Chapter 4: Geocentric Models

Quang Nguyen

10/3/2020

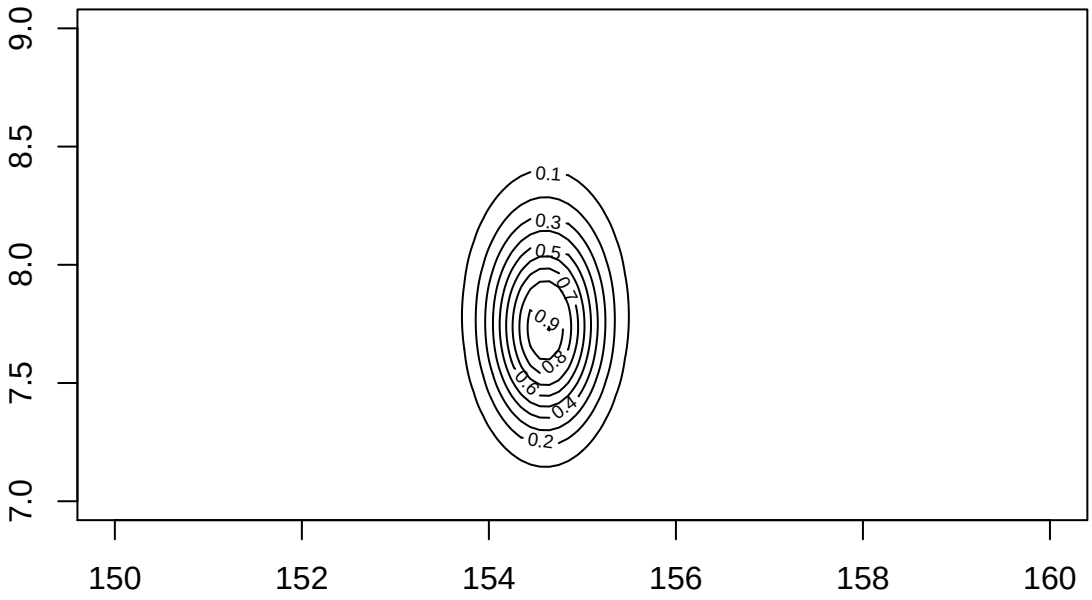
```
library(rethinking)
library(glue)
library(tidyverse)
```

```
data("Howell1")
d <- Howell1
d2 <- d %>% filter(age >= 18)
mu.list <- seq(from=150, to=160, length.out = 100)
sigma.list <- seq(from = 7, to = 9, length.out = 100)
post <- expand.grid(mu = mu.list, sigma = sigma.list)

# calculate likelihood from grid of values
post$LL <- sapply(1:nrow(post), function(x){
  sum(dnorm(d2$height, post$mu[x], post$sigma[x], log = TRUE))
})

# evaluate numerator as likelihood combined with prior
post$prod <- post$LL + dnorm(post$mu, 170, 20, TRUE) + dunif(post$sigma, 0, 50, TRUE)

# normalize by maximum
post$prob <- exp(post$prod - max(post$prod))

print("Contour plots")
```

