Portfolio 1

Andreas Møldrup Holst

2025-03-01

```
## Install pacman if not install
# install.packages('pacman')

pacman::p_load(rethinking)
```

Question 1:

Which of the following statements corresponds to the expression: Pr(Monday|rain)? (1) The probability of rain on Monday. (2) The probability of rain, given that it is Monday. (3) The probability that it is Monday, given that it is raining. (4) The probability that it is Monday and that it is raining.

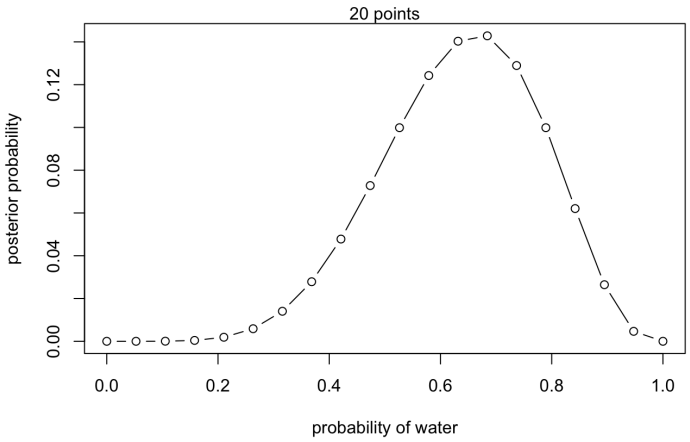
Answer:

In terms of the expression Pr(Monday|rain), the statement 3 is the correct answer.

Question 2:

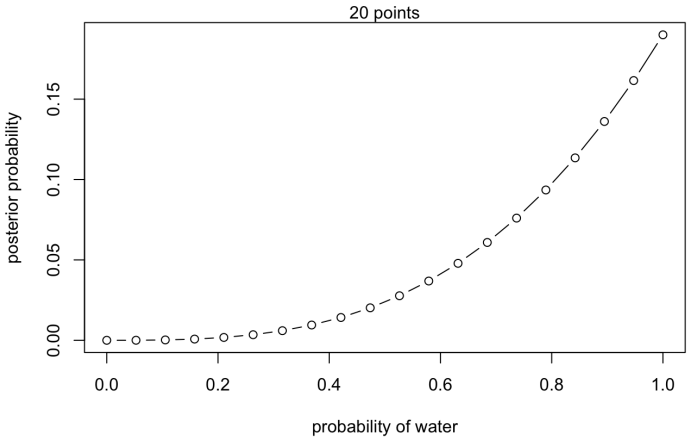
Recall the globe tossing model from the chapter.

```
#Define grid
p_grid <- seq(from=0, to=1, length.out=20)
#Define prior
prior <- rep(1 ,20)
#Compute likelihood at each value in grid
likelihood <- dbinom(6, size=9, prob=p_grid)
#Compute product of likelihood and prior
unstd.posterior <- likelihood*prior
#Standardize the posterior, so it sums to 1
posterior <- unstd.posterior/sum(unstd.posterior)

#Plot
plot(p_grid, posterior, type="b", xlab="probability of water",
     ylab="posterior probability")
mtext("20 points")
```



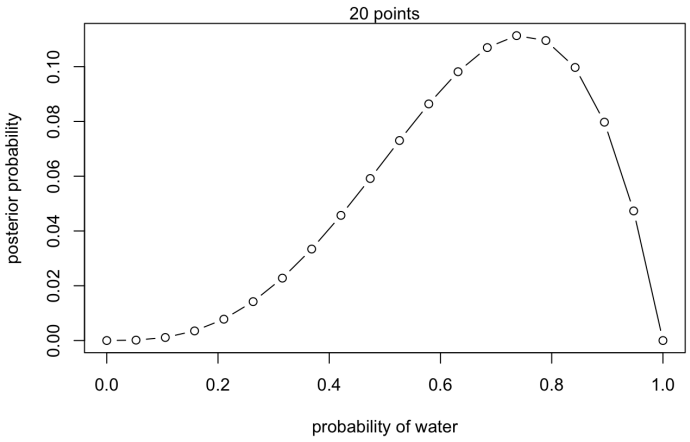
Compute and plot the grid approximate posterior distribution for each of the following sets of observations. In each case, assume a uniform prior for p. (1) W, W, W

```
#Define grid
p_grid <- seq(from=0, to=1, length.out=20)
#Define prior
prior <- rep(1, 20)
#Compute likelihood at each value in grid
likelihood_1 <- dbinom(3, size=3, prob=p_grid)
#Compute product of likelihood and prior
unstd.posterior_1 <- likelihood_1*prior
#Standardize the posterior, so it sums to 1
posterior_1 <- unstd.posterior_1/sum(unstd.posterior_1)

#Plot
plot(p_grid, posterior_1, type="b", xlab="probability of water",
     ylab="posterior probability")
mtext("20 points")
```



