Chapter 3. Sampling the imaginary

19/11/2021

Here we'll solve the medium to hard exercises for the chapter.

```
library(rethinking)
library(magrittr)
library(ggplot2)
library(dplyr)
library(purrr)
library(tidyr)
```

Easy

The easy problems use the samples from the posterior distribution for the globe tossing example. This code will give you a specific set of samples, so that you can check your answers exactly

```
p_grid <-  seq( from = 0, to =1 , length.out = 1000)
prior <- rep(1, 1000)
likelihood <- dbinom( 6, size=9, prob=p_grid) # Assume 6 water in 9 tosses
posterior <- likelihood * prior
posterior <- posterior / sum(posterior)

set.seed(100)
samples <- sample(p_grid , prob = posterior, size = 1e4, replace = TRUE)
```

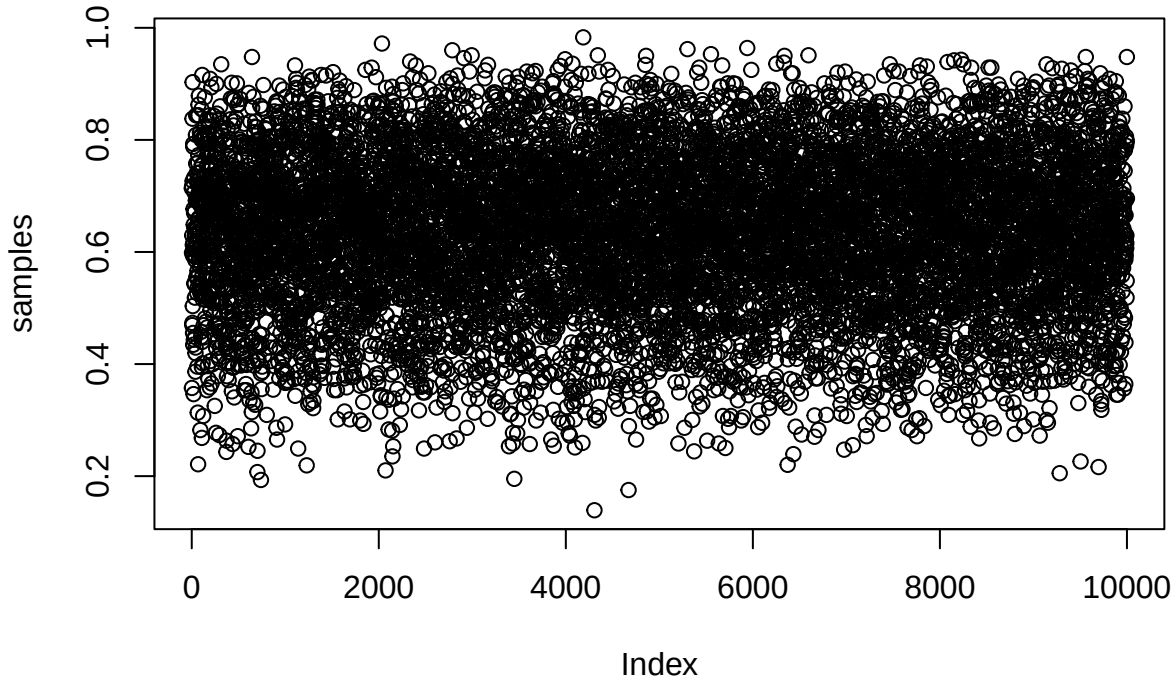
Use the values in samples to answer the questions that follow.

3E1

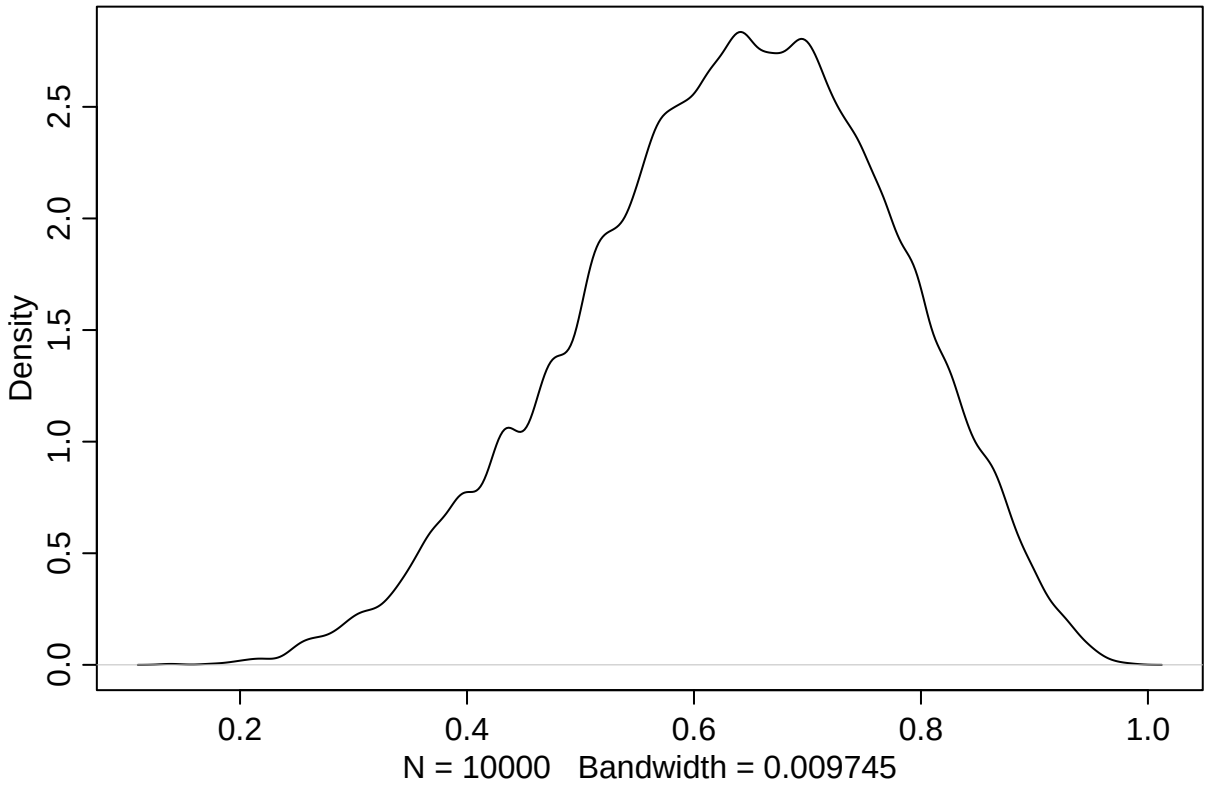
How much posterior probability lies below p = 0.2?

