**PRACTICAL NO- 1**

**AIM-** Compute measures of dispersion to calculate range, variance, standard deviation, coefficient of variation.

**BRIEFING OF THE AIM**

Measures of dispersion are summary statistics that describe the amount of variability or spread in a dataset. While measures of central tendency (like the mean) locate the center of a dataset, measures of dispersion quantify how spread-out the data points are from that center and from each other. A low dispersion indicates that the data points tend to be clustered closely together, while a high dispersion signifies that they are spread over a wider range. The four main measures are:

**1. Range:** The difference between the highest and lowest values in a dataset. It is the most straightforward measure of spread. However, because it only uses two data points, it can be a misleading representation of the overall distribution if there are one or more extreme outliers.

**2. Variance (σ²):** The average of the squared differences of each data point from the dataset's mean. A larger variance indicates greater spread. The primary drawback of variance is that its units are the square of the units of the original data (e.g., if the data is in meters, the variance is in meters squared), which makes it difficult to interpret in a real-world context.

**3. Standard Deviation (σ):** The square root of the variance (σ = √Variance). It is the most common and powerful measure of spread because it is expressed in the same units as the original data. A low standard deviation indicates that values are close to the mean, while a high standard deviation indicates that values are spread out over a wider range. For many datasets (those with a normal distribution), approximately 68% of values lie within one standard deviation of the mean.

**4. Coefficient of Variation (CV):** The ratio of the standard deviation to the mean, often expressed as a percentage (CV = (σ / μ) \* 100). It is a relative, unitless measure of dispersion. This makes it ideal for comparing the level of variability between two or more datasets, even if they have different means or are measured in different units.

**Libraries Used:**

**NumPy:** A fundamental package for numerical computation in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. Its array-based operations are significantly more efficient than standard Python lists for numerical tasks.

\* We will use numpy.ptp() (peak-to-peak) to find the range.

\* We will use numpy.var() to calculate the variance.

\* We will use numpy.std() to calculate the standard deviation.

\* The Coefficient of Variation is not a built-in NumPy function but is easily calculated using the outputs of numpy.std() and numpy.mean().

INDIVIDUAL SERIES

CODE:

**import** numpy **as** np

data = np.array([2, 4, 4, 4, 5, 5, 7, 9])

print(f"Individual Series Data: {data}")

print("-------------------------------------")

range\_value = np.ptp(data)

print(f"Range: {range\_value}")

variance\_value = np.var(data)

print(f"Variance: {variance\_value:.2f}")

std\_deviation\_value = np.std(data)

print(f"Standard Deviation: {std\_deviation\_value:.2f}")

mean\_value = np.mean(data)

**if** mean\_value != 0:

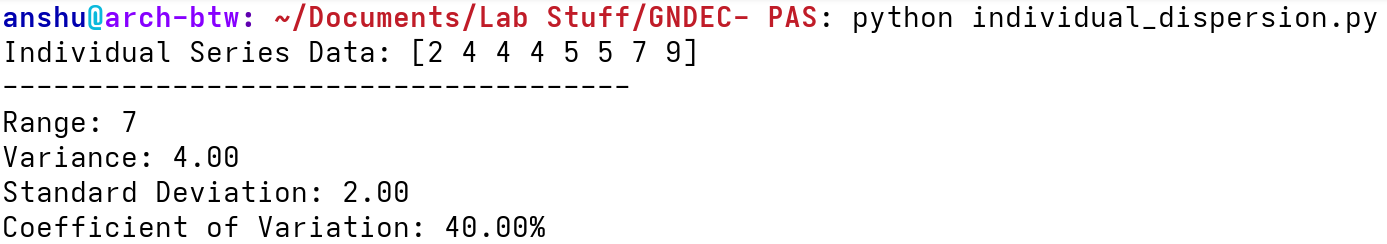
coeff\_variation = (std\_deviation\_value / mean\_value) \* 100

print(f"Coefficient of Variation: {coeff\_variation:.2f}%")

**else**:

print("Coefficient of Variation: Not applicable (mean is zero)")

OUTPUT:



DISCRETE SERIES

CODE:

**import** numpy **as** np

x = np.array([2, 3, 4, 5, 6])

f = np.array([2, 4, 5, 3, 1])

data = np.repeat(x, f)

print(f"Discrete Series Data (x): {x}")

print(f"Frequencies (f): {f}")

print("-------------------------------------")

range\_value = np.ptp(x)

print(f"Range: {range\_value}")

variance\_value = np.var(data)

print(f"Variance: {variance\_value:.2f}")

std\_deviation\_value = np.std(data)

print(f"Standard Deviation: {std\_deviation\_value:.2f}")

mean\_value = np.average(x, weights=f)

**if** mean\_value != 0:

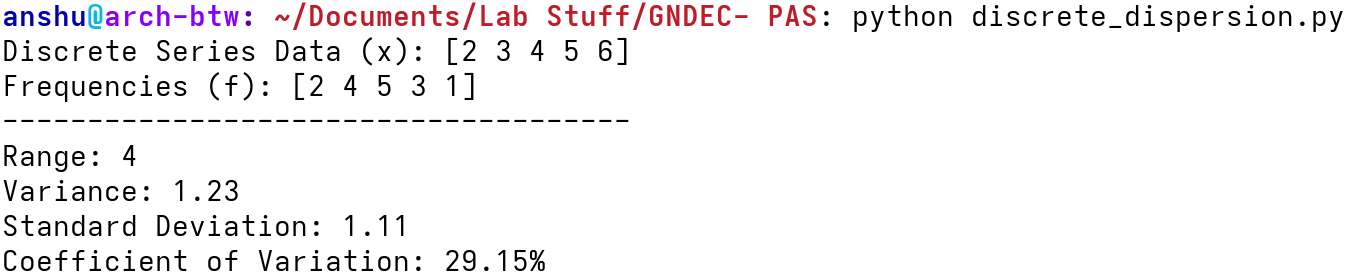
coeff\_variation = (std\_deviation\_value / mean\_value) \* 100

print(f"Coefficient of Variation: {coeff\_variation:.2f}%")

**else**:

print("Coefficient of Variation: Not applicable (mean is zero)")

OUTPUT:



CONTINUOUS SERIES

CODE:

**import** numpy **as** np

intervals = [(0, 10), (10, 20), (20, 30), (30, 40), (40, 50)]

frequency = np.array([5, 15, 25, 8, 7])

midpoints = np.array([(low + high) / 2 **for** low, high **in** intervals])

lower\_bounds = np.array([low **for** low, high **in** intervals])

upper\_bounds = np.array([high **for** low, high **in** intervals])

print(f"Intervals: {intervals}")

print(f"Frequencies: {frequency}")

print("-------------------------------------")

range\_value = upper\_bounds.max() - lower\_bounds.min()

print(f"Range: {range\_value}")

mean\_value = np.average(midpoints, weights=frequency)

variance\_value = np.average((midpoints - mean\_value)\*\*2, weights=frequency)

print(f"Variance: {variance\_value:.2f}")

std\_deviation\_value = np.sqrt(variance\_value)

print(f"Standard Deviation: {std\_deviation\_value:.2f}")

**if** mean\_value != 0:

coeff\_variation = (std\_deviation\_value / mean\_value) \* 100

print(f"Coefficient of Variation: {coeff\_variation:.2f}%")

**else**:

print("Coefficient of Variation: Not applicable (mean is zero)")

OUTPUT:

