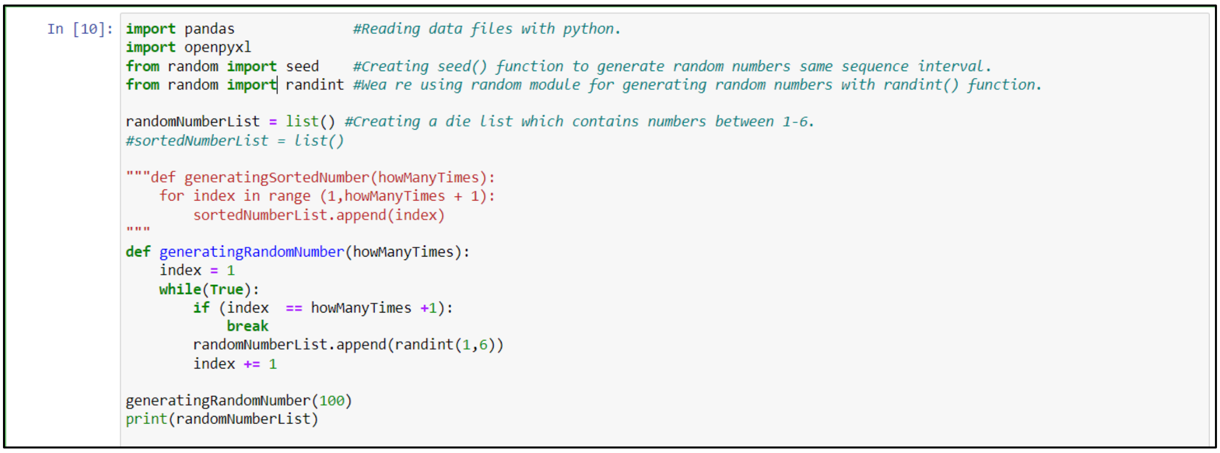
GE 204 PROJECT 1 DATA

STEP 1:

- I am using Python for generate randomly number.

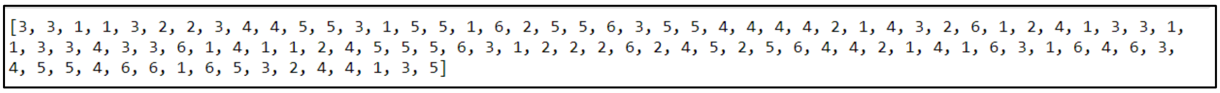
STEP 2:

- The code of the first 100 rolls experiment:



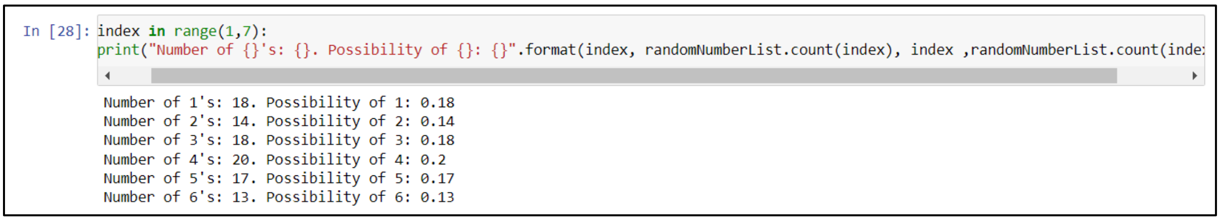
```
In [10]: import pandas              #Reading data files with python.
         import openpyxl
         from random import seed    #Creating seed() function to generate random numbers same sequence interval.
         from random import randint #Wea re using random module for generating random numbers with randint() function.

         randomNumberList = list() #Creating a die list which contains numbers between 1-6.
         #sortedNumberList = list()

         """def generatingSortedNumber(howManyTimes):
             for index in range (1,howManyTimes + 1):
                 sortedNumberList.append(index)
         """
         def generatingRandomNumber(howManyTimes):
             index = 1
             while(True):
                 if (index  == howManyTimes +1):
                     break
                 randomNumberList.append(randint(1,6))
                 index += 1

         generatingRandomNumber(100)
         print(randomNumberList)
```

- Output of the code overall first 100 rolls in array format as following:



```
[3, 3, 1, 1, 3, 2, 2, 3, 4, 4, 5, 5, 3, 1, 5, 5, 1, 6, 2, 5, 5, 6, 3, 5, 5, 4, 4, 4, 4, 2, 1, 4, 3, 2, 6, 1, 2, 4, 1, 3, 3, 1,
1, 3, 3, 4, 3, 3, 6, 1, 4, 1, 1, 2, 4, 5, 5, 5, 6, 3, 1, 2, 2, 2, 6, 2, 4, 5, 2, 5, 6, 4, 4, 2, 1, 4, 1, 6, 3, 1, 6, 4, 6, 3,
4, 5, 5, 4, 6, 6, 1, 6, 5, 3, 2, 4, 4, 1, 3, 5]
```

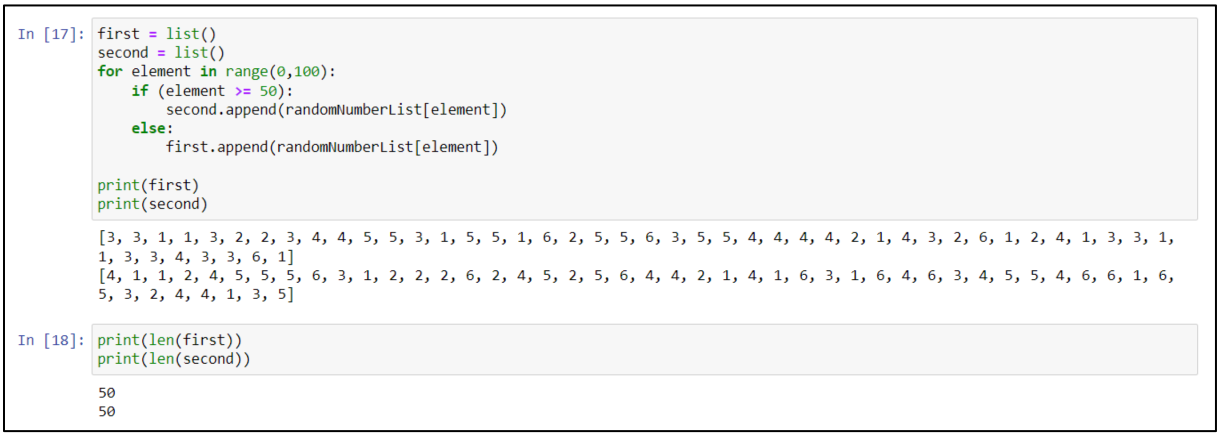
- Total times of the experiment in first 100 rolls:



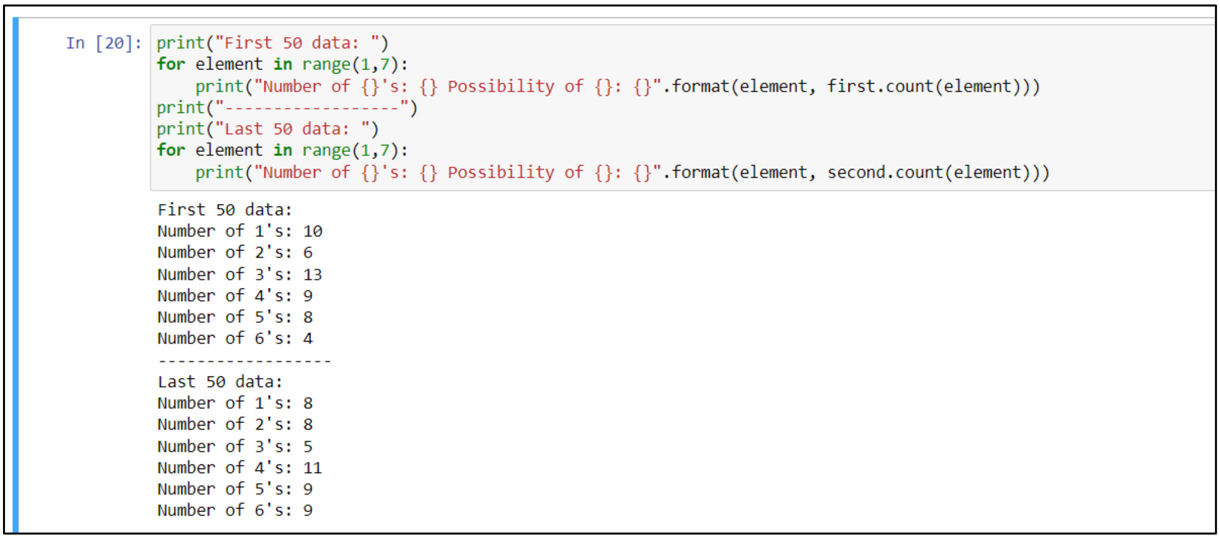
```
In [28]: index in range(1,7):
         print("Number of {}'s: {}. Possibility of {}: {}".format(index, randomNumberList.count(index), index ,randomNumberList.count(inde

          Number of 1's: 18. Possibility of 1: 0.18
          Number of 2's: 14. Possibility of 2: 0.14
          Number of 3's: 18. Possibility of 3: 0.18
          Number of 4's: 20. Possibility of 4: 0.2
          Number of 5's: 17. Possibility of 5: 0.17
          Number of 6's: 13. Possibility of 6: 0.13
```

STEP 3:

- The code of the FIRST AND LAST 50 rolls are shown and make sure that the total numbers of each equal 100.



```
In [17]: first = list()
         second = list()
         for element in range(0,100):
             if (element >= 50):
                 second.append(randomNumberList[element])
             else:
                 first.append(randomNumberList[element])

         print(first)
         print(second)

         [3, 3, 1, 1, 3, 2, 2, 3, 4, 4, 5, 5, 3, 1, 5, 5, 1, 6, 2, 5, 5, 6, 3, 5, 5, 4, 4, 4, 4, 2, 1, 4, 3, 2, 6, 1, 2, 4, 1, 3, 3, 1,
         1, 3, 3, 4, 3, 3, 6, 1]
         [4, 1, 1, 2, 4, 5, 5, 5, 6, 3, 1, 2, 2, 2, 6, 2, 4, 5, 2, 5, 6, 4, 4, 2, 1, 4, 1, 6, 3, 1, 6, 4, 6, 3, 4, 5, 5, 4, 6, 6, 1, 6,
         5, 3, 2, 4, 4, 1, 3, 5]

In [18]: print(len(first))
         print(len(second))

         50
         50
```

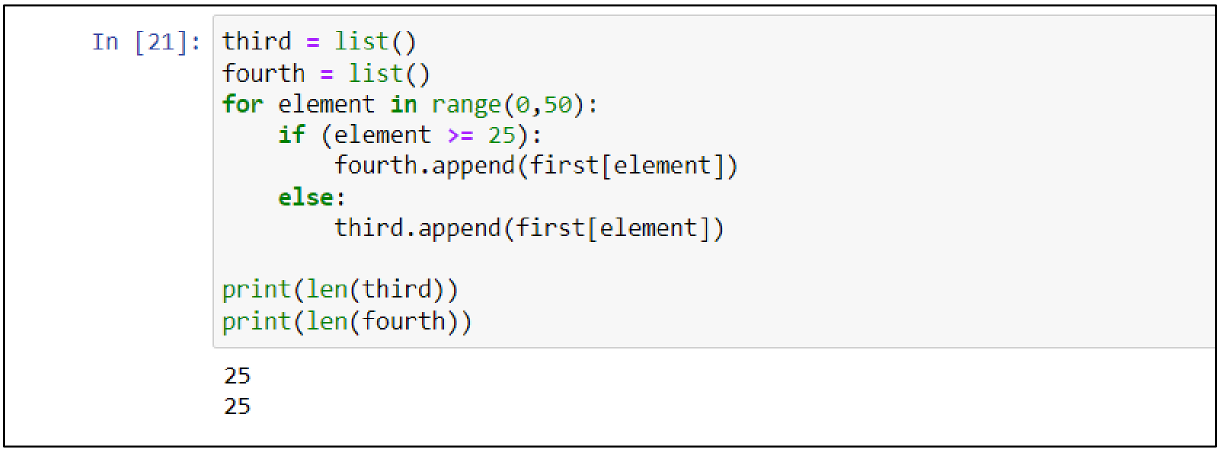


In [20]:

```
print("First 50 data: ")
for element in range(1,7):
    print("Number of {}'s: {} Possibility of {}: {}".format(element, first.count(element)))
print("------------------")
print("Last 50 data: ")
for element in range(1,7):
    print("Number of {}'s: {} Possibility of {}: {}".format(element, second.count(element)))
```

```
First 50 data:
Number of 1's: 10
Number of 2's: 6
Number of 3's: 13
Number of 4's: 9
Number of 5's: 8
Number of 6's: 4
------------------
Last 50 data:
Number of 1's: 8
Number of 2's: 8
Number of 3's: 5
Number of 4's: 11
Number of 5's: 9
Number of 6's: 9
```

