Lineare Modelle

QM2, Thema 5

AWM, HS Ansbach

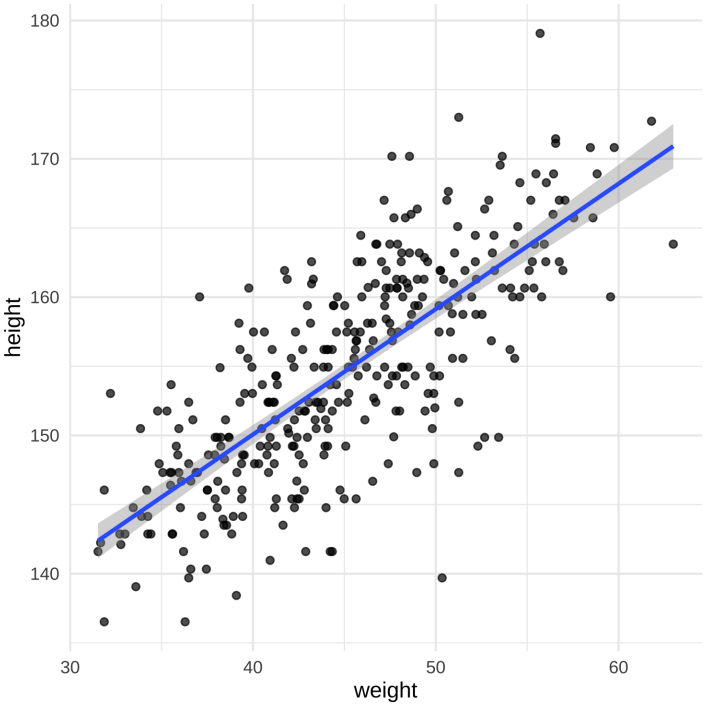
Gliederung

1. Teil 1: Die Post-Verteilung der Regression berechnen
2. Teil 2: Die Post-Verteilung befragen
3. Hinweise
4. Literatur

Post-Verteilung der Regression

Einfache Regression

- Die (einfache) Regression prüft, inwieweit zwei Variablen, Y und X linear zusammenhängen.
- Je mehr sie zusammenhängen, desto besser kann man X nutzen, um Y vorherzusagen (und umgekehrt).
- Hängen X und Y zusammen, heißt das nicht (unbedingt), dass es einen *kausalen* Zusammenhang zwische X und Y gibt.
- Linear bedeutet, der Zusammenhang ist additiv und konstant: wenn X um eine Einheit steigt, steigt Y immer um b Einheiten.



Statistiken zum !Kung-Datensatz

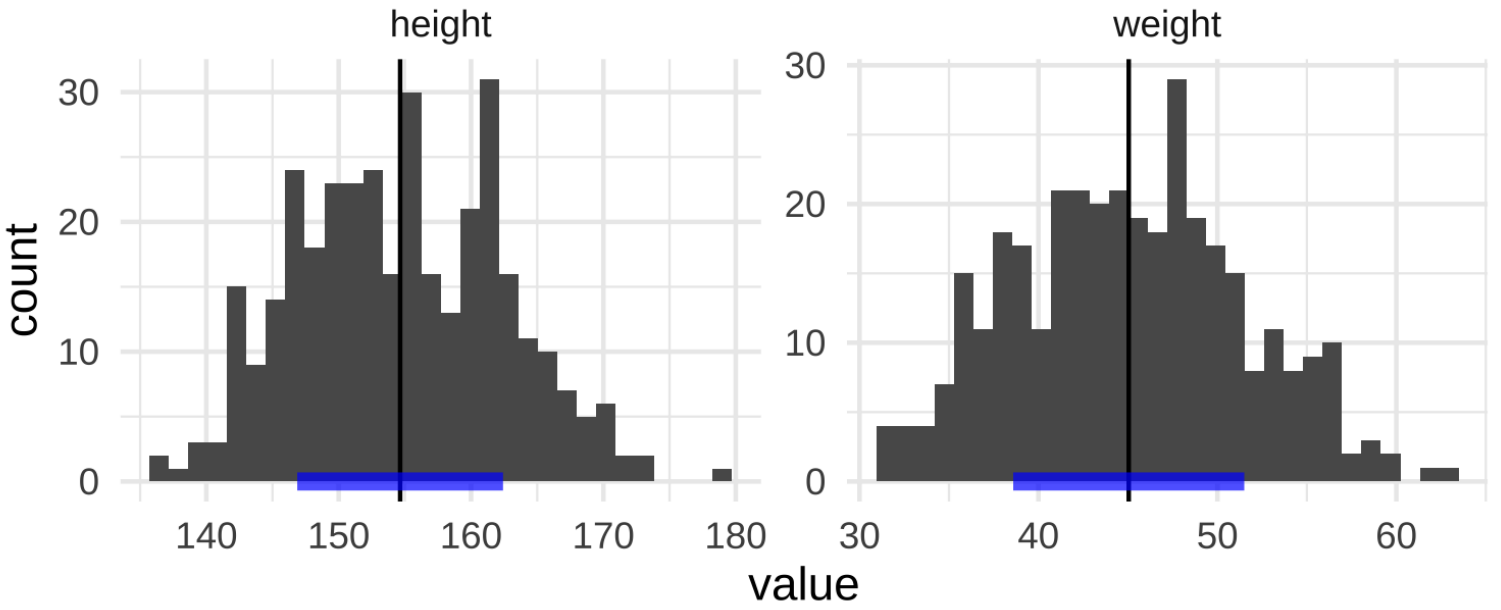
Datenquelle

```
library(tidyverse)
library(rstatix)
Kung_path <- "https://tinyurl.com/jr7ckxxj"  # Datenquelle s.o.

d <- read_dsv(Kung_path)

d2 <-
  d %>%
  filter(age > 18)

get_summary_stats(d2)
```

variable	n	min	max	median	q1	q3	iqr	mad	mean	sd	se	ci
age	346.0	19.0	88.0	40.0	29.0	51.0	22.0	16.3	41.5	15.8	0.8	1.7
height	346.0	136.5	179.1	154.3	148.6	160.7	12.1	8.5	154.6	7.8	0.4	0.8
male	346.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	0.5	0.5	0.0	0.1
weight	346.0	31.5	63.0	45.0	40.3	49.4	9.0	6.7	45.0	6.5	0.3	0.7

Das mittlere Körpergewicht (`weight`) liegt bei ca. 45kg (sd 7 kg).

