Internship in Project 4:
Identifying the causal loci in a perturbed genome

Kalki Bhavsar
AU1841029
BTech ICT 2018-2022

Question 1) Take a pair of fair dice and roll them say 100000 times.
(a) Find the most frequent combination of events.
(b) Find the most frequent sum of the outcomes from the two dice.
(c) plot the outcomes of (a) and (b) from 100000 throws
and indicate the mean and mode of the two distributions on the plot.

Question 2) Take a pair of dice, one of them is loaded and the other fair and roll them say 100000 times. The loaded dice gives a particular outcome more frequently, i.e., say the number 3 occurs with a probability of 1/4 rather than 1/6. The fair dice, on the other hand, outputs the number 4 with the probability of 1/6.

Again here,
(a) Find the most frequent combination of events.
(b) Find the most frequent sum of the outcomes from the two dice.
(c) plot the outcomes of (a) and (b) from 100000 throws and
indicate the mean and mode of the two distributions on the plot.

Installations required:
>> python

>> pip
>> Any python editor (I used PyCharm and made project in virtual environment)
https://www.jetbrains.com/pycharm/download/
(download the community version)

Imports:
~matplotlib:
 pip install matplotlib   (code on how to install using pycharm terminal)

~ random: inbuilt in pycharm
~ math: inbuilt in pycharm

Check versions for all dependencies:
> pip --version (in pycharm terminal)
> python --version (in pycharm terminal)
>>> import matplotlib
... print('matplotlib: {}'.format(matplotlib.__version__))   (in pycharm console)

Versions used:
pip 19.0.3
Python 3.7.3
matplotlib: 3.2.1

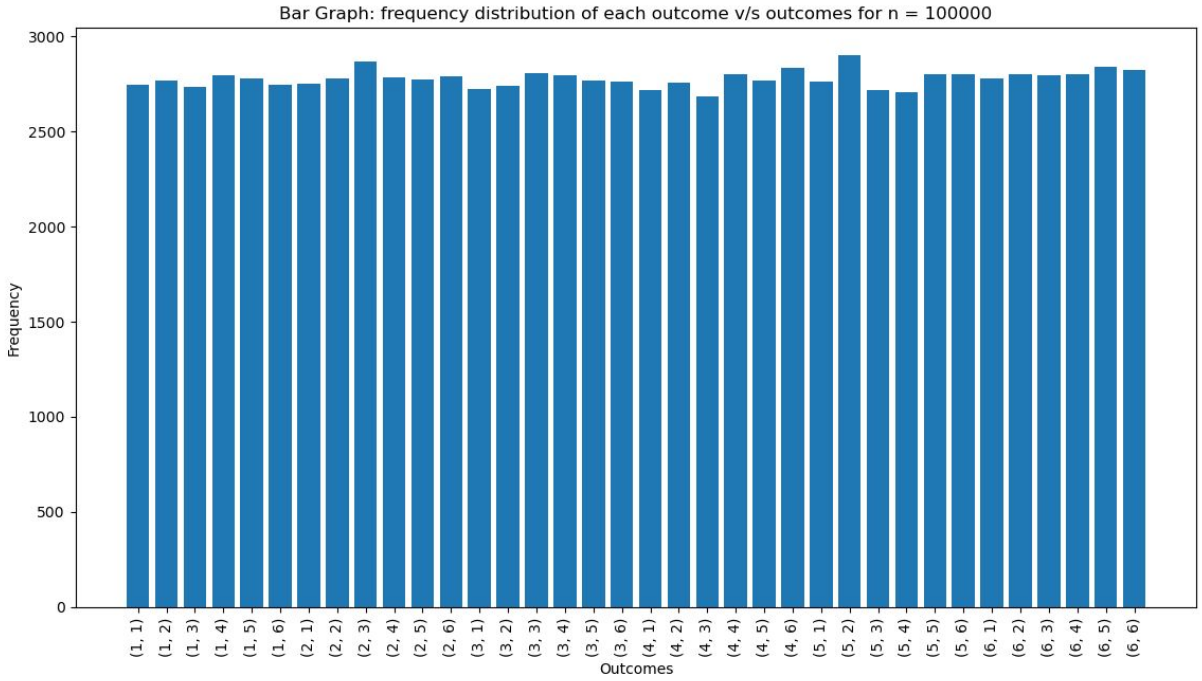
Solution 1)

```
import matplotlib.pyplot as plt
import random
import math
"""
sample_space ={(1,1), (1,2), (1,3), (1,4), (1,5), (1,6),
                (2,1), (2,2), (2,3), (2,4), (2,5), (2,6),
                (3,1), (3,2), (3,3), (3,4), (3,5), (3,6),
                (4,1), (4,2), (4,3), (4,4), (4,5), (4,6),
                (5,1), (5,2), (5,3), (5,4), (5,5), (5,6),
                (6,1), (6,2), (6,3), (6,4), (6,5), (6,6)
        }
print(sample_space)
"""
sample_space = {}
outcomes_count = [[0,0,0,0,0,0], [0,0,0,0,0,0], [0,0,0,0,0,0], [0,0,0,0,0,0], [0,0,0,0,0,0], [0,0,0,0,0,0]]
outcomes_sum_count = [0,0,0,0,0,0,0,0,0,0,0]
outcomes_probability = [[0,0,0,0,0,0], [0,0,0,0,0,0], [0,0,0,0,0,0], [0,0,0,0,0,0], [0,0,0,0,0,0], [0,0,0,0,0,0]]
outcomes_sum_probability = [0,0,0,0,0,0,0,0,0,0,0]

n = 100000
for i in range(1, n+1):
    die_1 = random.randint(1, 6)
    die_2 = random.randint(1, 6)
    outcome = (die_1, die_2)
    outcomes_count[die_1-1][die_2-1] += 1
    if (outcome in sample_space):
        sample_space[outcome] += 1
    else:
        sample_space[outcome] = 1
print(outcomes_count)
X = []
Y = []

"""Since sum of all outcomes = n, the mean = n/36
Since this is a completely random event, we can not exactly say that whenever,
we do this experiment, we will get this particular value the mode.
However the mode is nearly equal to mean since all the events are equally likely
as seen the bar graph
"""
mean = math.ceil(n/36)
mode = [0, 0]
mode_freq = 0
for i in range(6):
    for j in range(6):
        outcome = outcomes_count[i][j]
        Y += [outcome]
        x_label = "("+str(i+1)+", "+str(j+1)+")"
        X += [x_label]
        outcomes_sum_count[i+j] += outcomes_count[i][j]
        outcomes_probability[i][j] = outcomes_count[i][j] / n
        if mode_freq < outcomes_count[i][j]:
            mode = [i+1, j+1]
            mode_freq = outcomes_count[i][j]
```