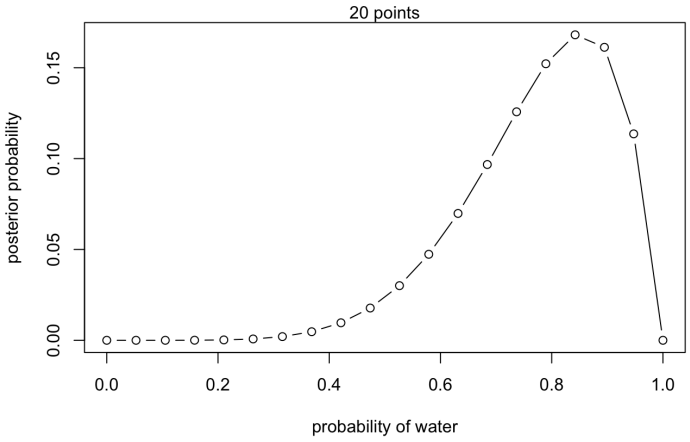
2. W, W, W, L

```
#Define grid
p_grid <- seq(from=0, to=1, length.out=20)
#Define prior
prior <- rep(1, 20)
#Compute likelihood at each value in grid
likelihood_2 <- dbinom(3, size=4, prob=p_grid)
#Compute product of likelihood and prior
unstd.posterior_2 <- likelihood_2*prior
#Standardize the posterior, so it sums to 1
posterior_2 <- unstd.posterior_2/sum(unstd.posterior_2)

#Plot
plot(p_grid, posterior_2, type="b", xlab="probability of water",
     ylab="posterior probability")
mtext("20 points")
```



3. W, W, W, L, W, W, W

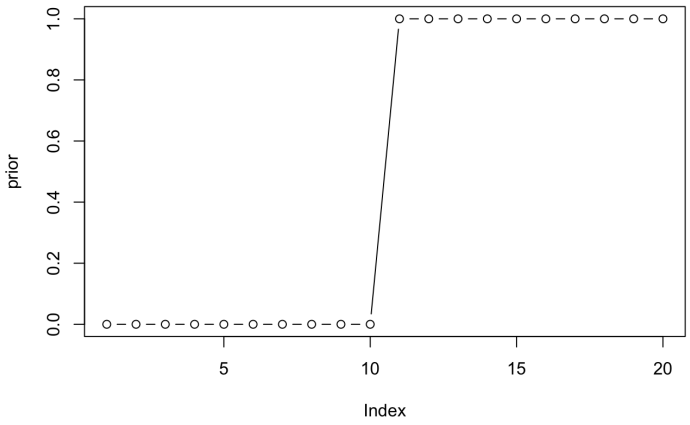
```
#Define grid
p_grid <- seq(from=0, to=1, length.out=20)
#Define prior
prior <- rep(1, 20)
#Compute likelihood at each value in grid
likelihood_3 <- dbinom(6, size=7, prob=p_grid)
#Compute product of likelihood and prior
unstd.posterior_3 <- likelihood_3*prior
#Standardize the posterior, so it sums to 1
posterior_3 <- unstd.posterior_3/sum(unstd.posterior_3)

#Plot
plot(p_grid, posterior_3, type="b", xlab="probability of water",
     ylab="posterior probability")
mtext("20 points")
```



Question 3:

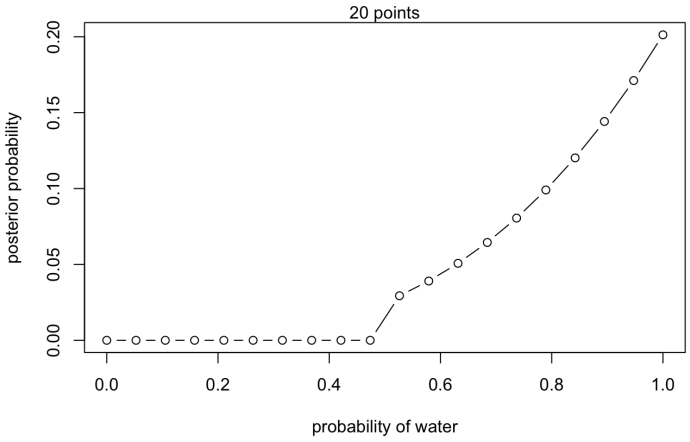
Now assume a prior for p that is equal to zero when p < 0.5 and is a positive constant when p ≥ 0.5. Again compute and plot the grid approximate posterior distribution for each of the sets of observations in the problem just above.

```
prior <- c(rep(0 ,10), rep(1 ,10))

plot(prior, type = "b")
```



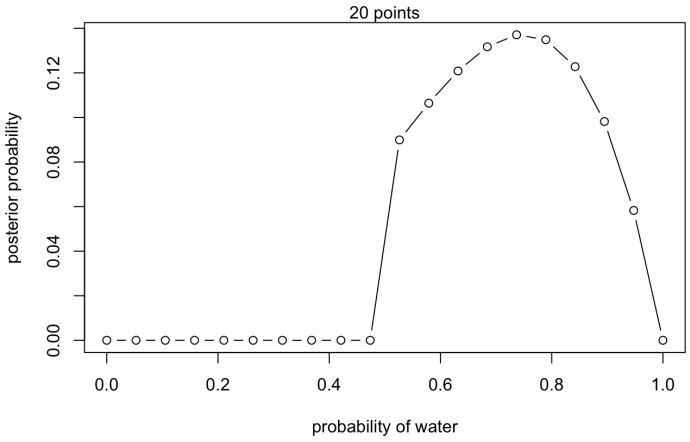
1. W, W, W

```
#Define grid
p_grid <- seq(from=0, to=1, length.out=20)

#Define prior
prior_new <- c(rep(0 , 10), rep(1 ,10))

#Compute likelihood at each value in grid
likelihood_3_1 <- dbinom(3, size=3, prob=p_grid)
#Compute product of likelihood and prior
unstd.posterior_3_1 <- likelihood_3_1*prior_new
#Standardize the posterior, so it sums to 1
posterior_3_1 <- unstd.posterior_3_1/sum(unstd.posterior_3_1)

#Plot
plot(p_grid, posterior_3_1, type="b", xlab="probability of water",
      ylab="posterior probability")
mtext("20 points")
```



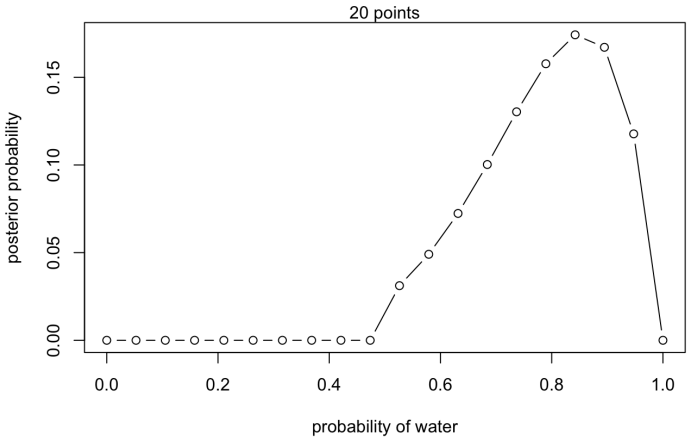
2. W, W, W, L

```
#Compute likelihood at each value in grid
likelihood_3_2 <- dbinom(3, size=4, prob=p_grid)
#Compute product of likelihood and prior
unstd.posterior_3_2 <- likelihood_3_2*prior_new
#Standardize the posterior, so it sums to 1
posterior_3_2 <- unstd.posterior_3_2/sum(unstd.posterior_3_2)

#Plot
plot(p_grid, posterior_3_2, type="b", xlab="probability of water",
      ylab="posterior probability")
mtext( "20 points" )
```



