AMRUTVAHINI COLLEGE OF ENGINEERING, SANGAMNER

BE(E&C)	Data Science & Visualization Lab	
Experiment No.: 02	Calculating Mean, Median, variance and plotting Correlation and Normal Distribution of a data using Python.	Page: /10

Aim: Calculating Mean, Median, variance and plotting Correlation and Normal Distribution of a data using Python.

Software Used: Python 3.12, Jupyter Notebook.

Learning Objective

1. Understand the basic terminologies in Statistics and Probability using python.

Learning Outcomes:

After performing the experiment students will be able to-

1. Calculate mean, median, mode and variance using python.
2. Plot the graph of correlation and normal distribution of a given data using different libraries of python.

Theory:

Statistics-

Statistics is the discipline that concerns the collection, organization, analysis, interpretation, and presentation of data. Statistics are used in virtually all scientific disciplines such as the physical and social sciences, as well as in business, the humanities, government, and manufacturing.

Two types of statistical methods are used in analyzing data: descriptive statistics and inferential statistics.

Descriptive Statistics-

Descriptive statistics mostly focus on the central tendency, variability, and distribution of sample data. Central tendency means the estimate of the characteristics, a typical element of a sample or population, and includes descriptive statistics such as mean, median, and mode.

Mean-

A mean is the average value of a set of data points. The arithmetic mean is the sum of the elements along the axis divided by the number of elements.

$$\text{Mean} = \frac{\text{Sum of all data values}}{\text{Number of data values}}$$

Symbolically,

$$\bar{x} = \frac{\sum x}{n}$$

where $\bar{x}$ (read as 'x bar') is the mean of the set of x values,
$\sum x$ is the sum of all the x values, and
n is the number of x values.

Example

```
import numpy as np
data=[168,170,150,160,182,140,175,191,152,150]
print(data)
mean=np.mean(data)
print(mean)
```

Output-

```
163.8
```

```
import numpy as np
a = np.array([[1, 2], [3, 4]])
print(a)
np.mean(a)
```

Output-

```
2.5
```

```
arr = np.random.randint(10,10000, size = 50)
print(arr)
mean = np.mean(arr)
mean
```

Output-

```
[5456 6178 2590 5183 4204 5389 5896 7354 3419 2014 7422 3560 1670
6817 9574 8268 7946 7093 5126 4160 1981 4001 1536   84 1571 7314
4207 9179 1715 2340 8850 7225 8009 2131 2087 4320 7174 1352 4760
6953 3692 7539 4554  639 6333 9350 2184 6264 3500  838]

4820.02
```

Median-

The median is the middle value in a set of data points when they are arranged in order (ascending or descending).

Even Number of Observations: When the number of data points is even, the median is interpolated by taking the average of the two middle values.

If the total number of observation is even, then the median formula is:

$$Median = \frac{(\frac{n}{2})^{th}\ term + (\frac{n}{2}+1)^{th}\ term}{2}$$

where n is the number of observations.

```
#median for even count of samples:
import numpy as np
data=[168,170,150,160,182,140,175,191,152,150]
data.sort()
print(data)
median=np.median(data)
print(median)
```

Output-

```
[140, 150, 150, 152, 160, 168, 170, 175, 182, 191]
164.0
```

Odd Number of Observations: When the number of data points is even, the median is interpolated by taking the average of the two middle values.

If the total number of observations given is odd, then the formula to calculate the median is:

$$Median = (\frac{n+1}{2})^{th}\ term$$

where n is the number of observations.

```
#median for odd count of samples:
import numpy as np
data =[168,170,150,160,182,140,175,191,152]
data.sort()
print(data)
median=np.median(data)
print(median)
```

Output-

```
[140, 150, 152, 160, 168, 170, 175, 182, 191]
168.0
```

Mode-

A mode is defined as the value that has a higher frequency in a given set of values. It is the value that appears the most number of times.

Mode Formula for Grouped Data:

$$Mode = l + \left(\frac{f_1 - f_0}{2f_1 - f_0 - f_2}\right) \times h$$

Where, l = lower limit of the modal class, h = size of the class interval, f_1 = frequency of the modal class, f_0 = frequency of the class preceding the modal class, f_2 = frequency of the class, succeeding the modal class modal class = class having highest frequency

```
import numpy as np
import statistics as stats
data = [168,170,150,160,182,140,175,191,152,150]
stats.mode(data)
```

Output-

```
150
```

Variability-

Variability refers to a set of statistics that show how much difference there is among the elements of a sample or population along the characteristics measured, and includes metrics such as range, variance, and standard deviation.