Text-book code for Gaussian modelling of height.

```
## [1] "Contour plots"
```

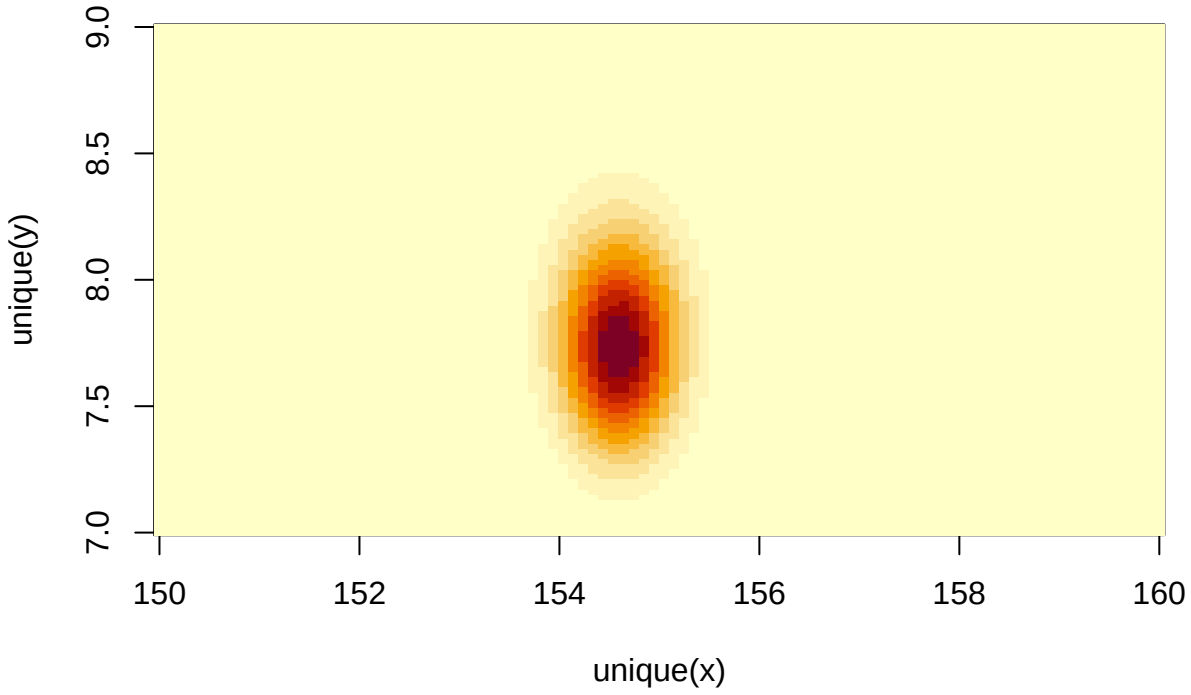
```
contour_xyz(post$mu, post$sigma, post$prob)
```




```
print("Heatmap")
```

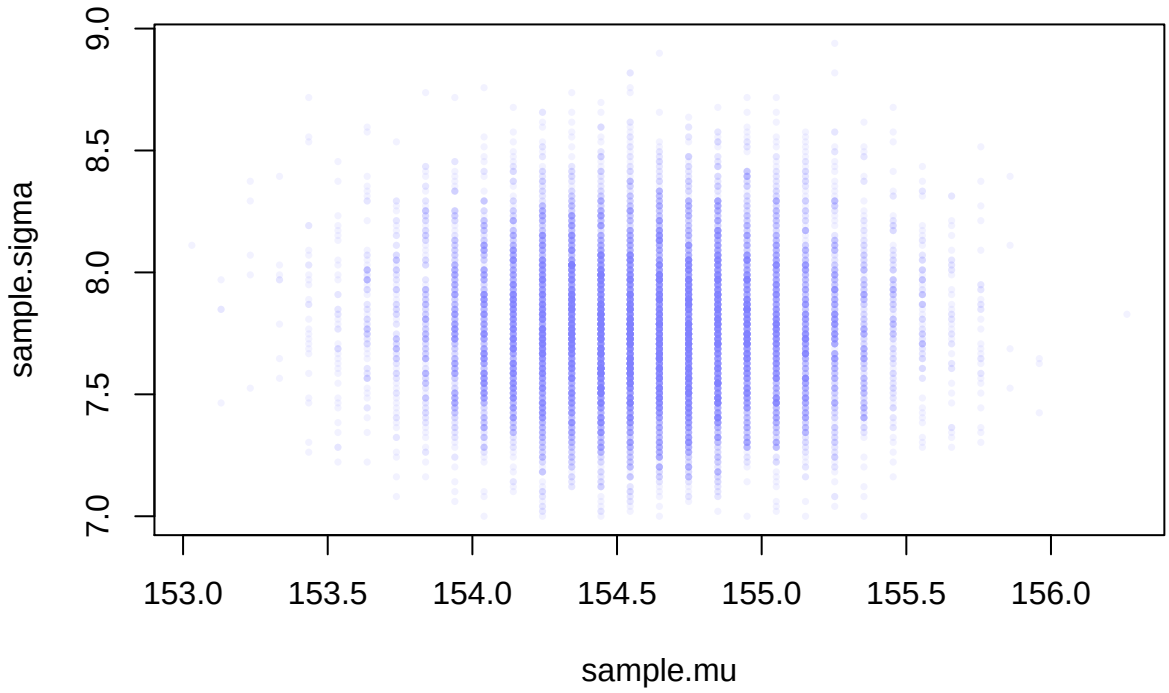
```
## [1] "Heatmap"
```

```
image_xyz(post$mu, post$sigma, post$prob)
```