3. W, W, W, L, W, W, W

```
#Compute likelihood at each value in grid
likelihood_3_3 <- dbinom(6, size=7 ,prob=p_grid)
#Compute product of likelihood and prior
unstd.posterior_3_3 <- likelihood_3_3*prior_new
#Standardize the posterior, so it sums to 1
posterior_3_3 <- unstd.posterior_3_3/sum(unstd.posterior_3_3)

#Plot
plot(p_grid, posterior_3_3, type="b", xlab="probability of water",
     ylab="posterior probability")
mtext("20 points")
```



Question 4:

Suppose there are two globes, one for Earth and one for Mars. The Earth globe is 70% covered in water. The Mars globe is 100% land. Further suppose that one of these globes—you don't know which—was tossed in the air and produced a "land" observation. Assume that each globe was equally likely to be tossed. Show that the posterior probability that the globe was the Earth, conditional on seeing "land" (Pr(Earth|land)), is 0.23.

```
#Using Bayes theorem
prior <- 0.5 #Probability of Earth

likelihood <- 1-0.7 #Likelihood of land on Earth

marginal_probability <- 0.3*0.5+1*0.5 #Probability of land on Earth plus on non-Earth (Mars)

posterior_E_L <- (prior*likelihood)/marginal_probability #Bayes theorem

posterior_E_L #Result = 0.23
```

```
## [1] 0.2307692
```

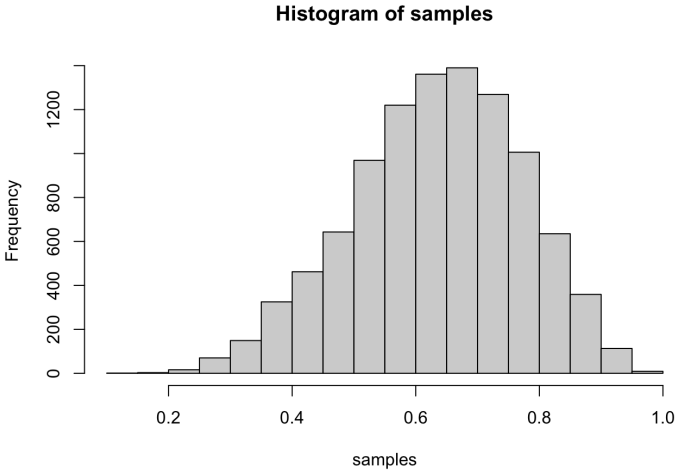
Answer:

By following Bayes' theorem, we can compute the probability that the globe was the Earth given seeing a land observation (Pr(Earth|land)) is 0.23.

Question 5:

Using a globe tossing example, resulting in 6 waters from 9 tosses:

```
p_grid <- seq(from=0, to=1, length.out=1000)
prior <- rep(1, 1000)
likelihood <- dbinom(6, size=9, prob=p_grid)
posterior <- likelihood*prior
posterior <- posterior/sum(posterior)
set.seed(100)
samples <- sample(p_grid, prob=posterior, size=1e4, replace=TRUE)
hist(samples)
```



Use the values in samples to answer the questions that follow (Monte Carlo Method):

```
# How much posterior probability lies below p = 0.2?
sum((samples<0.2)/length(samples))
```

```
## [1] 4e-04
```

```
# How much posterior probability lies above p = 0.8?
sum((samples>0.8)/length(samples))
```

```
## [1] 0.1116
```

```
# How much posterior probability lies between p = 0.2 and p = 0.8?
sum((samples>0.2 & samples<0.8)/length(samples))
```

```
## [1] 0.888
```

```
# 20% of the posterior probability lies below which value of p?
quantile(samples, 0.2)
```

```
##       20%
## 0.5185185
```

```
# 20% of the posterior probability lies above which value of p?
quantile(samples, 0.8)
```

```
##       80%
## 0.7557558
```

```
# Which values of p contain the narrowest interval equal to 66% of the posterior probability?
HPDI(samples, prob = 0.66)
```

```
##     |0.66     0.66|
## 0.5085085 0.7737738
```

```
# Which values of p contain 66% of the posterior probability, assuming equal posterior probability both below and
above the interval?
PI(samples, prob = 0.66)
```

```
##       17%       83%
## 0.5025025 0.7697698
```

Question 6:

These data indicate the gender (male=1, female=0) of officially reported first and second born children in 100 two-child families.

```
birth1 <- c(1,0,0,0,1,1,0,1,0,1,0,0,1,1,0,1,1,0,0,0,1,0,0,0,1,0,
0,0,0,1,1,1,0,1,0,1,1,1,0,1,0,1,1,0,1,0,0,1,1,0,1,0,0,0,0,0,0,0,
1,1,0,1,0,0,1,0,0,0,1,0,0,1,1,1,1,0,1,0,1,1,1,1,1,0,0,1,0,1,1,0,
1,0,1,1,1,0,1,1,1,1)
birth2 <- c(0,1,0,1,0,1,1,1,0,0,1,1,1,1,1,0,0,1,1,1,0,0,1,1,1,0,
1,1,1,0,1,1,1,1,0,1,0,0,1,1,1,1,1,0,0,1,0,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,0,1,1,0,1,1,0,1,1,1,1,0,0,0,0,0,0,1,0,0,0,1,1,0,0,1,1,0,0,1,
0,0,0,1,1,1,1,0,0,0)
```

So for example, the first family in the data reported a boy (1) and then a girl (0). The second family reported a girl (0) and then a boy (1). The third family reported two girls.

Use these vectors as data. So for example to compute the total number of boys born across all of these births, you could use:

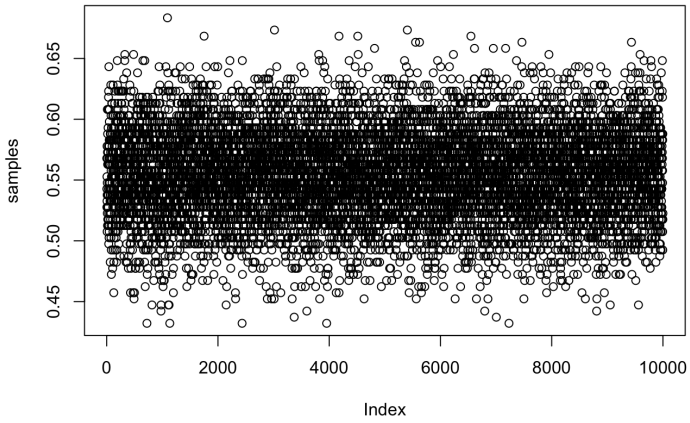
```
sum(birth1)+sum(birth2)
```

```
## [1] 111
```

Question 6.1

Using this data and grid approximation, compute the posterior distribution for the probability of a birth being a boy. Assume a uniform prior probability. Which parameter value maximizes the posterior probability?

```
set.seed(100)
p_grid <- seq(from = 0, to = 1, length.out = 200)
prior <- rep(1, 200)
likelihood <- dbinom(111, size = 200, prob = p_grid)
posterior <- likelihood*prior
posterior <- posterior/sum(posterior)
samples <- sample(p_grid, prob = posterior, size = 1e4, replace = TRUE)
plot(samples)
```




```
p_grid[which.max(posterior) ]
```

```
## [1] 0.5527638
```

Answer:

The parameter value which maximises the posterior probability of birth being a boy is 0.553.

Question 6.2

Using the sample function, draw 10,000 random parameter values from the posterior distribution you calculated above. Use these samples to estimate the 50%, 89%, and 97% highest posterior density intervals.

```
samples_child <- sample(p_grid, prob = posterior, size = 10.000, replace = TRUE)
HPDI(samples_child, prob = 0.5)
```

```
##      |0.5      0.5|
## 0.5527638 0.5879397
```

```
HPDI(samples_child, prob = 0.89)
```

```
##     |0.89     0.89|
## 0.5025126 0.5879397
```

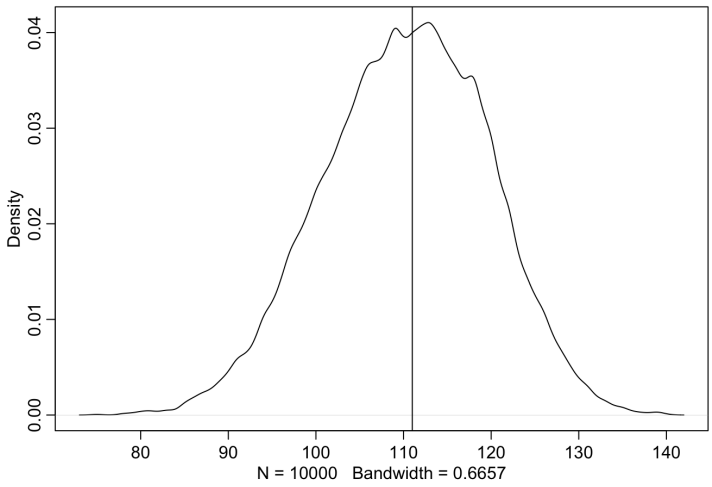
```
HPDI(samples_child, prob = 0.97)
```

```
##     |0.97     0.97|
## 0.5025126 0.5879397
```

