Brandon's Lab Code

September 30, 2023

This assignment will be reviewed by peers based upon a given rubric. Make sure to keep your answers clear and concise while demonstrating an understanding of the material. Be sure to give all requested information in markdown cells. It is recommended to utilize Latex.

0.0.1 Problem 1

The Birthday Problem: This is a classic problem that has a nonintuitive answer. Suppose there are N students in a room.

Part a) What is the probability that at least two of them have the same birthday (month and day)? (Assume that each day is equally likely to be a student's birthday, that there are no sets of twins, and that there are 365 days in the year. Do not include leap years).

Note: Jupyter has two types of cells: Programming and Markdown. Programming is where you will create and run R code. The Markdown cells are where you will type out expalantions and mathematical expressions. Here is a document on Markdown some basic markdwon syntax. Also feel free to look at the underlying markdown of any of the provided cells to see how we use markdown.

$$\begin{aligned} P(\text{At least two have same birthday}) &= ? \\ &= \text{YOUR ANSWER HERE} \end{aligned}$$

[ ]:
```
#Answer in detial on word doc PDF
```

Part b) How large must N be so that the probability that at least two of them have the same birthday is at least 1/2?

[102]:
```
#Answer in detail on word doc PDF
```

23

Part c) Plot the number of students on the x-axis versus the probability that at least two of them have the same birthday on the y-axis.

[31]:

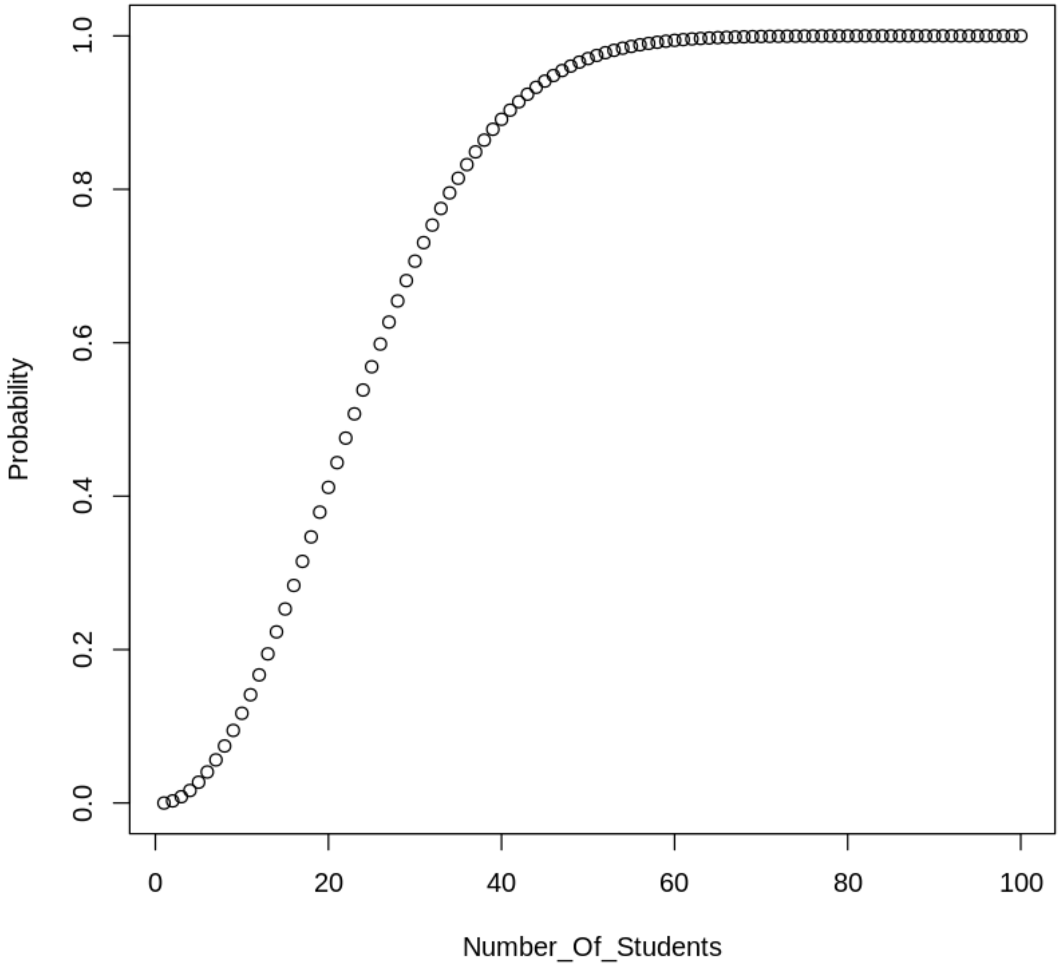
```
Number_Of_Students = 1:100
#
Probability = rep(0,100)

#Itirating through each Student 1-100
for (N in Number_Of_Students)
{
    factor = 1 #increment by 1

        for (x in seq(365,365-N+1))
                {
                    factor = factor*x #mulitplies each individual probability in demonomiator - binomial distribution
                }

    Probability[N] = 1 - factor/(365^N) #finds final probability by subtracting result from 1 or 100%
}

#plot graph
plot(Number_Of_Students, Probability)
```