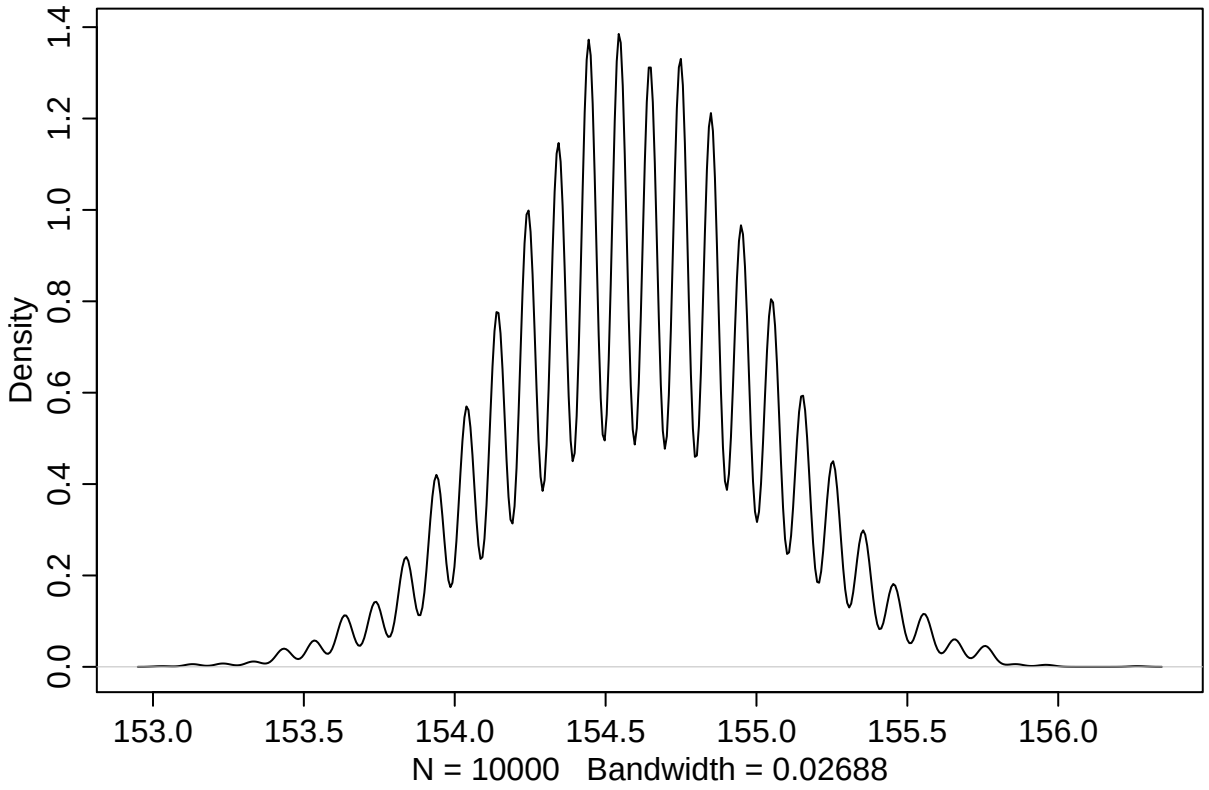


Sampling from the posterior

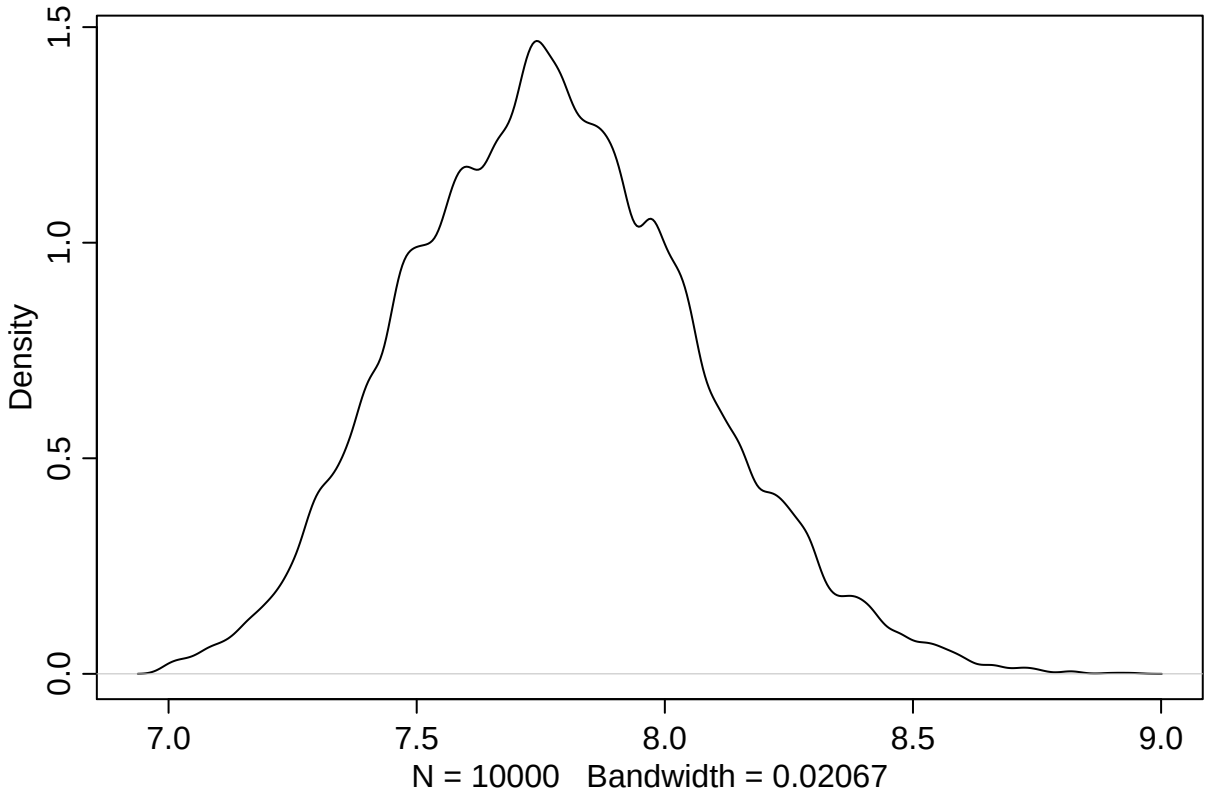
```
# sampling rows based on posterior
sample.rows <- sample(1:nrow(post), size = 1e4, replace = T, prob = post$prob)

# retrieve parameter values
sample.mu <- post$mu[sample.rows]
sample.sigma <- post$sigma[sample.rows]

plot(sample.mu, sample.sigma, cex = 0.5, pch = 16, col = col.alpha(rangi2,0.1))
```




```
# Marginal distribution
dens(sample.mu)
```




```
dens(sample.sigma)
```




```
PI(sample.mu)
```

```
##       5%      94%
## 153.9394 155.2525
```

```
PI(sample.sigma)
```

```
##       5%      94%
## 7.323232 8.252525
```

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

m4.1 <- quap(flist, data = d2)
precis(m4.1)
```

Using the quadratic approximation

```
##             mean        sd       5.5%      94.5%
## mu    154.605808 0.4118269 153.947629 155.263987
## sigma   7.728187 0.2910897   7.262969   8.193404
```

`quap` estimates posterior distribution by approximation. It needs a start point to start the gradient procedure, which will be chosen at random from the prior if start values are not specified. Use `alist` when defining a

list of formulas because alist does not evaluate terms in the list.

