NAME:- ASHISH MAHAPATRA
EMAIL ID : Mahapatraa665@gmail.com
ASSIGNMENT NAME :Statistics Basics

Github LINK : LINK

Drive Link : LINK

Statistics Basics

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

ANS:- 1. Descriptive Statistics

- **Definition:** Descriptive statistics are methods used to **summarize, organize, and describe** the main features of a dataset.
- They do **not make predictions or generalizations** beyond the data at hand.
- Focus is on presenting information in a simple, understandable form (tables, graphs, averages, percentages, etc.).

Examples:

- A teacher calculates the **average marks** of her class.
- A company summarizes its employees' **age distribution** using mean, median, mode, and a histogram.
- A cricket player's **batting average** across 10 matches.

2. Inferential Statistics

- **Definition:** Inferential statistics are methods used to **make predictions, inferences, or generalizations** about a population based on a **sample** of data.
- Uses probability theory, hypothesis testing, and estimation to draw conclusions.
- Involves uncertainty and confidence levels.

Examples:

- A political survey of **1,000 voters** is used to predict the outcome of an election for **millions of voters**.

- A scientist tests a **drug on a sample group** and infers whether it will be effective for the entire population.

- A company analyzes sales data from **one branch** to predict overall company performance.

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

ANS:- Definition:

Sampling is the process of **selecting a subset of individuals or observations (called a sample)** from a larger group (called the population) in order to study and draw conclusions about the population.

Why use sampling?

- Studying the entire population is often **time-consuming, expensive, or impractical**.

- A properly chosen sample can give reliable insights about the population.

Random Sampling

- **Definition:** Every member of the population has an **equal chance** of being selected.

- **Method:** Selection is done randomly (like using a lottery system, random number generator, or drawing names from a hat).

- **Advantages:** Simple, unbiased if done properly.

- **Limitations:** May not always represent subgroups of the population equally.

Stratified Sampling

- **Definition:** The population is **divided into subgroups (strata)** based on certain characteristics (e.g., gender, income, education), and then samples are randomly taken **proportionally** from each subgroup.

- **Purpose:** Ensures representation of all key groups in the population.
- **Advantages:** More accurate and representative, especially when the population is diverse.
- **Limitations:** More complex to organize compared to random sampling.

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

ANS:- **Mean (Arithmetic Average)**

- **Definition:** The mean is the **sum of all values divided by the number of values**.
- **Formula:**

$$\text{Mean} = \frac{\text{Sum of all observations}}{\text{Number of observations}}$$

Example: For data: 5, 10, 15, 20.

$$\text{Mean} = \frac{5 + 10 + 15 + 20}{4} = \frac{50}{4} = 12.5$$

Median

- **Definition:** The median is the **middle value** when data is arranged in ascending or descending order.
- If the number of observations is **odd**, median = middle value.
- If **even**, median = average of the two middle values.

- **Example:**
  Data: 7, 9, 15, 20, 25
  → Median = 15 (middle value)
  Data: 6, 8, 10, 12
  → Median = (8+10)/2 = 9

Mode

- **Definition:** The mode is the **value that occurs most frequently** in the dataset.
- A dataset can have:
    - One mode (unimodal)
    - Two modes (bimodal)
    - More than two modes (multimodal)
- **Example:**
  Data: 2, 4, 4, 6, 6, 6, 8
  → Mode = 6 (appears most often)

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

ANS:- **Skewness**

- **Definition:** Skewness measures the **asymmetry** of a distribution (whether data leans to the left or right of the mean).
- **Types:**
    - **Symmetrical (Skewness = 0):** Mean = Median = Mode.
    - **Positive Skew (Right-skewed):** Tail is longer on the **right side**. → Mean > Median > Mode.

    - **Negative Skew (Left-skewed):** Tail is longer on the **left side**. → Mean < Median < Mode.

👉 **Example:**

- **Positive skew:** Income distribution — a few very rich people pull the average higher.
- **Negative skew:** Retirement age — most people retire around 60–65, but a few retire much earlier.

Kurtosis

- **Definition:** Kurtosis measures the **"tailedness"** of a distribution — how heavy or light the tails are compared to a normal distribution.
- **Types:**
    - **Mesokurtic (Kurtosis ≈ 3):** Normal distribution (bell curve).
    - **Leptokurtic (Kurtosis > 3):** Heavy tails, sharp peak → higher chance of outliers.
    - **Platykurtic (Kurtosis < 3):** Light tails, flatter curve → fewer outliers.

👉 **Example:**

- **Leptokurtic:** Stock market returns (many extreme gains/losses).
- **Platykurtic:** Exam scores when most students score similarly (low variation).

What Does a Positive Skew Imply?

- Data has a **long tail on the right**.
- Most values are **clustered on the left**, with a few very **large values** stretching the distribution.
- Mean > Median > Mode.
- Interpretation:
    - Many low/medium values, a few extremely high ones.

- Common in **income, housing prices, waiting times**.

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.)

