Statistical Rethinking

Novica Nakov

2022-01-21

Contents

Chapter 1

bookdown

Statistical Rethinking e . . (PDF).

. . , . R4DS. .

Chapter 2

Small Worlds and Large Worlds

Exercises from Chapter 2 of the book.

```r
library(tidyverse)
library(patchwork)
```

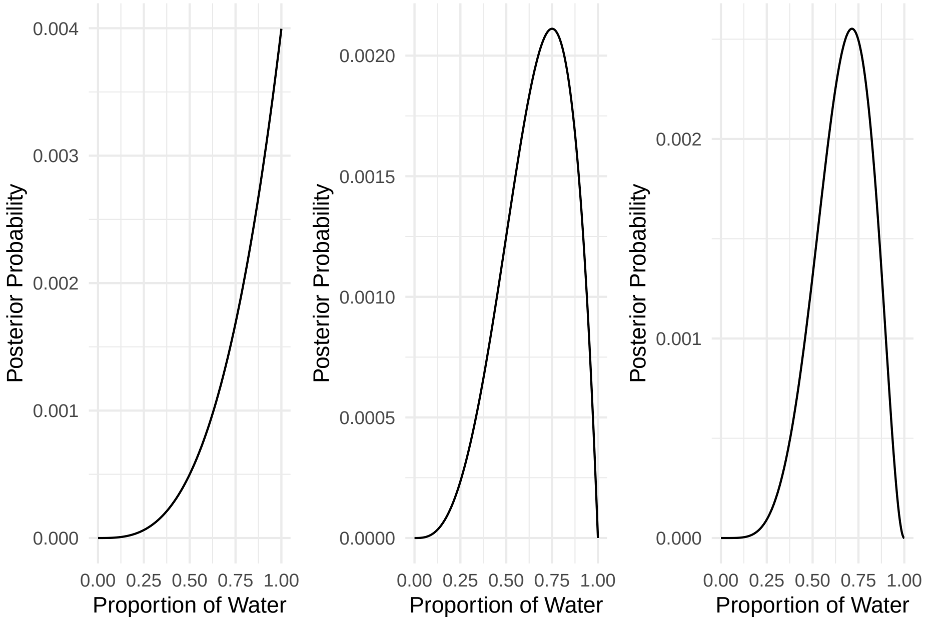
2M1. Recall the globe tossing model from the chapter. Compute and plot the grid approximate posterior distribution for each of the following sets of observations. In each case, assume a uniform prior for p. 1. W, W, W 2. W, W, W, L 3. L, W, W, L, W, W, W

There are three datasets for this problem. Let's store them in a list for ease of work below.

```r
data_1 <- c("W", "W", "W")
data_2 <- c("W", "W", "W", "L")
data_3 <- c("L", "W", "W", "L", "W", "W", "W")
data_lst <- list(data_1, data_2, data_3)
names(data_lst) <- c("data_1", "data_2", "data_3")
```

```r
# define grid same for all
p_grid <- seq(from = 0, to = 1, length.out = 1000)

# define prior same for all
prob_p <- rep(1, 1000)

#compute posterior

posterior <- function(data, p_grid, prob_p, plot = TRUE) {
```

```
# compute likelihood at each value in grid
prob_data <- dbinom(
  sum(data == "W"),
  size=length(data) ,
  prob=p_grid )
# compute product of likelihood and prior
posterior <- prob_data * prob_p
# standardize the posterior, so it sums to 1
posterior <- posterior / sum(posterior)

data_for_plot <- data.frame(param = p_grid, posterior = posterior)

if (plot) {
 p <-  ggplot(data_for_plot) +
   aes(x = param, y = posterior ) +
   geom_line() +
   labs(x = "Proportion of Water", y = "Posterior Probability") +
   theme_minimal()
 p
} else {
  posterior
}

}


plots <- lapply(data_lst, posterior, p_grid, prob_p)

plots[[1]]+ plots[[2]]+ plots[[3]]
```

