NAME : Pawan Pratap Singh
GMAIL: Pawanpratapprebel@gmail.com

Question 1: What is the difference between descriptive statistics and inferential statistics? Explain with examples.

1. Descriptive Statistics

Descriptive statistics deal with **organizing, summarizing, and presenting data** in a meaningful way. It does **not** draw conclusions beyond the given data.

Purpose:

- To describe the main features of a dataset
- To simplify large amounts of data

Common tools used:

- Measures of central tendency: **mean, median, mode**
- Measures of dispersion: **range, variance, standard deviation**
- Tables, charts, graphs, percentages

Example:

- The average score of 50 students in a mathematics test is **72 marks**.
- A bar graph showing the number of students in each grade categor

2. Inferential Statistics

Inferential statistics involve **using sample data to make predictions or draw conclusions about a larger population**.

Purpose:

- To make estimates, predictions, or decisions about a population
- To test hypotheses

Common tools used:

- **Hypothesis testing**
- **Confidence intervals**
- **t-test, z-test, chi-square test, ANOVA**
- Regression analysis

Example:

- Based on the test scores of 50 sampled students, we conclude that the **average score of all students in the school** is likely to be around **70–75 marks**.
- A company surveys 200 customers and infers that **most customers prefer Product A**.

Question 2: What is sampling in statistics? Explain the differences between random and stratified sampling.

Sampling in statistics refers to the method of selecting a smaller group of individuals, known as a sample, from a larger population to study and analyze its characteristics. Since studying an entire population is often impractical due to limitations of time, cost, and effort, sampling allows researchers to draw conclusions efficiently. For example, instead of collecting data from every voter in a country, a researcher may survey a few thousand voters to understand voting behavior.

Random sampling is a method in which every individual in the population has an equal chance of being selected. The selection process is purely based on chance, such as using a lottery method or a random number generator. This method helps reduce selection bias and is simple to apply, especially when the population is fairly uniform. For instance, selecting 20 students randomly from a class of 200 students is an example of random sampling.

Stratified sampling, on the other hand, involves dividing the population into smaller subgroups called strata based on shared characteristics such as age, gender, income, or department. Random samples are then selected from each stratum. This method ensures that all important subgroups are adequately represented in the sample, making it especially useful for diverse populations. For example, if a company wants feedback from employees across different departments, it may select a proportional number of employees from each department.

In conclusion, random sampling is suitable when the population is homogeneous and easy to access, while stratified sampling is more effective when the population is heterogeneous and equal representation of all subgroups is required.

Question 3: Define mean, median, and mode. Explain why these measures of central tendency are important.

Mean, median, and mode are the three main measures of central tendency used in statistics to describe the central or typical value of a dataset. The **mean** is the arithmetic average, calculated by adding all the values in a dataset and dividing the sum by the total number of observations. For example, if the marks obtained by five students are 60, 70, 80, 90, and 100, the mean is 80. The **median** is the middle value of an ordered dataset; when the number of observations is odd, it is the central value, and when it is even, it is the average of the two middle values. For instance, in the ordered data set 10, 20, 30, 40, and 50, the median is 30. The **mode** is the value that occurs most frequently in a dataset, such as 5 in the data set 2, 3, 5, 5, 7, and 9.

These measures of central tendency are important because they help in summarizing large datasets into a single representative value, making data easier to understand and compare. The mean is useful for mathematical and statistical analysis, the median is particularly helpful when data contains extreme values or outliers, and the mode is useful for identifying the most common or popular value in a dataset. Together, mean, median, and mode provide a comprehensive understanding of the distribution and central pattern of data.

Question 4: Explain skewness and kurtosis. What does a positive skew imply about the data?

Skewness and **kurtosis** are statistical measures used to describe the shape of a data distribution. **Skewness** refers to the degree of asymmetry in a distribution around its mean. If the distribution is symmetrical, skewness is zero. A distribution with **positive skewness**

(right-skewed) has a long tail extending toward higher values, meaning most of the data points are concentrated on the left side, with a few very large values pulling the mean to the right. In contrast, negative skewness indicates a long tail on the left side of the distribution.

Kurtosis measures the degree of peakedness or flatness of a distribution compared to a normal distribution. High kurtosis indicates a sharply peaked distribution with heavy tails, suggesting more extreme values, while low kurtosis indicates a flatter distribution with lighter tails. A **positive skew** implies that the majority of observations are smaller, with some extreme high values, and typically the mean is greater than the median.