Thought Question (Ungraded) Thought question (Ungraded): Would you be surprised if there were 100 students in the room and no two of them had the same birthday? What would that tell you about that set of students?

[ ]:
```
##That would be very suprising since it would only take about 23 of them to␣
 ↪reach a .50 probability of 2 birthdays matching.
```

1 Problem 2

One of the most beneficial aspects of R, when it comes to probability, is that it allows us to simulate data and random events. In the following problem, you are going to become familiar with these simulation functions and techniques.

Part a)

Let X be a random variable for the number rolled on a fair, six-sided die. How would we go about simulating X?

Start by creating a list of numbers [1, 6]. Then use the `sample()` function with our list of numbers to simulate **a single** roll of the die, as in simulate X. We would recommend looking at the documentation for `sample()`, found here, or by executing `?sample` in a Jupyter cell.

[3]:

```
x = 1:6
n = 1
sample(x, n)
```

6

Part b)

In our initial problem, we said that X comes from a fair die, meaning each value is equally likely to be rolled. Because our die has 6 sides, each side should appear about $1/6^{th}$ of the time. How would we confirm that our simulation is fair?

What if we generate multiple instances of X? That way, we could compare if the simulated probabilities match the theoretical probabilities (i.e. are all 1/6).

Generate 12 instances of X and calculate the proportion of occurances for each face. Do your simulated results appear to come from a fair die? Now generate 120 instances of X and look at the proportion of each face. What do you notice?

Note: Each time you run your simulations, you will get different values. If you want to guarantee that your simulation will result in the same values each time, use the `set.seed()` function. This function will allow your simulations to be reproducable.

[84]:

```
x = 1:6
#Take sum of occurances for given value rolled 1 through 6, and find frequency/
 ↪proportion
#load as a table
"Frequency of 12 Rolls"
set.seed(112358)
table(sample(x, 12, replace = TRUE))/12

#Repeat for 120 rolls
"Frequency of 120 Rolls"
set.seed(112358)
table(sample(x, 120, replace = TRUE))/120

#Repeat for 10000 rolls for fun
"Frequency of 10000 Rolls"
set.seed(112358)
table(sample(x, 10000, replace = TRUE))/10000
```

```
#Visuals for fun
hist(a,
     main = "Proportion of Occurances for 12 Rolls",
     col = "aquamarine1",
     labels = TRUE,
     xlab = "Face of Die Roll"
    )
hist(b,
     main = "Proportion of Occurances for 120 Rolls",
     col = "firebrick2",
     labels = TRUE,
     xlab = "Face of Die Roll"
    )

hist(c,
     main = "Proportion of Occurances for 10000 Rolls",
     col = "coral",
     labels = TRUE,
     xlab = "Face of Die Roll"
    )
```

’Frequency of 12 Rolls’

