Here’s my solution to the medium exercises in chapter 3 of McElreath’s Statistical Rethinking, 2nd edition.

Assuming Earth has 70% water cover, and we observe water 8 times out of 15 globe tosses, let’s calculate some posterior quantities with two choices of prior: uniform and step.

p\_true <- 0.7

W <- 8

N <- 15

granularity <- 1000 # points on the grid

**Uniform Prior**

We calculate the grid approximation of the posterior as shown in the book.

m1\_grid <- tibble(p = seq(0, 1, length.out = granularity)) %>%

mutate(prior = 1)

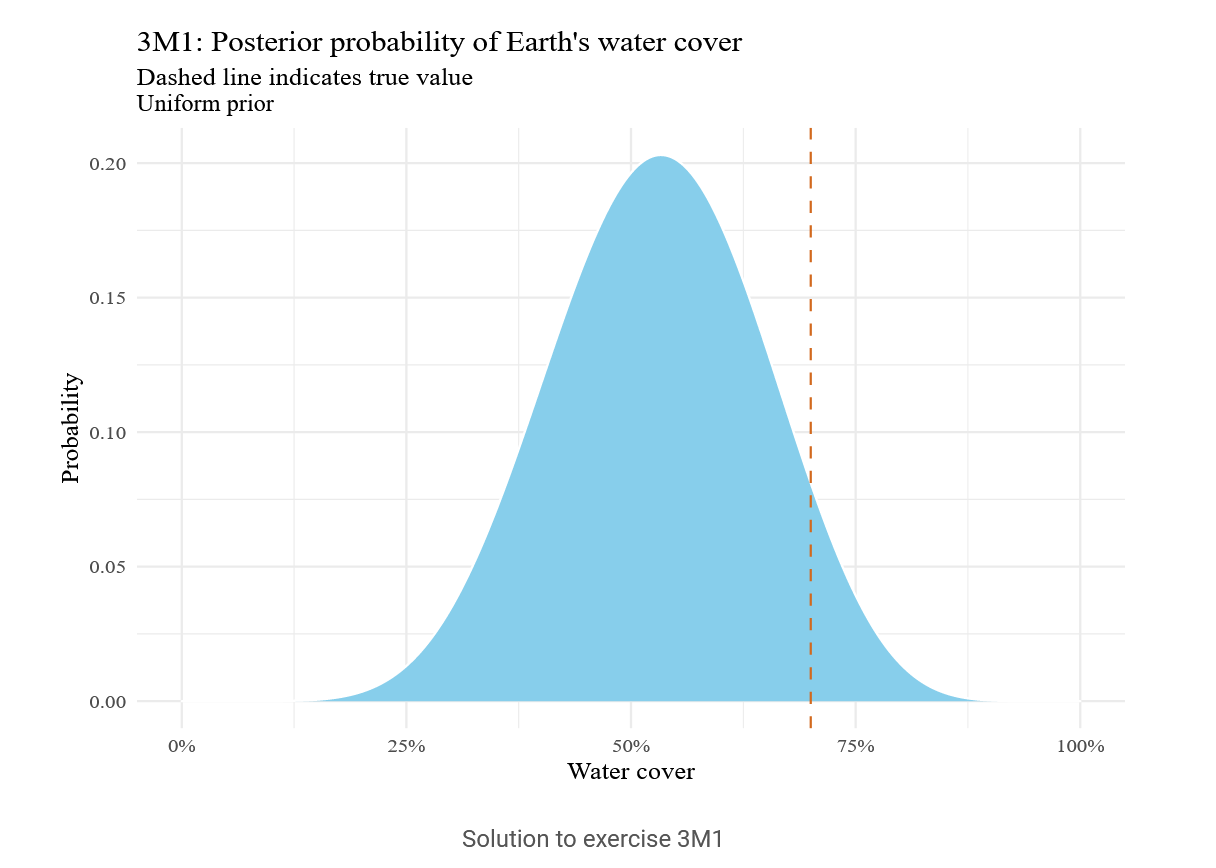
m1\_posterior <- m1\_grid %>%

mutate(

likelihood = dbinom(W, N, p),

posterior = prior \* likelihood

)



Solution to exercise 3M1

We can get draws from our posterior by sampling the water cover values many times with replacement, each value being drawn in proportion to the posterior probability. We can then just summarise these draws to get the desired interval.

m2\_samples <- m1\_posterior %>%

sample\_n(10000, replace = T, weight = posterior)

