

# Forschungsfragen für binäre AV

Kapitel 9

# Gliederung

1. Teil 1: Grundlagen der logistischen Regression
2. Teil 2: Metrische UV
3. Teil 3: Prioris
4. Teil 4: Binäre UV
5. Teil 5: Modellgüte
6. Hinweise

# Teil 1

## Grundlagen der logistischen Regression

# Lineare Modelle für binäre AV?

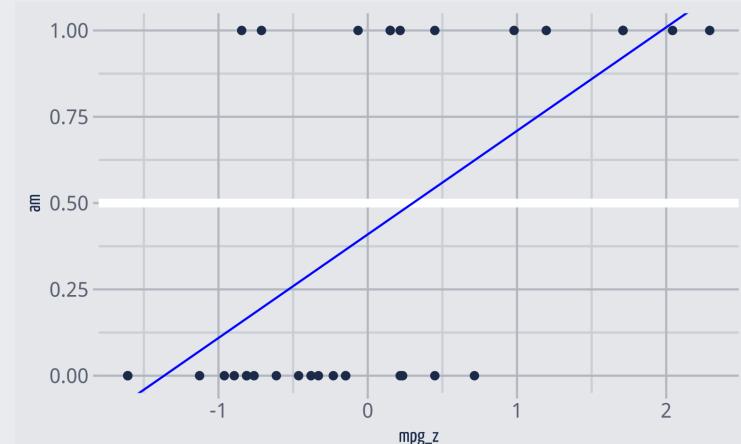
*Forschungsfrage:* Kann man anhand des Spritverbrauchs vorhersagen, ob ein Auto eine Automatik- bzw. ein manuelle Schaltung hat? Anders gesagt: Hängen Spritverbrauch und Getriebeart? (Datensatz mtcars)

```
data(mtcars)
mtcars <-
  mtcars %>%
  mutate(mpg_z = scale(mpg))
```

```
m91 <-
stan_glm(am ~ mpg_z,
         data = mtcars,
         refresh = 0)
coef(m91)
```

```
## (Intercept)      mpg_z
##       0.409      0.300
```

Wir können die Vorhersagen des Modells, d.h.  $\hat{y}_i$ , als *Wahrscheinlichkeit* interpretieren (für am=1).



$Pr(\text{am} = 1 | m91, \text{mpg}_z = 0) = 0.46$   
: Die Wahrscheinlichkeit einer manuelle Schaltung, gegeben einem durchschnittlichen Verbrauch (und dem Modell m91) liegt bei knapp 50%.

# Lineare Modelle running wild

Wie groß ist die Wahrscheinlichkeit für eine manuelle Schaltung ...

- ... bei  $\text{mpg\_z} = -2$ ?

```
(predict(m91, newdata = data.frame(mpg_z = -2)))
```

```
##      1  
## -0.187
```

$Pr(\hat{y}) < 0$  macht keinen Sinn. ⚡

- ... bei  $\text{mpg\_z} = +2$ ?

```
(predict(m91, newdata = data.frame(mpg_z = 2)))
```

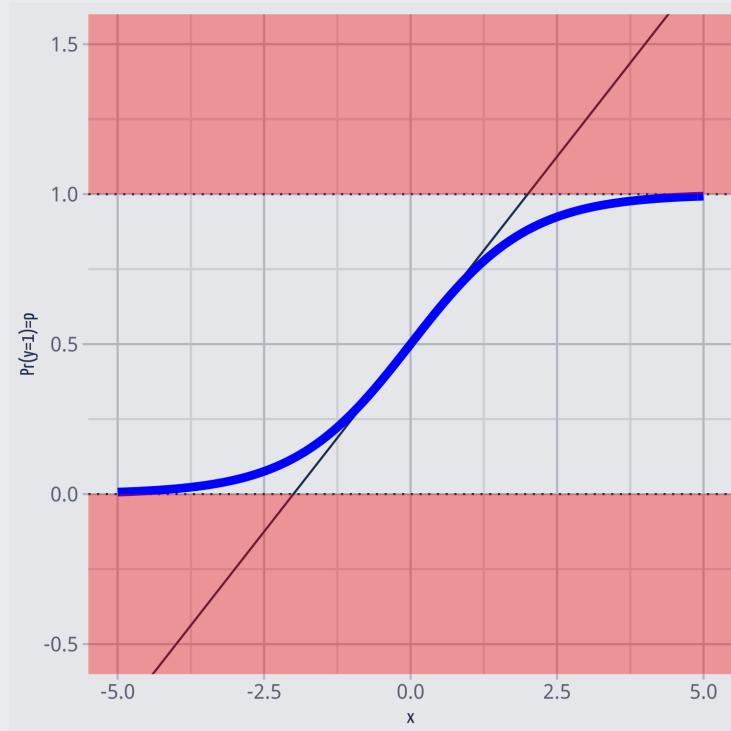
```
##      1  
## 1.01
```

$Pr(\hat{y}) > 1$  macht keinen Sinn. ⚡

Schauen Sie sich mal die Vorhersage an für  $\text{mpg\_z}=5$  😠

# Wir müssen die Regressionsgerade umbiegen

... wenn der vorhergesagte Wert eine Wahrscheinlichkeit,  $p_i$ , ist.



- Die *schwarze* Gerade verlässt den Wertebereich der Wahrscheinlichkeit.
- Die *blaue* Kurve,  $f$ , bleibt im erlaubten Bereich,  $\Pr(y) \in [0, 1]$ .
- Wir müssen also die linke oder die rechte Seite des linearen Modells transformieren:

$$p_i = f(\alpha + \beta \cdot x) \text{ bzw.:}$$

$$f(p) = \alpha + \beta \cdot x$$

- $f$  nennt man eine *Link-Funktion*.

# Verallgemeinerte lineare Modelle zur Rettung

- Für metrische AV mit theoretisch unendlichen Grenzen des Wertebereichs haben wir bisher eine Normalverteilung verwendet:

$$y_i \sim \mathcal{N}(\mu_i, \sigma)$$

- Dann ist die Normalverteilung eine voraussetzungsarme Wahl (maximiert die Entropie).
- Aber wenn die AV *binär* ist bzw. *Häufigkeiten* modelliert, braucht man eine Variable die nur positive Werte zulässt.
- In diesem Fall passt die *Binomialverteilung*, *Bin* oder  $\mathcal{B}$ , gut und ist voraussetzungsarm (maximiert die Entropie):

$$y_i \sim \text{Bin}(n, p_i)$$
$$f(p_i) = \alpha + \beta x$$

- Eine binäre Variablen ist eine Häufigkeitsvariable mit  $\text{Bin}(1, p)$ .
- Diese Verallgemeinerung des linearen Modells bezeichnet man als *generalisiertes lineares Modell* (generalized linear model).

