Generalised linear mixed models (GLMM) und die logistische Regression

Jonathan Harrington

Die R-Befehle: glmm.txt

```
library(lme4)
anna = read.table(paste(pfad, "anna.txt", sep="/"))
annalang = read.table(paste(pfad, "annalang.txt", sep="/"))
lax = read.table(paste(pfad, "lax.txt", sep="/"))
```

Die abhängige Variable

Wie im Mixed Model gibt es in einem GLMM mindestens einen Random Factor, mit dem die Variabilität ausgeklammert werden kann.

In einem MM ist der Response (abhängige Variable) numerisch; in einem GLMM binär und kategorial (wie in der logistischen Regression).

zB

ja ja nein ja nein...
"b", "d", "d", "d", "b", "b"...
"rot", "rot", "grün", "rot"...

aber nicht

"b", "d", "g", "g", "d"...
"rot", "gelb", "grün", "rot",
"schwarz..."

Fixed factors, random factors

Ähnlich wie in einem MM wird differenziert prinzipiell zwischen **fixed factors** (sollen geprüft werden) und **random factors** (sollen ausgeklammert werden)

MM: Reaktionszeiten (RT) wurden von 10 Versuchsperson in 200 verschiedenen Wörtern gemessen. Die Wörter unterschieden sich in der Wortlänge (1 vs 2 Silben) und Vokal (/i, u, a/). Inwiefern haben Wortlänge und Vokal einen Einfluss auf die RTs?

GLMM: 10 Hörer mussten in 200 Wörtern entscheiden ob das Wort ein Funktions- oder Inhaltswort war. Die Wörter unterschieden sich in der Wortlänge (1 vs 2 Silben) und Vokal (/i, u, a/). Inwiefern haben Wortlänge und Vokal einen Einfluss auf die Entscheidungen?

	Response	Fixed factor(s)	Random factor(s)
MM:	Reaktionszeiten	Wortlänge, Vokal	Sprecher, Wort
GLMM:	I vs F Entscheidungen		Hörer, Wort

Parameter-Einschätzung: MM und GLMM

In einem MM und GLMM werden zwei Parameter *m* (Neigung) und *k* (Intercept) eingeschätzt, um den Abstand zwischen tatsächlichen und eingeschätzten Werten zu minimieren. Für MM ist die Formel dafür ähnlich wie in der linearen Regression für GLMM wie in der logistischen Regression

MM

y: zB die Reaktionszeiten, $\hat{y}$ deren Einschätzung

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

GLMM

p: zB Proportion von "Inhalt"-Antworten, $\hat{p}$ deren Einschätzung

$$\hat{p} = \frac{e^{(mx+k)}}{1+e^{(mx+k)}}$$

Anders als die lineare oder lineare Regression wird getrennt pro Stufe des Random Faktors (zB pro Sprecher) einen *k* und ggf. einen *m* berechnet (also sprecher- und/oder wortspezifische Berechung dieser Parameter, wenn Vpn und Wort als random factors genannt werden)

Parameter-Einschätzung und Random Factors: MM und GLMM

GLMM: 10 Hörer mussten in 200 Wörtern entscheiden ob das Wort ein Funktions- oder Inhaltswort war. Die Wörter unterschieden sich in der Wortlänge (1 vs 2 Silben) und Vokal (/i, u, a/). Inwiefern haben Wortlänge und Vokal einen Einfluss auf die Entscheidungen?

Zwei Möglichkeiten den Sprecher (oder irgeneinen Factor) als Random festzulegen

1. Berechnung von einem sprecherspezifischen *k*

Die Sprechervariation wird herausgerechnet, **ohne die Stufenaufteilung zu berücksichtigen**

R syntax: (1 | Sprecher)

2. Berechnung sprecherspezifische *k* und m

die Sprechervariation wird getrennt pro Stufe des **genannten** fixed factors herausgerechnet

(1+Wortlänge | Sprecher)

= die Sprechervariation wäre getrennt aus einsilbigen und zweisilbigen Wörtern ausgeklammert.

Fixed factors: MM und GLMM

In einem MM (und RM-Anova) wird geprüft, ob ein, oder mehrere Faktoren, den Response signifikant beeinflussen. zB

MM: wird Reaktionszeiten vom Wortlänge (1 vs 2 Silben) und Vokal (/i, u, a/) signifikant beeinflusst – und interagieren diese Faktoren?