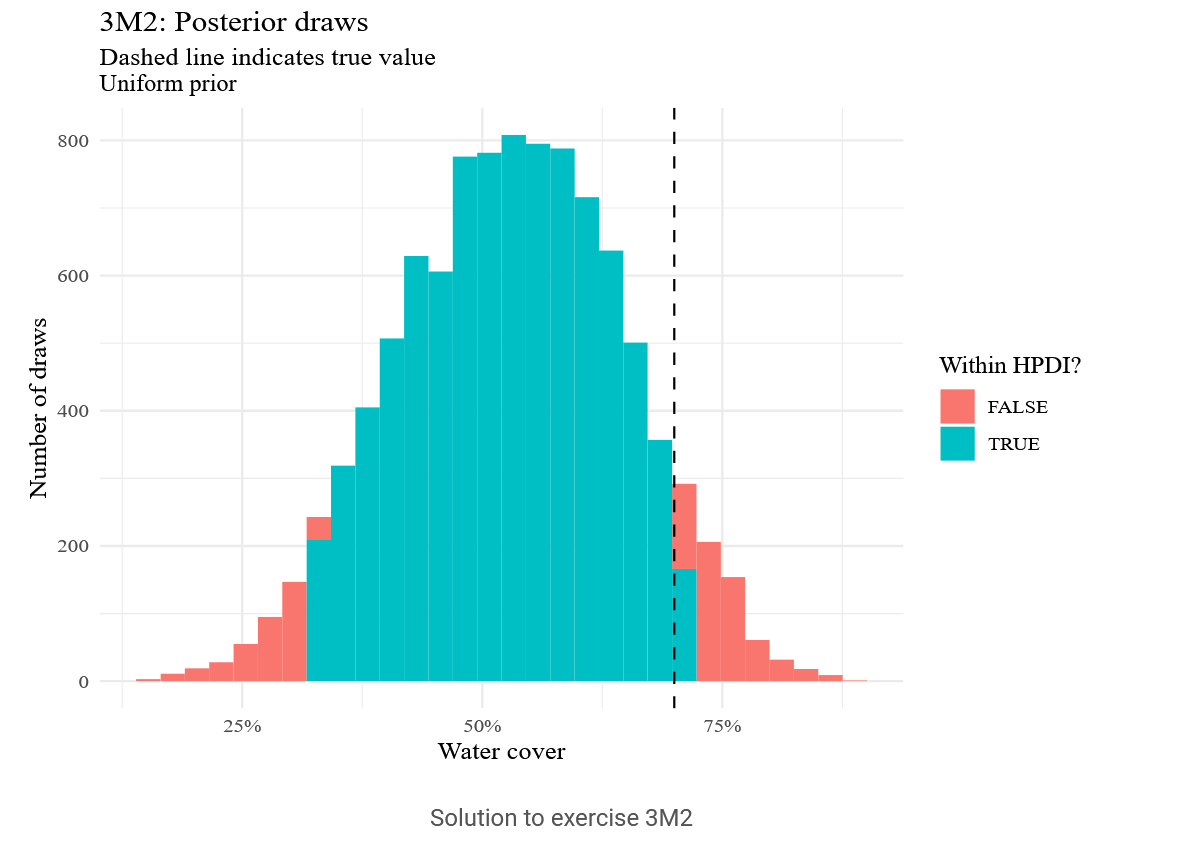
m2\_hpdi <- HPDI(m2\_samples$p, prob = 0.9)

m2\_hpdi

|0.9 0.9|

0.3223223 0.7097097

The histogram looks as follows. This is much the same as the previous graph, but calculated from the samples.



Solution to exercise 3M2

To get the posterior predictive sample, we take our posterior draws of *p*

, then use them to draw a random number of observed water tosses out of 15. The fraction of posterior predictive samples with a given value is then the posterior predictive probability of that value.