Range-

The Range is the difference between the lowest and highest values.

Variance-

Variance measures variability from the average or mean. It is calculated by taking the differences between each number in the data set and the mean, then squaring the differences to make them positive, and finally dividing the sum of the squares by the number of values in the data set. Informally, variance estimates how far a set of numbers (random) are spread out from their mean value.

Variance is calculated by using the following formula:

$$\sigma^2 = \frac{\sum_{i=1}^{n} (X_i - \mu)^2}{n}$$

Where,

X_i : i^{th} elements in the data set

μ : the population mean

N : Population size

```
import numpy as np
data = [168,170,150,160,182,140,175,191,152,150]
mean=np.mean(data)
print(mean)
variance= np.var(data)
print(variance)
```

Output-

```
163.8
235.35999999999999
```

Standard Deviation-

Standard Deviation is a measure which shows how much variation (such as spread, dispersion) from the mean exists. It is square root of the variance and denoted by Sigma (σ).

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}$$

σ = population standard deviation
N = the size of the population
x_i = each value from the population
μ = the population mean

```
import numpy as np
data=[168,170,150,160,182,140,175,191,152,150]
print(data)
standard_deviation=np.std(data)
print(standard_deviation)
```

Output-

```
[168, 170, 150, 160, 182, 140, 175, 191, 152, 150]
15.341447128612085
```

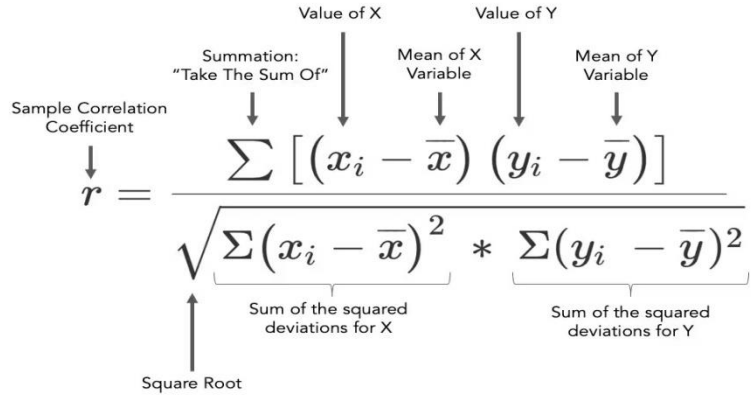
BE(E&C)	Data Science & Visualization Lab	
Experiment No.: 02	Calculating Mean, Median, variance and plotting Correlation and Normal Distribution of a data using Python.	Page: /10

Correlation -

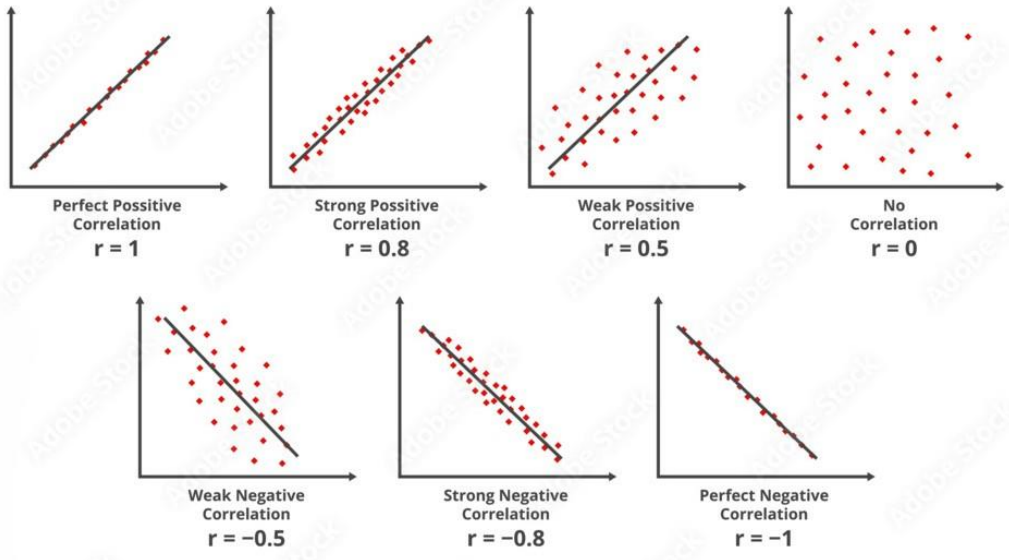
Correlation refers to a process for establishing the relationships between two variables. In statistics, Correlation studies and measures the direction and extent of relationship among variables, so the correlation measures co-variation, not causation.

