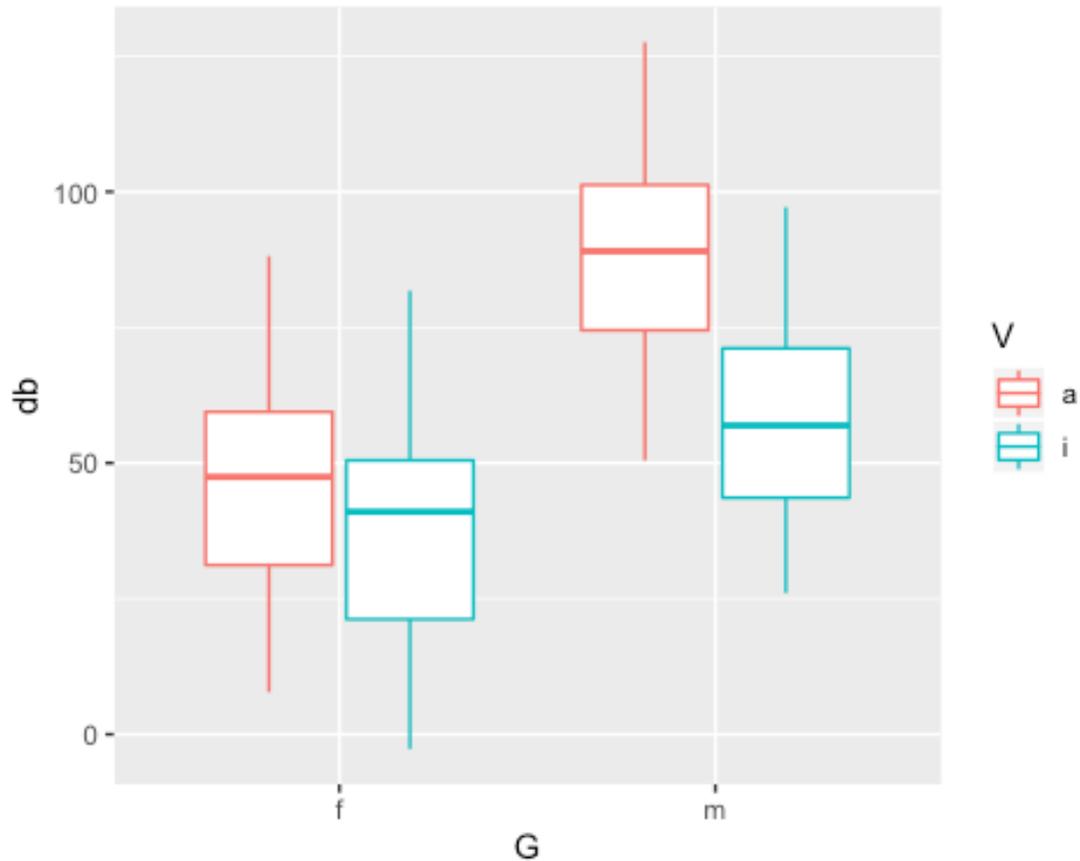
Zusammen:  $(V+G|Wort) + (V|Vpn)$  (und NICHT  $(V|Wort) + (G|Wort) + (V|Vpn) + (1|Vpn)$ ), denn die Faktoren, deren Einfluss ausgeklammert werden soll (hier also Wort und Vpn), sollen nur einmal in der Formel auftauchen.

Daher:

```

ggplot(int) +
  aes(y = db, x = G, col = V) +
  geom_boxplot()

```



```
int.lmer = lmer(db ~ V * G + (V + G | Wort) + (V | Vpn), data = int)
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.0391176
## (tol = 0.002, component 1)
```