Answer 4e-04, or 0.0004

```
plot(samples)
```




```
dens(samples)
```




```
sum( samples < 0.2 ) / 1e4
```

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

3E2

How much posterior probability lies above p = 0.8?

Answer 0.1116

```
sum( samples > 0.8 ) / 1e4
```

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

3E3

How much posterior probability lies above between p = 0.02 and p = 0.8?

Answer 0.888

```
sum( samples > 0.2 & samples < 0.8 ) / 1e4
```

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

3E4

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

Answer: 0.5185

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

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

3E5

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

Answer: 0.2442

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

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

3E6

Which values of p contain the narrowest interval equal to 66% of the posterior probability?

Answer: 0.5085 and 0.7738

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

```
##     |0.66     0.66|
## 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?

Answer: 0.5025 and 0.7698

```
quantile(samples, c(0.17, 0.83)) # Classic way
```

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

```
PI(samples, prob = 0.66) # McElreath's way
```

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

Medium

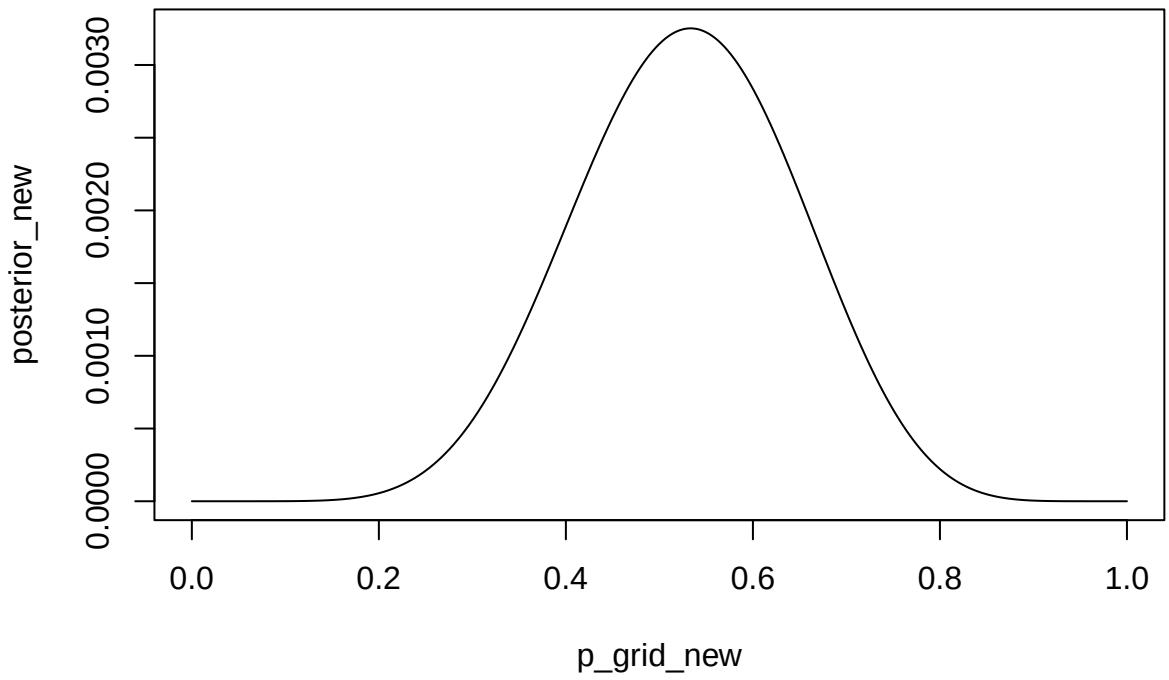
3M1

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.

Answer

Let's do as before, but changing the relevant observed data

