Statistical Rethinking - Problem Set Solutions

Tyler Burch

Chapter 2 problems

2E1.

> Which of the expressions below correspond to the statement:the probability of rain on Monday?

1. Pr(rain)
2. **Pr(rain|Monday)**
3. Pr(Monday|rain)
4. Pr(rain,Monday)/Pr(Monday)

2E2.

> 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.

2E3.

> Which of the expressions below correspond to the statement: the probability that it is Monday, given that it is raining?

1. **Pr(Monday|rain)**
2. Pr(rain|Monday)
3. Pr(rain|Monday)Pr(Monday)
4. Pr(rain|Monday)Pr(Monday)/Pr(rain)
5. Pr(Monday|rain)Pr(rain)/Pr(Monday)

2E4.

> The Bayesian statistician Bruno de Finetti (1906–1985) began his book on probability theory with the declaration: "PROBABILITY DOES NOT EXIST." The capitals appeared in the original, so I imagine de Finetti wanted us to shout this statement. What he meant is that probability is a device for describing uncertainty from the perspective of an observer with limited knowledge; it has no objective reality. Discuss the globe tossing example from the chapter, in light of this statement. What does it mean to say "the probability of water is 0.7"?

Based on the tosses that we've performed, on the globe we hold, we can expect 70% of future tosses to also land on water. The source of the uncertainty and limited knowledge is that the point we're landing on is random.

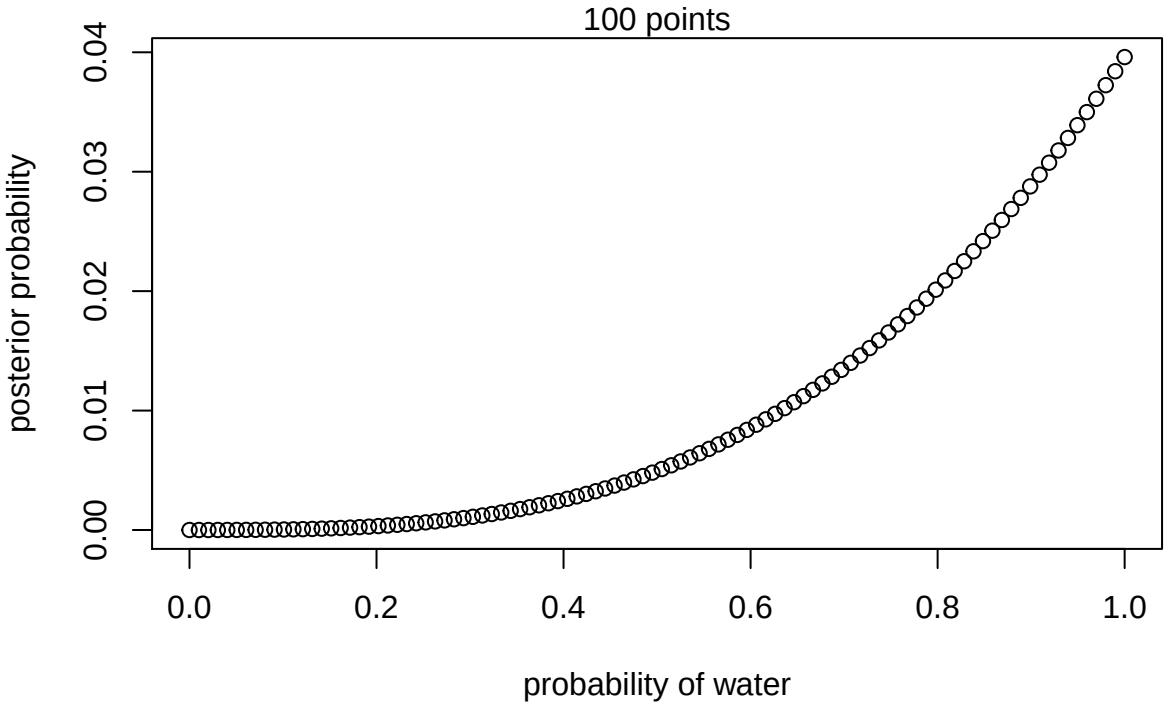
2M1.

