

# Statistical Rethinking Chapter 4 problems

\_\_\_Name: Ruijuan Li

**For 03/17/16**

**4E1**

the first one is the likelihood

**4E2**

two, mu and sigma

**4E3**

this formula should look very similar to the one on P83

**4M1 for the model definition below, simulate observed heights from the prior.**

```
library(rethinking)

## Loading required package: rstan

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.2.4

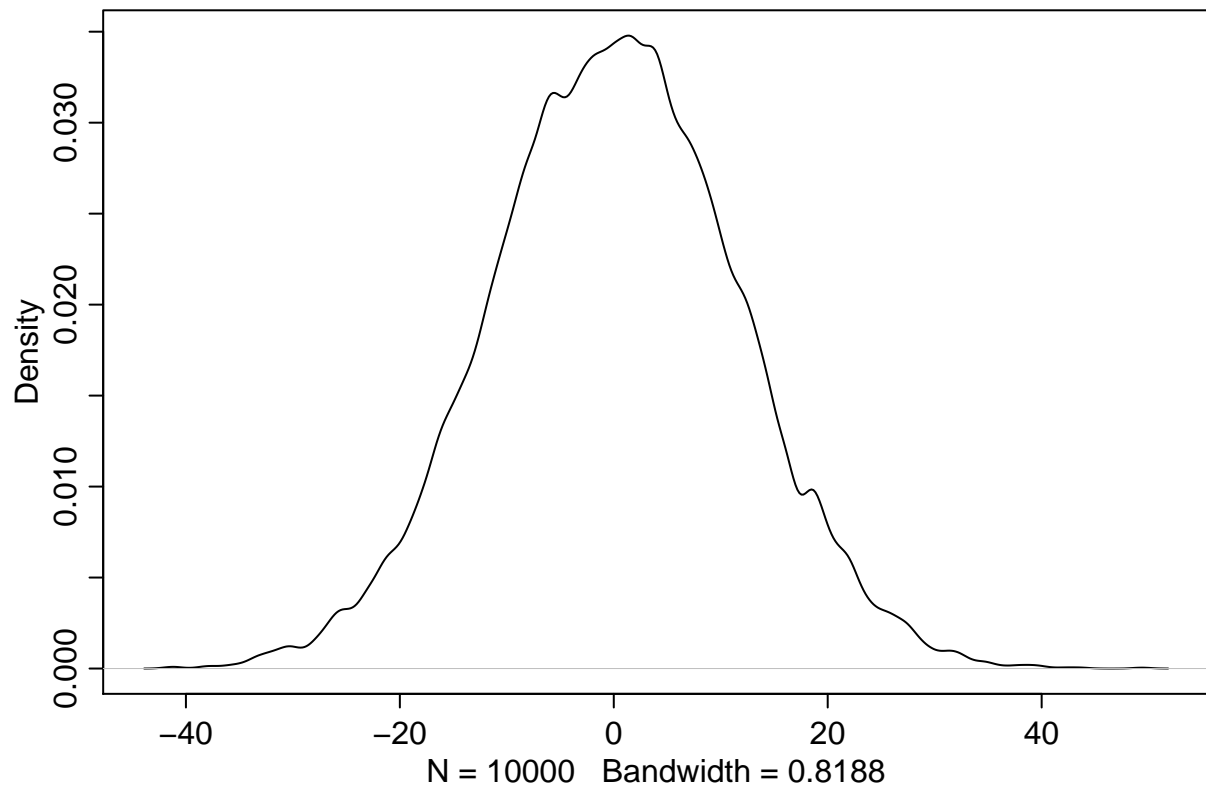
## rstan (Version 2.9.0-3, packaged: 2016-02-11 15:54:41 UTC, GitRev: 05c3d0058b6a)

## 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 1.58)

sample_mu <- rnorm(1e4, 0, 10)
sample_sigma <- runif(1e4, 0, 10)
prior <- rnorm(1e4, sample_mu, sample_sigma)
dens(prior)
```



4M2 translate the model into a map formula

```
data("Howell1")
d <- Howell1
d <- d[d$age >=18,]
m <- map(
  alist(
    height ~ dnorm(mu, sigma),
    mu ~ dnorm(0, 10),
    sigma ~ dunif(0, 10)
  ),
  data = d # prblem...
)
```