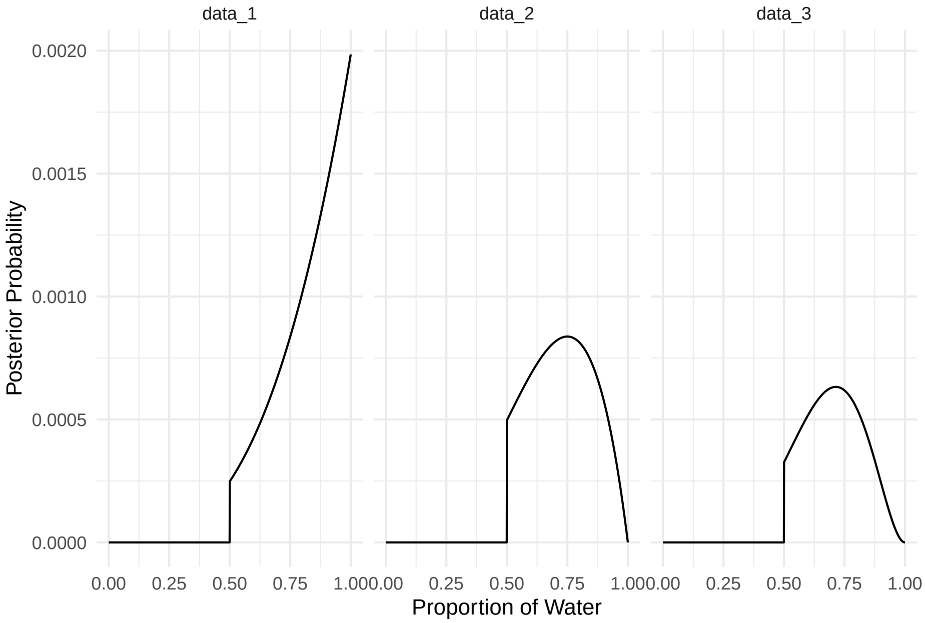


2M2. Now assume a prior for p that is equal to zero when $p < 0.5$ and is a positive constant when p 0.5. Again compute and plot the grid approximate posterior distribution for each of the sets of observations in the problem just above.

Exercise 2M2 is basically the same as 2M1. The only difference is the prior. Let's try it with a more of a tidyverse solution.

```
data_for_plots <-
  tibble(p_grid = seq(from = 0, to = 1, length.out = 1000)) %>%
  mutate(prior = if_else(p_grid < 0.5, 0, 1)) %>%
  mutate(data_1 = dbinom(sum(data_1 == "W"), size=length(data_1), prob = p_grid)) %>%
  mutate(data_2 = dbinom(sum(data_2 == "W"), size=length(data_2), prob = p_grid)) %>%
  mutate(data_3 = dbinom(sum(data_3 == "W"), size=length(data_3), prob = p_grid)) %>%
  pivot_longer(
    cols = starts_with("data"),
    names_to = "data",
    values_to = "observations"
  ) %>%
  mutate(unstd_posterior = observations * prior) %>%
  mutate(posterior = unstd_posterior / sum(unstd_posterior))
```

```
ggplot(data_for_plots) +
  aes(x = p_grid, y = posterior) +
  geom_line() +
  labs(x = "Proportion of Water", y = "Posterior Probability") +
  theme_minimal() +
  facet_wrap(vars(data), nrow = 1)
```



*2M3. 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.*1

$Pr(Earth|land) = 0.23$

$Pr(Earth) = Pr(Mars) = 0.5$

$Pr(land|Earth) = 1 - 0.7 = 0.3$

$Pr(land|Mars) = 1$

Bayes theorem:

$Pr(Earth|land) = Pr(land|Earth) * Pr(Earth)/Pr(land|Earth) * Pr(Earth) + Pr(land|Mars) * Pr(Mars)$

```
pr_earth_given_land <- 0.3 * 0.5 / (0.3 * 0.5 + 1 * 0.5)

pr_earth_given_land
```

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

2M4. Suppose you have a deck with only three cards. Each card has two sides, and each side is either black or white. One card has two black sides. The second card has one black and one white side. The third card has two white sides. Now suppose all three cards are placed in a bag and shuffled. Someone reaches into the bag and pulls out a card and places it flat on a table. A black side is shown facing up, but you don't know the color of the side facing down. Show that the probability that the other side is also black is 2/3. Use the counting method (Section 2 of the chapter) to approach this problem. This means counting up the ways that each card could produce the observed data (a black side facing up on the table).

```
bb_ways <- 2
wb_ways <- 1
ww_ways <- 0
data <- c(bb_ways, wb_ways, ww_ways)
prior <- c(1, 1, 1)
posterior <- prior * data
posterior <- posterior / sum(posterior)
posterior[1] == 2 / 3
```

```
## [1] TRUE
```

2M5. Now suppose there are four cards: B/B, B/W, W/W, and another B/B. Again suppose a card is drawn from the bag and a black side appears face up. Again calculate the probability that the other side is black.

```
bb_ways <- 2
wb_ways <- 1
ww_ways <- 0
data <- c(bb_ways, wb_ways, ww_ways)
prior <- c(2, 1, 1)
posterior <- prior * data
posterior <- posterior / sum(posterior)

posterior[1]
```

```
## [1] 0.8
```

2M6. Imagine that black ink is heavy, and so cards with black sides are heavier than cards with white sides. As a result, it's less likely that a card with black sides is pulled from the bag. So again assume there are three cards: B/B, B/W, and W/W. After experimenting a number of times, you conclude that for every way to pull the B/B card from the bag, there are 2 ways to pull the B/W card and 3 ways to pull the W/W card. Again suppose that a card is pulled and a black side appears face up. Show that the probability the other side is black is now 0.5. Use the counting method, as before.

```
bb_ways <- 2
wb_ways <- 1
ww_ways <- 0
data <- c(bb_ways, wb_ways, ww_ways)
prior <- c(1, 2, 3)
posterior <- prior * data
posterior <- posterior / sum(posterior)

posterior[1] == 0.5
```

```
## [1] TRUE
```

2M7. Assume again the original card problem, with a single card showing a black side face up. Before looking at the other side, we draw another card from the bag and lay it face up on the table. The face that is shown on the new card is white. Show that the probability that the first card, the one showing a black side, has black on its other side is now 0.75. Use the counting method, if you can. Hint: Treat this like the sequence of globe tosses, counting all the ways to see each observation, for each possible first card.

```
bb_ways <- 2*3
wb_ways <- 1*2
ww_ways <- 0*1
data <- c(bb_ways, wb_ways, ww_ways)
prior <- c(1, 1, 1)
posterior <- prior * data
posterior <- posterior / sum(posterior)

posterior[1] == 0.75
```

```
## [1] TRUE
```

2H1. Suppose there are two species of panda bear. Both are equally common in the wild and live in the same places. They look exactly alike and eat the same food, and there is yet no genetic assay capable of telling them apart. They differ however in their family sizes. Species A gives birth to twins 10% of the time, otherwise birthing a single infant. Species B births twins 20% of the time, otherwise birthing singleton infants. Assume these numbers are known with certainty, from many years of field research.

