Khulna University of Engineering & Technology, Khulna

Department of Biomedical Engineering

SESSIONAL REPORT

Course No: **BME 2152**

Experiment No : **03**

Name of the Experiment: **Experiment on the measures of central tendency using MATLAB.**

Remarks:

Date of performance : 1 January 2025

Date of submission : 8 January 2025

Name: **Md. Fuad Hasan Hamim**

Roll : **2215029**

Year : **2nd**

Term : **1st**

Objectives:

The main objectives of this sessional are as follows-

- To understand and analyze the concept of central tendency including different measures such as mean, mode, median, standard deviation, variance
- To apply MATLAB's built-in function as well as for loop to perform computation of these measures
- To learn how unusual or uneven data can change the results of mean, median, mode, and other measures.

Introduction:

Central tendency can be defined as a very basic concept within the statistics world; it provides a central or typical value for any distribution. It reflects the distribution of data in a summary form by one representative value. The three principal measures of central tendency are the mean, median, and mode. Along with these, variance and standard deviation explain the dispersal or scatter of values around the centricity values. These concepts bear immense importance regarding the distribution of data, for which computations may be performed on MATLAB through different built-in functions.

Mean/Average

The mean, or arithmetic mean, refers to an average of the data set that best describes the middle of the set. It is calculated by summing all values in a data set and then dividing the sum by the total number of data points. Mathematically, it can be represented as follows:

$$\text{Mean}(\mu)= \left\{\frac{1}{n}\right\} \sum_{i=1}^{n} x_i$$

Where x_i represents the individual data points and n is the total number of data points.

In MATLAB, the mean is computed by using the function mean. For example, if data is a vector containing the data set, then the command mean(data) returns the arithmetic mean. Mean is sensitive to outlier. A data point with value very high or very low affects the value of mean substantially.

Median

The median is the middle value of a sorted dataset. It is one measure of central tendency that is less sensitive to outliers than the mean. For a data set of n points ordered from smallest to largest, if n is odd, the median is the middle value in the ordered data set. If n is even, the median is the average of the two middle values. Mathematically the median can be calculated as:

Mathematically, for a group of data such as $x_1, x_2 \ldots x_n$ we can write as follows:

- If n is odd: Median=$x_{n+1}/_2$
- If n is even: Median=$\frac{x_{n/2}+x_{\frac{n}{2}+1}}{2}$

The median is less sensitive to outliers compared to the mean. This makes it a better measure of central tendency when dealing with skewed distributions.

In MATLAB, median is calculated via the function `median`. For instance, `median(data)` returns the median of the vector `data`. This measure is particularly helpful in the case of skewed distributions and distributions containing outliers.

Mode

The mode is the value that occurs most frequently in a data set. If no value occurs more than once, then the data set has no mode. If multiple values occur with the same highest frequency, then the data set is multimodal. Mathematically, the mode is the value x_m that maximizes the frequency count f(x_m).

The function `mode` can be used in MATLAB to compute the mode. As such, the statement `mode(data)` returns the most frequent value in the array `data`. This mode is quite useful in data that is either categorical or discrete since one is mostly interested in finding the most frequent element.

Standard Deviation

Standard deviation (σ) measures the extent of dispersion or variation in a dataset relative to its mean. A low standard deviation indicates that data points are closely clustered around the mean, while a high standard deviation signifies that the data points are more spread out.

The formula for standard deviation is:

$$\sigma = \sqrt{\frac{\sum_{j=1}^{N} (x_j - \bar{x})^2}{N}}$$

The standard deviation of a set of N numbers X_1, X_2, X_3,…., X_N is denoted by σ and x represent the deviations of each of the numbers X_j from the mean. In MATLAB, the standard deviation can be calculated using the std(data) function.

Variance

Variance (σ^2) is a measure of how data points differ from the mean. It reflects the degree to which the values in a dataset are spread out. Variance is calculated using the formula:

$$\sigma^2 = \frac{\sum_{j=1}^{N} (x_j - \bar{x})^2}{N}$$

This formula computes the average of the squared differences between each data point and the mean. Variance provides insight into the overall variability of the dataset. In MATLAB, the variance is computed using the var(data) function.