# Der Logit-Link

- Der *Logit-Link*,  $\mathcal{L}$ , logit, Log-Odds oder Logit-Funktion genannt, ordnet einen Parameter, der als Wahrscheinlichkeitsmasse definiert ist (und daher im Bereich von 0 bis 1 liegt), einem linearen Modell zu (das jeden beliebigen reellen Wert annehmen kann):

$$y_i \sim \mathcal{B}(n, p_i)$$
$$\text{logit}(p_i) = \alpha + \beta x_i$$

- Die Logit-Funktion  $\mathcal{L}$  ist definiert als der Logarithmus des Verhältnisses der Wahrscheinlichkeit zu Gegenwahrscheinlichkeit:

$$\mathcal{L} = \log \frac{p_i}{1 - p_i}$$

- Das Verhältnis der Wahrscheinlichkeit zu Gegenwahrscheinlichkeit nennt man auch *Odds*.
- Also:

$$\mathcal{L} = \log \frac{p_i}{1 - p_i} = \alpha + \beta x_i$$

# Inverser Logit

Um nach  $p$  aufzulösen, müssen wir einige Algebra bemühen:

$$\log \frac{p}{1-p} = \alpha + \beta x \quad \text{Exponentieren}$$

$$\frac{p}{1-p} = e^{\alpha+\beta x}$$

$$p_i = e^{\alpha+\beta x_i} (1 - p) \quad \text{Zur Vereinfachung: } x := e^{\alpha+\beta x_i}$$

$$\begin{aligned} p_i &= x(1 - p) \\ &= x - xp \end{aligned}$$

$$p + px = x$$

$$p(1 + x) = x$$

$$p = \frac{x}{1+x}$$

Lösen wir  $x$  wieder auf.

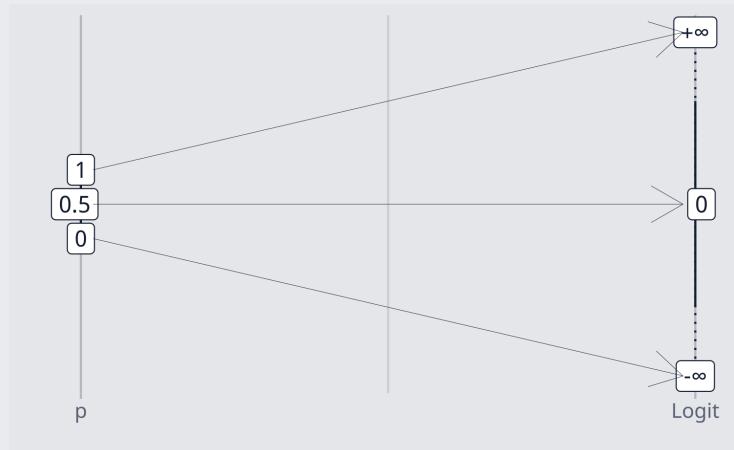
$$p = \frac{e^{\alpha+\beta x_i}}{1 + e^{\alpha+\beta x_i}}$$

Diese Funktion nennt man auch *inverser Logit*,  $\text{logit}^{-1}$ .

# Logit und Inverser Logit

## Logit

$$(0, 1) \rightarrow (-\infty, +\infty)$$

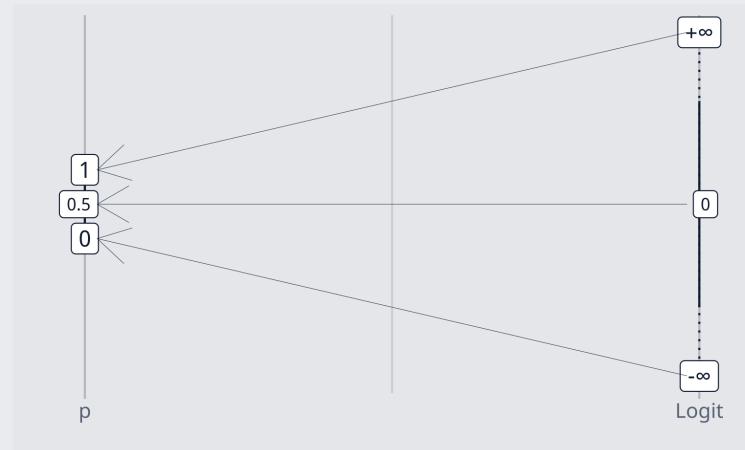


Praktisch, um Wahrscheinlichkeit zu modellieren.

$$p \rightarrow \boxed{\text{logit}} \rightarrow \alpha + \beta x$$

## Inv-Logit

$$(-\infty, +\infty) \rightarrow (0, 1)$$



Praktisch, um in Wahrscheinlichkeiten umzurechnen.

$$p \leftarrow \boxed{\text{inv-logit}} \leftarrow \alpha + \beta x$$

# Logistische Regression

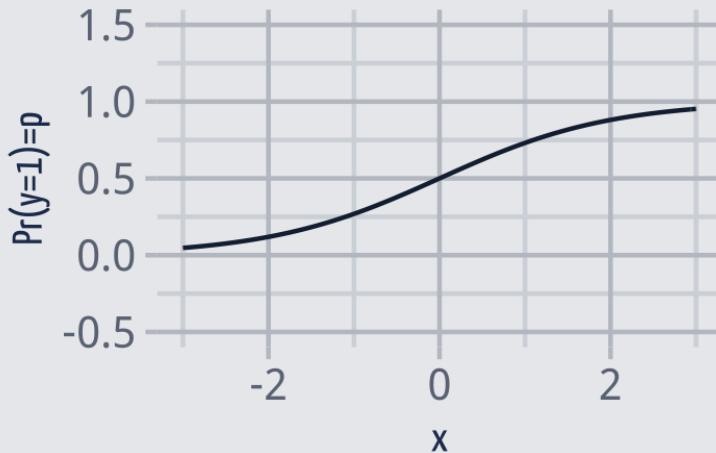
- Eine Regression mit binomial verteilter AV und Logit-Link nennt man *logistische Regression*.
- Man verwendet die logistische Regression um binomial verteilte AV zu modellieren, z.B.
  - Wie hoch ist die Wahrscheinlichkeit, dass ein Kunde das Produkt kauft?
  - Wie hoch ist die Wahrscheinlichkeit, dass ein Mitarbeiter kündigt?
  - Wie hoch ist die Wahrscheinlichkeit, die Klausur zu bestehen?
- Die logistische Regression ist eine normale, lineare Regression für den Logit der AV:

$$y_i \sim \mathcal{B}(n, p_i)$$
$$\text{logit}(p_i) = \alpha + \beta x_i$$

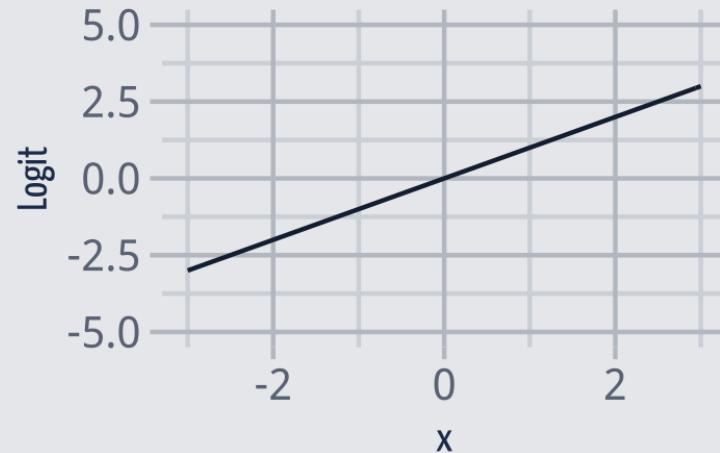
- Da es sich um eine normale, lineare Regression handelt, sind alle bekannten Methoden und Techniken der linearen Regression zulässig.
- Da Logits nicht einfach zu interpretieren sind, rechnet man nach der Berechnung des Modells den Logit häufig in Wahrscheinlichkeiten um.

# Die Koeffizienten sind schwer zu interpretieren

Logistische Regression



Lineare Regression



- In der logistischen Regression gilt *nicht* mehr, dass eine konstante Veränderung in der UV mit einer konstanten Veränderung in der AV einhergeht.
- Stattdessen geht eine konstante Veränderung in der UV mit einer konstanten Veränderung im *Logit* der AV einher.
- Beim logistischen Modell hier gilt, dass in der Nähe von  $x = 0$  die größte Veränderung in  $p$  von statten geht; je weiter weg von  $x = 0$ , desto geringer ist die Veränderung in  $p$ .

# Logits vs. Wahrscheinlichkeiten $p$

```
konvert_logits <-
  tibble(
    logit = c( -10, -3,
              -2, -1, -0.5, -.25,
              0,
              .25, .5, 1, 2,
              3, 10),
    p = invlogit(logit)
  )
```

| logit  | p    |
|--------|------|
| -10.00 | 0.00 |
| -3.00  | 0.05 |
| -2.00  | 0.12 |
| -1.00  | 0.27 |
| -0.50  | 0.38 |
| -0.25  | 0.44 |
| 0.00   | 0.50 |
| 0.25   | 0.56 |
| 0.50   | 0.62 |
| 1.00   | 0.73 |
| 2.00   | 0.88 |
| 3.00   | 0.95 |
| 10.00  | 1.00 |

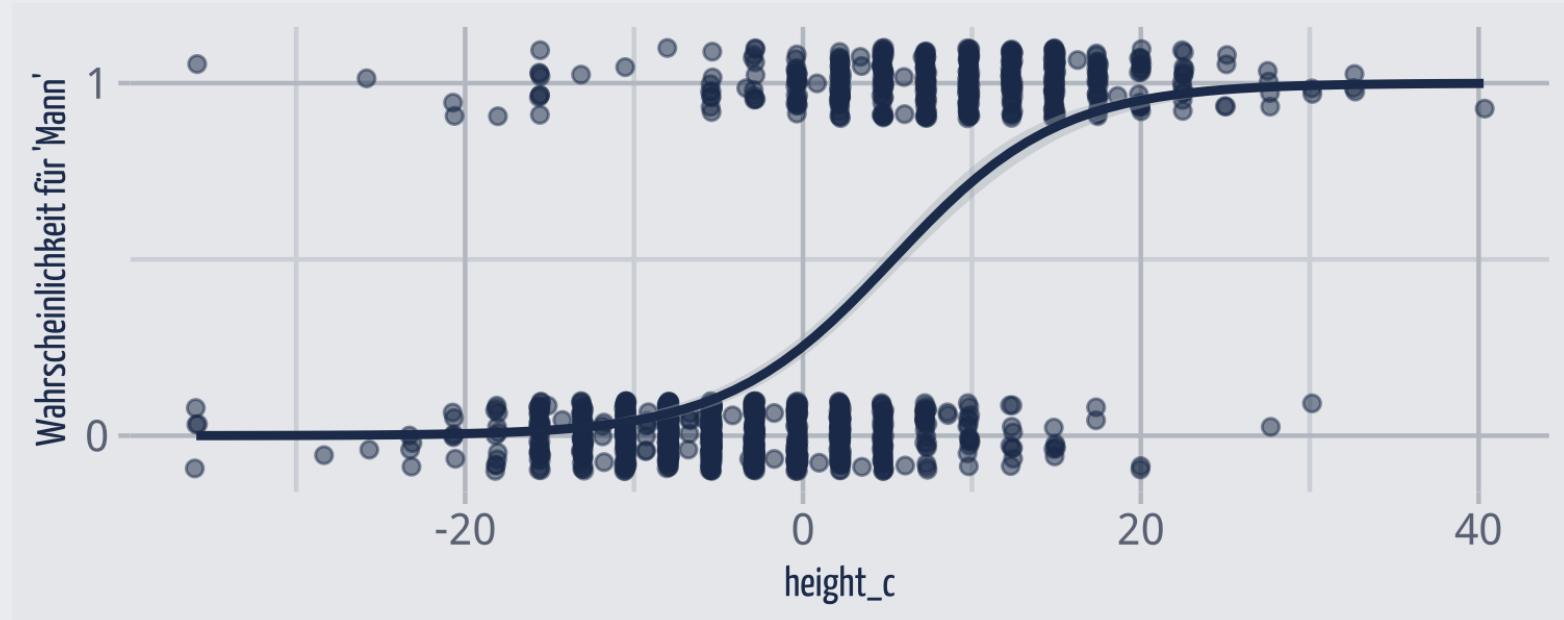
# Teil 2

## Metrische UV

# Geschlecht vorhersagen auf Basis der Körpergröße

Beschreibung des Datensatzes, Datenquelle

```
d <- read_csv(  
  "https://vincentarelbundock.github.io/Rdatasets/csv/openintro/speed_gend...
```



