Lab – 28

Student Name: Navneet P

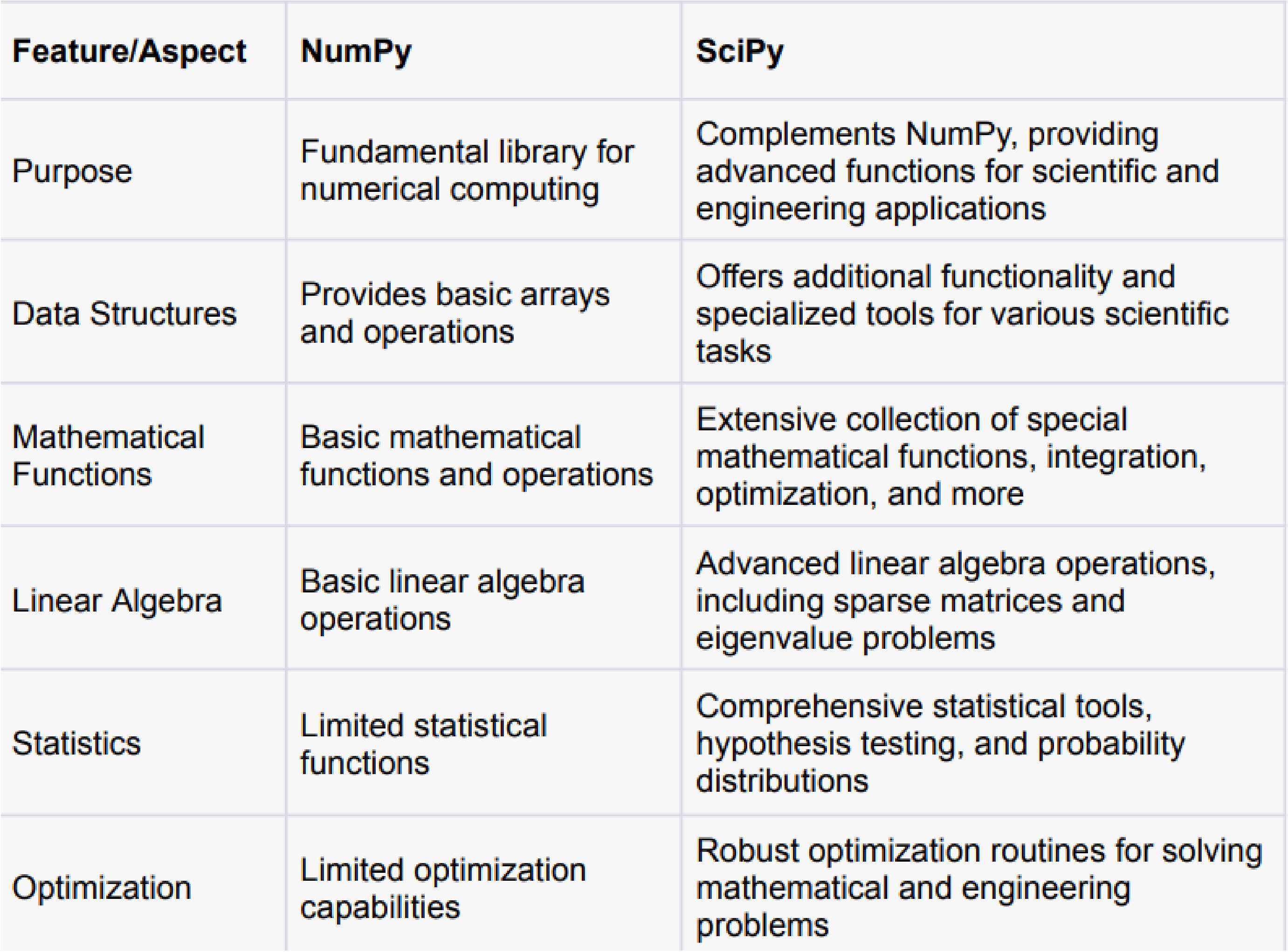
Student ID: AF0411619

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**.

