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,]
```

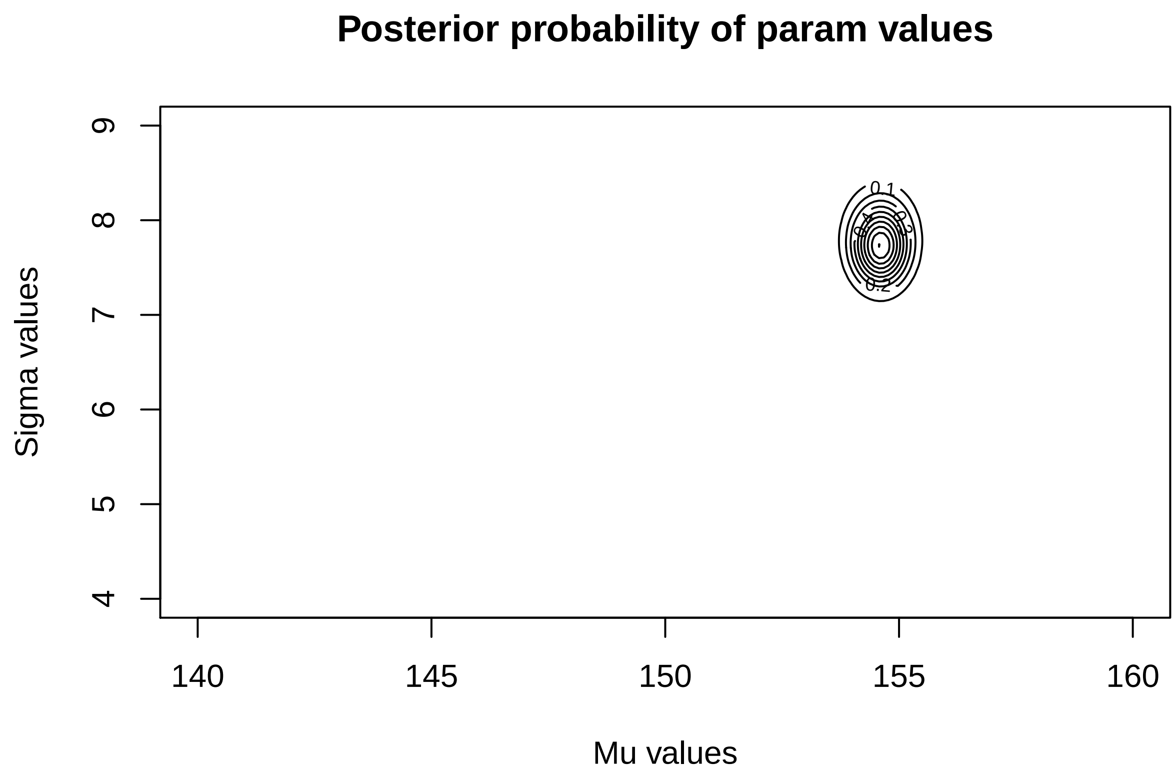
Model of height of adults:

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

Test of posterior distribution computation:

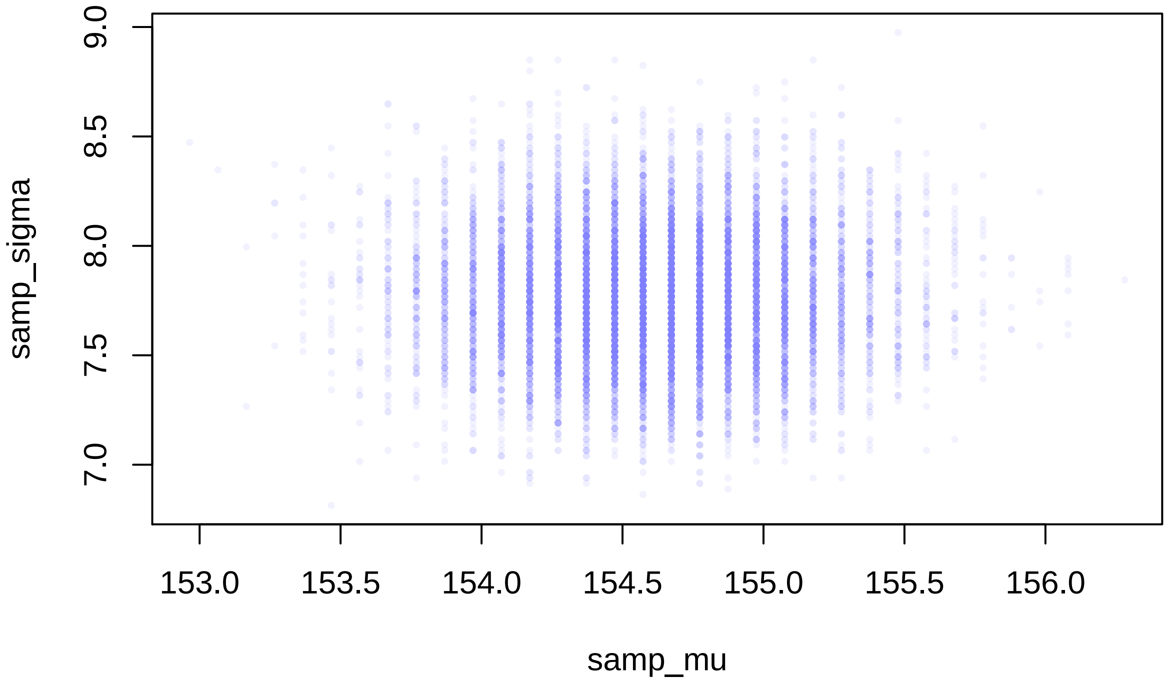
```
mu.list <- seq( from=140, to=160 , length.out=200 )
sigma.list <- seq( from=4 , to=9 , length.out=200 )
post <- expand.grid( mu=mu.list , sigma=sigma.list )
post$LL <- sapply( 1:nrow(post) , function(i) sum( dnorm(
                d2$height ,
                mean=post$mu[i] ,
                sd=post$sigma[i] ,
                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) )
```