Now suppose you are managing a captive panda breeding program. You have a new female panda of unknown species, and she has just given birth to twins. What is the probability that her next birth will also be twins?

, , , :

$Pr(twins2|twins1) = ?$

:

$Pr(twins2|twins1) = Pr(twins1, twins2)/Pr(twins)$

:

$Pr(twins|A) = 0.1$

$Pr(twins|B) = 0.2$

$Pr(A) = Pr(B) = 0.5$

:

$Pr(twins) = Pr(twins|A) * Pr(A) + Pr(twins|B) * Pr(B)$

$Pr(twins1, twins2) = Pr(twins|A) * Pr(twins|A) * Pr(A) + Pr(twins|B) * Pr(twins|B) * Pr(B) $

```
pr_twins <- 0.1 * 0.5 + 0.2 * 0.5

pr_twins1_and_twins2 <- 0.1 * 0.1 * 0.5 + 0.2 * 0.2 * 0.5

pr_twins_2_given_twins1 <- pr_twins1_and_twins2 / pr_twins

pr_twins_2_given_twins1
```

```
## [1] 0.1666667
```

2H2. Recall all the facts from the problem above. Now compute the probability that the panda we have is from species A, assuming we have observed only the first birth and that it was twins.

:

$Pr(A|twins1) = ?$

:

$Pr(A|twins1) = Pr(twins1|A) * Pr(A)/Pr(twins)$

:

```
pr_twins_given_a = 0.1 * 0.5 / pr_twins

pr_twins_given_a
```

```
## [1] 0.3333333
```

, , . .:

$Pr(A|twins1)$ $Pr(B|twins1)$.

```
p_twins_A <- 0.1
p_twins_B <- 0.2
data_twins <- c(p_twins_A, p_twins_B)
prior <- c(1, 1)
```

```
posterior <- prior * data_twins
posterior_2H2 <- posterior / sum(posterior)
names(posterior_2H2) = c("pr_A_twins", "pr_B_twins")
posterior_2H2
```

```
## pr_A_twins pr_B_twins
##  0.3333333  0.6666667
```

2H3. Continuing on from the previous problem, suppose the same panda mother has a second birth and that it is not twins, but a singleton infant. Compute the posterior probability that this panda is species A.

, :

$Pr(A|single) = ?$

bay sian updating

, *prior* . .:

$Pr(A|twins) = 0.3$

```
data_single <- c(1 - p_twins_A, 1 - p_twins_B)
prior_2H3 <- posterior_2H2
posterior <- prior_2H3 * data_single
posterior_2H3 <- posterior / sum(posterior)
names(posterior_2H3) = c("pr_A_single", "pr_B_single")
posterior_2H3
```

```
## pr_A_single pr_B_single
##        0.36        0.64
```

2H4. A common boast of Bayesian statisticians is that Bayesian inference makes it easy to use all of the data, even if the data are of different types.

So suppose now that a veterinarian comes along who has a new genetic test that she claims can identify the species of our mother panda. But the test, like all tests, is imperfect. This is the information you have about the test:

**The probability it correctly identifies a species A panda is 0.8.* **The probability it correctly identifies a species B panda is 0.65.*

The vet administers the test to your panda and tells you that the test is positive for species A. First ignore your previous information from the births and compute the posterior probability that your panda is species A. Then redo your calculation, now using the birth data as well.

:

$Pr(A|testA) = ?$

:

$Pr(testA|A) = 0.8$ $Pr(testA|B) = 1 - 0.65 = 0.35$ B 0.65 , , 0.35 .

```
test_correct_A <- 0.8
test_correct_B <- 0.35
data <- c(test_correct_A, test_correct_B)
prior <- c(1, 1)
posterior <- prior * data
posterior_2H4 <- posterior / sum(posterior)
names(posterior_2H4) = c("panda_is_A", "panda_is_B")
posterior_2H4
```

```
## panda_is_A panda_is_B
##  0.6956522  0.3043478
```

, , *prior* . . *posterior* .

```
prior <- posterior_2H3
posterior <- prior * data
posterior_2H4_2 <- posterior / sum(posterior)
names(posterior_2H4_2) = c("panda_is_A", "panda_is_B")
posterior_2H4_2
```

```
## panda_is_A panda_is_B
##     0.5625     0.4375
```

:

$Pr(A|testA, twins, single) = 0.56$

, 0.56.

Chapter 3

Sampling the Imaginary

Exercises from Chapter 3 of the book.

