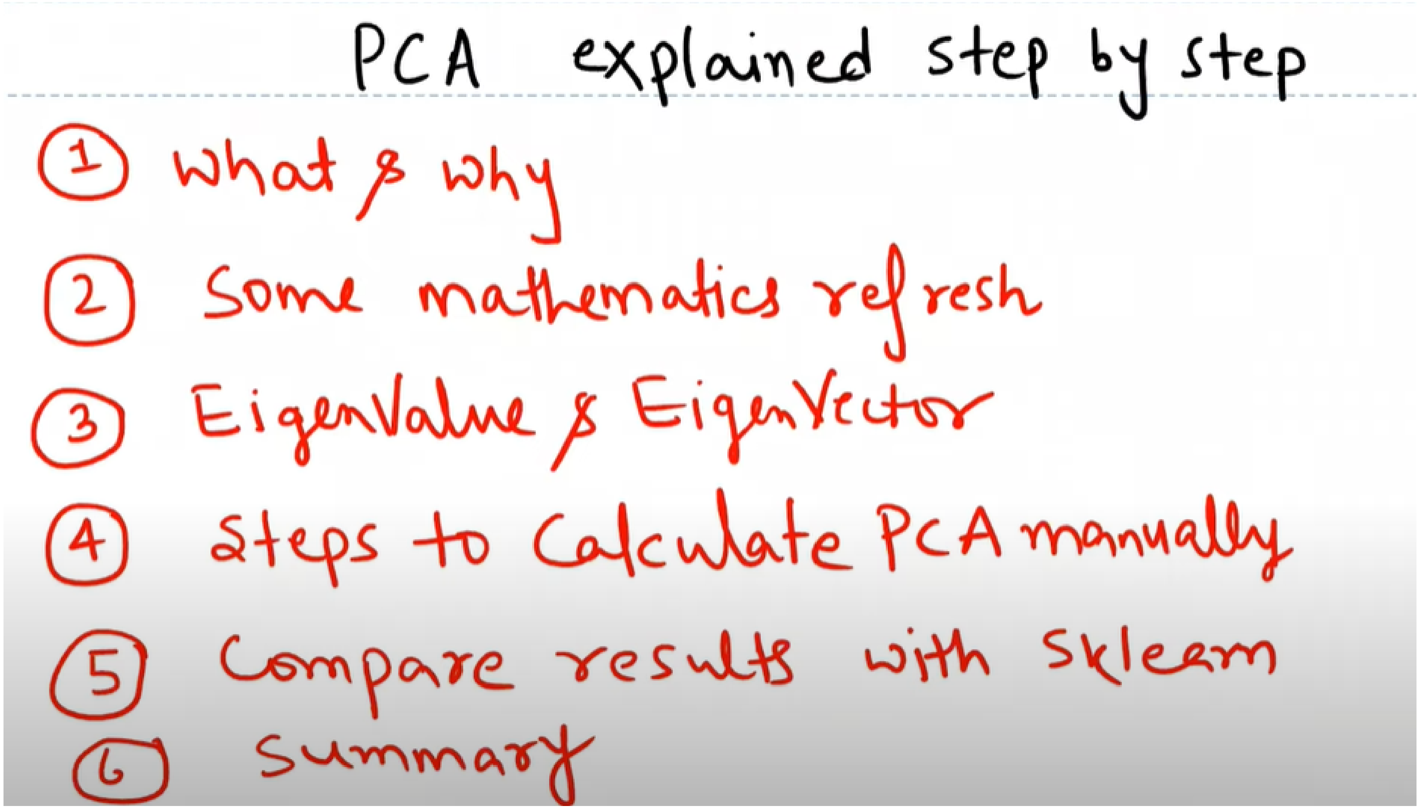
