Notes on Chapter 4 of Statistical Rethinking

Matthias Grenié

24 juillet 2016

This document are notes taken when reading chapter 4 of *Statistical Rethinking* from Richard McElreath

Notes

Linear regression specification using Bayesian statistics.

```
library(rethinking)
```

```
## Loading required package: rstan

## Loading required package: ggplot2

## Loading required package: StanHeaders

## rstan (Version 2.10.1, packaged: 2016-06-24 13:22:16 UTC, GitRev: 85f7a56811da)

## For execution on a local, multicore CPU with excess RAM we recommend calling
## rstan_options(auto_write = TRUE)
## options(mc.cores = parallel::detectCores())

## Loading required package: parallel

## rethinking (Version 1.59)
```

```
data(Howell1)
d2 = Howell1[Howell1$age >= 18,]
```

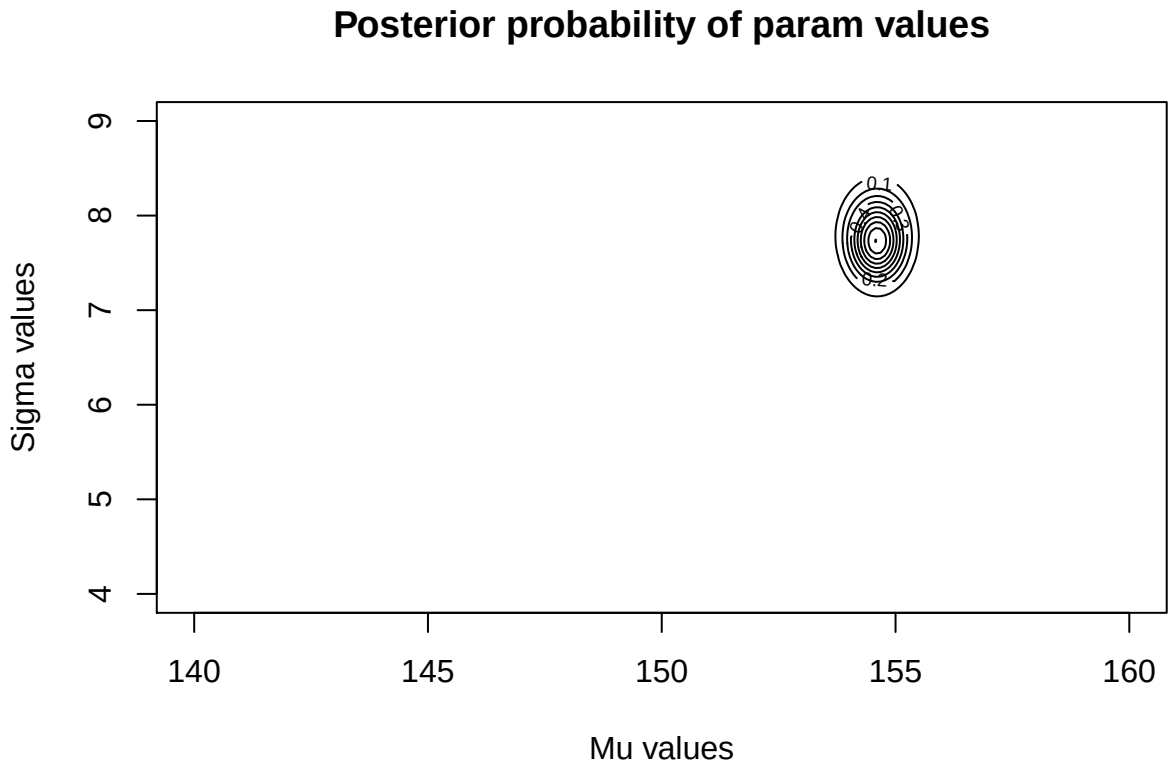
Model of height of adults:

$$\left.\begin{array}{ll} h_i & \sim \text{Normal}(\mu, \sigma), \\ \mu_i & \sim \text{Normal}(178, 20), \\ \sigma & \sim \text{Uniform}(0, 50) \end{array}\right\} \Leftrightarrow h_i = \mu + \epsilon_i, \epsilon_i \sim \text{Normal}(0, \sigma)$$

Test of posterior distribution computation:

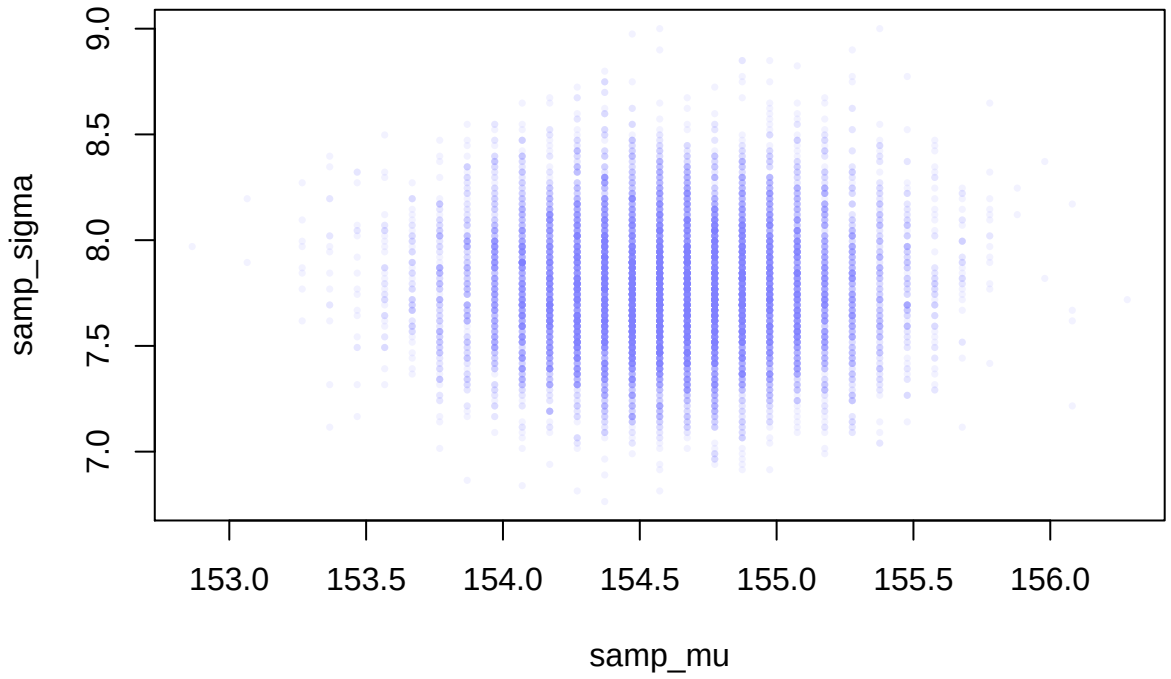
```
mu.list <- seq( from=140, to=160 , length.out=200 )
sigma.list <- seq( from=4 , to=9 , length.out=200 )
post <- expand.grid( mu=mu.list , sigma=sigma.list )
post$LL <- sapply( 1:nrow(post) , function(i) sum( dnorm(
                d2$height ,
                mean=post$mu[i] ,
                sd=post$sigma[i] ,
                log=TRUE ) ) )
post$prod <- post$LL + dnorm( post$mu , 178 , 20 , TRUE ) +
    dunif( post$sigma , 0 , 50 , TRUE )
post$prob <- exp( post$prod - max(post$prod) )
```

```
contour_xyz(post$mu, post$sigma, post$prob, xlab = "Mu values",
            ylab = "Sigma values", main = "Posterior probability of param values")
```



Now we can sample from posterior:

```
samp_rows = sample(1:nrow(post), size = 1e4, replace = TRUE, prob = post$prob)
samp_mu = post$mu[samp_rows]
samp_sigma = post$sigma[samp_rows]
plot(samp_mu, samp_sigma, cex = 0.5, pch = 16, col = col.alpha(rangi2, 0.1))
```



Using MAP

```
flist = alist(
  height ~ dnorm(mu, sigma),
  mu ~ dnorm(178, 20),
  sigma ~ dunif(0, 50)
)

m4.1 = map(flist, data = d2)
```

Predicting Height from weight

```
m4.3 = map(
  alist(
    height ~ dnorm(mu, sigma),
    mu <- alpha + beta*weight,
    alpha ~ dnorm(178, 100),
    beta ~ dnorm(0, 10),
    sigma ~ dunif(0, 50)
  ),
  data = d2)
```

Interpretation using table of estimates:

```
precis(m4.3, corr = TRUE)
```

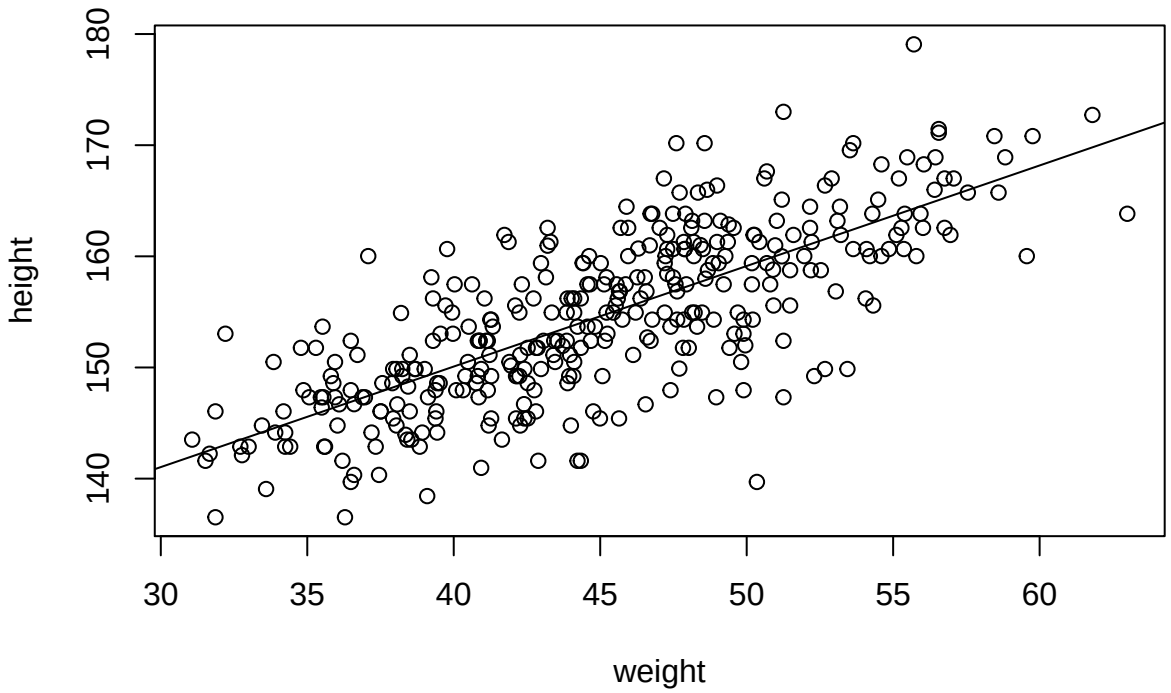
```
##         Mean StdDev   5.5%  94.5% alpha  beta sigma
```

```
## alpha 113.90   1.91 110.86 116.95  1.00 -0.99     0
## beta    0.90   0.04   0.84   0.97 -0.99  1.00     0
## sigma   5.07   0.19   4.77   5.38  0.00  0.00     1
```