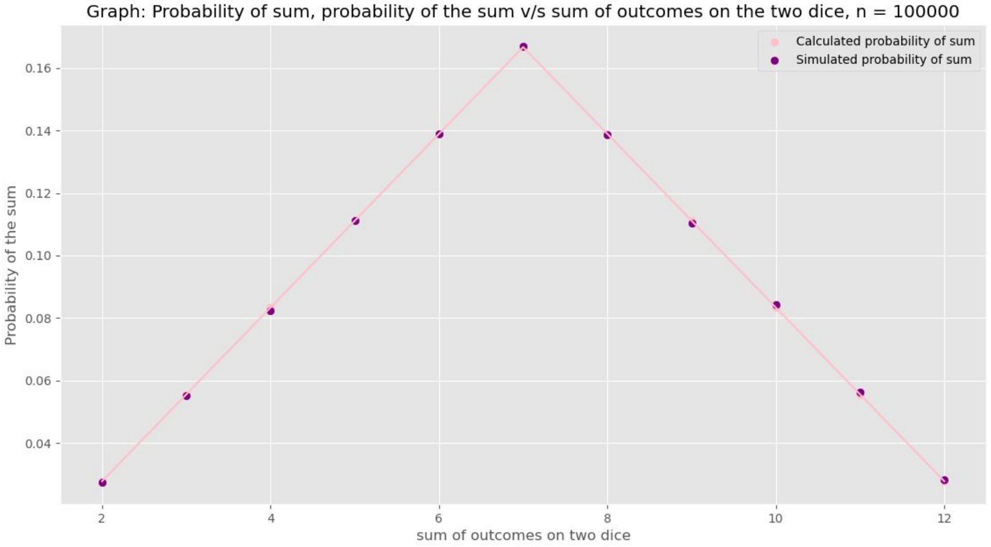
```
for i in range(len(outcomes_sum_count)):
    outcomes_sum_probability[i] = outcomes_sum_count[i]/n
print('\nSample space', sample_space)
print('\nOutcomes count: ',outcomes_count)
print('\nOutcomes_sum_count: ',outcomes_sum_count)
print('\nOutcomes probability: ',outcomes_probability)
print('\nOutcomes_sum_count_probability: ',outcomes_sum_probability)

print('\n\nY: ',Y)
print('\nX: ',X)

print('\n\n\t\tMean: ',mean)
print('\n\t\tMode: ',mode,' with frequency = ',mode_freq)
#print('length of X: ',len(X),' and length of Y: ',len(Y))
s = "Mean:"+str(mean)+" and Mode: "+str(mode)+" with frequency = " +str(mode_freq)
X2 = [2,3,4,5,6,7,8,9,10,11,12]

fig, ax = plt.subplots()
ax.set_xticklabels(X, rotation=90)
fig.suptitle(s, fontsize=14, fontweight='bold')
plt.title('Bar Graph: frequency distribution of each outcome v/s outcomes for n = 100000')
plt.xlabel('Outcomes')
plt.ylabel('Frequency')
plt.bar(X,Y)
plt.style.use('ggplot')
plt.show()


mean = 0
for i in range(len(outcomes_sum_probability)):
    mean += (i+2)*outcomes_sum_probability[i]

mode = 0
mode_freq = max(outcomes_sum_count)
for i in range(11):
    if mode_freq == outcomes_sum_count[i]:
        mode = i+2
s = "Mean:"+str(mean)+" and Mode: "+str(mode)+" with frequency = " +str(mode_freq)

fig = plt.figure()
calculated_probability_of_sum = [1/36, 2/36, 3/36, 4/36, 5/36, 6/36, 5/36, 4/36, 3/36, 2/36, 1/36]
plt.scatter(X2, calculated_probability_of_sum, label="Calculated probability of sum", color = 'pink')
plt.plot(X2, calculated_probability_of_sum, color = 'pink')
plt.scatter(X2, outcomes_sum_probability,  label="Simulated probability of sum",color = 'purple')
plt.xlabel('sum of outcomes on two dice')
plt.ylabel('Probability of the sum')

fig.suptitle(s, fontsize=14, fontweight='bold')
plt.title('Graph: Probability of sum, probability of the sum v/s sum of outcomes on the two dice, n = 100000')
plt.grid(True)
plt.legend()
plt.show()
```