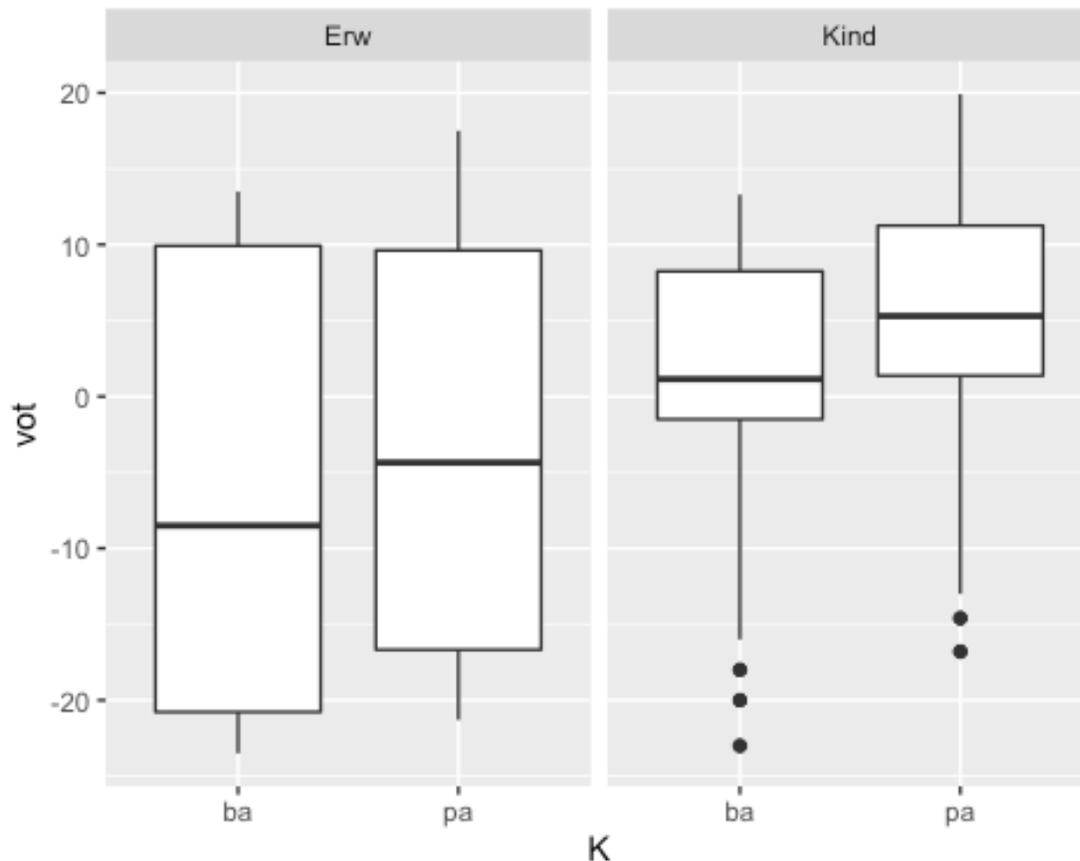
## 6. Die Prüfstatistik

Die step()-Funktion

- (a) vereinfacht das Modell wenn möglich.
- (b) prüft, ob die Faktoren in dem vereinfachten Modell signifikant sind.

VOT wurde in /ba, pa/ von Kindern und Erwachsenen produziert. **Inwiefern wird VOT von der Artikulationsstelle (Faktor K) und/oder von der Sprechergruppe (ob Kind oder Erwachsenen: Faktor Alter) beeinflusst?**

```
# Bild
ggplot(stimm) +
  aes(y = vot, x = K) +
  geom_boxplot() +
  facet_wrap(~ Alter)
```



```
# Mixed Model
```

```
stimm.lmer = lmer(vot ~ K * Alter + (K | Vpn), data = stimm)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.0304774
## (tol = 0.002, component 1)
```

```
stimm.step = step(stimm.lmer)
```

```
stimm.step
```

```
## Backward reduced random-effect table:
```

```
##
##           Eliminated npar logLik   AIC   LRT Df Pr(>Chisq)
## <none>                8 -297.30 610.59
## K in (K | Vpn)         0   6 -299.82 611.64 5.0428 2  0.08035 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Backward reduced fixed-effect table:
```

```
## Degrees of freedom method: Satterthwaite
```

```
##
##           Eliminated Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## K:Alter      1      1.554   1.554     1  6.0424  0.3523 0.5743198
## Alter        2      3.305   3.305     1  5.9996  0.7502 0.4197060
## K            0  180.853  180.853     1  6.9995 41.0544 0.0003648 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Model found:
## vot ~ K + (K | Vpn)
```

