**DATA MODELLING LAB WORKSHEET 2**

22BIT038

MUSAIB MAQBOOL WANI

PYTHON LIBRARIES

* **Python's Built-in Statistics Library**:
  + **Purpose**: Good for basic statistical calculations.
  + **Limitations**: Works well for smaller datasets and lacks advanced features.
* **NumPy**:
  + **Purpose**: Handles numerical computing efficiently with arrays.
  + **Features**: Great for performing basic statistical operations and working with large datasets.
* **SciPy**:
  + **Purpose**: Builds on NumPy for more advanced scientific computing.
  + **Features**: Includes scipy.stats for detailed statistical analysis.
* **pandas**:
  + **Purpose**: Designed for working with labeled data.
  + **Features**: Provides powerful tools for data manipulation with DataFrame (2D data) and Series (1D data).
* **Matplotlib**:
  + **Purpose**: Used for creating a wide range of static, animated, and interactive visualizations.
  + **Features**: Works well with NumPy, SciPy, and pandas to create charts, plots, and graphs.

**Types of Statistical Measures**

1. **Measures of Central Tendency**:
   * **Mean**: The average of all data points. Calculated by summing all values and dividing by the number of values.
     + Formula: Mean=∑XiN\text{Mean} = \frac{\sum X\_i}{N}Mean=N∑Xi​​
   * **Median**: The middle value when data points are ordered from smallest to largest. If there's an even number of data points, the median is the average of the two middle numbers.
   * **Mode**: The value that occurs most frequently in the dataset. There can be more than one mode or none at all if no value repeats.
2. **Measures of Variability (Dispersion)**:
   * **Range**: The difference between the maximum and minimum values in the dataset.
     + Formula: Range=Max−Min\text{Range} = \text{Max} - \text{Min}Range=Max−Min
   * **Variance**: Measures the average squared deviation of each data point from the mean. It gives an idea of the spread of the data.
     + Formula: Variance=∑(Xi−Mean)2N\text{Variance} = \frac{\sum (X\_i - \text{Mean})^2}{N}Variance=N∑(Xi​−Mean)2​
   * **Standard Deviation**: The square root of the variance. It provides a measure of the average distance of each data point from the mean.
     + Formula: Standard Deviation=Variance\text{Standard Deviation} = SQRT{Variance}}Standard Deviation=Variance​
   * **Interquartile Range (IQR)**: The range within which the central 50% of data points lie. Calculated as the difference between the 75th percentile (Q3) and the 25th percentile (Q1).
     + Formula: IQR=Q3−Q1\text{IQR} = Q3 - Q1IQR=Q3−Q1

Top of Form

Bottom of Form

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

# Sample Data

data = pd.DataFrame({

    'A': [10, 20, 30, 40, 50],

    'B': [15, 25, 35, 45, 55]

})

# Descriptive Statistics

mean\_A = data['A'].mean()

median\_A = data['A'].median()

mode\_A = data['A'].mode()[0]

std\_dev\_A = data['A'].std()

variance\_A = data['A'].var()

range\_A = data['A'].max() - data['A'].min()

mean\_B = data['B'].mean()

median\_B = data['B'].median()

mode\_B = data['B'].mode()[0]

std\_dev\_B = data['B'].std()

variance\_B = data['B'].var()

range\_B = data['B'].max() - data['B'].min()

print(f"Column A - Mean: {mean\_A}, Median: {median\_A}, Mode: {mode\_A}, Std Dev: {std\_dev\_A}, Variance: {variance\_A}, Range: {range\_A}")

print(f"Column B - Mean: {mean\_B}, Median: {median\_B}, Mode: {mode\_B}, Std Dev: {std\_dev\_B}, Variance: {variance\_B}, Range: {range\_B}")

# Summary of Descriptive Statistics

print(data.describe())

# Correlation Matrix

correlation\_matrix = data.corr()

print(correlation\_matrix)

# Visualizations

# Box Plot

sns.boxplot(data=data)

plt.title('Box Plot')

plt.show()

# Histograms

data.hist(bins=10, figsize=(10, 5))

plt.suptitle('Histograms')

plt.show()

# Pie Chart

sizes = data['A'].value\_counts()

plt.pie(sizes, labels=sizes.index, autopct='%1.1f%%')

plt.title('Pie Chart')

plt.show()

# Bar Chart

data['A'].value\_counts().plot(kind='bar')

plt.title('Bar Chart')

plt.xlabel('Value')

plt.ylabel('Frequency')

plt.show()

# X-Y Plot

plt.scatter(data['A'], data['B'])

plt.title('X-Y Plot')

plt.xlabel('A')

plt.ylabel('B')

plt.show()

# Heatmap

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm')

plt.title('Heatmap')

plt.show()

**OUTPUT**

Column A - Mean: 30.0, Median: 30.0, Mode: 10, Std Dev: 15.811388300841896, Variance: 250.0, Range: 40

Column B - Mean: 35.0, Median: 35.0, Mode: 15, Std Dev: 15.811388300841896, Variance: 250.0, Range: 40

A B

count 5.000000 5.000000

mean 30.000000 35.000000

std 15.811388 15.811388

min 10.000000 15.000000

25% 20.000000 25.000000

50% 30.000000 35.000000

75% 40.000000 45.000000

max 50.000000 55.000000

A B

A 1.0 1.0

B 1.0 1.0