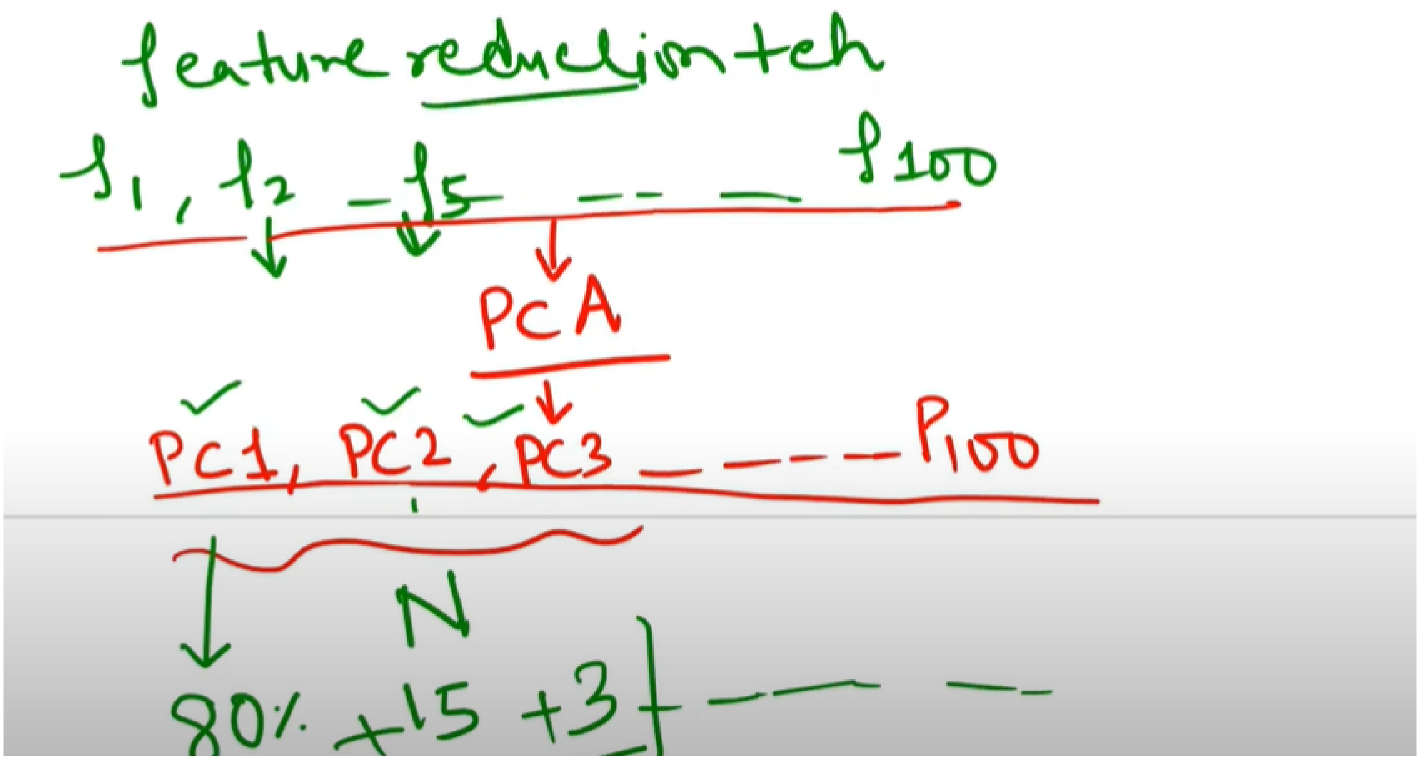
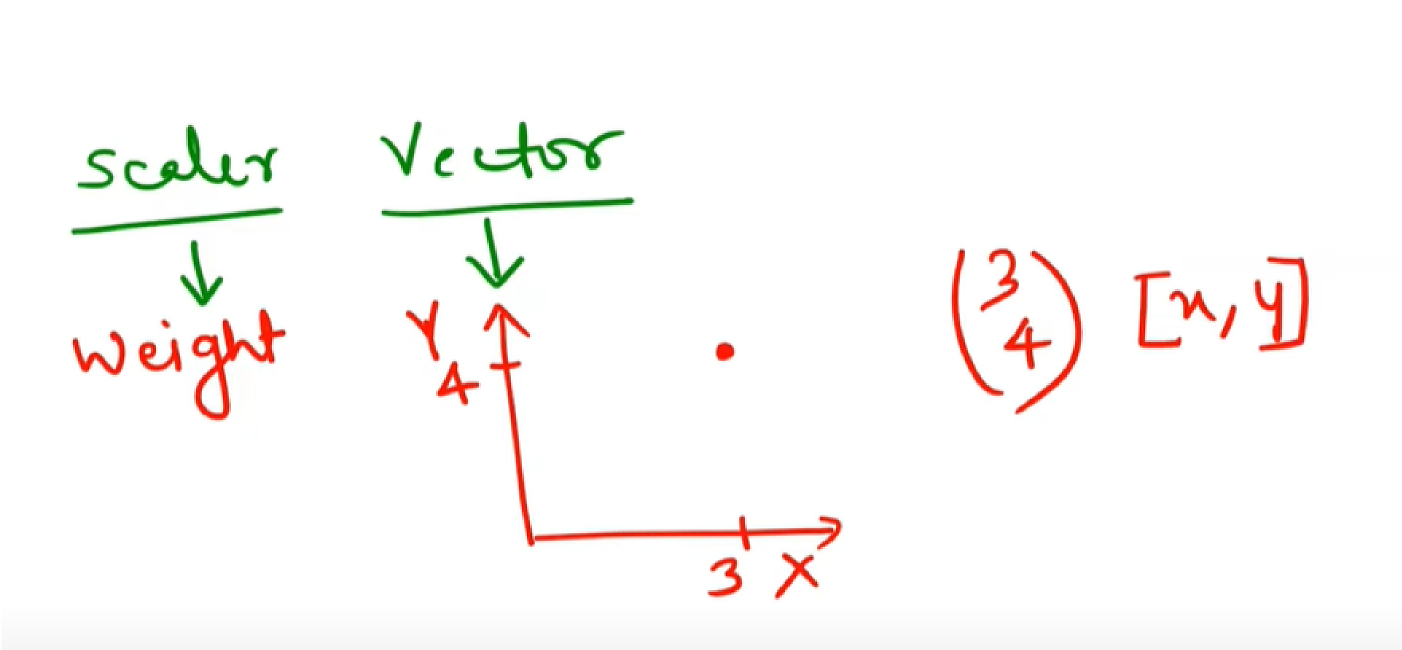