Strong negative correlation between `a` and `b`, can center weight to avoid this correlation:

```
d2$weight.c = d2$weight - mean(d2$weight)
m4.4 = map(
  alist(
    height ~ dnorm(mu, sigma),
    mu <- alpha + beta * weight.c,
    alpha ~ dnorm(178, 100),
    beta ~ dnorm(0, 10),
    sigma ~ dunif(0, 50)
  ),
  data = d2
)
```

```
plot(height ~ weight, data = d2)
abline(a = coef(m4.3)["alpha"], b = coef(m4.3)["beta"])
```




```
weight.seq = seq(from = 25, to = 75, by = 1)
mu = link(m4.3, data = data.frame(weight = weight.seq))
```

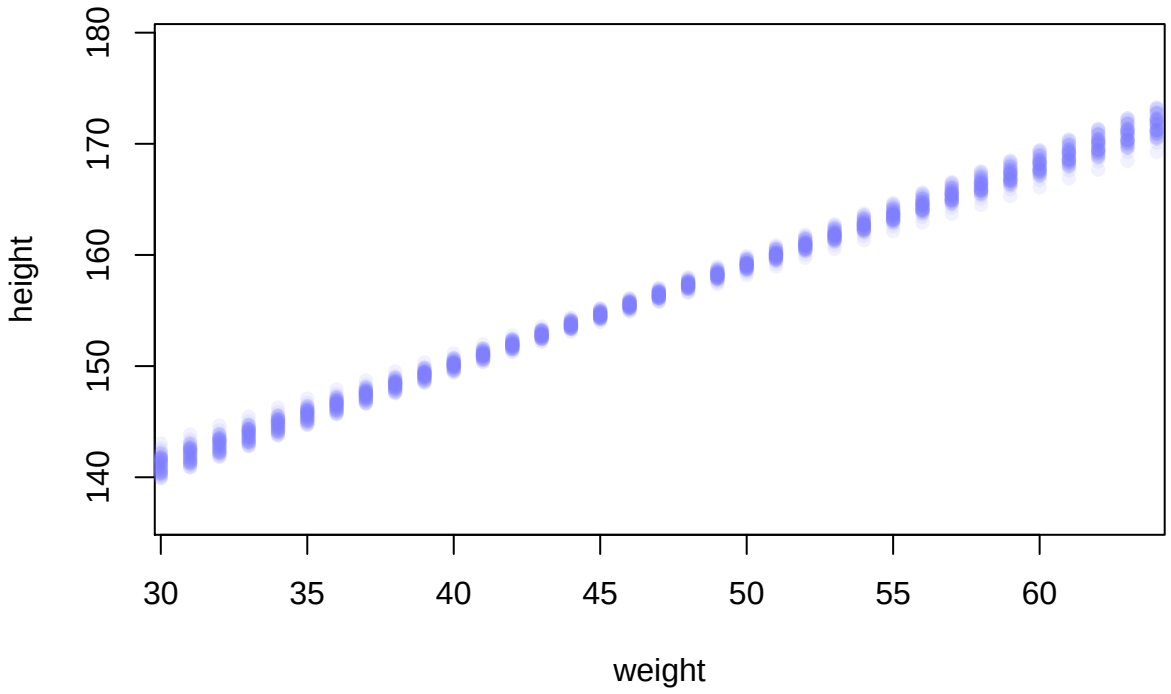
```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
```

```
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

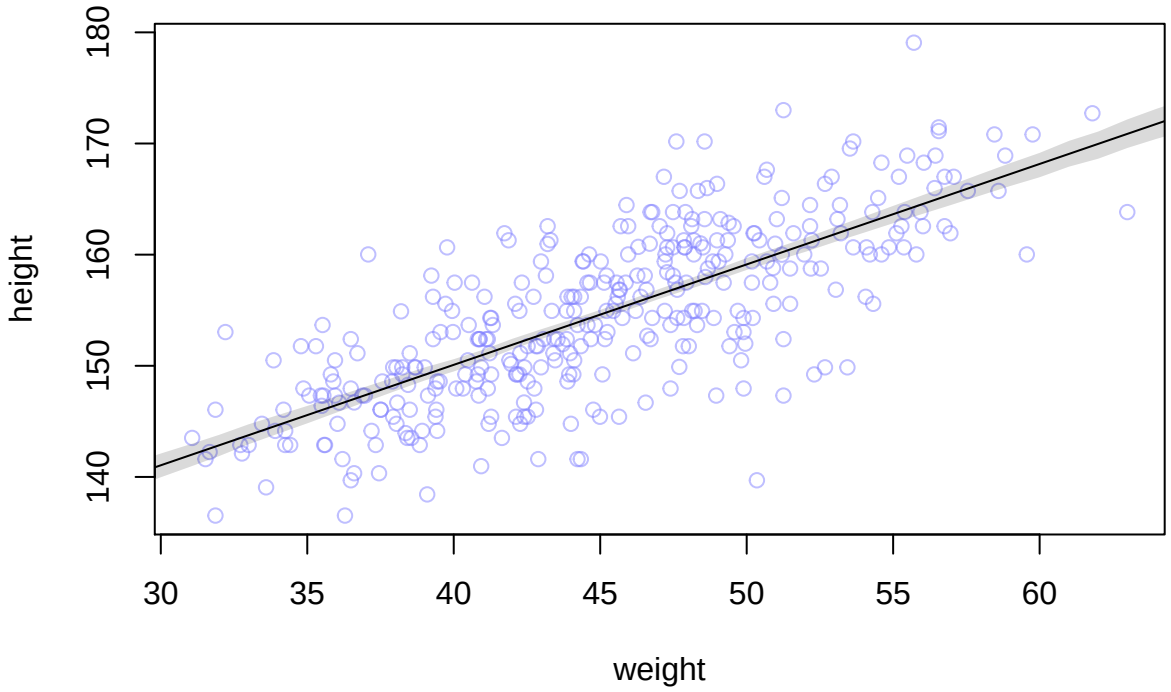
```
str(mu)
```

