Descriptive Statistics

- Today i am going to discuss Only Central Tendency i.e Mean,Median And Mode

1. Mean,Median,Mode With Libraries

Mean :

- SO What is mean ? Mean is just average of total number.

  Given set of numbers : [n1, n2, n3, n5, n6]

Sum of data-set = (n1 + n2 + n3 + n4 + n5) Number of data produced = 5

Average or arithmetic mean = (n1 + n2 + n3 + n4 + n5) / 5

First we have to import statistics then we can find mean

In [1]:

```
# Implement In Python

# Python program to demonstrate mean()

# Importing the statistics module
import statistics

# list of positive integer numbers
data1 = [1, 3, 4, 5, 7, 9, 2]

x = statistics.mean(data1)

# Printing the mean
print("Mean is :", x)
```

```
Mean is : 4.428571428571429
```

Applications :

Mean/Arithmetic average is one of the very important function, while working with statistics and large values. So, with the function like mean(), trending and featured values can be extracted from the large data sets.

Median

- median() function in the statistics module can be used to calculate median value from an unsorted data-list. The biggest advantage of using median() function is that the data-list does not need to be sorted before being sent as parameter to the median() function.
- It is the middle value of data Here It have two condition

  If n id odd - then middle value is median But if n is even then median is avg of two middle value

Eg. n is odd

8,7,9,6,5,4,2 Median = 6 (As it is middle value)

Eg. n is even

1,3,5,5,5,7,8,9,10,10 Median = 5+7/2 = 6 (Avg of two mid value) (Only +ve number)

In [3]:

```
# Python program to demonstrate median()

# Importing the statistics module
import statistics

# list of positive integer numbers
data1 = [8,7,9,6,5,4,2] #odd
data2 = [1,3,5,5,5,7,8,9,10,10] #even

x = statistics.median(data1)
y = statistics.median(data2)

# Printing the mean
print("Median is :", x)
print("Median is :", y)
```

```
Median is : 6
Median is : 6.0
```

In [4]:

```
# working of median() on various
# range of data-sets

# importing the statistics module
from statistics import median

# Importing fractions module as fr
from fractions import Fraction as fr

# tuple of positive integer numbers
data1 = (2, 3, 4, 5, 7, 9, 11)

# tuple of floating point values
data2 = (2.4, 5.1, 6.7, 8.9)

# tuple of fractional numbers
data3 = (fr(1, 2), fr(44, 12), fr(10, 3), fr(2, 3))

# tuple of a set of negative integers
data4 = (-5, -1, -12, -19, -3)

# tuple of set of positive
# and negative integers
data5 = (-1, -2, -3, -4, 4, 3, 2, 1)

# Printing the median of above datsets
print("Median of data-set 1 is % s" % (median(data1)))
print("Median of data-set 2 is % s" % (median(data2)))
print("Median of data-set 3 is % s" % (median(data3)))
print("Median of data-set 4 is % s" % (median(data4)))
print("Median of data-set 5 is % s" % (median(data5)))
```

```
Median of data-set 1 is 5
Median of data-set 2 is 5.9
Median of data-set 3 is 2
Median of data-set 4 is -5
Median of data-set 5 is 0.0
```

Applications : For practical applications, different measures of dispersion and population tendency are compared on basis how well the corresponding population values can be estimated.

Mode

- The mode of a set of data values is the value that appears most often. It is the value at which the data is most likely to be sampled. A mode of a continuous probability distribution is often considered to be any value x at which its probability density function has a locally maximum value, so any peak is a mode.
- It is the frequency of occurance of data
- Means which data occured more number of time

Eg.

3,9,5,10,10,1,7,8,5,5 Mode = 5 (It Occure 3 times)

Note - A dataset have more than one Mode Value .

In [5]:

```
# Python code to demonstrate the
# use of mode() function

import statistics

# positive integers.
set1 =[1, 2, 3, 3, 4, 4, 4, 5, 5, 6]

# Printing out mode of given data-set
print("Mode of given data set is % s" % (statistics.mode(set1)))
```

```
Mode of given data set is 4
```

Applications : The mode() is a statistics function and mostly used in Financial Sectors to compare values/prices with past details, calculate/predict probable future prices from a price distribution set. mean() is not used separately but along with two other pillars of statistics mean and median creates a very powerful tool which can be used to reveal any aspect of your data.

2. Now We Implement Mean,Median,Mode Without Libraries

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)

sum = sum(n_num)
mean = sum / n

print("Mean / Average is: " + str(mean))
```

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:
    medan1 = 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))
```

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: " + ', '.join(map(str, mode))

print(get_mode)
```