Question 6.3

Use rbinom to simulate 10,000 replicates of 200 births. You should end up with 10,000 numbers, each one a count of boys out of 200 births. Compare the distribution of predicted numbers of boys to the actual count in the data (111 boys out of 200 births). There are many good ways to visualize the simulations, but the dens command (part of the rethinking package) is probably the easiest way in this case. Does it look like the model fits the data well? That is, does the distribution of predictions include the actual observation as a central, likely outcome?

```
set.seed(100)
samples_birth <- rbinom(10000, size = 200, prob = samples_child)
dens(samples_birth)
abline(v = 111)
```



Answer: Based on the sampling of 10.000 birth simulations the model fit the data well since the actual observation of 111 is almost central/near the mode of the distribution.

Question 7.1

In the model definition below, which line is the likelihood?

$$y_i \sim Normal(\mu, \sigma) \mu \sim Normal(0, 10) \sigma \sim Exponential(1)$$

Answer:

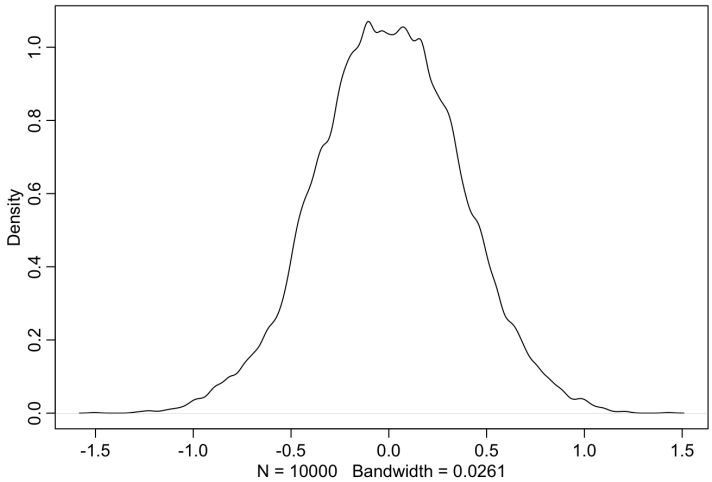
The model definition that describes the likelihood is the first line.

$$y_i \sim Normal(\mu, \sigma)$$

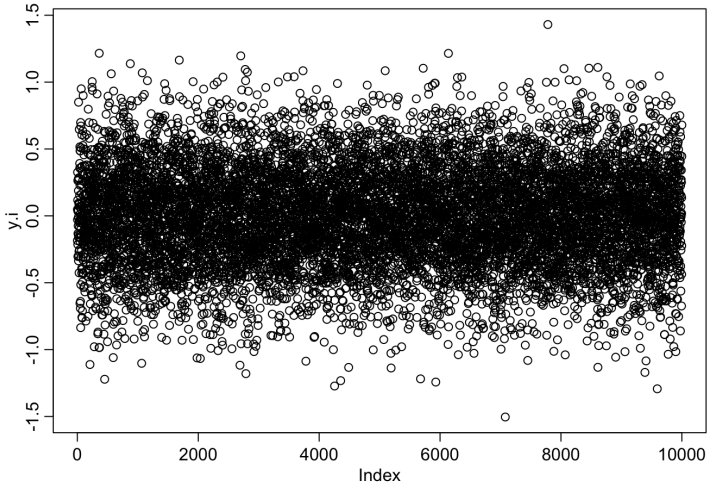
Question 7.2

For the model definition above, simulate observed y values from the prior (not the posterior).

```
# Simulate
set.seed(100)
mu <- dnorm(1e4, 0, 10)
sigma <- dexp(1)
y.i <- rnorm(1e4, mu, sigma)
dens(y.i)
```




```
plot(y.i)
```



