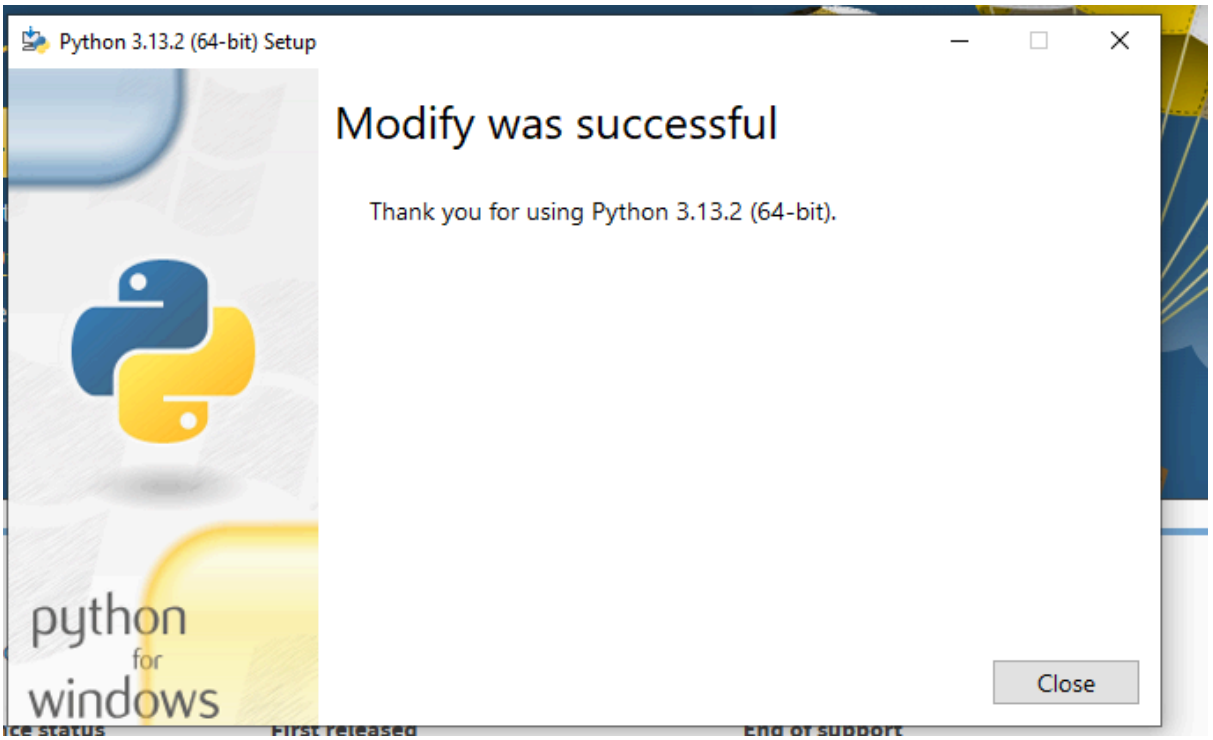
INSTALL OF PYTHON 3.13.2



Installation through the link Download Python | Python.org

AFTER INSTALLATION CHECKING IN COMMAND PROMPT:

```
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\student>python --version
Python 3.13.2

C:\Users\student>
```

Install of numerical python:

Type pip install numpy

```
C:\Users\student>pip install numpy
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: numpy in c:\users\student\appdata\local\packages\pythonsoftwarefoundation.python.3.1
3_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (2.2.3)

C:\Users\student>
```

After install we may check whether it is installed.

```
>>> import numpy
>>> print(numpy.__version__)
2.2.3
>>>
```

PERFORM SIMPLE OPERATION WITH NUMPY:

```
>>> import numpy as np
>>> arr = np.array([1,3,5,7,8,9])
>>> print("array:",arr)
array: [1 3 5 7 8 9]
>>> print("shape of the array:",arr.shape)
shape of the array: (6,)
>>> print("array size:",arr.size)
array size: 6
>>> print("sum of array:",np.sum(arr))
sum of array: 33
>>> print("mean of array:",np.mean(arr))
mean of array: 5.5
>>> print("std dev  of array:",np.std(arr))
std dev  of array: 2.8136571693556887
>>> matrix = np.array([[1,2,3,4],[5,6,7,8]])
>>> print("matrix:",matrix)
matrix: [[1 2 3 4]
 [5 6 7 8]]
>>> print("transpose:",matrix.T)
transpose: [[1 5]
 [2 6]
 [3 7]
 [4 8]]
>>>
```

It is simple based level
Move to further analysis

INFERENCE:

With the help of numpy we have a chance to add a array and work with array and 2D array and basic operation in array based and some descriptive analysis.