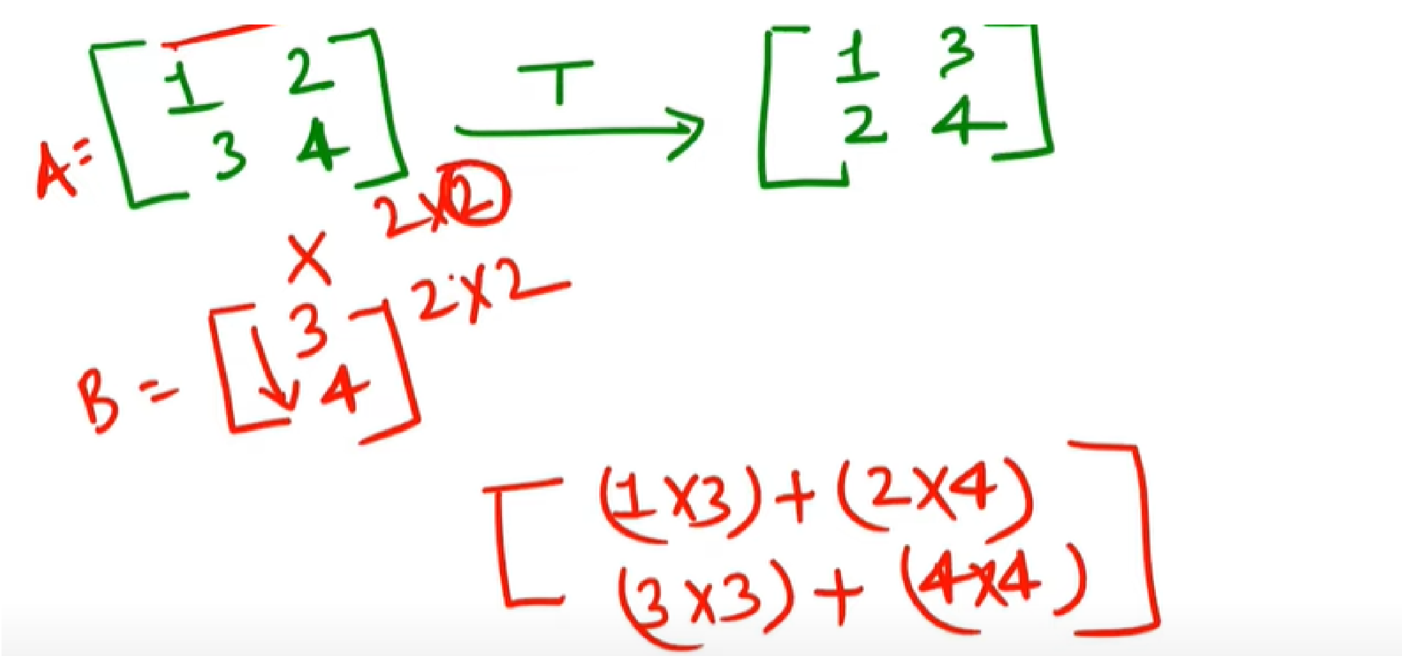