Question 7.3

Translate the model just above into a quap formula.

```
flist <- alist(
  # Write formula here:
y.i ~ dnorm(mu, sigma),
mu ~ dnorm(0, 10),
sigma ~ dexp(1)
)
```

Question 8.1

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 <- c(1,2,3,4,5)
weight_seq <- c(46.95, 43.72, 64.78, 32.59, 54.63)
expected_height <- rep(NA, 5)
CI_of_89 <- rep(NA, 5)

data <- data.frame(Individual, weight_seq, expected_height)
```

As a help, I have recreated the model predictions in here:

```
# Load data again, to make sure we start with a clean slate:
library(rethinking)
data(Howell1)
d <- Howell1
d2 <- d[d$age >= 18, ]

# Define model:
flist <- alist(
  height ~ dnorm(mu, sigma), # Predictive distribution
  mu ~ dnorm(170, 20), # Prior
  sigma ~ dunif(0, 50) # Prior
)

# define the average weight, x-bar
xbar <- mean(d2$weight)

# fit model
m8.1 <- quap(
alist(height ~ dnorm(mu, sigma),
mu <- a+b*(weight-xbar),
a ~ dnorm(170, 20),
b ~ dnorm(0, 1),
sigma ~ dunif(0, 50)
),
data=d2)

precis(m8.1) # Get output of parameters:
```

```
##              mean         sd        5.5%       94.5%
## a     154.5999045 0.27030736 154.1679011 155.0319079
## b       0.9034411 0.04189135   0.8364906   0.9703915
## sigma   5.0718773 0.19115438   4.7663757   5.3773789
```

```
mu_pred <- link(m8.1, data = data.frame(weight=weight_seq))

data$expected_height <- apply(mu_pred, 2, mean)

CI_of_89 <- apply(mu_pred, 2, PI, prob = 0.89)

data$lower_89 <- CI_of_89[1, ]
```

```
data$upper_89 <- CI_of_89[2, ]
head(data)
```

```
##   Individual weight_seq expected_height lower_89 upper_89
## 1          1      46.95        156.3740 155.9053 156.8371
## 2          2      43.72        153.4543 153.0114 153.8956
## 3          3      64.78        172.4912 171.0906 173.9377
## 4          4      32.59        143.3935 142.4279 144.3218
## 5          5      54.63        163.3162 162.4789 164.1131
```

Question 8.2

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.

```
# Load data again, to make sure we start with a clean slate:
library(rethinking)
data(Howell1)
d <- Howell1

# Filter data:
d3 <- d[d$age < 18, ]
```

(a) Fit a linear regression to these data, using quap. Present and interpret the estimates.

```
xbar <- mean(d3$weight)

m8.2 <- quap(
alist(height ~ dnorm(mu, sigma),
mu <- a+b*(weight-xbar),
a ~ dnorm(150, 20),
b ~ dnorm(0, 1),
sigma ~ dunif(0, 50)
),
data=d3)

precis(m8.2)
```

```
##             mean        sd       5.5%      94.5%
## a     108.357904 0.6084588 107.385470 109.330339
## b       2.707562 0.0681269   2.598682   2.816442
## sigma   8.434776 0.4302782   7.747109   9.122444
```

For every 10 units of increase in weight, how much taller does the model predict a child gets?

```
posterior_samples <- extract.samples(m8.2)
mean(posterior_samples$b)*10
```

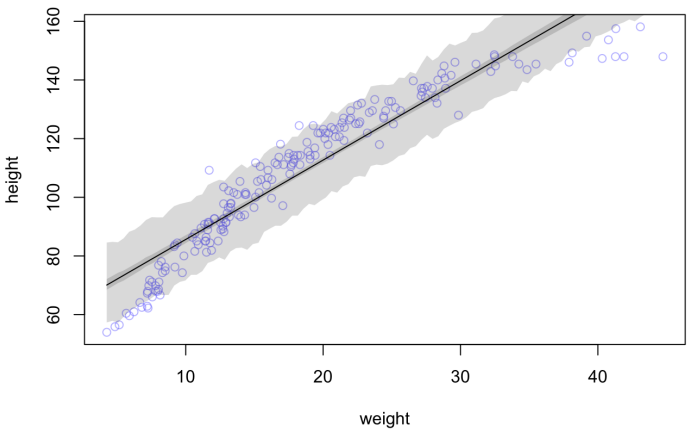
```
## [1] 27.07248
```

Answer:

For every 10 units of increase in weight, the model predicts an increase in heights of 27 cm.

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.