Visualisierung von `weight` und `height`



Vertikale Linie: Mittelwert
horizontale Linie: Std. Abweichung

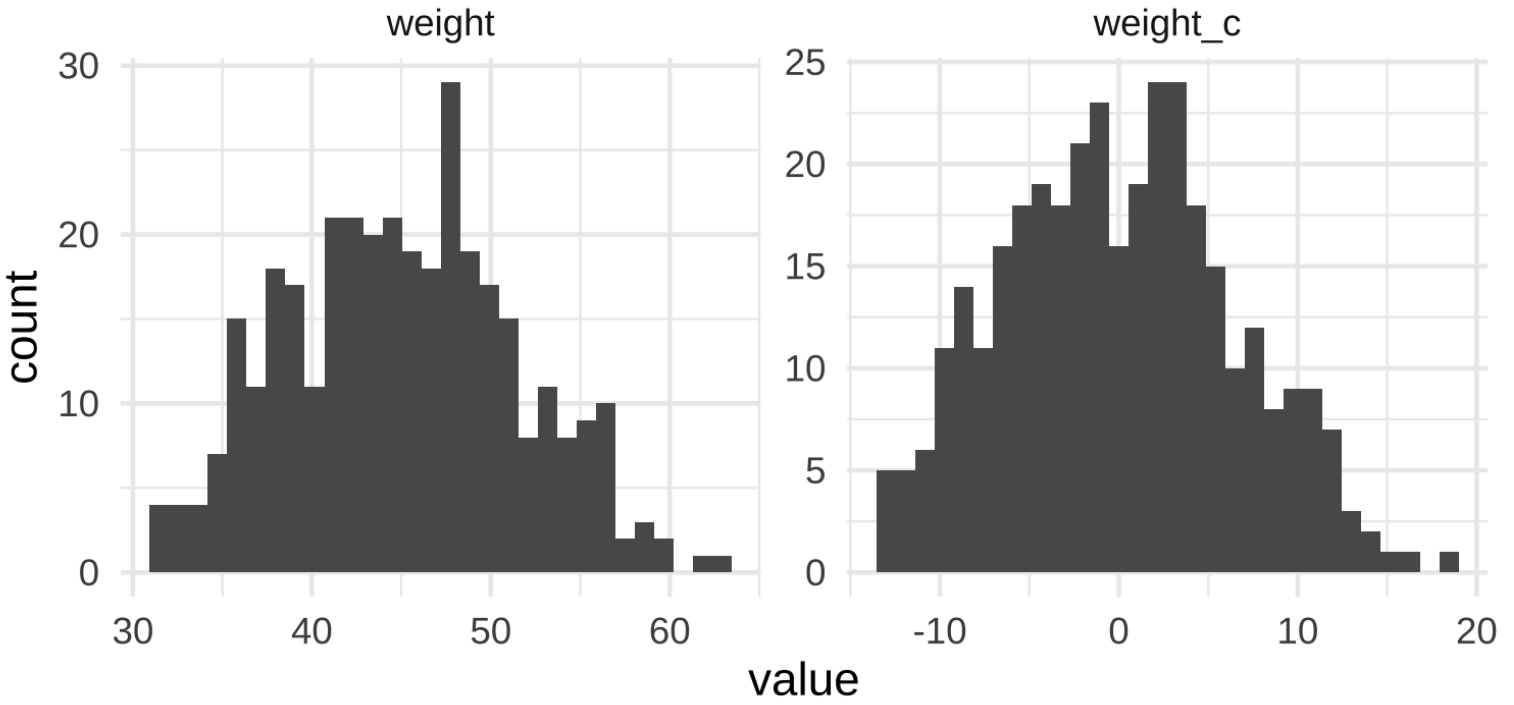
Prädiktor zentrieren 1/2

- Zieht man von jedem Gewichtswert den Mittelwert ab, so bekommt man die Abweichung des Gewichts vom Mittelwert (Präditkor "zentrieren").
- Wenn man den Prädiktor (`weight`) zentriert hat, ist der Achsenabschnitt, α, einfacher zu verstehen.
- In einem Modell mit zentriertem Prädiktor (`weight`) gibt der Achsenabschnitt die Größe einer Person mit durchschnittlichem Gewicht an.
- Würde man `weight` nicht zentrieren, gibt der Achsenabschnitt die Größe einer Person mit `weight=0` an, was nicht wirklich sinnvoll zu interpretieren ist.

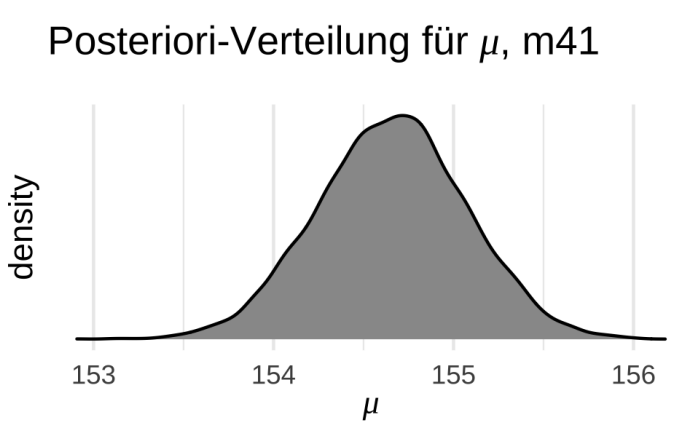
Prädiktor zentrieren 2/2

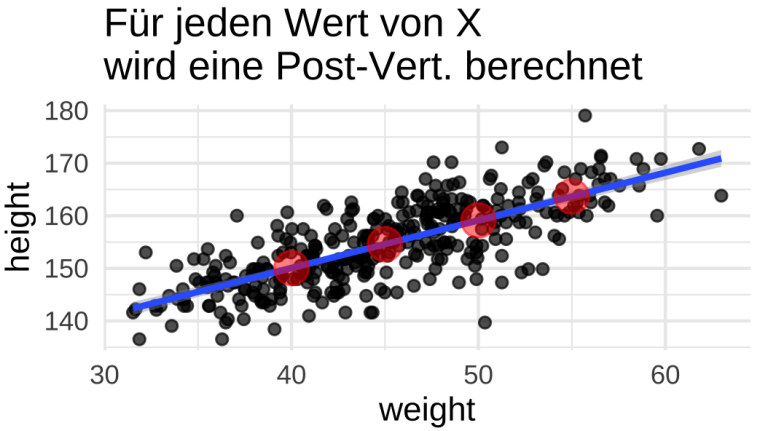
```
d2 <-
  d2 %>%
  mutate(
    weight_c = weight -
      mean(weight))
```

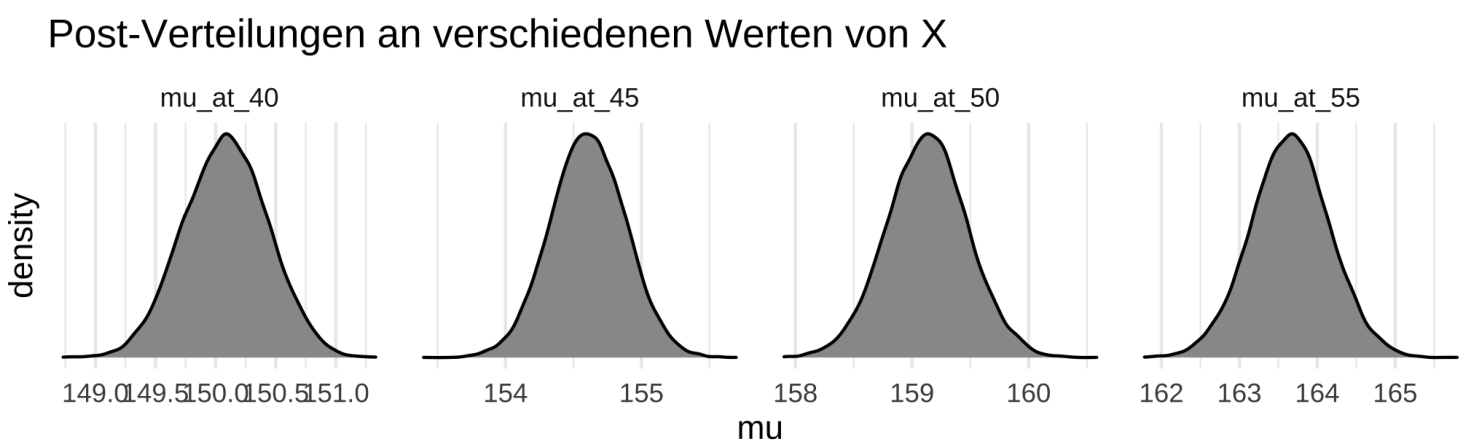
height	weight	age	male	weight_c
152	48	63	1	3
140	36	63	0	−9
137	32	65	0	−13