PCA explained step by step
1 What & why
2 Some mathematics refresh
3 EigenValue & EigenVector
4 steps to Calculate PCA manually
5 Compare results with Skleam
6 summary


feature reduction tech
f1, f2 — f5 — — — f100
PCA
PC1, PC2, PC3 — — — — P100
N
80% + 15 + 3

What is Scaler & Vector



Matrix Transpose $ Multipication



Eigen Value & Eigen Vector

$$\underline{A}\,\underline{V} = \lambda V$$

$\lambda \rightarrow$ Value, $V \rightarrow$ Vector

$$\begin{bmatrix} 3 & 6 \\ 5 & 4 \end{bmatrix} \times V = \lambda \times V$$

$$\begin{bmatrix} 3 & 6 \\ 5 & 4 \end{bmatrix} \times V = \lambda \times V \qquad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\underline{V}(\underline{M} - \lambda \underline{I}) = 0$$

$$\begin{bmatrix} 3 & 6 \\ 5 & 4 \end{bmatrix} \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}$$

$$\begin{bmatrix} 3 & 6 \\ 5 & 4 \end{bmatrix} \times V = \lambda \times V \qquad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\underline{V}(\underline{M} - \lambda \underline{I}) = 0$$

$$\begin{bmatrix} 3 & 6 \\ 5 & 4 \end{bmatrix} \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}$$

$$\begin{bmatrix} 3-\lambda & 6 \\ 5-0 & 4-\lambda \end{bmatrix} = 0$$

$$(3-\lambda)(4-\lambda) - 30 = 0$$

$$\lambda = 9, \lambda = -2$$

Steps for Developing PCA

→	Physics	Maths
s1	a	b
s2	c	d

Step1 → Define Data

step2 → Make Data mean

Double-click (or enter) to edit

Double-click (or enter) to edit

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from numpy.linalg import eig
```