```
plot(height ~ weight, d3, col=col.alpha(rangi2,0.5))

weight_seq <- seq(from = min(d3$weight), to = max(d3$weight), length.out = 100)

mu_pred <- link(m8.2, data = data.frame(weight=weight_seq))

mu_mean <- apply(mu_pred, 2, mean)
mu_HPDI <- apply(mu_pred, 2, HPDI, prob = 0.89)

height_sim <- sim(m8.2, data = data.frame(weight = weight_seq))
height_HPDI = apply(height_sim, 2, HPDI, prob = 0.89)

# draw MAP line
lines(weight_seq, mu_mean)
# draw HPDI region for line
shade(mu_HPDI, weight_seq)
# draw PI region for simulated heights
shade(height_HPDI, 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 to be doing a bad job of, and what you hypothesize would be a better model.

Answer:

Children do not grow exponentially, they will at some point decrease in the rate they increase in height while at some point stop growing all together regardless of their increase in weight. Shortened, the relationship between increasing in weight and height is not linear.

Question 8.3

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} \sim Normal(\mu_{i}, \sigma) \\ μ_{i} = \alpha + \beta log(w_{i}) \\ \alpha \sim Normal(178, 20) \\ \beta \sim Log − Normal(0, 1) \\ \sigma ∼ Uniform(0, 50)

```
m8.3 <- quap(
  alist(
    height ~ dnorm(mu, sigma),
    mu <- a + b*log(weight),
    a~ dnorm(178, 20),
    b ~ dlnorm(0, 1),
    sigma ~ dunif(0, 50)
  ),
  data = d
)

precis(m8.3)
```

```
##             mean        sd      5.5%      94.5%
## a     -22.874280 1.3342857 -25.00673 -20.741834
## b      46.817771 0.3823225  46.20675  47.428796
## sigma   5.137072 0.1558835   4.88794   5.386204
```

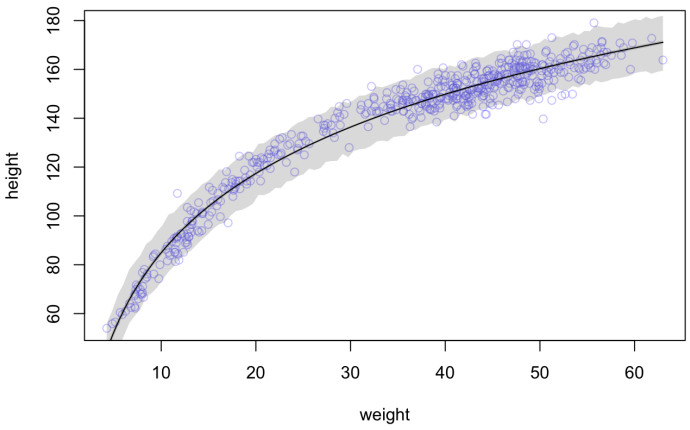
where h_{i} is the height of individual i and w_{i} is the weight (in kg) of individual i. The function for computing a natural log in R is just log(). Can you interpret the resulting estimates?

Answer:

The resulting estimates of the quadratic approximation infer that the relationship between weight and height now is modeled on a logarithmic scale, which implies a diminishing return effect. This means that small increases in weight leads to larger increases in height at lower weights, but this effect diminishes as weight increases resulting in the relationship to be non-linear.

(b) Begin with this plot:

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

weight_seq <- seq(from = min(d$weight), to = max(d$weight), length.out = 100)

mu_pred <- link(m8.3, data = data.frame(weight=weight_seq))

mu_mean <- apply(mu_pred, 2, mean)
mu_HPDI <- apply(mu_pred, 2, HPDI, prob = 0.97)

height_sim <- sim(m8.3, data = data.frame(weight = weight_seq))
height_HPDI = apply(height_sim, 2, HPDI, prob = 0.97)

lines(weight_seq, mu_mean)

shade(mu_HPDI, weight_seq)

shade(height_HPDI, weight_seq)
```



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.