Die logistische Regression

Jonathan Harrington & Ulrich Reubold

25. Juni 2019

```
library(ggplot2)
library(dplyr)
source(file.path(pfadu, "sig.fn.R"))
ovokal = read.table(file.path(pfadu, "ovokal.txt"))
pvp = read.table(file.path(pfadu, "pvp.txt"))
sz = read.table(file.path(pfadu, "sz.txt"))
```

Einleitung: Logistische Regression

Mit der logistischen Regression wird geprüft, ob Proportionen (abhängige Variable) von einem (oder mehreren) unabhängigen Faktoren beeinflusst werden.

- Die abhängige Variable ist immer kategorial und immer binär.
- Die unabhängige Variable kann numerisch oder kategorial (auch mehrstufig) sein.

Beispiele:

1. Inwiefern wird die Vokalisierung von einem finalen /l/ im Englischen (feel vs. 'feeu') vom Dialekt beeinflusst?

Abhängige Variable: Vokalisierung (kategorial, 2 Stufen: ja, nein) Unabhängige Variable: Dialekt (kategorial: 2 oder mehrere Stufen)

2. Wird ein Wort wie 'passt' in Augsburg im Vgl. zu München eher mit /ʃ/ produziert? Abhängige Variable: Frikativ (kategorial, 2 Stufen: /s/, /ʃ/) Unabhängige Variable: Dialekt (kategorial, 2 Stufen: Augsburg, München)

3. Ein offener Vokal in /lVm/ wird mit unterschiedlichen Dauern synthetisiert. Nimmt die Wahrnehmung von 'lahm' vs. 'Lamm' mit zunehmender Dauer zu? Abhängige Variable: Vokal (kategorial, 2 Stufen: /a:/, /a/) Unabhängige Variable: Dauer (kontinuierlich)

In der linearen (least-squares) Regression (siehe hier) wurde geprüft, inwiefern eine lineare Beziehung zwischen zwei Variablen, x und y vorliegt. Um dies zu tun, wurde eine Regressionslinie mit Steigung m und Intercept k an die Stichproben angepasst.

$yhut$ in...

$$yhut = mx + k$$

sind dann die eingeschätzen Werte, die auf der Regressionslinie liegen

Wir können aber eine solche Linie nicht an Proportionen anpassen, weil Proportionen zwischen 0 und 1 begrenzt sind (und die lineare Regression erwartet Werte zwischen ±unendlich). Stattdessen wird eine gerade Linie an sogenannte Log(Odds) angepasst

$$log(Odds) = mx + k$$

Odds (Odds = Gewinnchancen): P/Q.

- P ist 'Erfolg' - z.B. in Bsp. 3 Häufigkeit der 'lahm'-Urteile
- Q ist 'Misserfolg':Häufigkeit der 'Lamm'-Urteile
- $log(Odds)$ ist dann einfach $log(P/Q)$

$log(Odds)$ haben Werte zwischen $-\infty$ und $+\infty$ (∞ = 'unendlich').

1. Erfolg *P*, Misserfolg *Q*, Log-Odds *log(P/Q)*, und Proportionen *p*

```
head(ovokal)
```

```
##   Jahr Vokal Vpn
## 1 1950  hoch  S1
## 2 1950  hoch  S2
## 3 1950  hoch  S3
## 4 1950  hoch  S4
## 5 1950  hoch  S5
## 6 1950  hoch  S6
```

Zwischen 1950 und 2005 sollen Wörter wie 'lost' in einer aristokratischen Form der Standardaussprache von England immer weniger mit einem halb-geschlossenem Vokal /lo:st/ (nachfolgend: "hoch") und zunehmend mit einem halb-offenem Vokal /lɔst/ (nachfolgend: "tief") produziert. Ist dies wirklich der Fall? In anderen Worten:

- Wird der Vokal (hoch vs. tief = abhängige Variable) vom Jahr (1950... 2005 = unabhängige numerische Variable) beeinflusst?

```
################################## P ist 'Erfolg'
# Wir nehmen den zweiten Wert von
levels(ovokal$Vokal)
```

```
## [1] "hoch" "tief"
```

```
# als P
P = ovokal$Vokal == "tief"
################################## Q ist 'Misserfolg' (= nicht Erfolg)
Q = !P
# P, Q in den Data-Frame einbinden
ovokal = cbind(ovokal, P, Q)
################################## Die Summe von P, Q pro Jahr berechnen
### hierbei Vpn ignorieren (=über alle Antworten aufsummieren)
ovokal.m = ovokal%>%
  group_by(Jahr)%>%
  summarise(P=sum(P),Q=sum(Q))
ovokal.m
```

```
## # A tibble: 6 x 3
##    Jahr     P     Q
##   <int> <int> <int>
## 1  1950     5    30
## 2  1960    21    18
## 3  1971    26    15
## 4  1980    20    13
## 5  1993    32     4
## 6  2005    34     2
```

Reihe 1 bedeutet: es gab 5 Mal Erfolg (tief) und 30 Mal Misserfolg (hoch) in 1950.

Nun müssen wir die Proportionen mit P/(P+Q) berechnen:

```
# Proportionen berechnen = die Proportion vom Erfolg (0 <= p <= 1)
# neue Spalte p mit mutate() erstellen:
ovokal.m = ovokal.m %>%
  mutate(p = P/(P+Q))

ovokal.m
```

```
## # A tibble: 6 x 4
##    Jahr     P     Q     p
##   <int> <int> <int> <dbl>
## 1  1950     5    30 0.143
## 2  1960    21    18 0.538
## 3  1971    26    15 0.634
## 4  1980    20    13 0.606
## 5  1993    32     4 0.889
## 6  2005    34     2 0.944
```

Dann Log-odds ("lodd") berechnen: $log(P/Q)$:

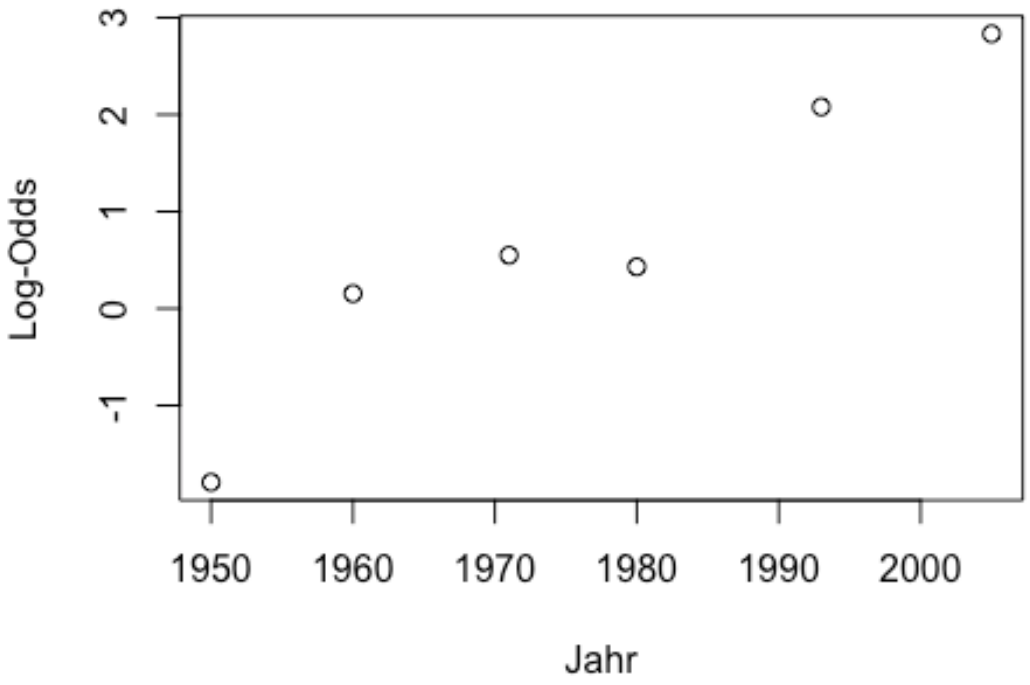
```
ovokal.m = ovokal.m %>%
  mutate(lodd = log(P/Q))
ovokal.m
```

```
## # A tibble: 6 x 5
##    Jahr     P     Q     p   lodd
##   <int> <int> <int> <dbl>  <dbl>
## 1  1950     5    30 0.143 -1.79
## 2  1960    21    18 0.538  0.154
## 3  1971    26    15 0.634  0.550
## 4  1980    20    13 0.606  0.431
## 5  1993    32     4 0.889  2.08
## 6  2005    34     2 0.944  2.83
```

Die logistische Regressionslinie

Abbildung im Raum *Log-Odds x unabhängige Variable*:

```
plot(lodd ~ Jahr, data = ovokal.m, ylab="Log-Odds")
```



Eine Regressionslinie in diesem Raum wird mit `glm(...,family=binomial)` berechnet. `glm()` steht für 'generalised linear model'.

Entweder Anwendung auf den ursprünglichen Data-Frame:

```
head(ovokal)
```

```
##   Jahr Vokal Vpn     P    Q
## 1 1950  hoch  S1 FALSE TRUE
## 2 1950  hoch  S2 FALSE TRUE
## 3 1950  hoch  S3 FALSE TRUE
## 4 1950  hoch  S4 FALSE TRUE
## 5 1950  hoch  S5 FALSE TRUE
## 6 1950  hoch  S6 FALSE TRUE
```

```
lreg = glm(Vokal ~ Jahr, family=binomial, data = ovokal)
```

Oder Anwendung auf die zusammengefassten Spalten P und Q in dem summierten Data-Frame:

```
lreg = glm(cbind(P, Q) ~ Jahr, family=binomial, data = ovokal.m)
```

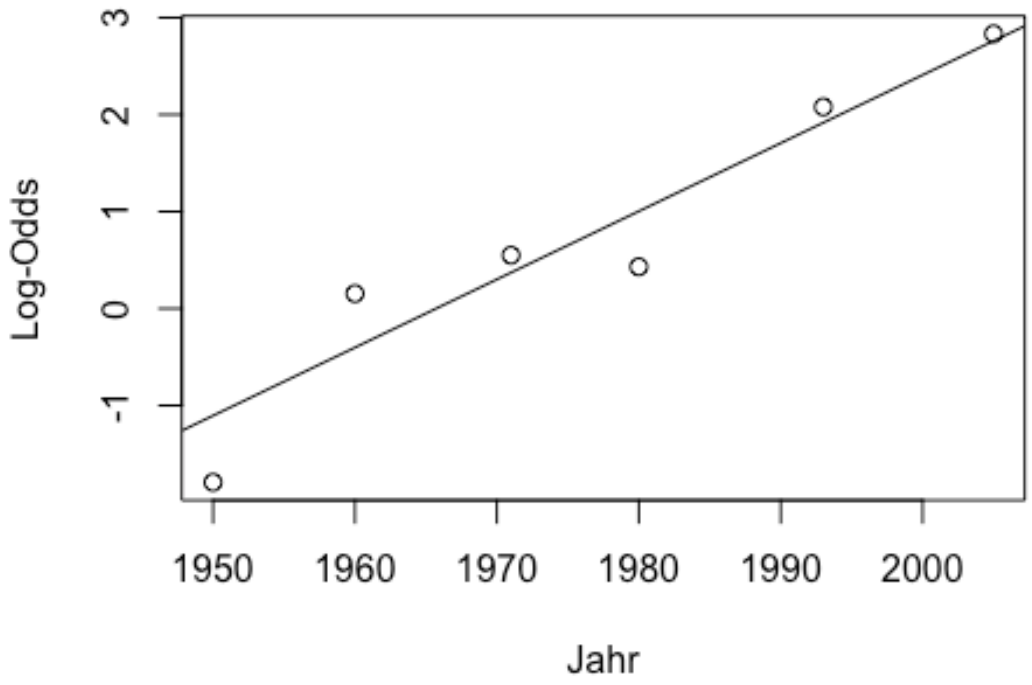
Intercept und Steigung

```
# Das Intercept und die Steigung sind hier:
coef(lreg)
```

```
##   (Intercept)          Jahr
## -138.11741925    0.07026313
```

Mit Hilfe dieser Maße kann die (lineare) Regressionslinie auf die Daten im Log-Odds Raum überlagert werden:

```
plot(lodd ~ Jahr, data = ovokal.m, ylab="Log-Odds")
abline(lreg)
```



Die Prüfstatistik (χ^2-Test)