```
m4.2 <- quap(
  alist(
    height ~ dnorm(mu, sigma),
    mu ~ dnorm(178, 0.1),
    sigma ~ dunif(0,50)
  ), data = d2
)

precis(m4.2)
```

Using quadratic approximation with a more informative prior

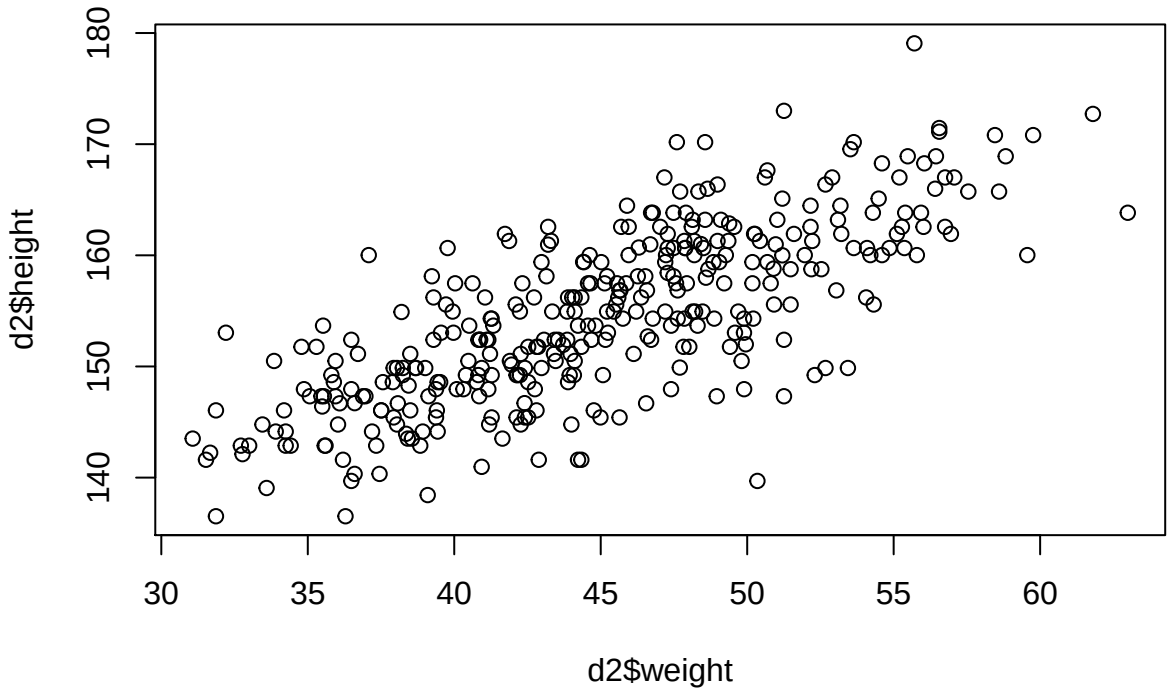
```
##            mean        sd      5.5%     94.5%
## mu    177.86378 0.1002354 177.70358 178.02397
## sigma  24.51822 0.9289839  23.03352  26.00291
```

```
post <- extract.samples(m4.1, n = 1e4)
precis(post)
```

Sampling from quap

```
##             mean        sd      5.5%      94.5%
## mu    154.609204 0.4134462 153.95264 155.272434
## sigma   7.725568 0.2914706   7.26359   8.194761
##                                                                                                  histogram
## mu                              <U+2581><U+2581><U+2581><U+2585><U+2587><U+2582><U+2581><U+2581>
## sigma <U+2581><U+2581><U+2581><U+2582><U+2585><U+2587><U+2587><U+2583><U+2581><U+2581><U+2581>
```