For 03/24/16

4E4

second line

4E5

three, alpha, beta, and sigma

### 4M3

```
yi ~ Normal(mui, sigma)
mui = a+b*xi
a ~ Normal(0, 50)
b ~ Normal(0, 10)
sigma ~ Uniform(0, 50)
```

### 4M4

```
hi ~ Normal(mui, sigma1)
mui = a + b*xi
a ~ Normal(mu2, sigma2)
b ~ Normal(mu3, sigma3)
sigma ~ Uniform(n, m)
```

### 4M5

```
mu2 = 120
b > 0
```

### 4M6

```
sigma <= 64
sigma2 <= 64
```

## For 03/31/16

### 4H1 predicted height or mean of height???

```
# import data
library(rethinking)
data("Howell1")
d <- Howell1

# model fit, normal linear regression

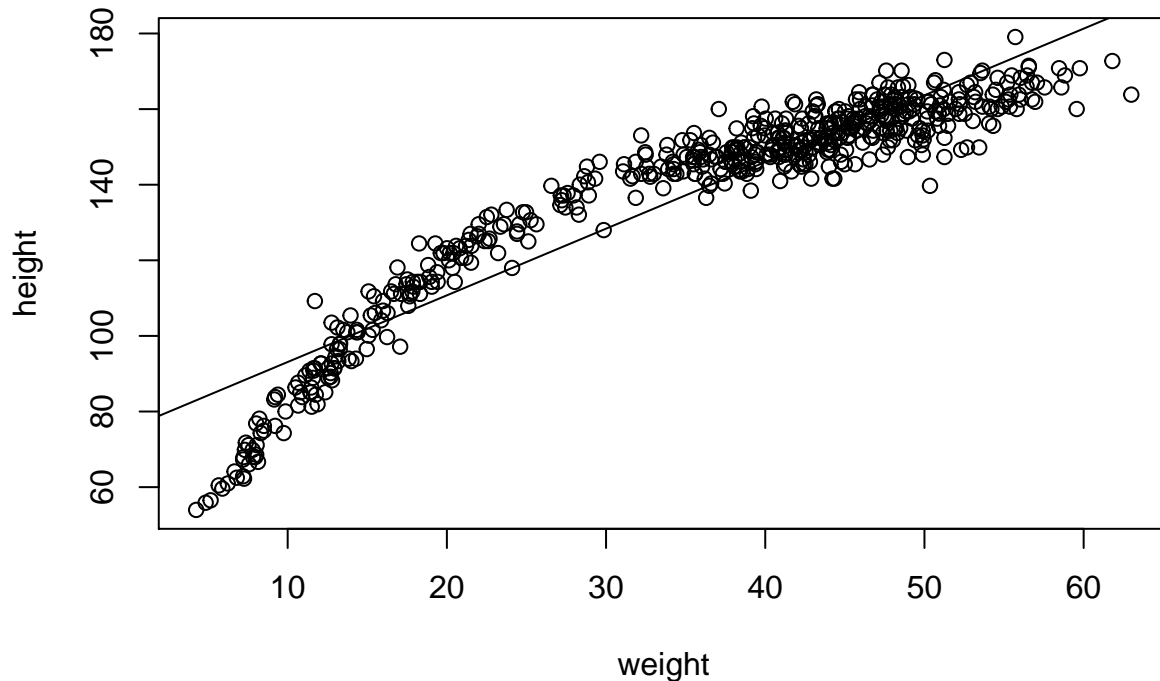
model1 <- map(
  alist(
    height ~ dnorm(mu, sigma),
    mu <- a + b*weight,
    a ~ dnorm(178, 100),
    b ~ dnorm(0, 10),
    sigma ~ dunif(0, 50)
```

```

),
data=d)

# plot the model
plot(height ~ weight, data=d)
abline(a=coef(model1)["a"], b = coef(model1)["b"])

```



```

# extract samples to form a posterior
post <- extract.samples(model1)
colnames(post)

```

```
## [1] "a"      "b"      "sigma"
```

```

# predict average height values
mu.link <- function(weight) post$a + post$b*weight
weight.seq <- c(46.95, 43.72, 64.78, 32.59, 54.63)
mu <- sapply(weight.seq, mu.link)
mu.mean <- apply(mu, 2, mean)
mu.mean