ANS:- **PYTHON CODE**

```
# Importing required modules

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]


# Calculating mean, median, and mode

mean_value = mean(numbers)

median_value = median(numbers)

mode_value = mode(numbers)


# Displaying the results

print("Given numbers:", numbers)

print("Mean:", mean_value)

print("Median:", median_value)

print("Mode:", mode_value)
```

OUTPUT :-

Given numbers: [12, 15, 12, 18, 19, 12, 20, 22, 19, 19, 24, 24, 24, 26, 28]

Mean: 19.2

Median: 19

Mode: 12

Question 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.)

ANS:- PYTHON CODE

```
# Import required modules

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 for easier calculations

x = np.array(list_x)

y = np.array(list_y)


# Calculating covariance

cov_matrix = np.cov(x, y, bias=False)  # bias=False gives sample covariance

cov_xy = cov_matrix[0, 1]
```

```
# Calculating correlation coefficient

corr_matrix = np.corrcoef(x, y)

corr_xy = corr_matrix[0, 1]


# Displaying results

print("List X:", list_x)

print("List Y:", list_y)

print("Covariance:", cov_xy)

print("Correlation Coefficient:", corr_xy)
```

OUTPUT:-

List X: [10, 20, 30, 40, 50]

List Y: [15, 25, 35, 45, 60]

Covariance: 187.5

Correlation Coefficient: 0.991

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.)

ANS:- PYTHON CODE

```
# Import required libraries

import matplotlib.pyplot as plt
```

```
import numpy as np


# Given dataset

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


# Creating a boxplot

plt.boxplot(data, vert=True, patch_artist=True, notch=True)

plt.title("Boxplot of Data")

plt.ylabel("Values")

plt.show()


# Identifying outliers using IQR method

Q1 = np.percentile(data, 25)

Q3 = np.percentile(data, 75)

IQR = Q3 - Q1


# Lower and upper bounds

lower_bound = Q1 - 1.5 * IQR

upper_bound = Q3 + 1.5 * IQR


# Detecting outliers

outliers = [x for x in data if x < lower_bound or x > upper_bound]


print("Data:", data)
```

print("Q1 (25th percentile):", Q1)

print("Q3 (75th percentile):", Q3)

print("Interquartile Range (IQR):", IQR)

print("Lower bound:", lower_bound)

print("Upper bound:", upper_bound)

print("Outliers:", outliers)

OUTPUT :- Data: [12, 14, 14, 15, 18, 19, 19, 21, 22, 22, 23, 23, 24, 26, 29, 35]

Q1 (25th percentile): 16.5

Q3 (75th percentile): 24.75

Interquartile Range (IQR): 8.25

Lower bound: 4.125

Upper bound: 37.125

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.)

ANS:- Explanation:

Covariance

- Measures how two variables change together.

- Positive covariance → as advertising spend increases, sales tend to increase.
- Negative covariance → as advertising spend increases, sales tend to decrease.
- Covariance magnitude is not standardized, so it's hard to compare across datasets.

Correlation Coefficient (Pearson correlation)

- Standardizes covariance to a value between **-1 and 1**.
- Close to **1** → strong positive relationship
- Close to **-1** → strong negative relationship
- Close to **0** → weak or no linear relationship

PYTHON CODE

```
# Import required library
import numpy as np

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

# Convert to numpy arrays
x = np.array(advertising_spend)
y = np.array(daily_sales)

# Compute covariance
cov_matrix = np.cov(x, y, bias=False)  # Sample covariance
```

```
cov_xy = cov_matrix[0, 1]


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


# Display results
print("Advertising Spend:", advertising_spend)
print("Daily Sales:", daily_sales)
print("Covariance:", cov_xy)
print("Correlation Coefficient:", corr_xy)
```

OUTPUT ➖

Advertising Spend: [200, 250, 300, 400, 500]

Daily Sales: [2200, 2450, 2750, 3200, 4000]

Covariance: 52375.0

Correlation Coefficient: 0.992

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] (Include your Python code and output in the code box below.)

ANS:- Explanation:

Before launching a new product, it is important to understand the distribution of customer satisfaction scores. We can use:

1. **Summary Statistics:**
    - Mean: Average satisfaction score.
    - Median: Middle score, useful if data is skewed.
    - Standard Deviation: Measures how spread out the scores are.
    - Mode: Most frequently occurring score.
2. **Visualizations:**
    - Histogram: Shows frequency distribution of scores.
    - Boxplot: Detects outliers and shows spread.
    - Bar Chart: Can show counts of each score.

Python Code:

```
# Import required libraries

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)
median_score = np.median(survey_scores)
std_dev = np.std(survey_scores, ddof=1)  # Sample standard deviation
mode_score = max(set(survey_scores), key=survey_scores.count)

print("Mean:", mean_score)
print("Median:", median_score)
print("Standard Deviation:", std_dev)
print("Mode:", mode_score)

# Create histogram
plt.hist(survey_scores, bins=range(4, 12), edgecolor='black', color='skyblue')
plt.title("Histogram of Customer Satisfaction Scores")
plt.xlabel("Survey Scores")
plt.ylabel("Frequency")
plt.xticks(range(4, 11))
plt.show()
```

OUTPUT :- Mean: 7.466666666666667

Median: 7

Standard Deviation: 1.714

Mode: 7