```
p_grid_new <-  seq( from = 0, to =1 , length.out = 1000)
prior_new <- rep(1, 1000)
likelihood_new <- dbinom( 8, size=15, prob=p_grid_new) # Assume 8 water in 15 tosses
posterior_new <- likelihood_new * prior_new
posterior_new <- posterior_new / sum(posterior_new)
plot( posterior_new ~ p_grid_new , type="l" )
```



3M2

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

Answer: 0.3293 and 0.7167

```
samples_new <- sample(p_grid_new , prob = posterior_new, size = 1e4, replace = TRUE)
HPDI(samples = samples_new, prob = 0.9)
```

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

3M3

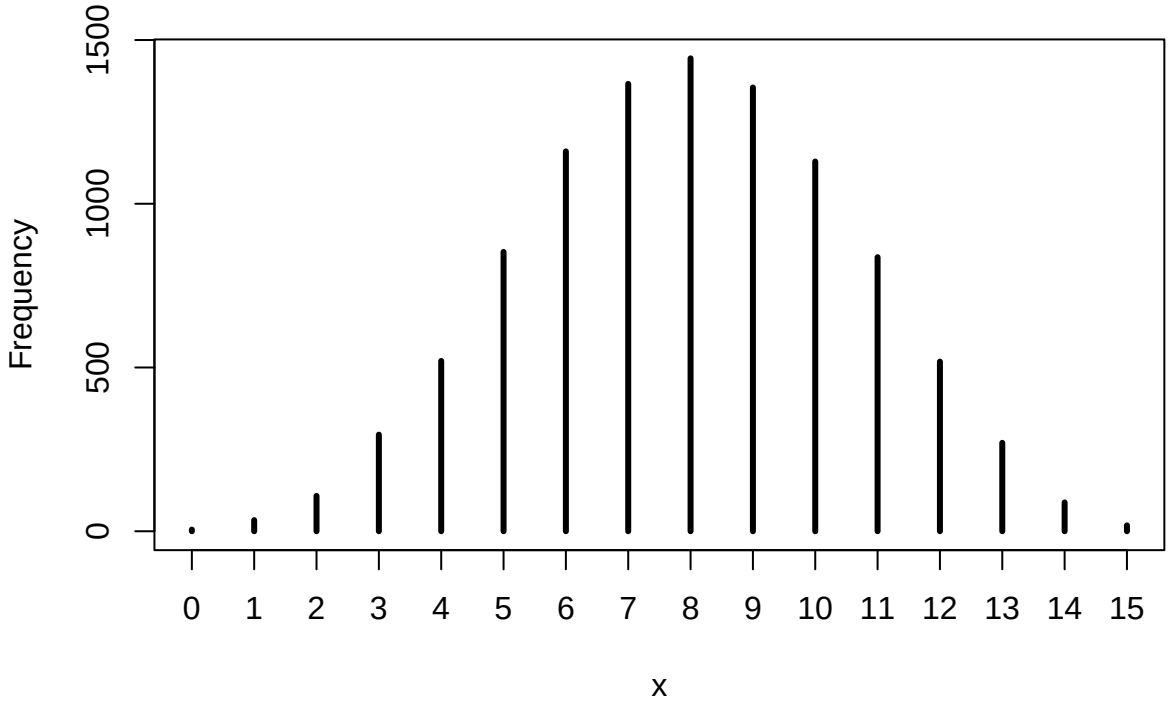
Construct a posterior predictive check for this model and data. This means simulate the distribution of samples, averaging over the posterior uncertainty of p. What is the probability of observing 8 water in 15 tosses?

Answer: 0.1409

```
w <- rbinom(1e4, size=15, prob = samples_new)
sum(w==8) / 1e4
```

```
## [1] 0.1444
```

```
simplehist(w)
```



3M4

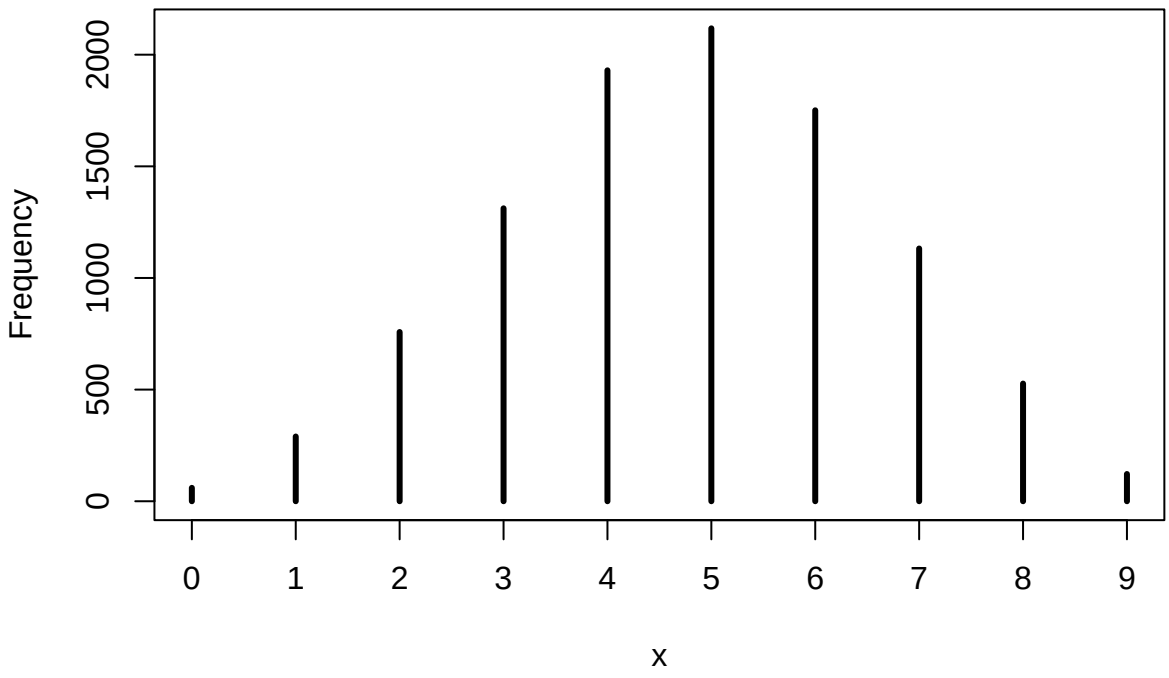
Using the posterior distribution constructed from the new (8/15) data, now calculate the probability of observing 6 water in 9 tosses

Answer: 0.1806