Simulation 1

Mean:2778 and Mode: [5, 2] with frequency = 2899

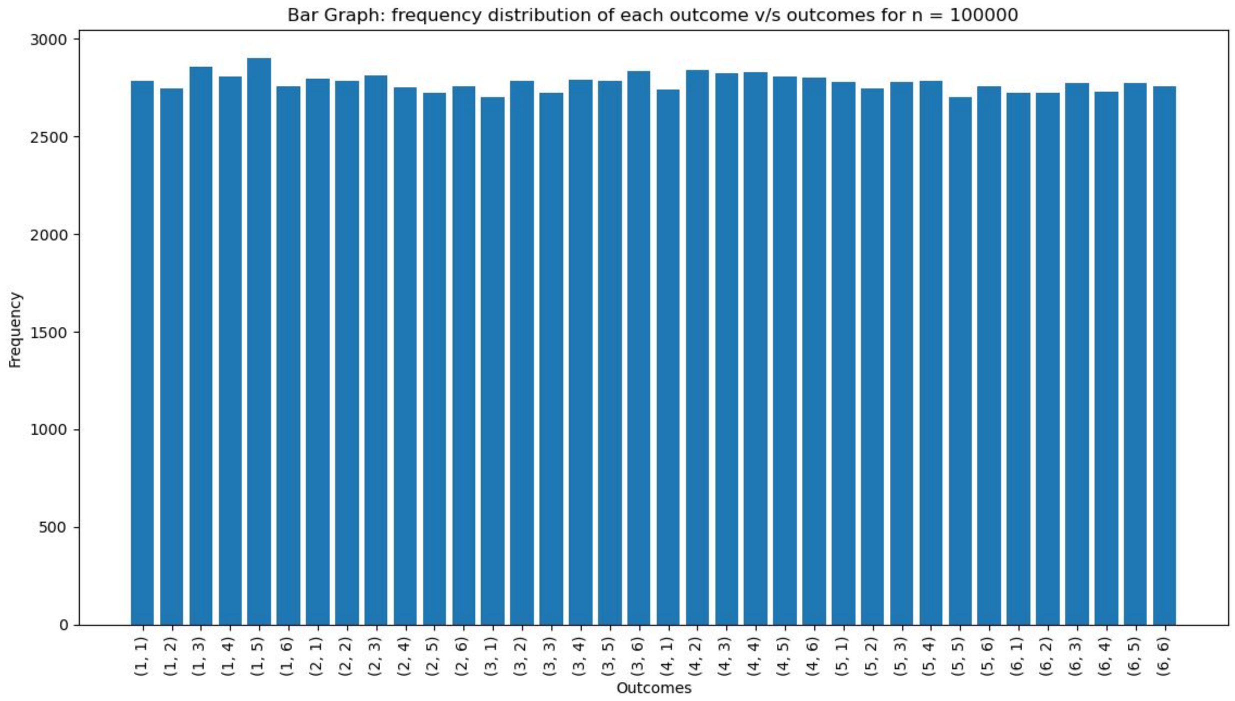

Bar Graph: frequency distribution of each outcome v/s outcomes for n = 100000

Mean:7.01281 and Mode: 7 with frequency = 16686

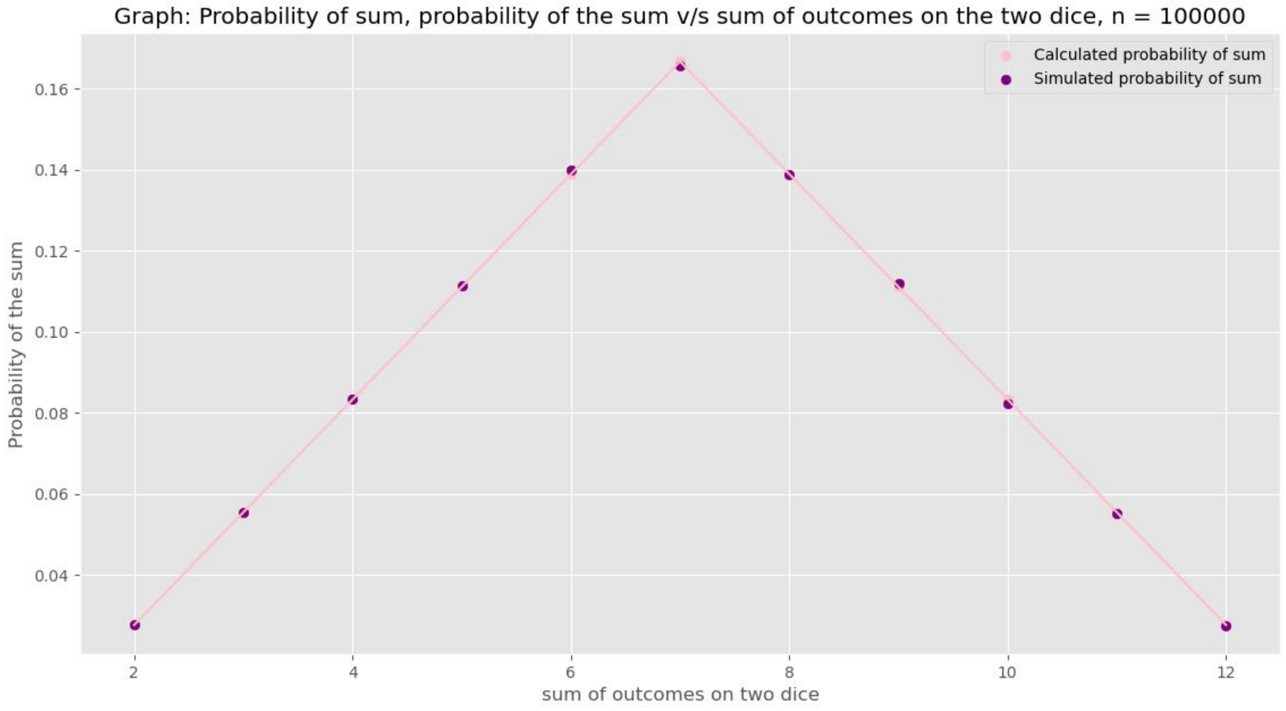

Graph: Probability of sum, probability of the sum v/s sum of outcomes on the two dice, n = 100000

