Lab – 29

Student Name: Navneet P

Student ID: AF0411619

Topic: Scipy Introduction

What is Scipy?

It is built on top of NumPy, another popular Python library for numerical computing, and provides additional functionality for various scientific and engineering tasks. Scipy is widely used in fields such as mathematics, physics, engineering, biology, and data science. Installation using **pip install scipy**.

SciPy vs NumPy

Feature/Aspect	NumPy	SciPy
Purpose	Fundamental library for numerical computing	Complements NumPy, providing advanced functions for scientific and engineering applications
Data Structures	Provides basic arrays and operations	Offers additional functionality and specialized tools for various scientific tasks
Mathematical Functions	Basic mathematical functions and operations	Extensive collection of special mathematical functions, integration, optimization, and more
Linear Algebra	Basic linear algebra operations	Advanced linear algebra operations, including sparse matrices and eigenvalue problems
Statistics	Limited statistical functions	Comprehensive statistical tools, hypothesis testing, and probability distributions
Optimization	Limited optimization capabilities	Robust optimization routines for solving mathematical and engineering problems

Question 1: Extract data from the given url and do the task mentioned below [Link](#) *Convert inches to centimeter.*

Solution:

Question 1

```python
# Import necessary libraries
import pandas as pd
from scipy.constants import inch, centi  # Import constants for unit conversions

# URL of the CSV file containing height data
url = "https://raw.githubusercontent.com/AnudipAE/DANLC/master/people_heights.csv"

# Read the data from the URL into a Pandas DataFrame
df = pd.read_csv(url)

# Print the original DataFrame to see the data
print(df)

# Convert heights from inches to centimeters
# 'inch' is a constant from scipy.constants which represents 1 inch in meters
# 'centi' is a constant representing 1 centimeter in meters
# The formula multiplies height in inches by the 'inch' constant to convert to meters,
# and then divides by 'centi' to convert from meters to centimeters.
df["Height (cm)"] = ((df["Height (inches)"] * inch) / centi).round(2)

# Print a message and the updated DataFrame with the new "Height (cm)" column
print("Updated data with new Height(cm) column")
print(df)

# Save the updated DataFrame to a new CSV file
df.to_csv("UpdatedHeights.csv", index=False)
```

Output:

```
          Name  Height (inches)
0     Person 1            60.03
1     Person 2            49.51
2     Person 3            82.97
3     Person 4            64.19
4     Person 5            54.42
..         ...              ...
95   Person 96            76.69
96   Person 97            68.06
97   Person 98            57.89
98   Person 99            63.56
99  Person 100            81.85

[100 rows x 2 columns]
```

```
Updated data with new Height(cm) column
          Name  Height (inches)  Height (cm)
0     Person 1            60.03       152.48
1     Person 2            49.51       125.76
2     Person 3            82.97       210.74
3     Person 4            64.19       163.04
4     Person 5            54.42       138.23
..         ...              ...          ...
95   Person 96            76.69       194.79
96   Person 97            68.06       172.87
97   Person 98            57.89       147.04
98   Person 99            63.56       161.44
99  Person 100            81.85       207.90

[100 rows x 3 columns]
```

Question 2: Extract data from the given url and do the task mentioned below [Link](#) *Convert Giga Byte to Mega Byte*.

Solution:

Question 2

```
# Import necessary libraries
import pandas as pd
from scipy.constants import mega, giga  # Import constants for unit conversions

# URL of the CSV file containing file size data
url = "https://raw.githubusercontent.com/AnudipAE/DANLC/master/file_size.csv"

# Read the data from the URL into a Pandas DataFrame
df = pd.read_csv(url)

# Print the original DataFrame to see the data
print(df)

# Convert file sizes from GB to MB
# 'giga' is a constant from scipy.constants which represents 1 gigabyte in bytes
# 'mega' is a constant representing 1 megabyte in bytes
# The formula multiplies the file size in gigabytes by the 'giga' constant to convert to
bytes,
# and then divides by 'mega' to convert from bytes to megabytes.
df["Size(MB)"] = (df["Size (GB)"] * giga / mega)

# Print a message and the updated DataFrame with the new "Size(MB)" column
print("Updated data with new Size(MB) column")
print(df)

# Save the updated DataFrame to a new CSV file
df.to_csv("UpdatedSize.csv", index=False)
```

Output:

```
      Filename  Size (GB)
0     file_1.txt       9.72
1     file_2.txt       9.81
2     file_3.txt       5.61
3     file_4.txt       4.58
4     file_5.txt       5.52
..           ...        ...
95   file_96.txt       1.29
96   file_97.txt       7.11
97   file_98.txt       4.86
98   file_99.txt       7.89
99  file_100.txt       5.52

[100 rows x 2 columns]
```

```
Updated data with new Size(MB) column
      Filename  Size (GB)  Size(MB)
0     file_1.txt       9.72    9720.0
1     file_2.txt       9.81    9810.0
2     file_3.txt       5.61    5610.0
3     file_4.txt       4.58    4580.0
4     file_5.txt       5.52    5520.0
..           ...        ...       ...
95   file_96.txt       1.29    1290.0
96   file_97.txt       7.11    7110.0
97   file_98.txt       4.86    4860.0
98   file_99.txt       7.89    7890.0
99  file_100.txt       5.52    5520.0

[100 rows x 3 columns]
```