# Zusammenhang von Körpergröße und 'Mann'?

*Forschungsfrage:* Ist der Zusammenhang von Körpergröße und 'Mann' positiv? Gehen also höhere Werte in Körpergröße height einher mit einer höheren Wahrscheinlichkeit, dass es sich um einen Mann handelt?

$$m_i \sim \mathcal{B}(1, p_i)$$

$$\log(p_i) = \alpha + \beta \cdot \text{height}_i$$

$$\alpha \sim \text{klären wir noch}$$

$$\beta \sim \text{klären wir noch}$$

- Die Variable  $m_i$  (ob eine Person ein Mann ist) wird als binomial verteilt angenommen mit der Wahrscheinlichkeit  $p_i$ .
- Die Häufigkeiten, die pro Person möglich sind, begrenzen sich auf 0 und 1 (solche Ereignisse nennt man auch *Bernoulli verteilt*).
- Pro Person wird der Logit von  $p_i$  modelliert als lineare Funktion der Körpergröße dieser Person.
- Entsprechend gilt auch:

$$p_i = \text{logit}^{-1}(\alpha + \beta \cdot \text{height}_i)$$

# Modell m92

```
d2 <- d %>%
  drop_na() %>%
  mutate(male = ifelse(gender == "male", 1, 0),
         height_cm = height * 2.54,
         height_c = height_cm - mean(height_cm))

m92 <- stan_glm(male ~ height_c,
                  family = binomial(link="logit"),
                  data = d2, refresh = 0)

coef(m92)
```

```
## (Intercept)      height_c
##       -1.079        0.202
```

- Die Modellgleichung kann man so schreiben:  
$$Pr(y = 1) = Pr(\text{male}) = \text{logit}^{-1}(-1.08 + 0.2 * \text{height})$$
- Bei einer mittleren Größe ( $\text{height}_c = 0$ ) ist der Logit für 'Mann' -1.08.
- Wenn dieser Wert kleiner ist als Null, ist die Wahrscheinlichkeit kleiner als 50%.
- Für jeden zusätzlichen Zentimeter Größe steigt die Wahrscheinlichkeit, dass wir 'Mann' vorhersagen um ca. 0.2 Logits.
- Der Zuwachs an Wahrscheinlichkeit ist *nicht* konstant pro zusätzlichen Logit.

# Umrechnen von Logits in Wahrscheinlichkeit

- Logits sind schwer zu interpretieren, rechnen wir in Wahrscheinlichkeit,  $p$  um.
- Der Achsenabschnitt im zentrierten (oder z-standardisierten) Modell gibt, den Y-Wert an für eine Beobachtung mit mittleren X-Wert:

```
invlogit <- plogis # Funktion, um Inv-Logit von R berechnen zu lassen  
invlogit(coef(m92)[1])
```

```
## (Intercept)  
##          0.254
```

- Bei einer Person mittleren Größe sagt unser Modell mit einer Wahrscheinlichkeit von ca. 0.25 vorher, dass es ein Mann ist.

```
invlogit(coef(m92)[1] + coef(m92)[2]*10)
```

```
## (Intercept)  
##          0.72
```

- Bei einer Person, die 10cm größer ist als der Mittelwert, geht unser Modell von einer Wahrscheinlichkeit von 0.72 aus.

# Post befragen: 95%-PI

```
posterior_interval(m92, prob = .95)
```

```
##           2.5%   97.5%
## (Intercept) -1.254 -0.918
## height_c     0.181  0.225
```

```
invlogit(c(-1.22, -0.95))
```

```
## [1] 0.228 0.279
```

Die Wahrscheinlichkeit, dass eine mittelgroße Person ein Mann ist, liegt zwischen ca. 23% und 28%, laut dem Modell.

Die Intervallgrenzen des Regressionsgewichts  $\beta$  sind schwieriger zu interpretieren, da die Veränderung in den Wahrscheinlichkeiten nicht konstant sind: Je größere eine Person, desto geringer der Zuwachs in Wahrscheinlichkeit (ein Mann zu sein) pro zusätzliche Logit-Einheit.

Zur Interpretation von  $\beta$  ist es meist sinnvoller, die Vorhersagen des Modells (in Wahrscheinlichkeitsmaßen) zu betrachten, die PPV also.

# Ist der Zusammenhang von Größe und 'Mann' positiv?

| Forschungsfrage: Ist der Zusammenhang von Körpergröße und 'Mann' positiv?

Zählen wir den Anteil der Stichproben, die ein positives  $\beta$  findet:

```
## # A tibble: 1 × 2
##   `height_c > 0`      n
##   <lgl>           <int>
## 1 TRUE            4000
```

Eigentlich hatten wir das schon hier gesehen:

```
##          2.5% 97.5%
## height_c 0.181 0.225
```

| Der Zusammenhang von Körpergröße und 'Mann' ist sehr sicher positiv, laut dem Modell: Je größer eine Person, desto höher die Wahrscheinlichkeit, dass sie ein Mann ist.

# PPV befragen 1

|  $Pr(\text{Mann} \mid \text{height}_c = 10, m92)$ ?

Ausgabe in Logit:

```
posterior_predict(m92, newdata = tibble(height_c = 10)) %>%  
  mean()
```

```
## [1] 0.725
```

Mit `invlogit()` kann man sich den Logit-Wert in Wahrscheinlichkeit umrechnen lassen.

Ausgabe direkt in Wahrscheinlichkeit:

```
m92_ppv_10 <- posterior_predict(m92, newdata = tibble(height_c = 10))
```

```
m92_ppv_10 %>%  
  mean()
```

```
## [1] 0.718
```

`posterior_predict()` zeigt uns die Vorhersagen aus der PPV für  $\alpha + \beta \cdot 10$ .

# PPV-Vorhersagen etwas genauer betrachtet

Betrachten wir die PPV für Personen der Größe -40, -30, ..., +40 cm größer als der Durchschnitt:

```
height_vec <- seq(-40, 40, by = 10)
heights_df <- tibble(height_c = height_vec)
m92_ppv <- posterior_predict(m92, newdata = heights_df) %>%
  as_tibble()
```

Die ersten paar Zeilen von m92\_ppv:

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

Die 9 Spalten entsprechen den 9 Werten von height\_vec, (-40, -30, ..., +40).