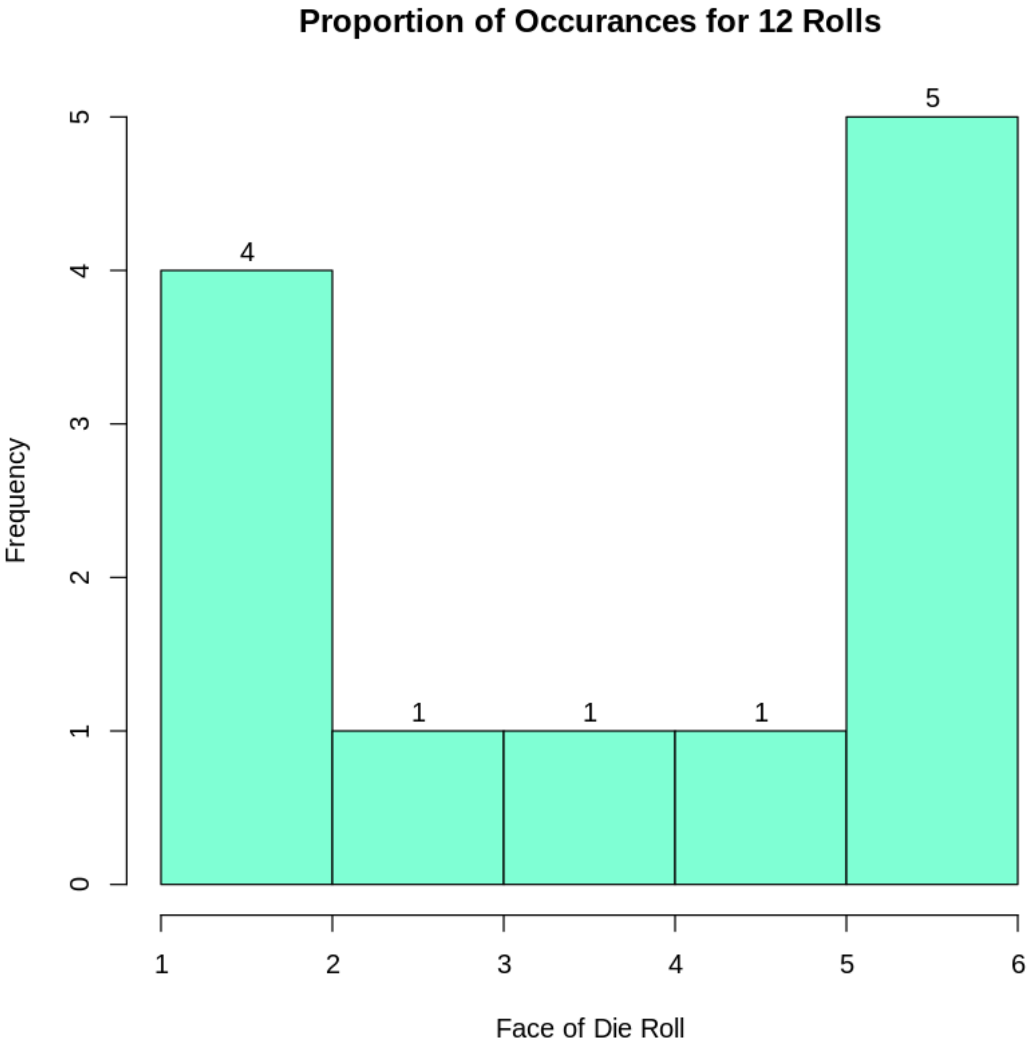
```
         1          2          3          4          5          6
0.08333333 0.25000000 0.25000000 0.16666667 0.08333333 0.16666667
```