MEDIUM OPERATION USING NUMPY:

```
>>> import numpy as np
>>> A = np.array([[2, 4, 6], [1, 3, 5], [7, 8, 9]])
>>> B = np.array([[1, 0, 1], [2, 3, 2], [4, 5, 4]])
>>>
>>> print("Matrix A:\n", A)
Matrix A:
 [[2 4 6]
 [1 3 5]
 [7 8 9]]
>>> print("\nMatrix B:\n", B)

Matrix B:
 [[1 0 1]
 [2 3 2]
 [4 5 4]]
>>> sum_matrix = A + B
>>> print("\nElement-wise Addition:\n", sum_matrix)

Element-wise Addition:
 [[ 3  4  7]
 [ 3  6  7]
 [11 13 13]]
>>> mul_matrix = A * B
>>> print("\nElement-wise Multiplication:\n", mul_matrix)

Element-wise Multiplication:
 [[ 2  0  6]
 [ 2  9 10]
 [28 40 36]]
>>> dot_product = np.dot(A, B)
>>> print("\nMatrix Multiplication (Dot Product):\n", dot_product)

Matrix Multiplication (Dot Product):
 [[34 42 34]
 [27 34 27]
 [59 69 59]]
>>> col_sum = np.sum(A, axis=0)
>>> print("\nColumn-wise Sum:\n", col_sum)

Column-wise Sum:
 [10 15 20]
>>> row_sum = np.sum(A, axis=1)
>>> print("\nRow-wise Sum:\n", row_sum)

Row-wise Sum:
 [12  9 24]
>>> max_value = np.max(A)
>>> min_value = np.min(A)
>>> print("\nMaximum Value in A:", max_value)

Maximum Value in A: 9
>>> print("Minimum Value in A:", min_value)
Minimum Value in A: 1
>>>
```

INFERENCE:

Here we have to analysis on matrix level because which is mandatory to know the basic operation in array that is matrix.

Our view is spread from 1D to 2D. So using numpy is a basic knowledge to work with python.

Lets move to on further libraries..

INSTALLATION OF SCIPY:

```
C:\Users\student>pip install scipy
Defaulting to user installation because normal site-packages is not writeable
Collecting scipy
  Downloading scipy-1.15.2-cp313-cp313-win_amd64.whl.metadata (60 kB)
Requirement already satisfied: numpy<2.5,>=1.23.5 in c:\users\student\appdata\local\packages\pythonsoftwarefoundat
ion.python.3.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from scipy) (2.2.3)
Downloading scipy-1.15.2-cp313-cp313-win_amd64.whl (41.0 MB)
   ---------------------------------------- 41.0/41.0 MB 9.9 MB/s eta 0:00:00
Installing collected packages: scipy
Successfully installed scipy-1.15.2

C:\Users\student>S
```

Try with numpy is basic operation so let we move to the analysis that is complex operation through scientific python.

```
>>> import numpy as np
>>> from scipy import linalg
>>> A = np.array([[4, 3], [3, 2]])
>>> det_A = linalg.det(A)
>>> print("Determinant of A:", det_A)
Determinant of A: -1.0
>>> inv_A = linalg.inv(A)
>>> print("Inverse of A:\n", inv_A)
Inverse of A:
 [[-2.  3.]
 [ 3. -4.]]
```

Here we use linalg from scipy.The functions `linalg.det(A)` and `linalg.inv(A)` are part of SciPy's linear algebra module (`scipy.linalg`).

INFERENCE:

- **Computes the determinant of matrix A.**
- **The determinant is a single scalar value that represents some key properties of a matrix, such as whether it is invertible.**
- **If det(A) = 0, the matrix is singular (non-invertible).**

SOLVING THE EQUATION (AX=B)

```
>>> import numpy as np
>>> from scipy import linalg
>>> A = np.array([[3, 2], [1, 4]])
>>> B = np.array([5, 6])
>>> x = linalg.solve(A, B)
>>> print("Solution x:", x)
Solution x: [0.8 1.3]
```

FINDING EIGEN VALUES AND EIGEN VECTORS:

```
>>> eigenvalues, eigenvectors = linalg.eig(A)
>>>
>>> print("Eigenvalues:", eigenvalues)
Eigenvalues: [2.+0.j 5.+0.j]
>>> print("Eigenvectors:\n", eigenvectors)
Eigenvectors:
 [[-0.89442719 -0.70710678]
 [ 0.4472136  -0.70710678]]
>>>
```

EXPONENTIAL OF THE MATRIX:

```
>>> exp_A = linalg.expm(A)
>>> print("Matrix Exponential:\n", exp_A)
Matrix Exponential:
 [[ 54.39709043  94.01606867]
 [ 47.00803433 101.40512477]]
```

TO FIND THE INTEGRAL:

```
from scipy import integrate

def f(x):
                    return x**2 + 2*x + 1

result, error = integrate.quad(f, 0, 1)

print("Integral result:", result)
tegral result: 2.3333333333333335
```

INFERENCE:

Compare to numpy the scipy is work with interior operation. It is best for engineering and physics field.