# PPV befragen 2

Wie groß ist die Wahrscheinlichkeit, dass eine Person, die -40, -30, ..., 40 cm größer ist als der Durchschnitt, als Mann klassifiziert wird?

```
m92_ppv %>%
  summarise(minus40 = mean(`1`),    # -40cm
            minus30 = mean(`2`),    # -30cm
            minus20 = mean(`3`),    # -20cm
            minus10 = mean(`4`),    # - 10 cm
            null_durchschnitt = mean(`5`))
# und so weiter
```

```
## # A tibble: 1 × 5
##   minus40 minus30 minus20
##     <dbl>   <dbl>   <dbl>
## 1       0  0.00025   0.004
## # ... with 2 more variables:
## #   minus10 <dbl>,
## #   null_durchschnitt <dbl>
```

# PPV befragen 3

Wie groß ist die Wahrscheinlichkeit, dass eine überdurchschnittlich große Person als Mann klassifiziert wird?

- Dazu bilden wir den Mittelwert der Spalten 5-9.
- Für diesen Zweck muss die Tabelle so umgeformt werden, dass die Wahrscheinlichkeiten aller 9 Gruppen in einer Spalte stehen (`pivot_wider()`).
- Dann filtern wir die Gruppen 5-9.

```
m92_ppv %>%
  pivot_longer(everything()) %>%
  filter(name == c("5", "6", "7", "8", "9")) %>%
  summarise(p_mann = mean(value))
```

```
## # A tibble: 1 × 1
##   p_mann
##     <dbl>
## 1 0.783
```

# Tabellen umformen von lang nach breit und zurück

- `pivot_longer()` von breit nach lang (gather)
- `pivot_wider()` von lang nach breit (spread)

wide

|   | id | x | y | z |
|---|----|---|---|---|
| 1 | a  | c | e |   |
| 2 | b  | d | f |   |

Garrick Aden-Buie [Quelle](#); vgl. auch dieses Tutorial

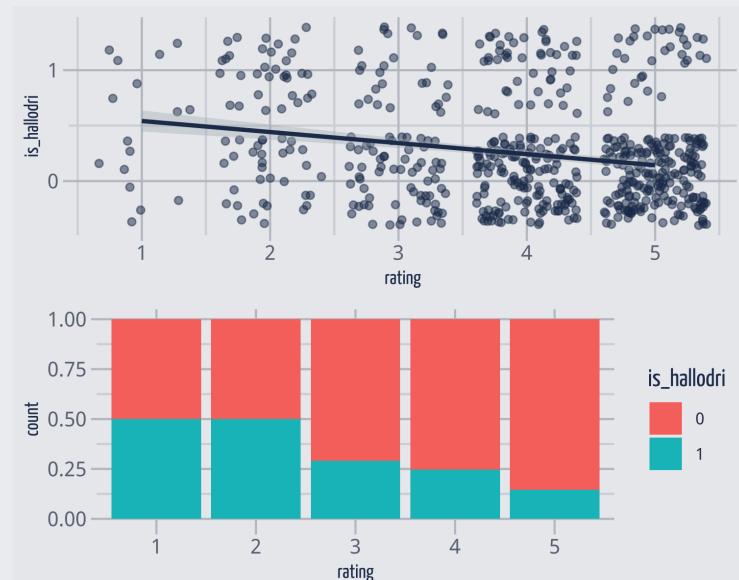
# Datensatz zu außerehelichen Affären

Beschreibung des Datensatzes, Datenquelle

```
d_hallodri <- read_csv(  
  "https://vincentarelbundock.github.io/Rdatasets/csv/AER/Affairs.csv")
```

EDA:

```
d_hallodri <-  
  d_hallodri %>%  
  mutate(is_hallodri =  
    ifelse(affairs > 0, 1, 0),  
    rating_z =  
      scale(rating))  
  
d_hallodri %>%  
  select(is_hallodri, rating_z) %>%  
  group_by(rating_z) %>%  
  get_summary_stats(  
    type = "mean_sd")
```



# Halldri-Modell

| Kovariiert die Wahrscheinlichkeit  $p_i$  für *Halldri*  $h_i$  (negativ) mit der (z-standardisierten) Ehezufriedenheit  $r_i$ ?

$$h_i \sim \mathcal{B}(1, p_i)$$

$$\log(p_i) = \alpha + \beta \cdot r_i$$

$$\alpha \sim \mathcal{N}(0, 2.5)$$

$$\beta \sim \mathcal{N}(0, 2.5)$$

```
m_hallodri1 <-  
  stan_glm(is_hallodri ~ rating_z, data = d_hallodri, refresh = 0,  
           family = binomial(link = "logit"))
```

💣 Der AV liegt eine metrische Variable zugrunde (`affairs`). Zumeist ist es sinnvoller, die informationsreichere metrische Variable zu modellieren. Die Dichotomisierung zu einer binären Variablen verschenkt viel Information. Hier zu didaktischen Zwecken.

# Halldri-Modell: Ergebnisse

```
posterior_interval(m_halldri1, pars = "(Intercept)") %>% invlogit()
```

```
##           5%      95%
## (Intercept) 0.208 0.266
```

```
posterior_interval(m_halldri1, pars = "rating_z")
```

```
##           5%      95%
## rating_z -0.718 -0.408
```

- Bei mittlerer Ehezufriedenheit liegt die Wahrscheinlichkeit eines Seitensprungs bei ca. 21% bis 27% (95%-PI).
- Je höher die Ehezufriedenheit, desto geringer die Wahrscheinlichkeit für einen Seitensprung.
- Laut dem Modell ist  $\beta$  mit hoher Wahrscheinlichkeit negativ.

# Vorhersagen bei den Hallordis

Wie hoch ist die Wahrscheinlichkeit für einen Seitensprung, wenn  
rating\_z = -3?

Punktschätzer:

```
predict(m_hallodri1,  
        newdata = tibble(rating_z=-3))
```

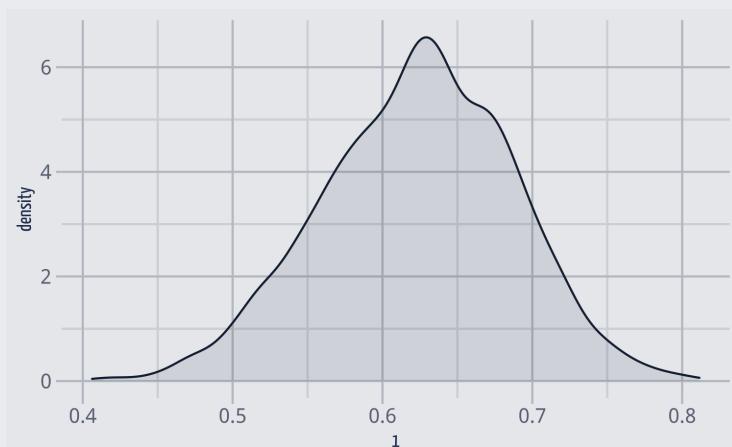
```
##      1  
## 0.514
```

95%-PI aus der Post-Verteilung (nicht  
PPV) als Wahrscheinlichkeit:

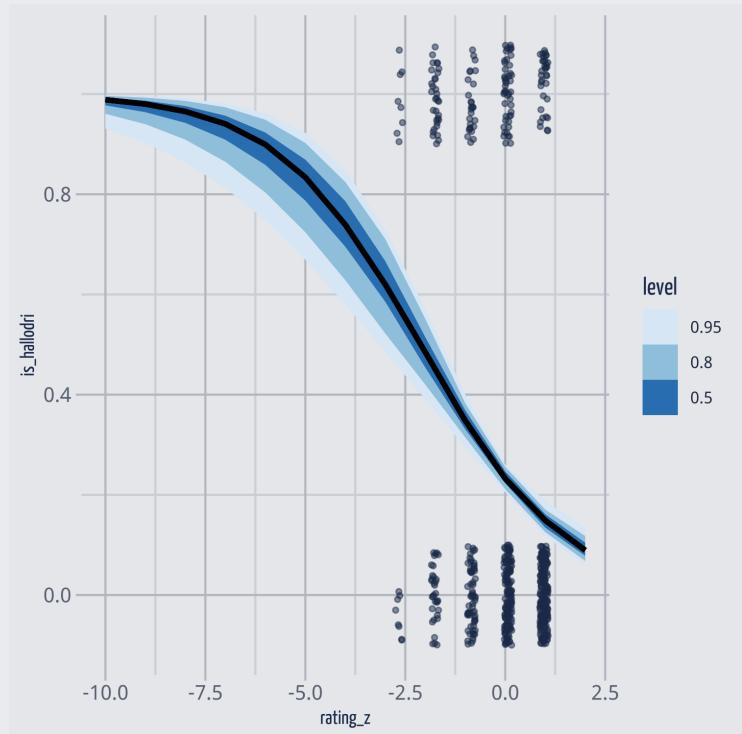
```
posterior_epred(m_hallodri1,  
                 newdata = tibble(rating_z = -3)  
                 quantile(prob = c(0.025, .975)))
```

```
## 2.5% 97.5%  
## 0.496 0.743
```

```
posterior_epred(m_hallodri1,  
                newdata = tibble(rating_z = -3)  
                as_tibble() %>%  
                ggplot(aes(x = `1`)) +  
                geom_density()
```



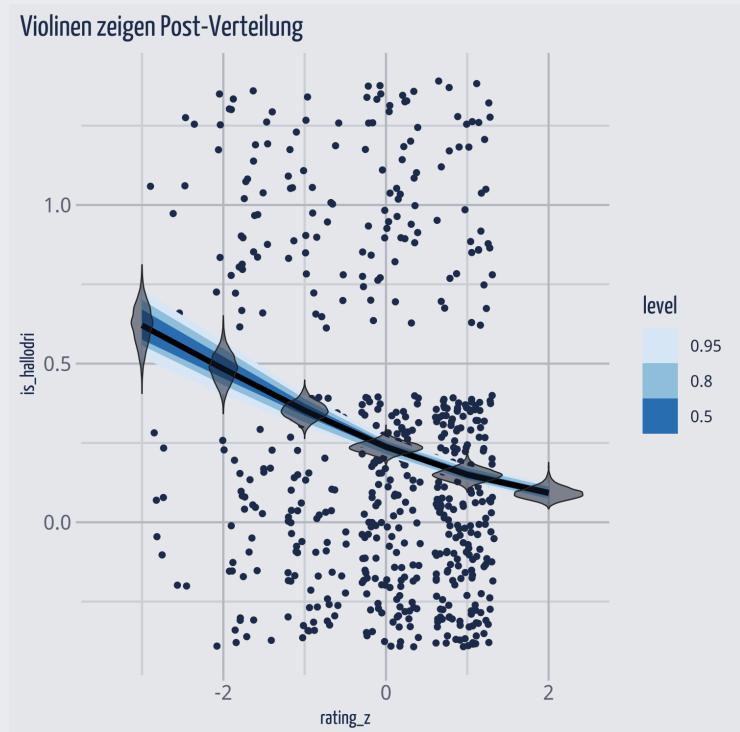
# Visualisierung der Modellfunktion



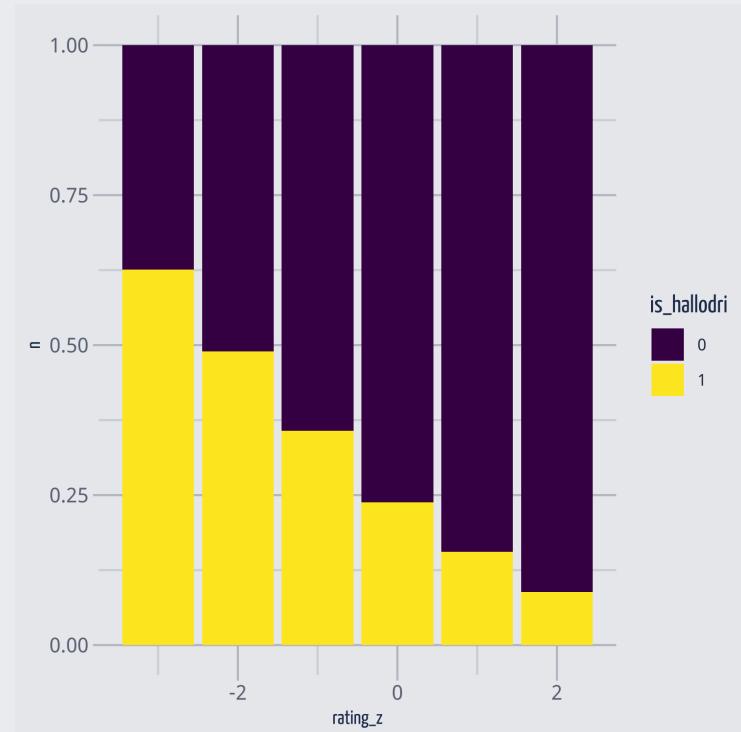
- Um den charakteristischen, s-förmigen Verlauf der logistischen Funktion zu zeigen, ist hier der Wertebereich des Prädiktors übermäßig nach links erweitert.
- Extreme Wahrscheinlichkeiten sind mit weniger Unsicherheit verbunden als mittlere Wahrscheinlichkeiten.