Für jede Ausprägung des Prädiktors brauchen wir eine Post-Verteilung







Modelldefinition von m43

- Für jede Ausprägung von `weight` wird eine Post-Verteilung für `height` berechnet.
- Der Mittelwert μ für jede Post-Verteilung ergibt sich aus dem linearen Modell (der Regression).
- Die Post-Verteilung berechnet sich auf Basis der Priori-Werte und des Likelihood (Bayes-Formel).
- Wir brauchen Priori-Werte für die Steigung β und den Achsenabschnitt α der Regressionsgeraden.
- Außerdem brauchen wir einen Priori-Wert, der die Streuung σ der Größe (`height`) angibt.
- Der Likelihood gibt an, wie wahrscheinlich ein Wert `height` ist, gegeben μ und σ.

$$
\begin{aligned}
\text{height}_i &\sim \text{Normal}(\mu_i, \sigma) && \text{Likelihood} \\
\mu_i &= \alpha + \beta \cdot \text{weight}_i && \text{Lineares Modell} \\
\alpha &\sim \text{Normal}(178, 20) && \text{Priori} \\
\beta &\sim \text{Normal}(0, 10) && \text{Priori} \\
\sigma &\sim \text{Uniform}(0, 50) && \text{Priori}
\end{aligned}
$$

Likelihood, m43

$$\text{height}_i \sim \text{Normal}(\mu_i, \sigma) \qquad \text{Likelihood}$$

- Der Likelihood von `m43` ist ähnlich zu den vorherigen Modellen (`m41, m42`).
- Nur gibt es jetzt ein kleines "Index-i" am μ und am h (h wie `heights`).
- Es gibt jetzt nicht mehr nur einen Mittelwert μ, sondern für jede Beobachtung (Zeile) einen Mittelwert μ_i.
- Lies etwa so:

> "Die Wahrscheinlichkeit, eine bestimmte Größe bei Person i zu beobachten, gegeben μ und σ ist normalverteilt (mit Mittelwert μ und Streuung σ)".

Regressionsformel, m43

$$\mu_i = \alpha + \beta \cdot \text{weight}_i \qquad \text{Lineares Modell}$$

- μ ist jetzt nicht mehr ein Parameter, der (stochastisch) geschätzt werden muss. μ wird jetzt (deterministisch) *berechnet*. Gegeben α und β ist μ ohne Ungewissheit bekannt.
- weight_i ist der Prädiktorwert (`weight`) der $i$ten Beobachtung, also einer !Kung-Person (Zeile i im Datensatz).
- Lies etwa so:

> "Der Mittelwert μ_i der $i$ten Person berechnet sich als Summe von α und $\beta \cdot \text{weight}_i$".

- μ_i ist eine lineare Funktion von `weight`.
- β gibt den Unterschied in `height` zweier Beobachtung an, die sich um eine Einheit in `weight` unterscheiden (Steigung der Regressionsgeraden).
- α gibt an, wie groß μ ist, wenn `weight` Null ist.

Priori-Werte der Regression, m43

$$
\begin{aligned}
\alpha &\sim \text{Normal}(178, 20) && \text{Priori} \\
\beta &\sim \text{Normal}(0, 10) && \text{Priori} \\
\sigma &\sim \text{Uniform}(0, 50) && \text{Priori}
\end{aligned}
$$

- Parameter sind hypothetische Kreaturen: Man kann sie nicht beobachten, sie existieren nicht wirklich. Ihre Verteilungen nennt man Priori-Verteilungen.
- α wurde in `m41` als μ bezeichnet, da wir dort eine "Regression ohne Prädiktoren" berechnet haben.
- σ ist uns schon als Parameter bekannt und behält seine Bedeutung.
- β fasst unser Vorwissen, ob und wie sehr der Zusammenhang zwischen Gewicht und Größe positiv (gleichsinnig ist).
  - 🤔 Moment. Dieser Prior, β erachtet positive und negative Zusammenhang als gleich wahrscheinlich?!

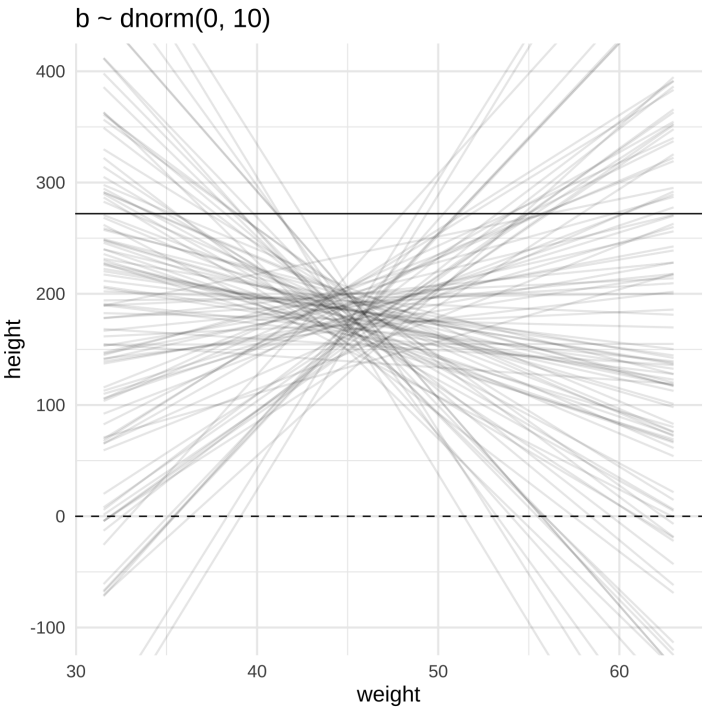
Prior-Prädiktiv-Simulation für m43

Wir simulieren 100 Regressionslinien:

```
n_lines <- 100
lines <-
  tibble(n = 1:n_lines,
         a = rnorm(n_lines,
                   mean = 178,
                   sd = 20),
           b = rnorm(n_lines,
                   mean = 0,
                   sd = 10))
```

n	a	b
1	194	−4
2	159	13
3	207	10

🤯 Oh nein! Viele dieser Regressionsgeraden sind unsinnig!