GLMM: liefert Ergebnisse in dem **immer nur zwei Stufen miteinander verglichen werden können**

Ein Hörer muss entscheiden, ob ein Wort ein Logatom ist oder nicht (ja/nein). Wird diese Entscheidung von der Wortlänge (1 oder 2 Silben) oder Vokal (/i, u, a/) signifikant beeinflusst?

Man kann prüfen, ob die Entscheidungen beeinflusst werden von:

1 vs 2 Silben. /i/ vs /u/ /u/ vs /a/

einsilbiger /i/ vs zwesilbiger /i/ einsilbiger /u/ vs zwesilbiger /a/

also immer nur paarweise Stufenkombinationen

GLMM: Ein fixed factor mit 2 Stufen

(Daten von Anna Rühl). Ein Spracherkennungssystem musste entscheiden, ob ein akustisches Signal prä- oder postaspiriert war. Die Aspirationssegmente wurden von verschiedenen Versuchspersonen produziert. Es wurde pro Segment ermittelt, ob richtig (Correct) oder falsch (Incorrect) erkannt wurde. Unterscheidet sich die Verteilung Correct:Incorrect in pre- vs. postaspirierte Segmente?

Es gibt zwei (und nur diese 2) Möglichkeiten den Response vorzubereiten:

Matrix
head(annalang)

Vektor
head(anna)

Fixed Factor:
Aspiration (2 Stufen: pre vs post)

Random Factor:
Sprecher

GLMM und Basis-Stufe

In einem GLMM wird immer der Unterschied zwischen einer Basis-Stufe und (paarweise) allen anderen Stufen des Faktors geprüft.

Die Basis-Stufe wird durch levels() vermittelt und kann durch relevel() geändert werden:

1. with(anna, levels(Asp))
   `"post" "pre"`

Asp2 = with(anna, relevel(Asp, "pre"))

levels(Asp2)

2. `"pre" "post"`

3. with(anna, levels(Cons))
   "k" "p" "t"

Ein GLMM prüft paarweise die Entfernung (und Signifikanz) der anderen Stufen zur Basis also:

(1) von pre zu post

(2) von post zu pre

(3) (i) von /p/ zu /k/ sowie (getrennt)
(ii) von /t/ zu /k/

lmer(... family = "binomial")

entweder:

```
annalang.lmer = lmer(Response ~ Asp + (1|Sp), family="binomial", data = annalang)

print(annalang.lmer, corr=F)
```

oder:

```
anna.lmer = lmer(cbind (Incorrect, Correct) ~ Asp + (1|Sp), family="binomial", data = anna)
print(anna.lmer, corr=F)
```

```
            Estimate Std. Error z value Pr(>|z|)
Asppre       -1.1279     0.1613  -6.992 2.72e-12 ***
```

Asppre = Die Stufe pre des fixed factors Aspiration (die Basis ist die andere Stufe (post) und wird nicht gezeigt)

Die Entfernung in Standardabweichungen der Normalverteilung (z-scores) zwischen pre und der Basis-Stufe (= post).

Ein GLMM mit Aspiration als fixed factor und Sprecher als random factor zeigte einen signifikanten Einfluss von Prä- vs. Postaspiration auf die Erkennungsrate ($z = 7.0$, $p < 0.001$).

GLMM: Ein kategorialer Faktor und 3 Stufen

Inwiefern wurde die Erkennungsrate von der Artikulationsstelle beeinflusst?

```
c.lmer = lmer(Response ~ Cons + (1|Sp), family="binomial", data = annalang)
```

```
print(c.lmer, corr=F)
```

```
Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.8787     0.1455   6.037 1.57e-09 ***
Consp         0.5327     0.1969   2.705  0.00682 **
Const        -0.2722     0.1789  -1.521  0.12821
```

/p/ ist 2.705 Standardabweichungen von /k/ entfernt

/t/ ist 1.521 Standardabweichungen von /k/ entfernt

GLMM: Ein kategorialer Factor und 3 Stufen

```
Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
Consp         0.5327     0.1969   2.705  0.00682 **
Const        -0.2722     0.1789  -1.521  0.12821
```

Um die /t/-/p/ Entfernung zu vermitteln müsste Cons mit Basis /p/ oder /t/ kodiert werden.

```
Cons2 = with(annalang, relevel(Cons, "p"))
d.lmer = lmer(Response ~ Cons2 + (1|Sp), family="binomial", data = annalang)
print(d.lmer, corr=F)
```

```
Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
Cons2k       -0.5327     0.1969  -2.705  0.00682 **
Cons2t       -0.8048     0.1930  -4.170 3.04e-05 ***
```

Ein GLMM mit fixed factor Artikulationsstelle und mit Sprecher als random factor zeigte signifikante Unterschiede zwischen /k/ und /p/ ($z = 2.7$, $p < 0.01$), und /p/ und /t/ ($z = 4.2$, $p < 0.001$). Der Unterschied zwischen /k/ und /t/ war nicht signifikant.