Die Prüfstatistik überprüft die Wahrscheinlichkeit, dass die Steigung von Null abweicht (heißt: wenn die Steigung Null wäre, dann wären alle log-odds gleich und wir hätten eine gerade Linie); dies wird mit einem sogenannten χ^2-Test oder Chi-Quadrat-Test gemacht (daher "Chisq" als Code für 'Chi-squared'):

```
anova(lreg, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: cbind(P, Q)
##
```

```
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                     5     69.363
## Jahr  1   61.121         4      8.242 5.367e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Ein mögliche Antwort auf die obige Frage wäre also: *Jahr hatte einen signifikanten Einfluss auf die Proportion von 'lost' mit tiefem/hohem Vokal* ($\chi^2[1] = 61.1, p < 0.001$).

Die Sigmoidal-Funktion (`sig()`) und der Umkipppunkt (`-k/m`)

Die Regressionslinie wird berechnet im Raum *Log-Odds x unabhängige Variable (Jahr)*. Dieselben Intercept- und Steigungswerte (k, m) können verwendet werden, um statt Log-Odds auf der y-Achse **Proportionen** abzubilden. In dem Fall wandelt sich die gerade Linie in eine sogenannte **Sigmoid**-Funktion um.

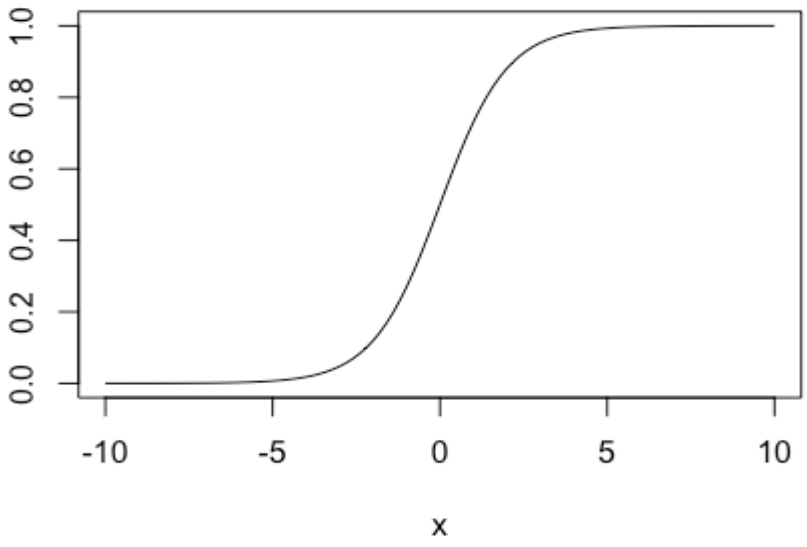
Die mathematische Formel für eine Sigmoid-Funktion lautet:

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

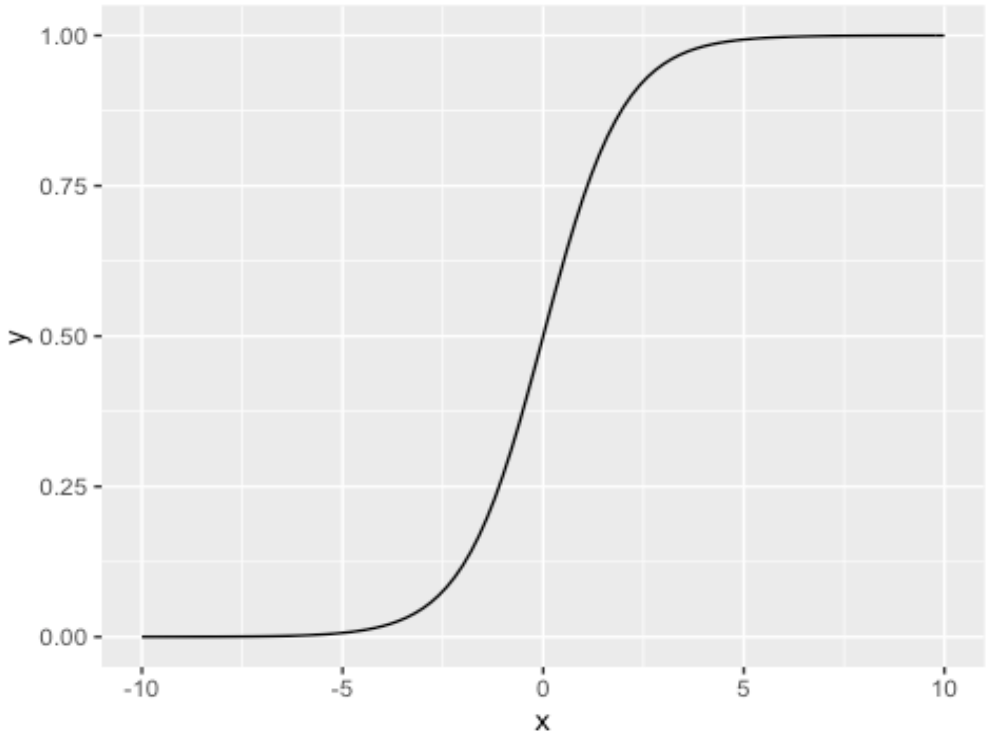
e ist die Exponentialfunktion, m und k sind die Steigung und das Intercept.

Umsetzung in R:

```
sig()
```

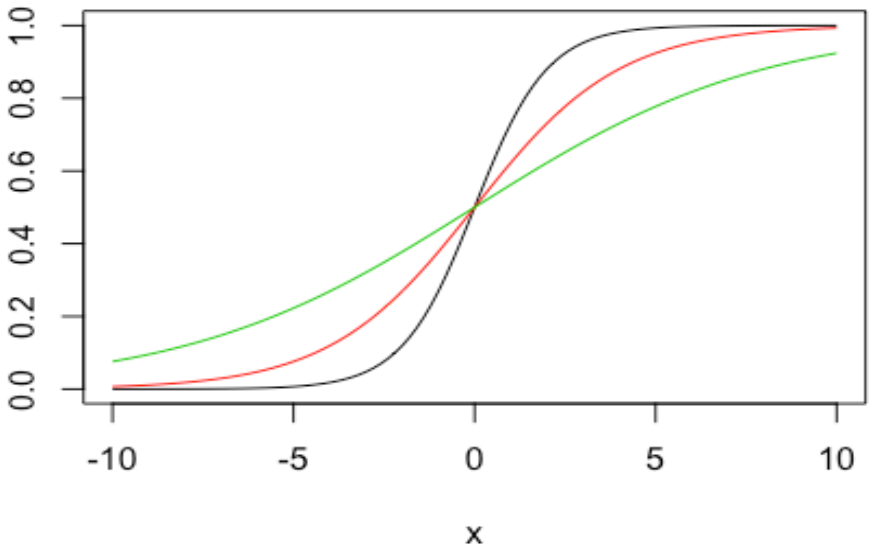


```
# oder
geom_sig()
```

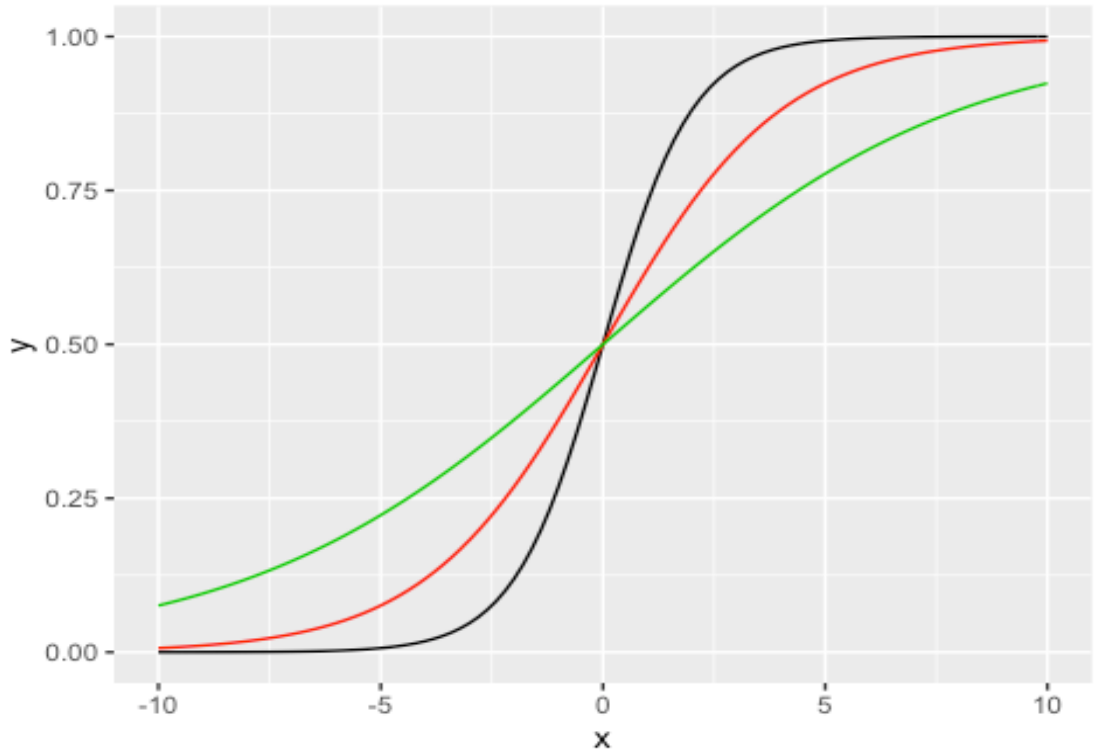


m ist die Steigung: je größer m, umso steiler kippt die S-Kurve um:

```
# Steigungen von 1, .5, , .25 (schwarz, rot, grün)
sig(m = c(1, .5, .25))
```

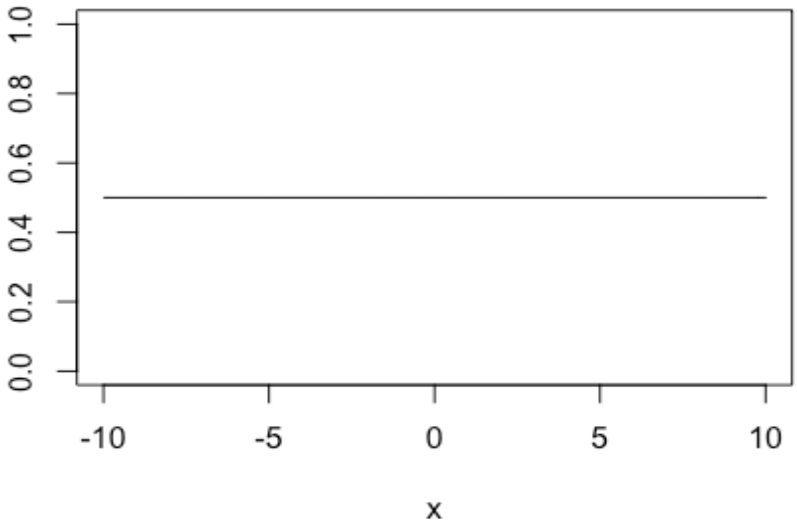