```

```
## [1] 158.2656 152.5699 189.7069 132.9433 171.8085
```

```

mu.HPDI <- apply(mu, 2, HPDI, prob=0.89)
mu.HPDI

```

```

##           [,1]      [,2]      [,3]      [,4]      [,5]
## |0.89| 157.4682 151.8393 188.2628 132.3282 170.7646
## 0.89| 159.0689 153.2819 191.0779 133.6071 172.8326

```

```

# predicted height: 158.2752 152.5763 189.7340 132.9389 171.8256
# 89% HPDI interval
# [,1]      [,2]      [,3]      [,4]      [,5]
# |0.89 157.4944 151.8958 188.3771 132.2422 170.7728
# 0.89| 159.1143 153.3666 191.2324 133.5591 172.8721

```

## 4H2

```

# get data
d2 <- d[d$age < 18, ]
nrow(d2)

```

```
## [1] 192
```

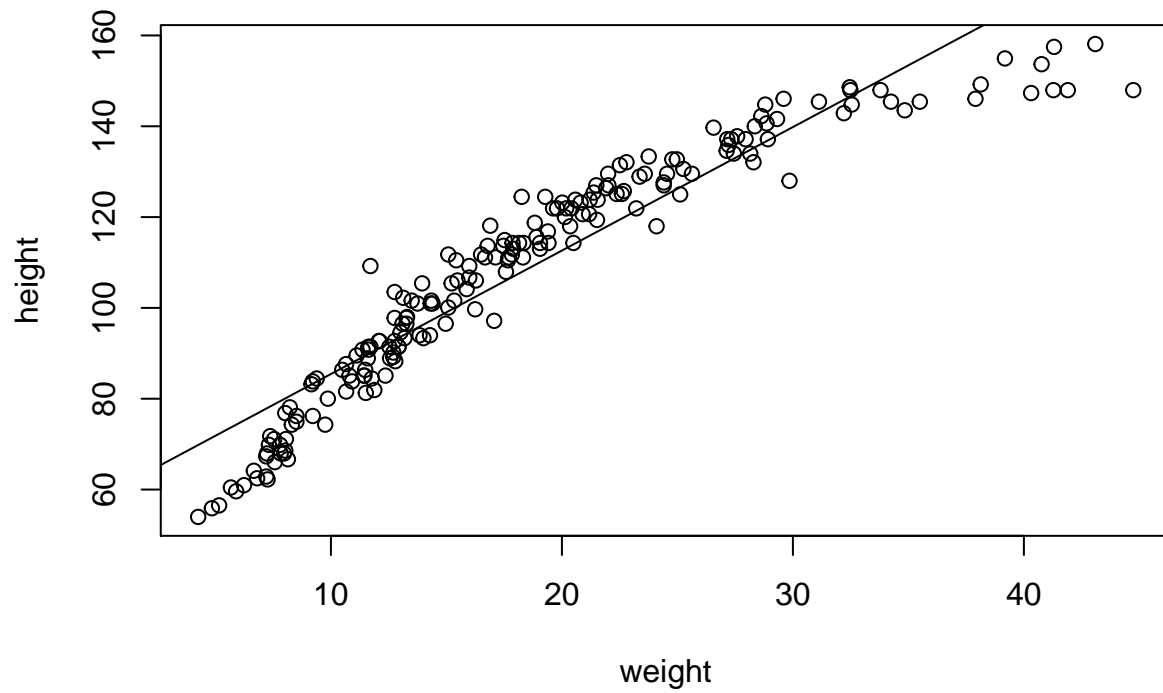
(a)

```

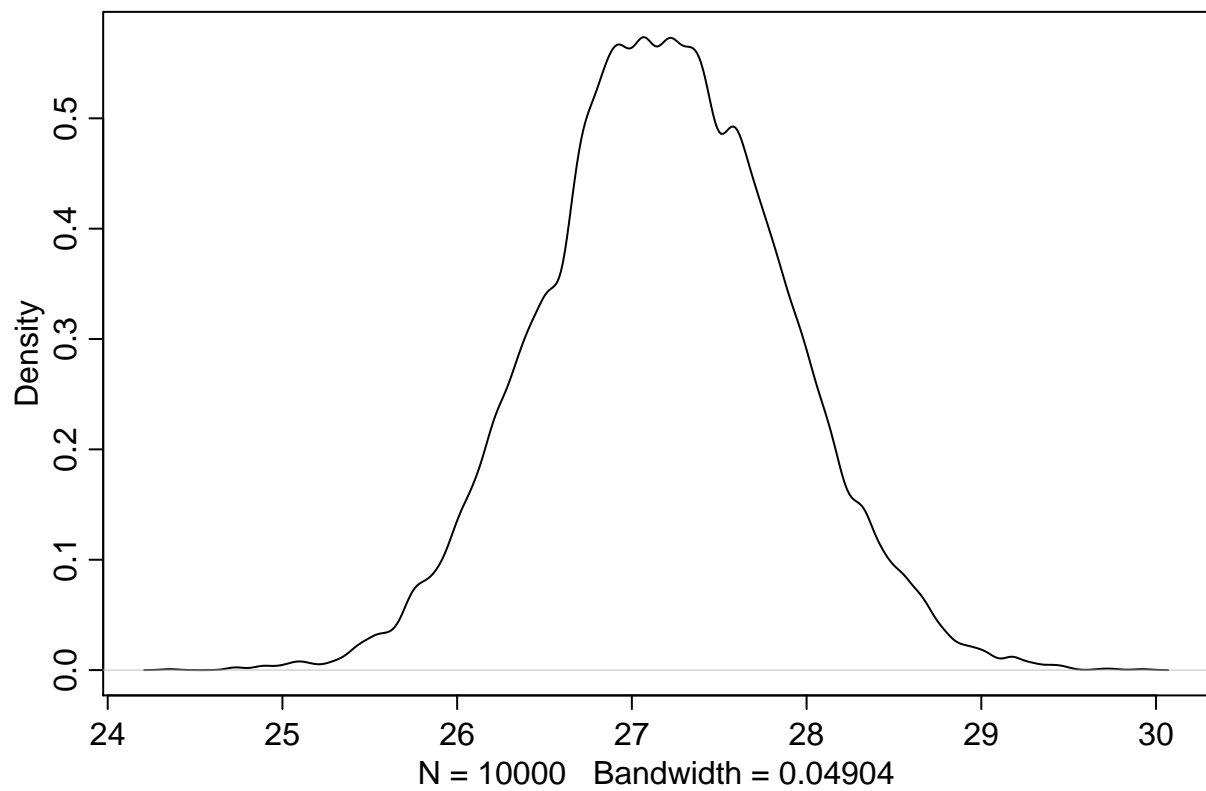
# fit a linear regression model
model2 <- map(
  alist(
    height ~ dnorm(mu, sigma),
    mu <- a + b*weight,
    a ~ dnorm(138, 100),
    b ~ dnorm(0, 10),
    sigma ~ dunif(0, 50)
  ),
  data=d2)

# plot
plot(height ~ weight, data=d2)
abline(a=coef(model2)["a"], b = coef(model2)["b"])