Parameter Berechnung und random factors

```
c.lmer = lmer(Response ~ Cons + (1|Sp), family="binomial", data = annalang)
```

In diesem Modell wird wegen (1|Sp) getrennt pro Sprecher ein Intercept berechnet (Ausklammerung der Sprechervariabilität, ohne die Aufteilung von einem fixed factor in Stufen zu berücksichtigen. Aus diesem Grund ist k unterschiedlich, *m* dasselbe):

coef(c.lmer)

```
    (Intercept)     Consp      Const
EBJ   0.6993940 0.5326656 -0.2721793
GGU   0.9919618 0.5326656 -0.2721793
JEI   0.6556235 0.5326656 -0.2721793
```

Hier wäre m und k pro Sprecher berechnet: die Sprechervariabilität wird pro Stufe von Cons ausgeklammert

```
c.lmer2 = lmer(Response ~ Cons + (1 + Cons|Sp),
family="binomial", data = annalang)
```

und es gibt unterschiedliche k und m Werte pro Sprecher:

```
$Sp
    (Intercept)       Consp       Const
EBJ   0.1695354  2.17175146 -0.10447699
GGU   0.6353814  1.17415791 -0.21263395
JEI   1.3797218  0.09419544 -2.21168461
```

Parameter Berechnung und random factors

Grundsätzlich soll das einfachere Modell (1|Random) verwendet werden, es sei denn (a) sich die beiden Modelle signifikant unterscheiden und (b) der AIC-Wert bei (1+Fixed|Random) kleiner wird (siehe ppt zur Regression). Dies kann mit anova() geprüft werden:

```
c.lmer = lmer(Response ~ Cons + (1|Sp),
family="binomial", data = annalang)

c.lmer2 = lmer(Response ~ Cons + (1 + Cons|Sp),
family="binomial", data = annalang)
```

anova(c.lmer, c.lmer2)

```
        Df     AIC     BIC  logLik  Chisq Chi Df Pr(>Chisq)
c.lmer   4 1016.71 1035.75 -504.35
c.lmer2  9  992.61 1035.47 -487.31 34.093      5  2.282e-06 ***
```

(daher wird (1 + Cons|Sp) bevorzugt)

```
library(lme4)

lax = read.table(paste(pfad, "lax.txt", sep="/"))
```

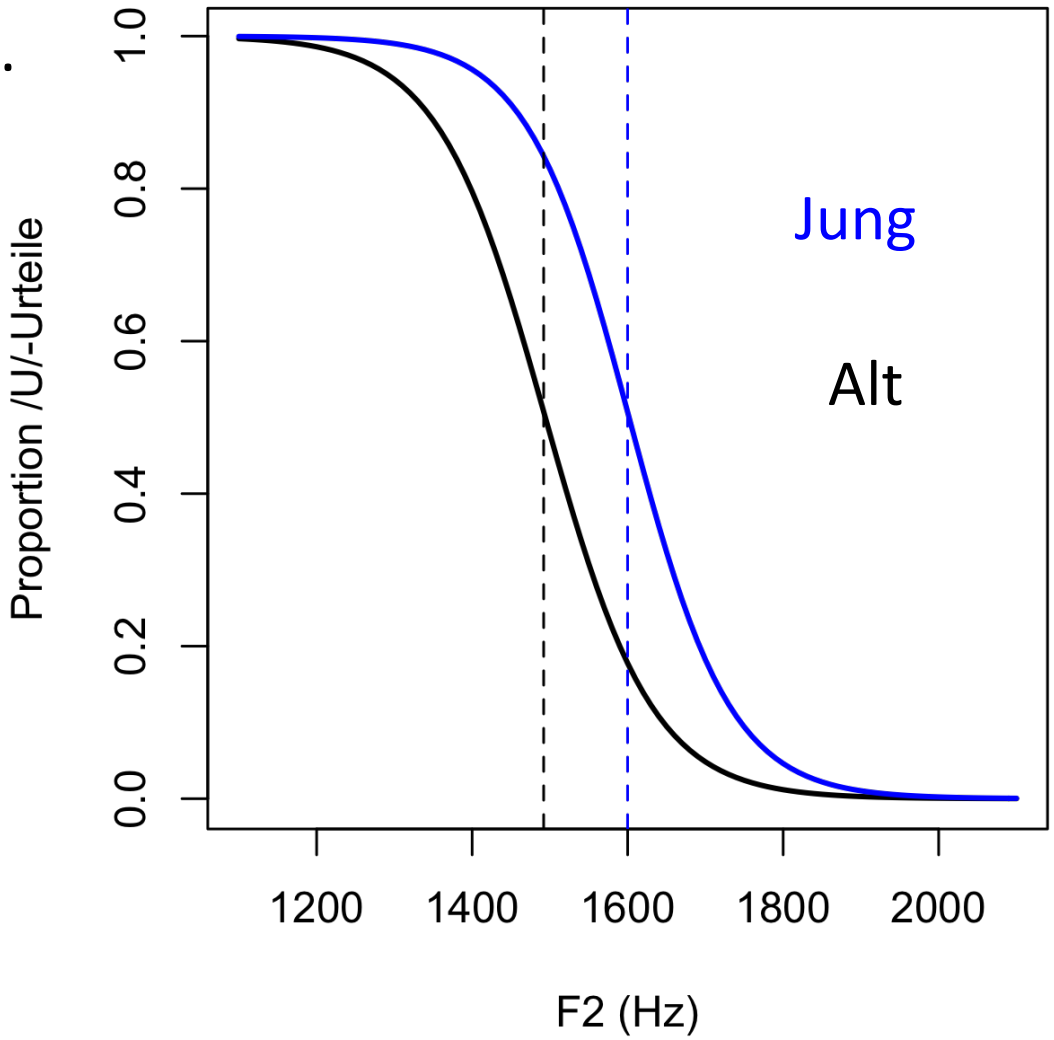
GLMM, Psychometrische Kurven, Umkipppunkte

Ein /ɪ-ʊ/ Kontinuum wurde durch Herabsenken von F2 synthetisiert

Es gab 2 Gruppen von Versuchspersonen: alt und jung.

Die Versuchspersonen mussten pro Stimulus entscheiden: war es /I/ oder /ʊ/ (forced-choice test)?

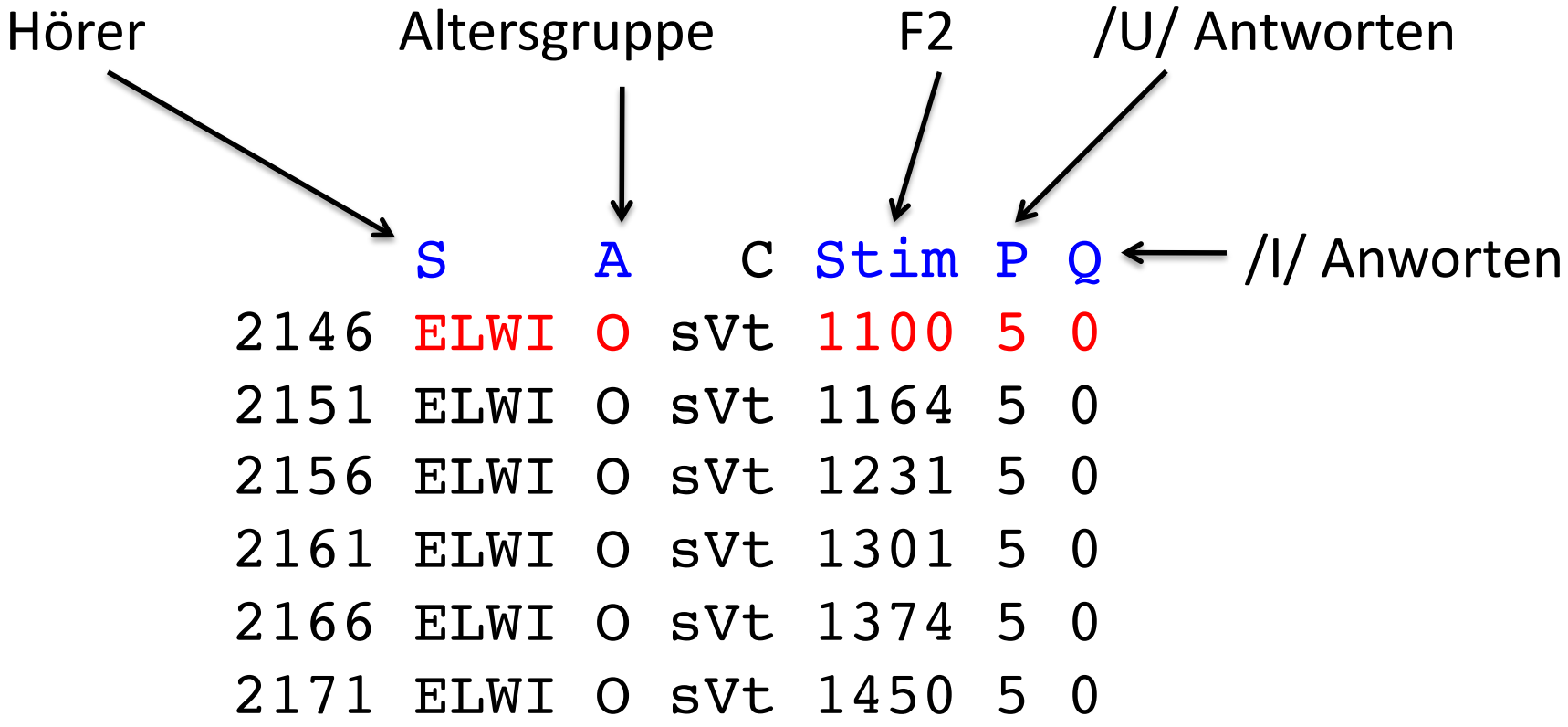
Inwiefern wird der Umkipppunkt und/oder die Neigung zwischen /I/ und /ʊ/ vom Alter beeinflusst?



GLMM, Psychometrische Kurven, Umkipppunkte

Data-Frame lax

Die relevanten Variablen



	S	A	C	Stim	P	Q
2146	ELWI	O	sVt	1100	5	0
2151	ELWI	O	sVt	1164	5	0
2156	ELWI	O	sVt	1231	5	0
2161	ELWI	O	sVt	1301	5	0
2166	ELWI	O	sVt	1374	5	0
2171	ELWI	O	sVt	1450	5	0

zB Hörer ELWI (Altersgruppe Old) antwortete 5 Mal mit /U/
(und kein Mal mit /I/) zu dem Vokal-Stimulus mit F2 = 1100 Hz.

Vorgang

Response: ein forced-choice, binäres Urteil: /U/ oder /I/

Fixed factor Stim: die numerischen Werte des Kontinuums

Random Factor

(1 + Stim | Vpn): Vermittelt *m* und *k* in dieser Formel pro Vpn.

$$p = \frac{e^{(mx+k)}}{1+e^{(mx+k)}}$$

50% Umkipppunkt

= -k/m (2 Werte pro Sprecher, eins pro Kontinuum)

↓

t-Test

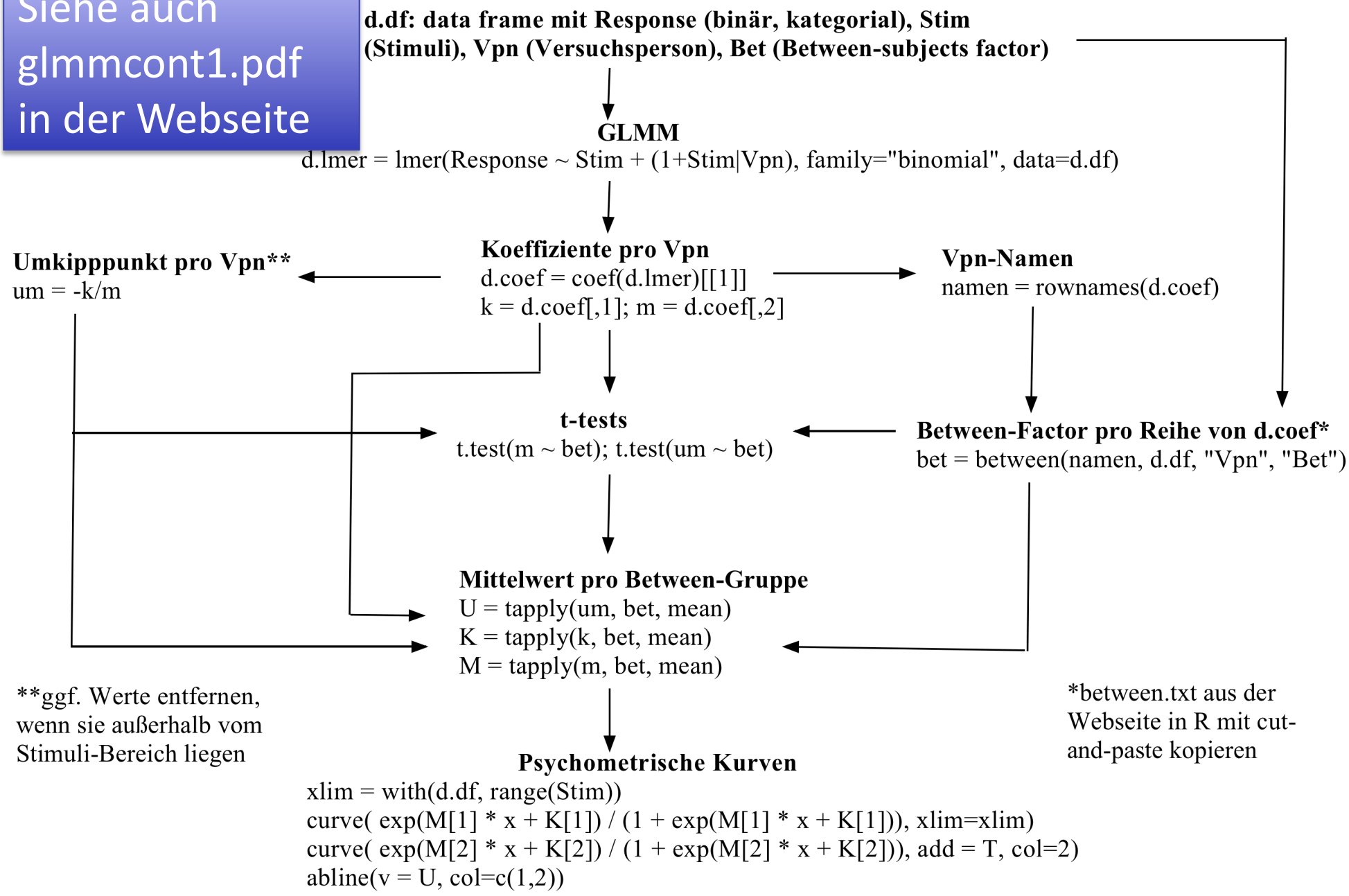
Haben jung vs. alt unterschiedliche Umkipppunkte oder unterschiedliche Neigungen?

↓

Psychometrische Kurve mit curve() erstellen, Um-Punkt überlagern (vorige Seite)

Ein Kontinuum (zB leiten-leiden), ein Between-Subjects Factor (zB Alter)

Siehe auch glmmcont1.pdf in der Webseite



Ein Between-Factor (Alter), Ein Within-Factor (Kontinuum)

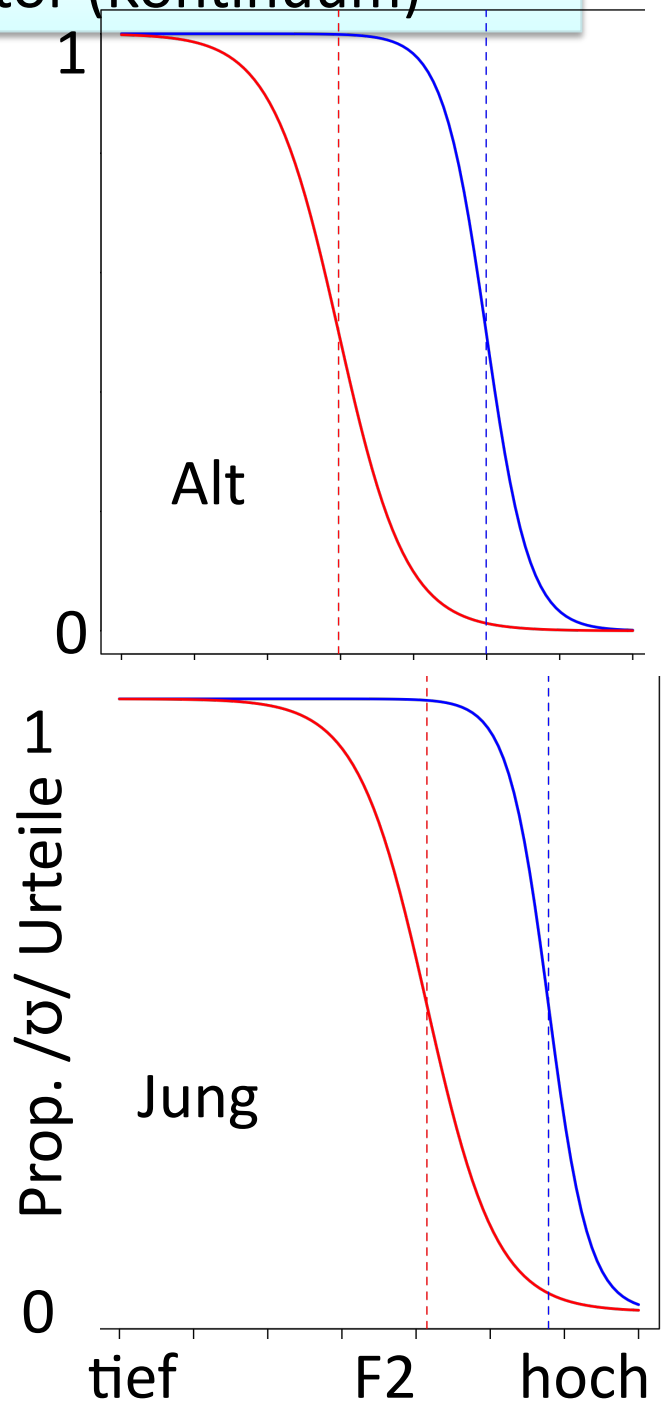
Zwei 13-stufige /ɪ-ʊ/ Kontinua wurden synthetisiert in einem alveolaren (/sɪt-sʊt/) und labialen Kontext (/wɪl-wʊl/).

Die Kontinua wurden erzeugt durch Herabsenken von F2 in 13 Schritten).

Es gab 2 Gruppen von Versuchspersonen: alt und jung.

Die Versuchspersonen mussten pro Stimulus entscheiden: war es /I/ oder /ʊ/?

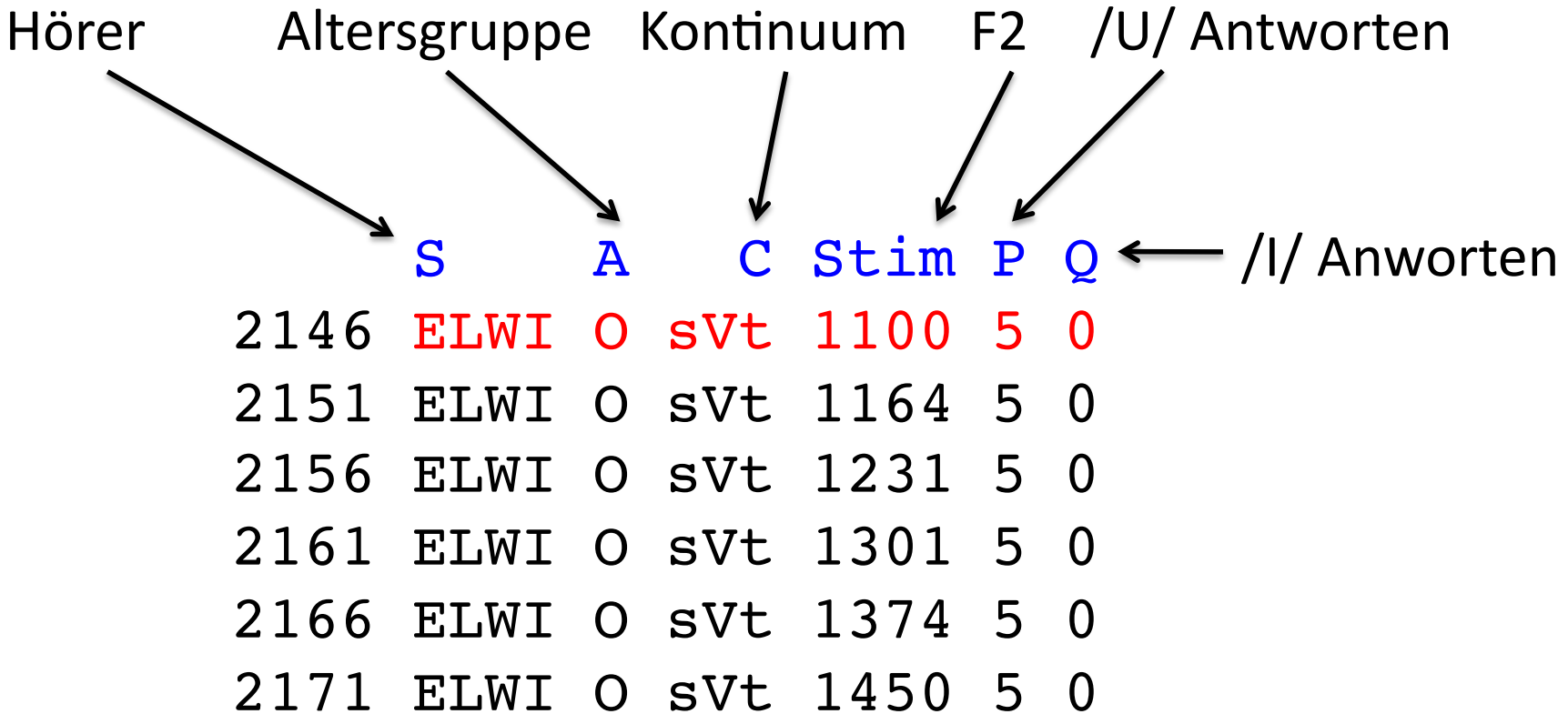
Die Fragen: wird der Umkipppunkt zwischen /I/ und /ʊ/ (a) vom Kontext (b) vom Alter beeinflusst?



GLMM, Psychometrische Kurven, Umkipppunkte

Data-Frame lax

Die relevanten Variablen



	S	A	C	Stim	P	Q
2146	ELWI	O	sVt	1100	5	0
2151	ELWI	O	sVt	1164	5	0
2156	ELWI	O	sVt	1231	5	0
2161	ELWI	O	sVt	1301	5	0
2166	ELWI	O	sVt	1374	5	0
2171	ELWI	O	sVt	1450	5	0

zB Hörer ELWI (Altersgruppe Old) antwortete 5 Mal mit /U/ (und kein Mal mit /I/) zu dem Vokal-Stimulus mit F2 = 1100 Hz in dem sVt (sit-soot) Kontinuum.

Vorgang

Genau wie vorher, aber:

(a) lmer() und die Berechnung der Koeffiziente und Umkipppunkte wird **getrennt pro Kontinuum durchgeführt**.

(b) Anstatt ein t-test benötigen wir ein RM-Anova oder MM da es jetzt mehrere Faktoren gibt: einen Between (Altersgruppe) und einen Within (Kontinuum).

Das letztere ist within, weil Kontinuum 2 Stufen hat (sVt, wVl), zu dem jeder Hörer eine Antwort gegeben hat.

2 Kontinua (zB leiten-leiden; baten-baden), ein Between-Subjects Factor (zB Alter)

data frame mit Response (binär, kategorial), Stim (Stimuli), Vpn (Versuchsperson), Bet (Between-subjects factor), Con (Die Continua mit 2 Stufen: a (zB leiten-leiden), b (zB baten-baden)

Continua trennen

```
temp = with(d.df, Con =="a")
a.df = d.df[temp,]; b.df = d.df[!temp,]
```

Ein GLMM pro Continuum

```
a.lmer = lmer(Response ~ Stim + (1+Stim|Vpn), family=binomial, data=a.df)
b.lmer = lmer(Response ~ Stim + (1+Stim|Vpn), family=binomial, data=b.df)
```

Vpn (Random factor)

```
a.namen = rownames(a.coef)
b.namen = rownames(b.coef)
V = factor(c(a.namen, b.namen))
```

Koeffiziente (k, m)

```
a.coef = as.matrix(coef(a.lmer)[[1]])
b.coef = as.matrix(coef(b.lmer)[[1]])
beide.coef = rbind(a.coef, b.coef)
colnames(beide.coef) = c("k", "m")
```

Umkipppunkt

```
um = -beide.coef[,1]/beide.coef[,2]
```

Between factor

```
a.bet = between(a.namen, a.df, "Vpn", "Bet")
b.bet = between(b.namen, b.df, "Vpn", "Bet")
B = factor(c(as.character(a.bet), as.character(b.bet)))
```

Continuum-Namen (Within factor)

```
Kon = factor(c(rep("a", nrow(a.coef)), rep("b", nrow(b.coef))))
```

Data-frame: Umkipppunkt, Koeffiziente, Vpn, Between-factor, Continuum-Namen

```
coef.df = data.frame(um, beide.coef, Vpn=V, Bet=B, Kontext=Kon, row.names=NULL)
```

Mittelwerte: Umkipppunkte und Koeffiziente für die Abbildungen

```
with(coef.df, apply(cbind(um, k, m), 2, tapply, paste(Kontext, Bet, sep="."), mean))
```

Prüfstatistik: MM oder RM-Anova zB

```
mm = lmer(um ~ Bet * Kontext + (1|Vpn), data=coef.df)
```

oder

```
um.t = Anova.prepare(coef.df[,-c(2,3)], c("d", "s", "b", "w"))  usw
```

Siehe auch glmmcont2.pdf in der Webseite