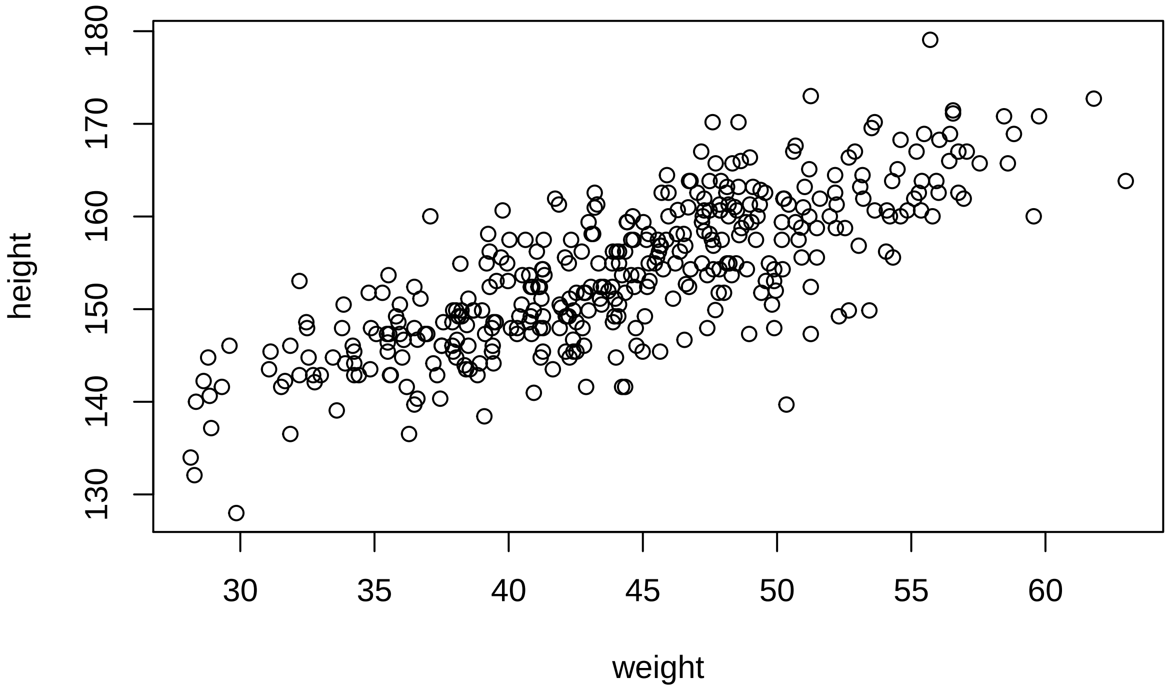
Statistical Rethinking Week 2

Jeff Shen

15 August, 2020

Problem 1

```
d2 <- d[d$weight > 28,]
plot(height ~ weight, data=d2)
```

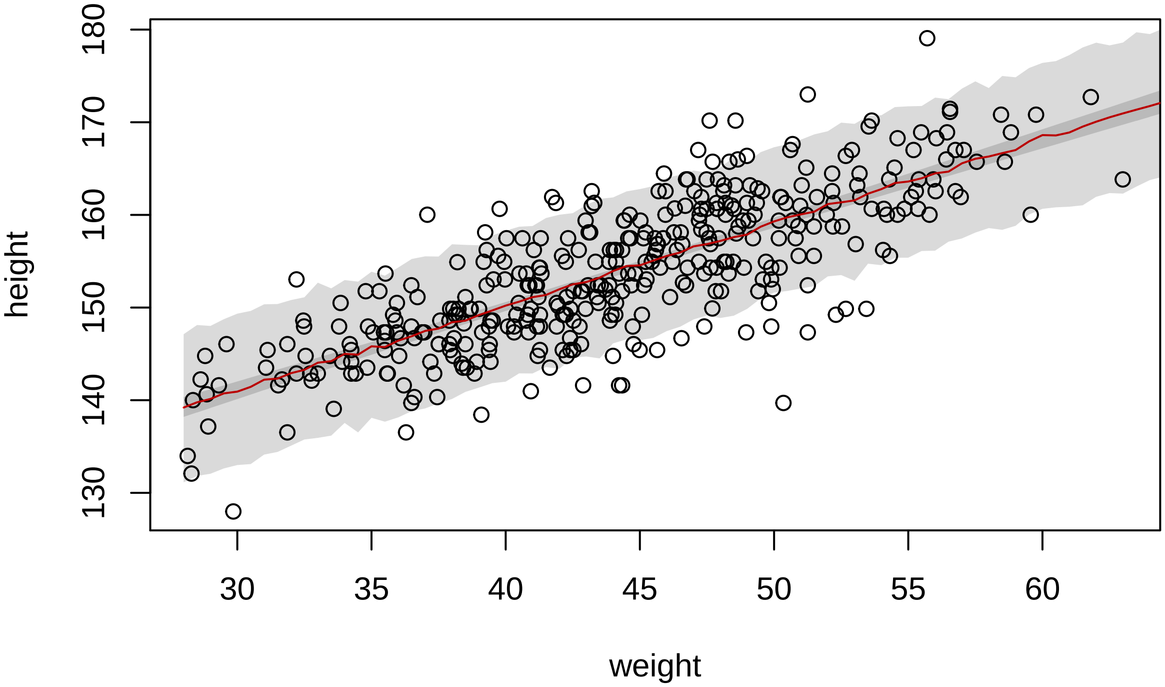


```
mdl <- quap(
    alist(
        height ~ dnorm(mu, sigma),
        mu <- a + b1 * (weight - mean(d$weight)),
        a ~ dnorm(160, 20),
        b1 ~ dlnorm(0, 1),
        sigma ~ dlnorm(3, 1)
    ), data=d2
)

precis(mdl)
```

```
##              mean        sd        5.5%       94.5%
## a     146.1446318 0.4089522 145.4910471 146.7982165
## b1      0.9024625 0.0370427   0.8432611   0.9616639
## sigma   5.0328852 0.1822956   4.7415417   5.3242287
```

```r
weight.seq <- seq(28, 70, by=0.5)
post <- link(mdl, data = list(weight=weight.seq))
mu.mean <- apply(post, 2, mean)
mu.PI <- apply(post, 2, PI, prob=0.89)
height.sim <- sim(mdl, data = list(weight=weight.seq))
height.mean <- apply(height.sim, 2, mean)
height.PI <- apply(height.sim, 2, PI, prob=0.89)
```

```r
plot(height ~ weight, data=d2)
lines(weight.seq, height.mean, col='red')
shade(mu.PI, weight.seq)
shade(height.PI, weight.seq)
```