```



```
# extract slope values b
post2 <- extract.samples(model2)
dens(post2$b*10)
```



```
HPDI(post2$b*10)
```

```
##      |0.89      0.89|  
## 26.00450 28.19752
```

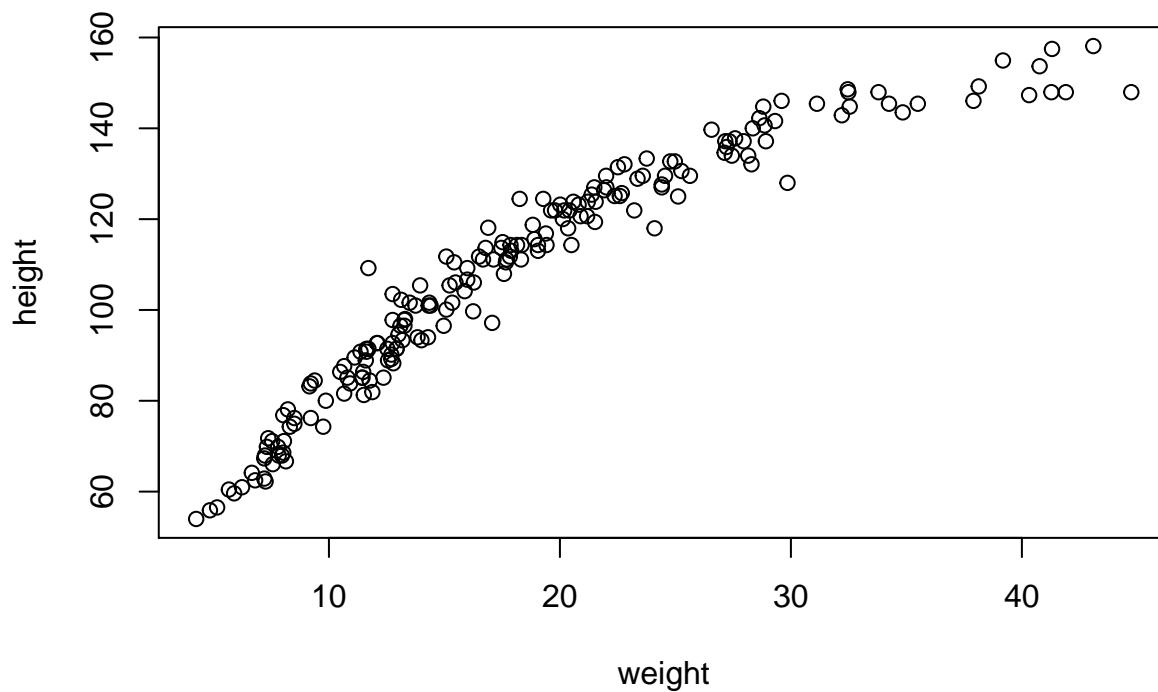
```
mean(post2$b)*10
```

```
## [1] 27.18334
```

```
# [1] 27.18586  
# |0.89      0.89|  
# 26.05893 28.25237
```

(b)

```
plot(height ~ weight, data=d2)
```



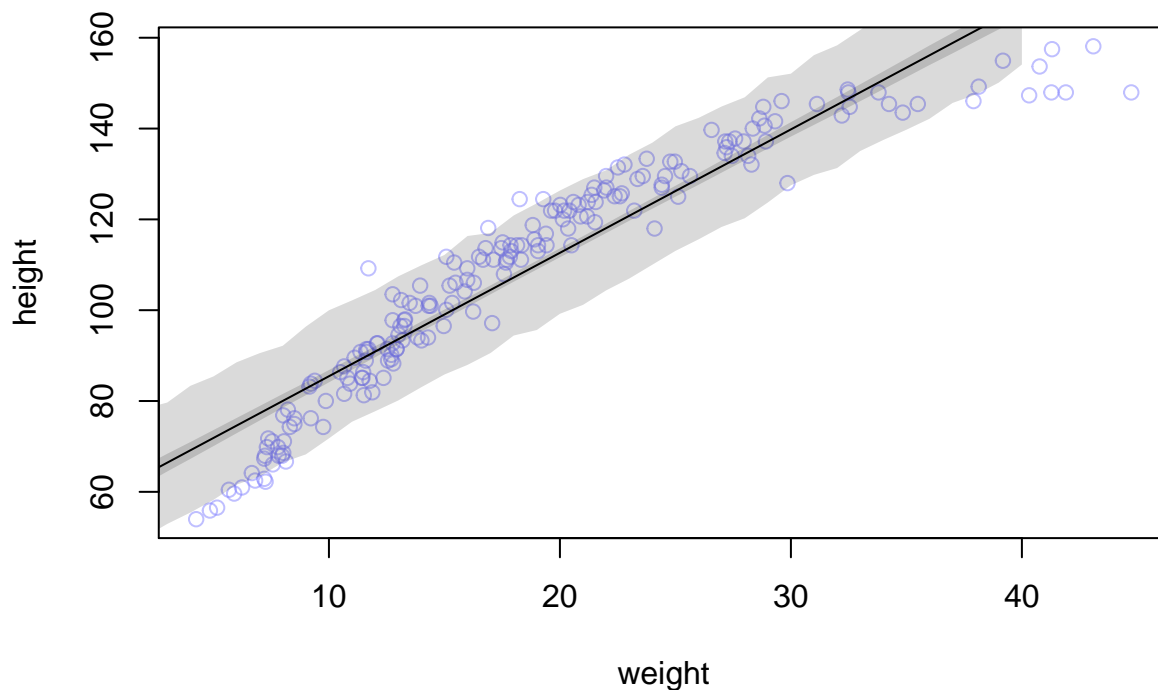
```
# use link to generate distribution of posterior values for mu  
mu.link <- function(weight) post2$a + post2$b*weight  
weight.seq <- seq(0, 40, by=1)  
mu <- sapply(weight.seq, mu.link)  
mu.mean <- apply(mu, 2, mean)  
mu.HPDI <- apply(mu, 2, HPDI, prob=0.89)  
  
# plot the distribution, the predicted mean, and the 89% interval  
plot(height ~ weight, data=d2, col=col.alpha(rangi2, 0.5))
```

```
lines(weight.seq, mu.mean)
shade(mu.HPDI, weight.seq)

# 89% interval for the predicted heights
sim.height <- sim(model2, 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)
shade(height.PI, weight.seq)
```