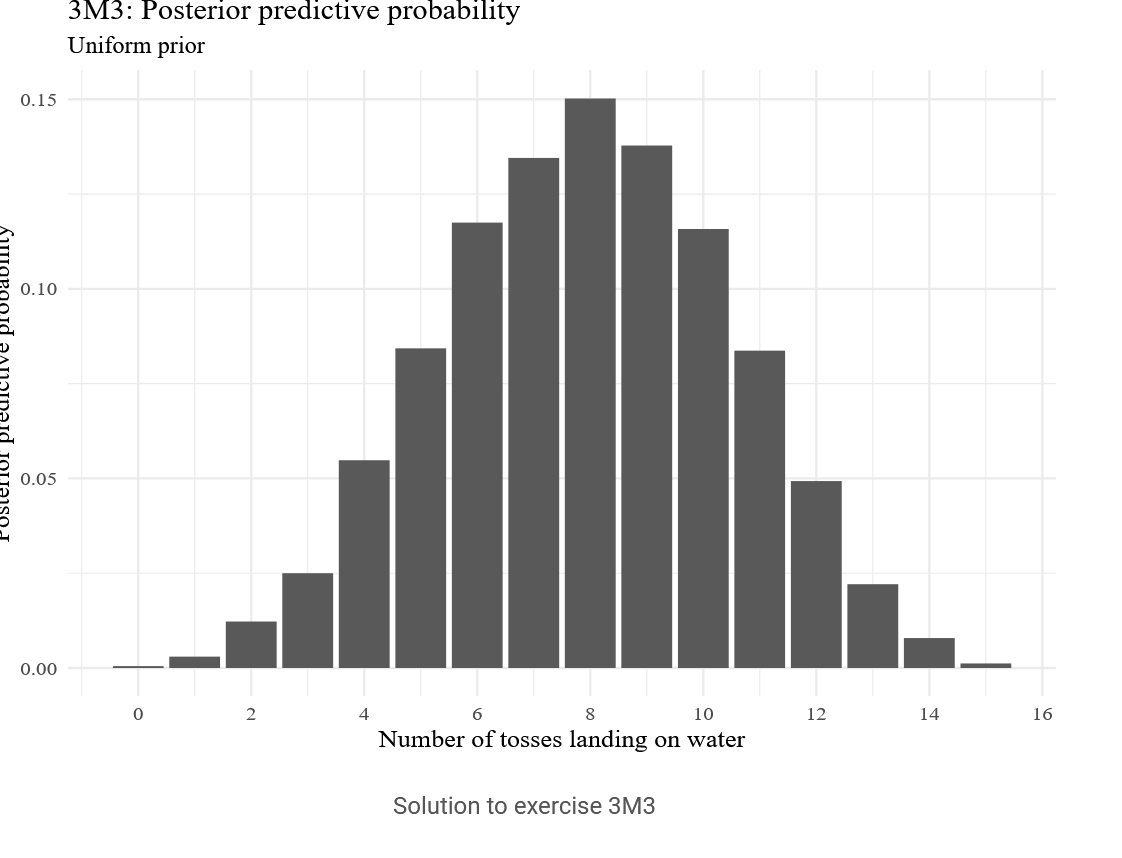
m3\_prob <- m2\_samples %>%

mutate(W = rbinom(n(), 15, p)) %>%

group\_by(W) %>%

tally() %>%

mutate(probability = n / sum(n))



Solution to exercise 3M3

We can also calculate the posterior predictive probabilities with a different number of tosses. Here with 9 tosses.