A scatter diagram is a diagram that shows the values of two variables X and Y, along with the way in which these two variables relate to each other.

Pearson Correlation Coefficient Formula:



Types of Correlation:




```
import numpy as np
orbital_period = np.array([88, 225, 365, 460, 687, 1200, 2223,
3831, 5854, 8756, 10015, 20687, 30765, 40190]) #days
dist_from_sun = np.array([58, 108, 250, 328, 486, 978, 1287,
1890, 2060, 2396, 3400, 3845, 4500, 4800]) #million km
```

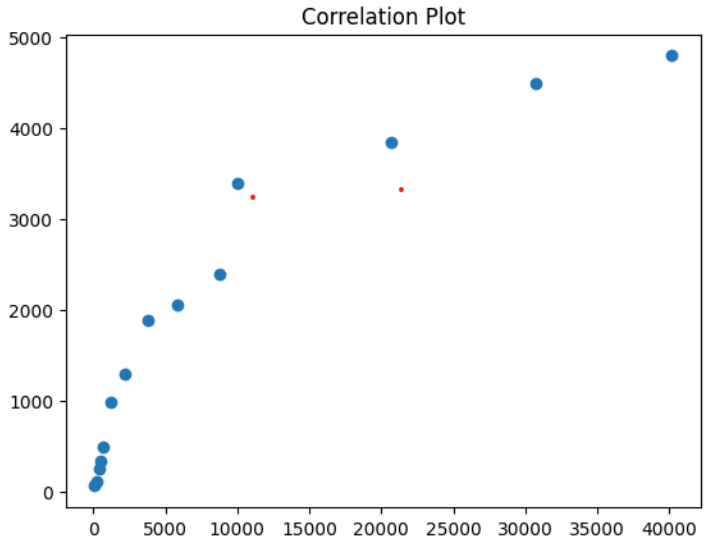
```
print(orbital_period, dist_from_sun)

#Show that a perfect monotonic relationship exists
correlation_coeff = np.corrcoef(orbital_period, dist_from_sun)
print(correlation_coeff )
```

Output-

```
[88   225   365   460   687  1200  2223  3831  5854  8756 10015
20687 30765 40190]
[58  108  250  328  486  978 1287 1890 2060 2396 3400 3845 4500
4800]
[[1.         0.91602461]
 [0.91602461 1.        ]]
```

```
import numpy as np
orbital_period = np.array([88, 225, 365, 460, 687, 1200, 2223,
3831, 5854, 8756, 10015, 20687, 30765, 40190])    #days
dist_from_sun = np.array([58, 108, 250, 328, 486, 978, 1287,
1890, 2060, 2396, 3400, 3845, 4500, 4800]) #million km

plt.scatter(orbital_period, dist_from_sun)
plt.title("Correlation Plot")
plt.xlabel = ("Orbital Period")
plt.ylabel = ("Distance from Sun")
plt.show()
```

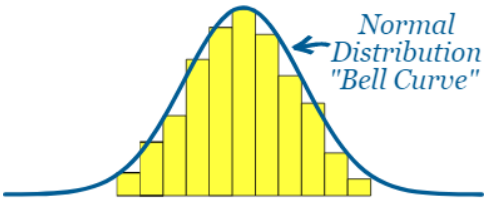
Output-



BE(E&C)	Data Science & Visualization Lab	
Experiment No.: 02	Calculating Mean, Median, variance and plotting Correlation and Normal Distribution of a data using Python.	Page: /10

Normal Distribution-

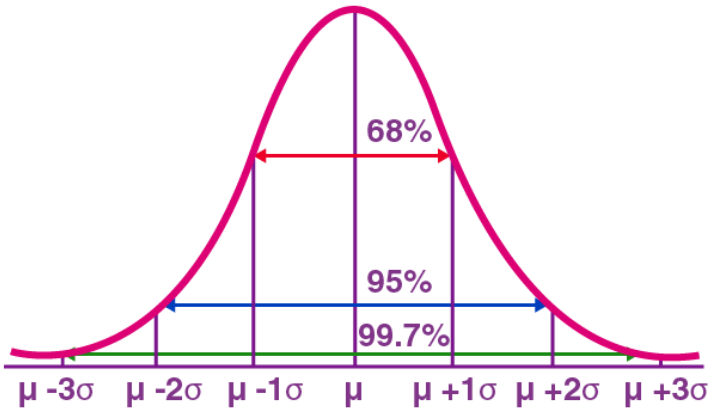
The Normal Distribution is defined by the probability density function for a continuous random variable in a system. It is also known as Gaussian distribution which is symmetric about its mean and has a bell-shaped curve.