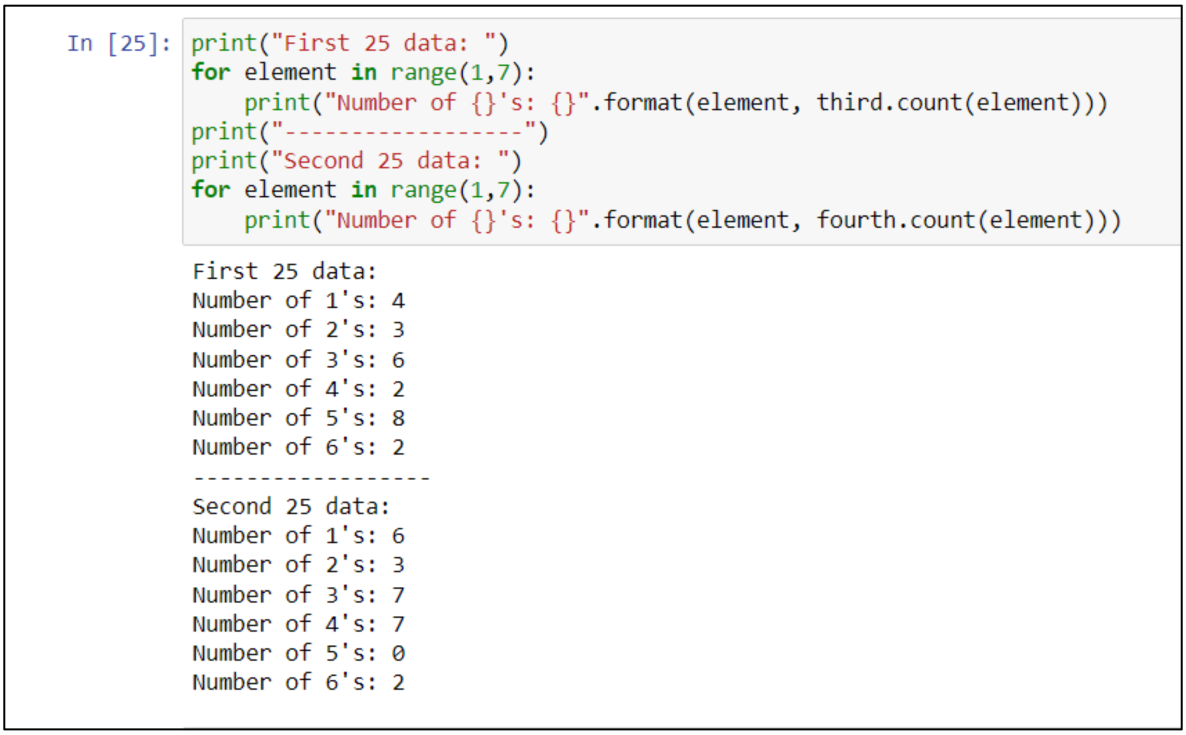
- FIRST AND SECOND 25 rolls are shown. Additionally, I want to make sure that total numbers of first and second 25 are equal 50.



In [21]:

```
third = list()
fourth = list()
for element in range(0,50):
    if (element >= 25):
        fourth.append(first[element])
    else:
        third.append(first[element])

print(len(third))
print(len(fourth))
```

```
25
25
```

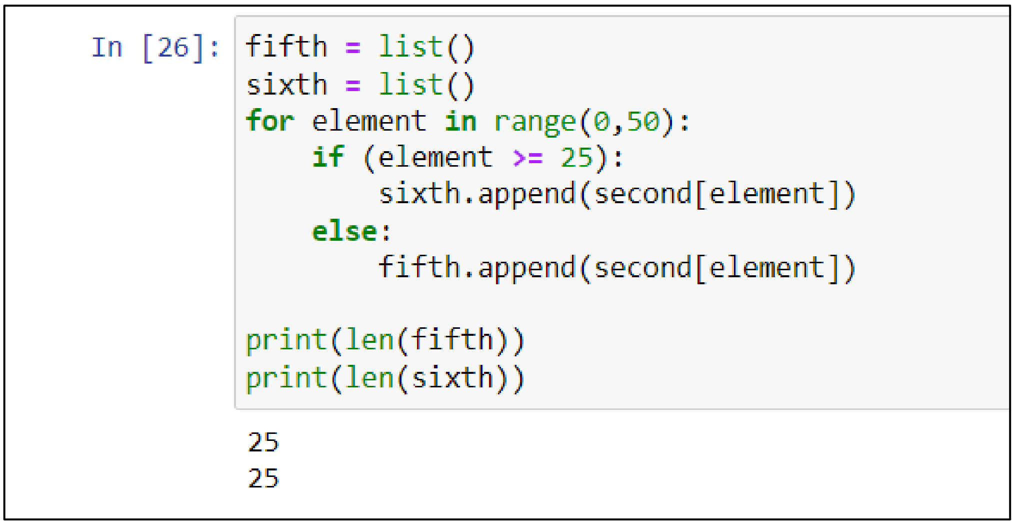


In [25]:

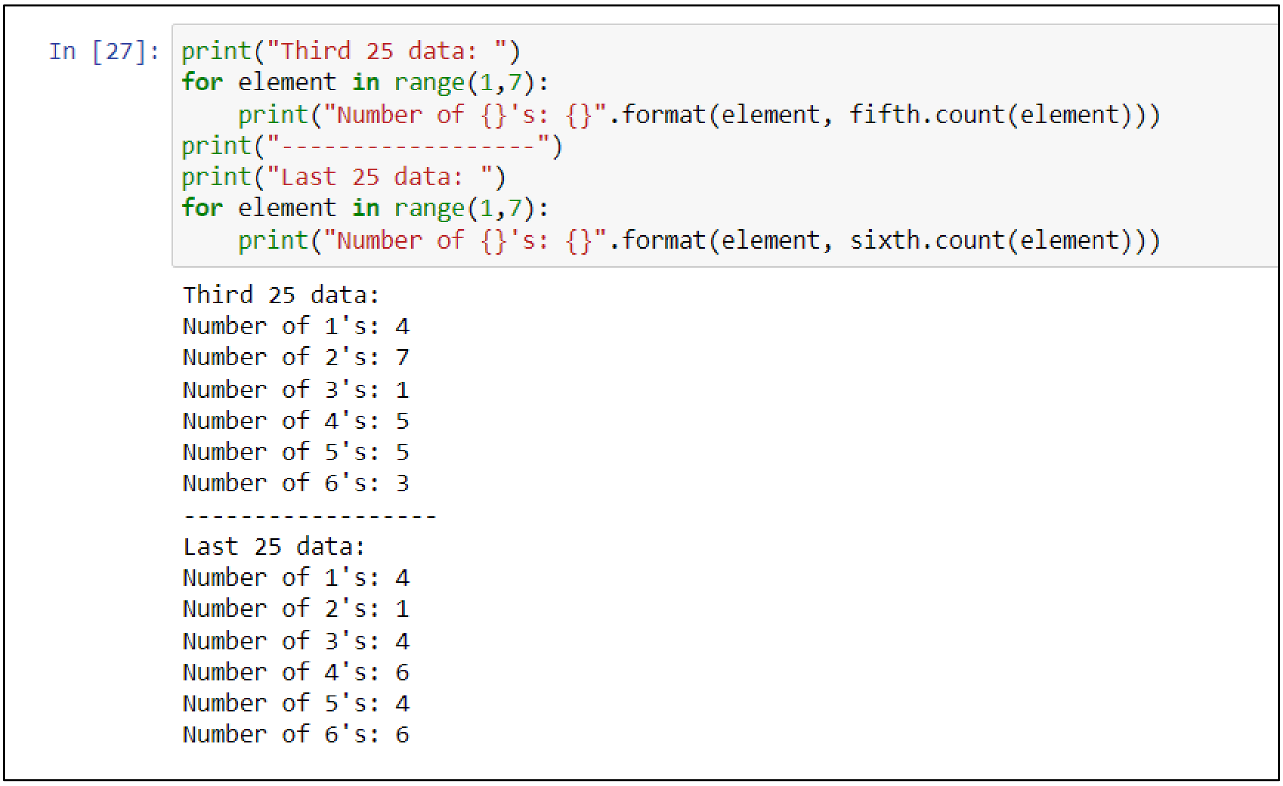
```
print("First 25 data: ")
for element in range(1,7):
    print("Number of {}'s: {}".format(element, third.count(element)))
print("------------------")
print("Second 25 data: ")
for element in range(1,7):
    print("Number of {}'s: {}".format(element, fourth.count(element)))
```

```
First 25 data:
Number of 1's: 4
Number of 2's: 3
Number of 3's: 6
Number of 4's: 2
Number of 5's: 8
Number of 6's: 2
------------------
Second 25 data:
Number of 1's: 6
Number of 2's: 3
Number of 3's: 7
Number of 4's: 7
Number of 5's: 0
Number of 6's: 2
```

