Don Bosco Institute of Technology, Kurla, Mumbai

SE SEM-IV Subject EM IV

MINI PROJECT- Probability Distribution Modelling

Academic Year: 2023-2024

Team members worked on this Project:

- 45 - Saniya Padwal
- 61 - Priyanshi Singh
- 65 - Niranjan Kumar Yadav

Contents of the Report:

Problem Statement:

A dice throwing experiment involves rolling two dice and recording the sum of the numbers obtained. The goal is to analyze the frequency distribution of the sums and observe how the distribution evolves with an increasing number of trials.

Record the number of times you obtain the sum as greater than 5.
Next perform 10 trials of throwing two dice for 10 times.
Step 1: Find the number of times you obtain the sum as greater than 5, in each trial- that is, Count Step 2: Draw the graph of Count vs frequency for all the trials put together.
Next perform 20 trials of throwing two dice for 10 times.
Repeat Steps 1 and 2.
Increase the number of trials to 30, 40, · · ·, 100, 200, · · ·, 1000, · · · Repeat Steps 1 and 2 in each case.

Objective:

The objective is to understand how the frequency distribution of the sums of two dice rolls evolves with an increasing number of trials and to identify any underlying distribution governing the outcomes.

Acknowledgement:

We would like to express gratitude to our Math's Faculty for giving us the opportunity to work on this project and hence arrive at a derived conclusion through their unwavering support and guidance.

The project concerning data analysis was done using real life application of statistical techniques. Secondly, a big thanks to our batch mates for helping us finding the necessary data and plotting the graphs.

Abstract:

In this Mini-Project we have successfully analyzed the given experiment to plot the required graphs and their data.

The main aim of the project was to obtain the frequency distribution of the data and hence the probability distribution modelling it. We derive a concept of Normal Distribution as the number of trials keep on increasing and hence, we get a bell-shaped curve whose shape is determined by two parameters: **the mean and the standard deviation**.

We could obtain the desired outcome by analyzing the random variables, conducting multiple trials, and drawing inferences based on appropriate graphing techniques.

What Problems were Tackled?

We certainly faced some difficulties in understanding the problem statement. To facilitate stepwise breakdown of the problem, we had to initiate our research through various articles on the topic and hence use Python language to understand and build the logic of the problem statement thereby plotting the required graphs.

Resources Used:

- **PyCharm** for setting up the development environment for the project.
- **Python** for writing code, importing the necessary libraries and hence implement the algorithm to get the desired output.
- **Libraries used in Python:**

1. **NumPy:** is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.
2. **Matplotlib:** Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

Graphs and their Analysis:

1. Problem:

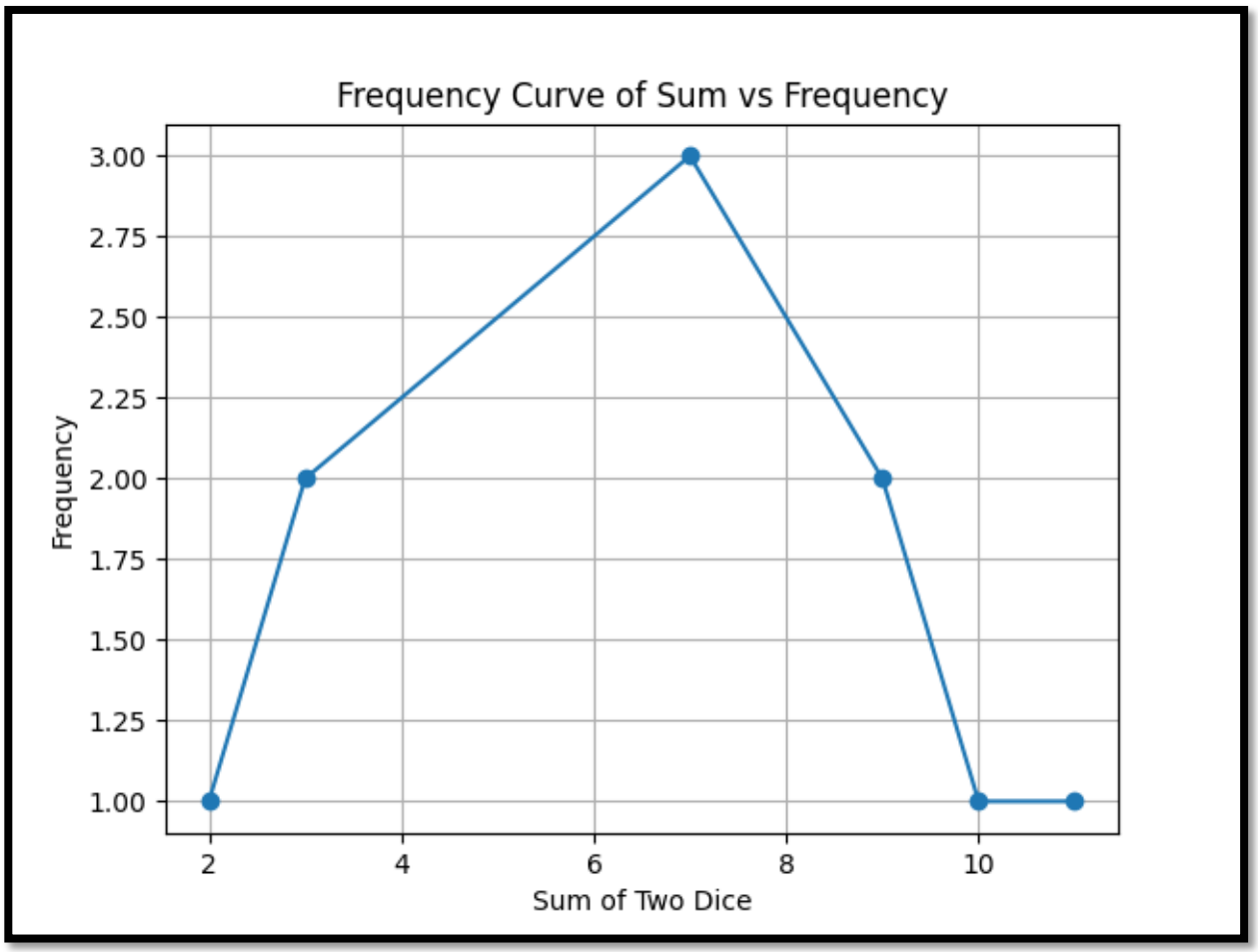
Throw two dice 10 times and record the sum of the numbers obtained.

Solution:

This following will simulate throwing two dice 10 times, record the sum of the numbers obtained, and plot the frequency curve of the sum versus frequency.

```
import random
import matplotlib.pyplot as plt


def throw_dice(num_throws):
    sums = []
    for _ in range(num_throws):
        # Simulate rolling two dice
        dice1 = random.randint(1, 6)
        dice2 = random.randint(1, 6)
        # Record the sum
        total_sum = dice1 + dice2
        sums.append(total_sum)
    return sums


def plot_frequency_curve(sums):
    frequency = {}
    # Count the frequency of each sum
    for s in sums:
        frequency[s] = frequency.get(s, 0) + 1

    # Sort the sums
    sums_sorted = sorted(frequency.keys())
    frequencies = [frequency[s] for s in sums_sorted]

    # Plot the frequency curve
    plt.plot(sums_sorted, frequencies, marker='o', linestyle='-')
    plt.xlabel('Sum of Two Dice')
    plt.ylabel('Frequency')
    plt.title('Frequency Curve of Sum vs Frequency')
    plt.grid(True)
    plt.show()


# Number of throws
num_throws = 10

# Perform the experiment
sums = throw_dice(num_throws)

# Plot the frequency curve
plot_frequency_curve(sums)
```

GRAPH:



2. Problem:

Record the number of times you obtain the sum as greater than 5

Solution:

When we run this code, it will print out the sums obtained and the number of times the sum is greater than 5, in addition to plotting the frequency curve.

```python
import random
import matplotlib.pyplot as plt


def throw_dice(num_throws):
    sums = []
    sum_greater_than_5 = 0  # Counter for sums greater than 5
    for _ in range(num_throws):
        # Simulate rolling two dice
        dice1 = random.randint(1, 6)
        dice2 = random.randint(1, 6)
        # Record the sum
        total_sum = dice1 + dice2
        sums.append(total_sum)
        # Check if sum is greater than 5
        if total_sum > 5:
            sum_greater_than_5 += 1
    return sums, sum_greater_than_5


def plot_frequency_curve(sums):
    frequency = {}
    # Count the frequency of each sum
    for s in sums:
        frequency[s] = frequency.get(s, 0) + 1

    # Sort the sums
    sums_sorted = sorted(frequency.keys())
    frequencies = [frequency[s] for s in sums_sorted]

    # Plot the frequency curve
    plt.plot(sums_sorted, frequencies, marker='o', linestyle='-')
    plt.xlabel('Sum of Two Dice')
    plt.ylabel('Frequency')
    plt.title('Frequency Curve of Sum vs Frequency')
    plt.grid(True)
    plt.show()


# Number of throws
num_throws = 10

# Perform the experiment
sums, count_greater_than_5 = throw_dice(num_throws)

print("Sums:", sums)
print("Number of times sum is greater than 5:", count_greater_than_5)

# Plot the frequency curve
plot_frequency_curve(sums)
```

Output:

Sums: [9, 10, 11, 7, 7, 9, 7, 2, 3, 3]
Number of times sum is greater than 5: - 7

Program:

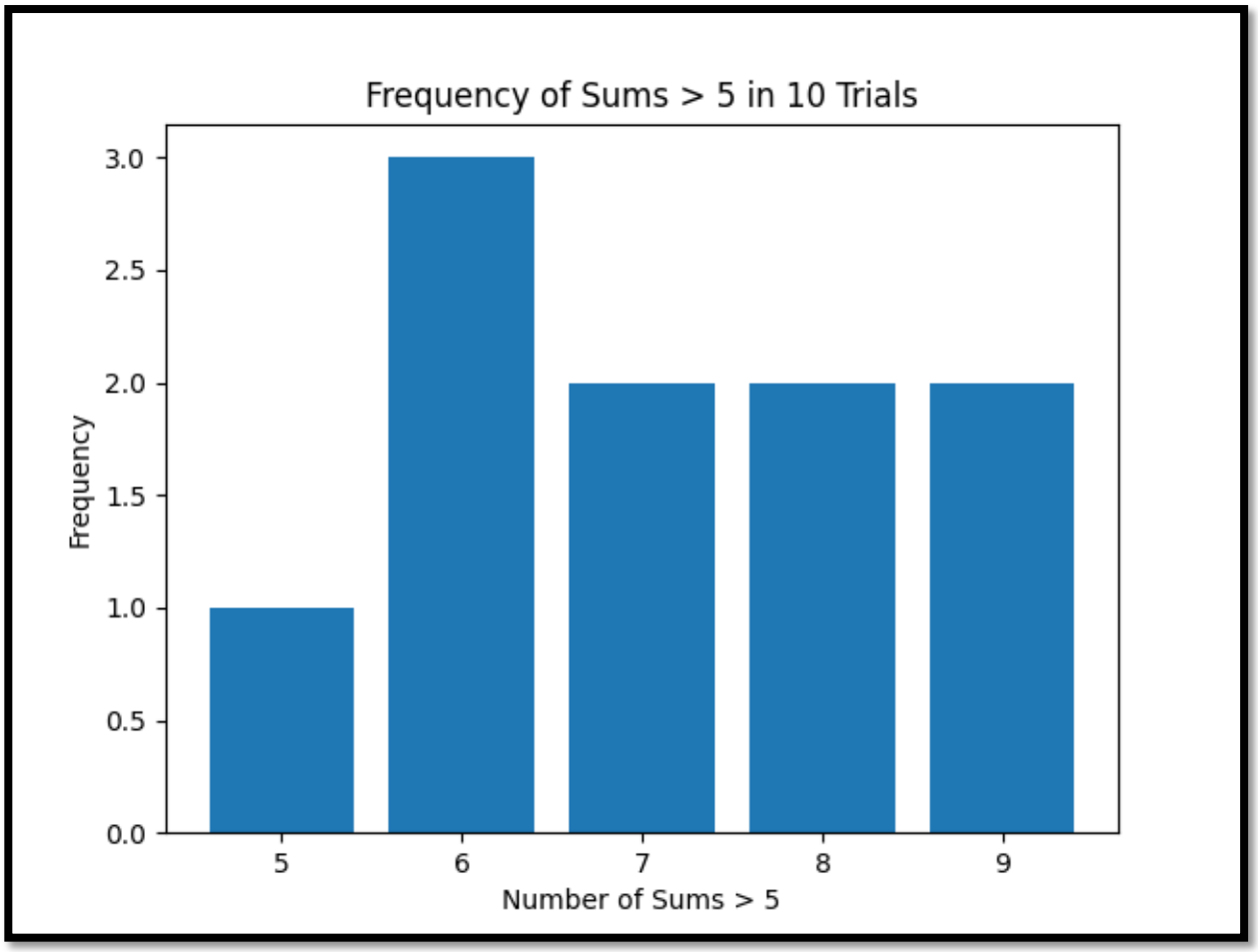
```
import numpy as np
import matplotlib.pyplot as plt


# Function to simulate throwing two dice for a given number of trials and
times
def throw_dice(num_trials, num_times):
    # Initialize arrays to store counts of sums greater than 5
    counts = np.zeros(num_trials)

    for trial in range(num_trials):
        # Roll two dice 'num_times' times
        rolls = np.random.randint(1, 7, size=(num_times, 2))
        # Calculate sum of each roll
        sums = np.sum(rolls, axis=1)
        # Count number of sums greater than 5
        counts[trial] = np.sum(sums > 5)

    return counts


# Function to plot the frequency of counts of sums greater than 5
def plot_frequency(counts, num_trials):
    # Count frequency of each count
    unique_counts, frequency = np.unique(counts, return_counts=True)

    # Plot bar graph
    plt.bar(unique_counts, frequency)
    plt.xlabel('Number of Sums > 5')
    plt.ylabel('Frequency')
    plt.title(f'Frequency of Sums > 5 in {num_trials} Trials')
    plt.show()


if __name__ == '__main__':

    # List of number of trials for each experiment
    num_trials_list = [10, 20, 30, 40, 100, 200, 1000, 10000, 100000]

    # Iterate through each number of trials
    for num_trials in num_trials_list:
        # Simulate throwing two dice for the specified number of trials and 10
times each
        counts = throw_dice(num_trials, 10)
        # Plot the frequency of counts of sums greater than 5
        plot_frequency(counts, num_trials)

        # Calculate mean, standard deviation, and total count
        mean_count = np.mean(counts)
        std_dev = np.std(counts)
        total_count = np.sum(counts)

        # Print statistics
        print(f'Number of Trials: {num_trials}')
        print(f'Mean: {mean_count}')
        print(f'Standard Deviation: {std_dev}')
```

```
        print(f'Total Count: {total_count}')
        print()
```

3. **Problem**:

    10 trials of throwing two dice for 10 times.

Solution:

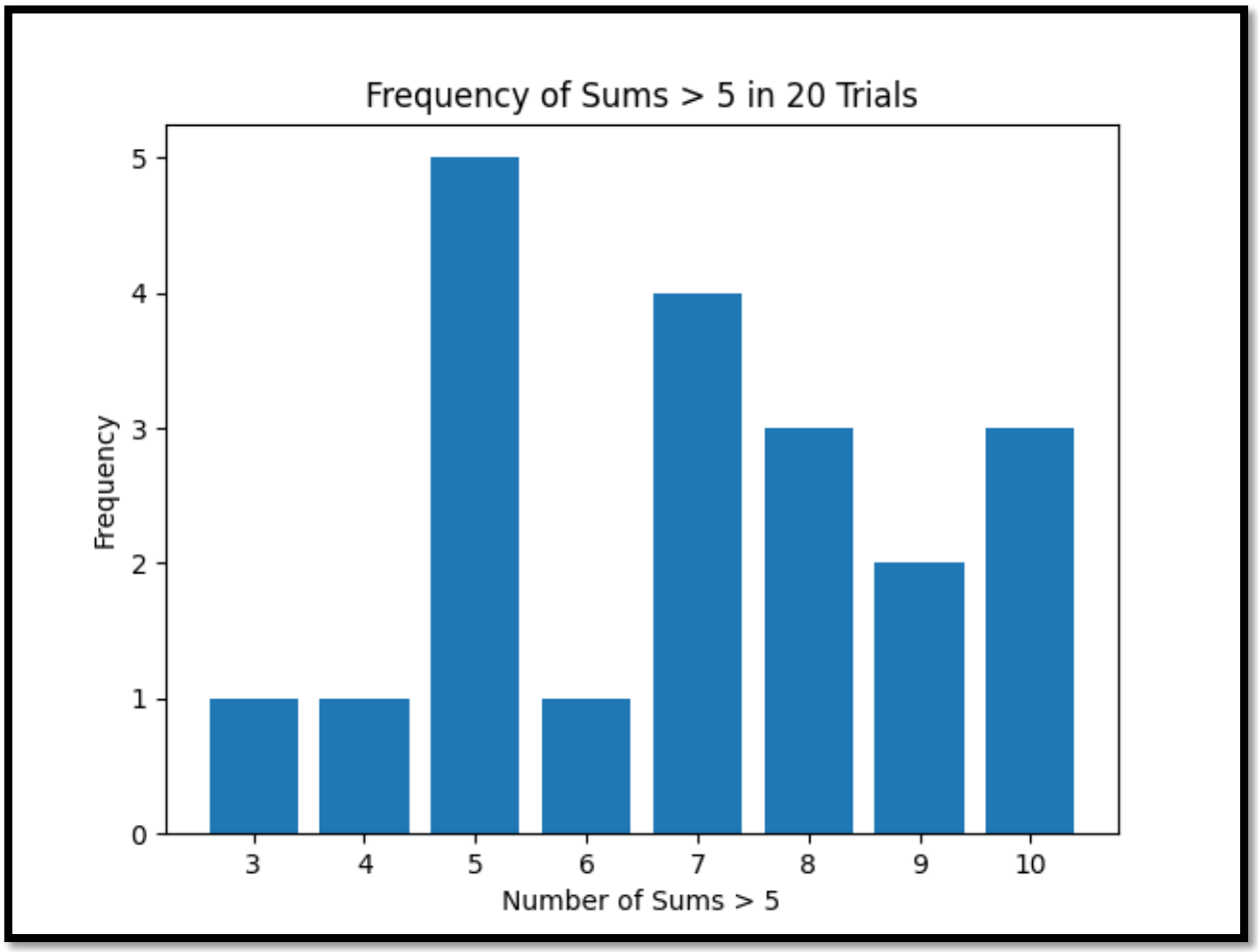


Number of Trials: **10**
Mean: **7.1**
Standard Deviation: **1.3**
Total Count: **71.0**

