Chapter 4 Practice Problems

```
library(rethinking)
library(knitr)
set.seed(1)
```

Easy

4E1: In the model definition below, which line is the likelihood?

y_i ~ Normal(μ, σ)
μ ~ Normal(0, 10)
σ ~ Exponential(1)

The first line is the likelihood. The second and third lines are priors for the parameters μ and σ.

4E2: In the model definition just above, how many parameters are in the posterior distribution?

There are two parameters in the posterior distribution, μ and σ.

4E3: Using the model definition above, write down the approximate form of Bayes' theorem that includes the proper likelihood and priors.

The model on p.84 and example on p.87 are similar to this question. The posterior distribution will be of the form:

$$Pr(parameters|data) = \frac{Pr(data|parameters) * Pr(parameters)}{Pr(data)}$$

Where the denominator represents the average likelihood of observing the data, weighted across all parameter values, and can be rewritten as:

$$Pr(data) = \int Pr(data|parameters) * Pr(parameters) * dp$$

In the case of this specific model we can plug in the values as follows, noting the need to compute the joint likelihood across all y_i by multiplying individual likelihoods and the double integral in the denominator since we have two parameters:

$$Pr(\mu, \sigma|y) = \frac{\prod_i Pr(y_i|\mu, \sigma) * Pr(\mu) * Pr(\sigma)}{\int\int \prod_i Pr(y_i|\mu, \sigma) * Pr(\mu) * Pr(\sigma) * d\mu * d\sigma}$$

Together with the likelihood and prior definitions, this gives us:

$$Pr(\mu, \sigma|y) = \frac{\prod_i Normal(y_i|\mu, \sigma) * Normal(\mu|0, 10) * Exponential(\sigma|1)}{\int\int \prod_i Normal(y_i|\mu, \sigma) * Normal(\mu|0, 10) * Exponential(\sigma|1) * d\mu * d\sigma}$$

4E4: In the model definition below, which line is the linear model?

y_i ~ Normal(μ, σ)
$\mu_i = \alpha + \beta x_i$
α ~ Normal(0, 10)

β ~ Normal(0, 1)
σ ~ Exponential(2)

The second line is the linear model. The first line is the likelihood and the last three lines are priors.

4E5: In the model definition just above, how many parameters are in the posterior distribution?

There are three parameters in the posterior distribution: α, β, and σ. Since μ is defined deterministically based on α and β, it does not need to be estimated as a model parameter in the posterior distribution.