Normal Distribution Formula:

$$f(x,\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

Where, x is the variable, μ is the mean, σ is the standard deviation




```
import numpy as np
def normal_dist(x, mean, sd):
    prob_density = (np.pi*sd) * np.exp(-0.5*((x-mean)/sd)**2)
    return prob_density
mean = 0
sd = 1
x = 1
result = normal_dist(x, mean, sd)
print(result)
```

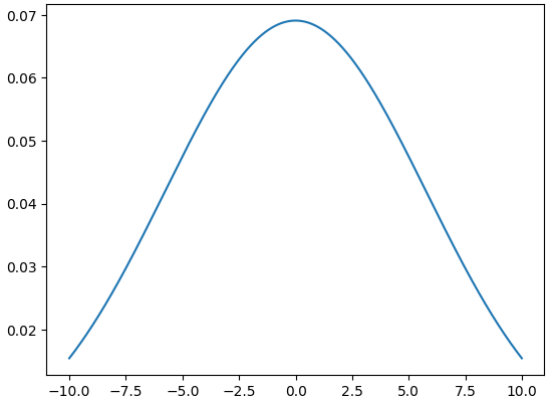
Output-

```
1.9054722647301798
```

BE(E&C)	Data Science & Visualization Lab	
Experiment No.: 02	Calculating Mean, Median, variance and plotting Correlation and Normal Distribution of a data using Python.	Page: /10

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm
import statistics

#Plot between -10 and 10 with .001 steps
x_axis = np.arange(-10, 10, 0.01)

#Calculating mean and standard deviation
mean = statistics.mean(x_axis)
sd = statistics.stdev(x_axis)

plt.plot(x_axis, norm.pdf(x_axis, mean, sd))
plt.show()
```

Output-

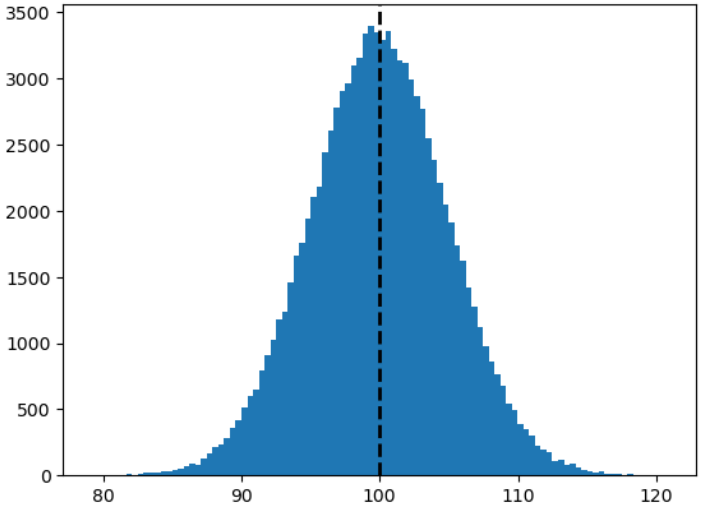



```
import numpy as np
#Mean of the distribution
Mean = 100
#satndard deviation of the distribution
Standard_deviation  = 5
 #size
size = 100000

#creating a normal distribution data
values = np.random.normal(Mean, Standard_deviation, size)

#plotting histograph
plt.hist(values, 100)

#plotting mean line
plt.axvline(values.mean(), color='k', linestyle='dashed',linewidth=2)
plt.show()
```

AMRUTVAHINI COLLEGE OF ENGINEERING, SANGAMNER

BE(E&C)	Data Science & Visualization Lab	
Experiment No.: 02	Calculating Mean, Median, variance and plotting Correlation and Normal Distribution of a data using Python.	Page: /10

Output-



Conclusion:

_____

_____

_____

_____

Questions:

1. Describe NumPy.
2. Describe the arrays in Python?
3. The runs scored by a batsman in 5 ODIs are 31,97,112, 63, and 12. Find the standard deviation of given data using numpy.
4. What is the difference between correlation and co-variance?
5. Explain the parameters needed to calculate the normal distribution.