```
Marks =np.array([[3,4],[2,8],[6,9]])
print(Marks)
```

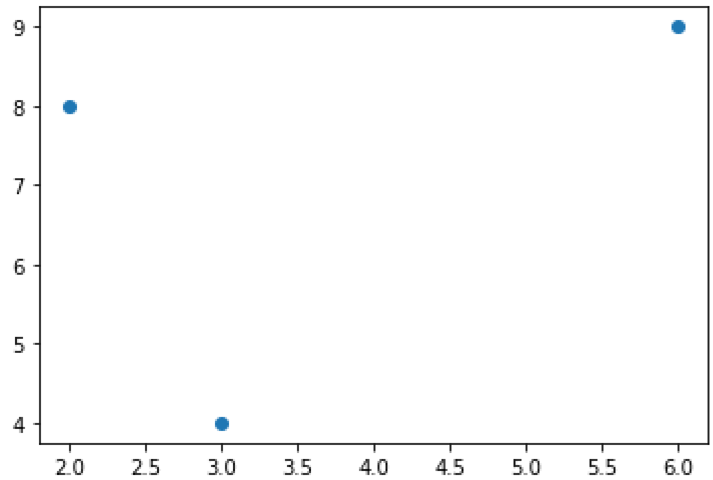
```
[[3 4]
 [2 8]
 [6 9]]
```

```
Marks_df= pd.DataFrame(Marks,columns=["Physics","Maths"])
Marks_df
```

	Physics	Maths
0	3	4
1	2	8
2	6	9

```
plt.scatter(Marks_df["Physics"],Marks_df["Maths"])
```

```
<matplotlib.collections.PathCollection at 0x7fb87a81d610>
```




```
#making data  mean Centric
Meanbycolumn=np.mean(Marks.T,axis=1)
print(Meanbycolumn)
```

```
[3.5 5.  7.5]
```

```
Scaled_Data = Marks- Meanbycolumn
print(Scaled_Data)
```

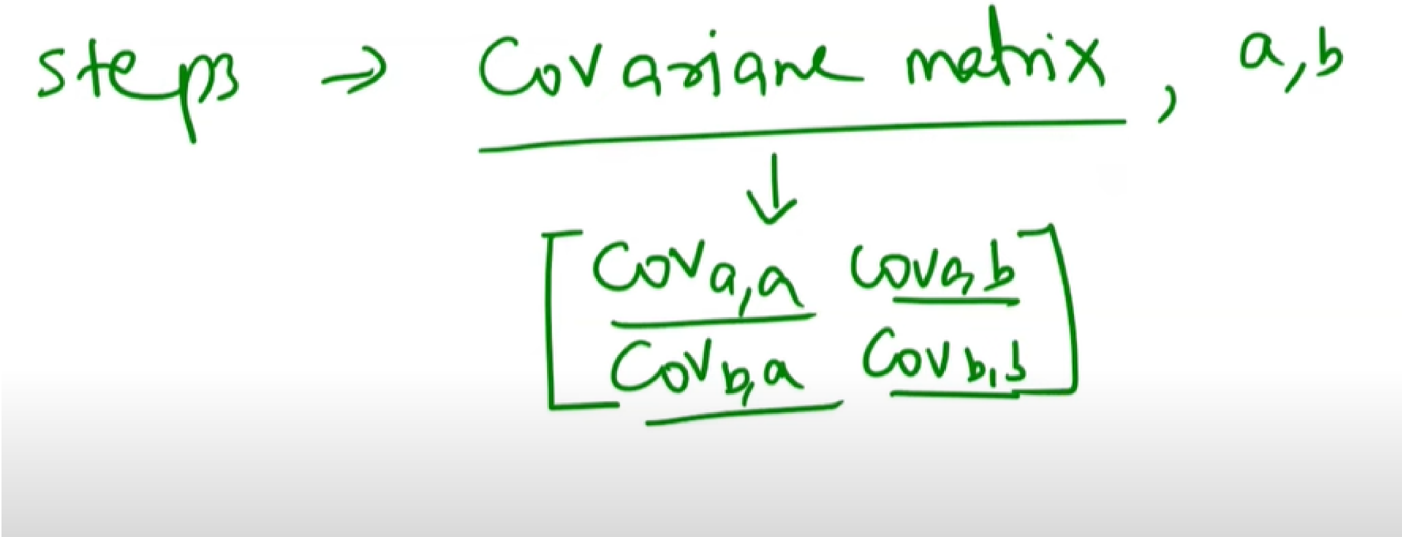
```
array([[-0.66666667, -3.        ],
       [-1.66666667,  1.        ],
       [ 2.33333333,  2.        ]])
```