> 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.

```r
globe_grid <- function(trial_list, grid_size){
  # define grid
  p_grid <- seq( from=0 , to=1 , length.out=grid_size )
  # define prior
  prior <- rep( 1 , grid_size )
  # compute likelihood at each value in grid
  likelihood <- dbinom( sum(trial_list) , size=length(trial_list) , 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(p_grid , posterior , type="b" ,
       xlab="probability of water" , ylab="posterior probability")
  mtext( sprintf("%i points", grid_size ))
}
```

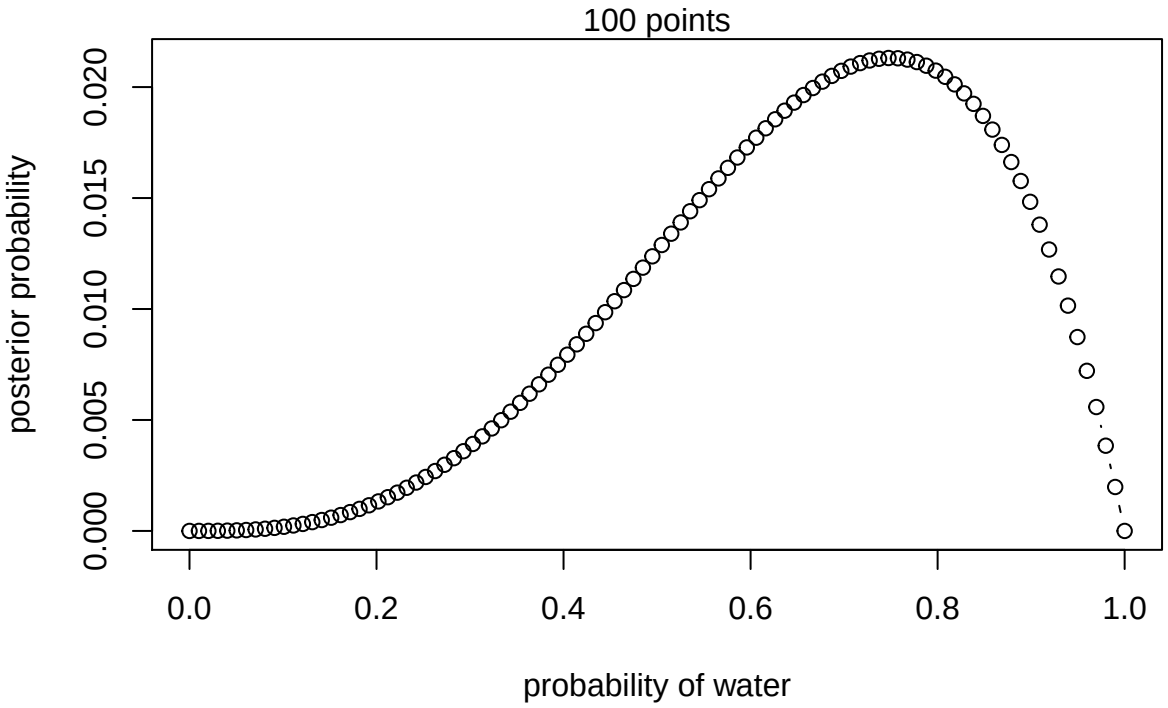
1. W, W, W

```r
globe_grid(c(1,1,1),100)
```



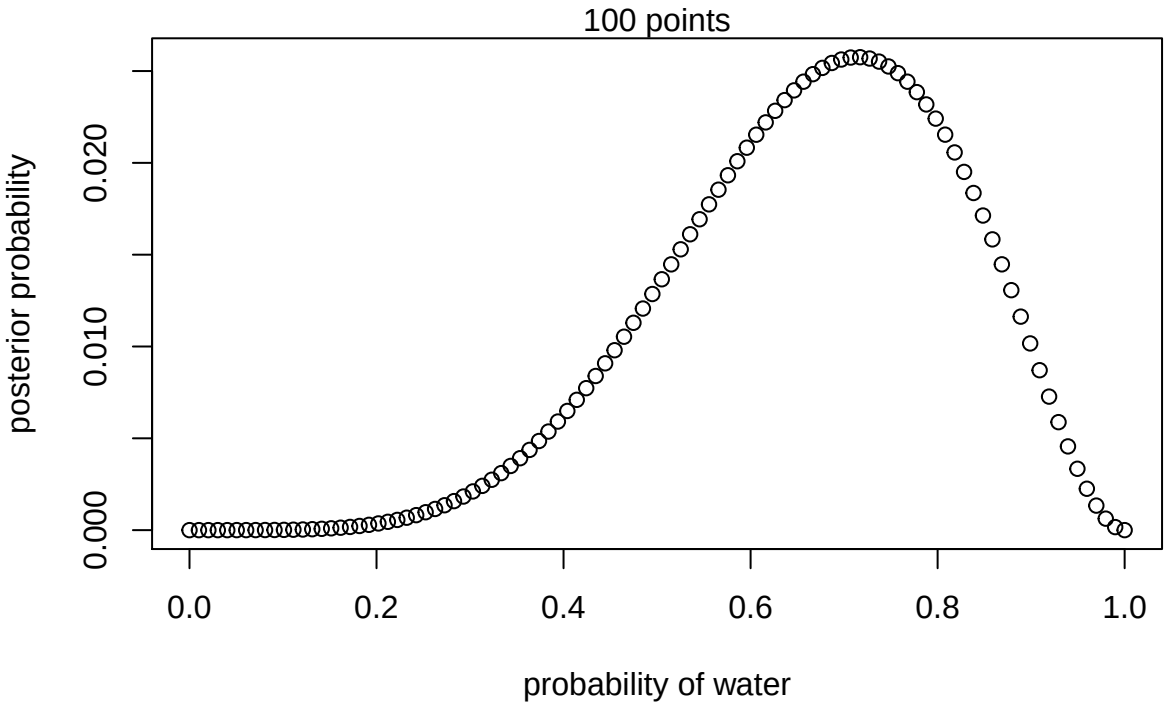
2. W, W, W, L

```r
globe_grid(c(1,1,1,0),100)
```



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

```
globe_grid(c(0,1,1,0,1,1,1),100)
```



2M2.

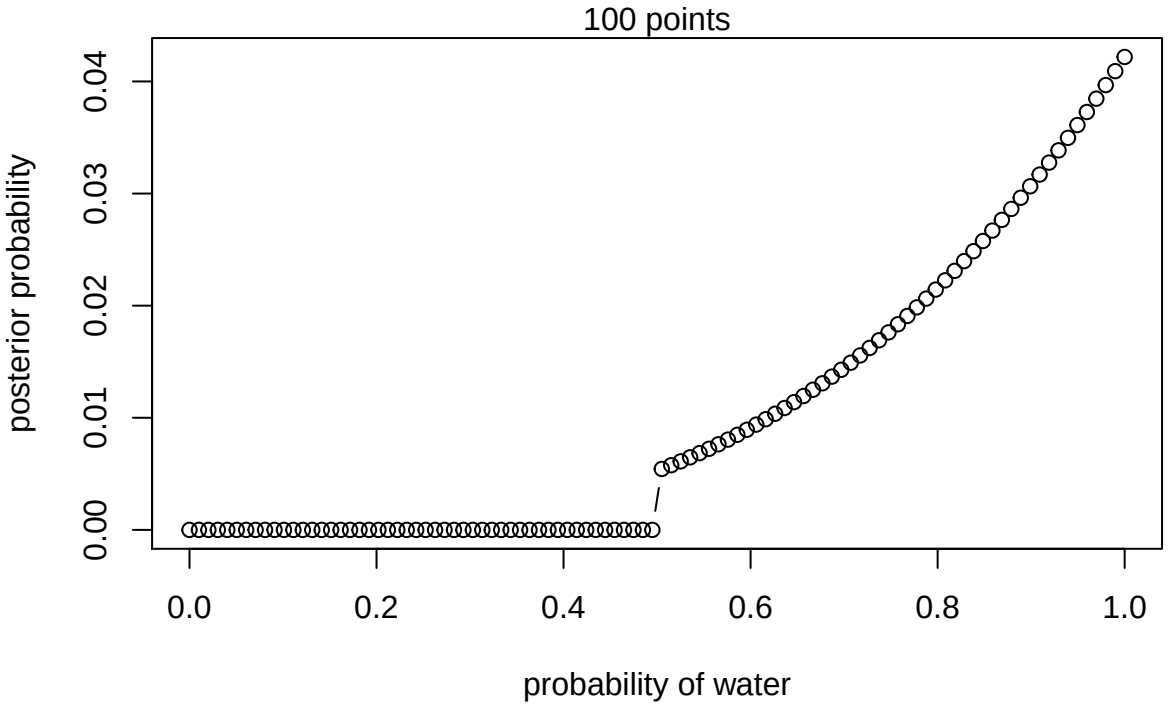
> Now assume a prior for p that is equal to zero when $p < 0.5$ and is a positive constant when $p \geq 0.5$. Again compute and plot the grid approximate posterior distribution for each of the sets of observations in the problem just above.