```
w <- rbinom(1e4, size=9, prob = samples_new)
sum(w==6) / 1e4
```

```
## [1] 0.1751
```

```
simplehist(w)
```

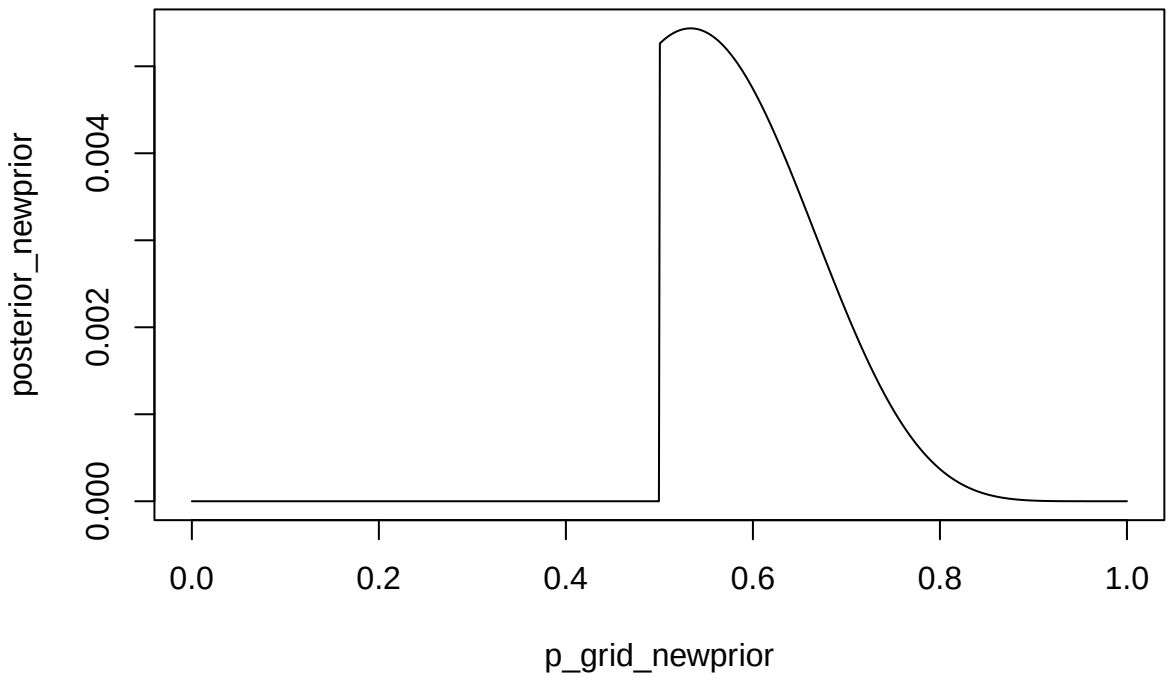


3M5

Start over at **3M1**, but 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 differences. What difference does the better prior make? If it helps, compare inferences (using both priors) to the true value p = 0.7.

Answer:

```
p_grid_newprior <-  seq( from = 0, to =1 , length.out = 1000)
prior_newprior <- rep(1, 1000)
prior_newprior[p_grid_newprior < 0.5] <- 0 # Switch to zero for grids below 0.5
likelihood_newprior <- dbinom( 8, size=15, prob=p_grid_newprior) # Assume 8 water in 15 tosses
posterior_newprior <- likelihood_newprior * prior_newprior
posterior_newprior <- posterior_newprior / sum(posterior_newprior)
plot( posterior_newprior ~ p_grid_newprior , type="l" )
```



Compute and compare

```
# Flat prior
HPDI(samples = samples_new, prob = 0.9)
```

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

```
# New prior
samples_newprior <- sample(p_grid_newprior , prob = posterior_newprior, size = 1e4, replace = TRUE)
HPDI(samples = samples_newprior, prob = 0.9)
```