```
Marks
```

```
array([[3, 4],
       [2, 8],
       [6, 9]])
```

```
print(Marks_df["Physics"].mean())
print(Marks_df["Maths"].mean())
```

```
3.6666666666666665
7.0
```



Covariance Matrix

Covariance matrix is a type of matrix that is used to represent the covariance values between pairs of elements given in a random vector. The covariance matrix can also be referred to as the variance covariance matrix. This is because the variance of each element is represented along the main diagonal of the matrix.

A covariance matrix is always a square matrix. Furthermore, it is positive semi-definite, and symmetric. This matrix is very useful in stochastic modeling and principle component analysis.

In this article, we will learn about the variance covariance matrix, its formula, examples, and various important properties associated with it.

What is Covariance Matrix?

Covariance matrix is a square matrix that displays the variance exhibited by elements of datasets and the covariance between a pair of datasets.

Variance is a measure of dispersion and can be defined as the spread of data from the mean of the given dataset.

Covariance is calculated between two variables and is used to measure how the two variables vary together.

Covariance Matrix Definition

Variance covariance matrix is defined as a square matrix where the diagonal elements represent the variance and the off-diagonal elements represent the covariance.

The covariance between two variables can be positive, negative, and zero.

A positive covariance indicates that the two variables have a positive relationship whereas negative covariance shows that they have a negative relationship.

If two elements do not vary together then they will display a zero covariance.

Covariance Matrix Example

Suppose there are two data sets X = {3, 2} and Y = {7, 4}.

The sample variance of dataset X = 0.5, and Y = 4.5.

The covariance between X and Y is 1.5. The covariance matrix is expressed as follows:

$$\begin{bmatrix} 0.5 & 1.5 \\ 1.5 & 4.5 \end{bmatrix}$$

Covariance Matrix Formula

Covariance Matrix Formula

$$\begin{bmatrix} Var(x_1) & \cdots\cdots & Cov(x_n,x_1) \\ \vdots & \ddots & \vdots \\ Cov(x_n,x_1) & \cdots\cdots & Var(x_n) \end{bmatrix}$$

To determine the covariance matrix, the formulas for variance and covariance are required. Depending upon the type of data available, the variance and covariance can be found for both sample data and population data. These formulas are given below.

Population Variance: var(x) = $\frac{\sum_1^n (x_i - \mu)^2}{n}$

Population Covariance: cov(x, y) = $\frac{\sum_1^n (x_i - \mu_x)(y_i - \mu_y)}{n}$

Sample Variance: var(x) = $\frac{\sum_1^n (x_i - \bar{x})^2}{n-1}$

Sample Covariance: cov(x, y) = $\frac{\sum_1^n (x_i - \bar{x})(y_i - \bar{y})}{n-1}$

μ = mean of population data.

$\bar{x}$ = mean of sample data.

n = number of observations in the dataset.

x_i = observations in dataset x.

Using these formulas, the general form of a variance covariance matrix is given as follows:

$$\begin{bmatrix} Var(x_1) & \cdots & Cov(x_1, x_n) \\ \vdots & \ddots & \vdots \\ \vdots & & \vdots \\ Cov(x_n, x_1) & \cdots & Var(x_n) \end{bmatrix}$$

Covariance Matrix 2 × 2

A 2 × 2 matrix is one which has 2 rows and 2 columns. The formula for a 2 × 2 covariance matrix is given as follows:

$$\begin{bmatrix} var(x) & cov(x, y) \\ cov(x, y) & var(y) \end{bmatrix}$$

Covariance Matrix 3 × 3

If there are 3 datasets, x, y, and z, then the formula to find the 3 × 3 covariance matrix is given below:

$$\begin{bmatrix} var(x) & cov(x, y) & cov(x, z) \\ cov(x, y) & var(y) & cov(y, z) \\ cov(x, z) & cov(y, z) & var(z) \end{bmatrix}$$