Change function prior definition to:

```
# define prior
prior <- (p_grid >= 0.5) * prior_const
```

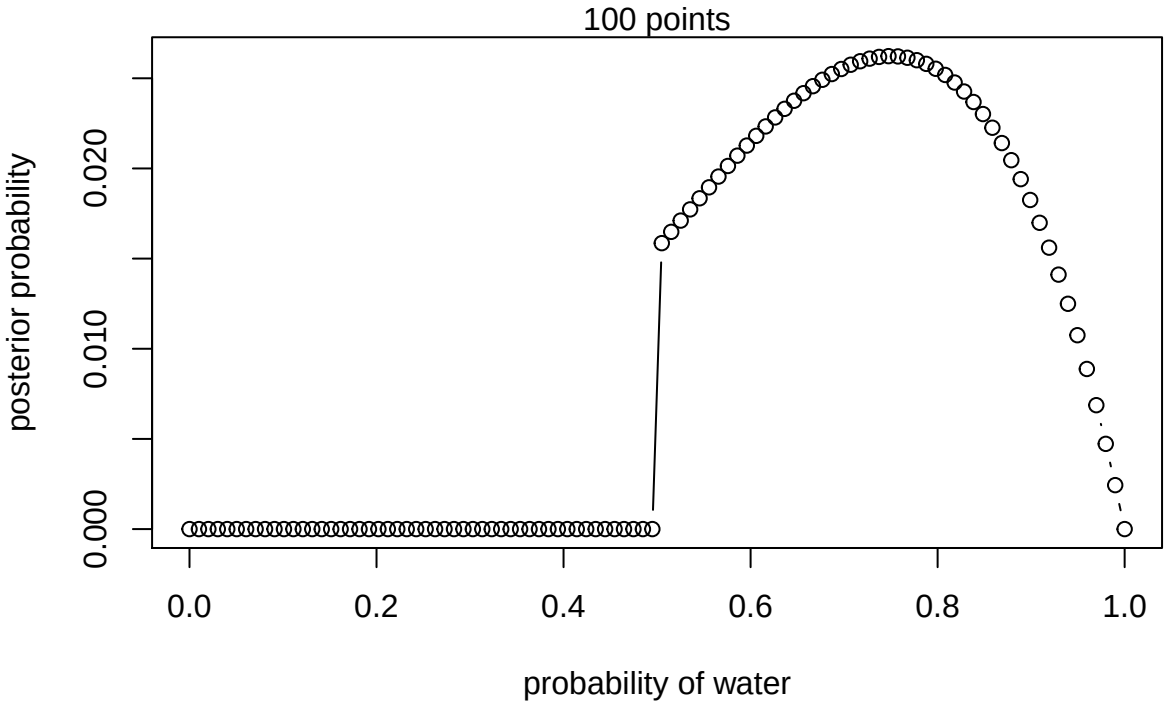
1. W, W, W

```
globe_grid_const(c(1,1,1),100, 1.0)
```



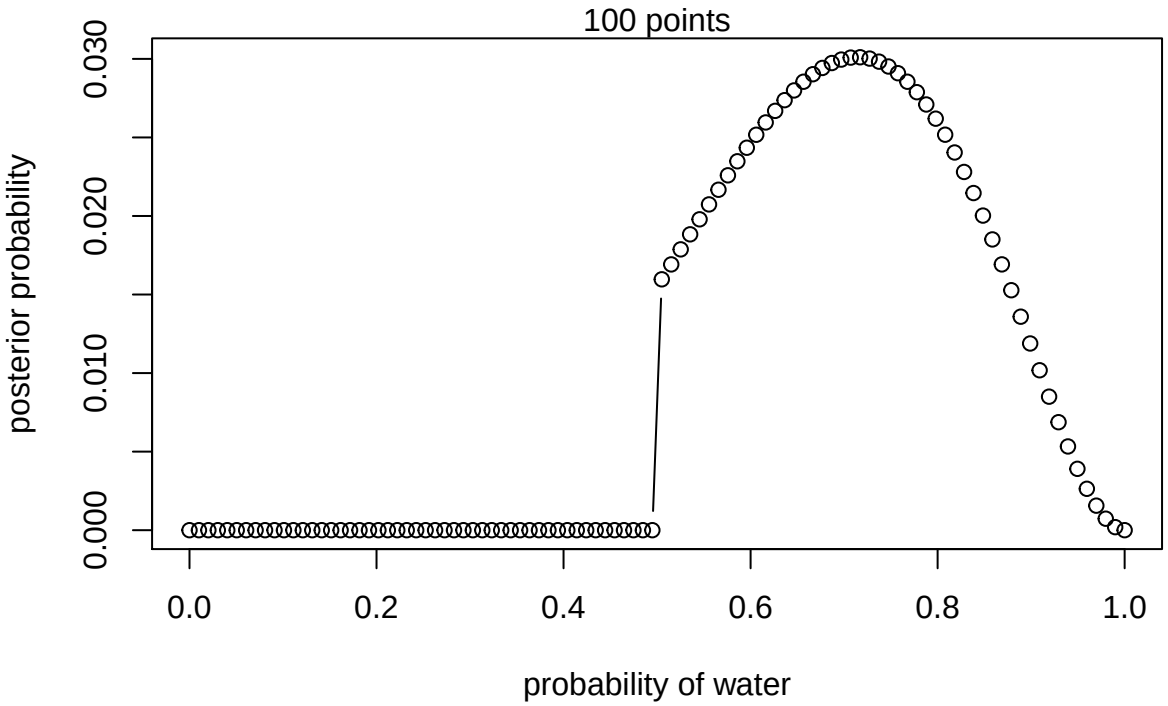
2. W, W, W, L

```
globe_grid_const(c(1,1,1,0),100, 1.0)
```



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

```
globe_grid_const(c(0,1,1,0,1,1,1),100,1)
```



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.

