

**PROGRAM:** Create a dataframe using a list of elements.

```
import numpy as np
import pandas as pd
data = np.array(['', 'Col1', 'Col2'], ['Row1', 1, 2],
['Row2', 3, 4])
print(pd.DataFrame(data=data[1:,1:],
index = data [1:,0], columns=data[0,1:]))
# Take a 2D array as input to your DataFrame
my_2darray = np.array([[1, 2, 3], [4, 5, 6]])
print(pd.DataFrame(my_2darray))
# Take a dictionary as input to your DataFrame
my_dict = {1: ['1', '3'], 2: ['1', '2'], 3: ['2', '4']}
print(pd.DataFrame(my_dict))
# Take a DataFrame as input to your DataFrame
my_df = pd.DataFrame(data=[4,5,6,7], index=range(0,4), columns=['A'])
print(pd.DataFrame(my_df))
# Take a Series as input to your DataFrame
my_series = pd.Series({"United Kingdom":"London", "India":"New Delhi", "United
States":"Washington", "Belgium":"Brussels"})
print(pd.DataFrame(my_series))
df = pd.DataFrame(np.array([[1, 2, 3], [4, 5, 6]]))
# Use the `shape` property print(df.shape)

# Or use the `len()` function with the `index` property
print(len(df.index))
```

Output:

```
***
      Col1 Col2
Row1      1    2
Row2      3    4
0      1    2
0      1    2    3
1      4    5    6
0      1    2    3
0      1    1    2
1      3    2    4
      A
0      4
1      5
2      6
3      7

United Kingdom      London
India              New Delhi
United States      Washington
Belgium            Brussels
2
```

**Result:**

Thus the working with Pandas data frames was successfully completed.

## PRACTICAL-1

Download, Install and Explore the Features of NumPy, SciPy, Jupyter, Statsmodels and Pandas  
Objective

To download, install, and explore the basic features of Python libraries NumPy, SciPy, Pandas, Statsmodels, and the Jupyter Notebook environment used for Data Science and Analytics.

## Software Requirements

- Operating System: Windows / Linux / macOS
- Python Version: Python 3.8 or above
- Internet Connection

### STEP 1: Install Python

1. Open a web browser.
2. Go to: <https://www.python.org>
3. Click on Downloads.
4. Download the latest Python 3.x version.
5. Run the installer.
6. IMPORTANT:  
Check the box "Add Python to PATH"
7. Click Install Now.
8. After installation, open Command Prompt and type:

`python --version`

If Python version is displayed, Python is installed successfully.

### STEP 2: Install Jupyter Notebook

Jupyter Notebook is used to write and run Python code interactively.

1. Open Command Prompt.
2. Run the command:

`pip install notebook`

3. After installation, start Jupyter Notebook using:

`jupyter notebook`

A browser window will open with Jupyter Dashboard.

### STEP 3: Install Required Python Packages

Install all required libraries using pip.

Run the following commands one by one in Command Prompt:

```
pip install numpy
pip install scipy
pip install pandas
pip install statsmodels
pip install matplotlib
pip install seaborn
```

All packages will be downloaded and installed automatically.

### STEP 4: Verify Installation in Jupyter Notebook

1. Open Jupyter Notebook.
2. Click New → Python 3 Notebook.
3. In a new cell, type:  
`import numpy as np`  
`import pandas as pd`  
`import scipy`  
`import statsmodels.api as sm`  
`print("All packages imported successfully")`

If no error appears, installation is successful.

### STEP 5: Explore NumPy Features

NumPy is used for numerical computing and array operations.

```
import numpy as np

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

print(arr)
print("Mean:", np.mean(arr))
print("Sum:", np.sum(arr))
```

Key NumPy Features

- Fast numerical operations
- Multi-dimensional arrays
- Mathematical functions

### **STEP 6: Explore Pandas Features**

Pandas is used for data handling and analysis.

```
import pandas as pd
data = {
    'Name': ['Amit', 'Sneha', 'Rohit'],
    'Marks': [85, 92, 78]
}
```

```
df = pd.DataFrame(data)
print(df)
```

Key Pandas Features

- DataFrames and Series
- Reading CSV/Excel files
- Data cleaning and manipulation

### **STEP 7: Explore SciPy Features**

SciPy is used for scientific and statistical computations.

```
from scipy import stats
data = [10, 20, 30, 40, 50]
print("Mean:", stats.tmean(data))
print("Standard Deviation:", stats.tstd(data))
```

Key SciPy Features

- Statistical functions
- Optimization
- Scientific calculations

### **STEP 8: Explore Statsmodels Features**

Statsmodels is used for statistical modeling and analysis.

```
import statsmodels.api as sm
data = [10, 20, 30, 40, 50]
print("Descriptive Statistics:")
print(sm.stats.DescrStatsW(data).summary())
```

Key Statsmodels Features

- Regression analysis
- ANOVA
- Statistical tests

### **STEP 9: Explore Jupyter Notebook Features**

Jupyter Notebook allows:

- Writing code in cells
- Running code step-by-step
- Adding text, equations, and outputs

Try:

- Markdown cells for theory
- Code cells for execution
- Saving notebooks as .ipynb