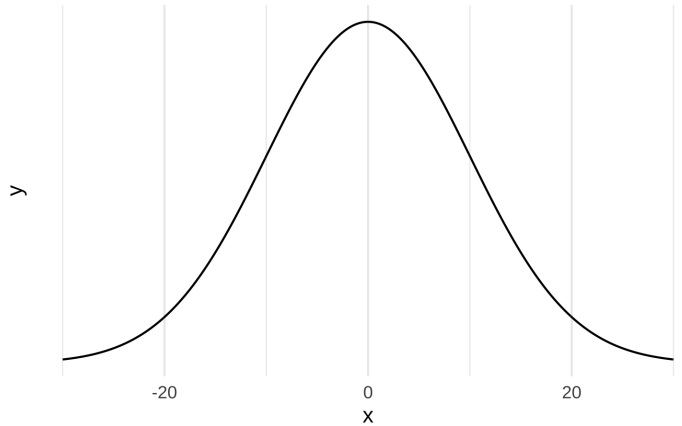


Die durchgezogene horizontale Linie gibt die Größe des größten Menschens, Robert Pershing Wadlow, an.

Wir müssen die Steigung zurecht stauchen

Oh no

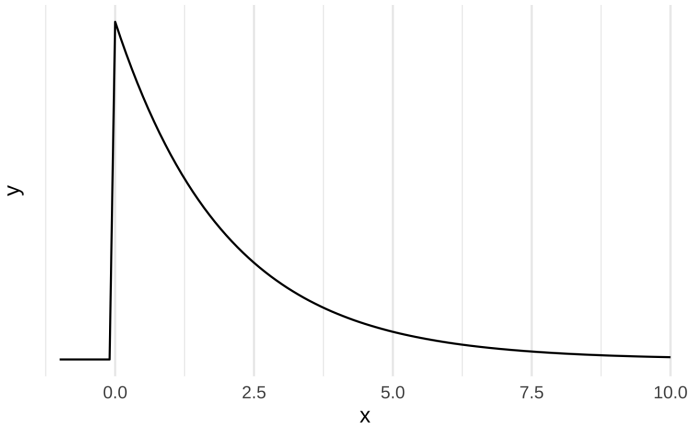
Eine Normalverteilung mit viel Streuung:



🤯 $\beta = -20$ ist gut möglich: Pro kg Gewicht sind Menschen im Schnitt 20cm kleiner, laut dem Modell. Quatsch.

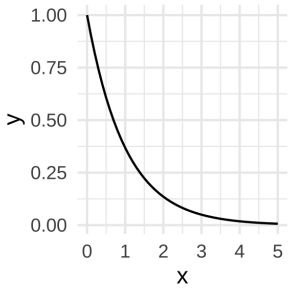
Oh yes

Wir bräuchten eher so eine Verteilung:



😍 Wo gibt's diese Verteilung?

Darf ich vorstellen: Die Exponential-Verteilung 🍾



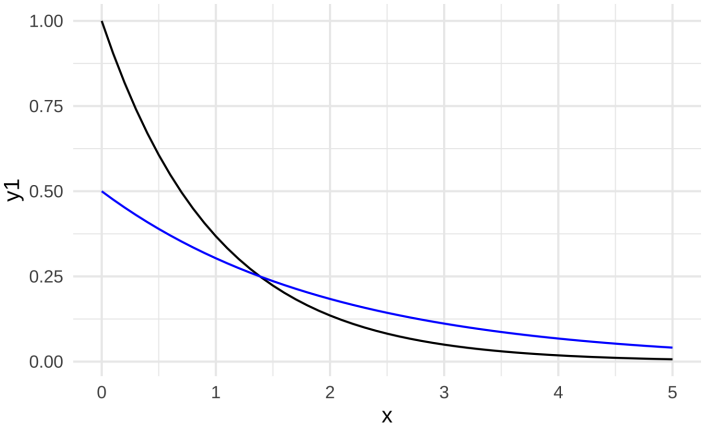
$\beta \sim \text{Exp}(1)$

- Eine *Exp*onentialverteilung ist nur für positive Werte, $x > 0$, definiert.
- Sie ist eine praktische Wahl, wenn man einen Parameter auf einen positiven Wertebereich bändigen möchte.
- Steigt X um eine Einheit, so verändert sich Y um einen konstanten Faktor.
- Sie hat nur einen Parameter, λ; $\frac{1}{\lambda}$ gibt die Streuung ("Gestrecktheit") der Verteilung an.
- Die Verteilung für β ist plausibel:
  - Nur positive Steigungen
  - Keine sehr starken Zusammenhänge.

Simulieren wir mal die Priori-Prädiktiv-Verteilung und schauen, was passiert.

Exponentialverteilung mit R

```
d <-
  tibble(
    x = seq(0, 5,.1),
    y1 = dexp(x, rate = 1),
    y2 = dexp(x, rate = 0.5)
  )

d %>%
  ggplot(aes(x)) +
  geom_line(aes(y = y1)) +
  geom_line(aes(y = y2),
            color = "blue")
```