First find total probability of land:

$$\Pr(\text{land}) = \Pr(\text{land}|\text{Earth})\Pr(\text{Earth}) + \Pr(\text{land}|\text{Mars})\Pr(\text{Mars}) = (0.3)(0.5) + (1.0)(0.5) = 0.65$$

Now solve for probability of Earth, given we have land:

$$\Pr(\text{Earth}|\text{land}) = \frac{\Pr(\text{land}|\text{Earth})\Pr(\text{Earth})}{\Pr(\text{land})} = \frac{(0.3)(0.5)}{0.65} \approx 0.23$$

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).

Enumerating all possible scenarios (b=black, w=white):

1. w up; w down (w/w)

2. w up; w down (w/w)
3. w up; b down (w/b)
4. b up; w down (w/b)
5. b up; b down (b/b)
6. b up; b down (b/b)

Observation eliminates 1-3, so 3-6 remain. 2/3 of those are b/b card, so this is our solution.

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.

Add to the prior cases:

7. b up; b down (new b/b)
8. b up; b down (new b/b)

Again we can eliminate 1-3 from observation. This leaves 5 cases (3-8). Of those 4 are b/b, so 4/5.

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.

Same cases as before, but now amend rates

up	down	card	rate
w	w	w/w	3
w	w	w/w	3
w	b	w/b	2
b	w	w/b	2
b	b	b/b	1
b	b	b/b	1

Now, we can cancel the first three cases, since we've pulled black. 2 b/b options at rate 1, 1 w/b option at rate 2. That means 2 positive chances out of 4 total rate chances, $2/4 = 0.5$.

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.

Looking at scenarios that match data:

1. b/b, w/b
2. b/b (flipped), w/b
3. b/b, w/w
4. b/b, w/w (flipped)

5. b/b (flipped), w/w
6. b/b (flipped), w/w (flipped)
7. b/w, w/w
8. b/w, w/w (flipped)

1-6 are desired, 7-8 are not; therefore 6/8 or 75%.

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?

```
# Givens
rate_a <- .5
rate_b <- .5
twin_rate_a <- .1
twin_rate_b <- .2
# Need to solve:
# P(twins) = P(twins|A)*P(A) + P(twins|B)P(B)
sum_probability_twins <- rate_a * twin_rate_a + rate_b * twin_rate_b # norm factor

pA_given_twins <- (twin_rate_a * rate_a) / sum_probability_twins
pB_given_twins <- (twin_rate_b * rate_b) / sum_probability_twins
p_twins <- twin_rate_a * pA_given_twins + twin_rate_b * pB_given_twins
p_twins
```

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

16.7% chance

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.

```
pA_given_twins
```

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

33% chance

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.

```
# Givens
updated_rate_a <- pA_given_twins
updated_rate_b <- pB_given_twins
single_rate_a <- 1-twin_rate_a
single_rate_b <- 1-twin_rate_b

# Repeat calculations for new single birth
# norm factor
```

```
sum_probability_single <- single_rate_a * twin_rate_a +
  single_rate_b * twin_rate_b
# Calculate probabilities
pA_given_single <- (single_rate_a * twin_rate_a) / sum_probability_single
pB_given_single <- (single_rate_b * twin_rate_b) / sum_probability_single
pA_given_single
```

```
## [1] 0.36
```

36% chance

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.

Starting with data-free solution:

```
# Givens
a_given_pos <- 0.8
b_given_pos <- 1 - a_given_pos
b_given_neg <- 0.65
b_given_pos <- 1 - b_given_neg

pA_given_test <- a_given_pos * rate_a /
                 (a_given_pos * rate_a + b_given_pos * rate_b)
pA_given_test
```

```
## [1] 0.6956522
```