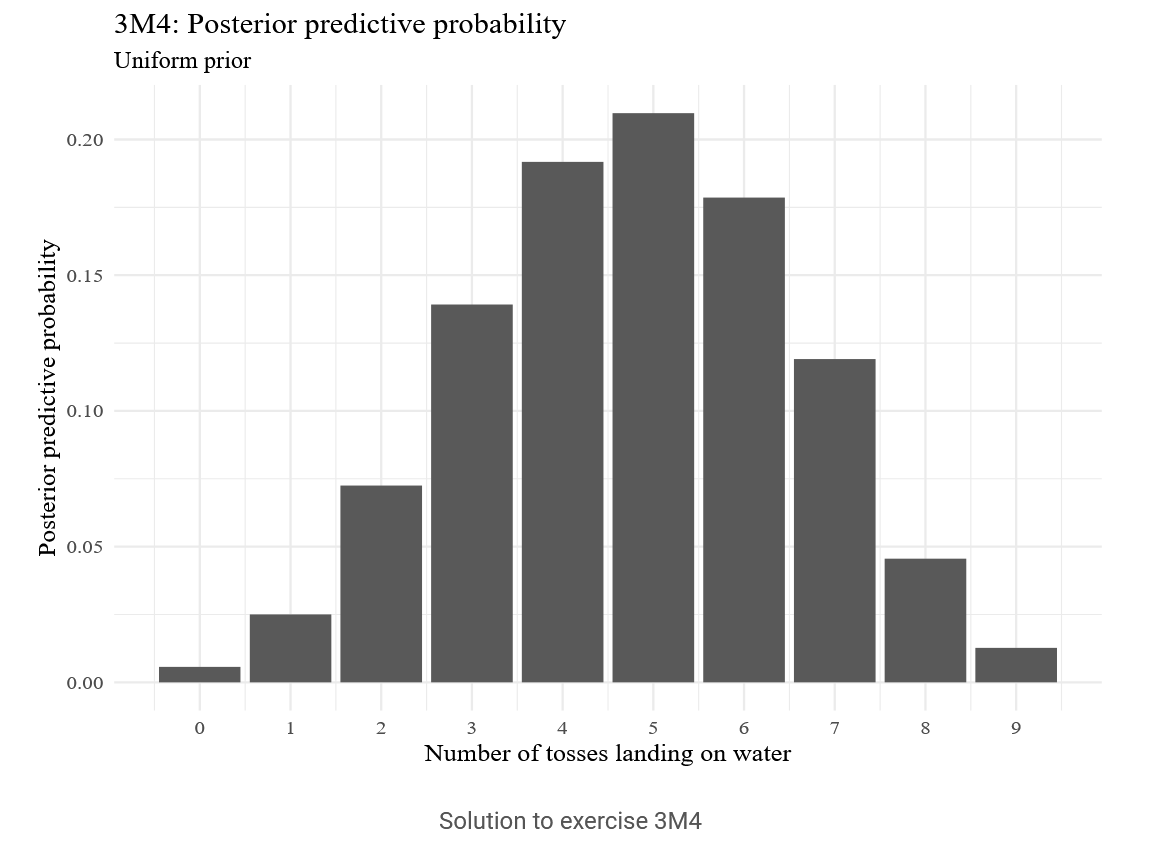
m4\_prob <- m2\_samples %>%

mutate(W = rbinom(n(), 9, p)) %>%

group\_by(W) %>%

tally() %>%

mutate(probability = n / sum(n))



Solution to exercise 3M4

**Step Prior**

Now we repeat the same steps but with the step prior instead of the uniform prior. We’ll just repeat it without comment.

m5\_grid <- m1\_grid %>%

mutate(prior = if\_else(p < 0.5, 0, 1))