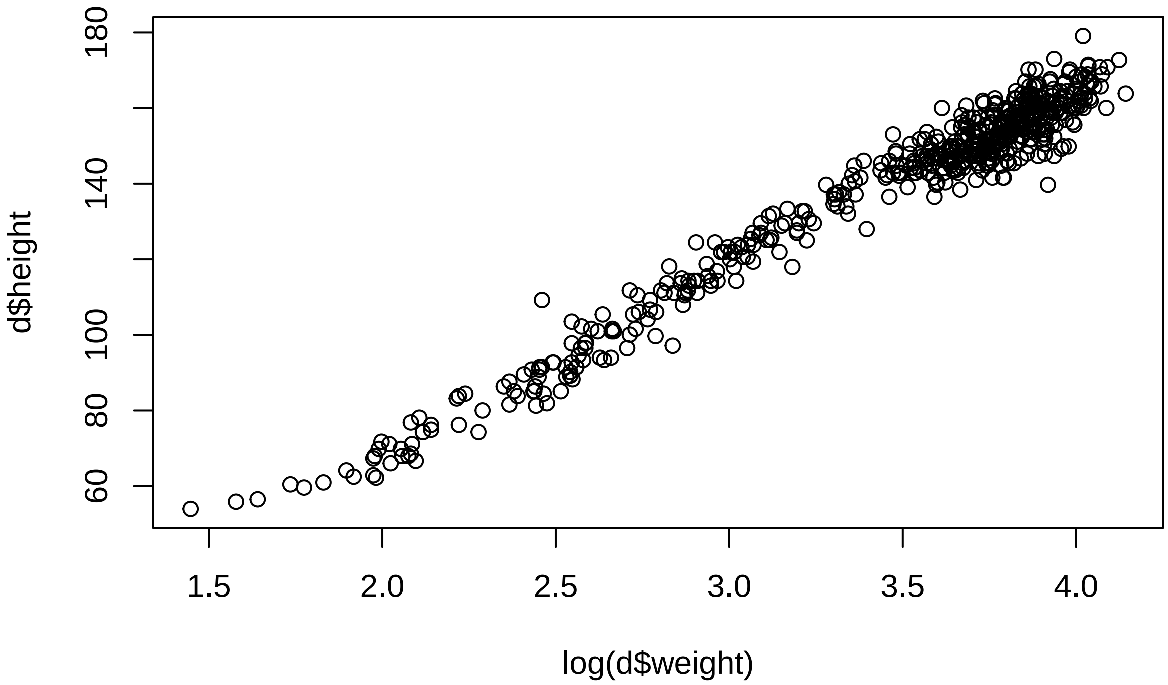


```r
tgt <- data.frame(weight=c(45, 40, 65, 31, 53))
tgt$height.mean <- sim(mdl, tgt) %>% apply(2, mean)
tgt$height.pi.lower <- apply(sim(mdl, tgt), 2, PI, prob=0.89)[1,]
tgt$height.pi.upper <- apply(sim(mdl, tgt), 2, PI, prob=0.89)[2,]
tgt
```

```
##   weight height.mean height.pi.lower height.pi.upper
## 1     45    154.5578        146.8321        163.1123
## 2     40    150.2055        142.1879        158.4008
## 3     65    172.5593        164.7544        180.7191
## 4     31    141.8766        133.1783        150.4153
## 5     53    161.8286        153.5497        169.1988
```