Probability of A given test returns an "A" reading is 69.6%

Now adding the data:

```
# P(A| positive test, twins, single) =
# P(positive test|A) * P(twins|A) * P(single|A) * P(A) /
# P(postive test, twins, single)
numerator <- a_given_pos * twin_rate_a * single_rate_a * rate_a

# P(positive test, twins, single) =
# P(positive test|A) * P(twins|A) *  P(single|A) * P(A) +
# (positive test|B) * P(twins|B) *  P(single|B) * P(B)
denom <- numerator + b_given_pos * twin_rate_b * single_rate_b * rate_b

numerator/denom
```

```
## [1] 0.5625
```

56.25% chance

Chapter 3 problems

```
## Loading required package: rstan
```

```
## Loading required package: StanHeaders
```

```
## Loading required package: ggplot2
```

```
## rstan (Version 2.19.3, GitRev: 2e1f913d3ca3)
```

```
## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
```

```
## Loading required package: parallel
```

```
## rethinking (Version 2.10)
```

```
##
## Attaching package: 'rethinking'
```

```
## The following object is masked from 'package:stats':
##
##     rstudent
```

Given:

```
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)
trial_size <- 1e4 # Tyler added
samples <- sample( p_grid , prob=posterior , size=trial_size , replace=TRUE )
```

3E1.

How much posterior probability lies below p = 0.2?

```
sum(samples < 0.2) / trial_size
```

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

3E2.

How much posterior probability lies above p = 0.8?

```
sum(samples > 0.8) / trial_size
```

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

3E3.

How much posterior probability lies between p = 0.2 and p = 0.8?

```
sum(samples < 0.8 & samples > 0.2) / trial_size
```

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

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, 1-0.2)
```

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

3E6.

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

```
HPDI(samples,prob=.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?

```
PI(samples,prob=.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.

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

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, size=1e5, replace=TRUE)
HPDI(samples, prob=0.9)
```

```
##      |0.9      0.9|
## 0.3413413 0.7267267
```

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

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

```
## [1] 0.1473
```

3M4.

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

```
simulations <- 1e4
w_2 <- rbinom( simulations, size=9 , prob=samples )
sum(w==6)/simulations
```

```
## [1] 0.1108
```

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 )
prior_const <- 1
prior <- (p_grid >= 0.5) * prior_const
prob_data_new <- dbinom( 8 , size=15 , prob=p_grid )
posterior_new <- prob_data_new * prior
posterior_new <- posterior_new / sum(posterior)
samples_new <- sample(p_grid, prob=posterior_new, size=1e5, replace=TRUE)
```

Tackling all the old problems:

```
print("Problem 2")
```

```
## [1] "Problem 2"
```

```
HPDI(samples_new, prob=0.9)
```

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

```
print("Problem 3")
```

```
## [1] "Problem 3"
```

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

```
## [1] 0.1634
```

```
print("Problem 4")
```

```
## [1] "Problem 4"
```

```
w_2 <- rbinom( simulations, size=9 , prob=samples_new )
sum(w==6)/simulations
```

```
## [1] 0.0659
```

HPDI is far narrower. Likelihood of 8/15 is slightly increased, likelihood of 6/9 increases considerably - effectively we've removed the opportunity for fewer than 50% water cases to be considered, which will subsequently increase the likelihood of all >50% cases.

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?

```
interval_width <- 1
nSimulations <- 0
p <- 0.7
while (interval_width > 0.05)
{
  nSimulations <- nSimulations + 10
  p_grid <- seq( from=0 , to=1 , length.out=1000 )
  prob_p <- rep( 1 , 1000 )

  # Simulate data
  simulations <- nSimulations
  likelihood <- dbinom( round(simulations*p), size=simulations, prob=p_grid )
  posterior <- likelihood * prob_p
  posterior <- posterior / sum(posterior)
  #print(posterior)
  trial_size <- 1e4
  #print(trial_size)
  samples <- sample( p_grid , prob=posterior , size=trial_size , replace=TRUE )
  interval_width <- quantile(samples,0.995) - quantile(samples, 0.005)
}
nSimulations
```

```
## [1] 2200
```

About 2200 trials.

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?

```
all_births <- c(birth1,birth2)
p_grid <- seq( from=0 , to=1 , length.out=1000 )
prior <- rep( 1 , 1000 ) #Uniform
likelihood <- dbinom( sum(all_births) , size=length(all_births) , prob=p_grid )
posterior <- likelihood * prior
posterior <- posterior / sum(posterior)
p_grid[which.max(posterior)]
```

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

3H2.

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

```
trial_size <- 10000
samples <- sample( p_grid , prob=posterior , size=trial_size , replace=TRUE )
HPDI(samples, prob=.5)
```