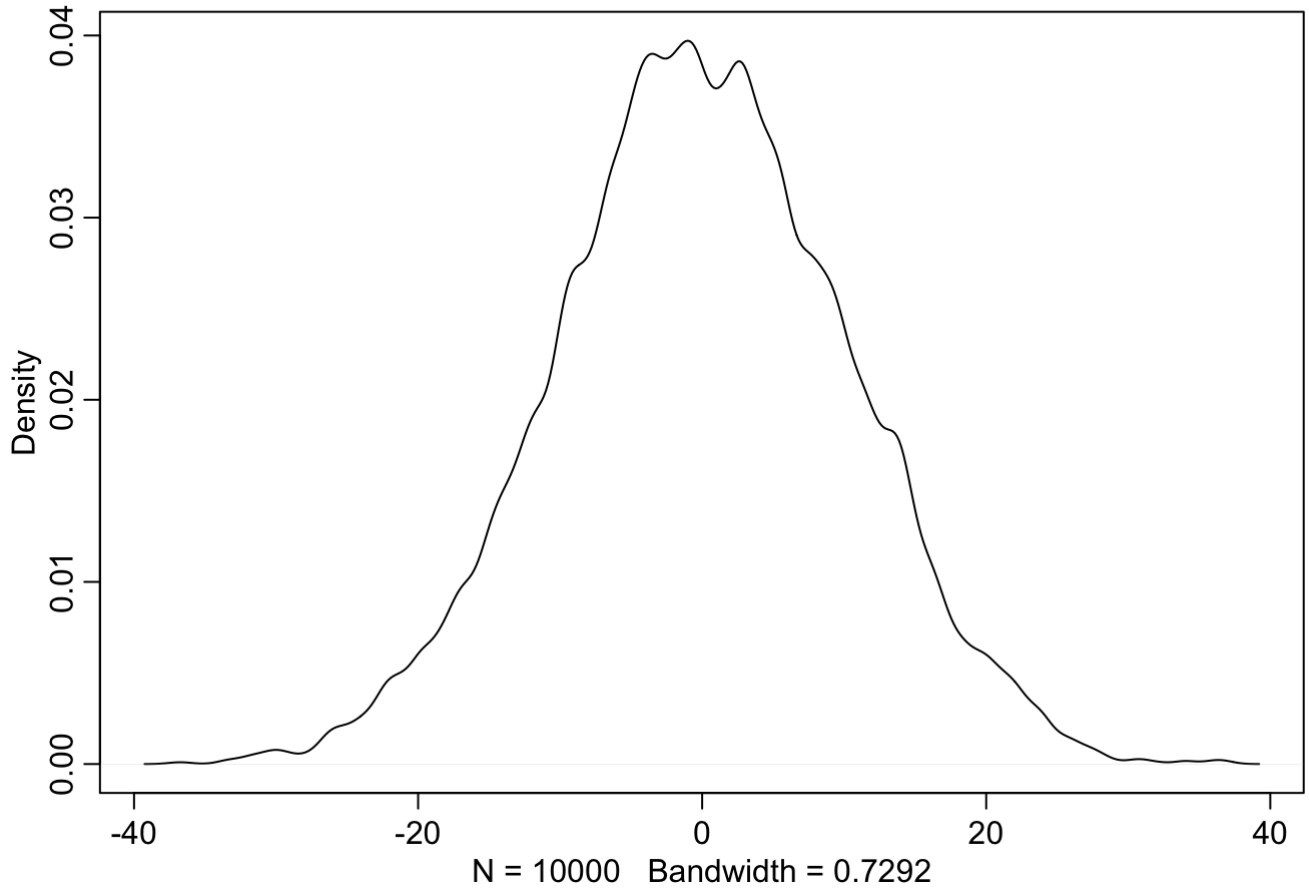
Medium

4M1: For the model definition below, simulate observed y values from the prior (not the posterior).

y_i ~ Normal(μ, σ)
μ ~ Normal(0, 10)
σ ~ Exponential(1)

R code 4.14 on p.85 is similar to this question. We can simulate y values from the prior with:

```
sample_mu <- rnorm(1e4, 0, 10)
sample_sigma <- rexp(1e4, 1)
prior_y <- rnorm(1e4, sample_mu, sample_sigma)
dens(prior_y)
```



4M2: Translate the model just above into a quap formula.

```
m4M1 <- quap(
  alist(
    y ~ dnorm(mu, sigma),
    mu ~ dnorm(0, 10),
    sigma ~ dexp(1)
  ), data = insertDataHere)
```

4M3: Translate the quap formula below into a mathematical model definition.

```
flist <- alist(
  y ~ dnorm(mu, sigma),
  mu <- a + b * x,
  a ~ dnorm(0, 10),
  b ~ dunif(0, 1),
  sigma ~ dexp(1)
)
```

y_i ~ Normal(μ, σ)
μ = a + b * x_i
a ~ Normal(0, 10)
b ~ Uniform(0, 1)
σ ~ Exponential(1)

4M4: A sample of students is measured for height each year for 3 years. After the third year, you want to fit a linear regression predicting height using year as a predictor. Write down the mathematical model definition for this regression, using any variable names and priors you chose. Be prepared to defend your choice of priors.

I based my model definition on the model on p.96, with a different prior for α since we are working with students. Each of the priors is quite general and leave room for improvement in 4M5 and 4M6. Note that we don't need to center the year values, but it's probably good practice for future techniques.

$Height_i$ ~ Normal(μ, σ)
$\mu = \alpha + \beta$ * ($Year_i$ - $\overline{Year}$)
α ~ Normal(150, 100)
β ~ Normal(0, 10)
σ ~ Uniform(0, 50)

4M5: Now suppose I remind you that every student got taller each year. Does this information lead you to change your choice of priors? How?