```r
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:

```r
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)
```

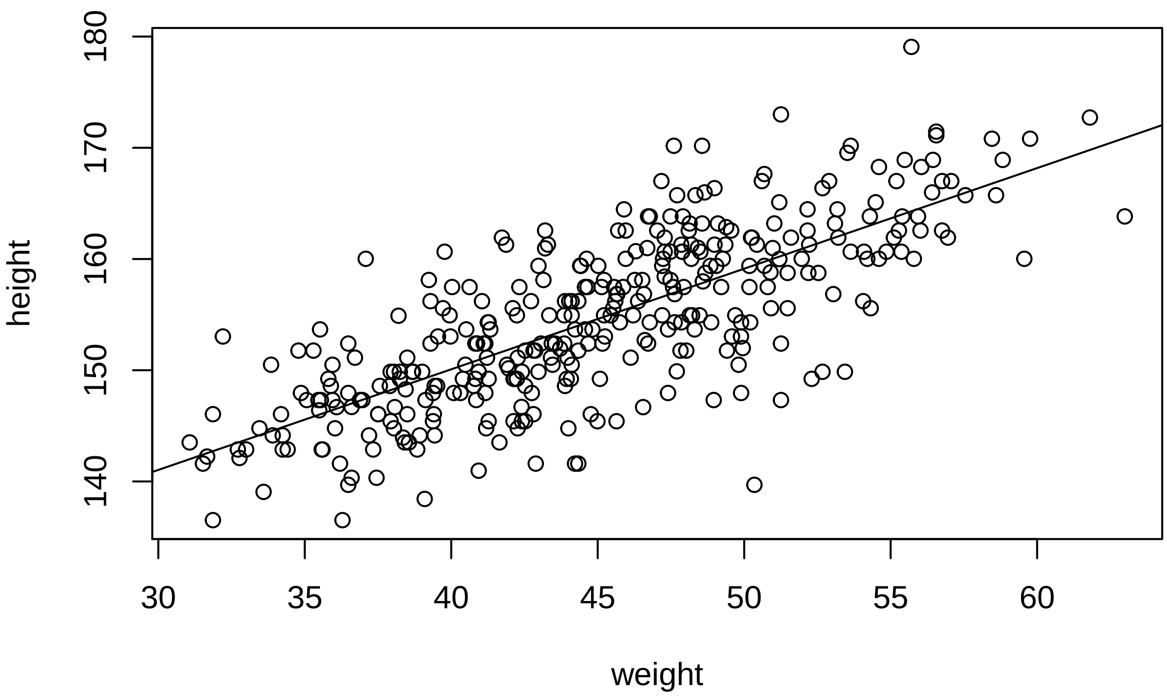
```
##          Mean StdDev   5.5%  94.5% alpha  beta sigma
```

```
## 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:

```r
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
)
```

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




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

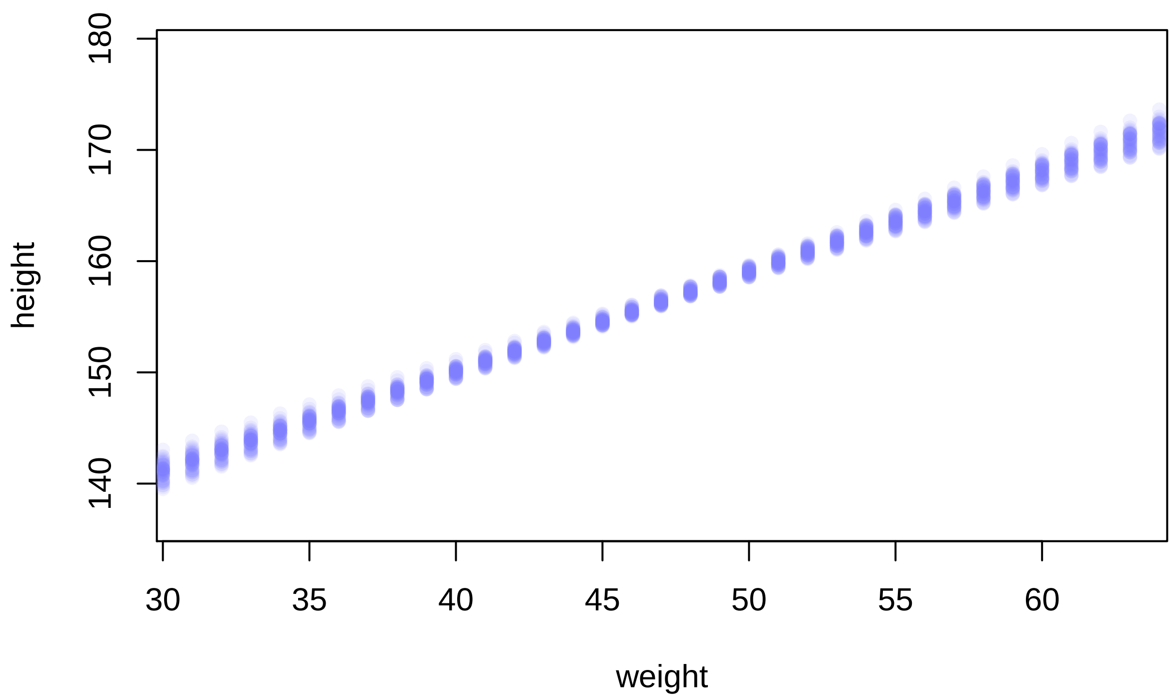
```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
```

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

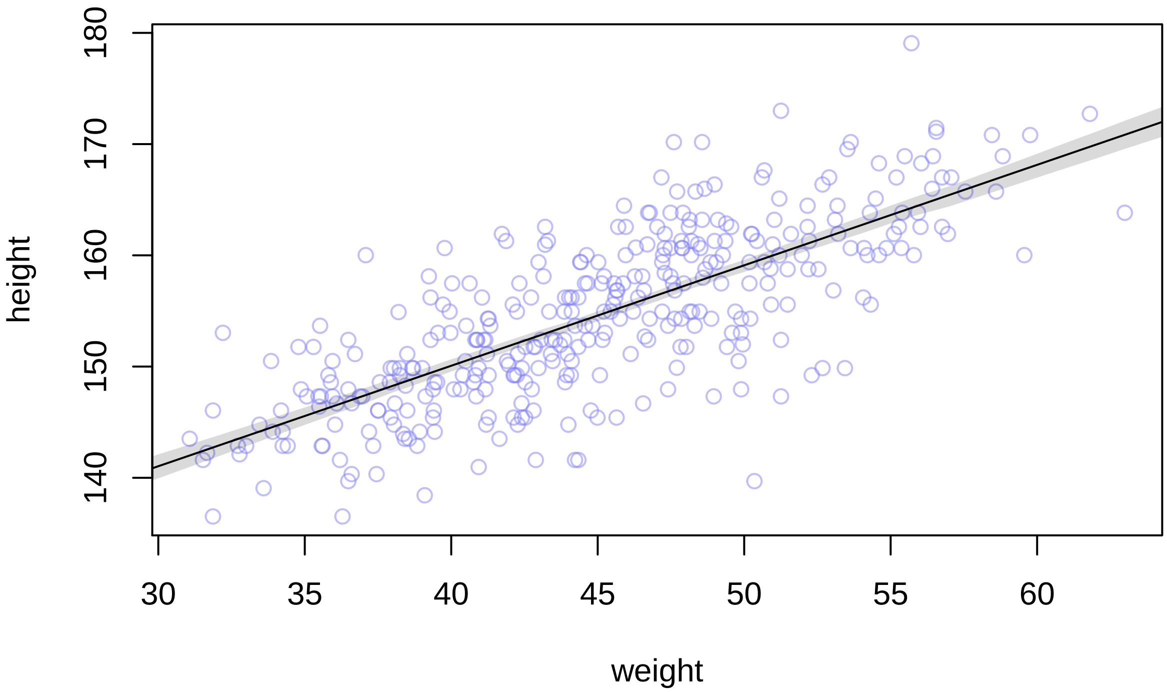
```r
str(mu)
```

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