## Teil (a): Modell-Vereinfachung

Brauchen wir wirklich  $K + \text{Alter} + K:\text{Alter} + (K|Vpn)$ ?

```
# Backward reduced random-effect table:
#
#           Eliminated npar  LogLik    AIC    LRT Df Pr(>Chisq)
# <none>                8 -297.30 610.59
# K in (K | Vpn)         0   6 -299.82 611.64 5.0439 2    0.0803 .
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

1. Random effects ist fast signifikant ( $p = 0.08$ ). Daher wird  $(K|Vpn)$  behalten. Daher '0' bei 'eliminated', d.h.  $(K|Vpn)$  wurde nicht durch  $(1|Vpn)$  ersetzt.
2. Bei den Fixed effects.  $K:\text{Alter}$  und  $\text{Alter}$  sind nicht signifikant, und wurden aus dem Modell entfernt. Aber Faktor  $K$  bleibt ('0').

```
# Backward reduced fixed-effect table:
# Degrees of freedom method: Satterthwaite
#
#           Eliminated  Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
# K:Alter          1   1.547   1.547    1     6  0.3511 0.5751381
# Alter            2   3.305   3.305    1     6  0.7501 0.4197122
# K                 0 180.861 180.861    1     7 41.0563 0.0003647 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Das vereinfachte Modell aus 1. und 2. war:

```
get_model(stimm.step)
## Linear mixed model fit by REML ['lmerModLmerTest']
## Formula: vot ~ K + (K | Vpn)
## Data: stimm
## REML criterion at convergence: 604.3563
## Random effects:
## Groups   Name                Std.Dev. Corr
## Vpn      (Intercept) 13.064
##          Kpa           1.365  -0.24
## Residual                    2.099
## Number of obs: 128, groups: Vpn, 8
## Fixed Effects:
## (Intercept)                Kpa
## -2.432                    3.900
```

Dies entspricht:

```
lmer(vot ~ K + (K | Vpn), data = stimm)

## Linear mixed model fit by REML ['lmerModLmerTest']
## Formula: vot ~ K + (K | Vpn)
## Data: stimm
## REML criterion at convergence: 604.3563
## Random effects:
## Groups Name Std.Dev. Corr
## Vpn (Intercept) 13.064
## Kpa 1.365 -0.24
## Residual 2.099
## Number of obs: 128, groups: Vpn, 8
## Fixed Effects:
## (Intercept) Kpa
## -2.432 3.900
```

Dies wird auch ganz unten berichtet:

```
stimm.step

## Backward reduced random-effect table:
##
## Eliminated npar logLik AIC LRT Df Pr(>Chisq)
## <none> 8 -297.30 610.59
## K in (K | Vpn) 0 6 -299.82 611.64 5.0428 2 0.08035 .
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Backward reduced fixed-effect table:
## Degrees of freedom method: Satterthwaite
##
## Eliminated Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## K:Alter 1 1.554 1.554 1 6.0424 0.3523 0.5743198
## Alter 2 3.305 3.305 1 5.9996 0.7502 0.4197060
## K 0 180.853 180.853 1 6.9995 41.0544 0.0003648 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Model found:
## vot ~ K + (K | Vpn)
```

## Teil B: Die Prüfstatistik

Die Prüfstatistik für das vereinfachte Modell ist somit:

```
anova(get_model(stimm.step))

## Type III Analysis of Variance Table with Satterthwaite's method
## Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## K 180.85 180.85 1 6.9995 41.054 0.0003648 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Dies ist das gleiche, als hätte man das Folgende ausgeführt:

```
stimm2.lmer = lmer(vot ~ K + (K | Vpn), data = stimm)
anova(stimm2.lmer)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##      Sum Sq Mean Sq NumDF  DenDF F value    Pr(>F)
## K 180.85  180.85      1 6.9995  41.054 0.0003648 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Daher die Schlussfolgerung: \*\*VOT wurde signifikant von der Artikulationsstelle ( $F[1,7] = 41.1, p < 0.001$ ), jedoch nicht vom Alter beeinflusst, und es gab auch keine Interaktion zwischen diesen Faktoren.

