Practical No. 1

Title: Write a Program to find the mean, median, standard deviation and mode using user defined functions.

Objective:

The objective of this practical is to implement and understand the computation of fundamental statistical measures, namely **Mean, Median, Mode, and Standard Deviation**, using Python. These measures provide insights into the distribution and characteristics of a given dataset.

Introduction:

In data analysis and statistics, various measures are used to summarize a dataset. The four key statistical measures covered in this practical are:

1. **Mean** – The average value of the dataset.
2. **Median** – The middle value when the data is sorted.
3. **Mode** – The most frequently occurring value(s).
4. **Standard Deviation** – A measure of how much the data deviates from the mean.

Understanding these statistical measures is crucial in data science, machine learning, and various analytical fields. Python provides an efficient way to compute these metrics using user-defined functions.

Requirements

Before starting, ensure that you have the following:

- **Python 3.x installed** on your system.
- **Basic knowledge of programming concepts** such as functions, loops, lists, and conditionals.
- **A text editor or IDE** like VS Code, PyCharm, or Jupiter Notebook.

Procedure:

Step 1: Take User Input

- The user is prompted to enter a set of numbers separated by spaces.
- The input is then converted into a list of integers for further processing.

Step 2: Sorting the Numbers

- Sorting the numbers is necessary to calculate the **median** correctly.
- Python's built-in sort () function is used to arrange the numbers in ascending order.

Step 3: Calculate Mean

- The **mean (average)** is calculated by summing all the numbers and dividing by the total count.
- Formula: Mean=∑XN\text {Mean} = \frac {\sum X} {N}Mean=N∑X where **ΣX** is the sum of all values, and **N** is the total number of values.

Step 4: Calculate Median

- If the total number of elements is **odd**, the middle element is taken as the median.
- If the total number of elements is **even**, the median is computed as the average of the two middle elements.
- Formula:
  Median={XN+12,if N is oddXN2+XN2+12,if N is even\text{Median} = \begin{cases} X_{\frac{N+1}{2}}, & \text{if } N \text{ is odd} \\ \frac{X_{\frac{N}{2}} + X_{\frac{N}{2}+1}}{2}, & \text{if } N \text{ is even} \end{cases}Median={X2N+1,2X2N+X2N+1,
  if N is oddif N is even

Step 5: Calculate Mode

- The mode is the most frequently occurring number in the dataset.
- If multiple numbers have the same highest frequency, they are all considered modes.

Step 6: Calculate Standard Deviation

- The standard deviation measures how much the values deviate from the mean.
- Formula: σ=∑(X−X¯)2N\sigma = \sqrt {\frac {\sum (X - \bar{X}) ^2} {N}}σ=N∑(X−X¯)2 where **X** is each data point, $\bar{X}$ is the mean, and **N** is the total number of elements.

Step 7: Display the Results

- The calculated Mean, Median, Mode, and Standard Deviation are displayed to the user.

Result:

The program successfully computes the **Mean, Median, Mode, and Standard Deviation** of a given dataset using user-defined functions. The output accurately represents the dataset's central tendency and dispersion.

Code

```
def calculate_mean(numbers):
    return sum(numbers) / len(numbers)

def calculate_median(numbers):
    numbers.sort()
    n = len(numbers)
    middle = n // 2
    return (numbers[middle - 1] + numbers[middle]) / 2 if n % 2 == 0 else numbers[middle]

def calculate_standard_deviation(numbers):
    mean = calculate_mean(numbers)
    variance = sum((x - mean) ** 2 for x in numbers) / len(numbers)
    return variance ** 0.5

def calculate_mode(numbers):
    from collections import Counter
    frequency = Counter(numbers)
    max_count = max(frequency.values())
    return [key for key, value in frequency.items() if value == max_count]

# Input and Execution
numbers = list(map(int, input("Enter numbers separated by spaces: ").split()))

print(f"Mean: {calculate_mean(numbers)}")
print(f"Median: {calculate_median(numbers)}")
print(f"Standard Deviation: {calculate_standard_deviation(numbers)}")
print(f"Mode: {calculate_mode(numbers)}")
```