- Probabilities of the third and fourth 25 are shown. I want to make sure that total numbers are THIRD AND FOURTH 25 rolls are equal 50:

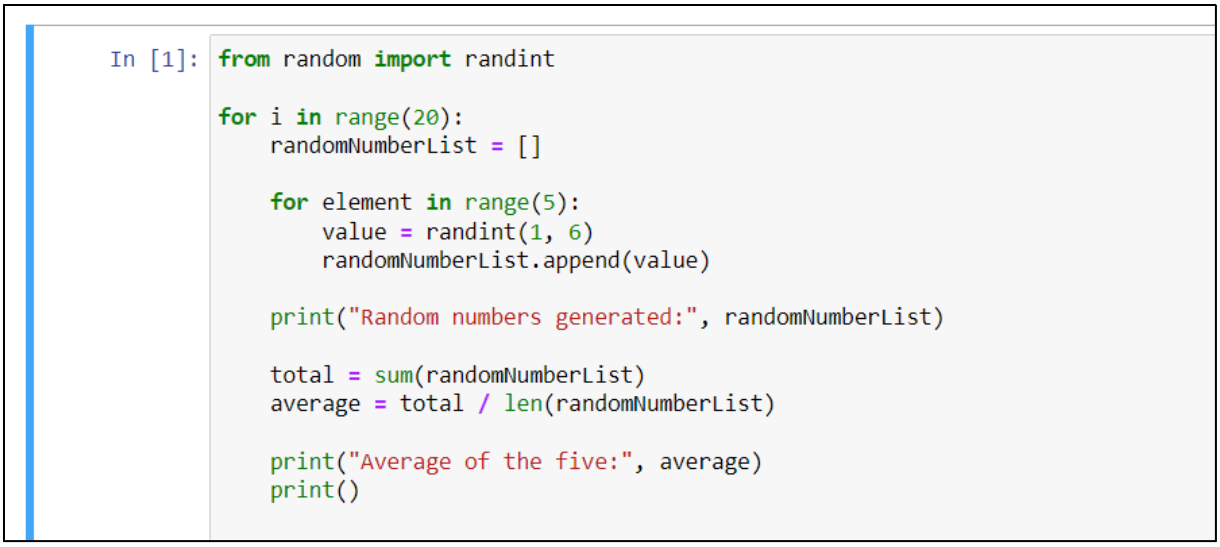


```
In [26]: fifth = list()
         sixth = list()
         for element in range(0,50):
             if (element >= 25):
                 sixth.append(second[element])
             else:
                 fifth.append(second[element])

         print(len(fifth))
         print(len(sixth))

         25
         25
```



```
In [27]: print("Third 25 data: ")
         for element in range(1,7):
             print("Number of {}'s: {}".format(element, fifth.count(element)))
         print("------------------")
         print("Last 25 data: ")
         for element in range(1,7):
             print("Number of {}'s: {}".format(element, sixth.count(element)))

         Third 25 data:
         Number of 1's: 4
         Number of 2's: 7
         Number of 3's: 1
         Number of 4's: 5
         Number of 5's: 5
         Number of 6's: 3
         ------------------
         Last 25 data:
         Number of 1's: 4
         Number of 2's: 1
         Number of 3's: 4
         Number of 4's: 6
         Number of 5's: 4
         Number of 6's: 6
```