```
##      |0.5      0.5|
## 0.5305305 0.5775776
```

```
HPDI(samples, prob=.89)
```

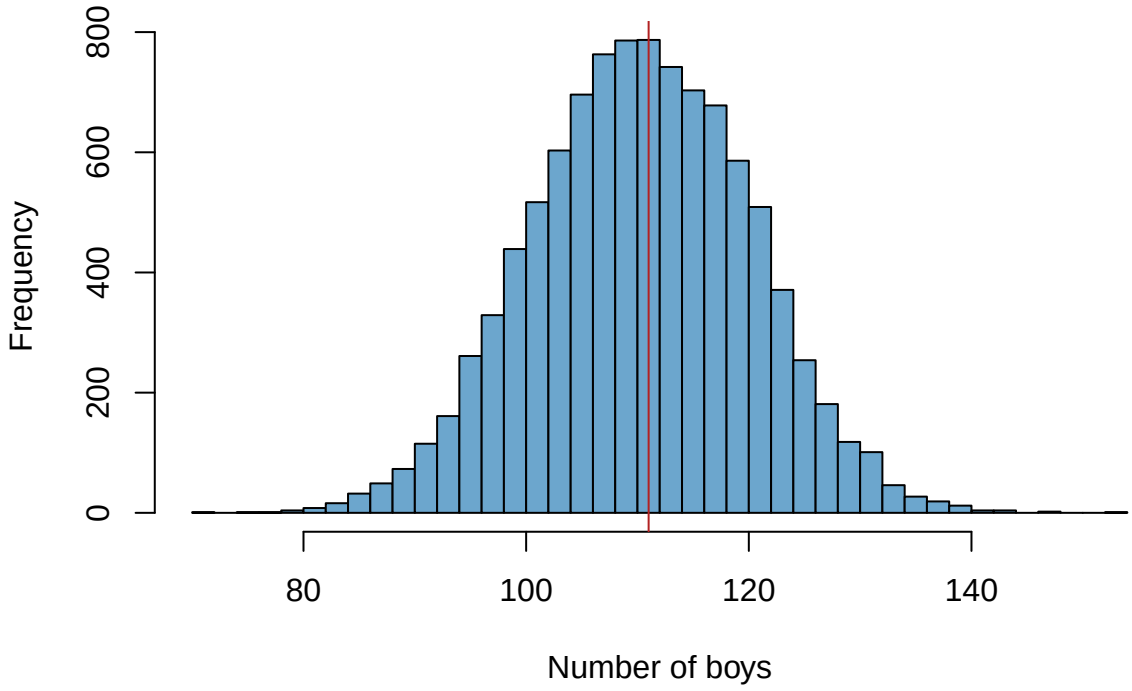
```
##     |0.89     0.89|
## 0.5005005 0.6116116
```

```
HPDI(samples, prob=.97)
```

```
##     |0.97     0.97|
## 0.4794795 0.6296296
```

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?

```
sim <- rbinom(10000, size=200, prob=samples)
hist(sim, c="skyblue3", breaks=50, xlab="Number of boys", main="")
abline(v=sum(all_births), col="firebrick")
```

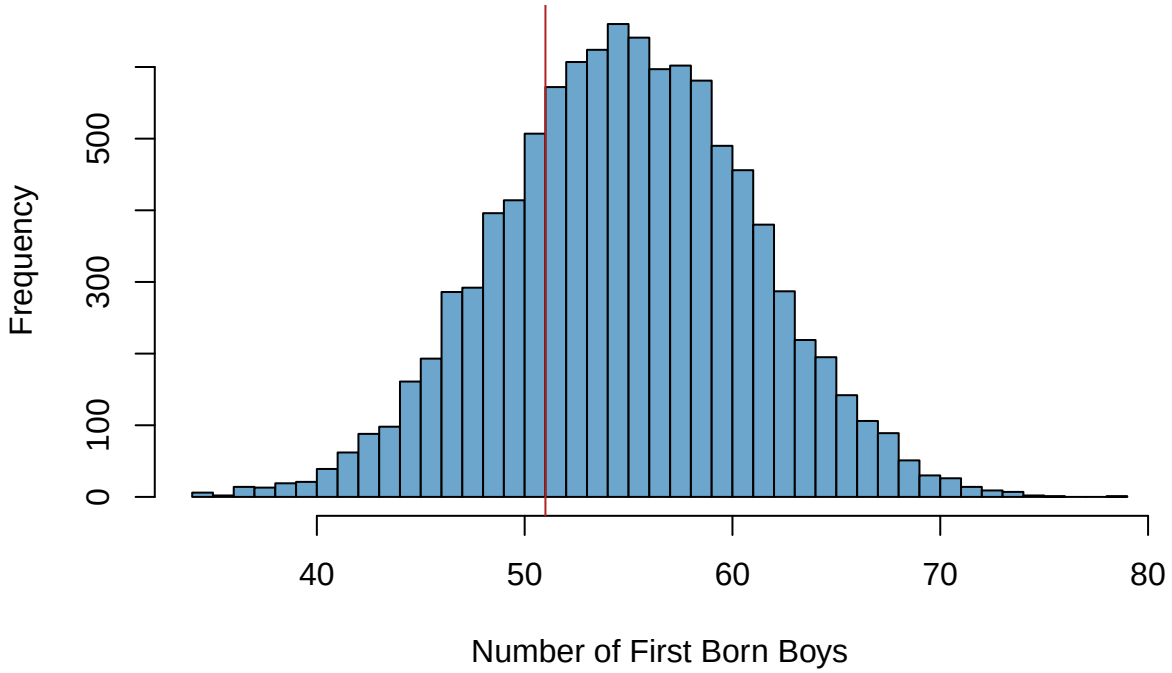


Fits right on mean, this seems like a likely outcome.

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?