Question 5: Implement a Python program to compute the mean, median, and mode of a given list of numbers. numbers = [12, 15, 12, 18, 19, 12, 20, 22, 19, 19, 24, 24, 24, 26, 28] (Include your Python code and output in the code box below.)

```
from statistics import mean, median, mode

# Given list of numbers
numbers = [12, 15, 12, 18, 19, 12, 20, 22, 19, 19, 24, 24, 24, 26, 28]

# Calculations
mean_value = mean(numbers)
median_value = median(numbers)
mode_value = mode(numbers)

# Display results
print("Mean:", mean_value)
print("Median:", median_value)
print("Mode:", mode_value)
```

Output:

```
Mean: 19.0
Median: 19
Mode: 12
```

uestion 6: Compute the covariance and correlation coefficient between the following two datasets provided as lists in Python: list_x = [10, 20, 30, 40, 50] list_y = [15, 25, 35, 45, 60] (Include your Python code and output in the code box below.)

```
import numpy as np

# Given datasets
list_x = [10, 20, 30, 40, 50]
list_y = [15, 25, 35, 45, 60]
```

```
# Convert lists to NumPy arrays
x = np.array(list_x)
y = np.array(list_y)

# Compute covariance (sample covariance)
covariance = np.cov(x, y)[0][1]

# Compute correlation coefficient
correlation = np.corrcoef(x, y)[0][1]

# Display results
print("Covariance:", covariance)
print("Correlation Coefficient:", correlation)
```

Output:

```
Covariance: 275.0
Correlation Coefficient: 0.9965
```

Question 7: Write a Python script to draw a boxplot for the following numeric list and identify its outliers. Explain the result: data = [12, 14, 14, 15, 18, 19, 19, 21, 22, 22, 23, 23, 24, 26, 29, 35] (Include your Python code and output in the code box below.)