4. **Problem:**

20 trials of throwing two dice for 10 times.

Solution:

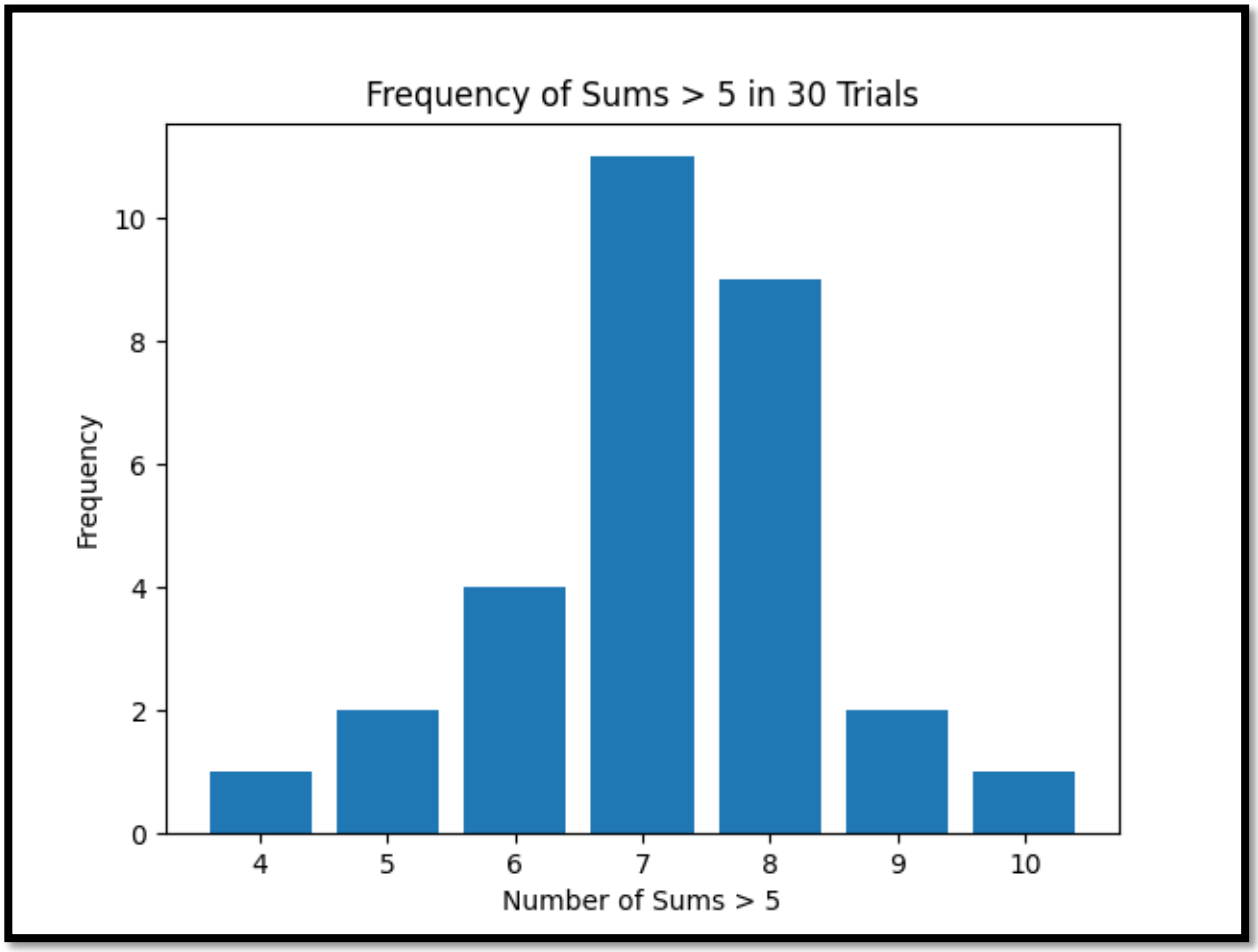


Number of Trials: **20**
Mean: **6.9**
Standard Deviation: **2.046948949045872**
Total Count: **138.0**

5. Problem:

30 trials of throwing two dice for 10 times.

Solution:

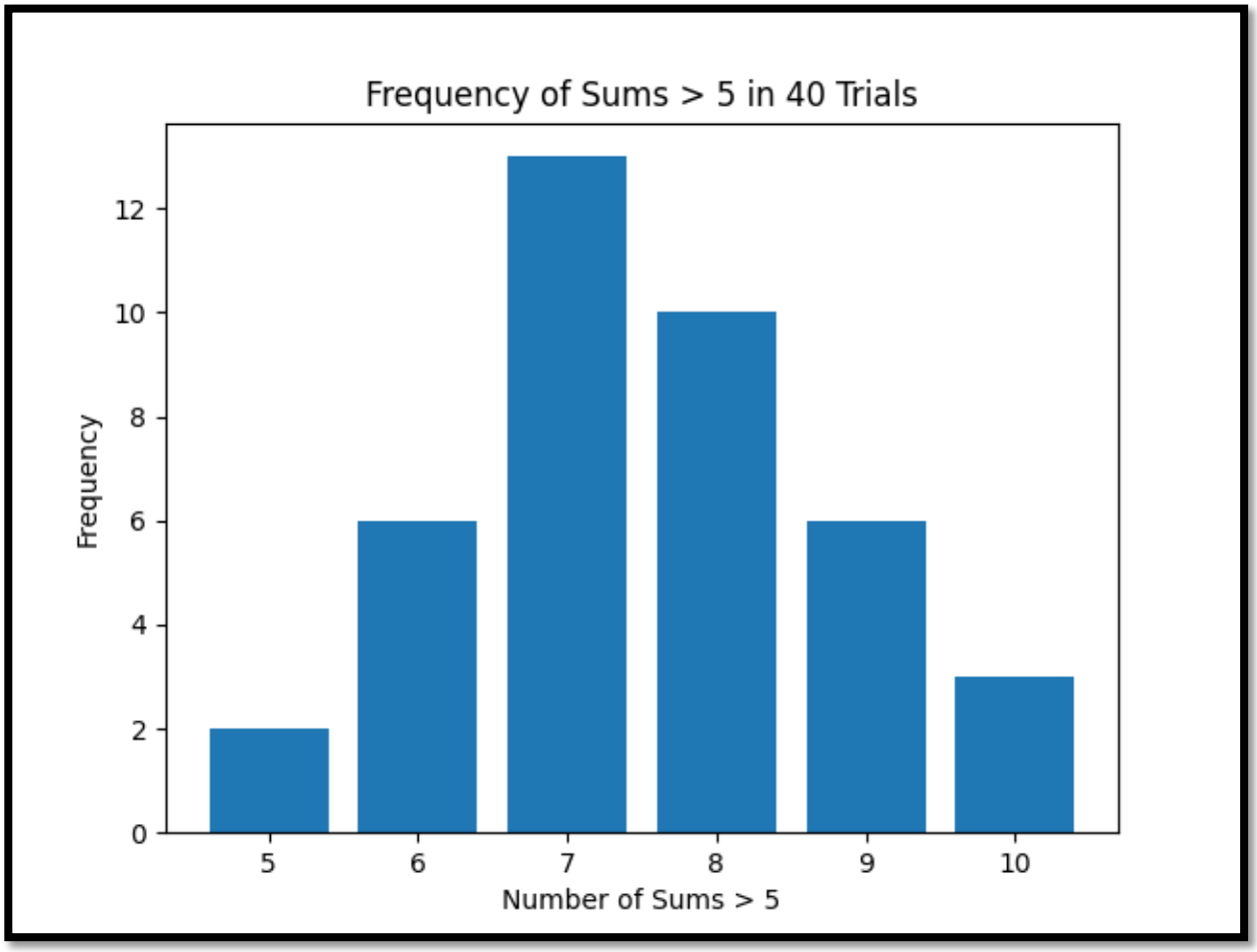


Number of Trials: **30**
Mean: **7.166666666666667**
Standard Deviation: **1.2405196043952262**
Total Count: **215.0**

6. **Problem:**

40 trials of throwing two dice for 10 times.

Solution:

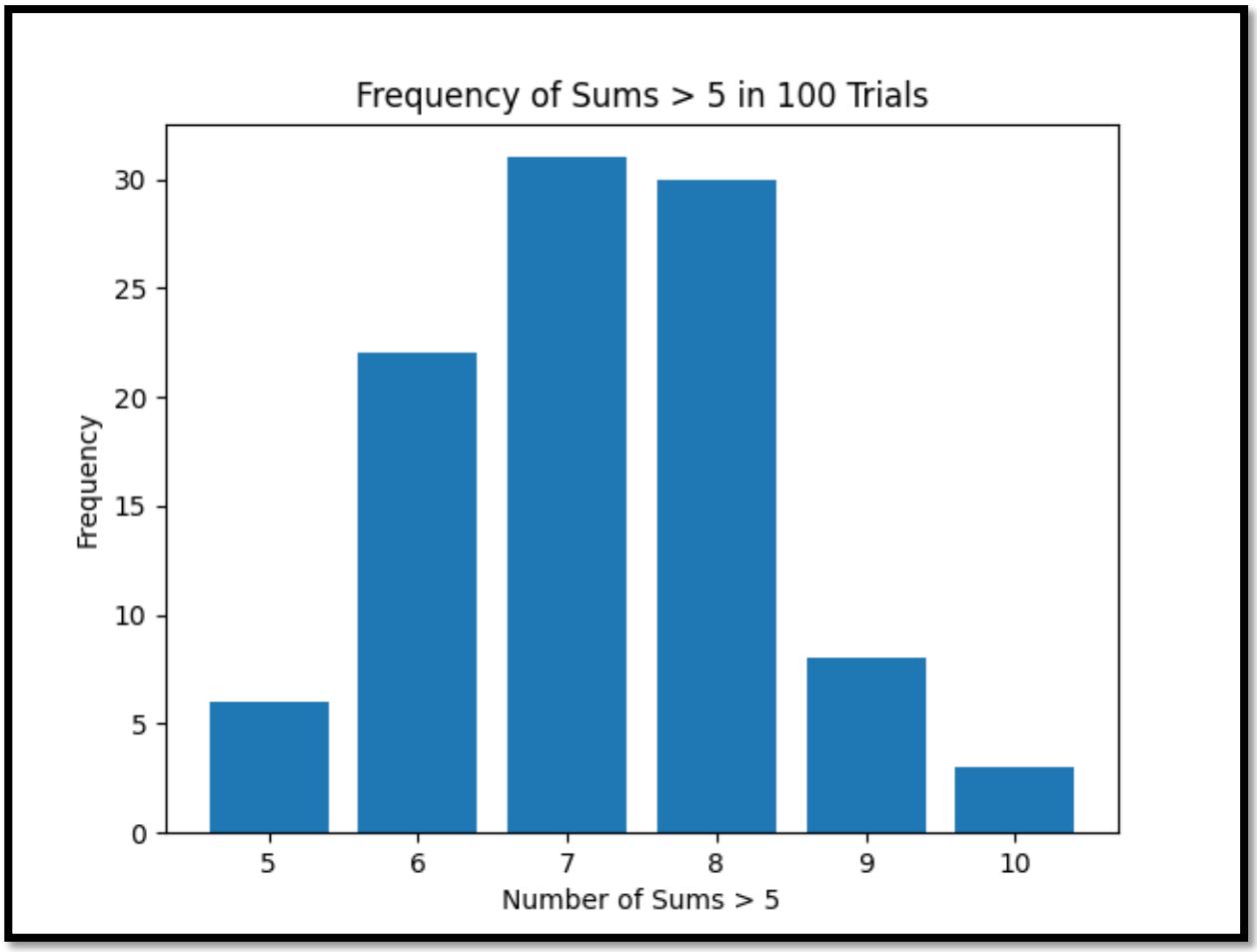


Number of Trials: **40**
Mean: **7.525**
Standard Deviation: **1.2646639869941738**
Total Count: **301.0**

7. Problem:

100 trials of throwing two dice for 10 times.