```r
# oder:
geom_sig(m = c(1, .5, .25))
```

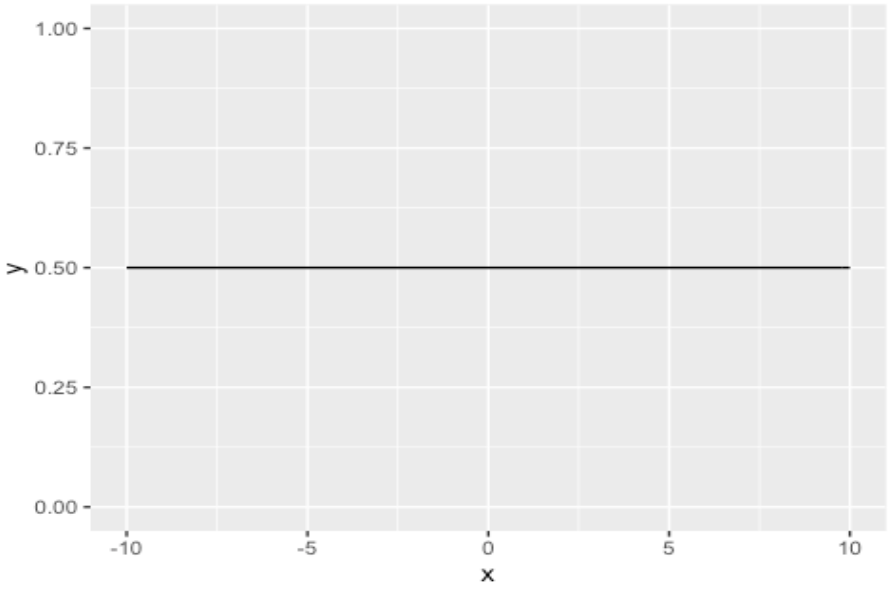


Wenn die Steigung 0 (Null;k = 0) ist, bekommt man eine gerade Linie um den Wert 0.5 auf der y-Achse:

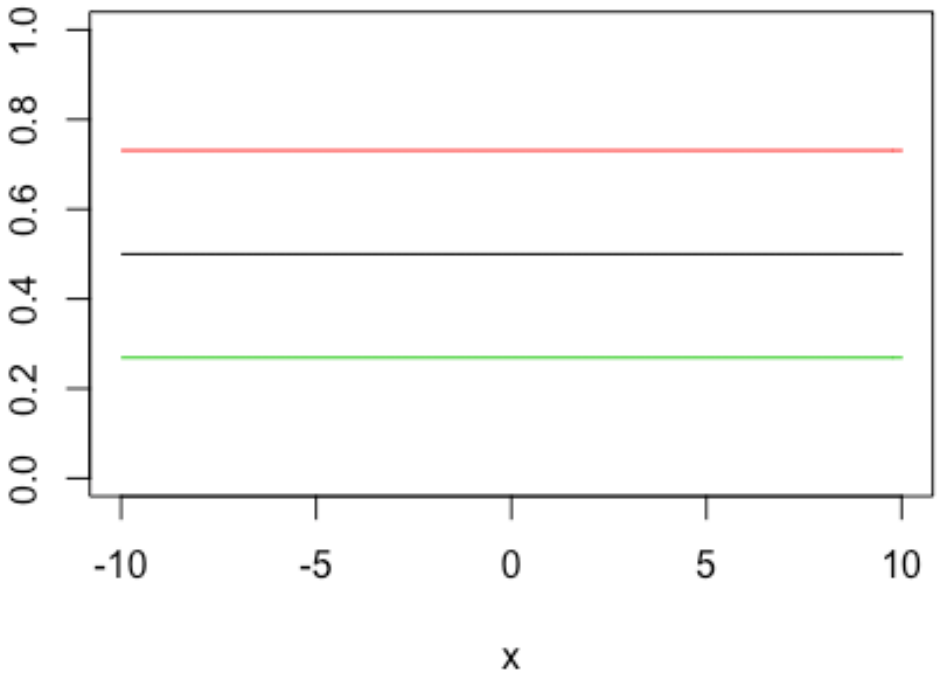
```r
sig(m = 0)
```



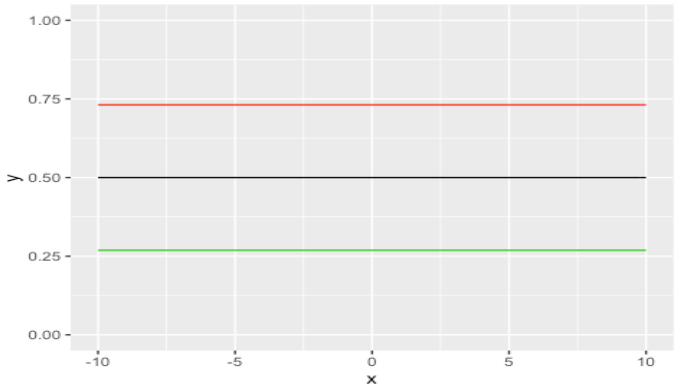
```r
# oder
geom_sig(m = 0)
```




```r
# Höhere/tiefere k-Werte verschieben die Linie um den 0.5 Wert
sig(k = c(0, 1, -1), m = 0)
```



```r
# oder
geom_sig(k = c(0, 1, -1), m = 0)
```

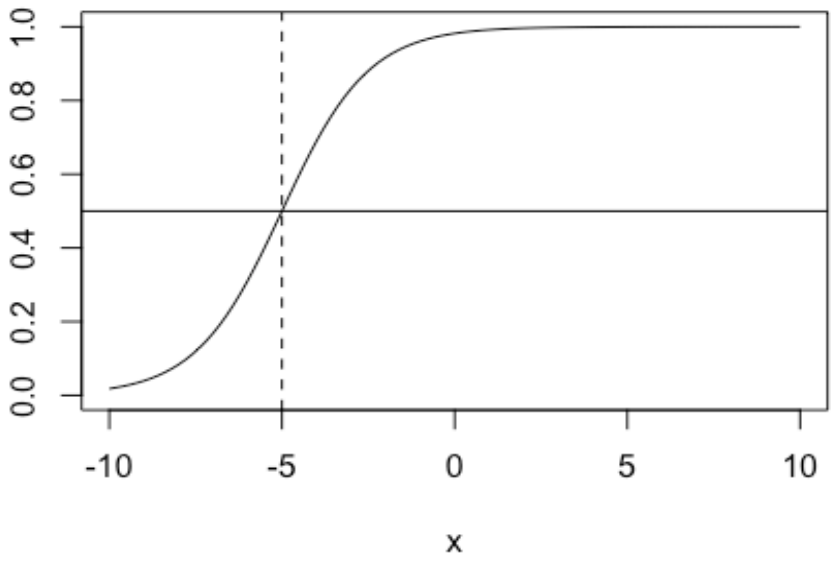


Der Umkipppunkt

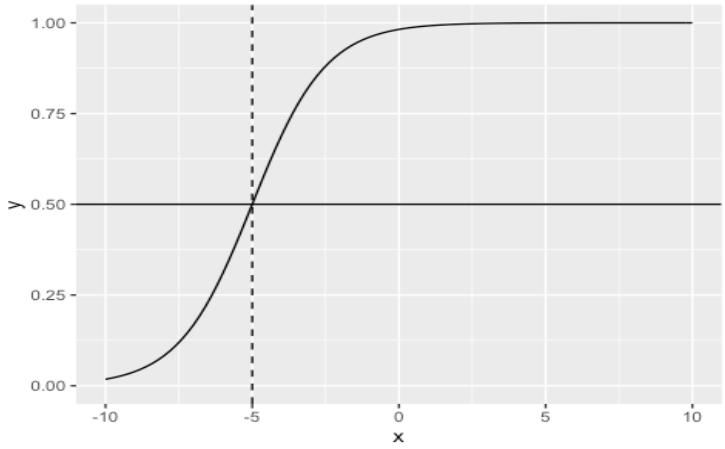
Der **Umkipppunkt** is der Punkt, zu dem (i) der Sigmoid **am steilsten** ist. *An diesem Punkt* **ist die Proportion** (auf der vertikalen Achse) **immer 0.5**.

Den Umkipppunkt bekommt man mit $-k/m$.