```
##  num [1:1000, 1:51] 137 138 135 136 136 ...
```

```
plot(height ~ weight, type = "n", data = d2)
  for (i in 1:51) {
    points(weight.seq, mu[i,], pch = 16, col = col.alpha(rangi2, 0.1))
  }
```




```
mu.mean = apply(mu, 2, mean)
mu.HPDI = apply(mu, 2, HPDI, prob = 0.89)
plot(height ~ weight, d2, col = col.alpha(rangi2, 0.5))
lines(weight.seq, mu.mean)
shade(mu.HPDI, weight.seq)
```

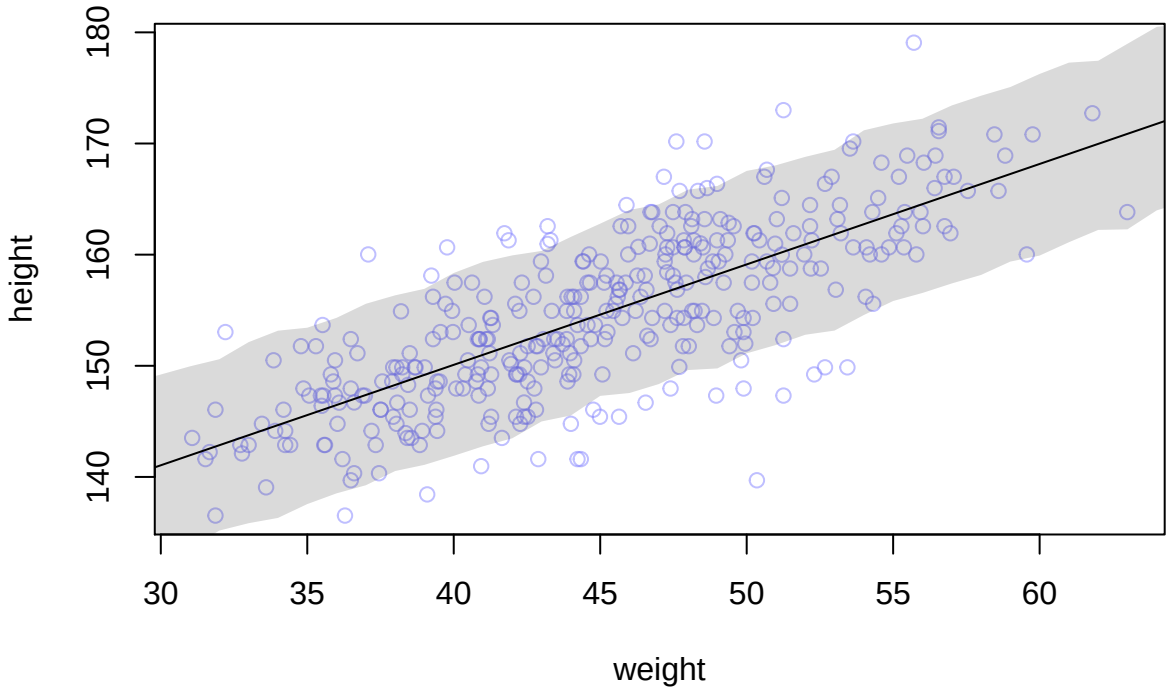


Shading indicates the 89% highest posterior density of interval of prediction of mean μ. Not the confidence interval of the prediction of height using weight exactly.