```r
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))
  }
```




```r
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)
```

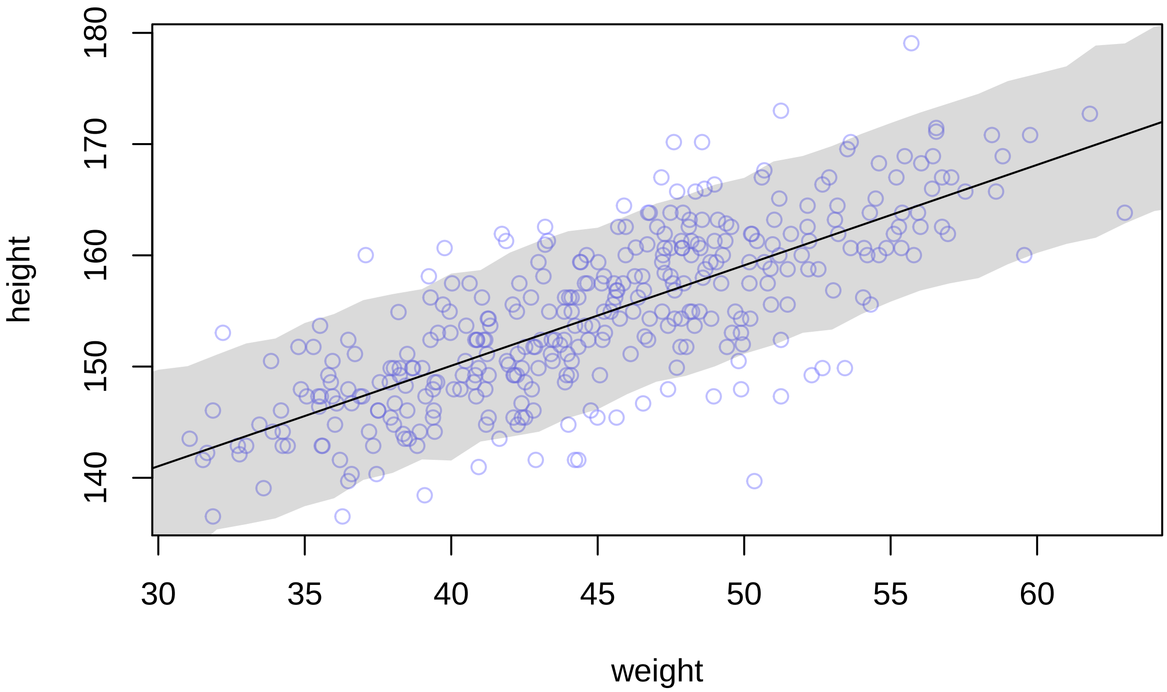


Shading indicates the 89% highest posterior density of interval of prediction of mean μ. Not the confidence interval of the prediction of height using weight exactly.