# Vorhersagen mit Ungewissheit visualisiert

Vorhersagen aus der Post-Verteilung  
für jede Stufe von rating\_z.



Vorhersagen aus der PPV für jede Stufe von rating\_z.



# Teil 3

## Prioris

# Priors bei den Hallodri

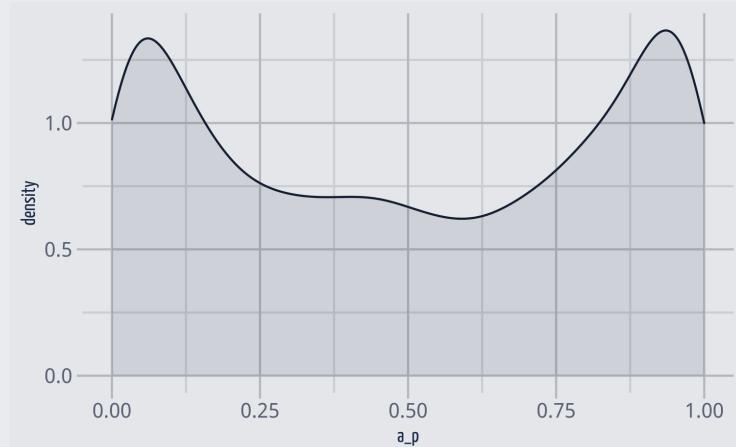
```
## Priors for model 'm_hallodri1'  
## -----  
## Intercept (after predictors centered)  
## ~ normal(location = 0, scale = 2.5)  
##  
## Coefficients  
## Specified prior:  
## ~ normal(location = 0, scale = 2.5)  
## Adjusted prior:  
## ~ normal(location = 0, scale = 2.5)  
## -----  
## See help('prior_summary.stanreg') for more details
```

- Unser Prädiktor ist z-standardisiert.
- Ein Prior für  $\sigma$  gibt es bei Binomialmodellen nicht.

# Priori-Analyse

```
m_hallodri2 <-
  stan_glm(is_hallodri ~ rating_z, data = d_hallodri, refresh = 0,
            prior_PD = TRUE,
            family = binomial(link = "logit"))
```

```
m_hallodri2 %>%
  as_tibble() %>%
  rename(a = `"(Intercept)"`) %>%
  mutate(a_p = invlogit(a)) %>%
  ggplot(aes(x = a_p)) +
  geom_density()
```

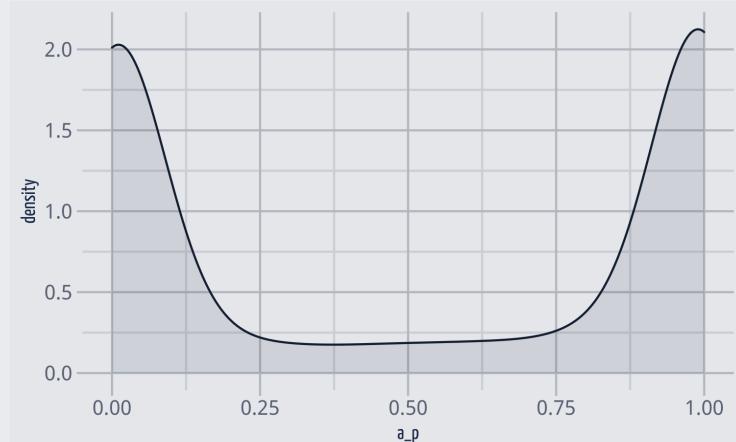


Unser Golem geht apriori von extremen Wahrscheinlichkeiten des Hallodritums aus: Entweder du bist einer ❤ oder du bist es nicht 😊.

# Uninformativer Prior

```
m_hallodri3 <-  
  stan_glm(is_hallodri ~ rating_z, data = d_hallodri, refresh = 0,  
           prior_PD = TRUE,  
           prior_intercept = normal(0, 10),  
           family = binomial(link = "logit"))
```

```
m_hallodri3 %>%  
  as_tibble() %>%  
  rename(a = `Intercept`) %>%  
  mutate(a_p = invlogit(a)) %>%  
  ggplot(aes(x = a_p)) +  
  geom_density()
```



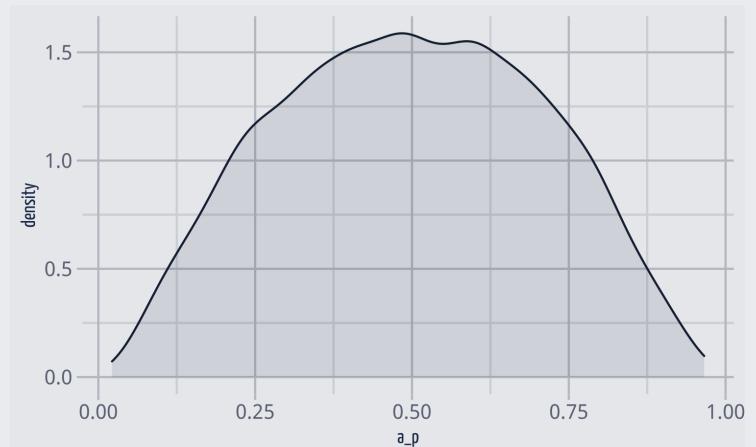
💡 Je flacher der Priori, desto extremer die Priori-Verteilung bei einem Binomial-Modell.

Wir sollten gemäßigte Prioris bevorzugen.

# Gemäßiger Prior

```
m_hallodri4 <-  
  stan_glm(is_hallodri ~ rating_z, data = d_hallodri, refresh = 0,  
           prior_PD = TRUE,  
           prior_intercept = normal(0, 1),  
           family = binomial(link = "logit"))
```

```
m_hallodri4 %>%  
  as_tibble() %>%  
  rename(a = `Intercept`) %>%  
  mutate(a_p = invlogit(a)) %>%  
  ggplot(aes(x = a_p)) +  
  geom_density()
```



👍 Das sieht gut aus!

## Teil 4

### Binäre UV

# Logistische Regression nur mit Achsenabschnitt

- *Lineare* ("normale") Regression nur mit Achsenabschnitt ist identisch zur Schätzung eines *Mittelwerts*.
- Lineare Regression mit einer binären UV ist identisch zur Schätzung eines Unterschieds im Mittelwert zwischen zwei Gruppen.
- Analog dazu entspricht eine *logistische* Regression nur mit Achsenabschnitt dem Schätzen eines *Anteils*.

50 Personen werden auf Ignoranzitis getestet (eine schlimme Krankheit), davon 10 Personen positiv. Was ist der Anteil in der Population?