```
sim <- rbinom(10000, size=100, prob=samples)
hist(sim, c="skyblue3", breaks=50, xlab="Number of First Born Boys", main="")
abline(v=sum(birth1), col="firebrick")
```



It's not on the maximum likelihood location, but it's still a reasonable value.

```
sprintf("Value: %i",sum(birth1))
```

```
## [1] "Value: 51"
```

```
PI(sim,prob=0.60)
```

```
## 20% 80%
##  50  61
```

The value is within the inner 60% of posterior density

3H5.

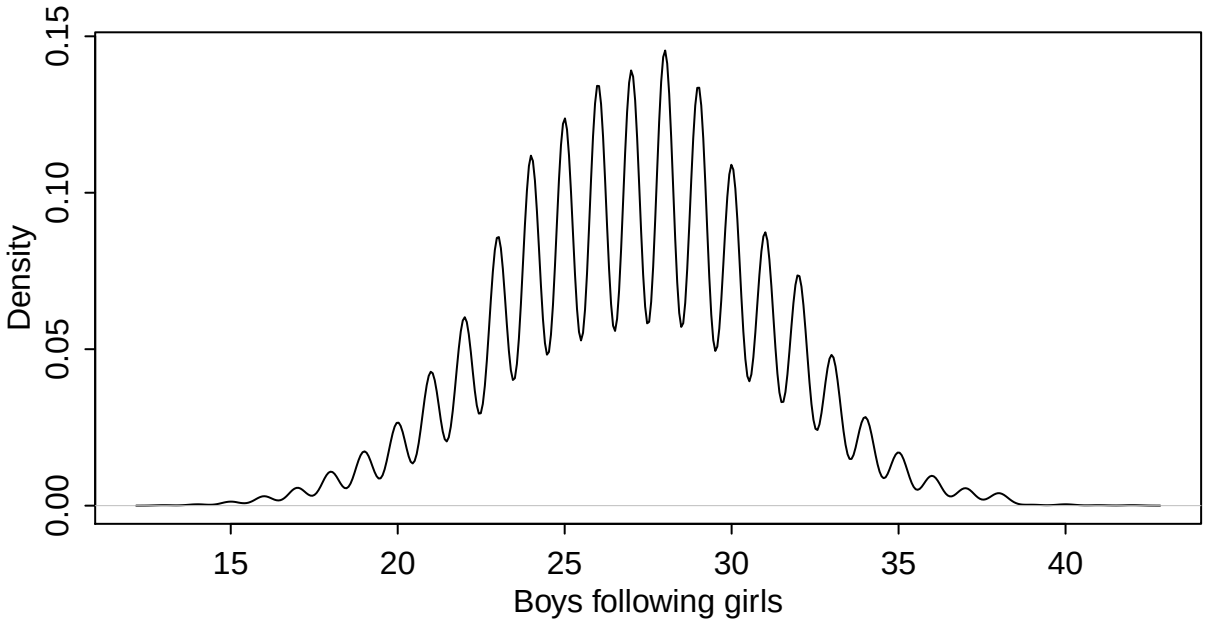
The model assumes that sex of first and second births are independent. To check this assumption, focus now on second births that followed female first borns. Compare 10,000 simulated counts of boys to only those second births that followed girls. To do this correctly, you need to count the number of 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?

```
boys_after_girls <- birth2[birth1==0 & birth2 ==1]
sum(boys_after_girls)
```

```
## [1] 39
```

39 cases of boys born after a girl

```
count_first_girls <- sum(birth1==0)
sim_girl <- rbinom(10000, size=count_first_girls, prob=samples)
dens(sim_girl, xlab="Boys following girls", main="")
```



This doesn't look like anything normal. The biggest thing is that binomial assumes that trials are independent, and it's very possible these are not.