INSTALLATION OF JUPYTER NOTEBOOK

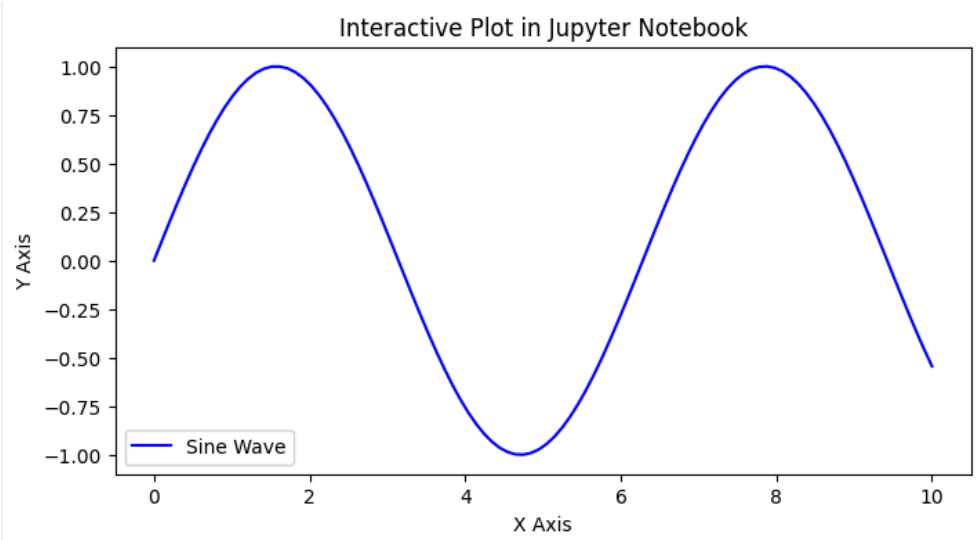
```
C:\Users\student>jupyter notebook
[I 2025-03-03 15:12:59.377 ServerApp] Extension package jupyter_lsp took 0.8547s to import
[I 2025-03-03 15:13:00.333 ServerApp] Extension package jupyter_server_terminals took 0.9149s to import
[I 2025-03-03 15:13:04.902 ServerApp] jupyter_lsp | extension was successfully linked.
[I 2025-03-03 15:13:04.907 ServerApp] jupyter_server_terminals | extension was successfully linked.
[I 2025-03-03 15:13:04.915 ServerApp] jupyterlab | extension was successfully linked.
[I 2025-03-03 15:13:04.921 ServerApp] notebook | extension was successfully linked.
[I 2025-03-03 15:13:18.078 ServerApp] notebook_shim | extension was successfully linked.
[I 2025-03-03 15:13:18.400 ServerApp] notebook_shim | extension was successfully loaded.
[I 2025-03-03 15:13:18.403 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2025-03-03 15:13:18.404 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[I 2025-03-03 15:13:18.495 LabApp] JupyterLab extension loaded from C:\Users\student\AppData\Local\Packages\Pytho
reFoundation.Python.3.13_qbz5n2kfra8p0\LocalCache\local-packages\Python313\site-packages\jupyterlab
```

OPERATIONS IN JUPYTER NOTEBOOK

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

x = np.linspace(0, 10, 100)

y = np.sin(x)

df = pd.DataFrame({'X Values': x, 'Y Values': y})

df.head()

plt.figure(figsize=(8, 4))
```

```
plt.plot(x, y, label='Sine Wave', color='b')

plt.xlabel('X Axis')

plt.ylabel('Y Axis')

plt.title('Interactive Plot in Jupyter Notebook')

plt.legend()

plt.show()
```

Output:



INSTALLATION OF STATSMODEL:

```
C:\Users\student>pip install statsmodels
Defaulting to user installation because normal site-packages is not writeable
Collecting statsmodels
  Downloading statsmodels-0.14.4-cp313-cp313-win_amd64.whl.metadata (9.5 kB)
Requirement already satisfied: numpy<3,>=1.22.3 in c:\users\student\appdata\local\packages\pythonsoftwarefoundat
on.3.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from statsmodels) (2.2.3)
Requirement already satisfied: scipy!=1.9.2,>=1.8 in c:\users\student\appdata\local\packages\pythonsoftwarefound
thon.3.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from statsmodels) (1.15.2)
Requirement already satisfied: pandas!=2.1.0,>=1.4 in c:\users\student\appdata\local\packages\pythonsoftwarefoun
ython.3.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from statsmodels) (2.2.3)
Collecting patsy>=0.5.6 (from statsmodels)
  Downloading patsy-1.0.1-py2.py3-none-any.whl.metadata (3.3 kB)
Requirement already satisfied: packaging>=21.3 in c:\users\student\appdata\local\packages\pythonsoftwarefoundati
n.3.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from statsmodels) (24.2)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\student\appdata\local\packages\pythonsoftwaref
n.python.3.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from pandas!=2.1.0,>=1.4->statsmo
.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\student\appdata\local\packages\pythonsoftwarefoundation.
.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from pandas!=2.1.0,>=1.4->statsmodels) (202
Requirement already satisfied: tzdata>=2022.7 in c:\users\student\appdata\local\packages\pythonsoftwarefoundatio
.3.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from pandas!=2.1.0,>=1.4->statsmodels) (2
Requirement already satisfied: six>=1.5 in c:\users\student\appdata\local\packages\pythonsoftwarefoundation.pyth
qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from python-dateutil>=2.8.2->pandas!=2.1.0,>=1.
models) (1.17.0)
Downloading statsmodels-0.14.4-cp313-cp313-win_amd64.whl (9.8 MB)
   ---------------------------------------- 9.8/9.8 MB 10.4 MB/s eta 0:00:00
```

```
>>> import statsmodels
>>> print(statsmodels.__version__)
0.14.4
>>>
```

```
>>> import numpy as np
>>> import statsmodels.api as sm
>>>
>>> # Data: Age and Salary as features, Purchase (0 or 1) as
>>> X = np.array([[22, 20000], [30, 30000], [40, 40000], [50,
>>> y = np.array([0, 0, 1, 1, 1])
>>>
>>> X = sm.add_constant(X)  # Add intercept
>>>
>>> # Fit logistic regression model
>>> model = sm.Logit(y, X).fit(disp=False)
C:\Users\student\AppData\Local\Packages\PythonSoftwareFoundat
iction detected, parameter may not be identified
  warnings.warn(msg, category=PerfectSeparationWarning)
C:\Users\student\AppData\Local\Packages\PythonSoftwareFoundat
iction detected, parameter may not be identified
  warnings.warn(msg, category=PerfectSeparationWarning)
C:\Users\student\AppData\Local\Packages\PythonSoftwareFoundat
iction detected, parameter may not be identified
  warnings.warn(msg, category=PerfectSeparationWarning)
C:\Users\student\AppData\Local\Packages\PythonSoftwareFoundat
iction detected, parameter may not be identified
  warnings.warn(msg, category=PerfectSeparationWarning)
C:\Users\student\AppData\Local\Packages\PythonSoftwareFoundat
iction detected, parameter may not be identified
  warnings.warn(msg, category=PerfectSeparationWarning)
C:\Users\student\AppData\Local\Packages\PythonSoftwareFoundat
iction detected, parameter may not be identified
  warnings.warn(msg, category=PerfectSeparationWarning)
C:\Users\student\AppData\Local\Packages\PythonSoftwareFoundat
iction detected, parameter may not be identified
  warnings.warn(msg, category=PerfectSeparationWarning)
C:\Users\student\AppData\Local\Packages\PythonSoftwareFoundat
iction detected, parameter may not be identified
  warnings.warn(msg, category=PerfectSeparationWarning)
```

```
C:\Users\student\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.1
verge. Check mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
>>>
>>> # Print only key results
>>> print("Model Coefficients:", model.params)
Model Coefficients: [-1.64285251e+02  9.50879106e+00 -4.80073027e-03]
>>> print("Predicted Classes:", (model.predict(X) > 0.5).astype(int))
Predicted Classes: [0 0 1 1 1]
>>>
```

```
Predicted Classes: [0 0 1 1 1]
>>>
```

INSTALLATION OF PANDAS:

```
C:\Users\student>pip install pandas
Defaulting to user installation because normal site-packages is not writeable
Collecting pandas
  Downloading pandas-2.2.3-cp313-cp313-win_amd64.whl.metadata (19 kB)
Requirement already satisfied: numpy>=1.26.0 in c:\users\student\appdata\local\packages\pythonsoftware
3.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from pandas) (2.2.3)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\student\appdata\local\packages\pytho
n.python.3.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from pandas) (2.9.0.pos
Collecting pytz>=2020.1 (from pandas)
  Downloading pytz-2025.1-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.7 (from pandas)
  Downloading tzdata-2025.1-py2.py3-none-any.whl.metadata (1.4 kB)
Requirement already satisfied: six>=1.5 in c:\users\student\appdata\local\packages\pythonsoftwarefound
qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from python-dateutil>=2.8.2->pandas)
Downloading pandas-2.2.3-cp313-cp313-win_amd64.whl (11.5 MB)
   ---------------------------------------- 11.5/11.5 MB 8.8 MB/s eta 0:00:00
Downloading pytz-2025.1-py2.py3-none-any.whl (507 kB)
Downloading tzdata-2025.1-py2.py3-none-any.whl (346 kB)
Installing collected packages: pytz, tzdata, pandas
Successfully installed pandas-2.2.3 pytz-2025.1 tzdata-2025.1
```

```
>>> import pandas as pd
>>> print(pd.__version__)
2.2.3
>>>
```

CREATING DATAFRAMES AND PRINT IT

```
>>> import pandas as pd
>>>
>>> data = {'Name': ['Alice', 'Bob', 'Charlie'],
...         'Age': [25, 30, 35],
...                 'City': ['New York', 'London', 'Paris']}
>>>
>>> df = pd.DataFrame(data)
>>>
>>> print("DataFrame:\n", df)
DataFrame:
       Name  Age      City
0    Alice   25  New York
1      Bob   30    London
2  Charlie   35     Paris
>>> print("\nNames column:\n", df['Name'])

Names column:
0      Alice
1        Bob
2    Charlie
Name: Name, dtype: object
>>>
```

FILTERING THE DATA

```
>>> filtered_df = df[df['Age'] > 25]
>>> print("\nFiltered Data (Age > 25):\n", filtered_df)

Filtered Data (Age > 25):
      Name  Age    City
1      Bob   30  London
2  Charlie   35   Paris
>>>
>>> df['Salary'] = [50000, 60000, 70000]
>>> print("\nDataFrame after adding Salary column:\n", df)

DataFrame after adding Salary column:
      Name  Age      City  Salary
0    Alice   25  New York   50000
1      Bob   30    London   60000
2  Charlie   35     Paris   70000
>>>
```

SUMMARY STATISTICS:

```
Summary Statistics:
             Age   Salary
count    3.0      3.0
mean    30.0  60000.0
std      5.0  10000.0
min     25.0  50000.0
25%     27.5  55000.0
50%     30.0  60000.0
75%     32.5  65000.0
max     35.0  70000.0
```

ASCENDING THE VALUES:

```
>>> sorted_df = df.sort_values(by='Age', ascending=False)
>>> print("\nDataFrame Sorted by Age (Descending):\n", sorted_df

DataFrame Sorted by Age (Descending):
      Name  Age      City  Salary
2  Charlie   35     Paris   70000
1      Bob   30    London   60000
0    Alice   25  New York   50000
>>>
```

UNIQUE VALUES:

```
>>> unique_cities = df['City'].unique()
>>> print("\nUnique Cities:\n", unique_cities)

Unique Cities:
 ['New York' 'London' 'Paris']
>>>
>>> city_counts = df['City'].value_counts()
>>> print("\nCount of Each City:\n", city_counts)

Count of Each City:
 City
New York    1
London      1
Paris       1
Name: count, dtype: int64
>>>
```

MERGING THE VALUES:

```
>>> merged_df = pd.merge(df, extra_data, on='Name')
>>> print("\nMerged DataFrame:\n", merged_df)

Merged DataFrame:
      Name  Age      City  Salary_x  Salary_y
0    Alice   25  New York     50000     50000
1      Bob   30    London     60000     60000
2  Charlie   35     Paris     70000     55000
```

HANDLING THE MISSING VALUES:

```
>>> df.loc[2, 'Salary'] = None  # Introducing a missing value
>>> print("\nDataFrame with Missing Value:\n", df)

DataFrame with Missing Value:
      Name  Age      City   Salary
0    Alice   25  New York  50000.0
1      Bob   30    London  60000.0
2  Charlie   35     Paris      NaN
>>>
>>> df_filled = df.fillna(df['Salary'].mean())
>>> print("\nDataFrame after Filling Missing Values:\n", df_filled)

DataFrame after Filling Missing Values:
      Name  Age      City   Salary
0    Alice   25  New York  50000.0
1      Bob   30    London  60000.0
2  Charlie   35     Paris  55000.0
>>>
```

INFERENCE:

The analysis helps identify trends in age and salary, correlation, and missing values while providing insights through grouping, sorting etc..