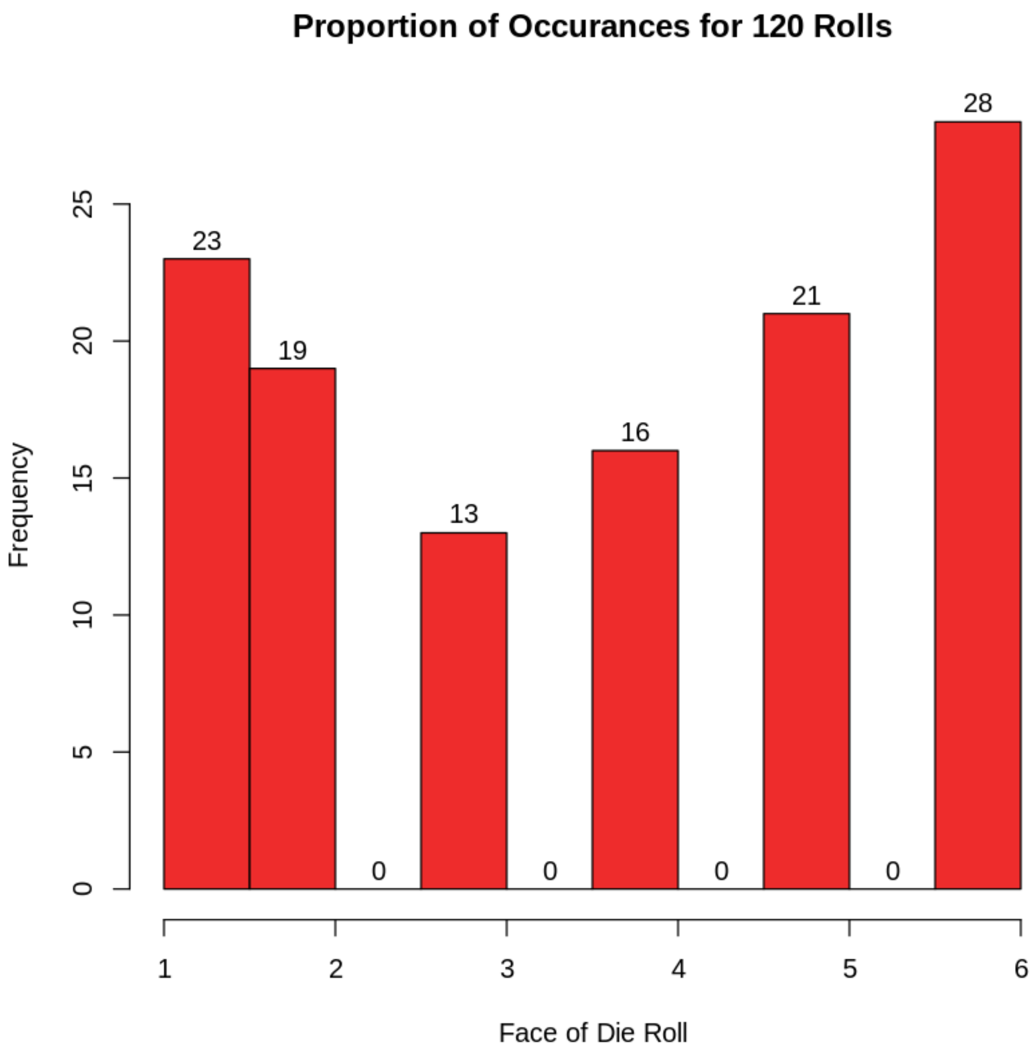
’Frequency of 120 Rolls’