```
sim.height = sim(m4.3, data = list(weight = weight.seq))
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```
height.PI = apply(sim.height, 2, PI, prob = 0.89)

# Plot
plot(height ~ weight, d2, col = col.alpha(rangi2, 0.5))
lines(weight.seq, mu.mean)
shade(height.PI, weight.seq)
```



Including children (Polynomial regression)

```
d = Howell1
d$weight.s = (d$weight - mean(d$weight)) / sd(d$weight)

d$weight.s2 = d$weight.s^2

m4.5 = map(
  alist(
    height ~ dnorm(mu, sigma),
    mu <- alpha + beta1 * weight.s + beta2 * weight.s2,
    alpha ~ dnorm(178, 100),
    beta1 ~ dnorm(0, 10),
    beta2 ~ dnorm(0, 10),
    sigma ~ dunif(0, 59)
  ),
  data = d
)
```

```
# Get idea of posterior distribution and prediction
weight.seq = seq(-2.2, to = 2.2, length.out = 30)
pred_data = list(weight.s = weight.seq, weight.s2 = weight.seq^2)
mu = link(m4.5, data = pred_data)
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
```

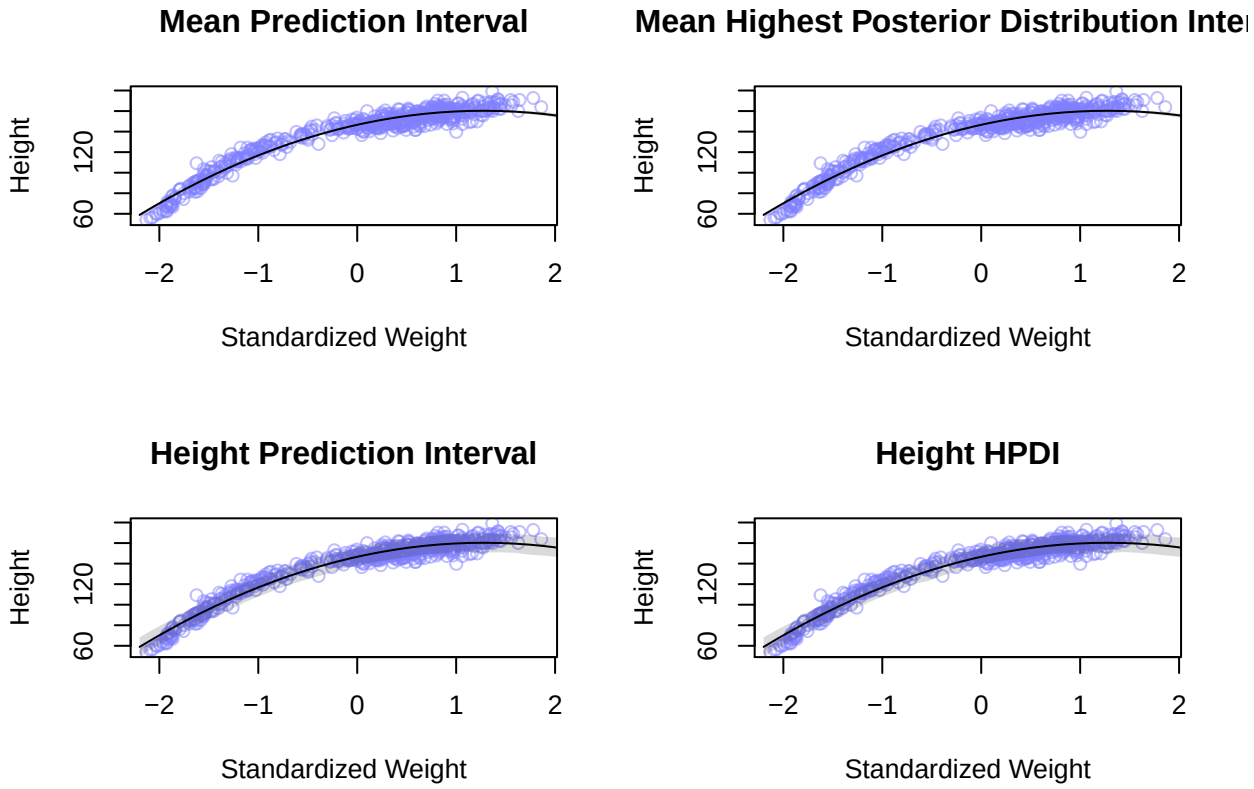
```
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```
mu.mean = apply(mu, 2, mean)
mu.HPDI = apply(mu, 2, HPDI, prob = 0.89)
mu.PI = apply(mu, 2, PI, prob = 0.89)

sim.height = sim(m4.5, data = pred_data)
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```
height.PI = apply(sim.height, 2, PI, prob = 0.89)
height.HPDI = apply(sim.height, 2, HPDI, prob = 0.89)
```