Solution:

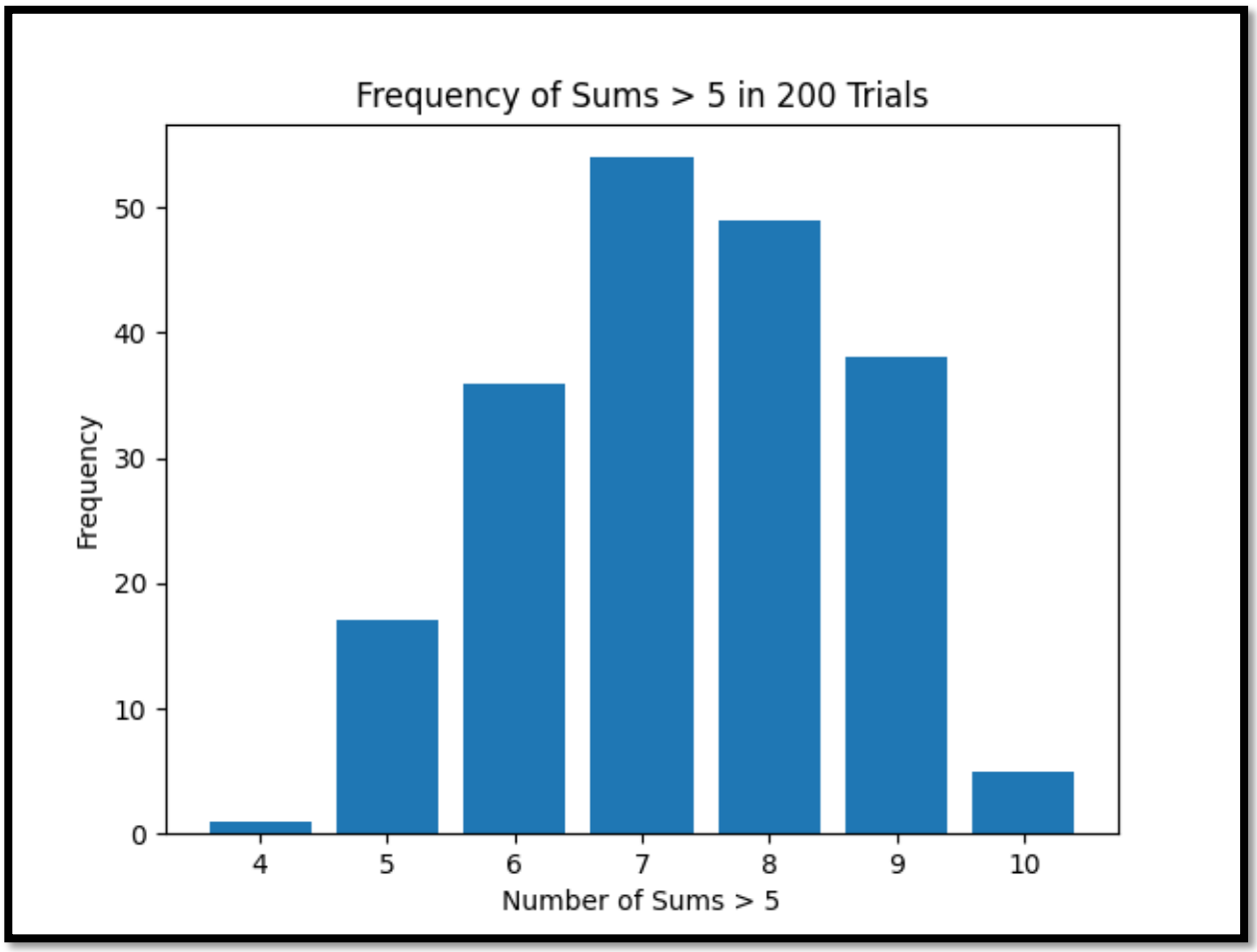


Number of Trials: **100**
Mean: **7.21**
Standard Deviation: **1.1427598172844544**
Total Count: **721.0**

8. Problem:

200 trials of throwing two dice for 10 times.

Solution:

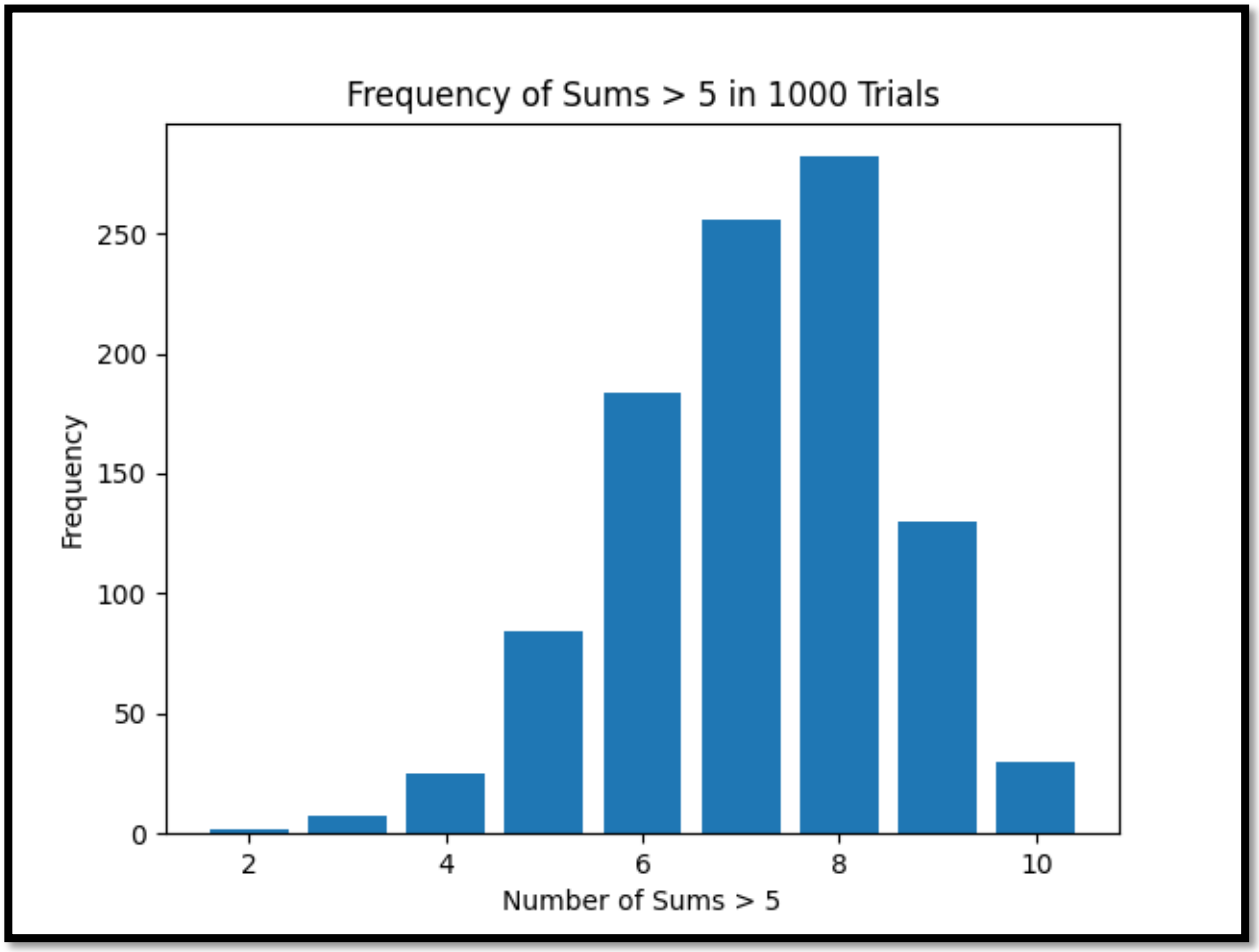


Number of Trials: **200**
Mean: **7.335**
Standard Deviation: **1.2972181774859617**
Total Count: **1467.0**

9. Problem:

1000 trials of throwing two dice for 10 times.

Solution:

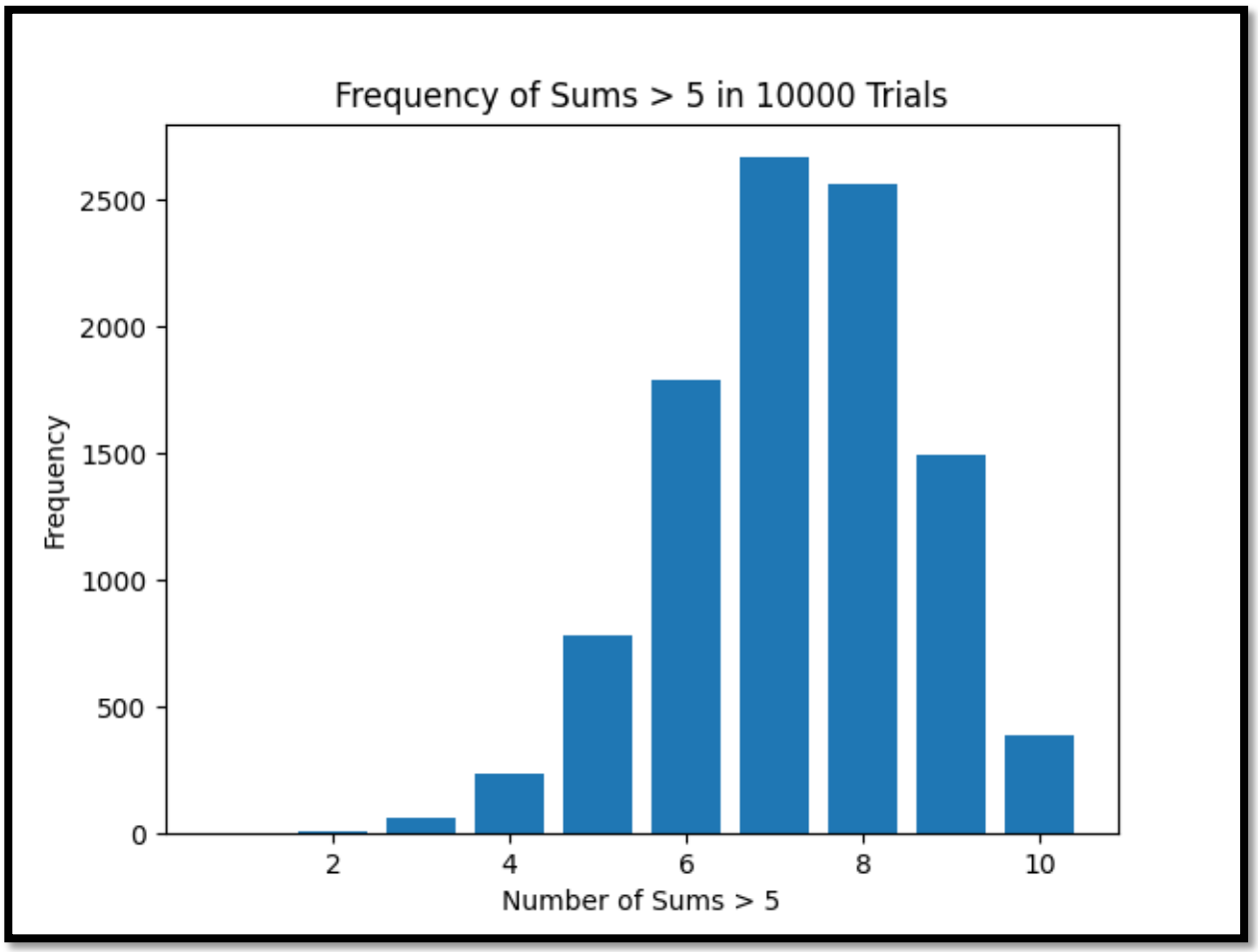


Number of Trials: 1000
Mean: 7.167
Standard Deviation: 1.3968217495443003
Total Count: 7167.0

10. Problem:

10000 trials of throwing two dice for 10 times.

