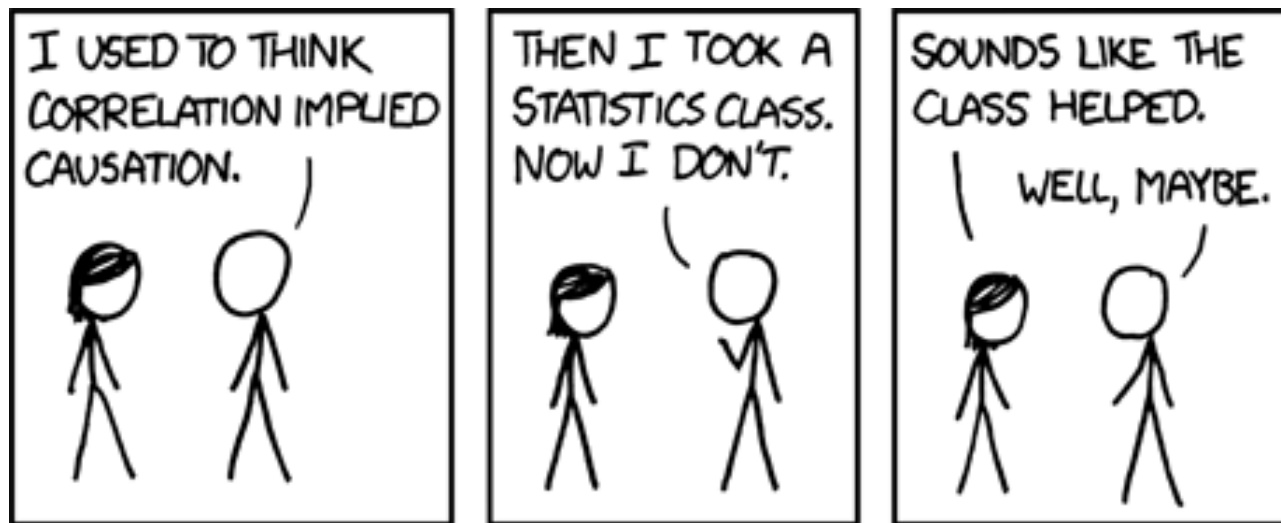


# Binary Logistic Regression & Linear Mixed Effects Models

R Workshop am ZAS, 24.5.2011



[XKCD – „Correlation“]

Amir Zeldes

[amir.zeldes@rz.hu-berlin.de](mailto:amir.zeldes@rz.hu-berlin.de)

# Themen

- Vorhersagekräftige Modelle mit binärer logistischer Regression
- Umgang mit zufälliger Varianz anhand von Mixed Effects Models

Daten herunterladen:

[http://zope.linguistik.hu-berlin.de/institut/professuren/korpuslinguistik/mitarbeiterinnen/amir/ZAS\\_data.zip](http://zope.linguistik.hu-berlin.de/institut/professuren/korpuslinguistik/mitarbeiterinnen/amir/ZAS_data.zip)

Folien:

[https://korpling.german.hu-berlin.de/~amir/ZAS\\_Stat\\_2011.pdf](https://korpling.german.hu-berlin.de/~amir/ZAS_Stat_2011.pdf)

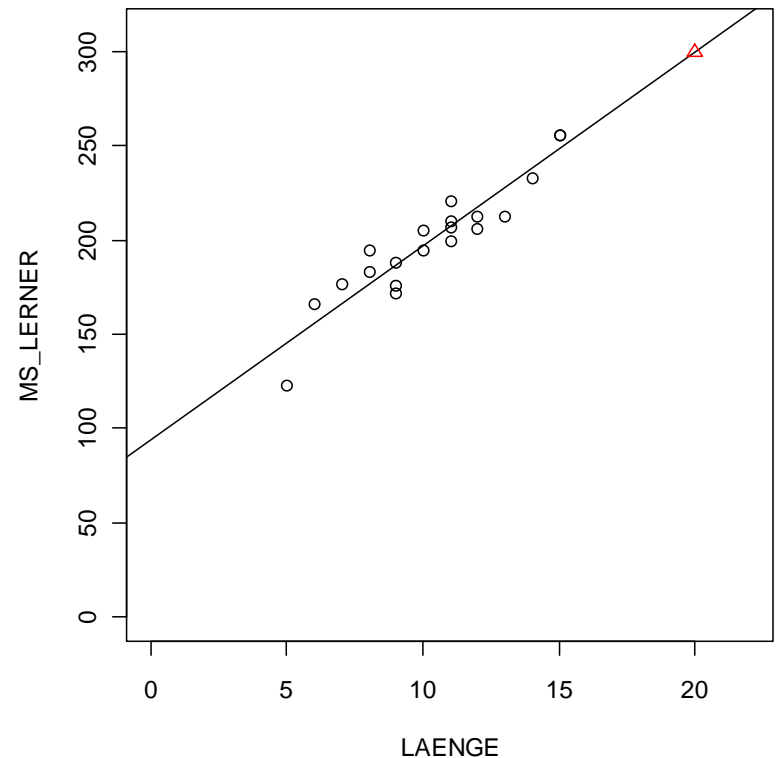
# Voraussetzungen

- Varianz & SD
- Signifikanz
- Statistische Tests (z, t, ANOVA)
- Korrelation ( $r^2$ )
  
- Umgang mit Vektoren, Dateien etc. in R
- lme4, Design, languageR, (amap)

# Binary Logistic Regression

# Was ist BLR?

- Korrelationen bilden eine Grundlage der statistischen Arbeit
- Lineare Modelle ermöglichen Vorhersagen, sind oft sehr hilfreich
- Aber was macht man, wenn man mehr als zwei Faktoren hat?



# BLR – die Idee

- Nehmen wir an, wir möchten eine binäre (ja-nein) Entscheidung erklären:
  - Scrambling im MF: Dat-Akk oder Akk-Dat?
  - Haging Topik: ja oder nein?
  - Genitiv mit -es oder -s?
  - ...
- Die Entscheidung kann mit vielen Faktoren zusammenhängen
  - Wie viele Silben?
  - Ist die Phrase pronominal?
  - ...

# BLR – die Idee

- Jeder Faktor kann etwas beitragen:
  - Dat. ist Pronominal  $\rightarrow$   $p(\text{dat vor akk})$  steigt
  - Akk. ist kürzer  $\rightarrow$   $p(\text{akk vor dat})$  steigt ...
- Es kann aber auch zu Interaktionen kommen:
  - Länge in Silben ist nur relevant, wenn der Referent bekannt (vorerwähnt) ist
  - Einfluss von Länge ist noch stärker, wenn das Wort ein Kompositum ist
  - ...

# BLR – die Idee

- Wir möchten für beliebig viele nominal- bzw. verhältnisskalierte Faktoren sagen,
  - ob sie zur Entscheidung etwas beitragen (Signifikanz)
  - wie stark der Beitrag ist (Effektstärke, Erklärungskraft)
  - dasselbe für Interaktionen von mehreren Faktoren
- Achtung: unsere Beobachtungen sollen unabhängig sein!

# Die Daten

- Als Beispiel untersuchen wir die englische **Ditransitive Alternation**:  
(Daten aus Gries 2008; vgl. ausführlichere Studien von Bresnan et al.)
  - give the book to John (prepositional)
  - give John the book (ditransitive)
- Die abhängige Variable: Wahl der Konstruktion
- Die unabhängigen Variablen:
  - Art des Verbs: change of possession (Besitzwechsel, bspw. *give*) oder nicht (*show*)
  - Aktivierungsgrad von: Agens, Thema, Recipient

# Laden der Daten - *DatAlt.tab*

```
> library(Design)
> DatAlt <- read.table(file.choose(),header=T)
> str(DatAlt)
'data.frame':  304 obs. of  7 variables:
 $ FALL      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ KONSTRUKTION1: int  1 0 1 0 0 1 1 0 0 0 ...
 $ KONSTRUKTION2: Factor w/ 2 levels "ditransitiv",...: 2 1
 2 1 1 2 2 1 1 1 ...
 $ V_CHANGPOSS : Factor w/ 2 levels "ja","nein": 2 2 2 2
 1 2 2 2 1 2 ...
 $ AGENT_ACT   : int  0 0 9 0 4 0 9 0 9 8 ...
 $ REC_ACT     : int  0 9 0 9 9 0 0 9 7 8 ...
 $ PAT_ACT     : int  0 0 0 0 0 0 0 0 0 0 ...
```

