Experiment 1: Import and Export Data files

Program:

|  |
| --- |
| import pandas as pd  import os  # ImportDataFiles  base\_dir = os.path.dirname(os.path.abspath(\_\_file\_\_))  file\_path = os.path.join(base\_dir, '..', 'dataset', 'cancer.csv')  df = pd.read\_csv(file\_path)  print(df.head())  # ExportDataFiles  with open("output/output.txt", "w") as f:      f.write("Hello World\n")      f.write(df.head().to\_string()) |

Output:

|  |
| --- |
|  |

Experiment2: Measure of Central Tendency

Program

|  |
| --- |
| import numpy as np  import pandas as pd  ages = [1,2,2,3]  print("Mean: " + str(np.mean(ages)))  # Mean  print("Median: " + str(np.median(ages)))  # Median  print("StandardDeviation: " + str(np.std(ages)))  # Standard Deviation  print("Variance: " + str(np.var(ages)))  # Variance  df  = pd.DataFrame(ages, columns=['ages'])  print("Mode: " + str(df.ages.mode()))  # Mode  q1 = np.percentile(ages, 25)  q3 = np.percentile(ages, 75)  iqr = q3 - q1  print(f"Interquartile Range (IQR): {iqr}")  print("Range: " + str(np.max(ages) - np.min(ages)))  # Range |

Output:

|  |
| --- |
|  |

Without using BuiltIn functions

Program:

|  |
| --- |
| def find\_mean(data):      n = len(data)      total = sum(data)      mean = total / n      return mean  def find\_median(data):      sorted\_data = sorted(data)      n = len(sorted\_data)      mid = n // 2        if n % 2 == 0:          median = (sorted\_data[mid - 1] + sorted\_data[mid]) / 2      else:          median = sorted\_data[mid]      return median  def find\_mode(data):      frequency = {}      for value in data:          frequency[value] = frequency.get(value, 0) + 1      max\_freq = max(frequency.values())      modes = [key for key, val in frequency.items() if val == max\_freq]        if len(modes) == len(frequency):          return "No mode"      elif len(modes) == 1:          return modes[0]      else:          return modes  def find\_std\_deviation(data):      mean = find\_mean(data)      squared\_diff = [(x - mean) \*\* 2 for x in data]      variance = sum(squared\_diff) / len(data)      std\_dev = variance \*\* 0.5      return std\_dev  data = [69, 74, 68, 70, 72, 67, 66, 70, 76, 68, 72, 79, 74, 67, 66, 71, 74]  print("Data:", data)  print("Mean:", find\_mean(data))  print("Median:", find\_median(data))  print("Mode:", find\_mode(data))  print("Standard Deviation:", round(find\_std\_deviation(data), 2)) |

Output

|  |
| --- |
|  |