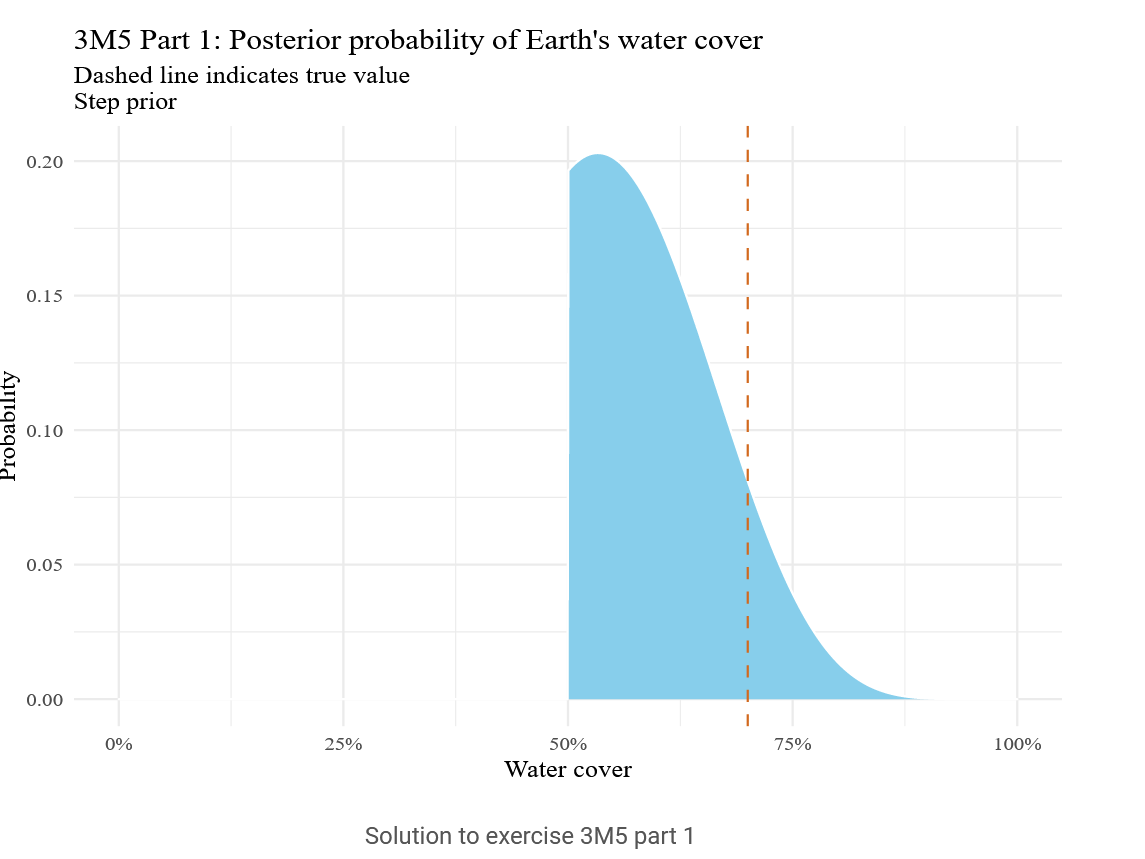
m5\_posterior <- m5\_grid %>%

mutate(

likelihood = dbinom(W, N, p),

posterior = prior \* likelihood

)



Solution to exercise 3M5 part 1

m5\_samples <- m5\_posterior %>%

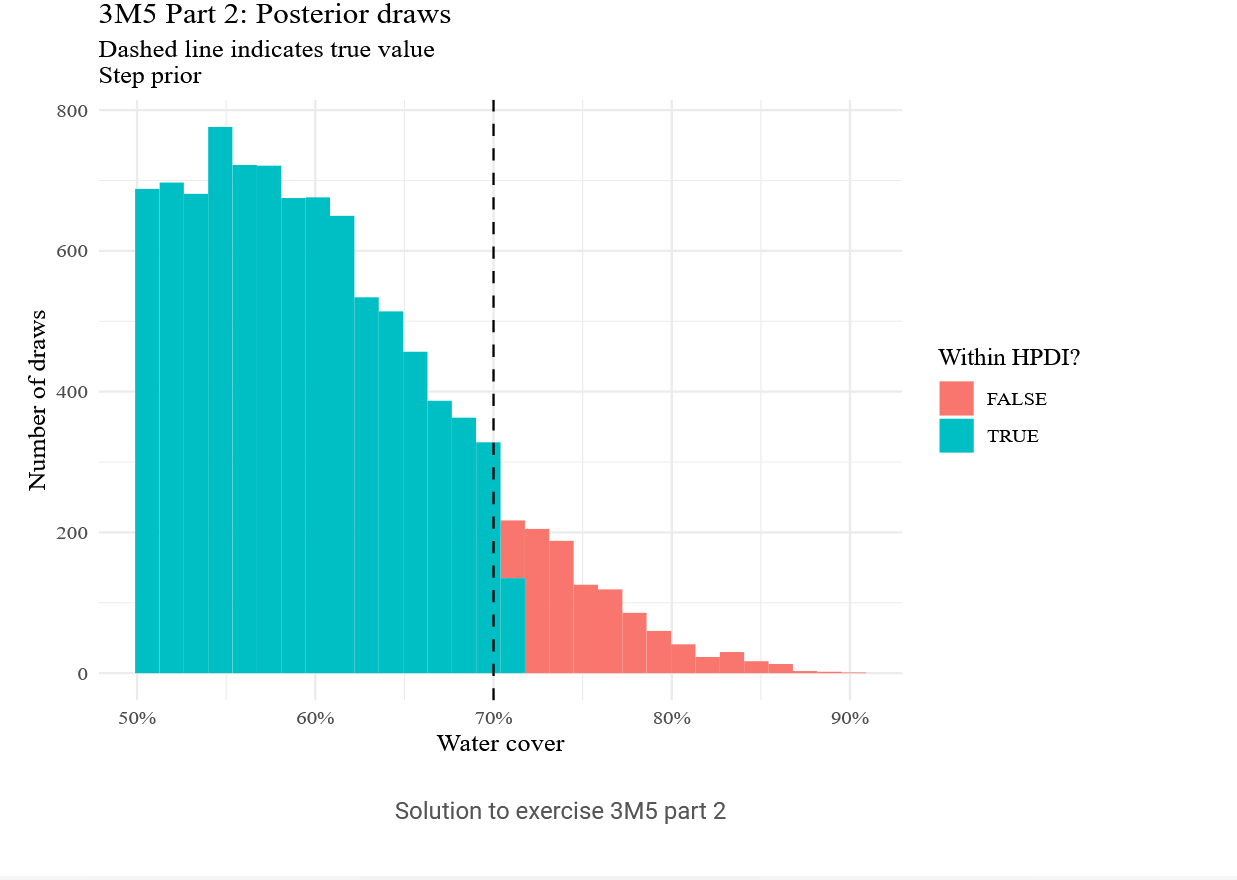
sample\_n(10000, replace = T, weight = posterior)