Simulation 2

Mean:2778 and Mode: [1, 5] with frequency = 2900

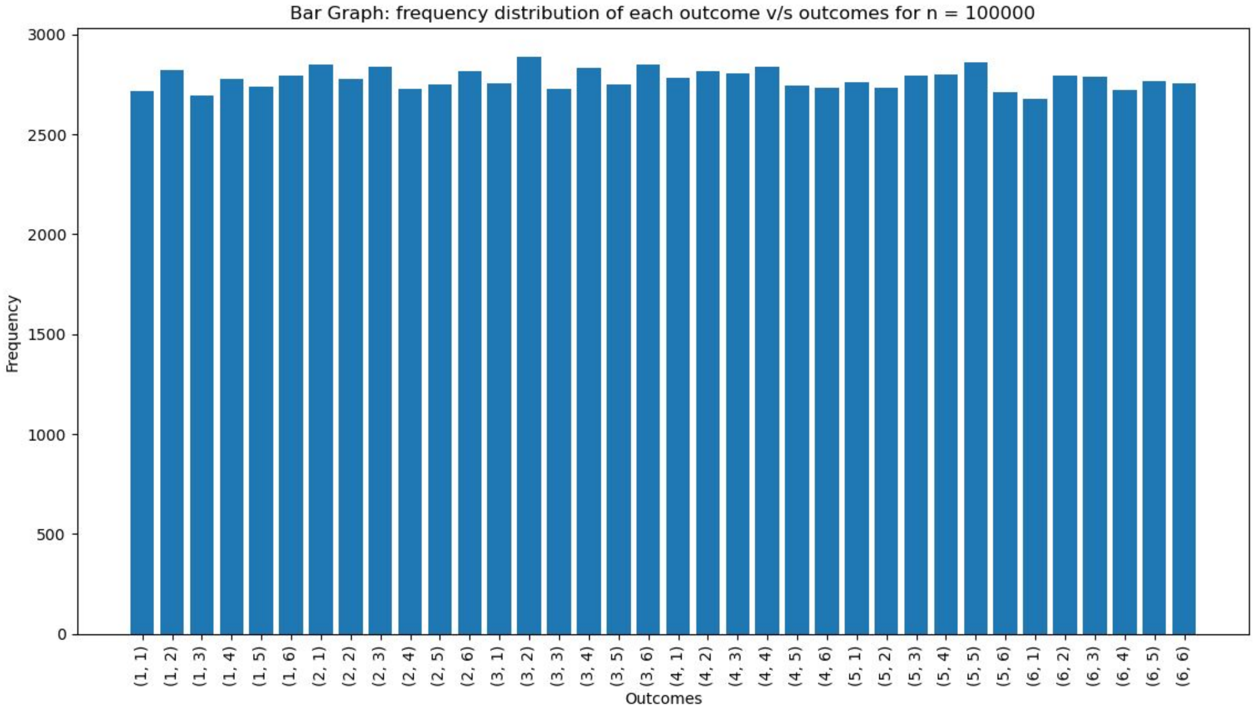


Mean:6.99489 and Mode: 7 with frequency = 16566



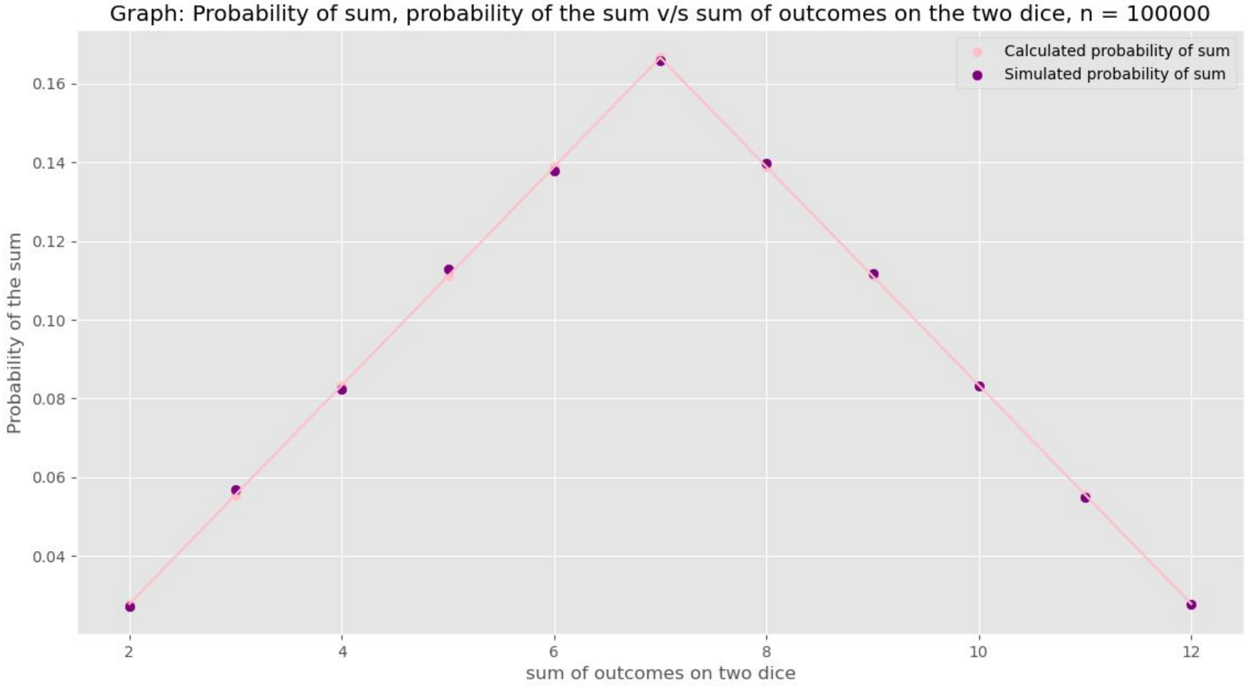
Simulation 3

Mean:2778 and Mode: [3, 2] with frequency = 2886



Bar Graph: frequency distribution of each outcome v/s outcomes for n = 100000

Mean:6.996910000000001 and Mode: 7 with frequency = 16594



Graph: Probability of sum, probability of the sum v/s sum of outcomes on the two dice, n = 100000