```
plot(d2$height ~ d2$weight)
```



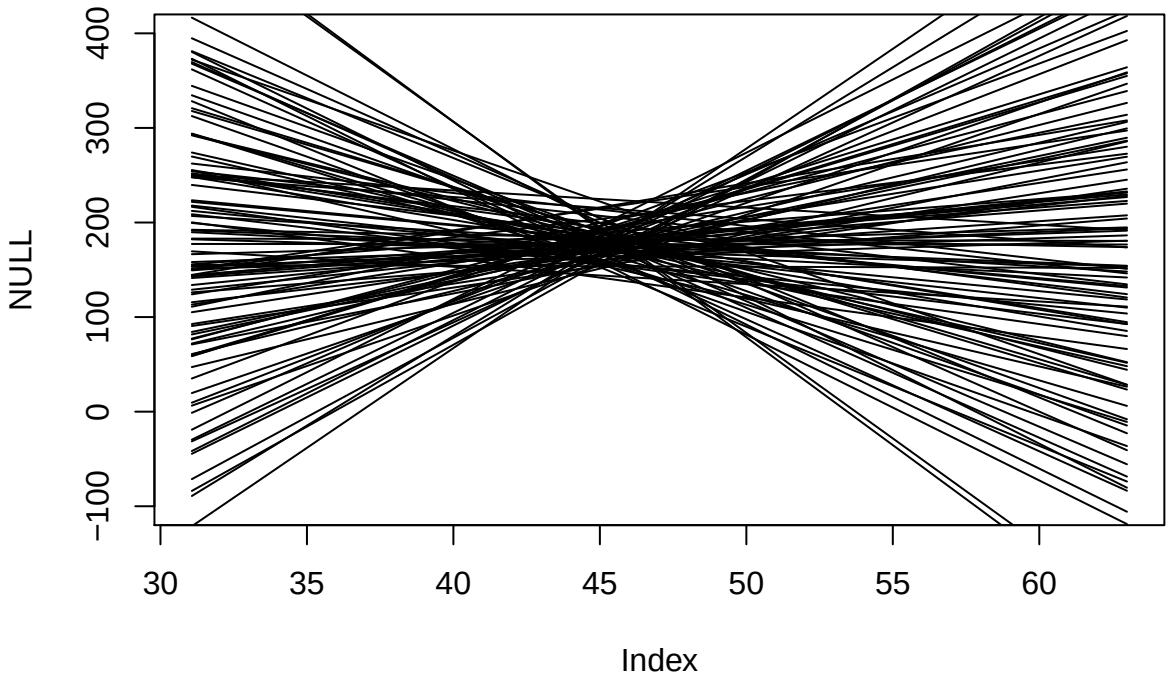
Linear prediction

Simulating to get the prior.

```
N <- 100
a <- rnorm(N, 178, 20)
b <- rnorm(N, 0, 10)
b2 <- rlnorm(N, 0, 1)
xbar <- mean(d2$weight)
print("Getting the normal prior")
```