```
# Plot of data and model
base_plot = function() {
  plot(height ~ weight.s, d, col = col.alpha(rangi2, 0.5),
       xlab = "Standardized Weight", ylab = "Height")
  lines(weight.seq, mu.mean)
}

par(mfrow = c(2, 2))

base_plot()
title(main = "Mean Prediction Interval")
shade(mu.PI, weight.seq)

base_plot()
title(main = "Mean Highest Posterior Distribution Interval")
shade(mu.HPDI, weight.seq)

base_plot()
title(main = "Height Prediction Interval")
shade(height.PI, weight.seq)

base_plot()
title(main = "Height HPDI")
shade(height.HPDI, weight.seq)
```




```
par(mfrow = c(1, 1))
```

Practice

Easy

4E1

The likelihood is $y_i \sim \text{Normal}(\mu, \sigma)$

4E2

There are **two** parameters in the posterior distribution (μ and σ).

4E3

$$P(\mu, \sigma|y) = \frac{\prod_i \text{Normal}(y_i|\mu, \sigma)\text{Normal}(\mu|0, 10)\text{Uniform}(\sigma|0, 10)}{\int \text{Normal}(y_i|\mu, \sigma)\text{Normal}(\mu|0, 10)\text{Uniform}(\sigma|0, 10)d\mu d\sigma}$$

4E4

The line with the linear model is $\mu_i = \alpha + \beta x_i$.

4E5

There are **three** parameters in the posterior distribution (α, β and σ).

Medium

4M1

Need to sample from the prior:

```
samp_mu = rnorm(100, 0, 10)
samp_sigma = runif(100, 0, 10)

N = sample(1:length(samp_mu), size = 10)
samp_heights = rnorm(10, mean = samp_mu[N], sd = samp_sigma[N])
```

4M2

```
map(
  alist(
    y ~ dnorm(mu, sigma),
    mu ~ dnorm(0, 10),
    sigma ~ dunif(0, 10)
  )
)
```

4M3

$$
\begin{aligned}
y_i &\sim \text{Normal}(\mu_i, \sigma), \\
\mu_i &= a + b \times x_i, \\
a &\sim \text{Normal}(0, 50), \\
b &\sim \text{Normal}(0, 10), \\
\sigma &\sim \text{Uniform}(0, 50)
\end{aligned}
$$

4M4

$$
\begin{aligned}
\text{height}_i &\sim \text{Normal}(\mu_i, \sigma), \\
\mu_i &= a + b\text{year}_i, \\
a &\sim \text{Normal}(178, 100), \\
b &\sim \text{Normal}(0, 10), \\
\sigma &\sim \text{Uniform}(0, 50)
\end{aligned}
$$

Uninformative prior for b and σ and a not so constrained one because of lack of information.

4M5

Yes, it changes our choice of prior. Because if in the first year the height of student is around 120cm you could guess that a is around 120 cm and center its distribution around this value. An d if each student gets taller every year, than we could assume that b is positive, so assume for example a strictly positive prior for b.

4M6

We can argue if it is never more than 64cm that the prior distribution of σ can be Uniform between 0 and 64.

Hard

4H1

```
# We can use the model used in 4.3

given_weight = data.frame(ind = 1:5, weight = c(46.95, 43.72, 64.78, 32.59, 54.63))

pred_height = link(m4.3, data = list(weight = c(46.95, 43.72, 64.78, 32.59, 54.63)))
```

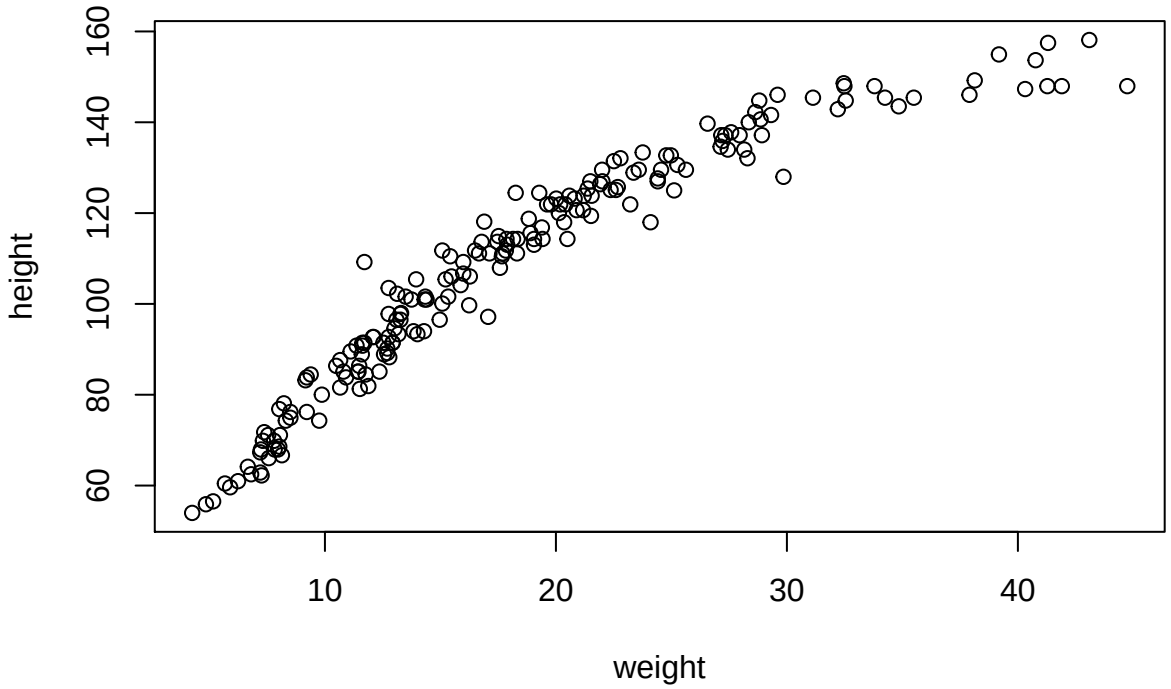
```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```
given_weight$pred_height = apply(pred_height, 2, mean)
height_HPDI = apply(pred_height, 2, HPDI, prob = 0.89)