```
##      |0.9      0.9|
## 0.5005005 0.7117117
```

Now posterior distribution is way narrower, since we're constraining our prior. While it doesn't change much on the upper side, it does remove a good deal on the lower side.

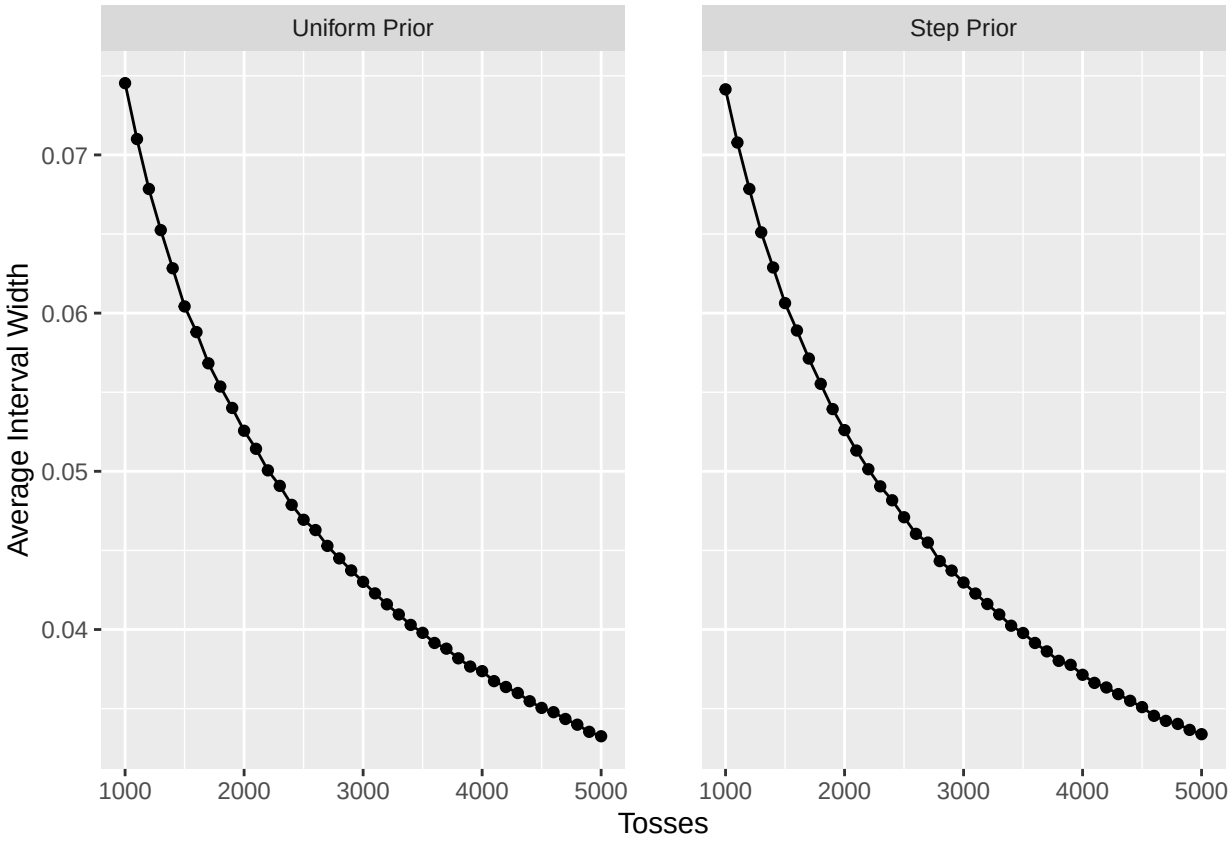
3M6

Suppose you want to estimate 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?

Answer:

Code borrowed from Jake Thompson's website.

```r
# We create some functions to run the simulations
## Run each simulation and compute the interval width
single_sim <- function(tosses, prior_type = c("uniform", "step")) {
  prior_type <- match.arg(prior_type)
  obs <- rbinom(1, size = tosses, prob = 0.7) # Here we extract one sample from a distribution with pro

  p_grid <- seq(from = 0, to = 1, length.out = 1000)
  prior <- rep(1, 1000)
  if (prior_type == "step") prior[1:500] <- 0

  likelihood <- dbinom(obs, size = tosses, prob = p_grid)
  posterior <- likelihood * prior
  posterior <- posterior / sum(posterior)

  samples <- sample(p_grid, prob = posterior, size = 1e4, replace = TRUE) # Sample from the posterior
  interval <- PI(samples, prob = 0.99)
  width <- interval[2] - interval[1]
}

# Create a table with the tosses and the prior type.
single_cond <- function(tosses, prior_type, reps = 100) {
  tibble(tosses = tosses,
         prior_type = prior_type,
         width = map_dbl(seq_len(reps), ~single_sim(tosses = tosses,
                                                    prior_type = prior_type)))
}

# Run simulations for each condition
simulation <- crossing(tosses = seq(1000, 5000, by = 100),
                       prior_type = c("uniform", "step")) %>%
  pmap_dfr(single_cond, reps = 100) %>%
  group_by(tosses, prior_type) %>%
  summarize(avg_width = mean(width), .groups = "drop") %>%
  mutate(prior_type = case_when(prior_type == "uniform" ~ "Uniform Prior",
                                prior_type == "step" ~ "Step Prior"),
         prior_type = factor(prior_type, levels = c("Uniform Prior",
                                                    "Step Prior")))