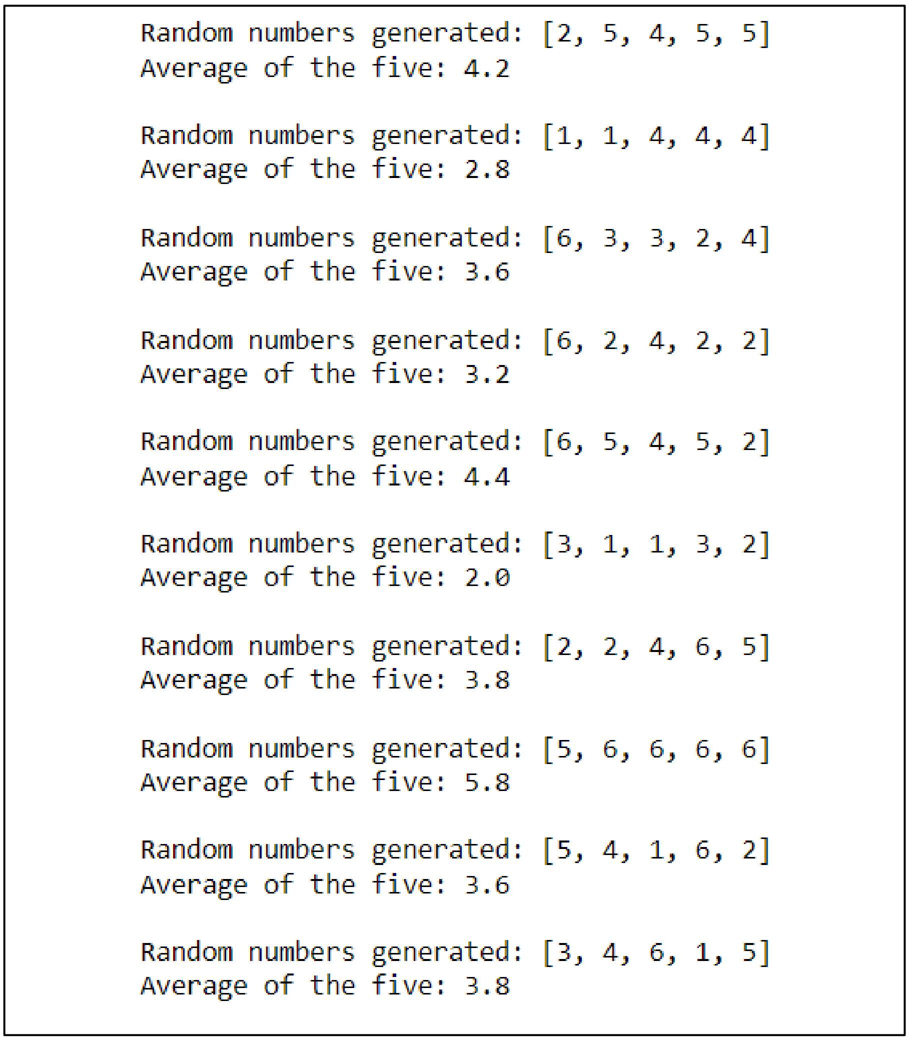
STEP 4: SECOND 100 ROLLS STAGE



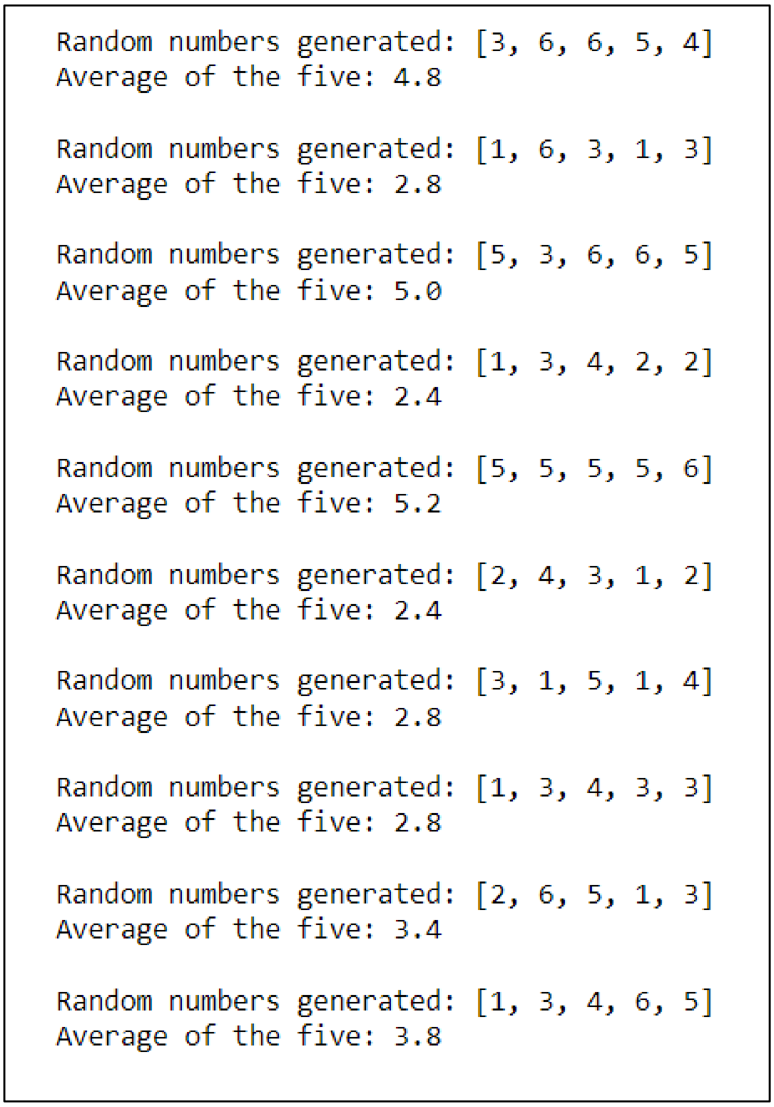
In [1]:

```python
from random import randint

for i in range(20):
    randomNumberList = []

    for element in range(5):
        value = randint(1, 6)
        randomNumberList.append(value)

    print("Random numbers generated:", randomNumberList)

    total = sum(randomNumberList)
    average = total / len(randomNumberList)

    print("Average of the five:", average)
    print()
```

- Output of the code above:

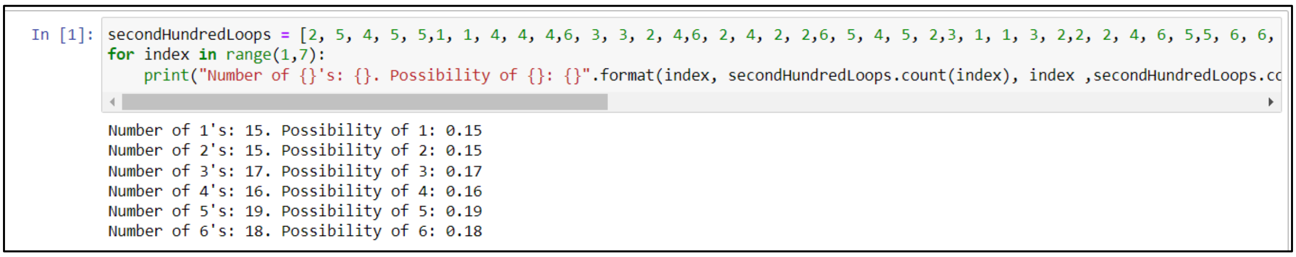


```
Random numbers generated: [2, 5, 4, 5, 5]
Average of the five: 4.2

Random numbers generated: [1, 1, 4, 4, 4]
Average of the five: 2.8

Random numbers generated: [6, 3, 3, 2, 4]
Average of the five: 3.6

Random numbers generated: [6, 2, 4, 2, 2]
Average of the five: 3.2

Random numbers generated: [6, 5, 4, 5, 2]
Average of the five: 4.4

Random numbers generated: [3, 1, 1, 3, 2]
Average of the five: 2.0

Random numbers generated: [2, 2, 4, 6, 5]
Average of the five: 3.8

Random numbers generated: [5, 6, 6, 6, 6]
Average of the five: 5.8

Random numbers generated: [5, 4, 1, 6, 2]
Average of the five: 3.6

Random numbers generated: [3, 4, 6, 1, 5]
Average of the five: 3.8
```



```
Random numbers generated: [3, 6, 6, 5, 4]
Average of the five: 4.8

Random numbers generated: [1, 6, 3, 1, 3]
Average of the five: 2.8

Random numbers generated: [5, 3, 6, 6, 5]
Average of the five: 5.0

Random numbers generated: [1, 3, 4, 2, 2]
Average of the five: 2.4

Random numbers generated: [5, 5, 5, 5, 6]
Average of the five: 5.2

Random numbers generated: [2, 4, 3, 1, 2]
Average of the five: 2.4

Random numbers generated: [3, 1, 5, 1, 4]
Average of the five: 2.8

Random numbers generated: [1, 3, 4, 3, 3]
Average of the five: 2.8

Random numbers generated: [2, 6, 5, 1, 3]
Average of the five: 3.4

Random numbers generated: [1, 3, 4, 6, 5]
Average of the five: 3.8
```