Since people do not shrink over time, the value for β should be non-negative. We can use a lognormal distribution to enforce this and adjust the prior for β to:

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

4M6: Now suppose I tell you that the variance among heights for students of the same age is never more than 64cm. How does this lead you to revise your priors?

Since the variance among heights for a given age is $\leq$ 64, we know that $\sigma \leq 8$. Then, we can adjust the prior for σ to:

$$\sigma \sim Uniform(0, 8)$$

Hard

4H1: The weights listed below were recorded in the !Kung census, but heights were not recorded for these individuals. Provide predicted heights and 89% intervals for each of these individuals. That is, fill in the table below, using model-based predictions.

Individual	weight	expected height	89% interval
1	46.95		
2	43.72		
3	64.78		
4	32.59		
5	54.63		

I used the linear model as in R code 4.42 on p.100-101:

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

xbar <- mean(d2$weight)

m4H1 <- 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)
```

We can then use the link function to generate samples and compute the corresponding values of the linear model for the desired weights. From these, we can generate the expected heights:

```
weights <- c(46.95, 43.72, 64.78, 32.59, 54.63)
mu <- link(m4H1, data = data.frame(weight = weights))
mu.mean <- apply(mu, 2, mean)
```

For the 89% intervals we also need to include σ, so we generate samples and compute the corresponding values of the linear model for the desired weights as above, but also generate draws from the normal distribution using those expected heights and the sample σs:

```
post <- extract.samples(m4H1)
mu.PI <- array(0, dim = c(2, length(weights)))

for(i in 1:length(weights)) {
  heights <- sapply(1:nrow(post), function(f) rnorm(1, post$a[f] + post$b[f] * (weights
[i] - xbar), post$sigma[f]))

  mu.PI[, i] <- PI(heights, prob = 0.89)
}
```

Displaying the relevant information in a nice table gives us:

```
mu_df <- data.frame("individual" = 1:5, "weight" = weights, "mean" = round(mu.mean, digi
ts = 2), "interval" = sapply(1:length(weights), function(f) paste(round(mu.PI[, f], digi
ts = 2), collapse = " - ")))

kable(mu_df, col.names = c("Individual", "weight", "expected height", "89% interval"), a
lign = "r")
```

Individual	weight	expected height	89% interval
1	46.95	156.36	148.26 - 164.36
2	43.72	153.44	145.38 - 161.78
3	64.78	172.47	164.28 - 180.62
4	32.59	143.39	135.23 - 151.49
5	54.63	163.30	155.12 - 171.35

4H2: Select out all the rows in the Howell1 data with ages below 18 years of age. If you do it right, you should end up with a new data frame with 192 rows in it.

We can filter only the individuals below 18 years of age with:

```
d2 <- d[d$age < 18, ]
```

(a) Fit a linear regression to these data, using quap. Present and interpret the estimates. For every 10 units of increase in weight, how much taller does the model predict a child gets?

I used the linear model as in R code 4.42 on p.100-101, with some more general priors for μ and b given that we are working with data for children only:

```
xbar <- mean(d2$weight)

m4H2 <- quap(
  alist(
    height ~ dnorm(mu, sigma),
    mu <- a + b * (weight - xbar),
    a ~ dnorm(100, 100),
    b ~ dlnorm(0, 10),
    sigma ~ dunif(0, 50)
  ),
  data = d2)
```

The estimates are:

```
precis(m4H2)
```

```
##             mean        sd       5.5%      94.5%
## a     108.318980 0.6088885 107.345859 109.292102
## b       2.718342 0.0683147   2.609162   2.827522
## sigma   8.437163 0.4305621   7.749041   9.125284
```

The estimate for a means that the expected height of a child of average weight is 108.32 cm. Next, the estimate for b means that for every 1 kg increase in weight, we expect a 2.72 cm increase in height. So for every 10 units of increase in weight, the model predicts that the child is 27.2 cm taller. Lastly, the estimate for σ means that the variance in height among children of a given weight is σ^2, or 71.19.

(b) Plot the raw data, with height on the vertical axis and weight on the horizontal axis. Superimpose the MAP regression line and 89% interval for the mean. Also superimpose the 89% interval for predicted heights.

This question asks for a similar plot to Figure 4.10 on p.112, which is derived in the preceding pages.

The MAP regression line is computed with R code 4.54 and 4.56 on p.108 by using the link function to generate samples for each desired weight, and then averaging across the samples for each weight:

```
weight.seq <- seq(from = 3, to = 50, by = 1)
mu <- link(m4H2, data = data.frame(weight = weight.seq))
mu.mean <- apply(mu, 2, mean)
```

The interval for the mean is computed with R code 4.56 on p.108 by finding the highest posterior density interval among the samples for each weight:

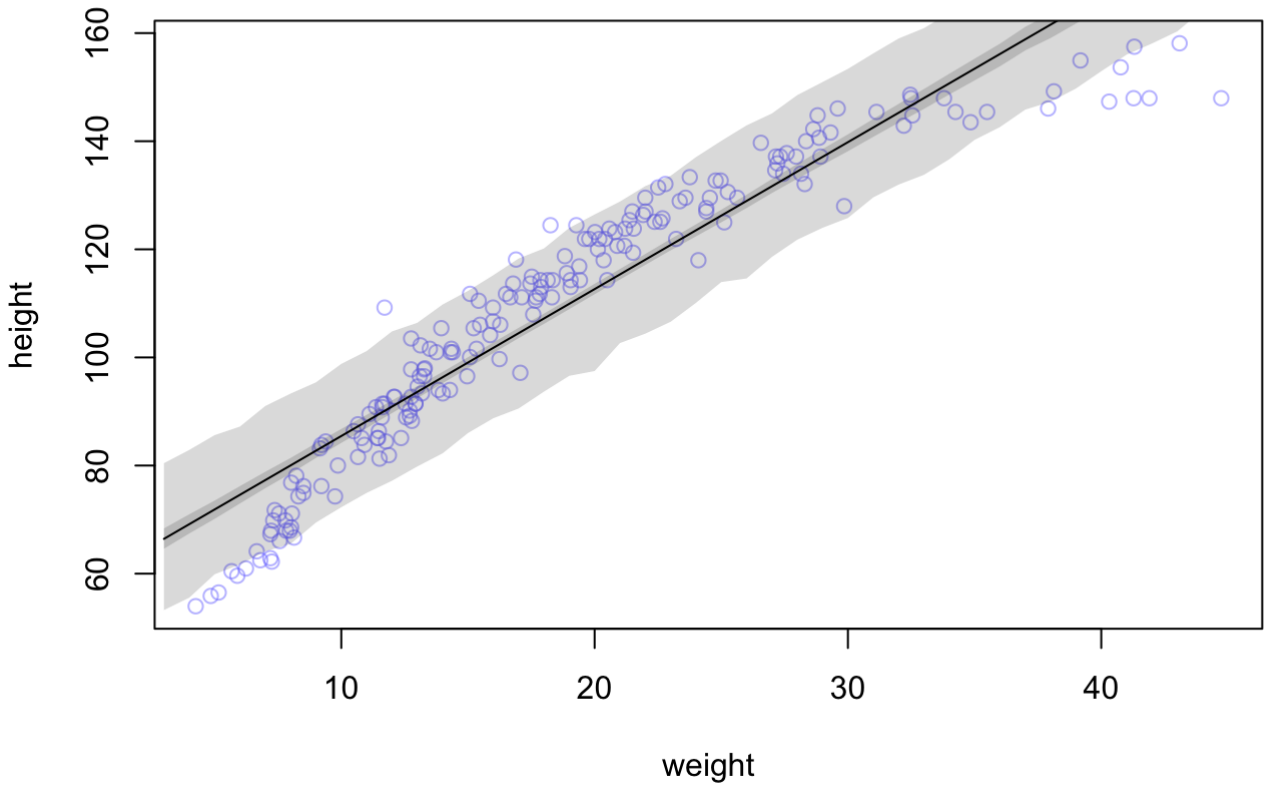
```
mu.HPDI <- apply(mu, 2, HPDI, prob = 0.89)
```

Lastly, the interval for predicted heights is computed with R code 4.59 - 4.60 on p.111 by using the sim function to sample heights from the Gaussian distribution for each desired weight, and then computing the desired interval among the samples for each weight:

```
sim.height <- sim(m4H2, data = list(weight = weight.seq))
height.PI <- apply(sim.height, 2, PI, prob = 0.89)
```

We can plot these together using R code 4.61 on p.111:

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



(c) What aspects of the model fit concern you? Describe the kinds of assumptions you would change, if any, to improve the model. You don't have to write any new code. Just explain what the model appears too be doing a bad job of, and what you hypothesize would be a better model.

The relationship between weight and height does not appear to be linear. The residuals vary systematically with respect to weight. At low weights the model tends to overpredict heights, at mid-weights the model tends to underpredict heights, and at high weights the model again tends to overpredict heights. I would change the form of what is currently our linear model to accommodate the curved shape of the relationship. Since the height increment per unit weight appears to level off at large weights, a logarithmic relationship may fit the data better.

4H3: Suppose a colleague of yours, who works on allometry, glances at the practice problems just above. Your colleague exclaims, "That's silly. Everyone knows that it's only the logarithm of body weight that scales with height!" Let's take your colleague's advice and see what happens.

(a) Model the relationship between height (cm) and the natural logarithm of weight (log-kg). Use the entire Howell1 data frame, all 544 rows, adults and non-adults. Fit this model, using quadratic approximation:

h_i ~ Normal(μ_i, σ)
$\mu_i = \alpha + \beta \log(w_i)$
α ~ Normal(178, 20)
β ~ Log-Normal(0, 1)
σ ~ Uniform(0, 50)

where h_i is the height of individual i and w_i is the weight (in kg) of individual i. The function for computing a natural log in R is just log. Can you interpret the resulting estimates?

```
m4H3 <- quap(
  alist(
    height ~ dnorm(mu, sigma),
    mu <- alpha + beta * log(weight),
    alpha ~ dnorm(178, 20),
    beta ~ dlnorm(0, 1),
    sigma ~ dunif(0, 50)
  ),
  data = d)
precis(m4H3)
```

```
##             mean        sd       5.5%      94.5%
## alpha -22.874993 1.3342651 -25.007407 -20.742580
## beta   46.817936 0.3823167  46.206920  47.428952
## sigma   5.136994 0.1558775   4.887871   5.386116
```

While nonsensical, α is the expected height (in cm) of an individual that weighs 1 kg. When weight = 1, log(weight) = 0 and the second line of the model simply becomes $\mu = \alpha$. Next, β is the height increment (in cm) that corresponds to a one order-of-magnitude increase in weight (in natural log scale, i.e. from e^2 ~ 7.39 to e^3 ~ 20.09). Lastly, σ can be interpreted as before, with σ^2 representing the variance in height among children of a given weight.

(b) Begin with this plot:

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

Then use samples from the quadratic approximate posterior of the model in (a) to superimpose on the plot: (1) the predicted mean height as a function of weight, (2) the 97% interval for the mean, and (3) the 97% interval for predicted heights.

This is essentially the same as in 4H2(b), with slight adjustments to account for the full range of weights:

```
weight.seq <- seq(from = 3, to = 70, by = 1)
mu <- link(m4H3, data = data.frame(weight = weight.seq))
mu.mean <- apply(mu, 2, mean)

mu.HPDI <- apply(mu, 2, HPDI, prob = 0.97)

sim.height <- sim(m4H3, data = list(weight = weight.seq))
height.PI <- apply(sim.height, 2, PI, prob = 0.97)

plot(height ~ weight, Howell1, col = col.alpha(rangi2, 0.4))
lines(weight.seq, mu.mean)
shade(mu.HPDI, weight.seq)
shade(height.PI, weight.seq)
```