Problem 2

```r
plot(d$height ~ log(d$weight))
```

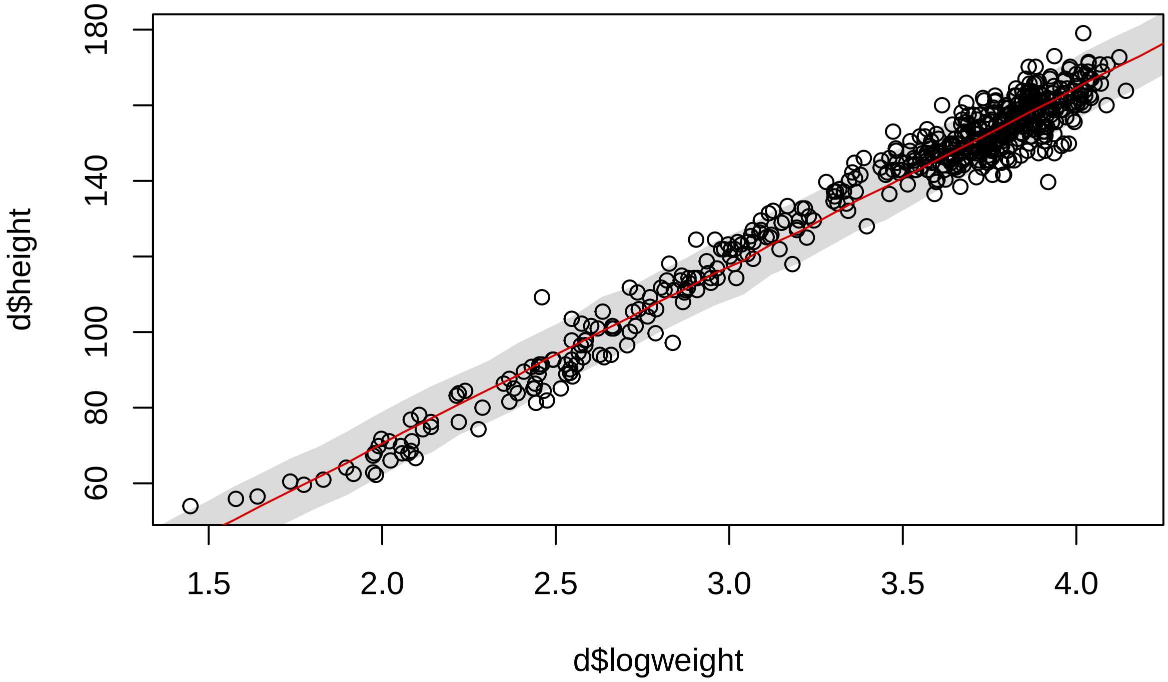


```r
d$logweight <- log(d$weight)
xbar <- mean(d$logweight)

mdl <- quap(
    alist(
        height ~ dnorm(mu, sigma),
        mu <- a + b * (logweight - xbar),
        a ~ dnorm(120, 10),
        b ~ dnorm(55, 10),
        sigma ~ dlnorm(1, 1)
    ), data=d
)

precis(mdl)
```

```
##              mean        sd      5.5%      94.5%
## a      138.254638 0.2197527 137.90343 138.605845
## b       47.096563 0.3817062  46.48652  47.706603
## sigma    5.126691 0.1551073   4.87880   5.374583
```