Ein ähnliches, aber nicht identisches Ergebnis ( $F$  wird kleiner,  $p$  daher größer):

```
anova(stimm.lmer)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##      Sum Sq Mean Sq NumDF  DenDF F value    Pr(>F)
## K      146.890 146.890      1 6.0424 33.3133 0.001152 **
## Alter      2.244   2.244      1 6.1555  0.5089 0.501749
## K:Alter    1.554   1.554      1 6.0424  0.3523 0.574320
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Die Methode mit `step()` und `anova(get_model(MODELL.step))` ist vorzuziehen; leider scheitert sie bei manchen Modellen (Fehlermeldung), und muss dann durch die Anwendung von `anova()` auf das ursprünglich erdachte Modell ersetzt werden.

## 7. Post-hoc Tests

Wenn fixed factors interagieren, sollen ähnlich wie in der Varianzanalyse post-hoc tests durchgeführt werden.

```
head(int)
```

```
##      db V G Vpn Wort
## 1 100 a m S1  w1
## 2  70 a m S2  w1
## 3  90 a m S3  w1
## 4  60 a m S4  w1
## 5  80 a m S5  w1
## 6  50 a f S6  w1
```

10 Sprecher produzierten 6 verschiedene Wörter entweder mit einem /i/ oder /a/ Vokal (Faktor V). Die Intensität wurde (db) wurde pro Vokal gemessen. Wir hatten festgestellt, dass V und G 'within' sind im Bezug zu Wort und nur V ist 'within' im Bezug zu Vpn. Daher hatten wir:

```
int.lmer = lmer(db ~ V * G + (V + G | Wort) + (V | Vpn), data = int)
```

```

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.0391176
## (tol = 0.002, component 1)

int.step = step(int.lmer)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.0029403
## (tol = 0.002, component 1)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.0117045
## (tol = 0.002, component 1)

## boundary (singular) fit: see ?isSingular

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.0117045
## (tol = 0.002, component 1)

## boundary (singular) fit: see ?isSingular

int.step

## Backward reduced random-effect table:
##
##           Eliminated npar  logLik      AIC      LRT Df Pr(>Chisq)
## <none>                14 -315.66  659.32
## G in (V + G | Wort)    1  11 -315.87  653.73   0.41  3   0.9375
## V in (V | Wort)       2   9 -316.45  650.89   1.16  2   0.5596
## V in (V | Vpn)        0   7 -407.66  829.31 182.42  2   <2e-16
## (1 | Wort)           0   8 -501.43 1018.85 369.96  1   <2e-16
##
## <none>
## G in (V + G | Wort)
## V in (V | Wort)
## V in (V | Vpn)      ***
## (1 | Wort)         ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Backward reduced fixed-effect table:
## Degrees of freedom method: Satterthwaite
##
##           Eliminated Sum Sq Mean Sq NumDF  DenDF F value  Pr(>F)
## V:G          0 31.975  31.975    1 8.0027  7.5304 0.02528 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Model found:
## db ~ V + G + (V | Vpn) + (1 | Wort) + V:G

```