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

3.2. Sampling to summarize section .

```
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 )
```

3E1. How much posterior probability lies below $p = 0.2$?

```
mean( samples < 0.2 )
```

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

3E2. How much posterior probability lies above $p = 0.8$?

```
mean( samples > 0.8 )
```

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

3E3. How much posterior probability lies between $p = 0.2$ and $p = 0.8$?

```
mean( samples > 0.2 & samples < 0.8 )
```

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

ll three should be 1.

```
sum(mean( samples < 0.2 ), mean( samples > 0.8 ), mean( samples > 0.2 & samples < 0.8 )
```

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

3E4. 20% of the posterior probability lies below which value of p?

```
quantile(samples, 0.2)
```

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

3E5. 20% of the posterior probability lies above which value of p?

```
quantile(samples, 0.8)
```

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

3E6. 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
```

Trying the tidybayes functions. The results are the same.

```
tidybayes::mode_hdi(samples, .width = .66)
```

```
##           y      ymin      ymax .width .point .interval
## 1 0.6477573 0.5085085 0.7737738   0.66   mode       hdi
```

```
tidybayes::hdi(samples, .width = .66)
```

```
##           [,1]      [,2]
## [1,] 0.5085085 0.7737738
```

3E7. Which values of p contain 66% of the posterior probability, assuming equal posterior probability both below and above the interval?

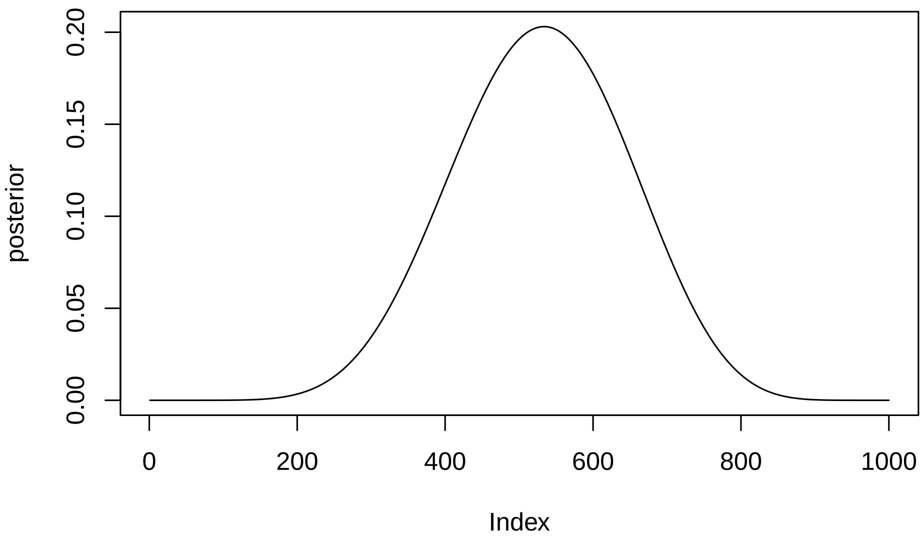
```
PI(samples, prob = 0.66)
```

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

3M1. Suppose the globe tossing data had turned out to be 8 water in 15 tosses. Construct the posterior distribution, using grid approximation. Use the same flat prior as before.

, `dbinom`.

```
p_grid <- seq(from=0, to=1, length.out=1000)
prior <- rep(1, 1000)
likelihood <- dbinom(8, size=15, prob=p_grid)
posterior <- likelihood * prior
posterior_3m1 <- posterior / sum(posterior)

plot(posterior, type = "l")
```



3M2. Draw 10,000 samples from the grid approximation from above. Then use the samples to calculate the 90% HPDI for p.

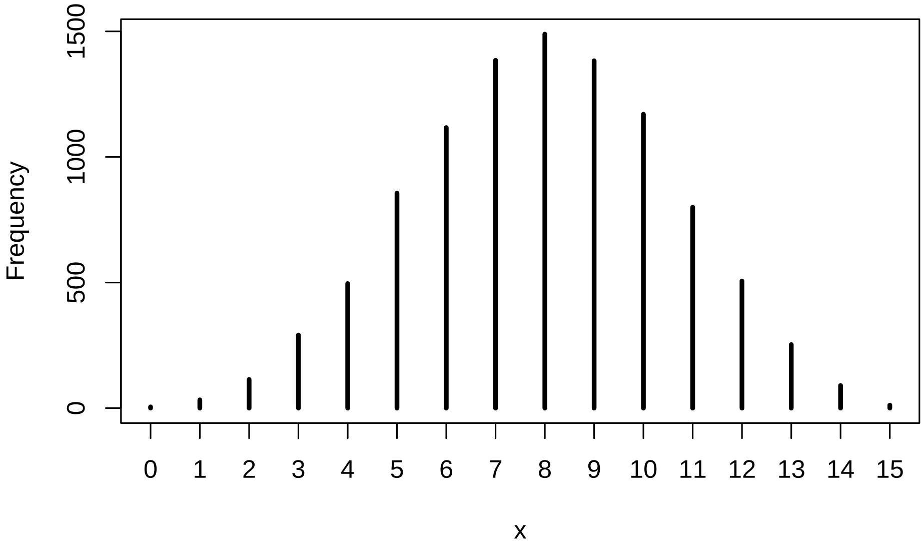
```
samples <- sample( p_grid , prob=posterior_3m1 , size=1e4 , replace=TRUE )

HPDI( samples , prob=0.9 )
```

```
##      |0.9      0.9|
## 0.3293293 0.7167167
```

3M3. Construct a posterior predictive check for this model and data. This means simulate the distribution of samples, averaging over the posterior uncertainty in p.

. 3.3.2 .