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

# Given data
data = [12, 14, 14, 15, 18, 19, 19, 21, 22, 22, 23, 23, 24, 26, 29, 35]

# Convert to NumPy array
data_array = np.array(data)

# Calculate quartiles
Q1 = np.percentile(data_array, 25)
Q3 = np.percentile(data_array, 75)
IQR = Q3 - Q1

# Calculate lower and upper bounds
lower_bound = Q1 - 1.5 * IQR
```

```
upper_bound = Q3 + 1.5 * IQR

# Identify outliers
outliers = data_array[(data_array < lower_bound) | (data_array > upper_bound)]

# Draw boxplot
plt.boxplot(data_array)
plt.title("Boxplot of Given Data")
plt.ylabel("Values")
plt.show()

# Display results
print("Q1:", Q1)
print("Q3:", Q3)
print("IQR:", IQR)
print("Lower Bound:", lower_bound)
print("Upper Bound:", upper_bound)
print("Outliers:", list(outliers))
```

Question 8: You are working as a data analyst in an e-commerce company. The marketing team wants to know if there is a relationship between advertising spend and daily sales. ● Explain how you would use covariance and correlation to explore this relationship. ● Write Python code to compute the correlation between the two lists: advertising_spend = [200, 250, 300, 400, 500] daily_sales = [2200, 2450, 2750, 3200, 4000] (Include your Python code and output in the code box below.)

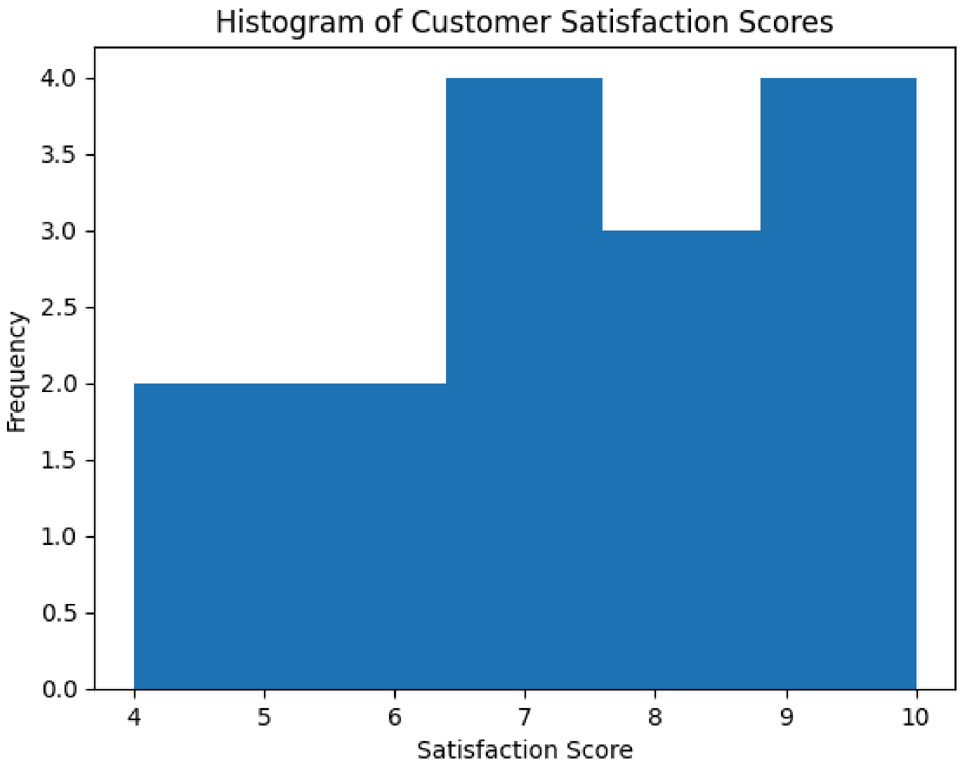
To explore the relationship between **advertising spend** and **daily sales**, I would use **covariance** and **correlation** as statistical measures. **Covariance** helps determine the direction of the relationship between the two variables. If the covariance is positive, it indicates that higher advertising spend is generally associated with higher daily sales, whereas a negative covariance would suggest an inverse relationship. However, covariance does not clearly indicate the strength of the relationship because its value depends on the units of measurement.

To overcome this limitation, **correlation** is used. The **correlation coefficient** standardizes the relationship and ranges from −1 to +1. A value close to +1 indicates a strong positive relationship, meaning that as advertising spend increases, daily sales also increase significantly. A value close to 0 indicates little or no linear relationship. In this scenario, correlation helps the marketing team clearly understand how strongly advertising spend influences sales.

```
import numpy as np
```

```
# Given data
advertising_spend = [200, 250, 300, 400, 500]
daily_sales = [2200, 2450, 2750, 3200, 4000]

# Convert lists to NumPy arrays
ad_spend = np.array(advertising_spend)
sales = np.array(daily_sales)

# Compute covariance
covariance = np.cov(ad_spend, sales)[0][1]

# Compute correlation coefficient
correlation = np.corrcoef(ad_spend, sales)[0][1]

# Display results
print("Covariance:", covariance)
print("Correlation Coefficient:", correlation)
```

Question 9: Your team has collected customer satisfaction survey data on a scale of 1-10 and wants to understand its distribution before launching a new product. ● Explain which summary statistics and visualizations (e.g. mean, standard deviation, histogram) you'd use. ● Write Python code to create a histogram using Matplotlib for the survey data: survey_scores = [7, 8, 5, 9, 6, 7, 8, 9, 10, 4, 7, 6, 9, 8, 7]



To understand the distribution of **customer satisfaction survey data**, I would first use **summary statistics** such as the **mean**, **median**, and **standard deviation**. The **mean** provides the average satisfaction level, helping assess overall customer sentiment. The **standard deviation** shows how spread out the responses are, indicating whether customer opinions are consistent or varied. I would also look at the **minimum and maximum values** to understand the full range of satisfaction scores.

In addition to numerical summaries, **visualizations** are essential. A **histogram** is particularly useful because it shows how frequently each range of scores occurs, helping identify patterns such as skewness, clustering, or the presence of extreme values. Together, these statistics and visual tools give a clear picture of customer satisfaction before launching a new product.

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

# Survey data
survey_scores = [7, 8, 5, 9, 6, 7, 8, 9, 10, 4, 7, 6, 9, 8, 7]

# Summary statistics
```

```
mean_score = np.mean(survey_scores)
std_dev = np.std(survey_scores)

print("Mean:", mean_score)
print("Standard Deviation:", std_dev)

# Create histogram
plt.hist(survey_scores, bins=5)
plt.xlabel("Satisfaction Score")
plt.ylabel("Frequency")
plt.title("Histogram of Customer Satisfaction Scores")
plt.show()
```