given_weight$HPDI_interval = apply(apply(height_HPDI, 2, round, digits = 2), 2, paste, collapse = " - ")

given_weight
```

```
##   ind weight pred_height   HPDI_interval
## 1   1  46.95    156.3627 155.97 - 156.84
## 2   2  43.72    153.4413 152.99 - 153.86
## 3   3  64.78    172.4886 171.13 - 173.87
## 4   4  32.59    143.3750 142.39 - 144.25
## 5   5  54.63    163.3087 162.58 - 164.09
```

4H2

```
children = Howell1[Howell1$age < 18,]

plot(height ~ weight, children)
```



We want to fit a linear regression between the height and the weight of each child, the model would be as follow:

$$
\begin{aligned}
h_i &\sim \text{Normal}(\mu_i, \sigma), \\
\mu_i &= a + b \times w_i, \\
a &\sim \text{Normal}(100, 100), \\
b &\sim \text{Normal}(0, 10), \\
\sigma &\sim \text{Uniform}(0, 50)
\end{aligned}
$$

which gives using `map()` function:

```
child_model = map(
  alist(
    height ~ dnorm(mu, sigma),
    mu <- a + b * weight,
    a ~ dnorm(100, 100),
    b ~ dnorm(0, 10),
    sigma ~ dunif(0, 50)
  ),
  data = children
)

precis(child_model)
```

```
##         Mean StdDev  5.5% 94.5%
## a     58.20   1.40 55.97 60.43
## b      2.72   0.07  2.61  2.83
## sigma  8.43   0.43  7.74  9.12
```

(a) As we see for every increase of ten units in mass a child is 27.2 units taller according to the model.

(b)

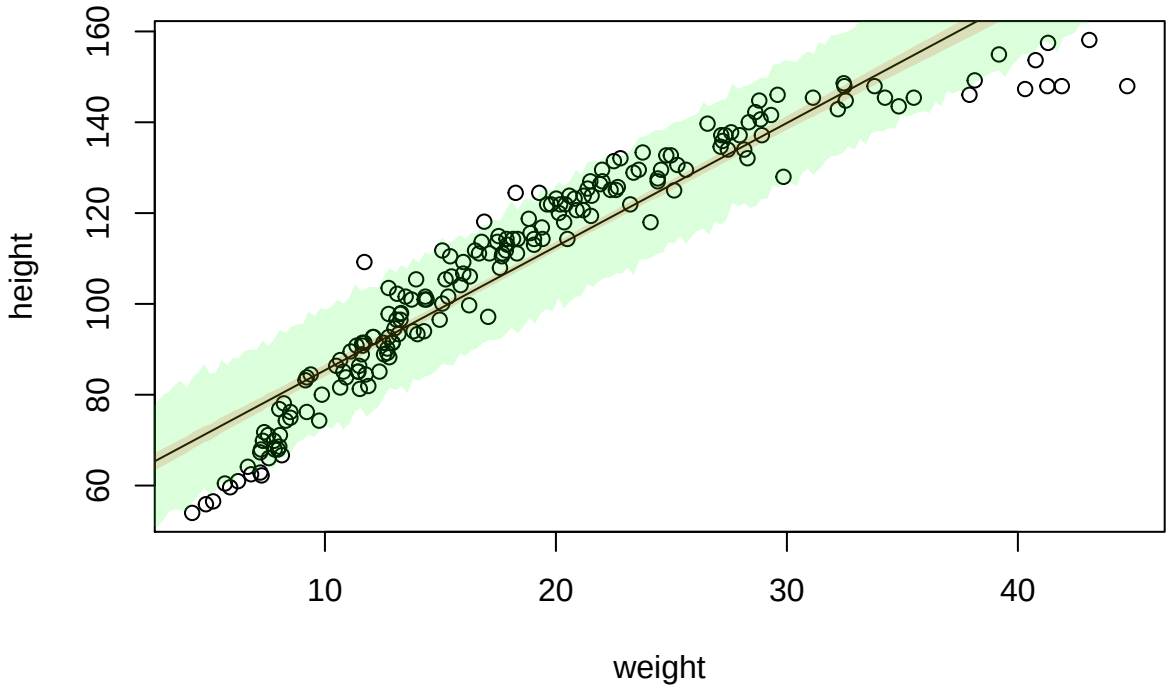
```
weight_seq = seq(0, 50, length.out = 200)