```
set.seed(123)

simulate_w <- rbinom( 1e4 , size=15 , prob=samples )

simplehist(simulate_w)
```



What is the probability of observing 8 water in 15 tosses?

```
mean(simulate_w == 8)
```

```
## [1] 0.1489
```

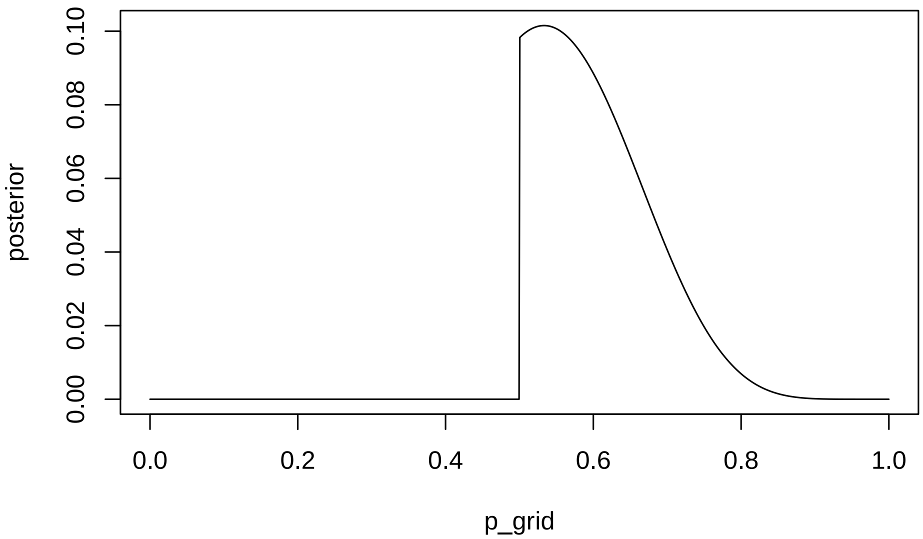
3M4. Using the posterior distribution constructed from the new (8/15) data, now calculate the probability of observing 6 water in 9 tosses.

```
set.seed(123)

simulate_w <- rbinom( 1e4 , size=9 , prob=samples )

mean(simulate_w == 6)
```

```
## [1] 0.1732
```

3M5. Start over at 3M1, but now use a prior that is zero below p = 0.5 and a constant above p = 0.5. This corresponds to prior information that a majority of the Earth's surface is water. Repeat each problem above and compare the inferences. What difference does the better prior make? If it helps, compare inferences (using both priors) to the true value p = 0.7.

```
p_grid <- seq( from=0 , to=1 , length.out=1000 )
prior3m5 <- ifelse(p_grid < 0.5, 0, 0.5)
likelihood <- dbinom( 8 , size=15 , prob=p_grid )
posterior <- likelihood * prior3m5
posterior_3m5 <- posterior / sum(posterior)

plot(p_grid, posterior , type = "l")
```



posterior 0.5

.

```
samples <- sample( p_grid , prob=posterior_3m5 , size=1e4 , replace=TRUE )

HPDI( samples , prob=0.9 )
```

```
##      |0.9      0.9|
## 0.5005005 0.7167167
```

HPDI 3 2.

3M6. Suppose you want to estimate the Earth's proportion of water very precisely. Specifically, you want the 99% percentile interval of the posterior distribution of p to be only 0.05 wide. This means the distance between the upper and lower bound of the interval should be 0.05. How many times will you have to toss the globe to do this?

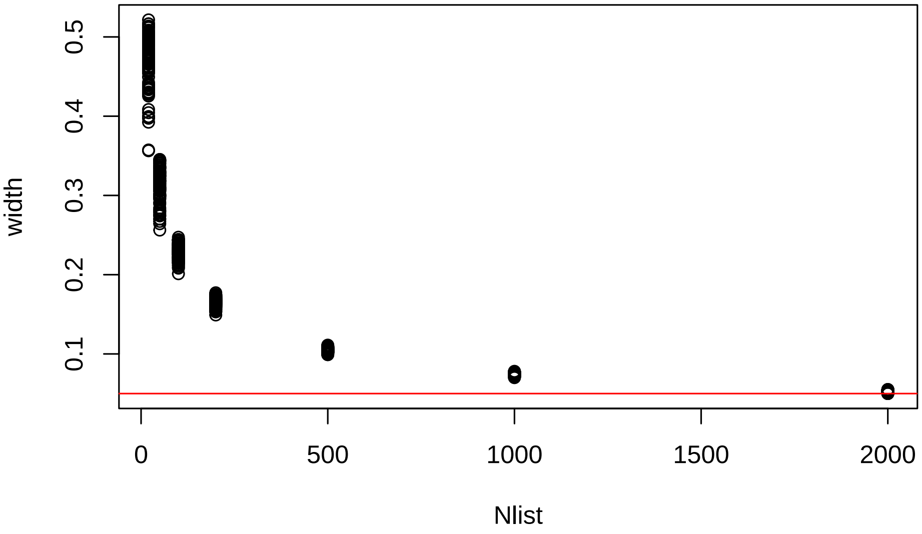
Richard McEarleth.

.

, .

(20 2000).