(c)

the model doesn't fit the current data very well, because it seems children with weight below 10kg and above 35kg had a slower growth rate as compared to children with weight between 10 to 35. While I made the assumption of steady growth rate, a polynomial regression with a squared curve may fit the current data better and can possibly serve as a better model.



### 4H3

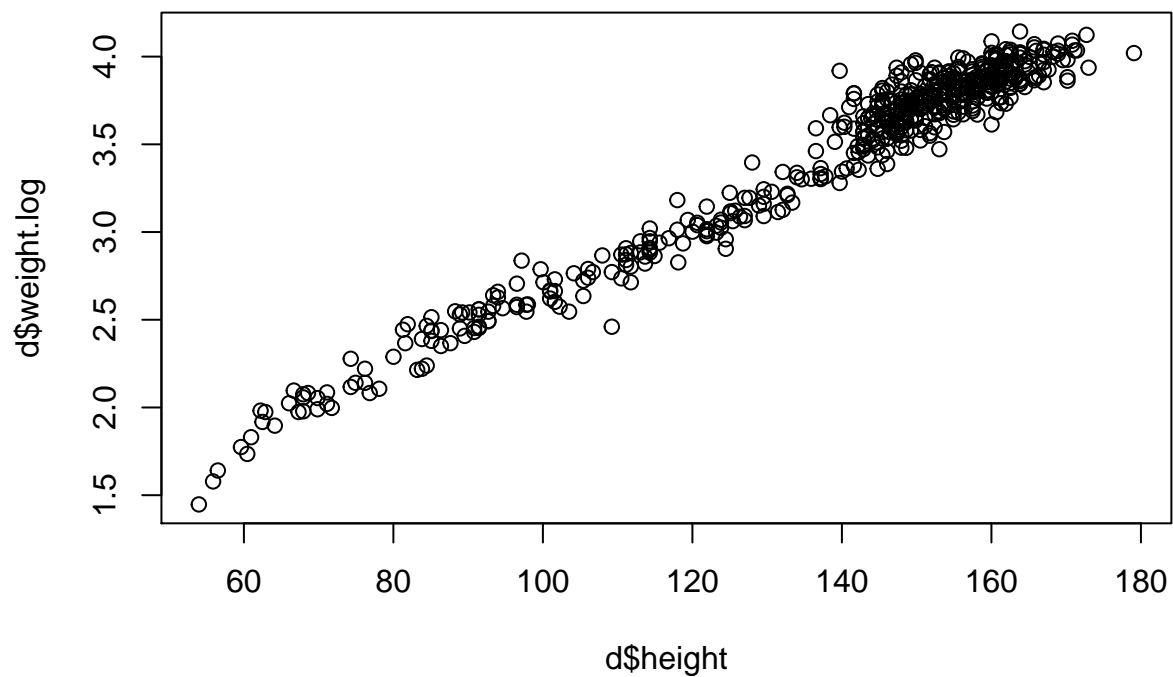
(a)

```
# import data
d <- Howell1
nrow(d)

## [1] 544

# fit model
d$weight.log <- log(d$weight) # cannot standardize, produce 0, prohibit log transformation
model3 <- map(
  alist(
    height ~ dnorm(mu, sigma),
    mu <- a + b * weight.log,
    a ~ dnorm(178, 100),
    b ~ dnorm(0, 100),
    sigma ~ dunif(0, 50)
  ),
  data = d)

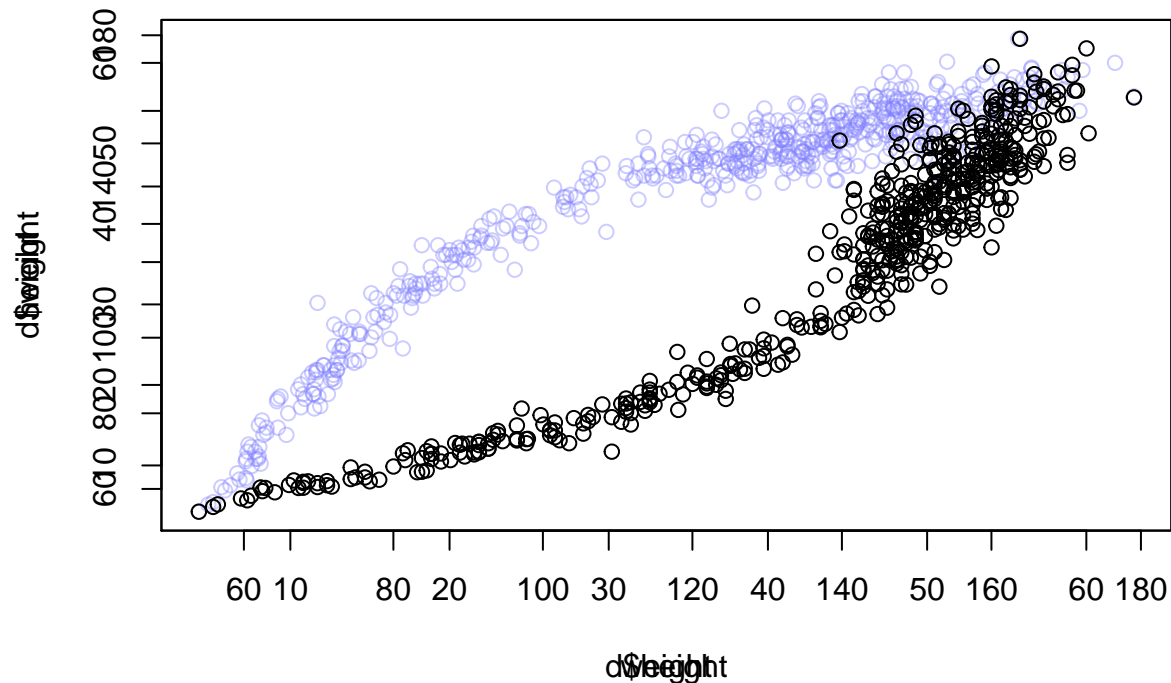
plot(d$height, d$weight.log)
```