```r
sig(k = 4, m = .8)
# Hier ist m = 0.8, k = 4
# Daher u = -k/m = -4/.8 = -5
#vertikale Linie:
abline(v = -5, lty="dashed")
#horizontale Linie:
abline(h = .5)
```

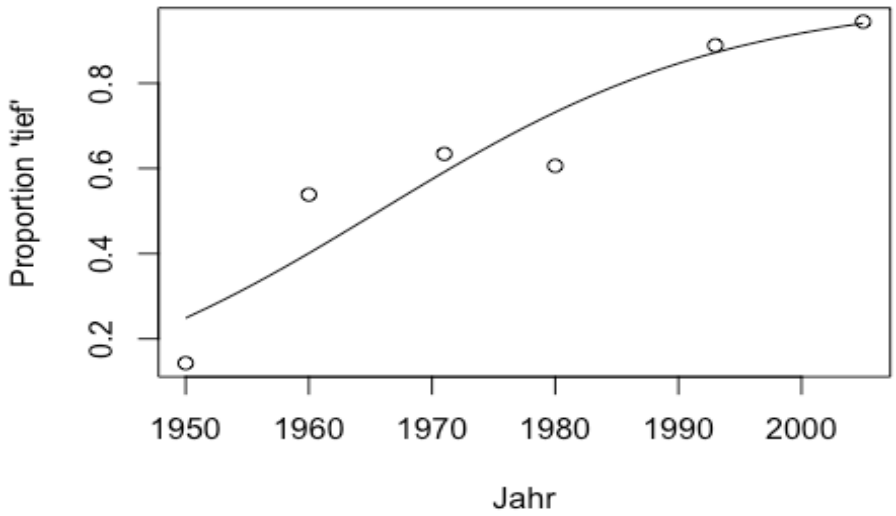


```r
# oder
geom_sig(k = 4, m = .8) +
  geom_vline(xintercept = -5,lty = "dashed") +
  geom_hline(yintercept = .5)
```

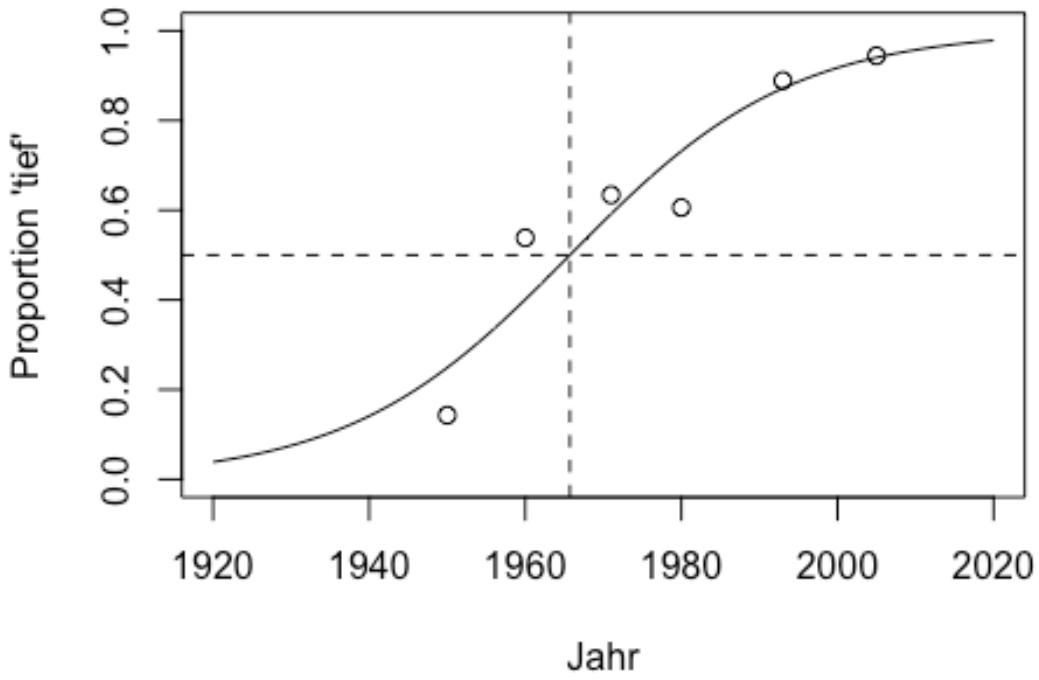


5. Proportionen abbilden

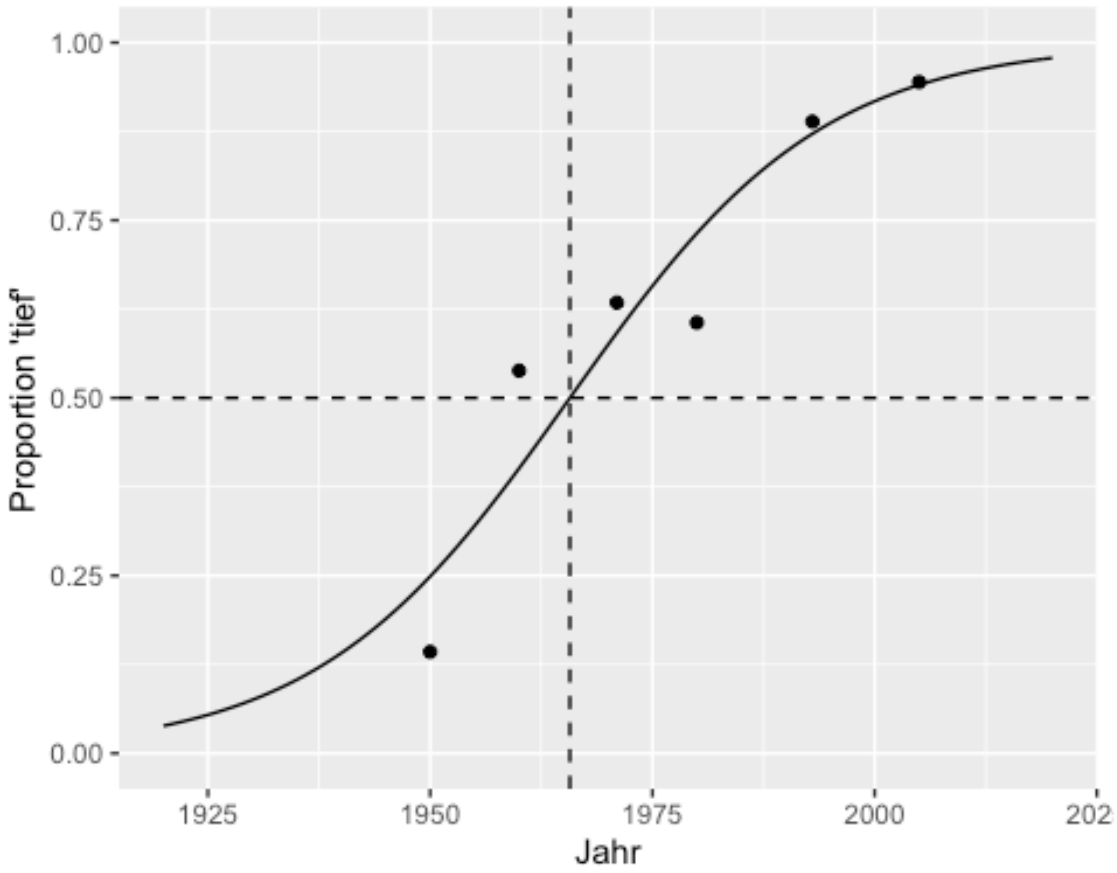
```r
plot(p ~ Jahr, data = ovokal.m, ylab = "Proportion 'tief'")
# von vorher
lreg = glm(Vokal ~ Jahr, family=binomial, data = ovokal)
# Intercept
lreg.k = coef(lreg)[1]
# Steigung
lreg.m = coef(lreg)[2]
# Angepasste Sigmoid
sig(lreg.k, lreg.m, add=T)
```