```
f <- function(N) {
  p_true <- 0.7
  W <- rbinom(1, size = N, prob = p_true)
  prob_grid <- seq(0, 1, length.out = 1000)
  prior <- rep(1, 1000)
  prob_data <- dbinom(W, size = N, prob = prob_grid)
  posterior <- prob_data * prior
  posterior <- posterior / sum(posterior)
  samples <- sample(p_grid, prob = posterior, size = 1e4, replace = TRUE)
  PI99 <- PI(samples, .99)
  as.numeric(PI99[2] - PI99[1])
}
Nlist <- c(20, 50, 100, 200, 500, 1000, 2000)
Nlist <- rep(Nlist, each = 100)
width <- sapply(Nlist, f)
plot(Nlist, width)
abline(h = 0.05, col = "red")
```

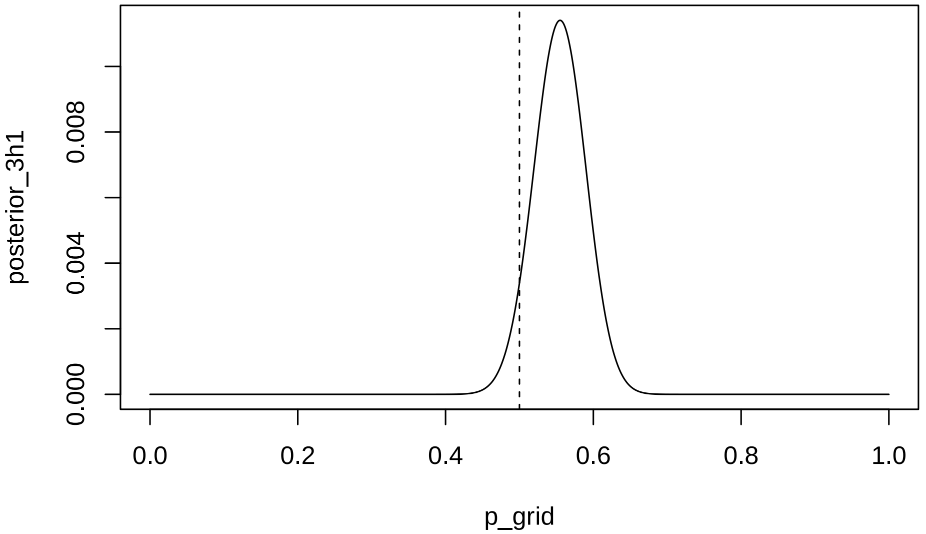


3H1. Using grid approximation, compute the posterior distribution for the probability of a birth being a boy. Assume a uniform prior probability.

```
data(homeworkch3)

all_boys <- sum(birth1) + sum(birth2)
```

```
p_grid <- seq( from=0 , to=1 , length.out=1000 )

prior3h1 <- prior <- rep(1,length(p_grid))

likelihood <- dbinom( all_boys , size=length(birth1) + length(birth2) , prob=p_grid )

posterior <- likelihood * prior3h1

posterior_3h1 <- posterior / sum(posterior)

plot(p_grid, posterior_3h1 , type="l" )
```

```
abline( v=0.5 , lty=2 )
```



Which parameter value maximizes the posterior probability?

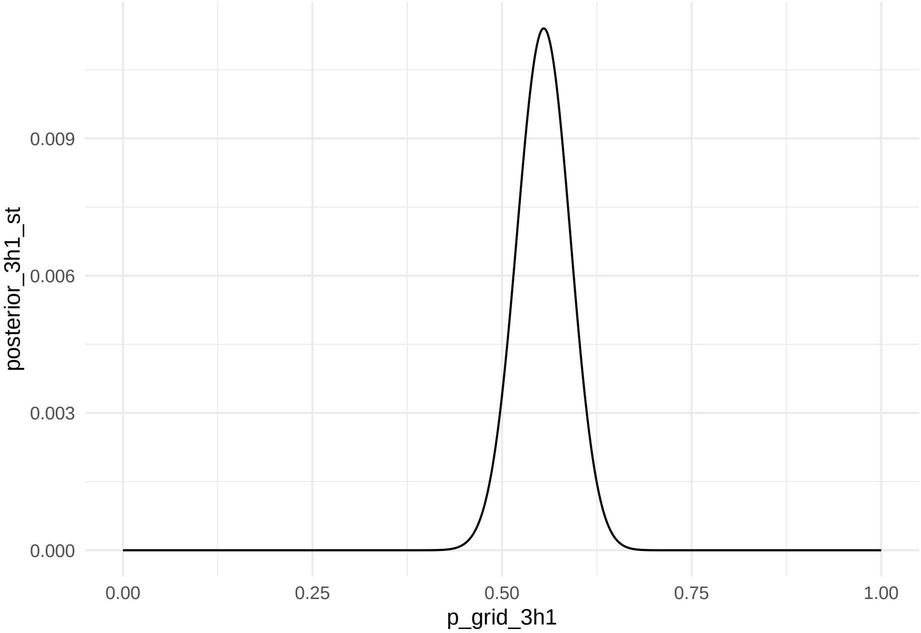
```
p_grid[ which.max(posterior) ]
```