How To Calculate Covariance Matrix?

The number of variables determines the dimension of a variance-covariance matrix.

For example, if there are two variables (or datasets) it indicates that the covariance matrix will be 2 dimensional.

Suppose the math and science scores of 3 students are given as follows:

Student	Math (X)	Science (Y)
1	92	80
2	60	30
3	100	70

The steps to calculate the covariance matrix for the sample are given below:

- **Step 1:** Find the mean of one variable (X). This can be done by dividing the sum of all observations by the number of observations. Thus, (92 + 60 + 100) / 3 = 84
- **Step 2:** Subtract the mean from all observations; (92 - 84), (60 - 84), (100 - 84)
- **Step 3:** Take the sum of the squares of the differences obtained in the previous step. $(92 - 84)^2 + (60 - 84)^2 + (100 - 84)^2$.
- **Step 4:** Divide this value by 1 less than the total to get the sample variance of the first variable (X). var(X) = $[(92 - 84)^2 + (60 - 84)^2 + (100 - 84)^2] / (3 - 1) = 448$
- **Step 5:** Repeat steps 1 to 4 to find the variances of all variables. Using these steps, var(Y) = 700.
- **Step 6:** Choose a pair of variables (X and Y).
- **Step 7:** Subtract the mean of the first variable (X) from all observations; (92 - 84), (60 - 84), (100 - 84).

- **Step 8:** Repeat step 7 for the second variable (Y); (80 - 60), (30 - 60), (70 - 60).
- **Step 9:** Multiply the corresponding observations. (92 - 84)(80 - 60), (60 - 84)(30 - 60), (100 - 84)(70 - 60).
- **Step 10:** Add these values and divide them by (n - 1) to get the covariance. cov(x, y) = cov(y, x) = [(92 - 84)(80 - 60) + (60 - 84)(30 - 60) + (100 - 84)(70 - 60)] / (3 - 1) = 520.
- **Step 11:** Repeat steps 6 to 10 for different pairs of variables.
- **Step 12:** Now using the general formula for covariance matrix arrange these values in matrix form. Thus, the variance covariance matrix for the example is given as $\begin{bmatrix} 448 & 520 \\ 520 & 700 \end{bmatrix}$.

The same steps can be followed while calculating the covariance matrix for a population. The only difference is that the population variance and covariance formulas will be applied.

Example 3: How will you interpret the covariance matrix given below?

$$\begin{bmatrix} & X & Y & Z \\ X & 500 & 320 & -40 \\ Y & 320 & 340 & 0 \\ Z & -40 & 0 & 800 \end{bmatrix}$$

Solution: The variance covariance matrix can be interpreted as follows:

1) The diagonal elements 500, 340 and 800 indicate the variance in data sets X, Y and Z respectively. Y shows the lowest variance whereas Z displays the highest variance.

2) The covariance for X and Y is 320. As this is a positive number it means that when X increases (or decreases) Y also increases (or decreases)

3) The covariance for X and Z is -40. As it is a negative number it implies that when X increases Z decreases and vice - versa.

4) The covariance for Y and Z is 0. This means that there is no predictable relationship between the two data sets.

Q.2 Given a variance covariance matrix $\begin{array}{cc} & \begin{array}{cc} X & Y \end{array} \\ \begin{array}{c} X \\ Y \end{array} & \begin{bmatrix} 4 & 3 \\ 3 & 8 \end{bmatrix} \end{array}$. What is the variance of data set Y?

Double-click (or enter) to edit

Double-click (or enter) to edit

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from numpy.linalg import eig
```

```
Marks =np.array([[92,80],[60,30],[100,70]])
print(Marks)
```

```
[[ 92  80]
 [ 60  30]
 [100  70]]
```

```
Marks_df= pd.DataFrame(Marks,columns=["Maths(X)","Science(Y)"])
Marks_df
```

