# Question1: Write the program to compute mean, median, mode, standard deviation, variance, correlation, covariance of given data table format 10X3 matrix where C1 and C2 has integer data type and C3 has character type.

Program

|  |
| --- |
| import pandas as pd  import numpy as np  import math  marks = [98,56, "P" ,78,43, "P", 56,23, "F", 67,87, "P", 99,54, "P", 78,27, "F", 77,45, "P", 56,57, "P",  89,34, "F" ,44,43, "P"]  data = np.array(marks).reshape(10,3)  results\_mapping = {      "P": 1,      "F": 0  }  df = pd.DataFrame(data, columns=["sub1", "sub2", "result"])  df['sub1'] = pd.to\_numeric(df['sub1'])  df['sub2'] = pd.to\_numeric(df['sub2'])  df['result'] = df['result'].map(results\_mapping)  df['result'] = pd.to\_numeric(df['result'])  print("Mean")  print(df[['sub1', 'sub2', 'result']].mean())  print("\nMedian:")  print(df[['sub1', 'sub2', 'result']].median())  print("\nMode:")  print(df[['sub1', 'sub2', 'result']].mode())  print("\nStandard Deviation:")  print(df[['sub1', 'sub2', 'result']].std())  print("\nVariance:")  print(df[['sub1', 'sub2', 'result']].var())  print("\nCorrelation between sub1 and sub2:")  print(df['sub1'].corr(df['sub2']))  print("\nCovariance Matrix:")  print(df[['sub1', 'sub2', 'result']].cov())  print("\nResult Distribution:")  print(df['result'].value\_counts())  print("\nOverall Mean")  print(df.mean().mean())  print("\nOverall Median")  print(df.median().median())  print("\nOverall Mode")  print(df.mode().mode())  print("\nOverall Standard Deviation")  print(df.std().std())  print("\nOverall Variance")  print(df.var().var()) |

Output:

|  |
| --- |
| Mean  sub1 74.2  sub2 46.9  result 0.7  dtype: float64  Median:  sub1 77.5  sub2 44.0  result 1.0  dtype: float64  Mode:  sub1 sub2 result  0 56 43.0 1.0  1 78 NaN NaN  Standard Deviation:  sub1 18.449932  sub2 18.290556  result 0.483046  dtype: float64  Variance:  sub1 340.400000  sub2 334.544444  result 0.233333  dtype: float64  Correlation between sub1 and sub2:  0.061307764484815494  Covariance Matrix:  sub1 sub2 result  sub1 340.400000 20.688889 -0.044444  sub2 20.688889 334.544444 6.300000  result -0.044444 6.300000 0.233333  Result Distribution:  result  1 7  0 3  Name: count, dtype: int64  Overall Mean  40.6  Overall Median  44.0  Overall Mode  sub1 sub2 result  0 56 43.0 1.0  1 78 NaN NaN  Overall Standard Deviation  10.327486205085725  Overall Variance  37918.59460905351 |

# Program2: Draw a BarPlot for the following data of company and its profit. Also draw BoxPlot also.

Program:

|  |
| --- |
| import matplotlib.pyplot as plt  import numpy as np  companyname = ["Google", "Amazon", "Microsoft", "IBM"]  profit = [3000,5000, 6000, 4500]  plt.figure(figsize=(12,6))  plt.subplot(1,2,1)  plt.bar(companyname, profit)  plt.title("Bar Chart")  plt.xlabel("Company Name")  plt.ylabel("Profit Value")  plt.subplot(1,2,2)  plt.boxplot([profit], label=["Companies"])  plt.title("Box Chart")  plt.ylabel("Profit Value")  plt.tight\_layout()  plt.show() |

Output:

|  |
| --- |
|  |

# Program 3: Draw a line plot and scatter plot for the following data

Semester -> [1,2,3,4,5,6,7,8]

Grades -> [A+, A, A, D, B+, B, A, A]

Program:

|  |
| --- |
| import pandas as pd  import matplotlib.pyplot as plt  data = {      "Semester" : [1,2,3,4,5,6,7,8],      "Grades" : ["A+", "D", "A", "B+", "A+", "A", "D", "B+"]  }  df = pd.DataFrame(data)  print(df)  grade\_mapping = {      'A+' : 10,      'A': 9,      'B+' : 8,      'B': 7,      'D' : 4  }  #Converting grades to numeric values  df['Numeric\_Grades'] = df['Grades'].map(grade\_mapping)  plt.figure(figsize=(12,5))  plt.subplot(1,2,1)  plt.scatter(df['Semester'], df['Numeric\_Grades'], color='blue', marker='o')  plt.title("Semester Grades - Scatter Plot")  plt.xlabel("Semester")  plt.ylabel("Grades")  plt.grid(True, linestyle="--", alpha= 0.7)  plt.yticks(list(grade\_mapping.values()), list(grade\_mapping.keys()))  plt.subplot(1,2,2)  plt.plot(df['Semester'], df['Numeric\_Grades'], 'r-o',color='blue', linewidth=2, markersize  = 8)  plt.title("Semester Grades - Line Plot")  plt.xlabel("Semester")  plt.ylabel("Grades")  plt.yticks(list(grade\_mapping.values()), list(grade\_mapping.keys()))  plt.grid(True, linestyle="-", alpha= 0.7)  plt.tight\_layout()  plt.show() |

Output:

|  |
| --- |
|  |

# Program4: Clean the dataset with replacing all the column values which contains ? to NAN, create dataset 5X3 and update our csv file

Program

|  |
| --- |
| import numpy as np  import pandas as pd  pd.set\_option('future.no\_silent\_downcasting', True)  data = {      'Age': [25, '?', 30, 35, '?', 28],      'Income': ['?', 45000, 55000, '?', 65000, 48000],      'Education': ['Bachelors', '?', 'Masters', 'PhD', 'Bachelors', '?']  }  df = pd.DataFrame(data)  df = df.replace('?', np.nan)  print("\nDataFrame after replacing '?' with NaN:")  print(df)  print("\nData Info after cleaning:")  print(df.info())  print("\nNumber of NaN values in each column:")  print(df.isna().sum())  #Save the cleaned dataset  df.to\_csv("cleaned\_dataset.csv", index=False) |

Output:

|  |
| --- |
|  |