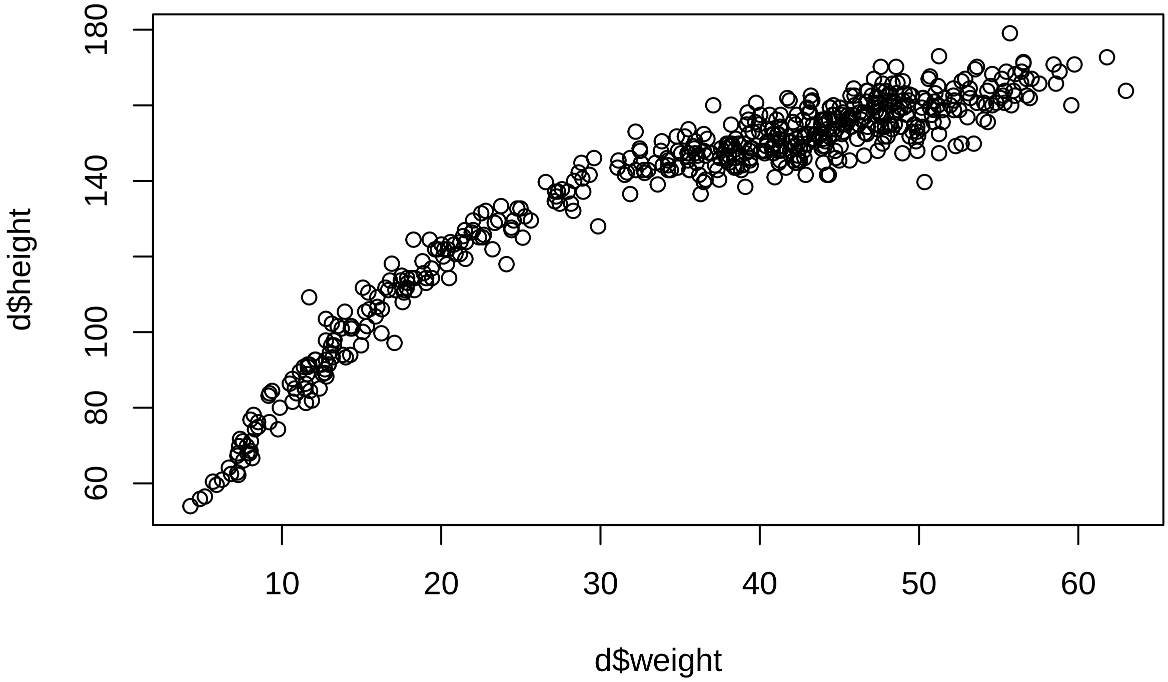
```r
weight.seq <- seq(from=floor(min(d$logweight)), to=ceiling(max(d$logweight)), length.out = 50)
post <- sim(mdl, list(logweight=weight.seq))
post.mean <- apply(post, 2, mean)
post.pi <- apply(post, 2, PI, prob=0.89)
```

```r
plot(d$height ~ d$logweight)
lines(weight.seq, post.mean, col='red')
shade(post.pi, weight.seq)
```


180
140
100
80
60
d$height
1.5
2.0
2.5
3.0
3.5
4.0
d$logweight

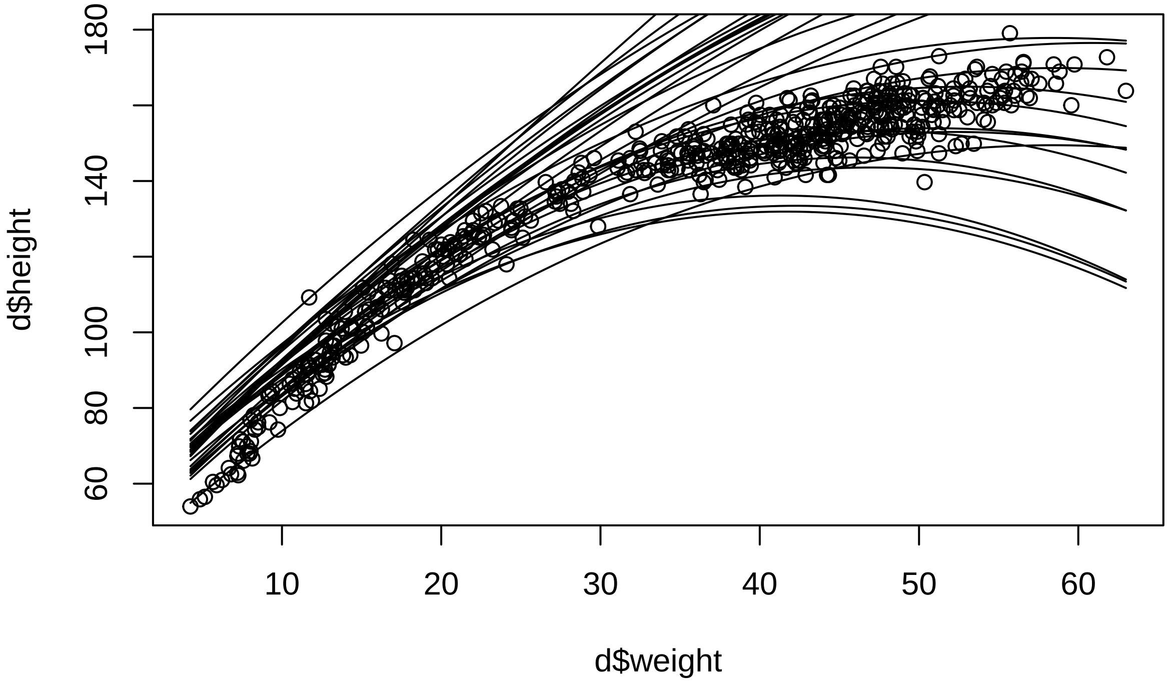
Problem 3

```
plot(d$height ~ d$weight)
```



```
priors <- data.frame(
    # height ~ dnorm(mu, sigma),
    # mu <- a + b1 * (weight - xbar) + b2 * (weight - xbar)^2,
    # a ~ dnorm(130, 20),
    # b1 ~ dnorm(0, 15),
    # b2 ~ dnorm(0, 15),
    # sigma ~ dlnorm(1, 1)
    a=rnorm(1e2, 50, 5),
    b1=rnorm(1e2, 4.2, 0.2),
    b2=runif(1e2, -0.05, -0.01)
)
```

```
plot(d$height ~ d$weight)
for (i in 1:30) {
    curve(priors$a[i] +
              priors$b1[i] * x +
              priors$b2[i] * x^2,
          from = min(d$weight),
          to = max(d$weight),
          add = TRUE)
}
```



Kind of ended up fitting the data by hand (exactly what I shouldn't be doing). Hard to modify the prior distributions especially for β_2 because it doesn't really have a physical meaning. Not really used to thinking of things scaling with the square of weight, so hard to estimate what it should be without looking at the data. Only know it's negative, have no idea what the scale should be.