	Maths(X)	Science(Y)
0	92	80
1	60	30
2	100	70

How to calculate mean in pandas and numpy using Axis

```
import numpy as np
import pandas as pd
```

```
arr=np.array([[1,2,3],[2,3,4]])
arr
```

```
array([[1, 2, 3],
       [2, 3, 4]])
```

```
print(np.sum(arr,axis=1))
```

```
[6 9]
```

```
#making data  mean Centric
Meanbycolumn=np.mean(arr.T,axis=1)
print(Meanbycolumn)
```

```
[1.5 2.5 3.5]
```

```
import numpy as np
import pandas as pd

df=pd.DataFrame({'col1':[1,2,3],'col2':[2,3,4]})
df
```

	col1	col2
0	1	2
1	2	3
2	3	4

```
print(df.sum(axis=1) )
```

```
0    3
1    5
2    7
dtype: int64
```

```
#making data  mean Centric
df_new=df.T
Meanbycolumn=df_new.mean(axis=1)
print(Meanbycolumn)
```

```
col1    2.0
col2    3.0
dtype: float64
```

```
#making data  mean Centric
Meanbycolumn=np.mean(Marks.T,axis=1)
print(Meanbycolumn)
```

```
[84. 60.]
```

Properties of Covariance Matrix

Covariance matrix is a very important tool used by data scientists to understand and analyze multivariate data. Listed below are the various properties of this matrix that make it extremely useful.

A covariance matrix is always a square matrix. This means that the number of rows of the matrix will be equal to the number of columns.

The matrix is symmetric. Suppose M is the covariance matrix then MT = M.

It is positive semi-definite. Let u be a column vector, uT is the transpose of that vector and M be the covariance matrix then uTMu ≥ 0.

All eigenvalues of the variance covariance matrix are real and non-negative.

```
#Find Covariance matrix of above scaled data
Cov_mat= np.cov(Scaled_Data.T)
Cov_mat
```

```
array([[4.33333333, 2.5       ],
       [2.5       , 7.        ]])
```

find the Eigen Value and Eigen Vector of the above Covariance matrix

step4 → find λ_1, λ_2

$\underline{v_1}, \underline{v_2}$

Double-click (or enter) to edit

```
Eval, Evec =eig(Cov_mat)
print(Eval)
print(Evec)
```

```
[2.83333333 8.5       ]
[[-0.85749293 -0.51449576]
 [ 0.51449576 -0.85749293]]
```

Get Original Data Projected to principal Components as new axis

```
Projected_data = Evec.T.dot(Scaled_Data.T)
print(Projected_data.T)
```

```
[[-9.71825316e-01  2.91547595e+00]
 [ 1.94365063e+00  1.11022302e-16]
 [-9.71825316e-01 -2.91547595e+00]]
```

```
from sklearn.decomposition import PCA
pca= PCA(n_components=2)
pca.fit_transform(Marks)
```

```
array([[ 2.91547595e+00, -9.71825316e-01],
       [-7.37588530e-16,  1.94365063e+00],
       [-2.91547595e+00, -9.71825316e-01]])
```

variance explanation ratio by each PCA

```
pca.explained_variance_ratio_
```

```
array([0.75, 0.25])
```

```
PCDF=pd.DataFrame(data=pca.fit_transform(Marks),columns=['PC1','PC2'])
PCDF
```

	PC1	PC2
0	2.915476e+00	-0.971825
1	-7.375885e-16	1.943651
2	-2.915476e+00	-0.971825

Double-click (or enter) to edit

Assigment1 Find the covariance and variance for matrix of following sample data.

Examples on Covariance Matrix

Example 1: Find the population covariance matrix for the following table.

Score	Age
68	29
60	26
58	30
40	35

Example 2: Find the covariance and variance for matrix of following sample data.

Example 2: Find the covariance matrix for the following sample data.

X	Y	Z
15	12.5	50
35	15.8	55
20	9.3	70
14	20.1	65
28	5.2	80

Double-click (or enter) to edit