Solution 2)

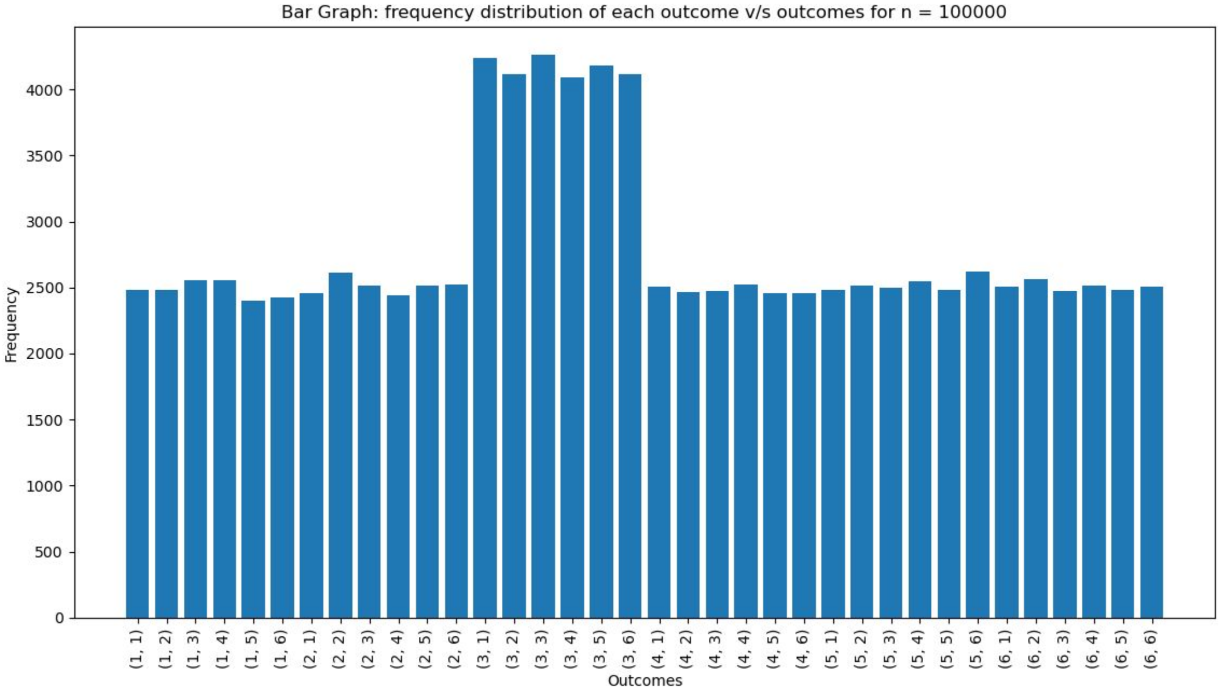
```
import matplotlib.pyplot as plt
import random
import math
"""
sample_space ={(1,1), (1,2), (1,3), (1,4), (1,5), (1,6),
            (2,1), (2,2), (2,3), (2,4), (2,5), (2,6),
            (3,1), (3,2), (3,3), (3,4), (3,5), (3,6),
            (4,1), (4,2), (4,3), (4,4), (4,5), (4,6),
            (5,1), (5,2), (5,3), (5,4), (5,5), (5,6),
            (6,1), (6,2), (6,3), (6,4), (6,5), (6,6)
      }
print(sample_space)
"""
sample_space = {}
outcomes_count = [[0,0,0,0,0,0], [0,0,0,0,0,0], [0,0,0,0,0,0], [0,0,0,0,0,0], [0,0,0,0,0,0], [0,0,0,0,0,0]]
outcomes_sum_count = [0,0,0,0,0,0,0,0,0,0,0]
outcomes_probability = [[0,0,0,0,0,0], [0,0,0,0,0,0], [0,0,0,0,0,0], [0,0,0,0,0,0], [0,0,0,0,0,0], [0,0,0,0,0,0]]
outcomes_sum_probability = [0,0,0,0,0,0,0,0,0,0,0]

n = 100000
for i in range(1, (n//4)+1):
    for j in range(3):
        die_1 = random.randint(1, 6)
        while(die_1==3):
            die_1 = random.randint(1, 6)

        die_2 = random.randint(1, 6)
        outcome = (die_1, die_2)
        outcomes_count[die_1-1][die_2-1] += 1

        if (outcome in sample_space):
            sample_space[outcome] += 1
        else:
            sample_space[outcome] = 1

    die_1 = 3
    die_2 = random.randint(1, 6)
    outcome = (die_1, die_2)
    outcomes_count[die_1 - 1][die_2 - 1] += 1

    if (outcome in sample_space):
        sample_space[outcome] += 1
    else:
        sample_space[outcome] = 1
print(outcomes_count)
X = []
Y = []
```