```r
# verifizieren, dass es wirklich ein Sigmoid ist!
# Mittel hierzu: die Abbildung verbreitern:
plot(p ~ Jahr, data = ovokal.m, xlim = c(1920, 2020), ylim = c(0, 1),
ylab = "Proportion 'tief'")
sig(lreg.k, lreg.m, add=T)
abline(v = -lreg.k/lreg.m, lty = "dashed")
abline(h = .5, lty="dashed")
```




```r
# das Gleiche mit (dem wesentlich komplizierteren Befehl)
ggplot(ovokal.m) +
  aes(y = p, x = Jahr) +
  geom_point() +
  ylab("Proportion 'tief'") +
  geom_sig(lreg.k, lreg.m, add=T) +
  geom_vline(xintercept = -lreg.k/lreg.m, lty = "dashed") +
  geom_hline(yintercept = .5, lty = "dashed") +
  scale_x_continuous(limits = c(1920,2020)) +
  scale_y_continuous(limits = c(0,1))
```



Die vertikale Linie zeigt den **Umkipppunkt** (= das Jahr, zu dem sich laut Modell die Entscheidungen von *hoch* auf *tief* wandelt):

```
-lreg.k/lreg.m

## (Intercept)
##    1965.717

# 1965.717
```

6. Umkipppunkte in einem synthetischen Kontinuum

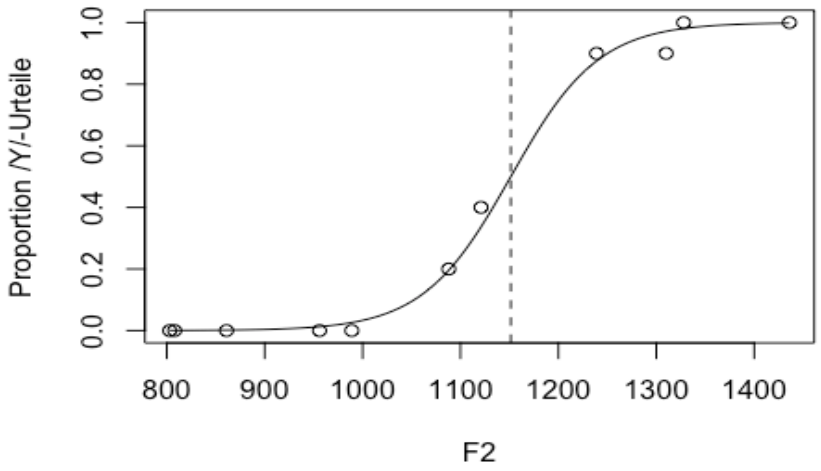
Ein 11-stufiges Kontinuum wurde synthesisert zwischen /pUp/ und /pYp/. Die Stimuli wurden 10 Mal einem Hoerer einzeln praesentiert. Der Hörer musste pro Stimulus entscheiden: PUPP oder PÜPP?

- **Zu welchem F2-Wert kommt der Umkipppunkt vor?** (= zu welchem F2-Wert kippt die Entscheidung um von PUPP auf PUEPP?)

Hierzu müssen wir P, Q, $Proportionen$, und die $logodds$ berechnen:

```
levels(pvp$Urteil)

## [1] "U" "Y"
```

```r
# Erfolg
P = pvp$Urteil == "Y"
# Misserfolg
Q = !P
# in den Data-Frame einbinden
pvp = cbind(pvp, P, Q)

# summieren pro F2-Wert
pvp.sum = pvp %>%
  group_by(F2) %>%
  summarise(P = sum(P),Q = sum(Q))

# Proportionen
pvp.sum = pvp.sum %>%
  mutate(p = P/(P+Q))

plot(p ~ F2, data = pvp.sum, ylab = "Proportion /Y/-Urteile")

# (k,m) der Sigmoid berechnen
pvp.glm = glm(Urteil ~ F2, family=binomial, data = pvp)
# oder
pvp.glm = glm(cbind(P, Q) ~ F2, family=binomial, data = pvp.sum)