Solution:

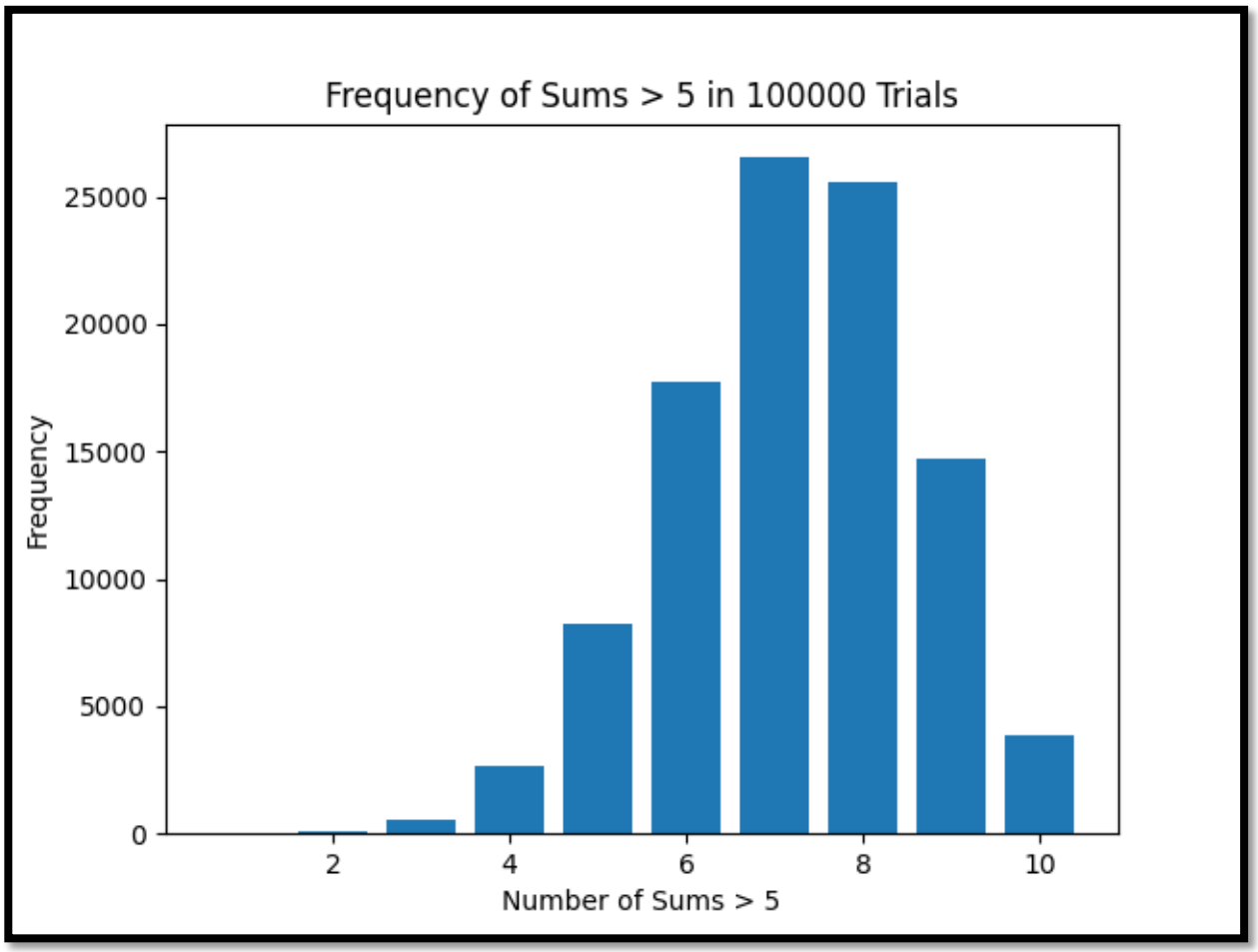


Number of Trials: 10000
Mean: 7.2352
Standard Deviation: 1.4080060227143916
Total Count: 72352.0

11. Problem:

100000 trials of throwing two dice for 10 times.

Solution:



Number of Trials: **100000**
Mean: **7.21785**
Standard Deviation: **1.415857117614627**
Total Count: **721785.0**

Findings

1. Mean and Standard Deviation:

- The mean values of the counts for sums greater than 5 across different numbers of trials range from approximately 6.9 to 7.525.
- This indicates that, on average, we can expect around 7 occurrences of sums greater than 5 per trial.
- The standard deviation values range from approximately 1.14 to 1.41. This indicates the variability or spread of the counts around the mean.
- A lower standard deviation suggests that the data points are closer to the mean, while a higher standard deviation indicates greater variability.

2. Total Count:

- The total count of sums greater than 5 increases as the number of trials increases.
- This is expected, as more trials provide more opportunities for the sums to occur.
- The total count follows a trend of approximately linear growth with the number of trials. This suggests a consistent increase in the number of occurrences with an increase in the number of trials.

3. Consistency of Results:

- As the number of trials increases, the mean and standard deviation values tend to stabilize.
- This indicates that the experiment outcomes become more consistent and reliable with a larger number of trials.

Conclusion:

In conclusion, the experiment demonstrates the frequency distribution of sums greater than 5 when throwing two dice for varying numbers of trials. The observations suggest a consistent trend in the occurrence of these sums, with reliable and predictable outcomes as the number of trials increases.