# Wie sehen die Daten aus?

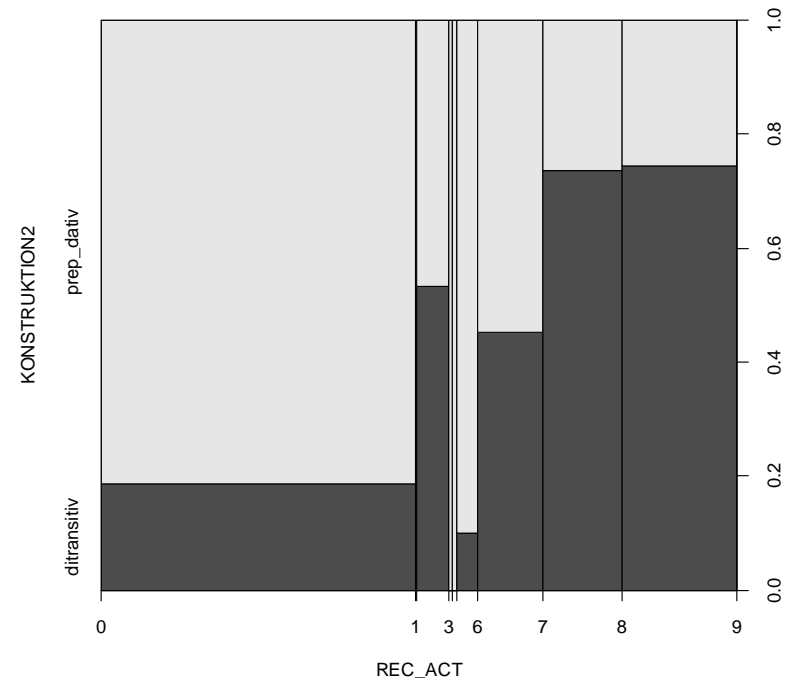
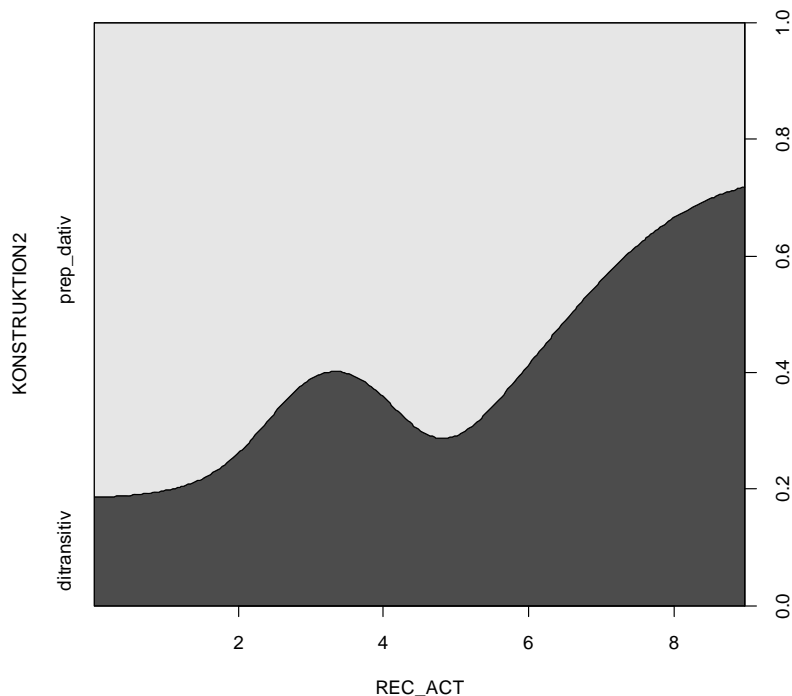
```
> attach(DatAlt)
```

```
> table(KONSTRUKTION2,  
V_CHANGPOS)
```

	ja	nein
ditransitiv	23	97
prep_dativ	12	168

# Wie sehen die Daten aus?

- > `cdplot(KONSTRUKTION2 ~ REC_ACT)`
- > `spineplot(KONSTRUKTION2~REC_ACT)`



# Unser 1. Modell

```
> DatAlt.lrm <- lrm(KONSTRUKTION1 ~ V_CHANGPOSS  
+ AGENT_ACT + REC_ACT + PAT_ACT +  
V_CHANGPOSS:AGENT_ACT +  
V_CHANGPOSS:REC_ACT +  
V_CHANGPOSS:PAT_ACT)  
> DatAlt.lrm
```

Logistic Regression Model

```
lrm(formula = KONSTRUKTION1 ~ V_CHANGPOSS +  
AGENT_ACT + REC_ACT +  
PAT_ACT + V_CHANGPOSS:AGENT_ACT +  
V_CHANGPOSS:REC_ACT +  
V_CHANGPOSS:PAT_ACT)
```

# Qualität des Modells

Frequencies of Responses

0	1
120	180

Frequencies of Missing Values Due to Each Variable

KONSTRUKTION1	V_CHANGPOSS	AGENT_ACT	REC_ACT	PAT_ACT
0	4	0	0	0

Obs	Max Deriv	Model L.R.	d.f.	P	C	Dxy	Gamma
300	3e-09	89.1	7	0	0.805	0.61	0.621
Tau-a	R2	Brier					
0.294	0.347	0.177					

# Koeffizienten

	Coef	S.E.	Wald Z	P
Intercept	-0.07675	0.8849	-0.09	0.9309
V_CHANGPOSS=nein	1.31410	0.9297	1.41	0.1575
AGENT_ACT	0.04013	0.1364	0.29	0.7685
REC_ACT	-0.32950	0.1369	-2.41	0.0161
PAT_ACT	0.24027	0.1431	1.68	0.0931
V_CHANGPOSS=nein* AGENT_ACT	0.06986	0.1414	0.49	0.6212
V_CHANGPOSS=nein* REC_ACT	0.05597	0.1424	0.39	0.6943
V_CHANGPOSS=nein* PAT_ACT	-0.31178	0.1525	-2.04	0.0409

# Was ist hier mathematisch los?

- Die logistische Funktion nimmt als Parameter  $z$  Werte zwischen:  $-\infty < z < \infty$
- liefert immer eine Zahl:  $0 \leq f(z) \leq 1$
- Perfekt, um Wahrscheinlichkeiten zu modellieren
- $z$  besteht aus beliebig vielen gewichteten Faktoren

$$f(z) = \frac{1}{1 + e^{-z}}$$

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k$$

# Was ist hier mathematisch los?

- R muss nur noch die optimalen Werte für die Konstanten  $\beta_i$  berechnen, in Anbetracht der Daten
- R kann uns sagen, wie genau diese  $\beta_i$  unsere Daten vorhersagen (post-hoc)
- Und wir können das Modell mit ungesehenen Daten testen (pre-hoc)

$$f(z) = \frac{1}{1 + e^{-z}}$$

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k$$

# Verbessertes Modell berechnen

- Wir verzichten auf die nicht-signifikanten Faktoren

```
> (DatAlt2.lrm <- lrm(KONSTRUKTION1 ~ V_CHANGPOSS +  
  REC_ACT + PAT_ACT + V_CHANGPOSS:PAT_ACT))
```

	Coef	S.E.	Wald Z	P
Intercept	0.03251	0.47498	0.07	0.9454
<b>V_CHANGPOSS=nein</b>	<b>1.72872</b>	<b>0.50178</b>	<b>3.45</b>	<b>0.0006</b>
REC_ACT	-0.27359	0.03636	-7.52	0.0000
<b>PAT_ACT</b>	<b>0.21749</b>	<b>0.10904</b>	<b>1.99</b>	<b>0.0461</b>
V_CHANGPOSS=nein *				
PAT_ACT	-0.27387	0.12173	-2.25	0.0245

# Effektstärke

- Wie viel größer wird die Chance (odds-ratio), die Dativkonstruktion zu wählen, nach jedem Faktor?

```
> (effect.sizes <-  
  exp(DatAlt2.lrm$coefficients))  
Intercept                V_CHANGPOSS=nein  
1.0330443                5.6334413  
REC_ACT                  PAT_ACT  
0.7606418                1.2429479  
V_CHANGPOSS=nein * PAT_ACT  
0.7604342
```

# Vorhersagen treffen

- Wenn wir die Werte der unabhängigen Variablen kennen, sind Vorhersagen möglich

```
> prediction <-  
  predict.lrm(DatAlt2.lrm)
```

```
> head(prediction)
```

	1	2	3
	1.7612305	-0.7011037	1.7612305
	4	5	6
	-0.7011037	-2.4298242	1.7612305

# oder mit ungesehenen Werten

```
> predict.lrm(DatAlt2.lrm,  
  newdata=list(V_CHANGPOSS=2,  
  REC_ACT=1, PAT_ACT=2) )
```

1

1.374878

# Binäre Darstellung

```
> prediction.bin <-  
  ifelse(prediction<=0,0,1)  
> (xtab<-  
  table(prediction.bin,KONSTRUKTION1))
```

```
                KONSTRUKTION1  
Prediction.bin  0    1  
                0    85  46  
                1    35  134
```

```
> sum(diag(xtab))/sum(xtab)  
[1] 0.73
```

# Zusammenfassung

- Einige Faktoren beeinflussen die Wahl:
  - Besitzwechsel=nein ist am wichtigsten (odds~5.63,  $p<0.001$ ) für die Präposition
  - Aktivierung des Rezipienten erhöht die Chancen auf Ditransitiv (odds~0.76,  $p<0.001$ )
  - Patiens ist gemischt:
    - Bei Besitzwechsel=nein: ditransitiv (odds~0.76,  $p<0.05$ )
    - Sonst: Präposition (odds~1.24,  $p<0.05$ )

# Und weiter? (Bresnan et al. 2007 i.a.)

	Coefficient	Odds Ratio PP	95% C.I.
inanimacy of recipient	2.54	12.67	5.56–28.87
nonpronominality of recipient	1.17	3.22	1.70–6.09
nongivenness of recipient	0.99	2.69	1.37–5.3
transfer semantic class	0.96	2.61	1.44–4.69
indefiniteness of recipient	0.85	2.35	1.25–4.43
plural number of theme	0.50	1.65	1.05–2.59
person of recipient	0.48	1.62	1.06–2.46
nongivenness of theme	–1.05	0.35	0.19–0.63
structural parallelism in dialogue	–1.13	0.32	0.22–0.47
nonpronominality of theme	–1.18	0.31	0.19–0.50
length difference (log scale)	–1.21	0.3	0.22–0.4
communication semantic class	–1.34	0.26	0.13–0.55
indefiniteness of theme	–1.37	0.25	0.15–0.44

# Zusammenfassung

- BLR kann beliebige ja-nein-Entscheidungen modellieren
- Anzahl der Faktoren / Interaktionen unbegrenzt...
- aber es gilt, das optimale Modell zu finden
  - Minimum an Faktoren
  - Nur signifikante Beiträge

# Mixed Effects Models

# Die ideale statistische Welt

- Wir möchten immer den Einfluss von bestimmten Variablen auf unsere Ergebnisse messen
- Alles andere zwischen unseren Messungen soll identisch bleiben
- Beispiel:
  - Wir messen Reaktionszeiten bei einer Lexical Decision Task für jeweils 10 Personen: Deutschlerner und Muttersprachler
  - Die Probanden und deren Antworten sollen in jeder Hinsicht nur von der Gruppe abhängen: L1 oder L2

# Statistik in der Wirklichkeit

- „Zufällige“ Varianz (nicht systematisch):
  - Bestimmte Probanden in jeder Gruppe sind schneller oder langsamer
  - Jeder Proband kann selbst mal schneller, mal langsamer reagieren
  - Probanden werden müde > langsamer
  - Probanden werden besser > schneller
  - ...
- Solche aus unserer Sicht **randomalen** Effekte können nicht kontrolliert werden
- Und auch nicht reproduziert werden!