# Koeffiziente
pvp.k = coef(pvp.glm)[1]
pvp.m = coef(pvp.glm)[2]
sig(pvp.k, pvp.m, add=T)

# Umkipppunkte
u = -pvp.k/pvp.m
abline(v = u, lty=2)
```



```r
# Die Wahrscheinlichkeit, dass die Urteile
# durch F2-Änderungen beeinflusst werden:

anova(pvp.glm, test = "Chisq")
```

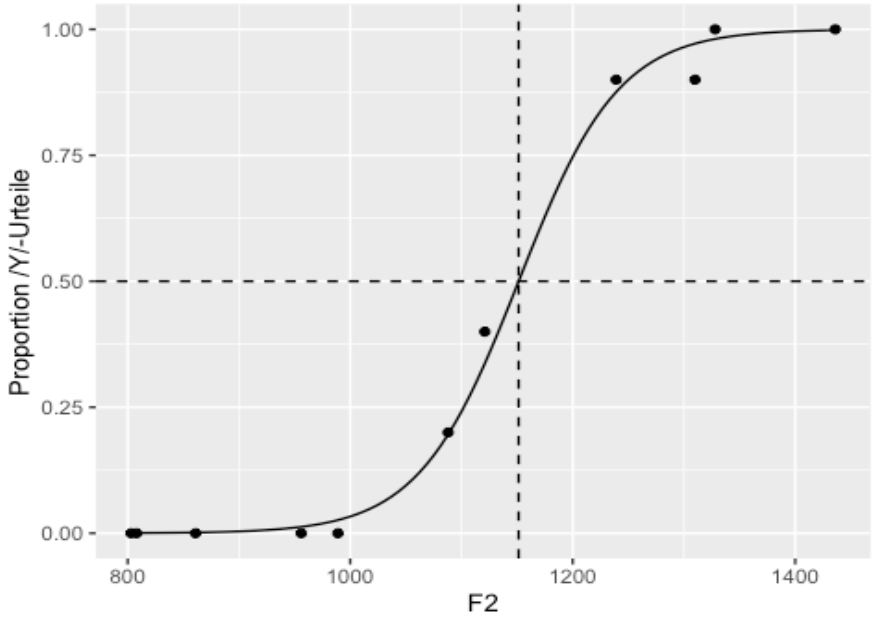
```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: cbind(P, Q)
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                    10     111.59
## F2    1   108.96         9       2.63 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Die Proportion von pUp/pYp-Antworten wurde signifikant von F2 beeinflusst

($\chi^2[1] = 109.0, p < 0.001$).

Der entsprechene Plot-Befehl in ggplot2-Syntax würde lauten:

```r
ggplot(pvp.sum) +
  aes(y = p, x = F2) +
  geom_point() +
  geom_sig(pvp.k,pvp.m,add=TRUE) +
  geom_vline(xintercept = -pvp.k/pvp.m, lty = "dashed") +
  geom_hline(yintercept = .5, lty = "dashed") +
  ylab("Proportion /Y/-Urteile")
```



7. Kategorialer unabhängiger Faktor

Die logistische Regression kann auf eine ähnliche Weise verwendet werden, wenn der *unabhängige Faktor* **kategorial** ist. Der wesentliche Unterschied ist, dass man nicht eine Sigmoidalkurve abzubilden braucht, und kein Umkipppunkt berechnet wird.

```
head(sz)
```

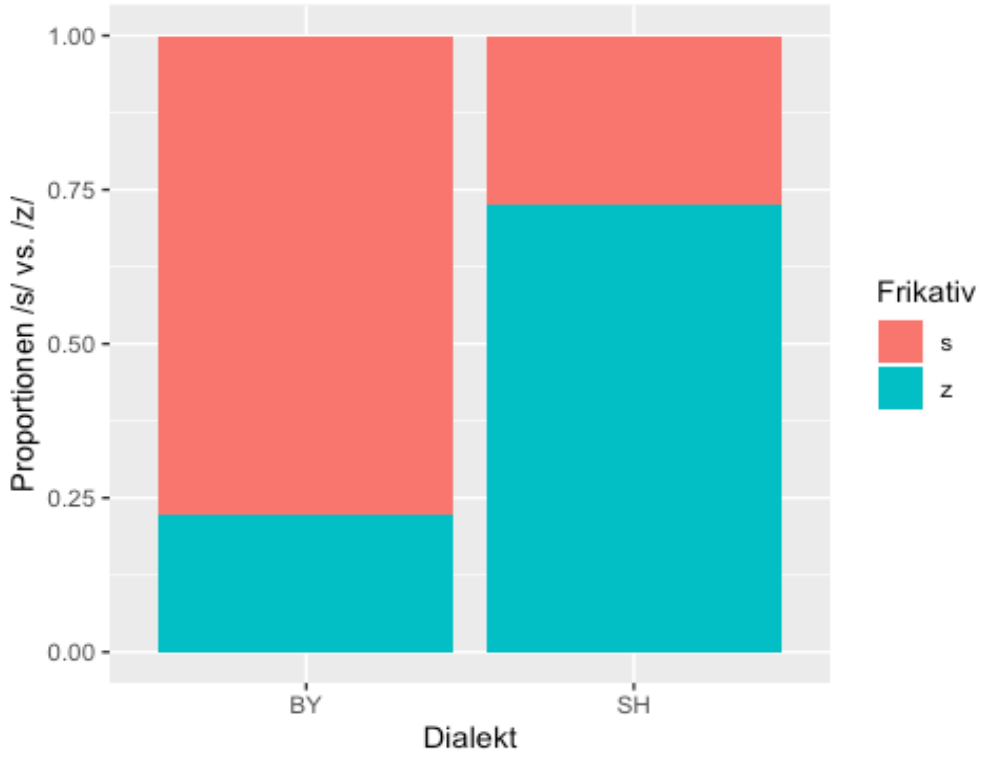
```
##   Frikativ Dialekt Vpn
## 1        z      SH  S1
## 2        z      SH  S2
## 3        z      SH  S3
## 4        z      SH  S4
## 5        s      SH  S5
## 6        s      SH  S6
```

20 Versuchspersonen, davon 9 aus Bayern, 11 aus Schleswig-Holstein, produzierten 'Sonne'.

Der initiale Frikativ wurde entweder als [z] oder [s] wahrgenommen.

- **Wird die Stimmhaftigkeit vom Dialekt beeinflusst?**

```
# Die Abbildung: beide Variablen sind kategorial, daher geom_bar()
ggplot(sz) +
  aes(fill = Frikativ, x = Dialekt) +
  geom_bar(position="fill") +
  ylab("Proportionen /s/ vs. /z/")
```



```r
# Test
sz.glm = glm(Frikativ ~ Dialekt, family=binomial, data = sz)

anova(sz.glm, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Frikativ
##
## Terms added sequentially (first to last)
##
##
##         Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                       19     27.726
## Dialekt  1   5.3002        18     22.426  0.02132 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Die [s]/[z] Verteilung wurde signifikant vom Dialekt beeinflusst

($\chi^2[1] = 5.3, p < 0.05$).