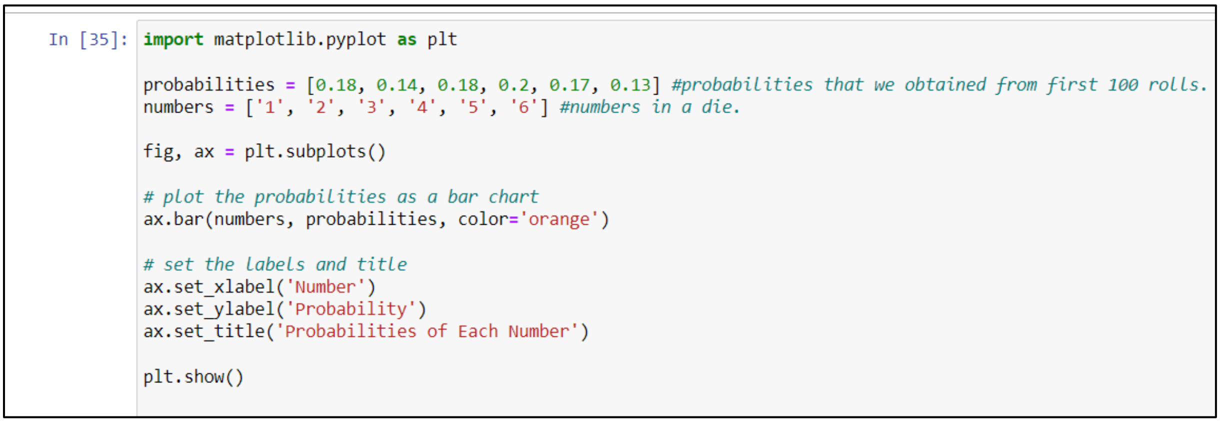
- Additionally, total number and possibilities of the second 100 rolls:



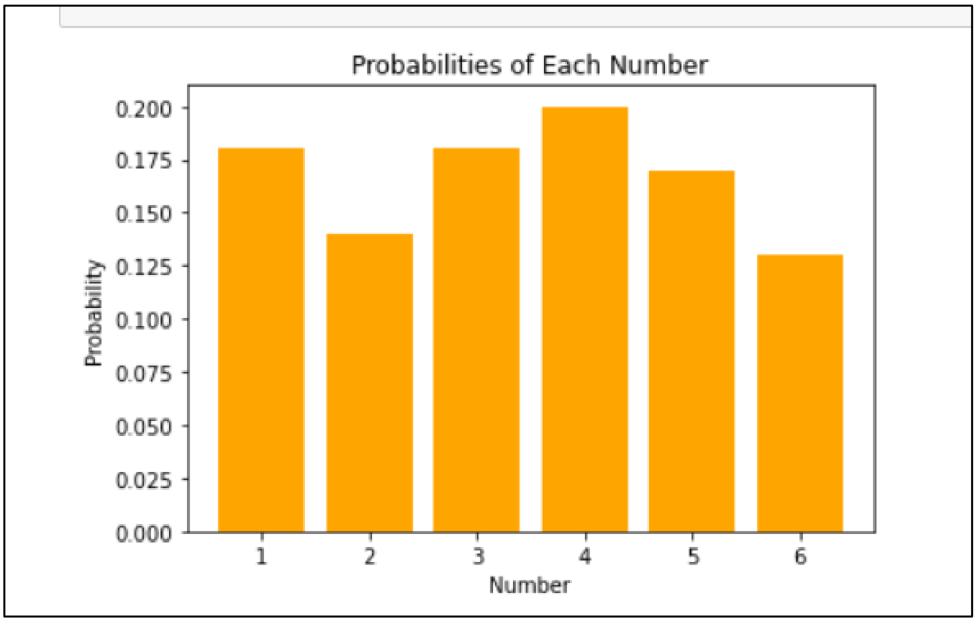
```
In [1]: secondHundredLoops = [2, 5, 4, 5, 5,1, 1, 4, 4, 4,6, 3, 3, 2, 4,6, 2, 4, 2, 2,6, 5, 4, 5, 2,3, 1, 1, 3, 2,2, 2, 4, 6, 5,5, 6, 6,
        for index in range(1,7):
            print("Number of {}'s: {}. Possibility of {}: {}".format(index, secondHundredLoops.count(index), index ,secondHundredLoops.cc

        Number of 1's: 15. Possibility of 1: 0.15
        Number of 2's: 15. Possibility of 2: 0.15
        Number of 3's: 17. Possibility of 3: 0.17
        Number of 4's: 16. Possibility of 4: 0.16
        Number of 5's: 19. Possibility of 5: 0.19
        Number of 6's: 18. Possibility of 6: 0.18
```

STEP 5:

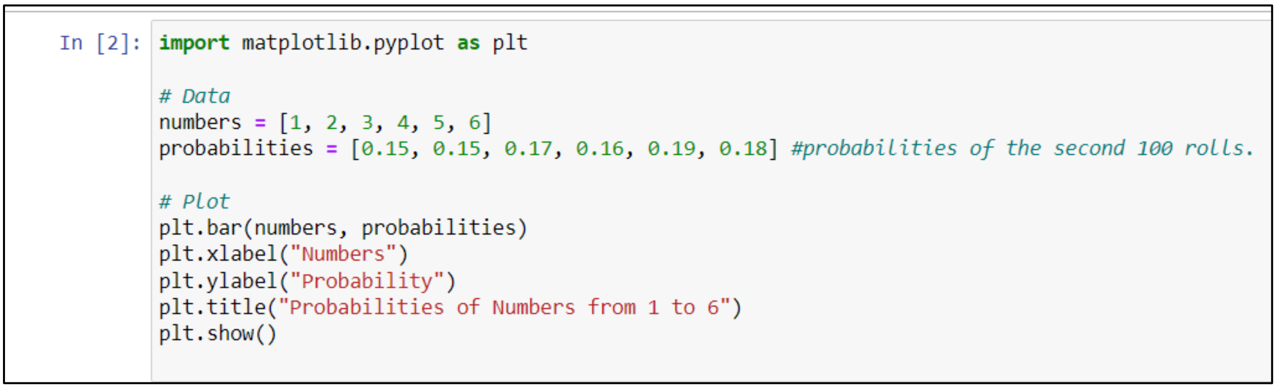
1. ***Plotting results for first 100 rolls:***