Both variance and standard deviation are closely related and serve to describe the dispersion of data. The standard deviation is simply the square root of the variance, and both are fundamental in statistical analysis for assessing consistency, spread, and reliability in datasets.

Geometric Mean

The geometric mean is a type of average that indicates the central tendency of a set of numbers by using the product of their values, as opposed to their sum (as in the arithmetic mean). It is particularly useful for data that is in multiplicative relationships or contains rates of change. The geometric mean is calculated as the nth root of the product of n numbers. For a set of numbers a_1, a_2, …. a_n, the geometric mean is expressed mathematically as:

$$G = \sqrt[n]{a_1 a_2 a_3 \dots a_n}$$

In MATLAB, the geometric mean can be calculated using the built-in geomean(data) function.

Harmonic Mean

The harmonic mean is another type of average, calculated as the reciprocal of the arithmetic mean of the reciprocals of a dataset. It is most commonly used in scenarios where rates or ratios are involved, such as speeds or densities. For a dataset consisting of n numbers $x_1, x_2, \ldots, x_n$, the harmonic mean is given by the formula:

$$H = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \frac{1}{x_3} + \cdots + \frac{1}{x_n}}$$

In MATLAB, the harmonic mean can be calculated using the harmmean(data) function.

RMS

The root mean square (RMS) or quadratic mean of a set of numbers X_1, X_2, X_3,…., X_N is sometimes denoted by $RMS = \sqrt{\frac{\sum_{j=1}^{\mathbb{N}} x_j^2}{N}}$

Task 1 || Finding sum and arithmetic mean of a set of numbers 1, 2, 3, 10 using for loop and built-in command.

Coding:

%Sum and Arithmetic Mean of a Set of Numbers (1, 2, 3, 10) Using for loop and built-in command %
clc;
clear all;
 	
a = [1, 2, 3, 10];
 	
% Using for loop
sum_loop = 0;
for i = 1:length(a)
    sum_loop = sum_loop + a(i);
end
mean_loop = sum_loop / length(a);
 	
% Using built-in commands
sum_builtin = sum(a);
mean_builtin = mean(a);
 	
fprintf("Sum using for loop is= %d\n", sum_loop);
fprintf("Mean using for loop is= %d\n", mean_loop);
fprintf("Sum using built-in command is= %d\n", sum_builtin);
fprintf("Sum using built-in command is= %d\n", mean_builtin);

Output:
Sum using for loop is= 16 Mean using for loop is= 4 Sum using built-in command is= 16 Sum using built-in command is= 4 >>
Comment:
Here, we see that Sum and Mean using for loop and built-in command both are same i.e. our operation is successful.

Task 2 || Finding sum and arithmetic mean of a set numbers (1, 2, 3, 10) with their respective frequencies (2, 1, 2, 1) using for loop and built-in command.

Coding:

% Finding sum and arithmetic mean of a set numbers (1, 2, 3, 10) with their respective frequencies (2, 1, 2, 1) using for loop and built-in command. %
clc;
clear all;
 	
a = [1, 2, 3, 10];
f = [2, 1, 2, 1];
 	
% Using for loop
sum1=0;
count1=0;
for i = 1:length(a)
    sum1 = sum1 + a(i)*f(i);
    count1 = count1 + f(i);
end
mean1 = sum1 / count1;
 	
% Using built-in commands
sum2 = sum(a.*f);
count2 = sum(f);
mean2 = sum2 / count2;
 	
fprintf("Sum using for loop is = %d\n", sum1);
fprintf("Mean using for loop is = %d\n", mean1);
fprintf("Sum using built-in command is = %d\n", sum2);
fprintf("Sum using built-in command is = %d\n", mean2);

Output:
Sum using for loop is = 20 Mean using for loop is = 3.333333e+00 Sum using built-in command is = 20 Sum using built-in command is = 3.333333e+00 >>
Comment:
Here, we also see that Sum and Mean using for loop and built-in command both are same and we could use arithmetic mean formula properly.

Task 3 || Show that, the arithmetic sum of the deviation of a set of numbers (1, 2, 3, 10) from their arithmetic mean is zero.