# Wie wird man diese Einflüsse los?

- Die Variablen, die wir kontrollieren können, heißen „fixed effects“
- Die unkontrollierbaren Variablen darüber hinaus heißen „random effects“
- Ein Modell, das beide Arten von Effekten berücksichtigt heißt **Mixed Effects Model**
- Es gilt, die zufälligen Einflüsse zu identifizieren und möglichst vorhersagekräftige Modelle zu entwickeln

# Vorannahmen

- Wir gehen davon aus, dass die zufälligen Effekte **normal verteilt** sind (sonst wären sie ja systematisch)
- der beobachtete Mittelwert ist mit beiden Effektgruppen (random + fixed) **linear korreliert**
- **Varianz** in jeder Gruppe hängt nicht vom Mittelwert ab

# Die lexdec-Daten (Baayen 2008, angepasst)

- Wir vergleichen die benötigte Zeit für die Lexical Decision Task und erheben:
  - 12 Muttersprachler + 9 Lerner a 79 Versuche
  - 44 Tiernamen / 33 Obst & Gemüse
  - Länge zwischen 3-10 Buchstaben
  - Korpusfrequenzen für jedes Wort
  - Die Reihenfolge der Versuche
- Fixed Effects:
  - L1, Wortklasse, Versuch, Länge, Frequenz
- Random Effects:
  - Person, Wort (da unsere Schlussfolgerungen auch für ungesehene Wörter und Personen gelten sollen)

# Die Daten

```
> lexdec <- read.table(file.choose(),header=T)
> str(lexdec)
'data.frame': 1659 obs. of 11 variables:
 $ ID      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Subject : Factor w/ 21 levels "A1","A2","A3",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ RT      : int  567 549 572 486 414 483 418 525 639 485 ...
 $ Trial    : int  23 27 29 30 32 33 34 38 41 42 ...
 $ Sex     : Factor w/ 2 levels "F","M": 1 1 1 1 1 1 1 1 1 1 ...
 $ Lang    : Factor w/ 2 levels "L1","L2": 1 1 1 1 1 1 1 1 1 1 ...
 $ Correct : Factor w/ 2 levels "correct","incorrect": 1 1 1 1 1 1 1 1 1 1
 ...
 $ Word    : Factor w/ 79 levels "almond","ant",...: 55 47 20 58 25 12 71
 69 62 1 ...
 $ Frequency: int  129 100 148 113 2138 58 116 111 21 67 ...
 $ Length  : int  3 4 6 4 3 10 10 8 6 6 ...
 $ Class   : Factor w/ 2 levels "animal","plant": 1 1 2 2 1 2 2 1 2 2 ...
```

# Die Daten

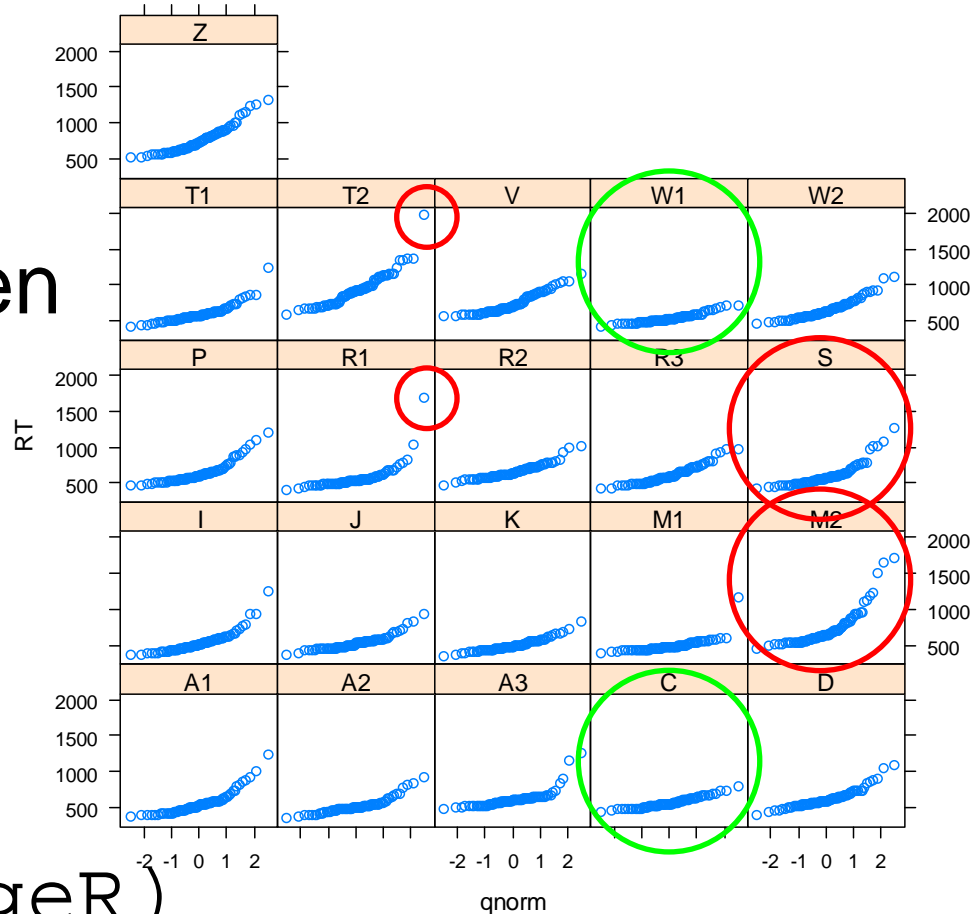
```
> head(lexdec)
```

ID	Subject	RT	Trial	Sex	Lang	Correct	Word	Frequency	Length
1	1	A1 567	23	F	L1	correct	owl	129	3 animal
2	2	A1 549	27	F	L1	correct	mole	100	4 animal
3	3	A1 572	29	F	L1	correct	cherry	148	6 plant
4	4	A1 486	30	F	L1	correct	pear	113	4 plant
5	5	A1 414	32	F	L1	correct	dog	2138	3 animal
6	6	A1 483	33	F	L1	correct	blackberry	58	10 plant

# Erste Ansicht der Daten

Wie sehen die Daten  
der Versuchspersonen  
aus?

Sind sie normal-  
verteilt?



```
>library(languageR)
```

```
>qqmath(~RT | Subject, data=lexdec)
```

# Daten bereinigen

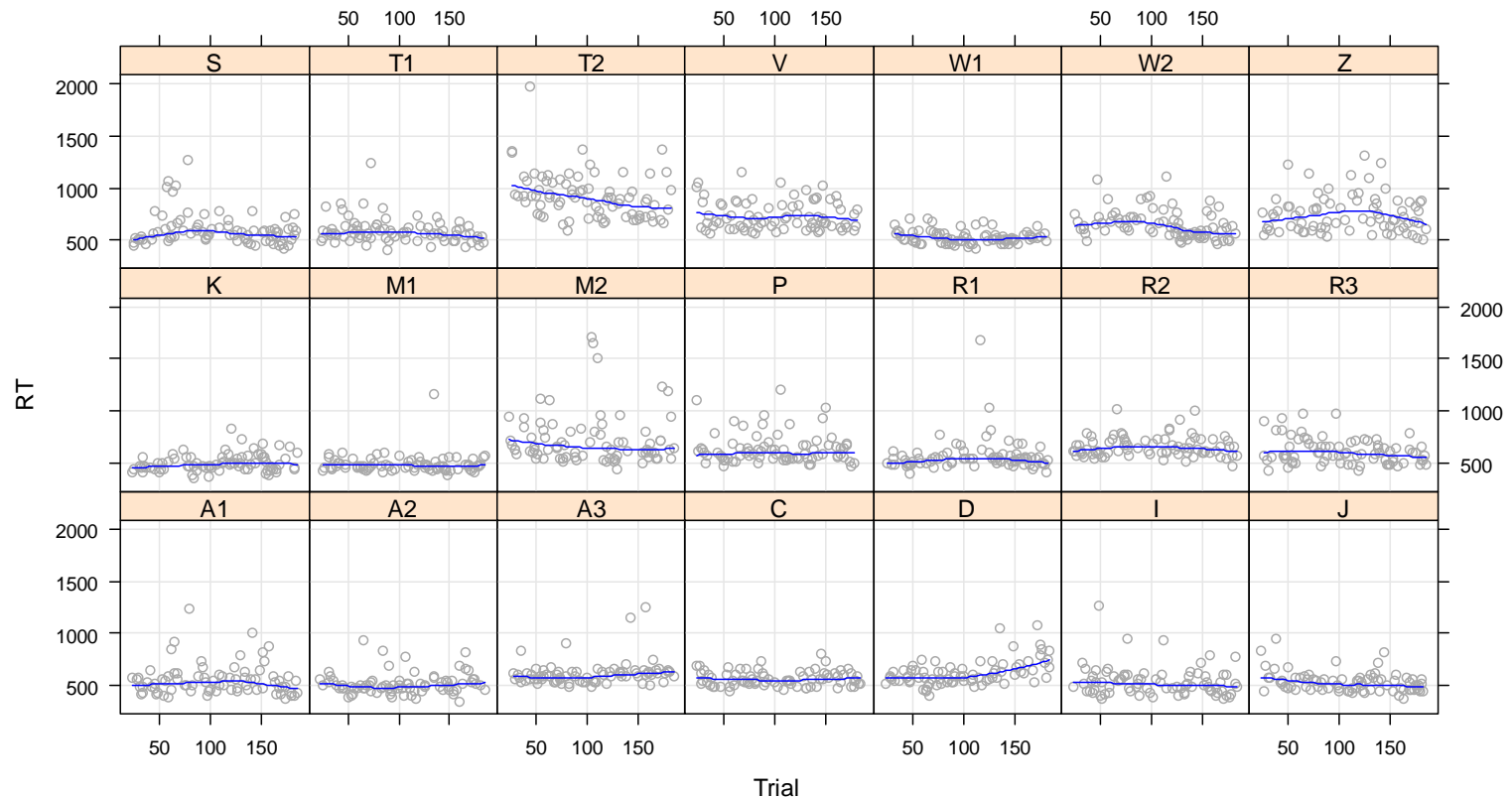
- Ausreißerwerte führen zu einem verzerrten Modell (müssen mit erklärt werden)
- Man darf Werte trotzdem nicht grundlos entfernen
- Bei Lexical Decision ignoriert man oft nicht normal verteilte, sehr hohe Werte
- Verdächtige Werte sollten genauer untersucht werden

# Einfaches Beispiel

- Wir wollen nur korrekte Antworten berücksichtigen
  - Keine Ausreißerwerte über 1100ms (Proband hat nicht spontan geantwortet)
- ```
> lexdec2 <- lexdec[lexdec$RT < 1100 & lexdec$Correct == "correct", ]
```

# 1. Effekt – Trial (Müdigkeit / Übung)

```
> xyloless.fnc(RT ~ Trial | Subject,  
data = lexdec2, ylab = "RT")
```



# Woran liegt die Varianz?

- Es kann sein, dass spätere Trials immer leichter/schwieriger sind (**fixed effect**)
- Es kann sein, dass manche Wörter schwieriger sind (**random effect**)
- Es kann sein, dass Probanden sich unterschiedlich verhalten (**random effect**)
- ...
- Und es kann sein, dass RT einfach zufällig schwankt („**residuelle**“ Zufälligkeit)

# Unser 1. LME-Modell

```
> library(lme4)
> (lexdec2.lmer <- lmer(RT~
Trial + (1|Subject) + (1|Word),
lexdec2))
```

Linear mixed model fit by REML

Formula: RT ~ Trial + (1 | Subject) +  
(1 | Word)

Data: lexdec2

| AIC   | BIC   | logLik | deviance | REMLdev |
|-------|-------|--------|----------|---------|
| 18914 | 18941 | -9452  | 18908    | 18904   |

**Unterschiedliche Qualitätsmaße  
(nützlich zum Vergleich mit anderen *lmer*)**

# Unser 1. LME-Modell

## Random effects:

| Groups          | Name        | Variance | Std.Dev. | Mittelwerte = |
|-----------------|-------------|----------|----------|---------------|
| <b>Word</b>     | (Intercept) | 1884.4   | 43.409   | <b>0</b>      |
| <b>Subject</b>  | (Intercept) | 7387.8   | 85.952   |               |
| <b>Residual</b> |             | 9557.5   | 97.763   |               |

Number of obs: 1558, groups: Word, 79; Subject, 21

## Fixed effects:

|                    | Estimate         | Std. Error | t value |
|--------------------|------------------|------------|---------|
| <b>(Intercept)</b> | <b>613.85289</b> | 20.32518   | 30.202  |
| <b>Trial</b>       | -0.12517         | 0.05332    | -2.348  |

„Ausgangszeit“

Correlation of Fixed Effects:  
(Intr)  
Trial -0.276

In einer großen  
Stichprobe (n>100)  
~ signifikant

# Was sagt uns das Modell?

Wie viel schneller welche Wörter sind (im Modell):

```
ranef ( lexdec2.lmer ) $Word
```

```
( Intercept )
```

|           |             |
|-----------|-------------|
| almond    | 1.7769858   |
| ant       | -27.4963614 |
| apple     | -64.3843336 |
| apricot   | -8.0652116  |
| asparagus | 61.4288199  |
| avocado   | 13.1246418  |
| ...       |             |

# Was sagt uns das Modell?

Gleichermaßen die Versuchspersonen:

```
ranef ( lexdec2.lmer ) $ Subject
```

```
A1    -58.523793
```

```
A2    -82.496253
```

```
R3     10.839131
```

```
S      -4.967746
```

```
T1    -18.721382
```

```
T2    239.602528...
```

# Was sagt uns das Modell?

- Das sind aber Werte für einzelne bekannte Personen und Wörter
- Viel wichtiger ist die Einschätzung der Abweichung durch die Random Effects Word und Subject
- und die Signifikanz und Stärke des Effekts von Trial



# Aber... Subjects waren doch unterschiedlich

- Trial hat einen Signifikanten Einfluss
- Warum werden manche Personen schneller und manche langsamer?
- **Interaktion von Person~Trial**

## 2. Modell

```
> (lexdec2.lmer2 <- lmer(RT~  
Trial+(1+Trial|Subject)+(1|Word)  
,lexdec2))
```

# 2. Modell

## Random effects:

| Groups          | Name               | Variance   | Std.Dev.  | Corr   |
|-----------------|--------------------|------------|-----------|--------|
| <b>Word</b>     | <b>(Intercept)</b> | 1.9218e+03 | 43.83802  |        |
| <b>Subject</b>  | <b>(Intercept)</b> | 1.3041e+04 | 114.19771 |        |
|                 | <b>Trial</b>       | 2.0376e-01 | 0.45139   | -0.723 |
| <b>Residual</b> |                    | 9.1295e+03 | 95.54825  |        |

Number of obs: 1558, groups: Word, 79; Subject, 21

## Fixed effects:

|                    | Estimate | Std. Error | t value |
|--------------------|----------|------------|---------|
| <b>(Intercept)</b> | 615.8006 | 26.1092    | 23.59   |
| <b>Trial</b>       | -0.1417  | 0.1116     | -1.27   |

Correlation of Fixed Effects:

(Intr)  
Trial -0.708

Nicht mehr  
signifikant

# Korrelationen in Ranef

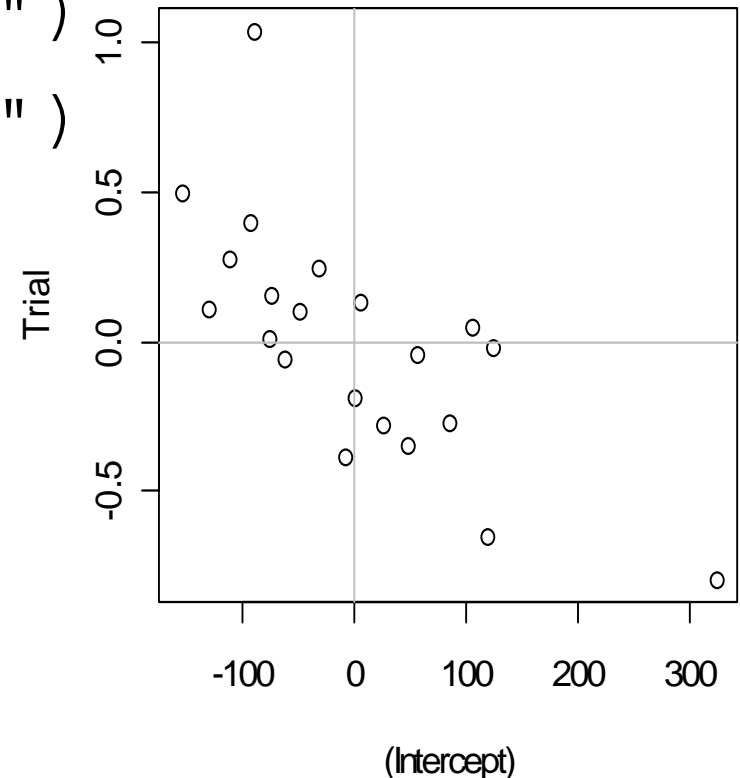
Wie wirkt Trial für jede Versuchsperson?

```
> plot(ranef(lexdec2.lmer2)$Subject)
```

```
> abline(h=0, col="grey")
```

```
> abline(v=0, col="grey")
```

- Wer langsam ist, wird schneller
- Wer schnell ist, wird langsamer



# Was nun?

- Wir wissen, dass Trial signifikant ist...
- ...aber für jede Person anders
- Das konnten wir in diesen Daten nicht kontrollieren: wir möchten diesen Effekt ignorieren
- Uns interessiert eigentlich der Unterschied **L1 : L2**

# 3. Modell

```
>(lexdec2.lmer3 <- lmer(RT ~  
  Trial+Lang+(1+Trial|Subject)+(1|Word), lexdec2))
```

Random effects:

| Groups          | Name               | Variance   | Std.Dev.  | Corr   |
|-----------------|--------------------|------------|-----------|--------|
| <b>Word</b>     | <b>(Intercept)</b> | 1.9253e+03 | 43.87858  |        |
| <b>Subject</b>  | <b>(Intercept)</b> | 1.1680e+04 | 108.07567 |        |
|                 | <b>Trial</b>       | 2.0397e-01 | 0.45163   | -0.794 |
| <b>Residual</b> |                    | 9.1286e+03 | 95.54384  |        |

Number of obs: 1558, groups: Word, 79; Subject, 21

Fixed effects:

|                    | Estimate | Std. Error | t value |
|--------------------|----------|------------|---------|
| <b>(Intercept)</b> | 576.9678 | 28.0744    | 20.551  |
| <b>Trial</b>       | -0.1411  | 0.1116     | -1.264  |
| <b>LangL2</b>      | 90.5206  | 30.4596    | 2.972   |

signifikant

# Interaktionen in den Fixed Effects?

- Wir berechnen nun ein Modell mit möglicher Interaktion von Wortlänge und Muttersprache (zwei **Fixed Effects**)

```
> (lexdec2.lmer4 <-  
  lmer(RT~Trial+Lang*Length+  
  (1+Trial|Subject)+(1|Word), lexdec2))
```

Fixed effects:

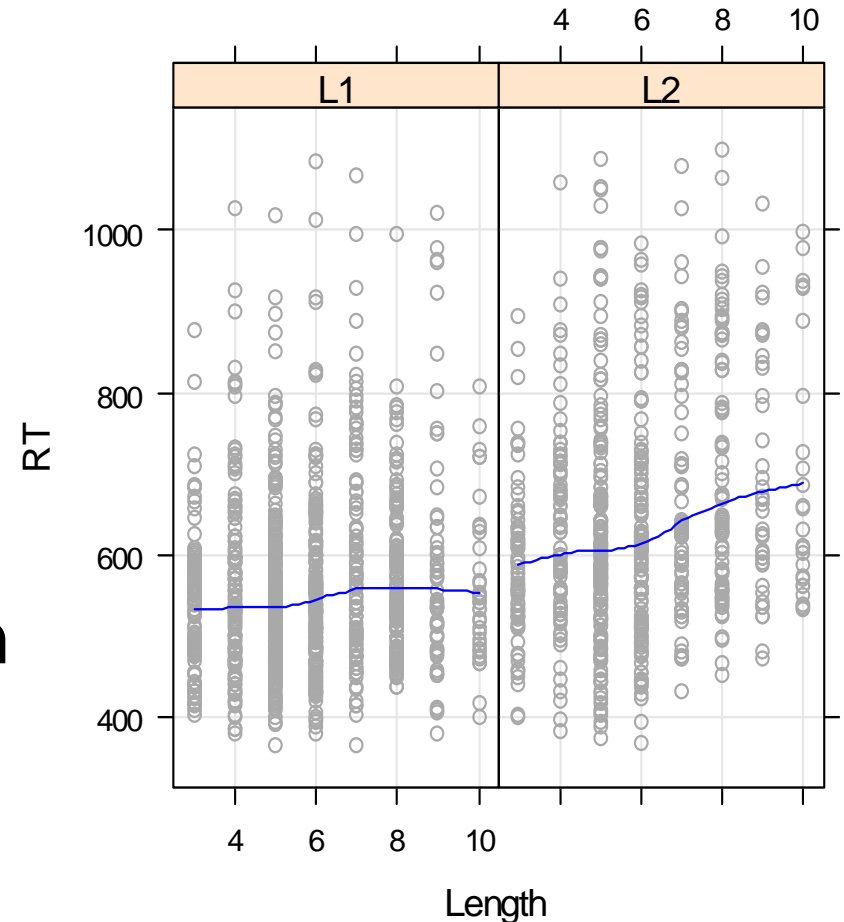
|                      | Estimate | Std. Error | t            | value |
|----------------------|----------|------------|--------------|-------|
| (Intercept)          | 538.0778 | 32.8641    | 16.373       |       |
| Trial                | -0.1494  | 0.1129     | -1.323       |       |
| LangL2               | 11.1011  | 34.2706    | 0.324        |       |
| Length               | 6.7915   | 2.8608     | 2.374        |       |
| <b>LangL2:Length</b> | 13.4186  | 2.6491     | <b>5.065</b> |       |

# Interaktionen in den Fixed Effects?

- Offenbar beeinflusst **Lang** die RT weniger, als eine Interaktion **Lang\*Length**:

```
> xylowess.fnc(  
  RT ~ Length |  
  Lang,  
  data = lexdec2)
```

- Nicht Muttersprachler sind bei kurzen Wörtern fast genauso schnell
- Bei den Muttersprachlern ist die Länge unwichtig



# Brauchen wir das alles?

```
>(lexdec2.lmer5 <-lmer(RT~  
  Trial+Lang*Length+  
  (1+Trial|Subject)+(1|Word) ,  
  lexdec2))
```

Fixed effects:

|               | Estimate | Std. Error | t value |
|---------------|----------|------------|---------|
| (Intercept)   | 537.6361 | 29.2931    | 18.354  |
| Trial         | -0.1446  | 0.1109     | -1.305  |
| LangL2        | 12.3659  | 35.1014    | 0.352   |
| Length        | 6.6621   | 1.8177     | 3.665   |
| LangL2:Length | 12.7339  | 2.8342     | 4.493   |

# Modelle vergleichen

- Die Modelle erklären einen Teil der Varianz
- Vergleich durch ANOVA möglich:

```
> anova(lexdec2.lmer4,lexdec2.lmer5)
```

```
Data: lexdec2
```

```
Models:
```

```
lexdec2.lmer5: RT ~ Trial + Lang * Length +  
(1 + Trial | Subject)
```

```
lexdec2.lmer4: RT ~ Trial + Lang + Lang * Length +  
(1 + Trial | Subject) + (1 |
```

```
lexdec2.lmer4:      Word)
```

|               | Df | AIC   | BIC   | logLik  | Chisq  | Chi | Df | Pr(>Chisq)    |
|---------------|----|-------|-------|---------|--------|-----|----|---------------|
| lexdec2.lmer5 | 9  | 18937 | 18985 | -9459.5 |        |     |    |               |
| lexdec2.lmer4 | 10 | 18838 | 18891 | -9408.7 | 101.53 |     | 1  | < 2.2e-16 *** |

```
---
```

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

# Was man mit dem besten Modell noch machen kann

- Erwartungswerte ausgeben lassen
- Simulationen ausführen
- Einschätzung der Erklärungskraft des Modells
- Erwartete Abweichung durch jeden Effekt

# fitted(modell.lmer)

- Die zu erwartende RT von jeder Person im nächsten Versuch:

```
> fitted(lexdec2.lmer4)
```

```
[1] 542.3780 564.8128 520.0987  
530.2310 508.1272 536.1893 ...
```

```
>
```

```
fitted(lexdec2.lmer4)[lexdec2$RT[lex  
dec2$Subject=="M2"]]
```

```
[1] 595.6023 563.5506 493.3604  
493.3604 593.4307 521.0490 ...
```

# Simulation anhand des Modells

```
> simulate(lexdec2.lmer4, nsim = 2)
  sim_1      sim_2
1 727.7082 514.9853
2 630.2731 493.5168
3 477.5391 296.6026
4 734.2604 625.7312
5 428.7836 636.8843
6 722.5493 664.8301
```

# Erklärungskraft

- Wir üblicherweise mit  $R^2$  geschätzt (die Varianz, die durch die Korrelation zwischen Modell und Beobachtungen erklärt wird)

```
> cor(fitted(lexdec2.lmer4), lexdec2$RT)^2  
[1] 0.5201528
```

# Zusammenfassung

- LMER eignen sich für verrauschte Daten
- Eine gute Methode, zufällige Störfaktoren und interessante Effekte auseinanderzuhalten
- Hauptentscheidung: welche Effekte sind „zufällig“?
- Danach: Modellfindung / Optimierung (ANOVA)

# Clustering

# Strukturen in unstrukturierten Daten

- Daten sind nie wirklich unstrukturiert:
  - Häufigkeitsverteilung der Werte
  - Reihenfolge
  - Dispersion
  - Deskriptive Statistiken
- Warum erkennen wir Strukturen nicht?
  - Zu viele Daten können nicht gleichzeitig wahrgenommen werden
  - Zusammenhang von sehr vielen **Variablen**
  - Die Kategorien, die uns interessieren, sind nicht explizit vorhanden, sondern von anderen Merkmalen indirekt ablesbar

# Explorative Verfahren

- Mit vertrauten Variablen und Verfahren kann man statistisch:
  - Daten beschreiben (Mittelwerte, Standardabweichung)
  - und Hypothesen überprüfen ( $\chi^2$ -Test, ANOVA)
- Was macht man, wenn man nicht weiß, **welche Variablen** die Daten aufteilen?
- Man möchte herausfinden, wie die Daten strukturiert sind
- Man möchte objektive Kategorien aufgrund der Daten **explorativ** entdecken

# Keine Hypothesen?

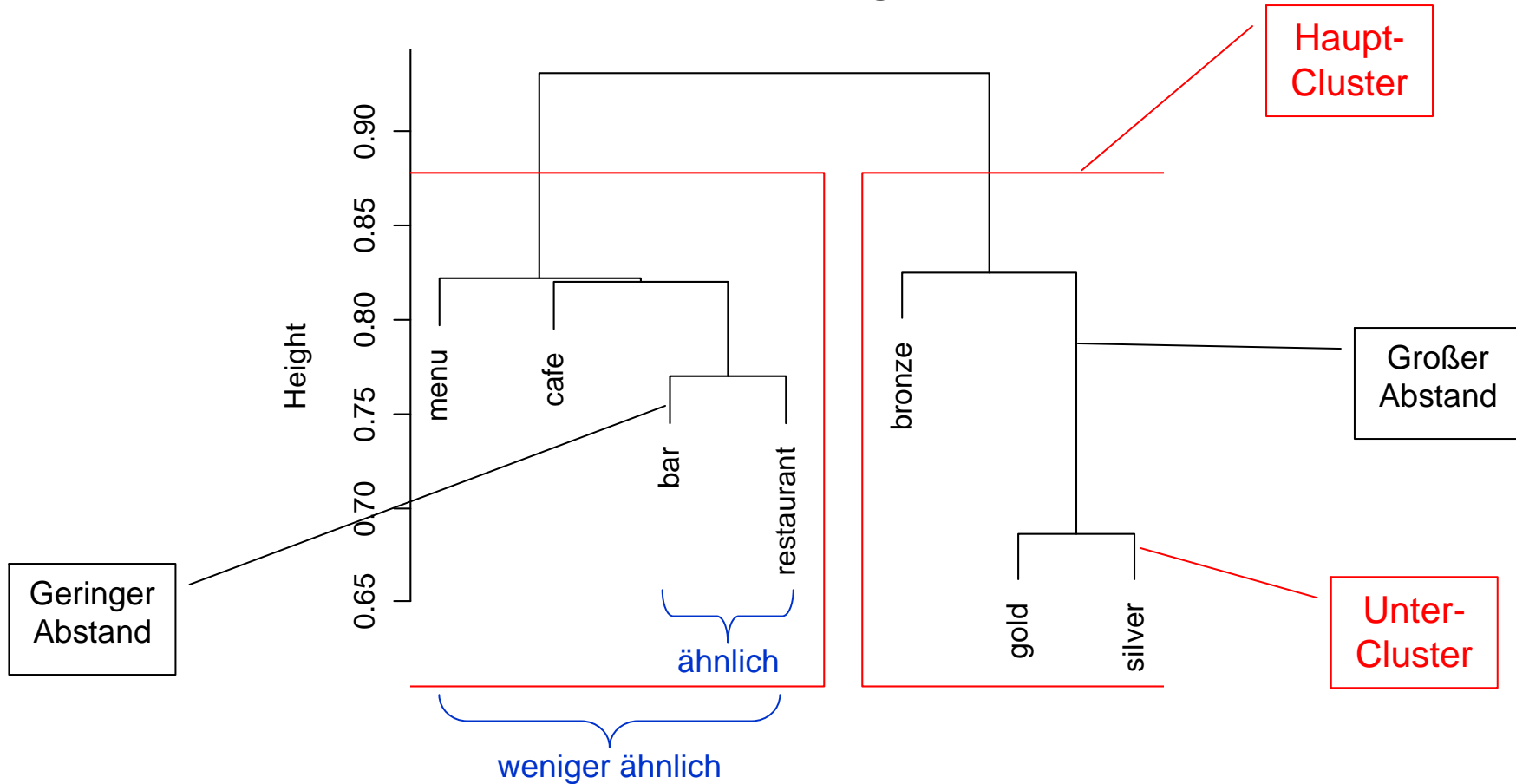
- In explorativen Verfahren gibt es keine Hypothesen, so wie man sie von statistischen Tests kennt
- Explorative Verfahren **generieren** Hypothesen
- Man nimmt an, dass die Daten sich aufgrund ihrer Merkmale in irgendwelche Gruppen teilen lassen
- Vorsicht:
  - Clustering ergibt aufgrund dieser Annahme **immer ein Ergebnis**
  - Das Ergebnis passt zu den Input-Daten – **Overfitting-Gefahr!**

# Das Ziel: Cluster bilden

- Es gibt viele Wege zur Gruppenbildung aus ungeordneten Daten
- Die besten Clustering-Verfahren sind:
  - Hierarchisch: Daten bilden Untergruppen, die wiederum zu Gruppen zusammengefasst werden (es gibt z.B. „Schwestergruppen“ etc.)
  - Proportional: der Abstand zwischen Schwesterpaaren kann unterschiedlich sein, je nach Ähnlichkeitsgrad

# Grafische Darstellung

Cluster Dendrogram



Dist(t\_kollokate, method = "pearson", diag = T, upper = T)  
hclust (\*, "single")

# Was heißt ähnlich?

- Um Ähnlichkeit zu berechnen, muss man jedes Element mit vergleichbaren **Merkmalen** versehen
- Das sind einfach Variablen:
  - jedes Blatt im Baum ist eine Beobachtung
  - für jede Beobachtung erheben wir Werte
  - können nominal- oder verhältnisskaliert sein

# Welche Merkmale können wir nehmen?

- Die Merkmale, die man benutzt, hängen von der Forschungsfrage und der Datenlage ab
- Hier: Clustering von Wortbedeutungen anhand von **Kollokationen**
- Wichtig dabei sind:
  - das Kookurenzkriterium (Operationalisierung)
  - das Kollokationsmaß (Deutung der Daten)

# Kollokationen als Merkmale/Variablen

- Wir erheben zunächst eine binäre Variable für jedes Wort, das wir Clustern wollen, mit jeder seiner Kollokationen
- Einfaches Kriterium:
  - Kommt in unseren Daten mit dem Wort *medal* vor: 1
  - Kommt vor dem Wort *medal* nicht vor: 0

# Beispiel: Nominalskala

- Die möglichen Ausprägungen für den Vergleich von *gold* und *silver*:

| Variable:<br><i>vor „medal“</i> | gold | $\neg$ gold |
|---------------------------------|------|-------------|
| silver                          | 1    | 0           |
| $\neg$ silver                   | 0    | 0           |

- In diesem Fall kommen beide Wörter vor *medal* vor, daher erhalten wir eine 1 oben links

# Schematisch

- Wir wiederholen die Erhebung für jede Variable und summieren die vier Zellen a, b, c, d nach folgendem Schema:

| Variable: $V$ | gold | $\neg$ gold |
|---------------|------|-------------|
| silver        | a    | b           |
| $\neg$ silver | c    | d           |

# Ähnlichkeit berechnen

Kommt vor  
*medal*

einsilbig

> gold = c(1, 1, 1, 1, 0, 0, 1, 0, 0, 0)

> silver = c(1, 1, 0, 1, 0, 1, 0, 1, 0, 1)

Ist Adjektiv und Nomen

- Sobald wir a,b,c,d gezählt haben, können wir die Ähnlichkeit berechnen:

$$\text{Ähnlichkeit} = \frac{a + w_1 \cdot d}{(a + w_1 \cdot d) + (w_2 \cdot (b + c))}$$

- Wert = 1 heißt identisch, 0 heißt völlig unterschiedlich
- Was sind  $w_1$  und  $w_2$ ?

# Die einstellbaren Koeffizienten

- $w_1$  und  $w_2$  sind einstellbar:
  - Jaccard-Koeffizient:  $w_1=0$  und  $w_2=1$
  - Simple-Match-Koeffizient:  $w_1=1$  und  $w_2=1$
  - Dice-Koeffizient:  $w_1=0$  und  $w_2=0.5$
- Wenn  $w_1=w_2=1$  hat die Anwesenheit eines Merkmals die gleiche Bedeutung wie seine Abwesenheit (dann benutzen wir *simple*)
- Das ist in der Linguistik selten der Fall:
  - wenn beide Wörter vor *medal* vorkommen ist das wichtig
  - wenn beide nicht vor *bark* vorkommen, ist es eher unwichtig
- Wir benutzen als erste Wahl den Jaccard-Koeffizienten, man kann aber immer verschiedene ausprobieren

# Verhältnisskala

- Oft haben wir mehr als nur nominale Merkmale:
  - Nicht nur ob etwas vor *medal* erscheint, sondern wie oft? Kollokationsstärke (Maß)?
  - Nicht nur einsilbig oder nicht, sondern wie viele Silben?
  - Frequenz von jedem Wort in unterschiedlichen Textsorten
  - ...

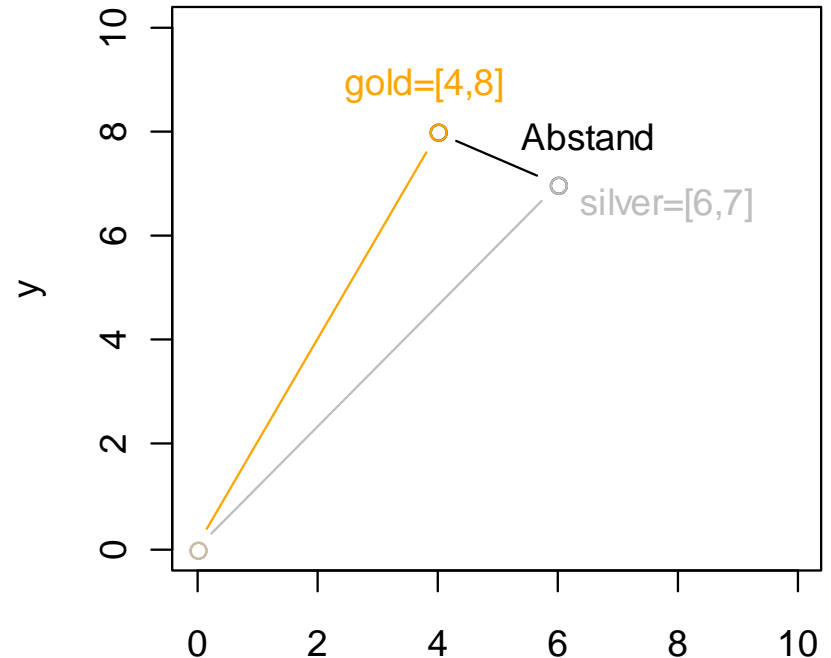
# Ähnlichkeit für Verhältnisskalen

- Nun sind unsere Vergleichsobjekte verhältnisskalierte Vektoren
- Beispiel gemeinsame Vorkommen:

|            | medal | food | eat | ring | pay | go |
|------------|-------|------|-----|------|-----|----|
| gold       | 19    | 0    | 0   | 14   | 2   | 3  |
| silver     | 12    | 1    | 0   | 3    | 0   | 0  |
| bar        | 0     | 5    | 12  | 0    | 0   | 23 |
| restaurant | 3     | 67   | 83  | 0    | 9   | 45 |

# Vektoren vergleichen

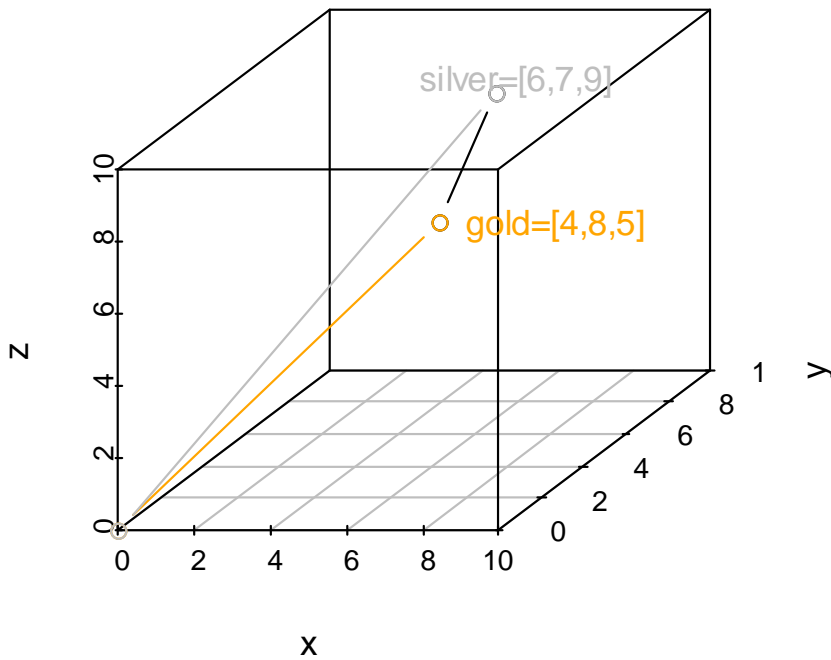
- Der intuitivste Vergleich zweier Vektoren ist ihr **Abstand**
- Wenn man sich zweistellige Vektoren als  $x,y$  Koordinaten vorstellt:



```
>plot(c(4,6), c(8,7), xlim=c(0,10), ylim=c(0,10),  
      type="b", xlab="x", ylab="y")  
>points(c(4,0), c(8,0), type="b", col="orange")  
>points(c(0,6), c(0,7), type="b", col="grey")  
>text(3.8,9, "gold=[4,8]", col="orange")  
>text(7.8,6.7, "silver=[6,7]", col="grey")  
>text(6.5,8,"Abstand")
```

# In 3 Dimensionen

- Das funktioniert genauso in 3D:



```
> library(scatterplot3d)
> my3d<-scatterplot3d(c(4,6), c(8,7), c(5,9),
  xlim=c(0,10), ylim=c(0,10),
  zlim=c(0,10), type="b", xlab="x",
  ylab="y", zlab="z")
> my3d$points3d(c(4,0), c(8,0), c(5,0), type="b",
  col="orange")
> my3d$points3d(c(0,6), c(0,7), c(0,9), type="b",
  col="grey")
> text(my3d$xyz.convert(6, 8, 9), col="grey",
  "silver=[6,7,9]")
> text(my3d$xyz.convert(7, 8, 5), col="orange",
  "gold=[4,8,5]")
```

# Und in $N$ Dimensionen

- Das können wir zwar schlecht darstellen...
- Aber die Formel für alle  $N$  ist einfach die sog. euklidische Distanz:

$$\sqrt{\left(\sum_{i=1}^N (a_i - b_i)^2\right)}$$

> `sqrt(sum(gold-silver)^2)`

\* (es gibt auch andere Distanzmaße, die wir hier außer Acht lassen)

# Distanzmatrix in der Library *amap*

- Mit *amap* geht das automatisch für ganze Tabellen:

```
> Library(amap)
> (cooc <- matrix(c(19,0,0,14,2,3, 12,1,0,3,0,0,
0,5,12,0,0,23), nrow=6, ncol=3,
dimnames= list(c("a","b","c","d","e","f"),
c("gold","silver","bar"))))
```

|   | gold | silver | bar |
|---|------|--------|-----|
| a | 19   | 12     | 0   |
| b | 0    | 1      | 5   |
| c | 0    | 0      | 12  |
| d | 14   | 3      | 0   |
| e | 2    | 0      | 0   |
| f | 3    | 0      | 23  |

# Distanzmatrix in der Library *amap*

- Die Vektoren müssen jedoch als Zeilen vorliegen, daher nutzen wir `t()` (transpose):

```
> t(cooc)
```

```
      a b  c  d e  f
gold  19 0  0 14 2  3
silver 12 1  0  3 0  0
bar    0 5 12  0 0 23
```

```
> Dist(t(cooc), method="euclidean", diag=TRUE,
      upper=TRUE)
```

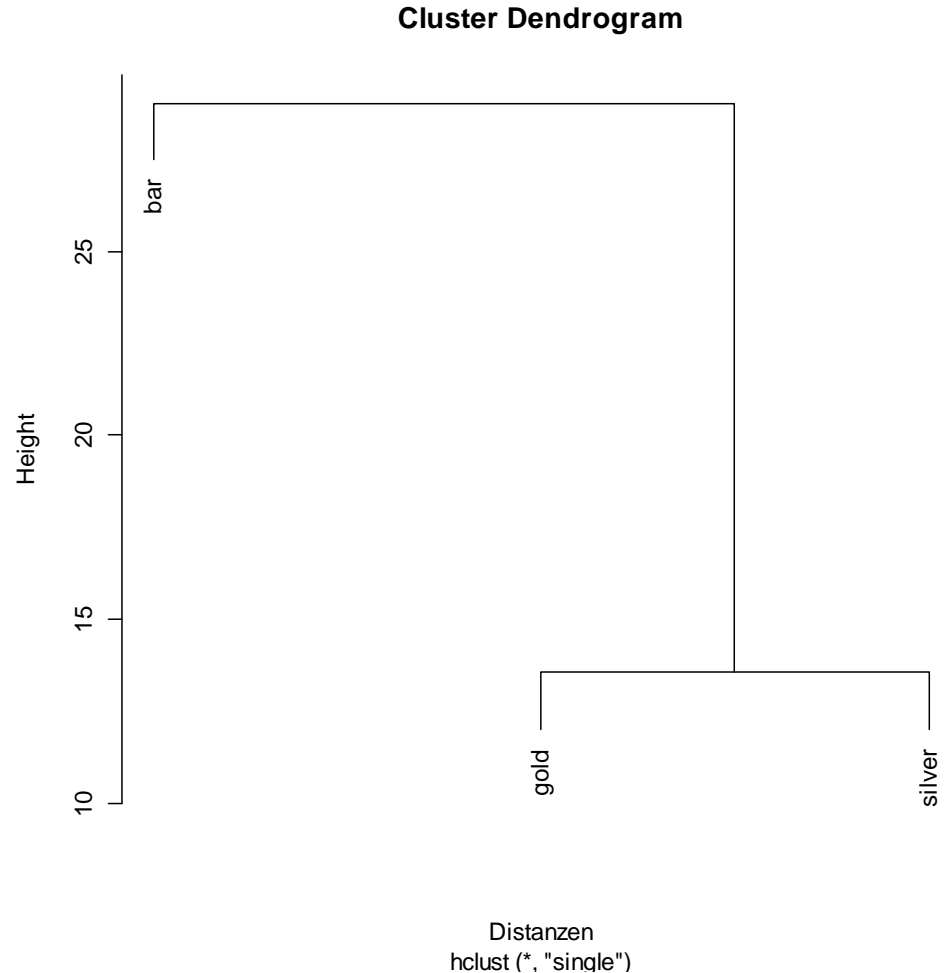
```
      gold  silver  bar
gold  0.00000 13.56466 33.61547
silver 13.56466  0.00000 29.01724
bar   33.61547 29.01724  0.00000
```

# Distanzmatrix clustern...

- Diese Matrix können wir nach folgendem Prinzip clustern (sog. „single linkage“ bzw. „nearest neighbor“):
  - Finde immer das Paar mit dem kleinsten Abstand und verbinde seine Bestandteile
  - Gilt der nächste kleinste Abstand zwischen einem schon gruppierten Vektor und einem anderen, verbinde sie alle zu einer größeren Gruppe
- D.h. zunächst silver+gold (kleinster Abstand)
- Dann bar+silver, aber silver ist schon in einer Gruppe, also: [bar+[silver+gold]]

# So entsteht unser Diagramm

```
> Distanzen <-  
  Dist(t(cooc),  
       method="euclidean",  
       diag=TRUE, upper=TRUE)  
> clust.ana <-  
  hclust(Distanzen,  
        method="single")  
> plot(clust.ana)
```



# Ein echtes Beispiel

- Wir möchten die Monate nach ihren Kollokationen klassifizieren
- In diesem Beispiel: attributive Adjektive aus einem Korpus mit 3,5 Mio Token
- Die zuklassifizierenden Wörter:
  - Januar
  - März
  - Mai
  - Juni
  - Juli
  - August
  - Oktober
  - Jahr (zur Kontrolle)



# Berechnung der Distanzmatrix

```
> distances <- Dist(t(kook[,2:9]),  
  method="euclidean", diag=TRUE, upper=TRUE)
```

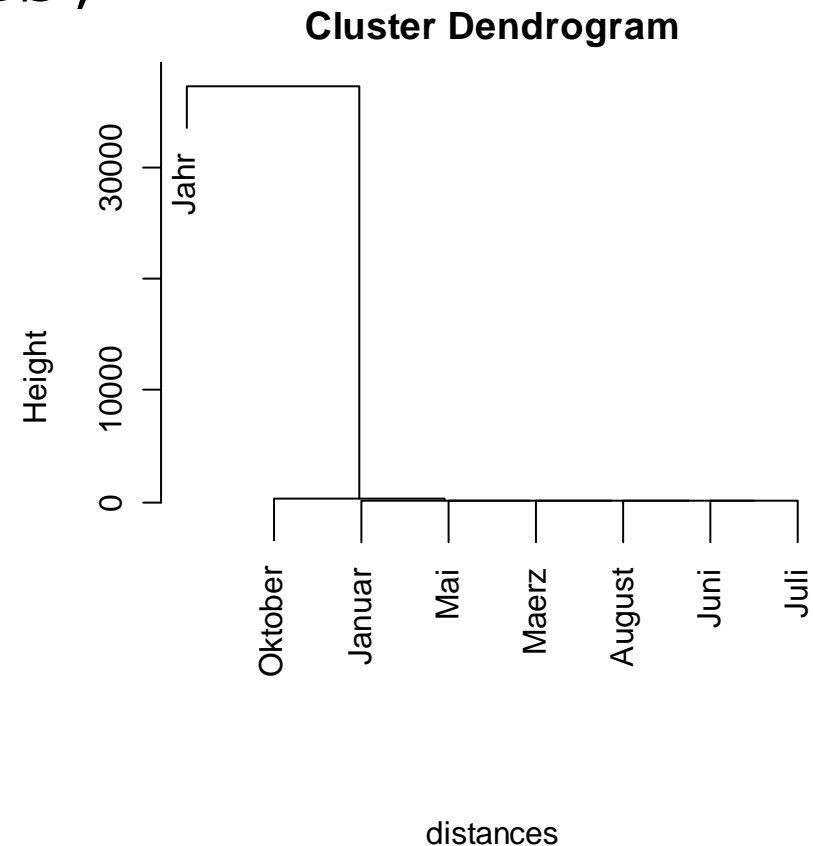
```
> distances
```

|        | Januar    | Maerz     | Mai...    |
|--------|-----------|-----------|-----------|
| Januar | 0.00000   | 96.15092  | 135.40310 |
| Maerz  | 96.15092  | 0.00000   | 103.31021 |
| Mai    | 135.40310 | 103.31021 | 0.00000   |
| ...    |           |           |           |