# Plot the results
ggplot(simulation, aes(x = tosses, y = avg_width)) +
  facet_wrap(~prior_type, nrow = 1) +
  geom_point() +
  geom_line() +
  labs(x = "Tosses", y = "Average Interval Width") +
  theme(panel.spacing.x = unit(2, "lines"))
```



Following these simulations, to get an interval width of 0.05 or smaller we'd need to toss the globe around 2,300 times.

Hard

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

```
data(homeworkch3)
```

We can compute the total number of boys born across all of these births

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

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

3H1

Using 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?

Answer:

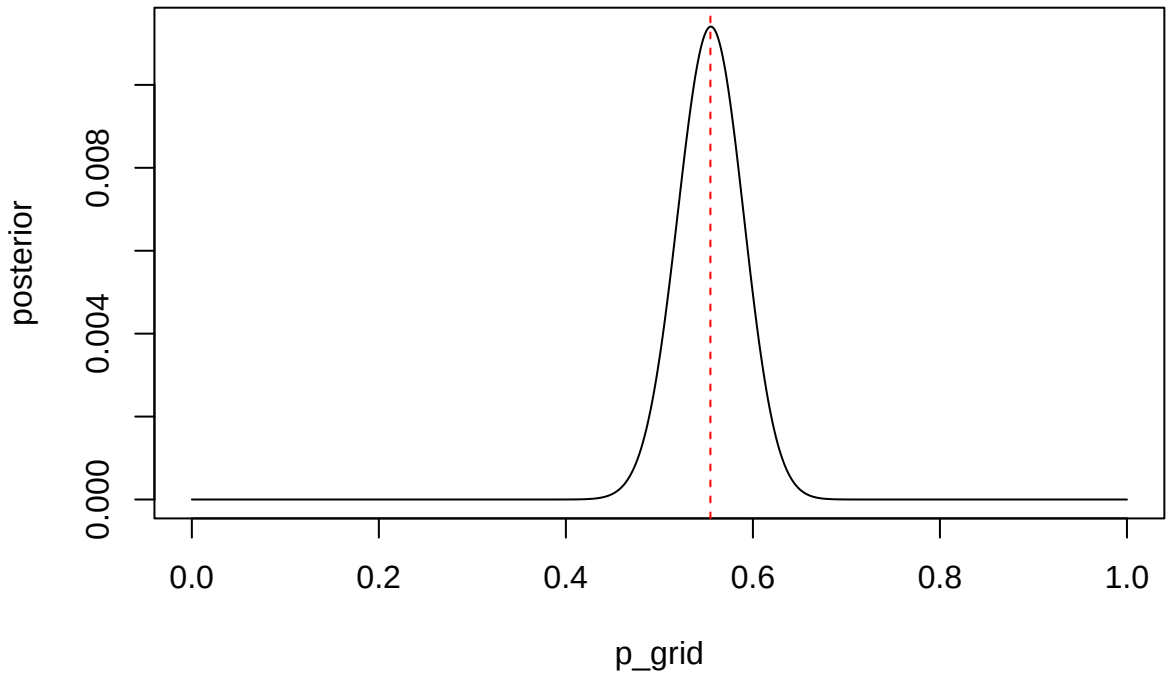
```
boys <- sum(birth1) + sum(birth2)
tot_births <- length(c(birth1, birth2))

p_grid <-  seq( from = 0, to =1 , length.out = 1000)
prior <- rep(1, 1000)
likelihood <- dbinom( boys, size=tot_births, prob=p_grid) # Assume 6 water in 9 tosses
```

```
posterior <- likelihood * prior
posterior <- posterior / sum(posterior)

loss <- sapply(p_grid, function(d) sum(posterior * abs(d - p_grid)))
p_grid[which.min(loss)]
```

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

So p = 0.55 naximises the posterior probability, which matches the highest point on the distribution

```
plot( posterior ~ p_grid , type="l" )
abline(v = p_grid[which.min(loss)], col="red", lty=2)
```



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.

Answer:

```
samples <- sample(p_grid , prob = posterior, size = 1e4, replace = TRUE)
HPDI(samples = samples, prob = 0.5)
```

```
##      |0.5      0.5|
## 0.5265265 0.5725726
```

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