```
## [1] 0.5545546
```

tidyverse :

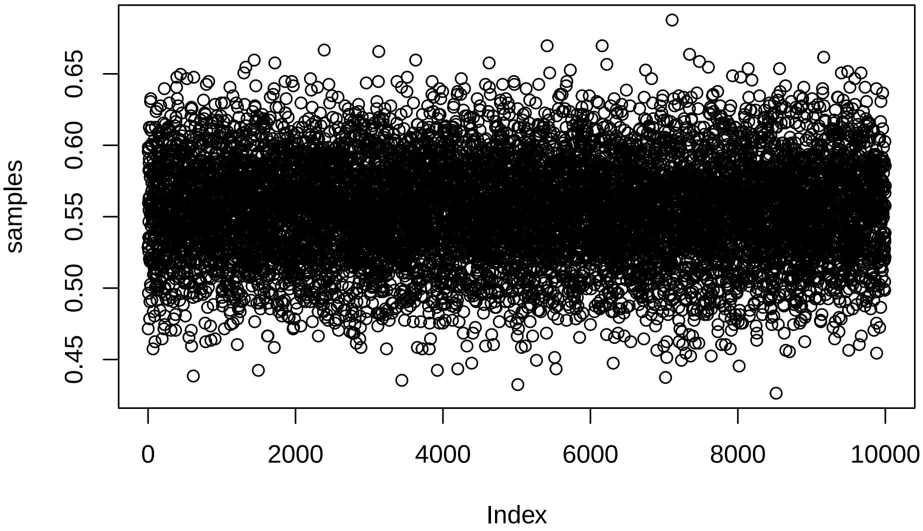
```
data_for_3h1 <-
  tibble(p_grid_3h1 = seq(from = 0, to = 1, length.out = 1000)) %>%
  mutate(prior_3h1 = 1) %>%
  mutate(data_for_3h1 = dbinom( all_boys , size=length(birth1) + length(birth2) , prob=p_grid_3h1
  mutate(posterior_3h1 = data_for_3h1 * prior_3h1) %>%
  mutate(posterior_3h1_st = posterior_3h1 / sum(posterior_3h1))
```

```
ggplot(data_for_3h1) +
  aes(x = p_grid_3h1, y = posterior_3h1_st) +
  geom_line() +
  theme_minimal()
```



3H2. 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 <- sample( data_for_3h1$p_grid_3h1 , prob=data_for_3h1$posterior_3h1_st , size=

plot(samples)
```



```
tidybayes::hdi(samples, .width = .50)
```

```
##           [,1]      [,2]
## [1,] 0.5255255 0.5725726
```

```
tidybayes::hdi(samples, .width = .89)
```

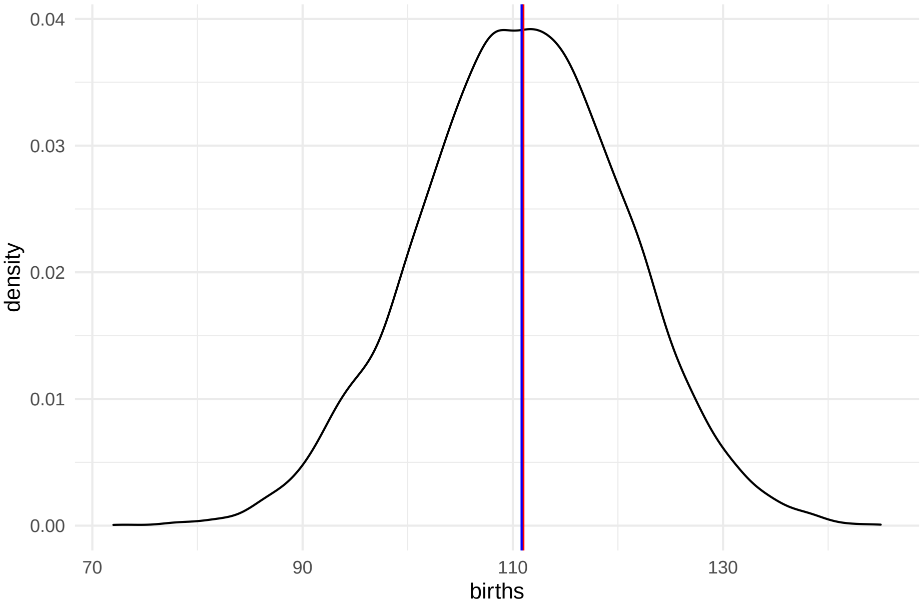
```
##           [,1]      [,2]
## [1,] 0.5005005 0.6126126
```

```
tidybayes::hdi(samples, .width = .97)
```

```
##           [,1]      [,2]
## [1,] 0.4774775 0.6276276
```

3H3. 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?