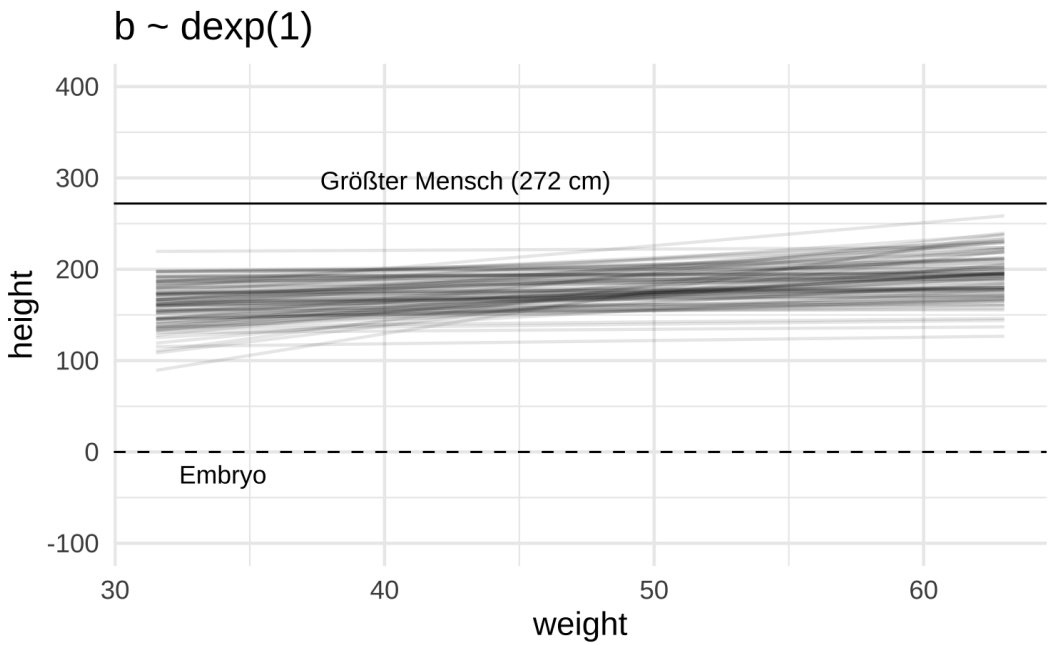


$\beta \sim \text{Exp}(1)$

$\beta \sim \text{Exp}(0.5)$

Priori-Prädiktiv-Simulation, 2. Versuch

$$\beta \sim \text{Exp}(1)$$

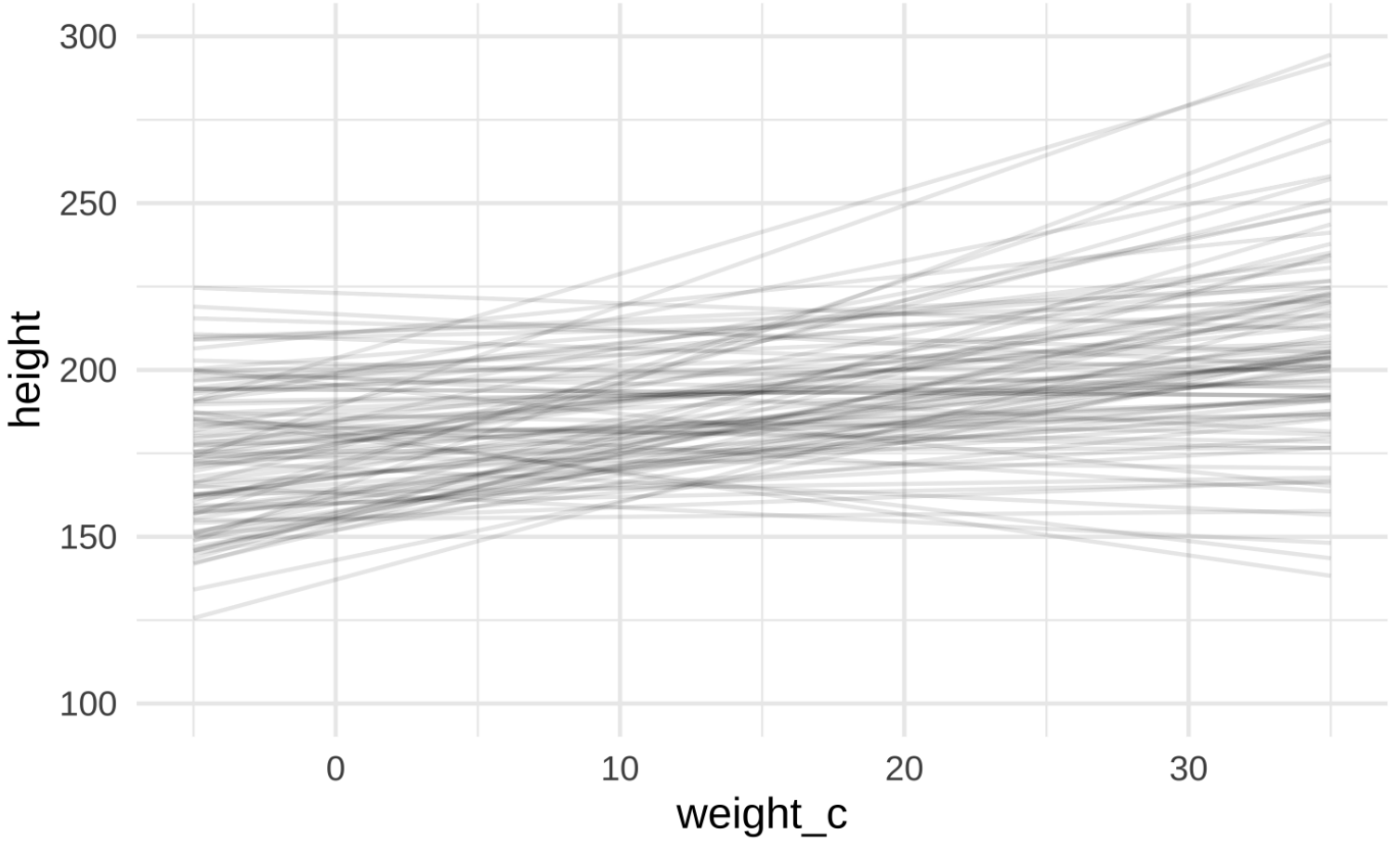


Das sieht gut aus; unsere Priori-Werte scheinen vernünftige Vorhersagen zu tätigen.

Prior-Prädiktiv mit R plotten: R-Code

```
n_lines <- 100  # 100 Regr.linien simulieren

lines1 <-
  tibble(n = 1:n_lines,
         a = rnorm(n_lines, mean = 178, sd = 20),  # Prior alpha
         b = rexp(n_lines, 1))  %>%   # Prior beta
  mutate(weight_c = 40-45,  # Gewicht einer leichten Person
         height = a + weight_c*b)  # Größe einer leichten Person

lines2 <-
  tibble(n = 1:n_lines,
         a = rnorm(n_lines, mean = 178, sd = 20),  # Prior alpha
         b = rexp(n_lines, 1))  %>%   # Prior beta
  mutate(weight_c = 80-45,  # Gewicht einer schweren Person
         height = a + weight_c*b)  # Größe einer schweren Person

lines_doppelt <-  # zwei Punkte pro Linie definieren eine Linie
  lines1 %>%
  bind_rows(lines2)

prior_pred_plot <- # ein Wertepaar von "n" ist eine Gruppe, d.h. eine Linie
  lines_doppelt %>%
  ggplot(aes(x = weight_c, y = height, group = n)) +
  geom_line(alpha = 0.1) +
  ylim(100, 300)
```

Prior-Prädiktiv mit R plotten: Ausgabe



Moment, kann hier jeder machen, was er will?

Es doch den einen, richtigen, objektiven Priori-Wert geben?!

Kann denn jeder hier machen, was er will?! Wo kommen wir da hin?!

> This is a mistake. There is no more a uniquely correct prior than there is a uniquely correct likelihood. Statistical models are machines for inference. Many machines will work, but some work better than others. Priors can be wrong, but only in the same sense that a kind of hammer can be wrong for building a table. McElreath (2020), p. 96.

Hier ist unser Modell, m43a

$$
\begin{aligned}
\text{height}_i &\sim \text{Normal}(\mu_i, \sigma) \\
\mu_i &= \alpha + \beta \cdot \text{weight}_i \\
\alpha &\sim \text{Normal}(178, 20) \\
\beta &\sim \text{Exp}(1) \\
\sigma &\sim \text{Uniform}(0, 50)
\end{aligned}
$$

```
# Zufallszahlen festlegen:
set.seed(42)
# Posteriori-Vert. berechnen:
m43a <-
  quap(
    alist(
      height ~ dnorm(mu, sigma),
      mu <- a + b*weight_c,
      a ~ dnorm(178, 20),
      b ~ dexp(1),
      sigma ~ dunif(0, 50)
    ),
    data = d2)
```

```
precis(m43a)
```

```
##              mean         sd       5.5%       94.5%
## a     154.6489113 0.27465244 154.209964 155.0878590
## b       0.9047461 0.04261317   0.836642   0.9728502
## sigma   5.1093085 0.19422260   4.798903   5.4197138
```

Voilà! Die Posteriori-Verteilung für `m43a`.