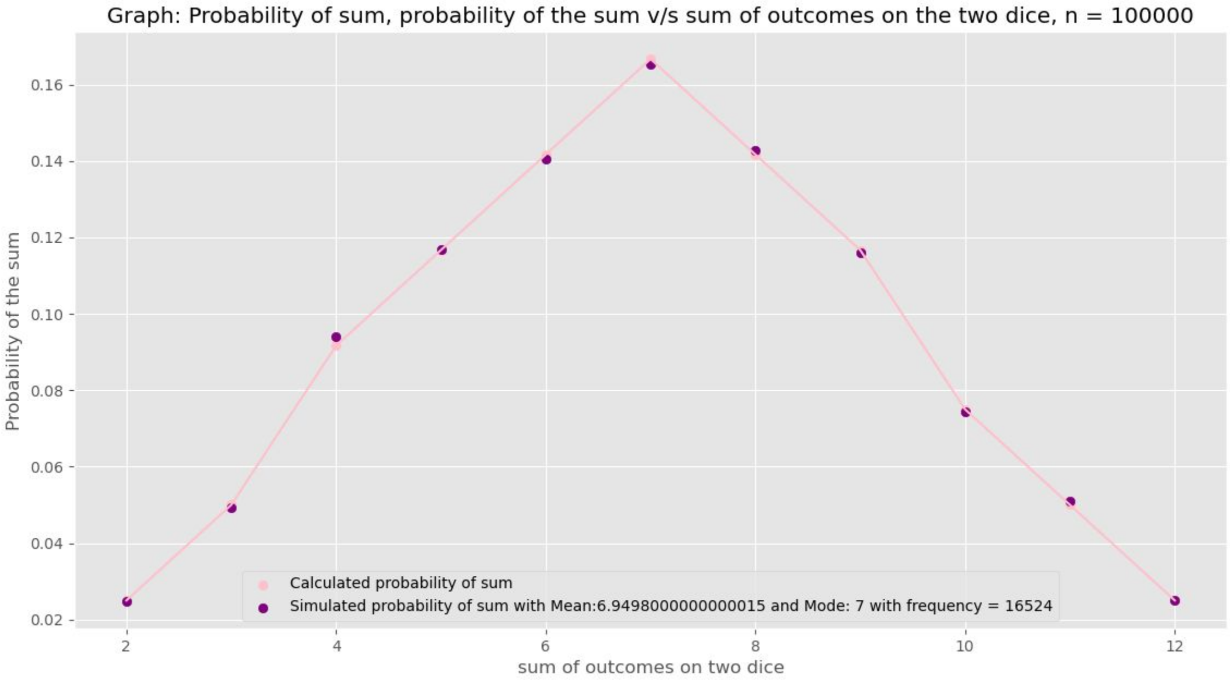
```
"""Since sum of all outcomes = n, the mean = n/36
Since this is a completely random event, we can not exactly say that whenever,
we do this experiment, we will get this particular value the mode.
However the mode is nearly equal to mean since all the events are equally likely
as seen the bar graph
"""
mean = math.ceil(n/36)

mode = [0, 0]
mode_freq = 0
for i in range(6):
   for j in range(6):
      outcome = outcomes_count[i][j]
      Y += [outcome]
      x_label = "("+str(i+1)+", "+str(j+1)+")"
      X += [x_label]
      outcomes_sum_count[i+j] += outcomes_count[i][j]
      outcomes_probability[i][j] = outcomes_count[i][j] / n

      if mode_freq < outcomes_count[i][j]:
         mode = [i+1, j+1]
         mode_freq = outcomes_count[i][j]

for i in range(len(outcomes_sum_count)):
   outcomes_sum_probability[i] = outcomes_sum_count[i]/n
print('\nSample space', sample_space)
print('\nOutcomes count: ',outcomes_count)
print('\nOutcomes_sum_count: ',outcomes_sum_count)
print('\nOutcomes probability: ',outcomes_probability)
print('\nOutcomes_sum_count_probability: ',outcomes_sum_probability)

print('\n\nY: ',Y)
print('\nX: ',X)

print('\n\n\t\tMean: ',mean)
print('\n\t\tMode: ',mode,' with frequency = ',mode_freq)
#print('length of X: ',len(X),' and length of Y: ',len(Y))
s = "Mean:"+str(mean)+" and Mode: "+str(mode)+" with frequency = " +str(mode_freq)
X2 = [2,3,4,5,6,7,8,9,10,11,12]

fig, ax = plt.subplots()
ax.set_xticklabels(X, rotation=90)
fig.suptitle(s, fontsize=14, fontweight='bold')
plt.title('Bar Graph: frequency distribution of each outcome v/s outcomes for n = 100000')
plt.xlabel('Outcomes')
plt.ylabel('Frequency')
plt.bar(X,Y)
plt.style.use('ggplot')
plt.show()


mean = 0
for i in range(len(outcomes_sum_probability)):
   mean += (i+2)*outcomes_sum_probability[i]
```

```
mode = 0
mode_freq = max(outcomes_sum_count)
for i in range(11):
    if mode_freq == outcomes_sum_count[i]:
        mode = i+2
s = "Mean:"+str(mean)+" and Mode: "+str(mode)+" with frequency = " +str(mode_freq)

fig = plt.figure()
p1 = [3/20, 3/20, 1/4, 3/20, 3/20, 3/20]
p2 = [1/6, 1/6, 1/6, 1/6, 1/6, 1/6]
p = [1/40, 1/40, 1/24, 1/40, 1/40, 1/40, 1/40, 1/40, 1/40, 1/40, 1/40]
calculated_probability_of_sum = []
calculated_probability_of_sum += [(p1[0]*p2[0])]
#2  (1, 1)

calculated_probability_of_sum += [((p1[0]*p2[1]) + (p1[1]*p2[0]))]
#3  (1,2), (2,1)

calculated_probability_of_sum += [((p1[0]*p2[2]) + (p1[2]*p2[0]) + (p1[1]*p2[1]))]
#4  (1,3), (3,1), (2, 2)

calculated_probability_of_sum += [((p1[0]*p2[3]) + (p1[3]*p2[0]) + (p1[1]*p2[2]) + (p1[2]*p2[1]))]
#5  (1,4), (4,1), (2, 3), (3, 2)

calculated_probability_of_sum += [((p1[0]*p2[4]) + (p1[4]*p2[0]) + (p1[1]*p2[3]) + (p1[3]*p2[1]) + (p1[2]*p2[2]))]
#6  (1,5), (5,1), (2, 4), (4, 2), (3, 3)

calculated_probability_of_sum += [((p1[0]*p2[5]) + (p1[5]*p2[0]) + (p1[1]*p2[4]) + (p1[4]*p2[1]) + (p1[2]*p2[3]) +
(p1[3]*p2[2]))]
#7  (1,6), (6,1), (2, 5), (5, 2), (3, 4), (4, 3)

calculated_probability_of_sum += [((p1[1]*p2[5]) + (p1[5]*p2[1]) + (p1[2]*p2[4]) + (p1[4]*p2[2]) + (p1[3]*p2[3]))]
#8  (2,6), (6,2), (3, 5), (5, 3), (4, 4)

calculated_probability_of_sum += [((p1[2]*p2[5]) + (p1[5]*p2[2]) + (p1[3]*p2[4]) + (p1[4]*p2[3]))]
#9  (3,6), (6,3), (4, 5), (5, 4)

calculated_probability_of_sum += [((p1[3]*p2[5]) + (p1[5]*p2[3]) + (p1[4]*p2[4]))]
#10  (4,6), (6,4), (5, 5)

calculated_probability_of_sum += [((p1[4]*p2[5]) + (p1[5]*p2[4]))]
#11  (5,6), (6,5)

calculated_probability_of_sum += [((p1[5]*p2[5]))]
#12  (6, 6)

print(calculated_probability_of_sum)
plt.scatter(X2, calculated_probability_of_sum, label="Calculated probability of sum", color = 'pink')
plt.plot(X2, calculated_probability_of_sum, color = 'pink')
plt.scatter(X2, outcomes_sum_probability,  label="Simulated probability of sum with "+s,color = 'purple')
plt.title('Graph: Probability of sum, probability of the sum v/s sum of outcomes on the two dice, n = 100000')
plt.xlabel('sum of outcomes on two dice')
plt.ylabel('Probability of the sum')
plt.style.use('ggplot')
plt.grid(True)
plt.legend()
plt.show()
```