Coding:

```
% Arithmetic Sum of Deviations from Mean is Zero %
clc;
clear all;

a = [1, 2, 3, 10];


mean1= sum(a) / length(a);

% Initialize deviation sum
dev_sum = 0;

% Calculate sum of deviations using a for loop
for i = 1:length(a)
    dev_sum = dev_sum + (a(i) - mean1);
end

fprintf("Arithmatic sum of deviation is %d", dev_sum);
```

Output:

```
Arithmatic sum of deviation is 0
>>
```

Comment:

In this experiment we get our desired output i.e. 0. So, the arithmetic sum of the deviation of a set of numbers from their arithmetic mean is zero.

Task 4 || Finding geometric mean (G), harmonic mean (H) and arithmetic mean ($\overline{x}$) of a set of numbers (2, 4, 8).

Coding:

```
% Task 4: Geometric, Harmonic, and Arithmetic Mean (2, 4, 8) %
clc;
clear all;

a = [2, 4, 8];

product = 1;              % For geometric mean
reciprocal_sum = 0;       % For harmonic mean
sum_numbers = 0;          % For arithmetic mean


for i = 1:length(a)
    product = product * a(i);                       % Multiply each number for geometric mean
    reciprocal_sum = reciprocal_sum + 1 / a(i);     % Add reciprocals for harmonic mean
    sum_numbers = sum_numbers + a(i);               % Add numbers for arithmetic mean
end

% Geometric mean
g_mean = product^(1 / length(a));

% Harmonic mean
h_mean = length(a) / reciprocal_sum;

% Arithmetic mean
a_mean = sum_numbers / length(a);

fprintf("Geometric Mean = %d\n", g_mean);
fprintf("Harmonic Mean = %d\n", h_mean);
fprintf("Arithmetic Mean = %d\n", a_mean);
```

Output:

```
Geometric Mean = 4.000000e+00
Harmonic Mean = 3.428571e+00
Arithmetic Mean = 4.666667e+00
>>
```

Comment:

Here, different means give different insights into the data.

Task 5 || Finding sum, mean, standard deviation, variance, RMS, of 1, 2, 3, 4, 5, 6 using built-in command.

Coding:

```
% Sum, Mean, Standard Deviation, Variance, RMS of (1, 2, 3, 4, 5, 6) using built-in command
%
clc;
clear all;

a = [1, 2, 3, 4, 5, 6];

sum1 = sum(a);
mean1 = mean(a);
sd = std(a);
v = var(a);
rms = sqrt(mean(a.^2));

fprintf("Sum = %d\n", sum1);
fprintf("Mean = %d\n", mean1);
fprintf("Standard Deviation = %d\n", sd);
fprintf("Variance = %d\n", v);
fprintf("RMS = %d\n", rms);
```

Output:

```
Sum = 21
Mean = 3.500000e+00
Standard Deviation = 1.870829e+00
Variance = 3.500000e+00
RMS = 3.894440e+00
>>
```

Comment:

Built-in commands gives advantages of reducing complex codes.

Task 6 || Finding Standard deviation and variance of 1, 2, 3, 4, 5, 6 using for loop.

Coding:

<pre>% Task 6: Standard Deviation and Variance Using For Loop
clc;
clear all;
 	
a = [1, 2, 3, 4, 5, 6];
mean1 = sum(a) / length(a);
sum1 = 0;
 	
for i = 1:length(a)
    sum1 = sum1 + (a(i) - mean1)^2;
end
var = sum1 / (length(a) - 1);
sd = sqrt(var);
 	
fprintf("Standard Deviation = %d\n", sd);
fprintf("Variance = %d", var);</pre>

Output:
Standard Deviation = 1.870829e+00 Variance = 3.500000e+00 >>
Comment:
Using loops for standard deviation and variance is less efficient but provides insight into the calculation process.

Conclusion:

The following sessional dealt with central tendency and dispersion such as arithmetic, geometric, and harmonic mean, standard deviation, and variance. First, the theoretical aspects were gone through and then their MATLAB implementations. In practice, by making use of several built-in functions, along with for loops, hands-on practice was got for computing statistical measures and visualizing data. These exercises powerfully demonstrated the influence of outliers on different measures of central tendency, with implications for the choice of appropriate statistical tools in relation to data characteristics. These are skills that will prove indispensable in many areas of programming and data analysis.