Group Project: James Scruggs (lead), Kyle, Paul, & Brandon

504 Probability

Today we will examine probability.

Simulating a Fair Coin Flip

Imagine flipping a fair coin. There are only two outcomes -- heads or tails (discounting weird outcomes like the coin being lost or landing on its edge). Suppose we assign a "1" to the outcome "heads" and a "0" to the outcome "tails." Now we have a random number -- 0 or 1, with equal chances for each.

We'd like to simulate that on the computer.

#Simulate a Fair Coin Flip

theta<-0.5 #this is a fair coin

N=100

flips<-sample(c(0,1),size=100,replace=TRUE,prob=c(1-theta,theta))

flips

Do this twice.

1. We have simulated flipping a coin 100 times, twice. How many times would you expect to see heads (ones) in each of those sets of 100 flips?

50

2. You can see how many heads you got by looking at the number of 1's. (*sum(flips)*). Chances are that the number of ones you got the first time is different from the number of ones you got the second time, and that these numbers are different from your prediction in number 1. Why is that the case?

It's only a probability. It's not guaranteed to be 50% every time.

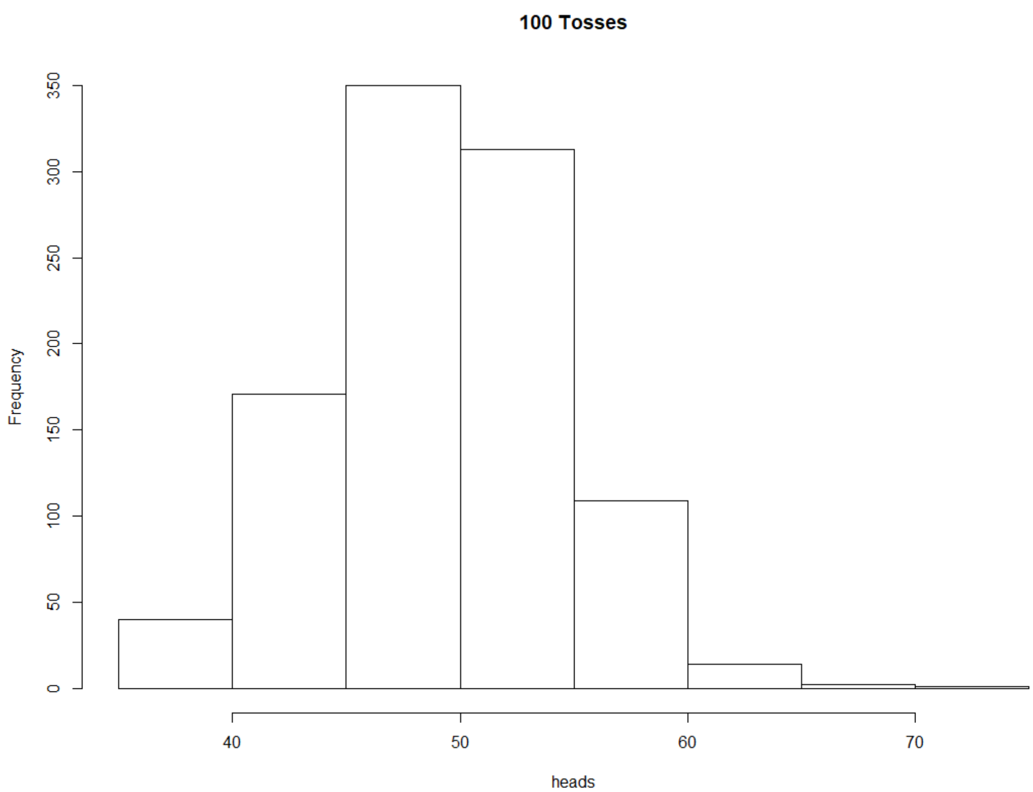
Simulating an Unfair Coin Flip

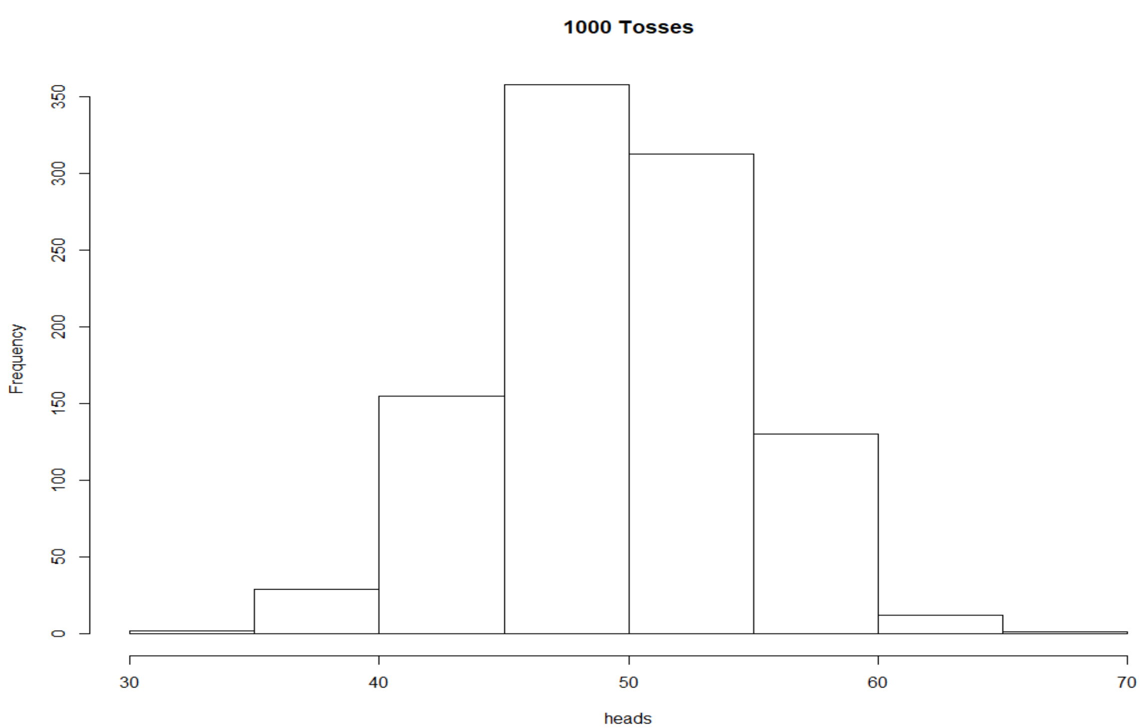
Suppose we want to simulate flipping a coin that is not fair. Suppose the coin is 90% likely to show heads, and only 10% likely to show tails. (*theta<-0.9*)

3. How many ones did you observe? Does this result agree with what your intuition would have predicted?

88 coins landed on heads. Yes, it is ALMOST 90%.

4. Do simulation for tossing coin for 100 and 1000 times. Count heads for each time with seed command =10 (seed =10). Plot histograms for both cases and explain the shape of histograms.





Simulating a Dice Roll

Suppose we want to simulate the rolling of a fair die.

#simulate the rolling of a fair die

sample.space=1:6

theta=1/6

n<-sample(sample.space, size = 100,replace = TRUE,prob = c(theta,theta,theta,theta,theta,theta))

5. Find the frequencies of each outcome and enter them below (*table(n)*). Are these frequencies all the same? Why or why not?

1: 15	4: 16
2: 15	5: 4
3: 28	6: 22

No, this is because the sample size is too small.

Simulating the Roll of Two Dice

Now we want to simulate rolling two fair dice. Then add the values together to get the total for both dice.

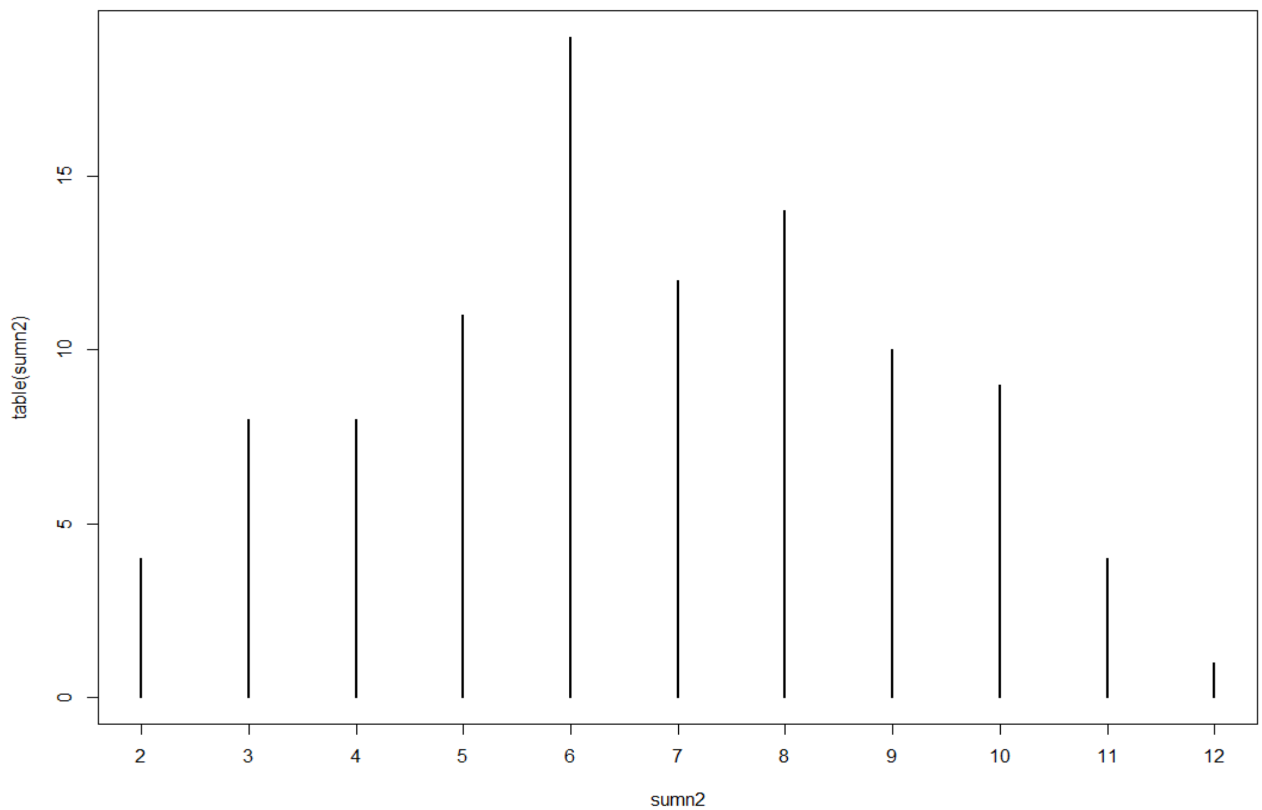
#simulate rolling two fair dice and add the values together to get the total for both dice

n2<-replicate(2,sample(sample.space,size = 100,replace = TRUE,prob = c(theta,theta,theta,theta,theta,theta))) #replicate does the sampling twice

n2 #Note that n2 is a matrix with 100 rows and 2 columns

sumn2<-apply(n2,1,sum) #apply does the same command on each element of a matrix. 1 is for rows, and 2 is for columns. This command has a 1, so it is summing each row.

6. Find the frequencies of the outcomes (*table(sumn2) or plot(table(sumn2))*). Enter below which outcome is most frequent (insert the plot). Does this match your expectation? What do you think would happen if we simulated 100,000 rolls instead of just 100? After you make a prediction, modify the above code to simulate 100,000 rolls and see if your prediction holds.



The most frequent is 6. I believe that if we increase the sample size to a 100,000 we will see a more uniform distribution. Our prediction does hold not hold.

Using Simulation to Examine the Birthday Problem

A famous problem in probability is the Birthday Problem: There are 25 people in a room. What is the probability that none of them share a birthday? We will simulate this situation.

#Using Simulation to Examine the Birthday Problem
birthday<-sample(1:365,size = 25,replace = TRUE, prob = rep(1/365,times=365))
table(birthday)

7. Did your data show any repeated birthdays?

Yes, just one. Day 319.

Repeat the assignment of birthdays.

8. Did you get the same values as the first time? Why not? Did your data show any repeated birthdays this time?

Nearly all are different because there 25 people out of 365 days is a small sample size (6.8%). No repeated birthdays this time.

Repeat the assignment of birthdays 20 times.

birthday20<-replicate(20,sample(1:365,size = 25,replace = TRUE,prob = rep(1/365,times=365)))
apply(birthday20,2, table)

To count the number of times there was a repeat, you can scan through the list above, or you can use the following:

temp <- apply(birthday20,2,duplicated)
#duplicated tests a vector for doubled elements. Now temp is a matrix of trues and falses

apply(temp,2,sum)
#A false is recorded as a zero, and a true as a one. Adding them up gives the number of duplicates.
#Any non-zero number is an assignment of birthdays with a duplicate.

sum(apply(temp,2,sum)==0)
#This adds the number of times there was no repeat, then subtract from 20 to get the number of repeats.

9. How many out of all 20 assignments had repeated birthdays ()?

7

The theoretical probability for how likely it is to have at least two people in a crowd of 25 share a birthday is 0.592. That is, it is expected to happen about 59% of the time.

10. Find the percent of time you saw repeated birthdays. (This is an experimental or empirical probability.) How does this compare to the theoretical probability? If we did 20,000 assignments instead of 20, how do you think that would affect the experimental probability? Change the above code to do 20,000 assignments.

I think empirical probability will be less than theoretical probability. I think if we did 20,000 assignments instead of 20, we would see a closer value to 59% of duplicates. After repeating my code with 20,000 assignments, I got 8741 duplicate birthdays which is higher than the previous experiment.

11. The code I gave you in the previous couple of problems didn't use loops. Write out how you could do the same thing using loops. (Pseudocode is fine. Or you can look up the syntax for a loop in R.)

```
for(person in 1:20)

{

    birthday<-sample(1:365,size = 25,replace = T, prob = rep(1/365,times=365))

    temp <- apply(birthday,2,duplicated) # count number of repeats.

    apply(temp,2,sum)

    temp2[person] = sum(apply(temp,2,sum)==0) }
```