```

# Das vereinfachte Modell:
#Model found:
# db ~ V + G + (V | Vpn) + (1 | Wort) + V:G
anova(get_model(int.step))

## Type III Analysis of Variance Table with Satterthwaite's method
##      Sum Sq Mean Sq NumDF  DenDF F value    Pr(>F)
## V    126.684 126.684     1  8.0027 29.8353 0.0005993 ***
## G     47.307  47.307     1  7.9988 11.1413 0.0102660 *
## V:G   31.975  31.975     1  8.0027  7.5304 0.0252779 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Es gibt also eine signifikante Interaktion. Daher:

```

library(emmeans)
int.ph = pairs(emmeans(get_model(int.step), ~ V:G))
int.ph

## contrast estimate SE df t.ratio p.value
## a,f - i,f 10.01 5.21 8.0 1.922 0.2918
## a,f - a,m -40.19 10.14 8.0 -3.965 0.0174
## a,f - i,m -9.97 9.74 10.6 -1.024 0.7398
## i,f - a,m -50.20 9.74 10.6 -5.156 0.0017
## i,f - i,m -19.98 9.32 8.0 -2.143 0.2187
## a,m - i,m 30.22 5.21 8.0 5.803 0.0018
##
## P value adjustment: tukey method for comparing a family of 4 estimates

```

Auch bei dem `pairs()`-Befehl wird die Korrektur für die Mehrfachtestung automatisch durchgeführt. Je nach Modell kann das mal eine Bonferroni-Korrektur sein, und mal andere Methoden (z.B. wie hier "Tukey"). Daher dies immer mitberichten. Wie von anderen post-hoc-Tests bekannt, können wir die Funktion `phsel()` benutzen, um nur die relevanten Vergleiche auszuwählen. Auch hier wird die Korrekturmethode dargestellt:

```

phsel(int.ph, 1)

## P value adjustment method: tukey

##      df      t  prob.adj sig
## a,f - i,f  8 1.921919 0.291821555
## a,m - i,m  8 5.802752 0.001809492 **

#oder auch einfach
phsel(int.ph)

## P value adjustment method: tukey

##      df      t  prob.adj sig
## a,f - i,f  8 1.921919 0.291821555
## a,m - i,m  8 5.802752 0.001809492 **

```

```
#und
phsel(int.ph, 2)

## P value adjustment method: tukey

##           df           t   prob.adj sig
## a,f - a,m   8 -3.965377 0.01744606   *
## i,f - i,m   8 -2.143433 0.21872497
```

Post-hoc Tukey korrigierte t-tests zeigten signifikante Unterschiede zwischen /i/ und /a/ in Männern ( $p < 0.01$ ), jedoch nicht in Frauen, und signifikante Unterschiede zwischen Männern und Frauen in /a/ ( $p < 0.05$ ), jedoch nicht in /i/.

## 8. Zusammenfassung:

Bedingung: Sie haben relativ viele Beobachtungen in NICHTGEMITTELT.df

1. Sie haben eine numerische abhängige Variable (AV)
2. Sie haben mehrere unabhängige Variablen (UV), von denen mindestens eine ausgeklammert werden soll (die Random Factors (RV), die anderen heißen Fixed Factors (FF))

Dann können Sie, bzw. müssen Sie (müssen, wenn mehr als 2 RFs da sind, oder wenn Sie fehlende Werte haben), ein Mixed Model anwenden:

```
NG.lmer=lmer(AV~FF1 * FF2 ... +
             (FF1+FF2|RF1) +
             (FF1+FF2|RF2),
             data=NICHTGEMITTELT.df)
```

...wenn FF1 und FF2 jeweils "within" in bezug zu RF1 bzw. RF2 sind; wäre z.B. FF1 in Bezug zu RF1 "between" -> (1|RF1)

Dann: NG.step=step(NG.lmer)

Wenn das klappt: anova(get\_model(NG.step)), ansonsten zur Not anova(NG.lmer) ggf. (bei gegebener Ingeraktion): post-hoc testen:

```
NG.ph = pairs(emmeans(get_model(NG.step), ~FF1:FF2))
```

```
oder, falls step() scheitert: NG.ph = pairs(emmeans(NG.lmer), ~FF1:FF2)
```

Paare auswählen: phsel(NG.ph,1) und phsel(NG.ph,2)

Dann Bericht schreiben ...

---

## Nachwort:

Wir müssen immer beachten, dass wir genügend Beobachtungen haben. Was "genügend" ist, hängt auch von der Anzahl der Faktoren (fixed und random) ab: bei nur einem Fixed und nur einem Random Factor geht man als Faustregel von mindestens 60 Beobachtungen aus. Aber: Nicht immer wird der `lmer()`-Befehl scheitern, wenn man eigentlich laut dieser Faustregel zuwenige Beobachtungen hat. Z.B. oben hatten wir

```
soa.lmer = lmer(F1 ~ Pos + (Pos | Vpn) + (Pos | W), data = soa)
## boundary (singular) fit: see ?isSingular
## Warning: Model failed to converge with 1 negative eigenvalue: -2.3e-01
```

...und das funktioniert ohne Probleme, aber wir bekommen eine Warnmeldung: Model failed to converge!

Das ist kein Wunder, denn soa hat nur 18 Beobachtungen:

```
soa
##      Vpn      W      Pos      F1
## 1   s1  Bart  final  493.5
## 2   s1  Pfad  final  532.5
## 3   s1  Start  final  514.5
## 4   s1  Bart  medial  482.0
## 5   s1  Pfad  medial  501.0
## 6   s1  Start  medial  497.0
## 7   s2  Bart  final  533.5
## 8   s2  Pfad  final  568.5
## 9   s2  Start  final  579.5
## 10  s2  Bart  medial  498.0
## 11  s2  Pfad  medial  551.0
## 12  s2  Start  medial  533.0
## 13  s3  Bart  final  531.5
## 14  s3  Pfad  final  561.5
## 15  s3  Start  final  571.5
## 16  s3  Bart  medial  477.0
## 17  s3  Pfad  medial  518.0
## 18  s3  Start  medial  535.0
```

... und das sind sicherlich zu wenige für ein Mixed Model, insbesondere bei drei unabhängigen Variablen (1 x FF, 2 x RF). Eine längere Warnmeldung erhalten wir erst bei Anwendung von `step()`:

```
step(soa.lmer)
## boundary (singular) fit: see ?isSingular
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : unable to evaluate scaled gradient
```

```

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge: degenerate Hessian with 1
## negative eigenvalues

## Warning: Model failed to converge with 1 negative eigenvalue: -1.8e-01

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : unable to evaluate scaled gradient

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge: degenerate Hessian with 1
## negative eigenvalues

## Warning: Model failed to converge with 1 negative eigenvalue: -1.8e-01

## boundary (singular) fit: see ?isSingular

## Backward reduced random-effect table:
##
##           Eliminated npar  logLik    AIC      LRT Df Pr(>Chisq)
## <none>                9 -68.498 155.00
## Pos in (Pos | W)      1  7 -68.511 151.02  0.0263  2  0.9869227
## Pos in (Pos | Vpn)   2  5 -69.544 149.09  2.0666  2  0.3558291
## (1 | W)                0  4 -75.934 159.87 12.7788  1  0.0003506
## (1 | Vpn)              0  4 -75.970 159.94 12.8516  1  0.0003372
##
## <none>
## Pos in (Pos | W)
## Pos in (Pos | Vpn)
## (1 | W)                ***
## (1 | Vpn)              ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Backward reduced fixed-effect table:
## Degrees of freedom method: Satterthwaite
##
##           Eliminated Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## Pos           0 4818.3  4818.3     1    12  38.898 4.339e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Model found:
## F1 ~ Pos + (1 | W) + (1 | Vpn)