Die Post-Verteilung befragen

Post-Verteilung 1: Mittelwerte von α und β

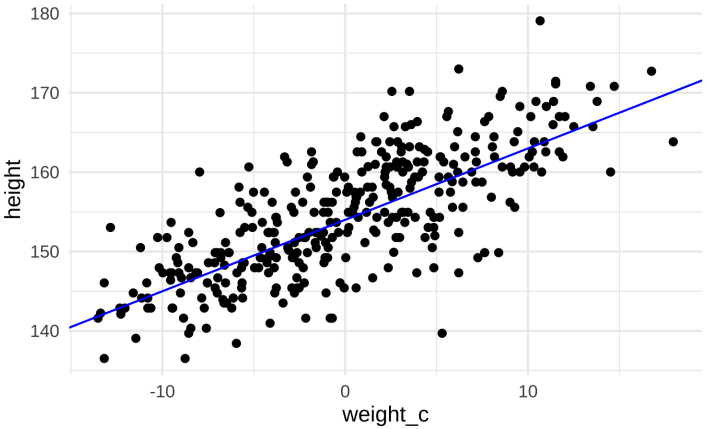
```
post_m43a <-
  extract.samples(m43a)
```

a	b	sigma
154.7	0.9	5.1
154.8	0.9	5.0
154.8	0.9	5.2

```
post_m43a_summary <-
  post_m43a %>%
  summarise(
    a_mean = mean(a),
    b_mean = mean(b),
    s_mean = mean(sigma))
```

a_mean	b_mean	s_mean
154.6	0.9	5.1

```
d2 %>%
  ggplot() +
  aes(x = weight_c, y = height) +
  geom_point() +
  geom_abline(
    slope = 0.9,
    intercept = 154,
    color = "blue")
```



Zentrale Statistiken zu den Parametern

In diesem Modell gibt es drei Parameter: μ, β, σ.

Mittelwerte

- Mittlere Größe?
- Schätzwert für den Zusammenhang von Gewicht und Größe?
- Schätzwert für Ungewissheit in der Schätzung der Größe?

```
post_m43a_summary
```

```
##     a_mean    b_mean   s_mean
## 1 154.6458 0.9051015 5.109428
```

Streuungen

- Wie unsicher sind wir uns in den Schätzungen der Parameter?

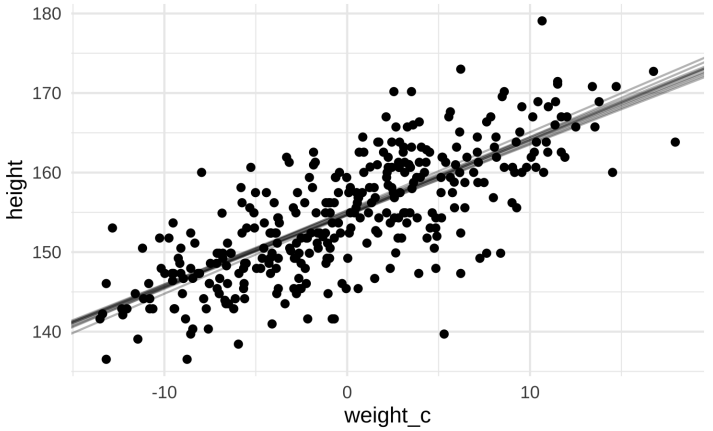
```
post_m43a_summary2 <-
  post_m43a %>%
  summarise(
    a_sd = sd(a),
    b_sd = sd(b),
    s_sd = sd(sigma))
```

a_sd	b_sd	s_sd
0.28	0.04	0.20

Post-Verteilung 2: Ungewissheit von α und β

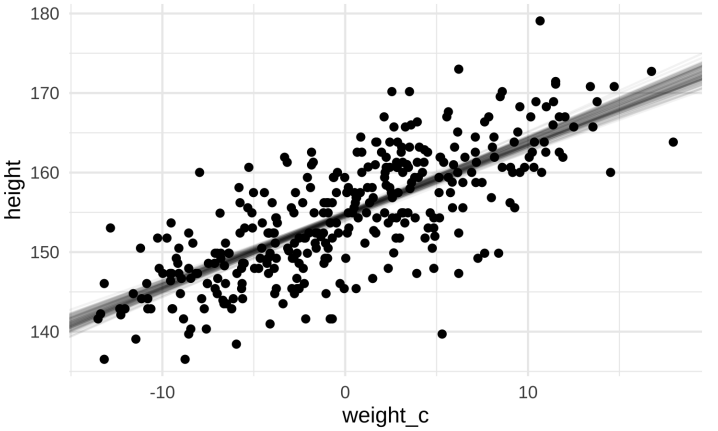
Die ersten 10 Stichproben

```
d2 %>%
  ggplot(aes(x = weight_c, y = height)) +
  geom_point() +
  geom_abline(data = post_m43a %>% slice_head(n=1
              aes(slope = b,
                  intercept = a),
              alpha = .3)
```



Die ersten 100 Stichproben

```
d2 %>%
  ggplot(aes(x = weight_c, y = height)) +
  geom_point() +
  geom_abline(data = post_m43a %>% slice_head(n=1
              aes(slope = b,
                  intercept = a),
              alpha = .05)
```



Fragen zum Achsenabschnitt (mittlere Größe)

- Welche mittlere Größe mit zu 50%, 90% Wskt. nicht überschritten?
- Welche mittlere Größe mit zu 95% Wskt. nicht unterschritten?
- Von wo bis wo reicht der innere 50%-Schätzbereich der mittleren Größe?
- Wie wahrscheinlich ist es, dass die mittlere Größe bei 155 cm oder mehr liegt?

```
##       q_50     q_90    q_05
## 1 154.6471 155.0014 154.187
```

```
##      pi_50
## 1 154.4585
## 2 154.8309
```

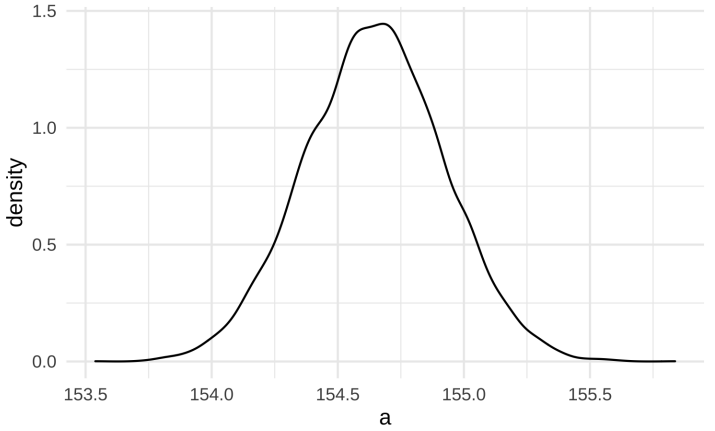
```
##   a >= 155    n
## 1    FALSE 8990
## 2     TRUE 1010
```

```
post_m43a %>%
  summarise(
    q_50 =
      quantile(a, prob = .5),
    q_90 =
      quantile(a, prob = .9),
    q_05 =
      quantile(a, prob = .05))

post_m43a %>%
  summarise(
    pi_50 =
    quantile(a,
             prob = c(.25, .75)))

post_m43a %>%
  count(a >= 155)
```

