**Python Libraries:** Numpy, Pandas, Matplotlib, and Seaborn

**1. Numpy**

Numpy (Numerical Python) is a library for numerical computations in Python. It provides support for arrays, matrices, and mathematical functions.

**Key Features:**

- Multi-dimensional arrays (ndarray).

- Mathematical operations on arrays.

- Linear algebra, Fourier transform, and random number capabilities.

**Commonly Used Functions:**

**1. Array Creation:**

python

import numpy as np

# Creating arrays

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

zeros = np.zeros((2, 2))

ones = np.ones((3, 3))

linspace = np.linspace(0, 10, 5) # Evenly spaced numbers

**2. Mathematical Operations:**

python

arr1 = np.array([1, 2, 3])

arr2 = np.array([4, 5, 6])

# Element-wise operations

sum\_arr = arr1 + arr2

product\_arr = arr1 \* arr2

sqrt\_arr = np.sqrt(arr1)

**3. Array Manipulation:**

python

arr = np.array([[1, 2, 3], [4, 5, 6]])

# Transpose

transposed = arr.T

# Reshape

reshaped = arr.reshape(3, 2)

**4. Random Numbers:**

python

random\_array = np.random.rand(3, 3) Random numbers between 0 and 1

[Click here to learn more](https://www.datacamp.com/tutorial/python-numpy-tutorial)

-------------------------------------------------------------------------------------------------------

**2. Pandas**

Pandas is a library for data manipulation and analysis. It provides data structures like Series and DataFrame.

Key Features:

- Data manipulation and analysis.

- Handling missing data.

- Merging, grouping, and pivoting data.

Commonly Used Functions:

1. **Data Structures:**

python

import pandas as pd

Series

series = pd.Series([1, 2, 3, 4])

DataFrame

data = {'Name': ['Alice', 'Bob'], 'Age': [25, 30]}

df = pd.DataFrame(data)

2. **Reading and Writing Data:**

python

Reading data

df = pd.read\_csv('data.csv')

Writing data

df.to\_csv('output.csv', index=False)

3. **Data Analysis:**

python

Viewing data

df.head()

df.describe()

# Filtering data

filtered = df[df['Age'] > 25]

4. **Handling Missing Data:**

python

df.fillna(0, inplace=True) # Replace NaNs with 0

df.dropna(inplace=True) # Remove rows with NaNs

[Click here to learn more](https://www.datacamp.com/tutorial/pandas)

-----------------------------------------------------------------------------------------------------

**3. Matplotlib**

Matplotlib is a plotting library for creating static, interactive, and animated visualizations in Python.

Key Features:

- Line plots, bar charts, scatter plots, and more.

- Highly customizable plots.

Commonly Used Functions:

1. **Line Plot:**

python

import matplotlib.pyplot as plt

x = [1, 2, 3, 4]

y = [10, 20, 25, 30]

plt.plot(x, y)

plt.title("Line Plot")

plt.xlabel("X-axis")

plt.ylabel("Y-axis")

plt.show()

2. **Bar Chart:**

python

categories = ['A', 'B', 'C']

values = [5, 7, 3]

plt.bar(categories, values)

plt.title("Bar Chart")

plt.show()

3. **Scatter Plot:**

python

x = [5, 7, 8, 7]

y = [8, 5, 6, 7]

plt.scatter(x, y)

plt.title("Scatter Plot")

plt.show()

4. **Histogram:**

python

data = [1, 2, 2, 3, 3, 3, 4, 4, 5]

plt.hist(data, bins=5)

plt.title("Histogram")

plt.show()

[Click here to learn](https://www.datacamp.com/tutorial/matplotlib-tutorial-python)

-----------------------------------------------------------------------------------------------------

**4. Seaborn**

Seaborn is a data visualization library built on top of Matplotlib. It provides an interface for creating informative and attractive statistical graphics.

Key Features:

- Easy integration with Pandas.

- Attractive default styles.

- Supports complex visualizations like heatmaps and violin plots.

Commonly Used Functions:

1. **Line Plot:**

python

import seaborn as sns

sns.set\_theme()

data = sns.load\_dataset("flights")

sns.lineplot(data=data, x="year", y="passengers")

plt.show()

2. **Bar Plot:**

python

sns.barplot(x=["A", "B", "C"], y=[5, 7, 3])

plt.show()

3. **Heatmap:**

python

data = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

sns.heatmap(data, annot=True)

plt.show()

**4. Pair Plot:**

python

iris = sns.load\_dataset("iris")

sns.pairplot(iris, hue="species")

plt.show()

[Click here to learn](https://www.datacamp.com/tutorial/seaborn-python-tutorial)

By mastering these libraries, you can efficiently handle data and create insightful visualizations for your projects.