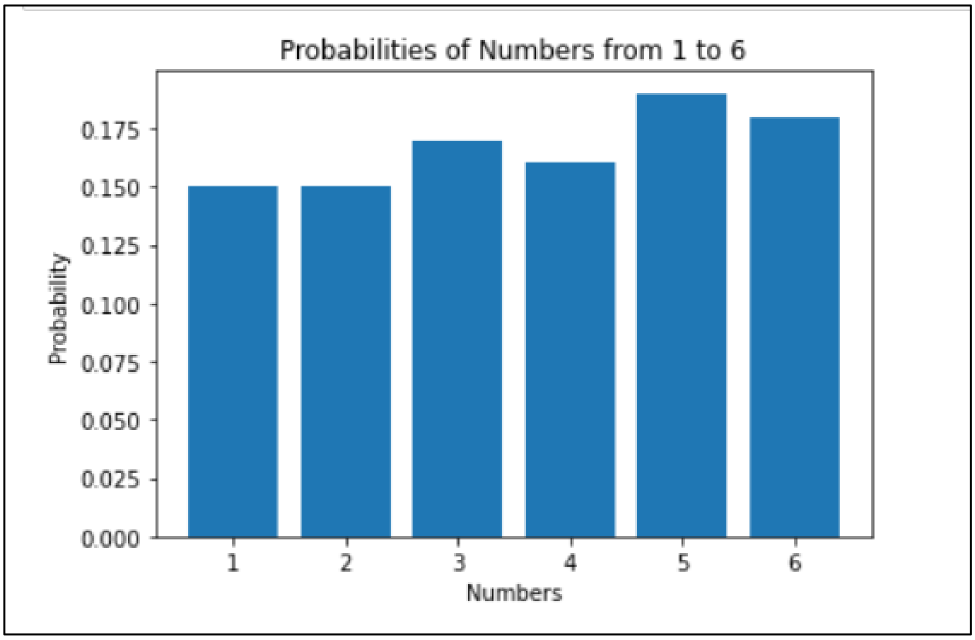
```
In [35]: import matplotlib.pyplot as plt

         probabilities = [0.18, 0.14, 0.18, 0.2, 0.17, 0.13] #probabilities that we obtained from first 100 rolls.
         numbers = ['1', '2', '3', '4', '5', '6'] #numbers in a die.

         fig, ax = plt.subplots()

         # plot the probabilities as a bar chart
         ax.bar(numbers, probabilities, color='orange')

         # set the labels and title
         ax.set_xlabel('Number')
         ax.set_ylabel('Probability')
         ax.set_title('Probabilities of Each Number')

         plt.show()
```



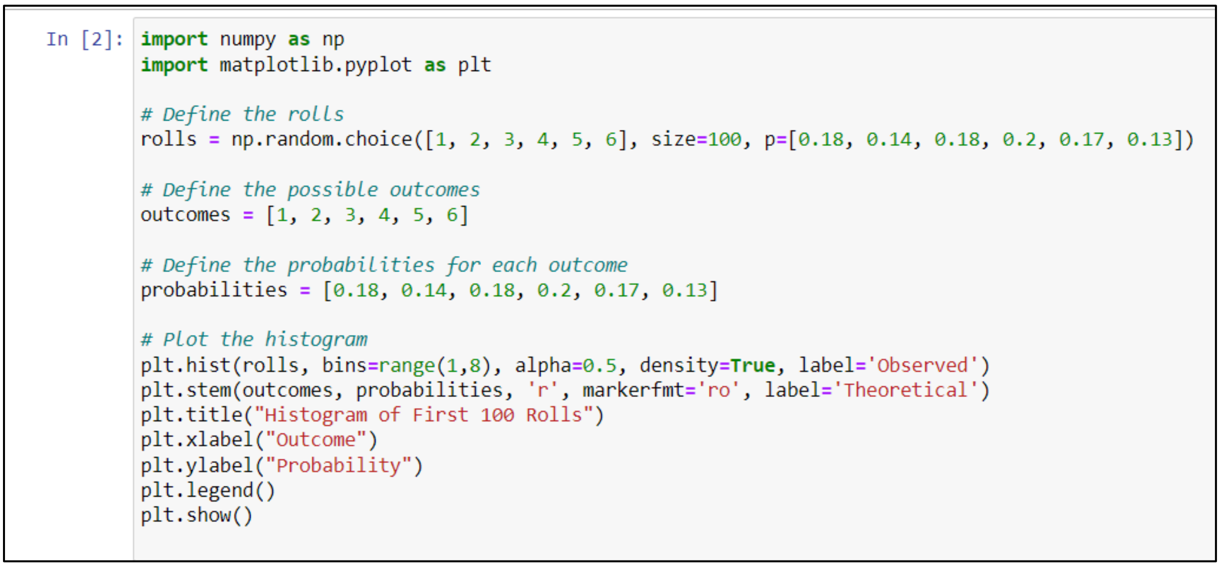
2. ***Plotting results for second 100 rolls:***



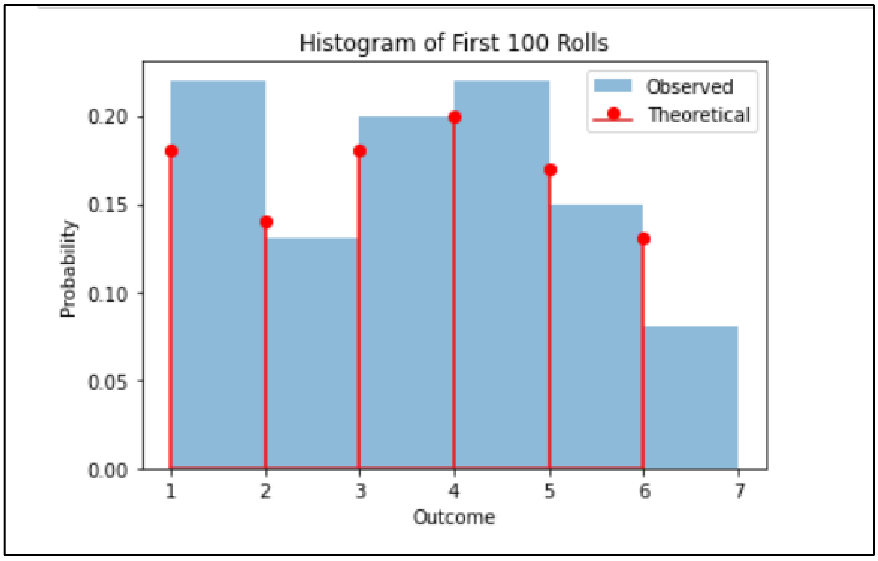
```
In [2]: import matplotlib.pyplot as plt

        # Data
        numbers = [1, 2, 3, 4, 5, 6]
        probabilities = [0.15, 0.15, 0.17, 0.16, 0.19, 0.18] #probabilities of the second 100 rolls.

        # Plot
        plt.bar(numbers, probabilities)
        plt.xlabel("Numbers")
        plt.ylabel("Probability")
        plt.title("Probabilities of Numbers from 1 to 6")
        plt.show()
```