(b)

```
plot(height ~ weight, data=Howell1,
col=col.alpha(rangi2, 0.4))

# the posterior from (a)
par(new=T)
plot(d$height, d$weight)
```



```
# one way for the rest #####
# use link to generate distribution of posterior values for mu
post3 <- extract.samples(model3)
mu.link <- function(weight) post3$a + post3$b*log(weight)
weight.seq <- seq(25, 70, by=1)
mu <- sapply(weight.seq, mu.link)
mu.mean <- apply(mu, 2, mean)
mu.HPDI <- apply(mu, 2, HPDI, prob=0.89)

# plot the distribution, the predicted mean, and the 89% interval
lines(weight.seq, mu.mean)
shade(mu.HPDI, weight.seq)

# 89% interval for the predicted heights
sim.height <- sim(model3, data = list(weight=weight.seq))
height.PI <- apply(sim.height, 2, PI, prob=0.89)
shade(height.PI, weight.seq)

### a second way for the rest #####
weight.seq <- seq(25, 70, by=1)
pred_dat <- list(weight = weight.seq, weight.log=log(weight.seq))
mu <- link(model3, data = pred_dat)
```

```
mu.mean <- apply(mu, 2, mean)
mu.PI <- apply(mu, 2, PI, prob=0.89)
sim.height <- sim(model3, data = pred_dat)
height.PI <- apply(sim.height, 2, PI, prob=0.89)

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