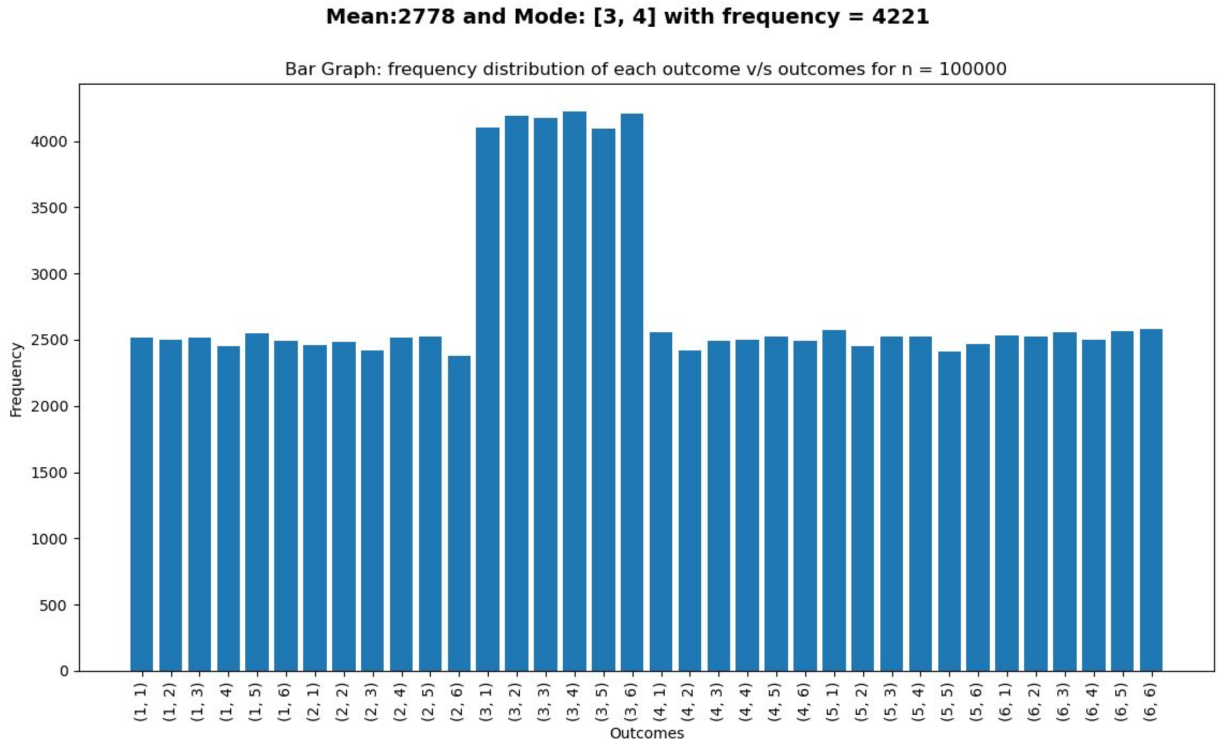
Simulation 1

Mean:2778 and Mode: [3, 3] with frequency = 4259


Bar Graph: frequency distribution of each outcome v/s outcomes for n = 100000

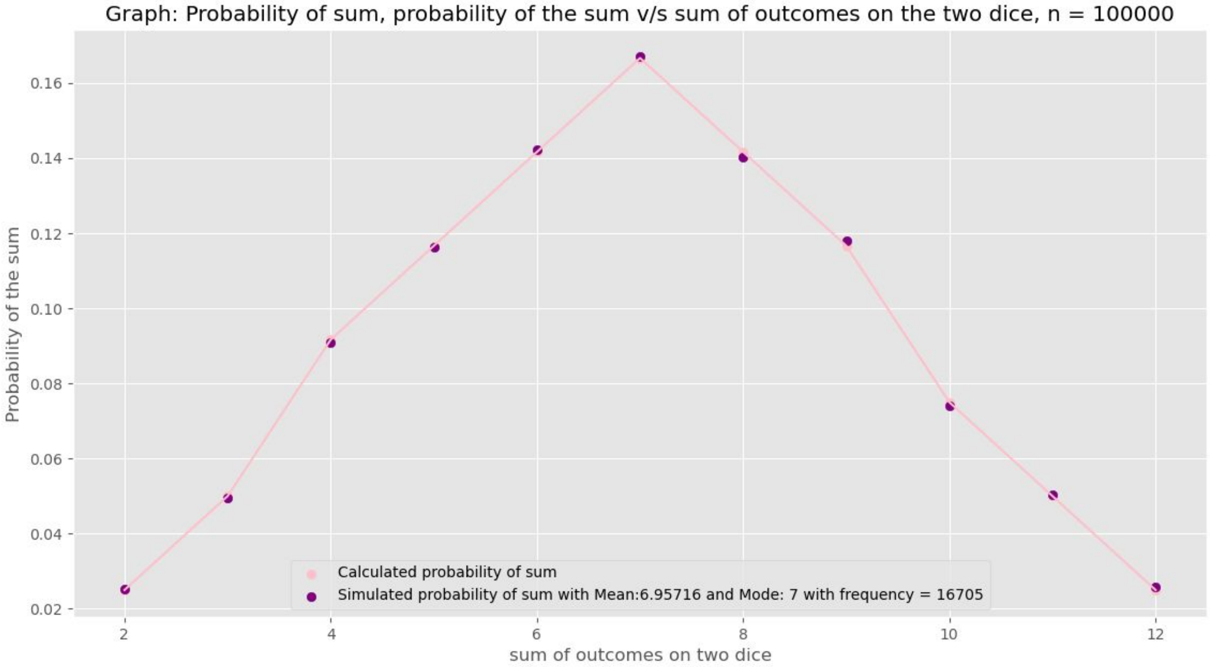

Graph: Probability of sum, probability of the sum v/s sum of outcomes on the two dice, n = 100000

Simulation 2



Mean:2778 and Mode: [3, 4] with frequency = 4221

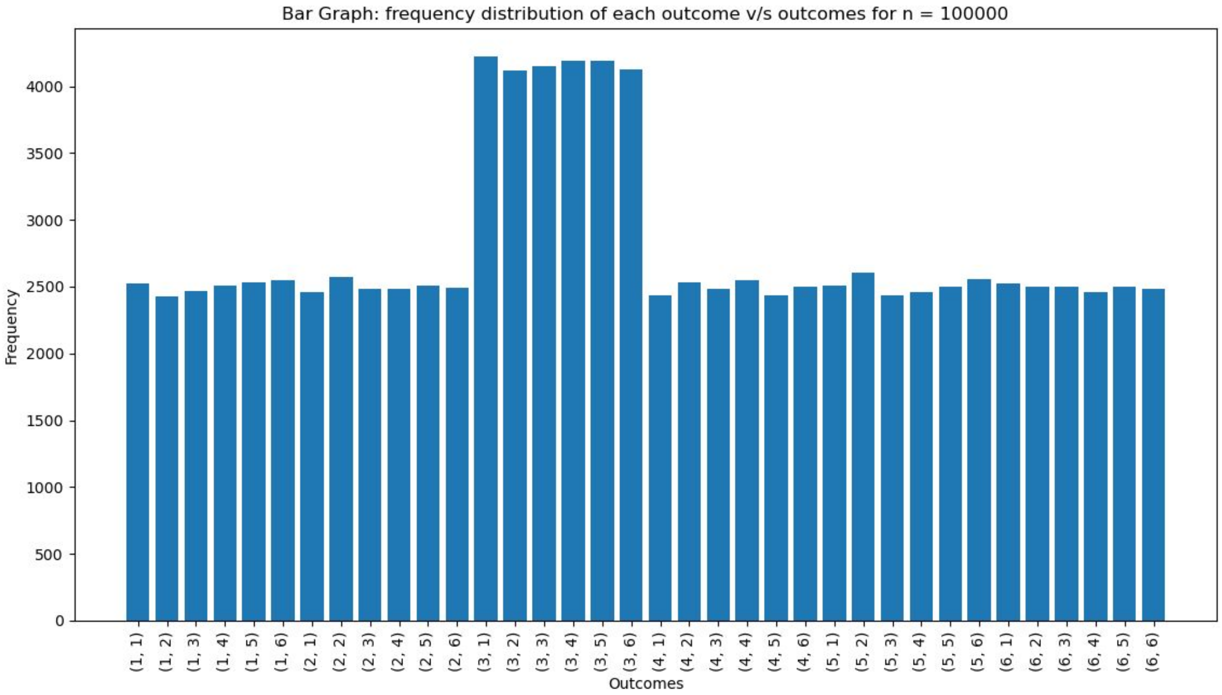
Bar Graph: frequency distribution of each outcome v/s outcomes for n = 100000



Graph: Probability of sum, probability of the sum v/s sum of outcomes on the two dice, n = 100000

Simulation 3

Mean:2778 and Mode: [3, 1] with frequency = 4221



Inferences:

The graphs plotted clearly shows that the theoretical probability (calculated probability) and probability calculated using Monte Carlo simulations (simulated probability) are the same and verified properly for multiple simulations. It can be seen that the mean and mode of the data are approximately equal and hence, the simulation was carried out successfully.