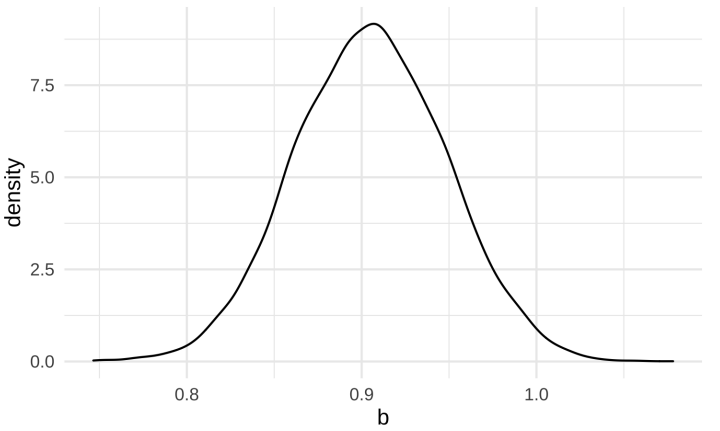
Ungewissheit von Achsenabschnitt und Steigung

... als Histogramme visualisiert

```
post_m43a %>%
  ggplot(aes(x = a)) +
  geom_density()
```




```
post_m43a %>%
  ggplot(aes(x = b)) +
  geom_density()
```

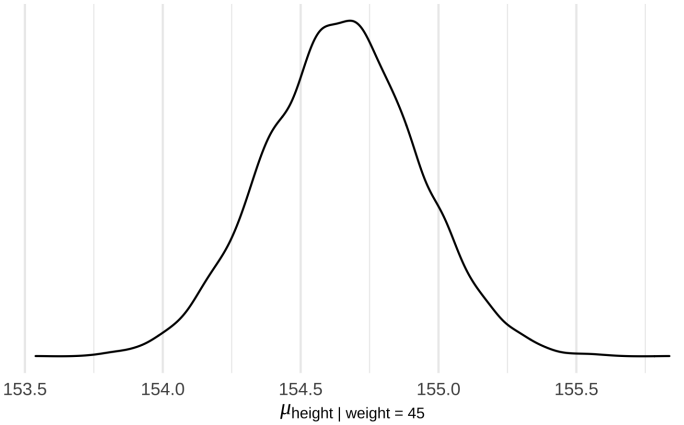


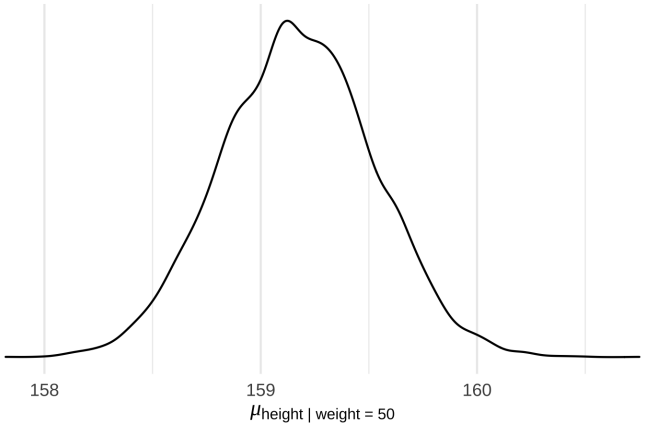
Ungewissheit für $\mu|\text{weight} = 45, 50$

```
mu_at_45_50 <-
  post_m43a %>%
  mutate(mu_at_45 = a,
         mu_at_50 = a + b * 5)
# 50kg ist 5 kg über dem MW
# b ist zentriert: b=0 ist MW von weight
```

```
mu_at_45_50 %>%
  ggplot(aes(x = mu_at_45)) +
  geom_density()
```

```
mu_at_45_50 %>%
  ggplot(aes(x = mu_at_50)) +
  geom_density()
```





Wie groß ist ein !Kung mit 50kg Gewicht im Mittel?

```
mu_at_45_50 %>%
  summarise(pi = quantile(mu_at_50, prob = c(0.5, .9)))
```

```
##         pi
## 1 159.1682
## 2 159.6245
```

Die mittlere Größe liegt mit 90% Wahrscheinlichkeit zwischen den beiden Werten.

Welche mittlere Größe wird mit 95% Wahrscheinlichkeit nicht überschritten, wenn die Person 45kg wiegt?

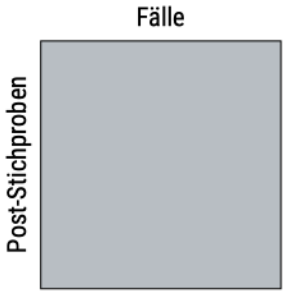
```
##      q_95
## 1 155.101
```

Posteriori-Verteilungen für alle Größen

Die Funktion `link()` erstellt eine Posteriori-Verteilung für jeden Wert des Prädiktors (`weight`), im Standard werden 1000 Stichproben aus der Posteriori-Verteilung gezogen (für jede Beobachtung im Datensatz `d2`).

```
weight_seq <-
  seq(-20,20, by = 1)
mu <- link(m43a,
  data = tibble(
    weight_c=weight_seq)) %>%
  as_tibble()

dim(mu)  # 1000 Zeilen, 41 Spalten
```

```
## [1] 1000   41
```



Auszug aus `mu`:

V1	V2	V3	V4	V5	V6	V7
136	137	138	139	140	141	142
136	137	138	139	140	141	142
137	138	139	140	140	141	142
137	138	139	140	141	142	143
136	137	138	139	140	141	142

In den *Zeilen* stehen die (1000) Stichproben aus der Posteriori-Verteilung; in den *Spalten* die (41) verschiedenen Gewichtswerte. In den Zellen steht jeweils die mittere geschätzte Größe.

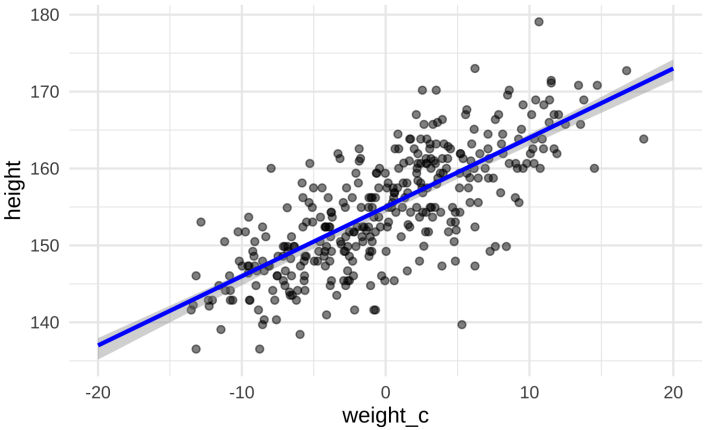
!Kung-Größen mit Schätzbereich für μ

```
mu_tidy <-
  mu %>%
  map_dfr(PI) %>%
  mutate(
    weight_c = weight_seq,
    mu = 155 + 0.9*weight_c)
```

5%	94%	weight_c	mu
135	138	−20	137
136	139	−19	138
137	140	−18	139
138	140	−17	140
139	141	−16	141

`PI()` berechnet im Default Intervalle mit 89%-Breite (die Zahl gefiel dem Autor der Funktion. 🤷)

```
d2 %>%
  ggplot(aes(x = weight_c)) +
  geom_point(aes(
    y = height),
    alpha = .5) +
  geom_smooth(data = mu_tidy,
              aes(y = mu,
                  ymin = `5%`,
                  ymax = `94%`),
              stat = "identity",
              color = "blue")
```