```
##     |0.89     0.89|
```

```
## 0.4964965 0.6076076
```

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

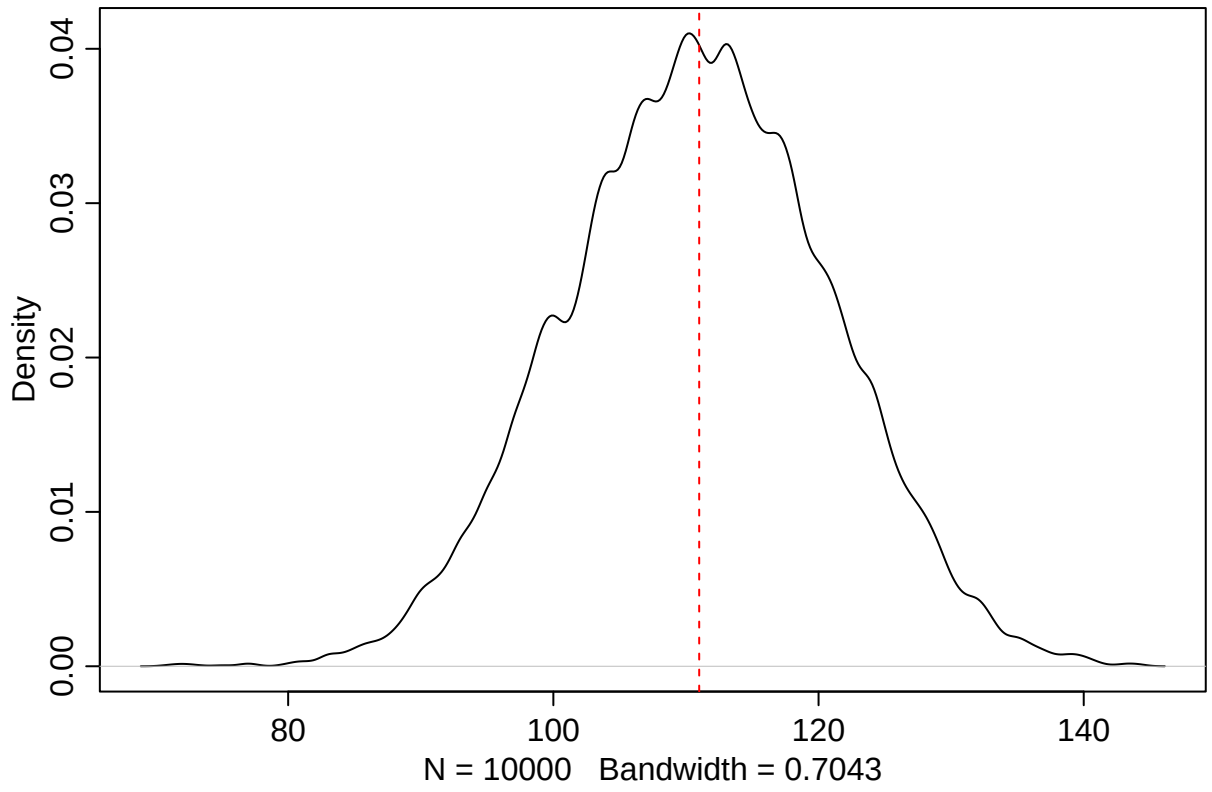
```
##     |0.97     0.97|
## 0.4754755 0.6266266
```

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 distributions 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 nclude the actual obervation as a central likely outcome?

Answer:

```
sims <- rbinom(10000, size = 200, prob = samples)
dens(sims)
abline(v = 111, col ="red", lty=2)
```



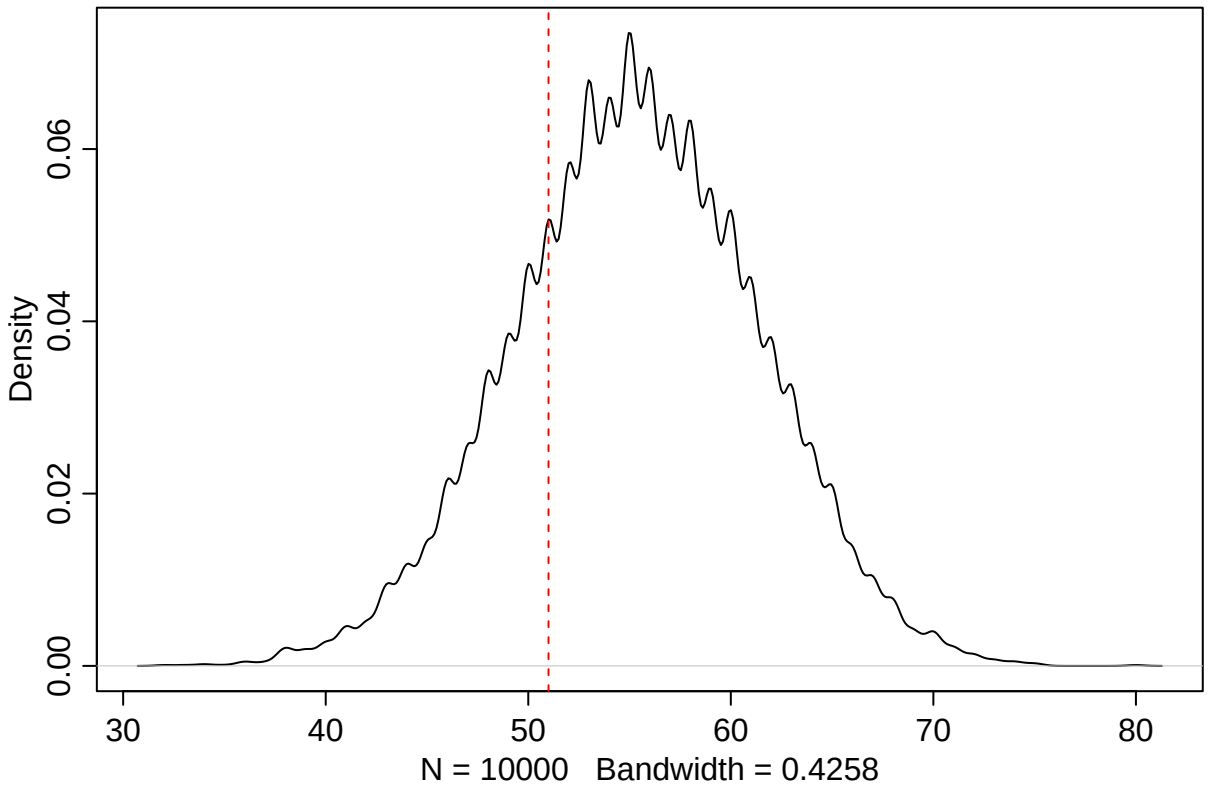
Yes, it looks about right!

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?

Answer:

```
boys_1 <- sum(birth1)
sims <- rbinom(10000, size = 100, prob = samples)
dens(sims)
abline(v = boys_1, col="red", lty=2)
```



Here the model seems a bit off. The model seem to overestimate the real count (51), the median of the model prediction being 56. This might be due to a higher proportion of boys born second.

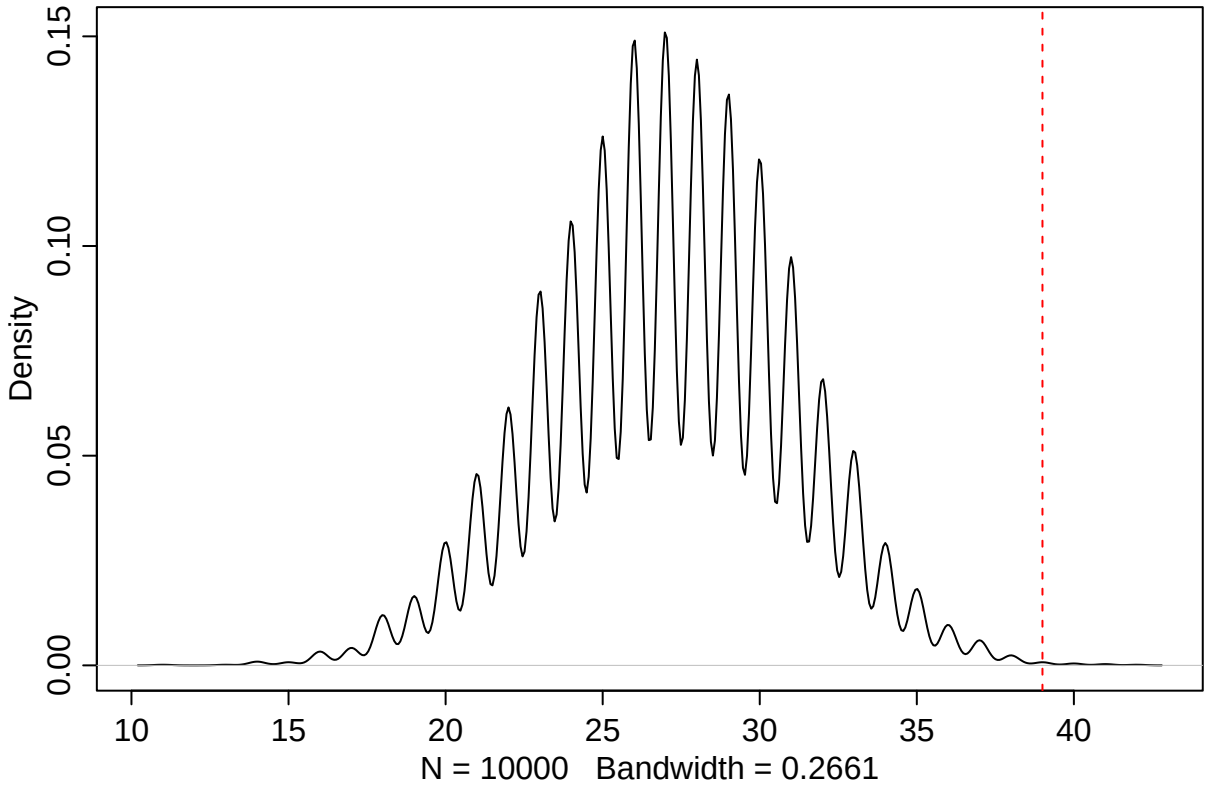
3H5

The model assumes that sex 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 atual observed count of boys following girls. How does the model look in this light? Any guesses what is going on in these data?

Answer:

```
girls_1st <- which(birth1 == 0)
boys_following_girls <- birth2[ girls_1st ]
boys_following_girls <- boys_following_girls[ boys_following_girls == 1]
boys_following_girls <- length(boys_following_girls) # 39

sims_bfg <- rbinom(10000, size = length(girls_1st), prob = samples)
dens(sims_bfg)
```

```r
abline(v = boys_following_girls, col = "red", lty =2)
```



Now the model underestimate the number of boys born after a girl, with a median = 27 boys compared to the actual 39 boys after a girl.

What seems to be happening is that boys are overrepresented as second children, being more likely to be born after a girl.

Hence, our assumption that births are independent seems to be violated.