child_mu = link(child_model, data = list(weight = weight_seq))
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```
child_mu_mean = apply(child_mu, 2, mean)
child_mu_hpdi = apply(child_mu, 2, HPDI, prob = 0.89)

child_pred_height = sim(child_model, data = list(weight = weight_seq))
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```
child_pred_height_hpdi = apply(child_pred_height, 2, HPDI, prob = 0.89)

plot(height ~ weight, data = children)
lines(weight_seq, child_mu_mean)
shade(child_mu_hpdi, weight_seq, col = col.alpha("red", 0.15))
shade(child_pred_height_hpdi, weight_seq, col = col.alpha("green", 0.15))
```



The red/brow shade represents the 89% HPDI for the mean, and the green shade represents the 89% for the predicted height.

(c) The model overestimates the height of light and heavy children (<10kg and >40kg), and it seems also to underestimate a bit the height of children around 20kg. It seems that adding a quadratic term to the model would improve its fit to the data. Meaning that height wouldn't vary linearly with weight.

4H3

```
d$log.weight = log(d$weight)

log_model = map(
  alist(
    height ~ dnorm(mu, sigma),
    mu <- alpha + beta * log.weight,
    alpha ~ dnorm(178, 100),
    beta ~ dnorm(0, 100),
    sigma ~ dunif(0, 50)
  ),
  data = d
)

precis(log_model)
```

```
##         Mean StdDev   5.5%  94.5%
## alpha -23.78   1.34 -25.92 -21.65
## beta   47.08   0.38  46.46  47.69
## sigma   5.13   0.16   4.89   5.38
```

(a) An increase in 1 log weight unit (= 2.7kg) increase the height by 47cm, and the theoretical height of an individual weighing 1kg would be −23.78 cm.

(b)

```r
seq_weight = seq(1, 70, by = 0.5)

seq_height = link(log_model, data = list(log.weight = log(seq_weight)))
```

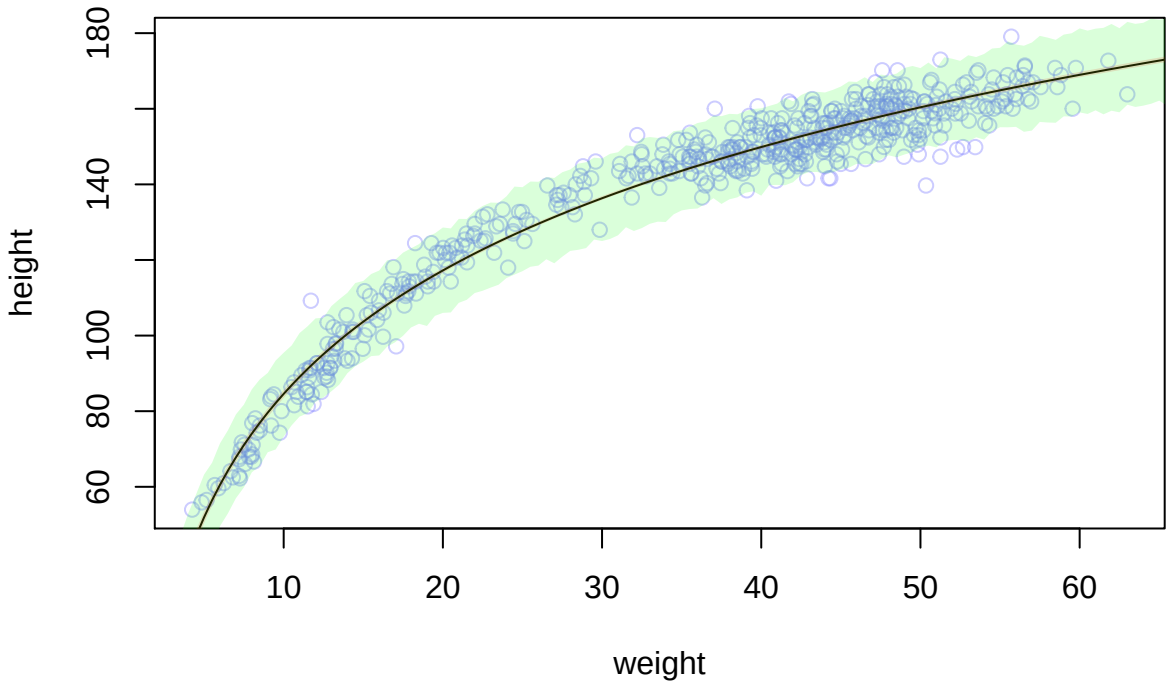
```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```r
mean_height = apply(seq_height, 2, mean)
mean_hpdi = apply(seq_height, 2, HPDI, prob = 0.97)

height_pred = sim(log_model, data = list(log.weight = log(seq_weight)))
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```r
height_pred_hpdi = apply(height_pred, 2, HPDI, prob = 0.97)
```

```r
plot(height ~ weight, data = Howell1, col = col.alpha(rangi2, 0.4))
lines(seq_weight, mean_height)
shade(mean_hpdi, seq_weight, col = col.alpha("red", 0.15))
shade(height_pred_hpdi, seq_weight, col = col.alpha("green", 0.15))
```


height
60
80
100
140
180
10
20
30
40
50
60
weight