# 1. Versuch

```
> plot(hclust(distances,  
method="single"))
```

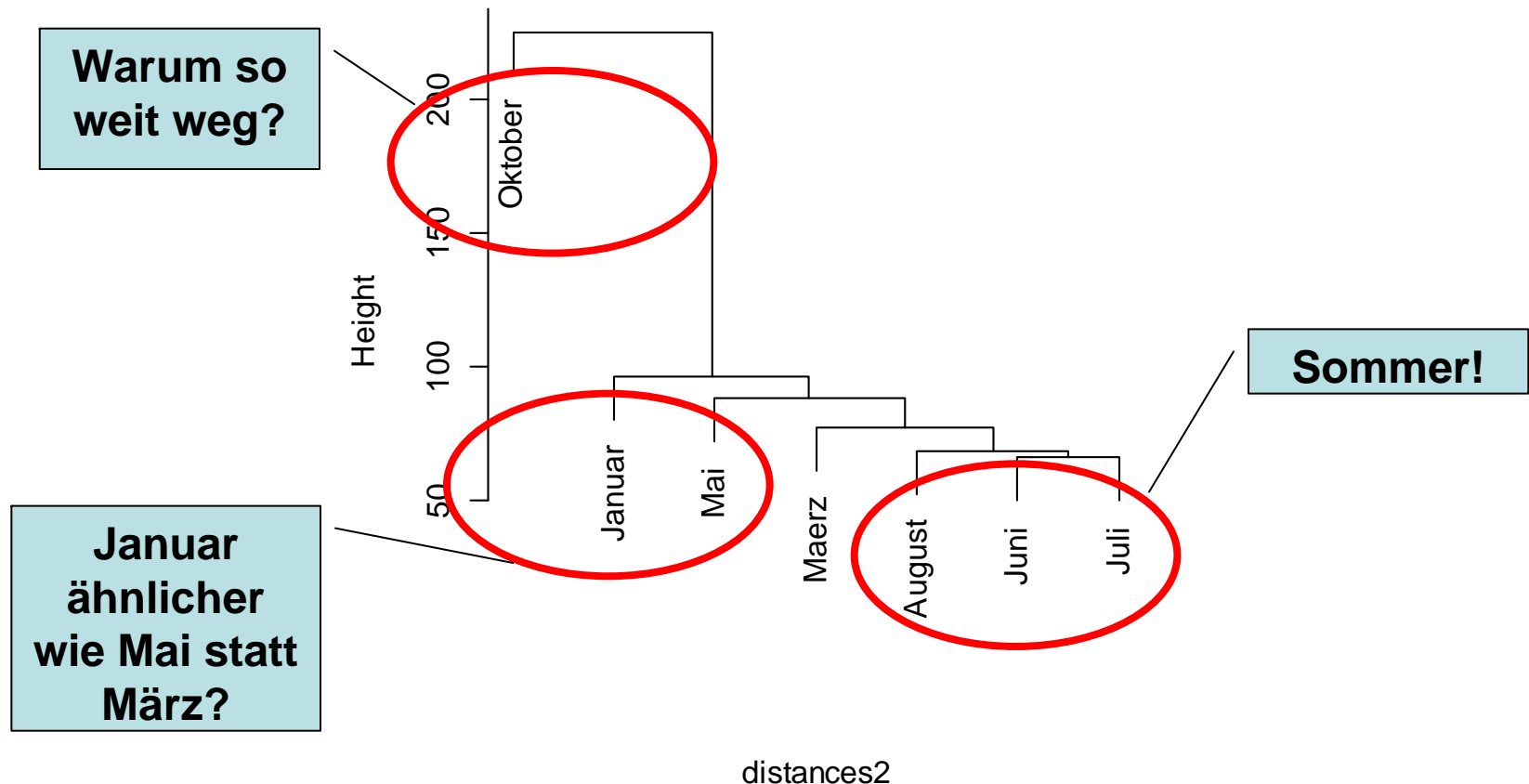
- Die Kontrolle funktioniert
- *Jahr* wird ausgeklammert
- ...aber wir können nichts sehen



# 2. Versuch

```
> distances2 <- Dist(t(kook[,2:8]),  
method="euclidean", diag=TRUE, upper=TRUE)  
> plot(hclust(distances2, method="single"))
```

Cluster Dendrogram



# Daten sortieren

```
>kook_akt <- kook[order(kook[, "Oktober"],  
  decreasing=T), ]  
>kook_akt[1:9, ]
```

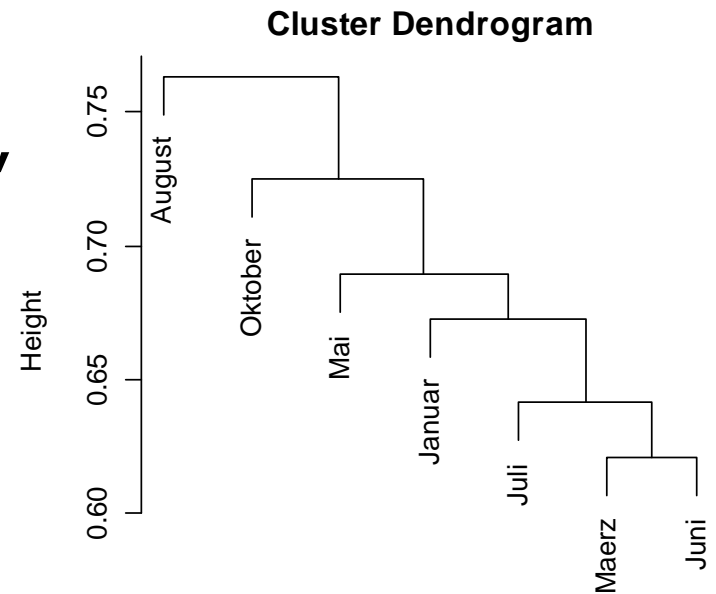
|     | ADJA     | Januar | Maerz | Mai | Juni | Juli | August | Oktober    | Jahr  |
|-----|----------|--------|-------|-----|------|------|--------|------------|-------|
| 883 | vergehen | 58     | 117   | 172 | 194  | 159  | 210    | <b>402</b> | 21486 |
| 565 | letzt    | 14     | 29    | 54  | 56   | 42   | 86     | <b>153</b> | 24609 |
| 57  | rot      | 0      | 0     | 0   | 0    | 0    | 0      | <b>66</b>  | 0     |
| 28  | golden   | 0      | 0     | 0   | 0    | 0    | 0      | <b>40</b>  | 43    |
| 313 | dritt    | 1      | 5     | 9   | 3    | 2    | 3      | 32         | 211   |
| 533 | kommen   | 46     | 47    | 54  | 49   | 22   | 19     | 31         | 5288  |
| 930 | vorig    | 1      | 3     | 8   | 11   | 9    | 9      | 27         | 1851  |
| 371 | erst     | 84     | 15    | 78  | 19   | 46   | 18     | 25         | 1346  |
| 37  | kalt     | 8      | 0     | 1   | 0    | 0    | 0      | 13         | 0     |

# Andere Methoden - Distanz

- Wie viele Gemeinsamkeiten?

```
> distances3 <-  
  Dist(t(kook[, 2:8]),  
       method="binary",  
       diag=TRUE,  
       upper=TRUE)
```

- oder maximum,  
 manhattan...



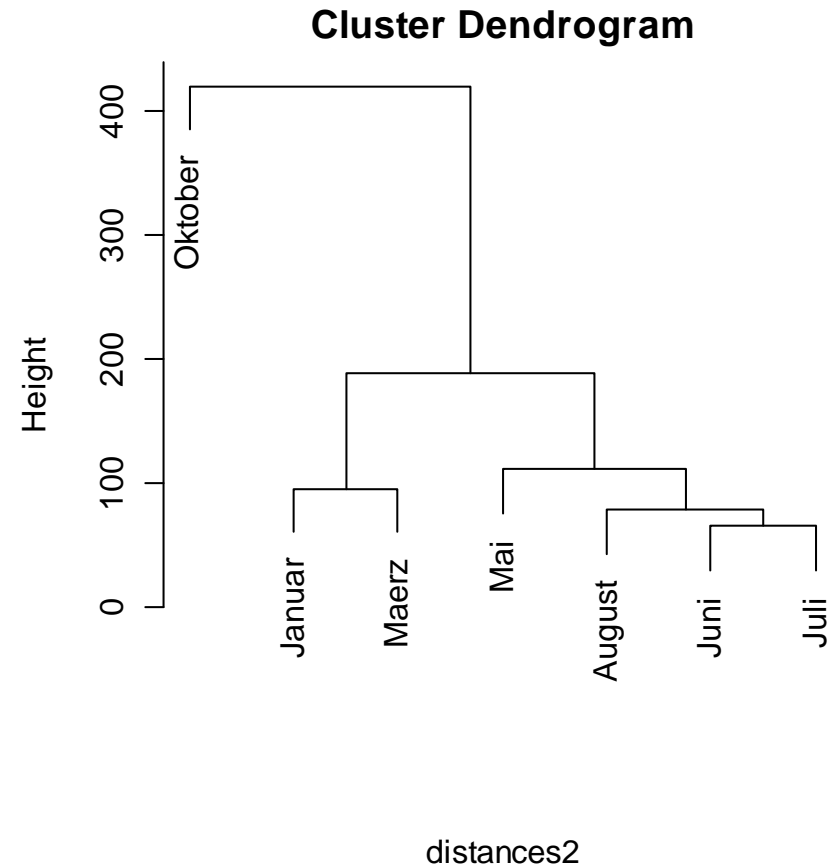
distances3

# Andere Methoden - Clusterbildung

- Die Methode „single“ betrachtet nur das ‚ähnlichste Blatt‘
- Andere Methoden betrachten Ähnlichkeit mit dem Mittelwert der Klasse, bspw. „ward“

```
>plot(hclust(distances2, method="ward"))
```

- Offene Frage: was ist „richtiger“?



# Zusammenfassung

- Clustering bildet Gruppen aus beliebigen Merkmalen
- Wie bei Dr. Oetker: „gelingt immer“
- Ergebnisse unterschiedlich je nach Methode
- Sehr viele weitere Anwendungen:
  - Ähnlichkeit zwischen Wörtern (Umgebung, vordefinierte Merkmale)
  - Probanden (ähnliche Antworten im Fragebogen, RT...)
  - Stimuli (produzieren ähnliche Ergebnisse, andere Merkmale...)

# Literatur

- Baayen, R. H. (2008). *Analyzing Linguistic Data. A Practical Introduction to Statistics using R*. Cambridge: Cambridge University Press.
- Bresnan, J., Cueni, A., Nikitina, T., and Baayen, R. H. (2007). Predicting the dative alternation. In Bouma, G., Kraemer, I., and Zwarts, J. (eds.), *Cognitive Foundations of Interpretation*, pages 69-94. Amsterdam: Royal Netherlands Academy of Science.
- Gries, S. T. (2008). *Statistik für Sprachwissenschaftler*. Göttingen: Vandenhoeck & Ruprecht.