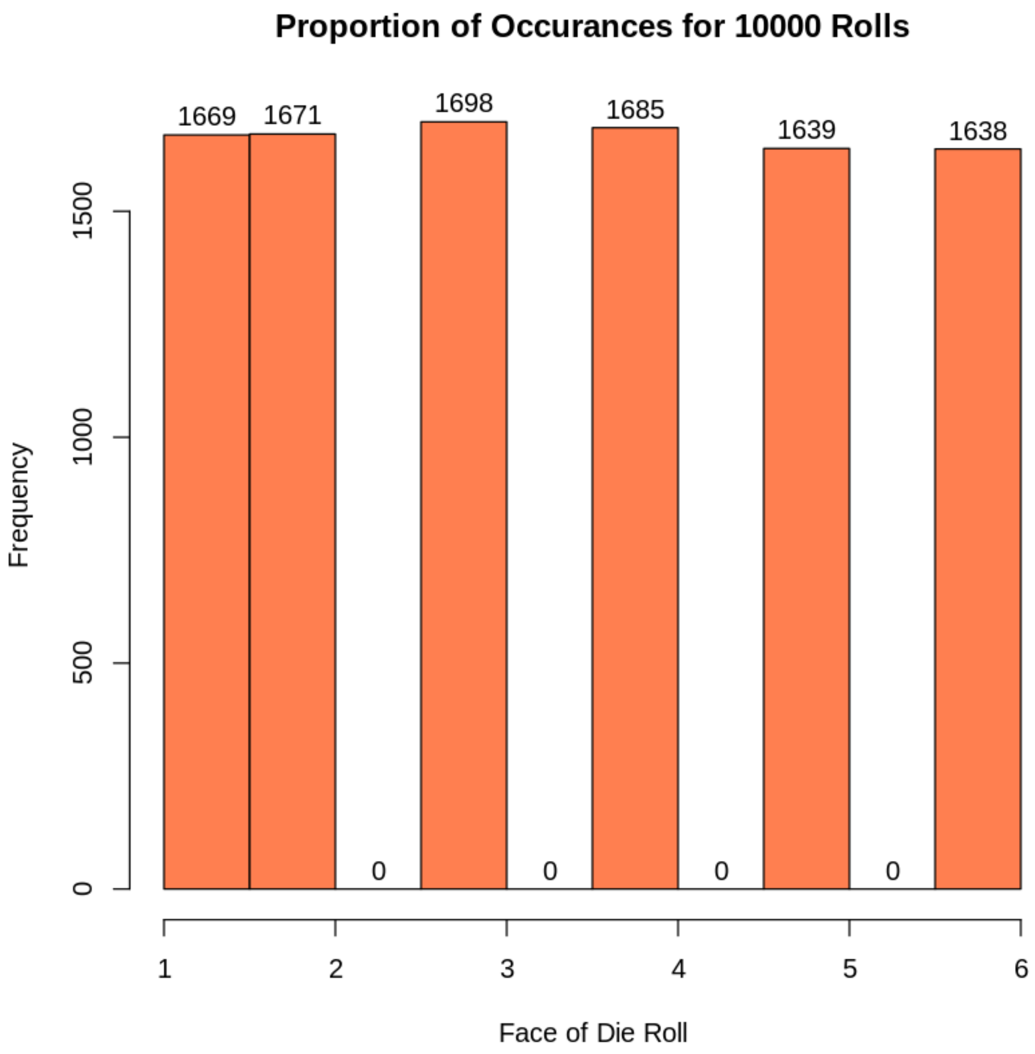
```
        1         2         3         4         5         6
0.1583333 0.1666667 0.2333333 0.1583333 0.1333333 0.1500000
```

’Frequency of 10000 Rolls’

```
     1      2      3      4      5      6
0.1648 0.1704 0.1699 0.1627 0.1594 0.1728
```


Proportion of Occurances for 12 Rolls
Frequency
0
1
2
3
4
5
4
1
1
1
5
1
2
3
4
5
6
Face of Die Roll


Proportion of Occurances for 120 Rolls
Frequency
0
5
10
15
20
25
23
19
0
13
0
16
0
21
0
28
1
2
3
4
5
6
Face of Die Roll



Part c)

What if our die is not fair? How would we simulate that?

Let's assume that Y comes from an unfair six-sided die, where $P(Y = 3) = 1/2$ and all other face values have an equal probability of occuring. Use the `sample()` function to simulate this situation. Then display the proportion of each face value, to confirm that the faces occur with the desired probabilities. Make sure that n is large enough to be confident in your answer.

[1]:

```
round(table(sample(1:6, 10e7, replace = TRUE, prob = c(1/6, 1/6, 1/2, 1/6, 1/6, 1/6)))/10e7 , 2)
```

```
   1    2    3    4    5    6 
0.13 0.13 0.37 0.13 0.12 0.12 
```

[ ]:

```
```