Körpergrößen simulieren: PPV

- Die Posteriori-Verteilung (`m43_post`) gibt uns Stichproben für die Parameter, d.i. α, σ, β des Modells, z.B. Zeile 1:

id	a	b	sigma
1	155	1	5

- Auf dieser Basis können wir $h_i \sim \mathcal{N}(\mu_i = \alpha, \sigma)$ schätzen, also eine tatsächliche Größe:

```
h_1 <- rnorm(n = 1, mean = 155, sd = 5)
h_1
```

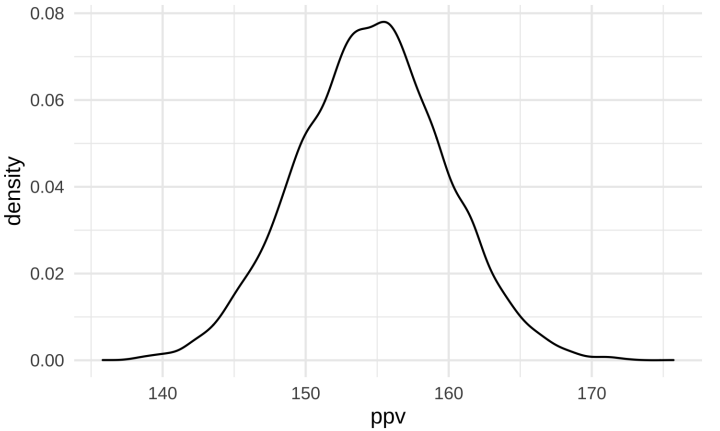
```
## [1] 163.6429
```

- Das wiederholen wir oft (z.B. 10^4 Mal).
- Voilà: Unsere Posteriori-Prädiktiv-Verteilung (PPV).

PPV plotten

```
set.seed(42)
ppv <-
  tibble(
    ppv = rnorm(
      n = 1e4,
      mean = post_m43a$a,
      sd = post_m43a$sigma))
```

```
ppv %>%
  ggplot(aes(x = ppv)) +
  geom_density()
```



Fragen an die PPV

- Wie groß sind die !Kung im Schnitt?
- Welche Größe wird von 90% der Personen nicht überschritten?
- Wie groß sind die 10% kleinsten?

```
ppv %>%
  summarise(
    q_50 = quantile(
      ppv, prob = .5),
    height_mean = mean(ppv),
    q_90 = quantile(
      ppv, prob = .9),
    q_10 = quantile(
      ppv, prob = .1)
  )
```

```
## # A tibble: 1 × 4
##    q_50 height_mean  q_90  q_10
##   <dbl>       <dbl> <dbl> <dbl>
## 1  155.        155.  161.  148.
```

- Was ist der 50% Bereich der Körpergröße?

```
ppv %>%
  summarise(
    pi_50 = quantile(ppv,
                     prob = c(.25, .75))
  )
```

```
## # A tibble: 2 × 1
##   pi_50
##   <dbl>
## 1  151.
## 2  158.
```

Vorhersage-Intervall

Simulieren wir die tatsächlichen Größen bedingt auf ein bestimmtes Körpergewicht:

id	a	b	sigma
1	155	1	5

- Auf dieser Basis können wir $h_i \sim \mathcal{N}(\mu_i = \alpha + \beta \cdot \text{weight}_i, \sigma)$ schätzen, also eine tatsächliche Größe, bedingt auf ein Gewicht, sagen wir 55 kg (10kg über dem Mittelwert):

```
h_1_55kg <- rnorm(n = 1, mean = 155 + 0.9*10, sd = 5)
h_1_55kg
```

```
## [1] 167.8589
```

- Das wiederholen wir oft (z.B. 10^4 Mal) für jeden Gewichtswert, der uns interessiert.
- Voilà: Unsere PPV bedingt auf die Gewichtswerte.
- Mit `sim()` können wir uns diese Arbeit abnehmen lassen.

Bedingte PPV visualisiert, 1

`sim()` berechnen eine PPV bedingt auf Gewichtswerte

```
sim_height <- sim(
  m43a,
  data = tibble(
    weight_c = weight_seq)
  ) %>%
  as_tibble()

dim(sim_height)
```

```
## [1] 1000   41
```

V1	V2	V3	V4	V5
135	130	154	135	133
126	133	136	145	141
141	139	129	141	138
134	128	142	135	143
154	137	142	141	137

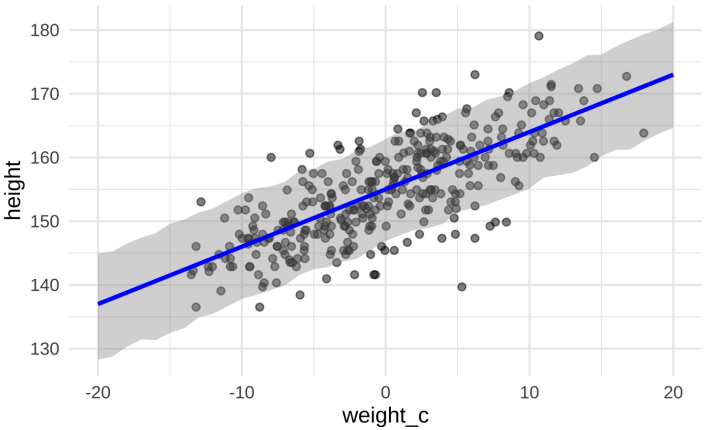
Die Tabelle sieht aus wie die vorherige, `(\mu)`, aber sie enthält simulierte Größenwerte, keine `(\mu)`-Werte.

Bedingte PPV visualisiert, 2

```
sim_height_tidy <-
  sim_height %>%
  map_dfr(PI) %>%
  mutate(
    weight_c = weight_seq,
    mu = 155 + 0.9*weight_c)
```

5%	94%	weight_c	mu
128	145	−20	137
129	145	−19	138
130	146	−18	139
131	147	−17	140
131	148	−16	141

```
d2 %>%
  ggplot(aes(x = weight_c)) +
  geom_point(aes(
    y = height),
    alpha = .5) +
  geom_smooth(data = sim_height_tidy,
              aes(y = mu,
                  ymin = `5%`,
                  ymax = `94%`),
              stat = "identity",
              color = "blue")
```



Hinweise

Zu diesem Skript

- Dieses Skript bezieht sich auf folgende Lehrbücher:
  - *Statistical Rethinking* (2. Auflage), Kapitel 4.4, McElreath (2020)
  - Der R-Code stammt aus Kurz (2021).
- Dieses Skript wurde erstellt am 2021-10-25 00:01:35 (WiSe 21).
- Lizenz: CC-BY
- Autor ist Sebastian Sauer.
- Um diese HTMLM-Folien korrekt darzustellen, ist eine Internet-Verbindung nötig.
- Eine PDF-Version kann erzeugt werden, indem man im Chrome-Browser druckt (Drucken als PDF).
- Mit der Taste ? bekommt man eine Hilfe über Shortcuts.

Literatur

Kurz, A. S. (2021). *Statistical rethinking with brms, ggplot2, and the tidyverse: Second edition*. URL: https://bookdown.org/content/4857/ (visited on Sep. 08, 2021).

McElreath, R. (2020). *Statistical rethinking: a Bayesian course with examples in R and Stan*. 2nd ed. CRC texts in statistical science. Boca Raton: Taylor and Francis, CRC Press. ISBN: 978-0-367-13991-9.