```
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 ]
```

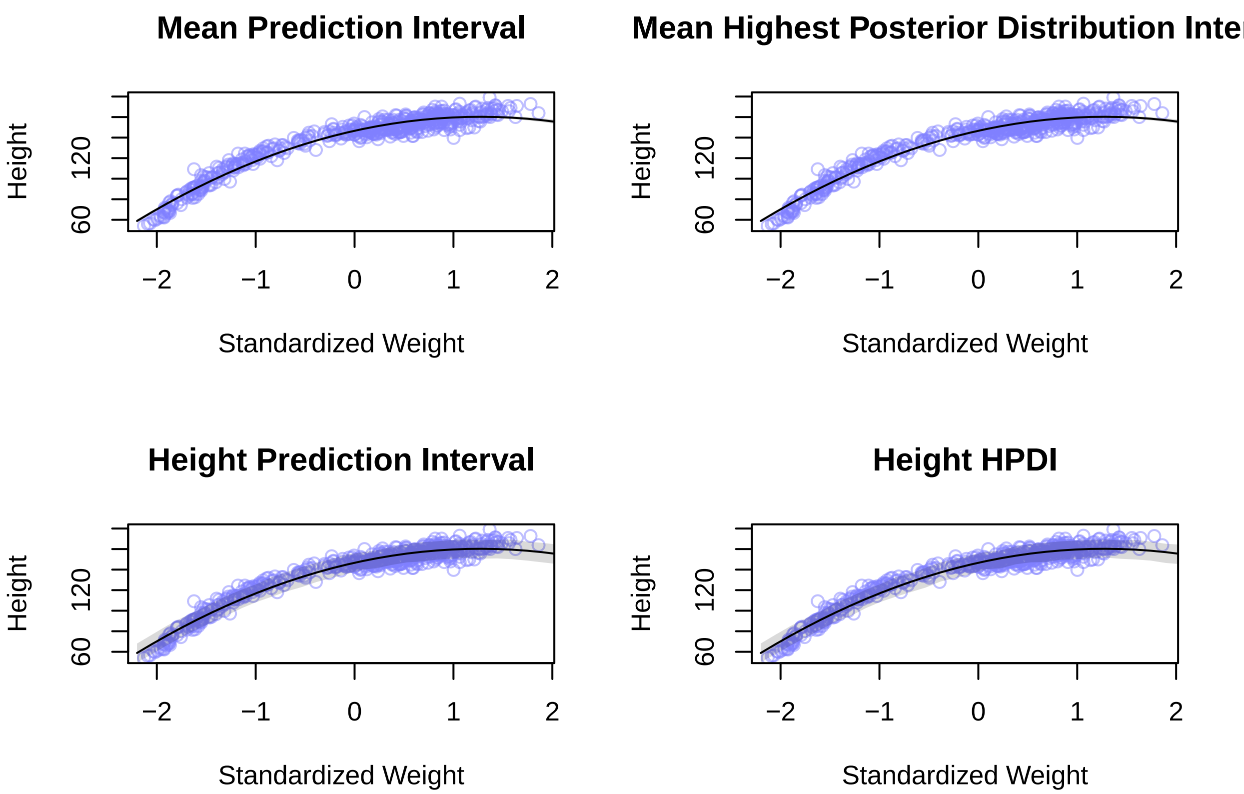
```
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```r
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 ]
```

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

```r
# 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)
```




```r
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

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