Vermutlich so etwa 20% (wenn die Stichprobe gut ist), plus minus ein bisschen.

Als logistische Regression:

```
y <- rep(c(0,1), c(40, 10)) # 40 mal 0, 10 mal 1
ignor_df <- tibble(y)
ignor_m <- stan_glm(y ~ 1, data = ignor_df, refresh = 0,
                      family = binomial(link = "logit"))
```

# Ergebnisse des Ignoranzitis-Modells

## Logit-Skala

Der Achsenabschnitt ist der Punktschätzer zum Anteil der Ignoranzitis-Positiven (in der Population):

```
coef(ignor_m)
```

```
## (Intercept)
## -1.38
```

Und hier die Ungenauigkeit (Standardfehler) für den Punktschätzer:

```
se(ignor_m)
```

```
## (Intercept)
## 0.355
```

## Wahrscheinlichkeits-Skala

Der Anteil in der Pr-Skala:

```
coef(ignor_m) %>% invlogit()
```

```
## (Intercept)
## 0.201
```

95%-PI des Punktschätzers:

```
UG <- (coef(ignor_m) -
  2*se(ignor_m)) %>% invlogit()
OG <- (coef(ignor_m) +
  2*se(ignor_m)) %>% invlogit()
```

UG: 0.11; OG: 0.34.

# Eine binäre UV

- Die logistische Regression mit einer UV ist äquivalent zum Vergleich zweier Anteile.

Zur Bekämpfung der Ignoranzitis wird das Medikament Lisliberin verabreicht. In der Experimentalgruppe (mit Lisliberin) liegt der Anteil von Ignoranzitis danach noch bei 5 von 50. In der Kontrollgruppe liegt der Anteil bei 20 von 60.

```
gruppe <- rep(c(0, 1), c(50, 60))
ignor <- rep(c(0, 1, 0, 1), c(45, 5, 40, 20))
lisliber_df <- tibble(gruppe, ignor)

lisliber_m <- stan_glm(ignor ~ gruppe, data = lisliber_df, refresh = 0,
                      family = binomial(link = "logit"))
coef(lisliber_m)
se(lisliber_m)

## (Intercept)      gruppe
##       -2.21      1.53
## (Intercept)      gruppe
##       0.487      0.558
```

# Ergebnisse zum Lisliber-Modell

Vorhersagen pro Gruppe in der Pr-Skala bekommt man auch mit `posterior_epred()`:

```
neu <- tibble(gruppe = c(0,1))
lisliber_post <- posterior_epred(lisliber_m, newdata = neu)
lisliber_post <- lisliber_post %>%
  as_tibble() %>%
  mutate(diff_gruppen = `2` - `1`)
```

Vorhersagen auf Basis der Post-Verteilung (ersten paar Stichproben):

| 1      | 2     | diff_gruppen |
|--------|-------|--------------|
| 0.1227 | 0.402 | 0.279        |
| 0.1851 | 0.348 | 0.163        |
| 0.0715 | 0.305 | 0.234        |

```
lisliber_post %>%
  summarise(
    diff_mw = mean(diff_gruppen),
    diff_sd = sd(diff_gruppen))
```

```
## # A tibble: 1 × 2
##       diff_mw   diff_sd
##         <dbl>     <dbl>
## 1      0.232    0.0738
```

# Teil 5

## Modellgüte

# Fehlerrate

Die *Fehlerrate* ist definiert als der Anteil für den eine der beiden folgenden Fehler zutrifft:

- Fehler 1:  $y_i = 1$  wenn  $Pr(\hat{y}_i) < 0.5$
- Fehler 2:  $y_i = 0$  wenn  $Pr(\hat{y}_i) > 0.5$
- Die Fehlerrate sollte immer kleiner sein als  $1/2$ : Sonst könnten wir alle  $\beta$ s auf 0 setzen und würden eine bessere Fehlerrate (besseren "Fit") bekommen.
- Man kann die Fehlerrate seines Modells mit dem *Nullmodell* vergleichen, das für alle  $y_i$  die gleiche Wahrscheinlichkeit annimmt.
- Das Nullmodell ist die Regression ohne Prädiktoren (nur mit Achsenabschnitt).
- Die Fehlerrate des Nullmodells entspricht dem Anteil  $p$  von  $y_i = 1$  oder  $1 - p$  (je nachdem, welcher Wert von beiden kleiner ist).

(Gelman, Hill, and Vehtari, 2021, S. 255)

# Fehlerrate berechnen für das Hallodri-Modell

```
d_hallodri <-  
  d_hallodri %>%  
  mutate(hallodri_pred = predict(m_hallodri1))  
  
d_hallodri <-  
  d_hallodri %>%  
  mutate(wrong_pred =  
    (hallodri_pred>0.5 & is_hallodri==0) | # Fehler 1  
    (hallodri_pred<0.5 & is_hallodri==1)) # Fehler 2  
  
error_rate <-  
  d_hallodri %>%  
  summarise(error_rate = mean(wrong_pred))  
  
error_rate  
  
## # A tibble: 1 × 1  
##   error_rate  
##       <dbl>  
## 1     0.250
```

# Hinweise

# Zu diesem Skript

- Dieses Skript bezieht sich auf folgende **Lehrbücher**:
  - Regression and other stories (Kap. 13); Statistical Rethinking (Kap. 10.2)
- Dieses Skript wurde erstellt am 2021-12-14 15:13:25.
- Lizenz: [MIT-Lizenz](#)
- Autor: Sebastian Sauer.
- Um diese HTML-Folien korrekt darzustellen, ist eine Internet-Verbindung nötig.
- Mit der Taste ? bekommt man eine Hilfe über Shortcuts.
- Wenn Sie die Endung .html in der URL mit .pdf ersetzen, bekommen Sie die PDF-Version der Datei.
- Alternativ können Sie im Browser Chrome die Folien als PDF drucken (klicken Sie auf den entsprechenden Menüpunkt).
- Den Quellcode der Skripte finden Sie [hier](#).
- Eine PDF-Version kann erzeugt werden, indem man im Chrome-Browser die Webseite druckt (Drucken als PDF).



Homepage

# Literatur

Diese R-Pakete wurden verwendet.

Gelman, A., J. Hill, and A. Vehtari (2021). *Regression and other stories*. Analytical methods for social research. Cambridge University Press.

McElreath, R. (2020). *Statistical rethinking: a Bayesian course with examples in R and Stan*. 2nd ed. CRC texts in statistical science. Taylor and Francis, CRC Press.

