Experiment - 3

Objective: Understand 1. Central Tendency Measures: Mean, Median, Mode 2. Measure of Dispersion: Variance, Standard Deviation

Central Tendency

A measure of **central tendency** (also referred to as measures of centre or central location) is a summary measure that attempts to describe a whole set of data with a single value that represents the middle or centre of its distribution.

There are three main measures of central tendency:

- mean
- median
- mode

Each of these measures describes a different indication of the typical or central value in the distribution.

- **Mean**

The mean is the sum of the value of each observation in a dataset divided by the number of observations. This is also known as the arithmetic average.

Looking at the retirement age distribution again:

54, 54, 54, 55, 56, 57, 57, 58, 58, 60, 60

The mean is calculated by adding together all the values (54+54+54+55+56+57+57+58+58+60+60 = 623) and dividing by the number of observations (11) which equals 56.6 years.

Advantage of the mean:
The mean can be used for both continuous and discrete numeric data.

Limitations of the mean:
The mean cannot be calculated for categorical data, as the values cannot be summed.

As the mean includes every value in the distribution the mean is influenced by outliers and skewed distributions.

- **Median**

The median is the middle value in distribution when the values are arranged in ascending or descending order.

The median divides the distribution in half (there are 50% of observations on either side of the median value). In a distribution with an odd number of observations, the median value is the middle value.

Looking at the retirement age distribution (which has 11 observations), the median is the middle value, which is 57 years:

54, 54, 54, 55, 56, 57, 57, 58, 58, 60, 60

When the distribution has an even number of observations, the median value is the mean of the two middle values. In the following distribution, the two middle values are 56 and 57, therefore the median equals 56.5 years:

52, 54, 54, 54, 55, 56, 57, 57, 58, 58, 60, 60

Advantage of the median:

The median is less affected by outliers and skewed data than the mean and is usually the preferred measure of central tendency when the distribution is not symmetrical.

Limitation of the median:

The median cannot be identified for categorical nominal data, as it cannot be logically ordered.

- **Mode**

The mode is the most commonly occurring value in a distribution. Consider this dataset showing the retirement age of 11 people, in whole years:

54, 54, 54, 55, 56, 57, 57, 58, 58, 60, 60

Following table shows a simple frequency distribution of the retirement age data.

Age	Frequency
54	3
55	1
56	1
57	2
58	2
60	2

The most commonly occurring value is 54, therefore the mode of this distribution is 54 years.

Advantage of the mode:

The mode has an advantage over the median and the mean as it can be found for both numerical and categorical (non-numerical) data.

Limitations of the mode:

The are some limitations to using the mode. In some distributions, the mode may not reflect the centre of the distribution very well. In some cases, particularly where the data are continuous, the distribution may have no mode at all (i.e. if all values are different).

Program 1: Write a python program to compute Central Tendency Measures: Mean

```
# Python program to print
# mean of elements
# list of elements to calculate mean
n_num = [1, 2, 3, 4, 5]
```

```
n = len(n_num)
get_sum = sum(n_num)
mean = get_sum / n
print("Mean / Average is: " + str(mean))
```

Output:

```
Mean / Average is: 3.0
```

Program 2: Write a python program to compute Central Tendency Measures: Median

```
# Python program to print
# median of elements
# list of elements to calculate median
n_num = [1, 2, 3, 4, 5]
n = len(n_num)
n_num.sort()
if n % 2 == 0:
 median1 = n_num[n//2]
 median2 = n_num[n//2 - 1]
 median = (median1 + median2)/2
else:
 median = n_num[n//2]
print("Median is: " + str(median))
```

Output:

```
Median is: 3
```

Program 3: Write a python program to compute Central Tendency Measures: Mode

```
# Python program to print
# mode of elements
from collections import Counter
# list of elements to calculate mode
n_num = [1, 2, 3, 4, 5, 5]
n = len(n_num)
data = Counter(n_num)
get_mode = dict(data)
mode = [k for k, v in get_mode.items() if v ==
max(list(data.values()))]
if len(mode) == n:
 get_mode = "No mode found"
else:
 get_mode = "Mode is / are: " + ', '.join(map(str, mode))

print(get_mode)
```

Output:

```
Mode is / are: 5
```

Measure of Dispersion

The **variance** is a measure of how far individual (numeric) values in a dataset are from the mean or average value. The variance is often used to quantify spread or dispersion. Spread is a characteristic of a sample or population that describes how much variability there is in it.

A high variance tells us that the values in our dataset are far from their mean. So, our data will have high levels of variability. On the other hand, a low variance tells us that the values are quite close to the mean. In this case, the data will have low levels of variability.

Standard deviation, on the other hand, is the square root of the variance that helps in measuring the expense of variation or dispersion in your dataset. It determines the deviation of each data point relative to the mean. A lower standard deviation indicates that the values are closer to the mean value. Again, a higher standard deviation indicates that the data are dispersed out in a wide range.

Program 3: Write a python program to compute Measure of Dispersion: Variance

```
def variance(val):
    numb = len(val)
    # m will have the mean value
    m = sum(val) / numb
    # Square deviations
    devi = [(x - m) ** 2 for x in val]
    # Variance
    variance = sum(devi) / numb
    return variance

print(variance([6, 6, 3, 9, 4, 3, 6, 9, 7, 8]))
```

Output:

```
4.49
```

Program 3: Write a python program to compute Measure of Dispersion: Standard Deviation

```
import math
# Finding the variance is essential before calculating the
standard deviation
def varinc(val, ddof=0):
    n = len(val)
    m = sum(val) / n
    return sum((x - m) ** 2 for x in val) / (n - ddof)
# finding the standard deviation
```

```python
def stddev(val):
    vari = varinc(val)
    stdev = math.sqrt(vari)
    return stdev

print(stddev([5, 9, 6, 2, 6, 3, 7, 4, 8, 6]))
```

Output:

```
2.0591260281974004
```