m5\_hpdi <- HPDI(m5\_samples$p, prob = 0.9)

m5\_hpdi

|0.9 0.9|

0.5005005 0.7107107



Solution to exercise 3M5 part 2

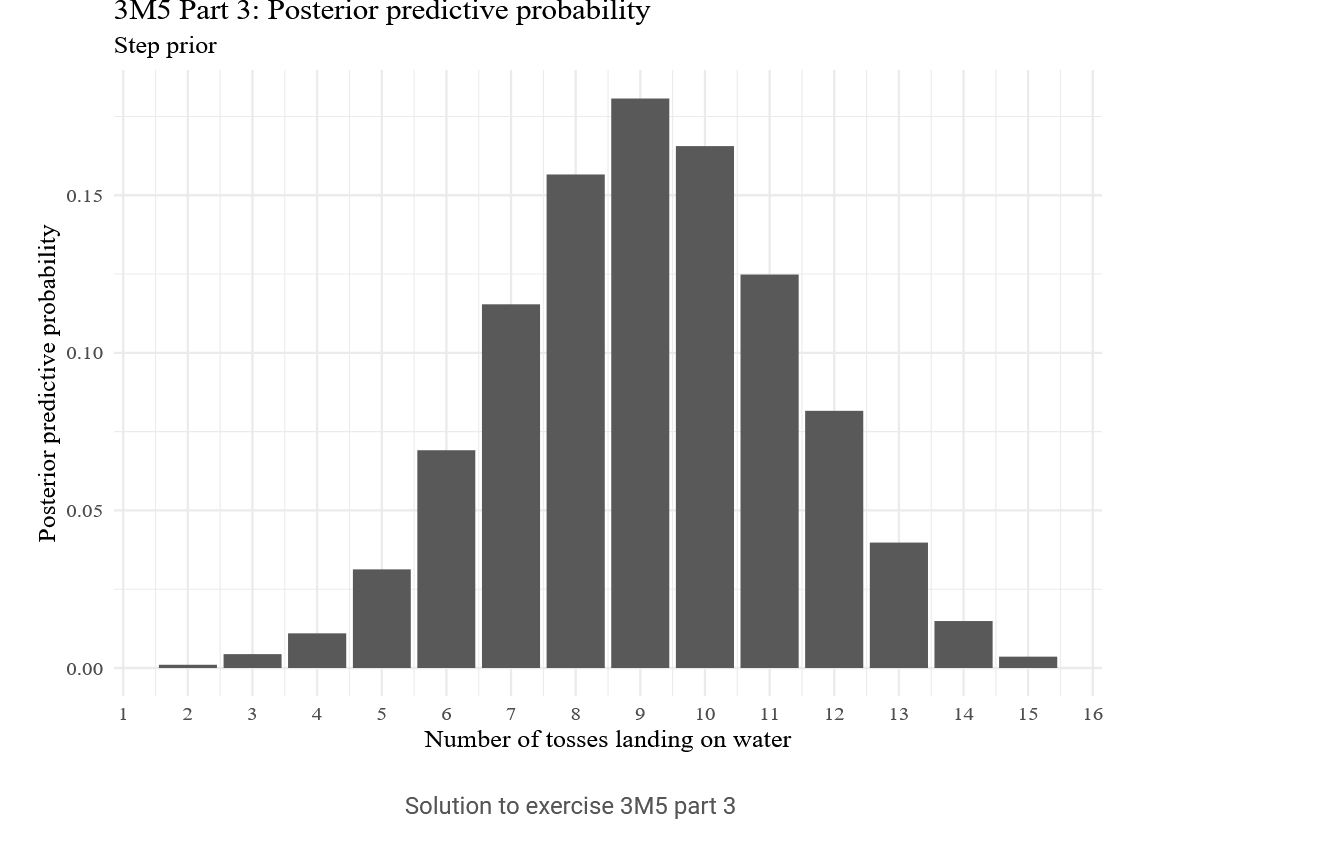
m5\_prob <- m5\_samples %>%

mutate(W = rbinom(n(), 15, p)) %>%

group\_by(W) %>%

tally() %>%

mutate(probability = n / sum(n))



Solution to exercise 3M5 part 3

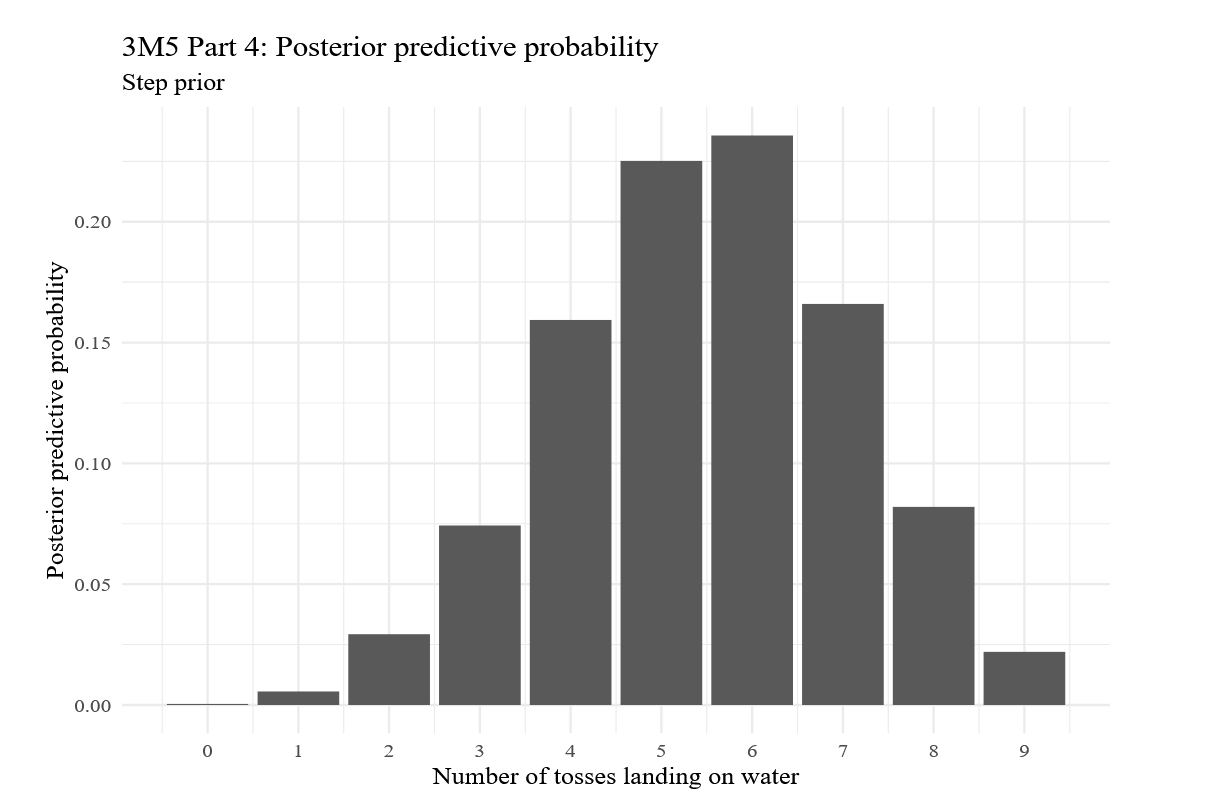
m5\_prob <- m5\_samples %>%

mutate(W = rbinom(n(), 9, p)) %>%

group\_by(W) %>%

tally() %>%

mutate(probability = n / sum(n))



Solution to exercise 3M5 part 4

Let’s compare the proportion of samples within 0.05 of the true value for each prior.

p\_close\_uniform <- m2\_samples %>%

group\_by(close = p %>% between(p\_true - 0.05, p\_true + 0.05)) %>%

tally() %>%

mutate(probability = n / sum(n)) %>%

filter(close) %>%

pull(probability)

p\_close\_step <- m5\_samples %>%

group\_by(close = p %>% between(p\_true - 0.05, p\_true + 0.05)) %>%

tally() %>%

mutate(probability = n / sum(n)) %>%

filter(close) %>%

pull(probability)

The probability of being close to the true value under the uniform and step priors is 0.1316 and 0.2157, respectively. The step prior thus has more mass around the true value.