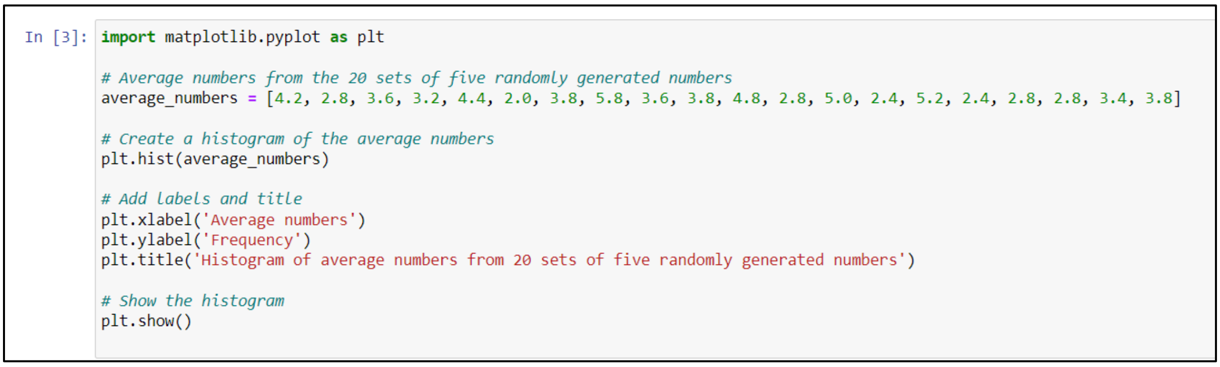
3. Histogram for the first 100 rolls obtained from step 2:



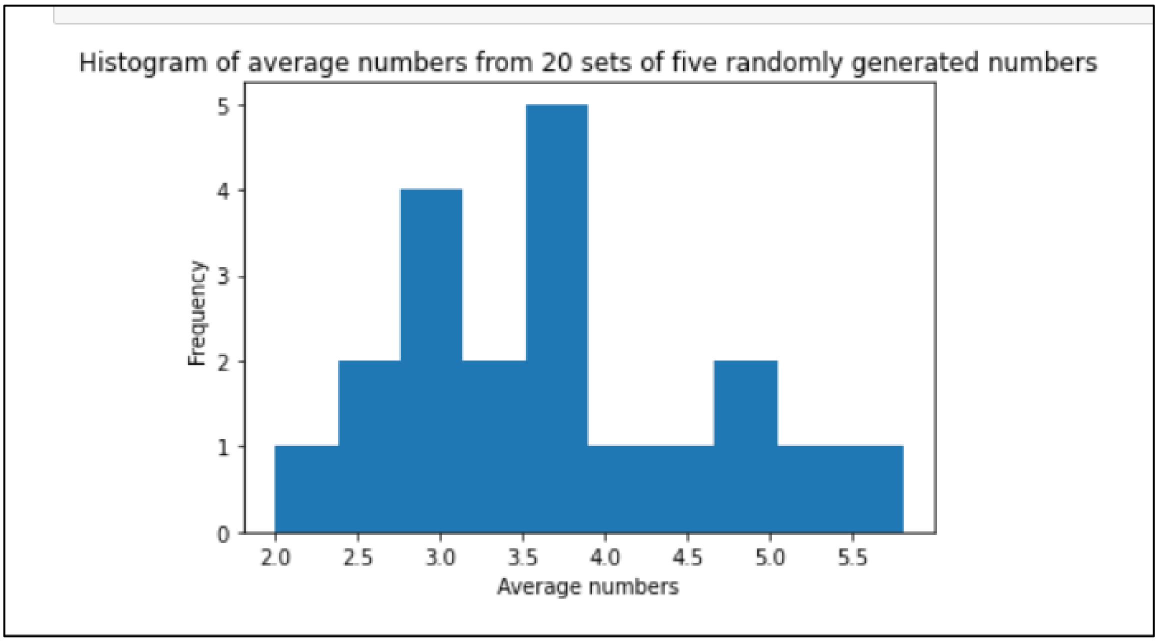
```
In [2]: import numpy as np
        import matplotlib.pyplot as plt

        # Define the rolls
        rolls = np.random.choice([1, 2, 3, 4, 5, 6], size=100, p=[0.18, 0.14, 0.18, 0.2, 0.17, 0.13])

        # Define the possible outcomes
        outcomes = [1, 2, 3, 4, 5, 6]

        # Define the probabilities for each outcome
        probabilities = [0.18, 0.14, 0.18, 0.2, 0.17, 0.13]

        # Plot the histogram
        plt.hist(rolls, bins=range(1,8), alpha=0.5, density=True, label='Observed')
        plt.stem(outcomes, probabilities, 'r', markerfmt='ro', label='Theoretical')
        plt.title("Histogram of First 100 Rolls")
        plt.xlabel("Outcome")
        plt.ylabel("Probability")
        plt.legend()
        plt.show()
```



4. *Histogram for the averages obtained from step 4:*



```
In [3]: import matplotlib.pyplot as plt

        # Average numbers from the 20 sets of five randomly generated numbers
        average_numbers = [4.2, 2.8, 3.6, 3.2, 4.4, 2.0, 3.8, 5.8, 3.6, 3.8, 4.8, 2.8, 5.0, 2.4, 5.2, 2.4, 2.8, 2.8, 3.4, 3.8]

        # Create a histogram of the average numbers
        plt.hist(average_numbers)

        # Add labels and title
        plt.xlabel('Average numbers')
        plt.ylabel('Frequency')
        plt.title('Histogram of average numbers from 20 sets of five randomly generated numbers')

        # Show the histogram
        plt.show()
```


Histogram of average numbers from 20 sets of five randomly generated numbers
Frequency
Average numbers