

# Python for Data Analysis & Visualization

---

## Complete Beginner-to-Intermediate Guide

---

**Instructor: Vinod**

---

### **1** Introduction

---

#### What You Will Learn

---

By the end of this session, you will be able to:

- Install Python and required libraries
  - Work with NumPy arrays
  - Manipulate data using Pandas
  - Clean and transform datasets
  - Create professional visualizations using:
    - Matplotlib
    - Seaborn
  - Perform basic exploratory data analysis (EDA)
- 

### **2** Software Installation Guide (Step-by-Step)

---

#### Option 1 (Recommended for Beginners): Install Anaconda

---

##### Step 1: Download Anaconda

Go to: <https://www.anaconda.com/products/distribution>

Download:

- Python 3.x version
- For Windows / macOS / Linux (as per your OS)

##### Step 2: Install Anaconda

- Double click installer

- Click Next
- Keep default settings
- Install for "Just Me"
- Add to PATH (if option available)
- Finish installation

### Step 3: Verify Installation

Open:

- Anaconda Prompt (Windows)
- Terminal (Mac/Linux)

Type:

```
conda --version  
python --version
```

---

## Option 2: Install Python + pip (Manual Method)

### Step 1: Install Python

Download from: <https://www.python.org/downloads/>

During installation:

- ☒ Check "Add Python to PATH"
- Click Install

Verify:

```
python --version  
pip --version
```

---

## 3 Install Required Libraries

If using Anaconda:

```
conda install numpy pandas matplotlib seaborn jupyter
```

If using pip:

```
pip install numpy pandas matplotlib seaborn jupyter
```

---

## 4 Launch Jupyter Notebook

---

In terminal:

```
jupyter notebook
```

Browser will open → Click **New** → **Python 3**

---

## 5 NumPy (Numerical Python)

---

### What is NumPy?

---

NumPy is a library used for:

- Numerical computation
- Multi-dimensional arrays
- Linear algebra
- Statistical operations

---

### Import NumPy

---

```
import numpy as np
```

---

### Creating Arrays

---

#### From List

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

#### 2D Array

```
arr2 = np.array([[1,2,3],  
                 [4,5,6]])
```

---

## Special Arrays

---

```
np.zeros((3,3))  
np.ones((2,4))  
np.eye(3)  
np.arange(0,10)  
np.linspace(0,10,5)
```

---

## Array Attributes

---

```
arr.shape  
arr.ndim  
arr.size  
arr.dtype
```

---

## Indexing & Slicing

---

```
arr[0]  
arr[1:4]  
  
arr2[0,1]
```

---

## Mathematical Operations

---

```
a = np.array([1,2,3])  
b = np.array([4,5,6])  
  
a + b  
a * b  
np.sqrt(a)  
np.mean(a)  
np.sum(a)
```

---

## Random Module

---

```
np.random.rand(3,3)
np.random.randint(1,10,5)
```

---

## 6 Pandas

---

### What is Pandas?

---

Pandas is used for:

- Data manipulation
  - Data cleaning
  - Working with structured data (tables)
- 

### Import Pandas

---

```
import pandas as pd
```

---

### Series

---

```
s = pd.Series([10,20,30])
```

---

### DataFrame

---

```
data = {
    "Name": ["Amit", "Ravi", "Sneha"],
    "Age": [25, 30, 28],
    "Salary": [50000, 60000, 55000]
}

df = pd.DataFrame(data)
```

---

## Load Data

---

### CSV

```
df = pd.read_csv("data.csv")
```

### Excel

```
df = pd.read_excel("data.xlsx")
```

## Inspect Data

---

```
df.head()  
df.tail()  
df.info()  
df.describe()  
df.shape  
df.columns
```

## Selecting Data

---

```
df["Age"]  
df[["Name", "Salary"]]  
  
df.loc[0]  
df.iloc[0]
```

## Filtering

---

```
df[df["Age"] > 26]
```

## Add New Column

---

```
df["Bonus"] = df["Salary"] * 0.10
```

---

## Handle Missing Values

---

```
df.isnull()  
df.dropna()  
df.fillna(0)
```

---

## GroupBy

---

```
df.groupby("Department")["Salary"].mean()
```

---

## Sorting

---

```
df.sort_values("Salary", ascending=False)
```

---

## Matplotlib

---

---

### What is Matplotlib?

---

Matplotlib is a plotting library used to create:

- Line charts
- Bar charts
- Pie charts
- Histograms
- Scatter plots

---

### Import

---

```
import matplotlib.pyplot as plt
```

## Line Plot

---

```
x = [1,2,3,4]
y = [10,20,25,30]

plt.plot(x,y)
plt.title("Simple Line Plot")
plt.xlabel("X Axis")
plt.ylabel("Y Axis")
plt.show()
```

## Bar Chart

---

```
names = ["A", "B", "C"]
marks = [80, 90, 75]

plt.bar(names, marks)
plt.show()
```

## Histogram

---

```
data = np.random.randn(1000)
plt.hist(data, bins=30)
plt.show()
```

## Scatter Plot

---

```
plt.scatter(x,y)
plt.show()
```

## Seaborn

---

### What is Seaborn?

---

Seaborn is built on top of Matplotlib. It provides:



- Beautiful statistical plots
- Better styling
- Built-in datasets

---

## Import

---

```
import seaborn as sns
```

---

## Built-in Dataset

---

```
df = sns.load_dataset("tips")  
df.head()
```

---

## Line Plot

---

```
sns.lineplot(x="total_bill", y="tip", data=df)  
plt.show()
```

---

## Bar Plot

---

```
sns.barplot(x="day", y="total_bill", data=df)  
plt.show()
```

---

## Box Plot

---

```
sns.boxplot(x="day", y="total_bill", data=df)
```

---

## Heatmap

---

```
corr = df.corr()  
sns.heatmap(corr, annot=True)
```

---

## 9 Mini Project (End-to-End Example)

---

### Step 1: Load Dataset

---

```
df = pd.read_csv("sales_data.csv")
```

### Step 2: Explore

---

```
df.info()  
df.describe()
```

### Step 3: Clean Data

---

```
df.dropna(inplace=True)
```

### Step 4: Analysis

---

```
df.groupby("Region")["Sales"].sum()
```

### Step 5: Visualization

---

```
sns.barplot(x="Region", y="Sales", data=df)  
plt.show()
```

---

## 10 Best Practices

---

- Always check missing values
- Always explore data using describe()
- Use proper labels in graphs

- Never modify original data directly (keep copy)
- Use meaningful variable names

---

## 1 1 Common Errors

---

Error	Reason	Fix
ModuleNotFoundError	Library not installed	pip install library
FileNotFoundError	Wrong path	Check file path
ValueError	Wrong data type	Check column types

---

## 1 2 Practice Exercises

---

1. Create a NumPy array of 50 random numbers and calculate mean & standard deviation.
  2. Load a CSV file and:
    - Show top 10 rows
    - Filter rows where value > 100
  3. Create:
    - Line plot
    - Histogram
    - Boxplot
  4. Use seaborn to analyze tips dataset.
- 

## 1 3 Summary

---

You have learned:

- Environment setup
- NumPy fundamentals
- Pandas data manipulation
- Data cleaning
- Data visualization
- Exploratory Data Analysis workflow