**Exercise 3M6**

Bayesian models are generative, meaning we can simulate new datasets according to our prior probabilities. We’ll simulate 10 datasets for each value of N of interest. We simulate a dataset by randomly choosing a p\_true from our uniform prior, then randomly choosing a W from the corresponding binomial distribution.

m6\_prior\_predictive <- crossing(

N = 200 \* (1:16),

iter = 1:10

) %>%

mutate(

p\_true = runif(n(), min=0, max=1),

W = rbinom(n(), N, p\_true)

)

For each of these simulated datasets, we grid approximate the posterior, take posterior samples, then calculate the HPDI.

m6\_grid <- tibble(p = seq(0, 1, length.out = granularity)) %>%

mutate(prior = 1)

m6\_posteriors <- m6\_prior\_predictive %>%

crossing(m6\_grid) %>%

group\_by(N, p\_true, iter) %>%

mutate(

likelihood = dbinom(W, N, p),

posterior = prior \* likelihood

)

m6\_samples <- m6\_posteriors %>%

sample\_n(1000, replace = TRUE, weight = posterior)

m6\_hpdi <- m6\_samples %>%

summarise(lo = HPDI(p, 0.99)[1], hi = HPDI(p, 0.99)[2]) %>%

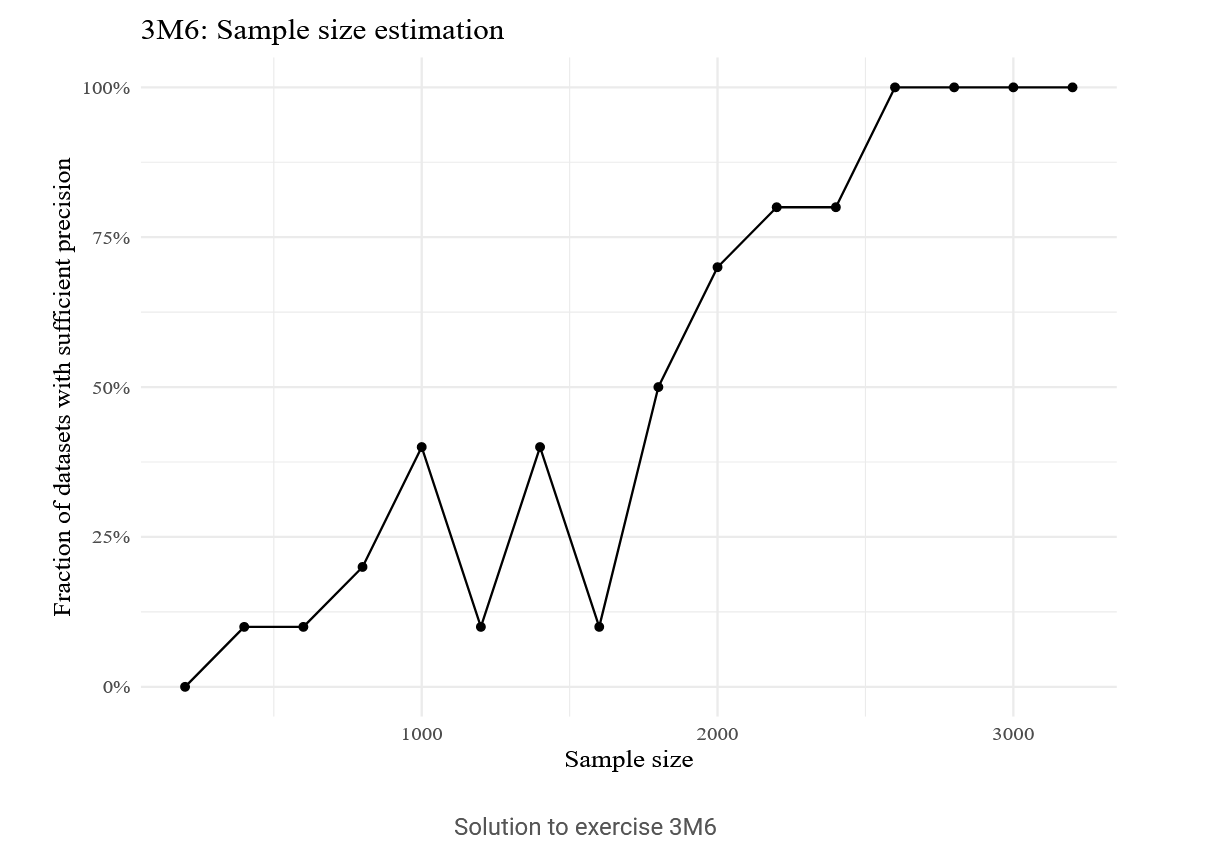
mutate(width = abs(hi - lo))

Now for each value of N, we check how many of the intervals have the desired width.

m6\_n <- m6\_hpdi %>%

group\_by(N) %>%

summarise(fraction = mean(width < 0.05))



Solution to exercise 3M6

Thus we expect a sample size around 2600-3000 to give us a sufficiently precise posterior estimation.