```
## [1] "Getting the normal prior"
```

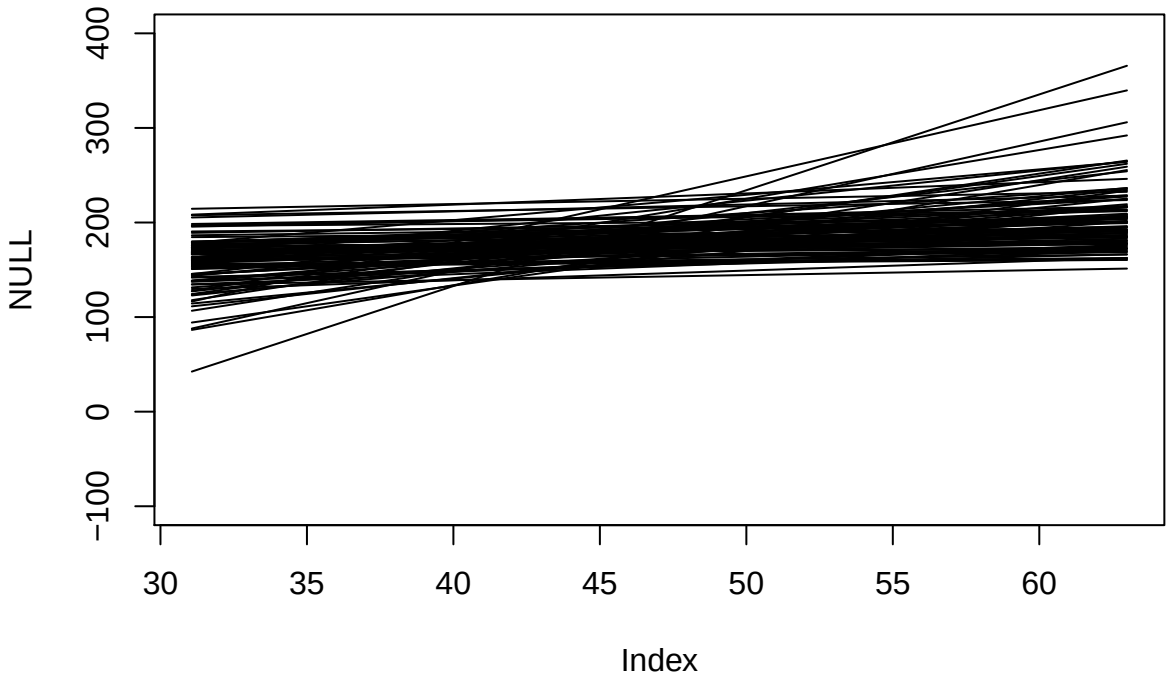
```
plot(NULL, xlim = range(d2$weight), ylim = c(-100,400))
for (i in 1:N){
  curve(a[i] + b[i]*(x - xbar), from = min(d2$weight),
        to=max(d2$weight),add = TRUE)
}
```




```
print("Getting the log-normal prior")
```

```
## [1] "Getting the log-normal prior"
```

```
plot(NULL, xlim = range(d2$weight), ylim = c(-100,400))
for (i in 1:N){
  curve(a[i] + b2[i]*(x - xbar), from = min(d2$weight),
        to=max(d2$weight),add = TRUE)
}
```



We fuss about priors for two reasons: 1. There are many analyses in which no amount of data makes the prior irrelevant. Non Bayesian methods also depend on such structural assumptions and are no better off. 2. Second, thinking about priors helps us develop better models, maybe even eventually going beyond geocentricism.

Rethinking: What is the correct prior? People often assume that there is a uniquely correct prior, which is wrong. Priors can be wrong in the same way that a hammer can be wrong for building a table. There exists guidelines to chose priors. Priors encode data and information before seeing data. Priors allow us to see consequences of beginning with different information. We can use priors to discourage certain parameter values such as negative associations between height and weight. When we don't know that much, we still now some information about the plausible range of values.

Rethinking: Prior predictive simulation and p-hacking If we choose to evaluate our choice of priors on observed data, that's cheating. The procedure performed is to try to choose priors based on pre-data knowledge. We're judging our choice of prior on general facts, not the sample.

Back to our regression model of weight and height which we have as:

$$h_i \sim Normal(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta(x_i - \bar{x})$$

$$\alpha \sim Normal(178, 20)$$

$$\beta \sim Log - Normal(0, 1)$$

$$\sigma \sim Uniform(0, 50)$$

We can encode this to our model

```
m4.3 <- quap(
  alist(
    height ~ dnorm(mu, sigma),
    mu <- a + b*(weight - xbar),
    a ~ dnorm(178,20),
    b ~ dlnorm(0,1),
    sigma ~ dunif(0,50)
  ), data = d2
)
```

Interpreting the posterior distribution