```

Da wir zuwenige Beobachtungen für ein so kompliziertes Mixed Model haben, sollten wir zunächst versuchen, die Struktur unserer Formel zu vereinfachen, und dabei zunächst bei den Random Factors beginnen. Zunächst könnten wir versuchen, Pos nicht als within-, sondern wie ein between-subjects-factor, bzw. between-words-factor zu behandeln (und genau das hat übrigens `step()` getan), also:

```

soa.lmer = lmer(F1 ~ Pos + (1 | Vpn) + (1 | W), data = soa)

soa.step = step(soa.lmer)
anova(get_model(soa.step))

## Type III Analysis of Variance Table with Satterthwaite's method
##      Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## Pos  4818.3  4818.3     1    12  38.898 4.339e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Wie Sie sehen können, bekommen wir auf diese Weise keine Warnmeldung! Aber wir erkaufen uns dies mit einem eigentlich etwas zu unterkomplexen Modell (denn eigentlich ist Pos sowohl in Bezug zu den Wörtern als auch zu den Versuchspersonen 'within'). Wir wir auch schon vorher gesehen haben, ist dieser Schritt der Modellvereinfachung aber genau das, was `step()` für uns automatisch macht: dasjenige Modell zu finden, das noch konvergiert, und dessen Faktoren (das umfasst sowohl fixed als auch random factors) einen nicht vernachlässigbaren Effekt auf die vorliegenden Daten hatten. Somit folgen wir also dem Grundsatz, das komplexeste Modell zu finden, das errechenbar ist, und dies soweit zu optimieren, dass Faktoren mit wenig Einfluss gleich weggelassen werden (womit wir dem Ideal des Prinzips der einfachsten Erklärung zu folgen versuchen).

Der gleiche Vorwurf eines möglicherweise zu unterkomplexen Modells gilt - und zwar sogar in stärkerem Maße - allerdings auch für die andere Alternative für die vorliegenden Daten mit wenigen Beobachtungen, nämlich die Durchführung einer Varianzanalyse; bedenken Sie hierbei aber auch, dass wir hierbei die Anzahl der auszuwertenden Beobachtungen durch mitteln sogar noch massiv verringern: zuerst müssen wir ja mitteln, da eine ANOVA nur einen einzigen Random Factor erlaubt, z.B. also 'mitteln über Wort'; in diesem Fall verbleiben dann für den Vergleich aber nur  $3(Vpn) \cdot 2(Pos) = 6$  Werte.

Genaugenommen könnten wir dann sogar einen gepaarten t-Test durchführen:

```

soa.m = soa %>%
  group_by(Vpn, Pos) %>%
  summarise(F1 = mean(F1))

soa.diff = soa.m %>%
  group_by(Vpn) %>%
  summarise(F1 = diff(F1))

shapiro.test(soa.diff$F1)

##
## Shapiro-Wilk normality test
##
## data:  soa.diff$F1
## W = 0.99903, p-value = 0.9404

t.test(soa.diff$F1)

```

```
##  
## One Sample t-test  
##  
## data: soa.diff$F1  
## t = -4.5932, df = 2, p-value = 0.04428  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -63.374837 -2.069608  
## sample estimates:  
## mean of x  
## -32.72222
```

Wir hätten in diesem Fall aber nur 2 Freiheitsgrade, da wir lediglich 3 (!) Werte hätten, bei denen wir überprüfen, ob sie signifikant von 0 abweichen.

Mit anderen Worten: mit so wenigen Daten kann man keine komplexen Fragestellungen beantworten! Bedenken Sie dies bitte immer in Ihren eigenen Experimentdesigns, und achten Sie auf möglichst viele Versuchspersonen, die möglichst viele Beobachtungstypen mit idealerweise vielen Wiederholungen produzieren!

---

Weitere Anmerkungen: Die Fixed Factors in MMs können kategorial sein (also Faktoren mit Stufen), können aber genauso auch numerisch sein (also wie bei `lm()` )

---

Drittens: Es gibt auch den Fall, dass die abhängige Variable nicht numerisch, sondern kategorial ist. Der Befehl heißt dann `glmer()` statt `lmer()` (`glmer()` von *generalized linear mixed effects regression models*) (vgl. `lm()` und `glm()` ).

`glmer()` kommt in diesem Seminar aber nicht vor!! Bedenken Sie aber, dass das einem Mixed Model mit logistischer Regression entspricht. Sollten Sie ein solches Modell für Ihre Abschlussarbeit benötigen, melden Sie sich bitte bei einem der Herren Harrington / Reubold.