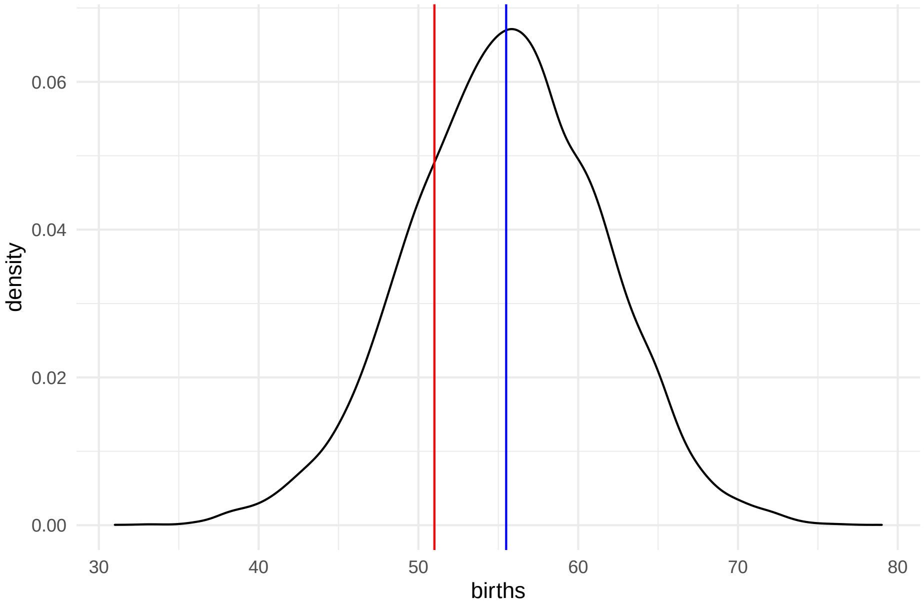
```
simulate <- tibble(births = rbinom( 10000 , size=200 , prob=samples ))

ggplot(simulate) +
  aes(x = births) +
  geom_density() +
  theme_minimal() +
  geom_vline(xintercept = sum(birth1 + birth2), color = "red") +
  geom_vline(xintercept = mean(simulate$births, na.rm = TRUE), color = "blue") +
  labs(caption = "Blue line is mean of simulated data, red line is number of boys from data.")
```



Blue line is mean of simulated data, red line is number of boys from data.

3H4. Now compare 10,000 counts of boys from 100 simulated first-borns only to the number of boys in the first births, birth1. How does the model look in this light?

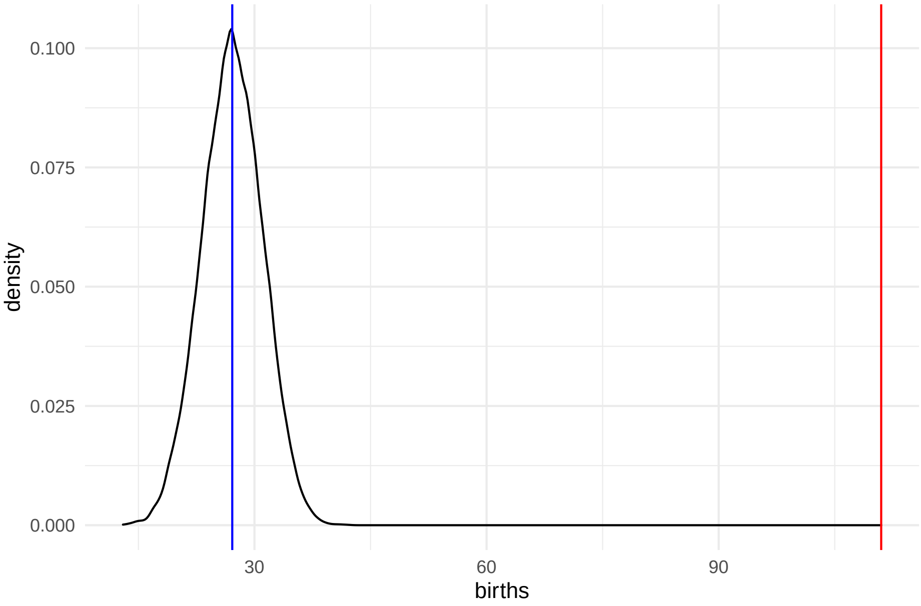
```
simulate <- tibble(births = rbinom( 10000 , size=100 , prob=samples ))

ggplot(simulate) +
  aes(x = births) +
  theme_minimal() +
  geom_density() +
  geom_vline(xintercept = sum(birth1), color = "red") +
  geom_vline(xintercept = mean(simulate$births, na.rm = TRUE), color = "blue") +
  labs(caption = "Blue line is mean of simulated data, red line is number of first-bor
```



Blue line is mean of simulated data, red line is number of first−born boys from data.

3H5. The model assumes that sex of first and second births are independent. To check this assumption, focus now on second births that followed female first borns. Compare 10,000 simulated counts of boys to only those second births that followed girls. To do this correctly, you need to count the number of first borns who were girls and simulate that many births, 10,000 times. Compare the counts of boys in your simulations to the actual observed count of boys following girls. How does the model look in this light? Any guesses what is going on in these data?

```
first_born_girls <- sum(birth1==0)

simulate <- tibble(births = rbinom(10000, size=first_born_girls, prob=samples))

ggplot(simulate) +
  aes(x = births) +
  geom_density() +
  theme_minimal() +
  geom_vline(xintercept = sum(birth1 + birth2), color = "red") +
  geom_vline(xintercept = mean(simulate$births, na.rm = TRUE), color = "blue") +
  labs(caption = "Blue line is mean of simulated data, red line is number of first-born girls fro
```



Blue line is mean of simulated data, red line is number of first−born girls from data.