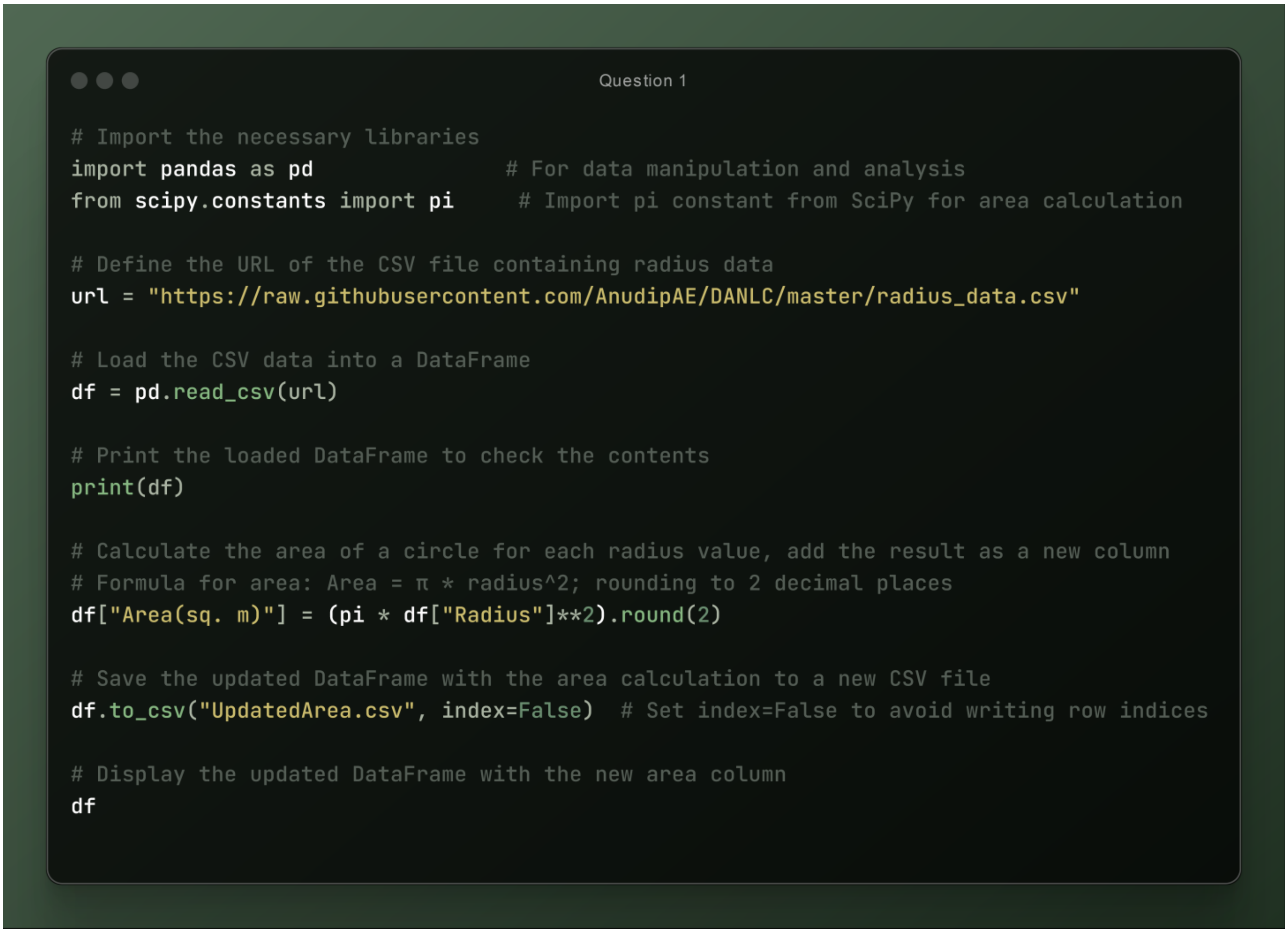
SciPy vs NumPy



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

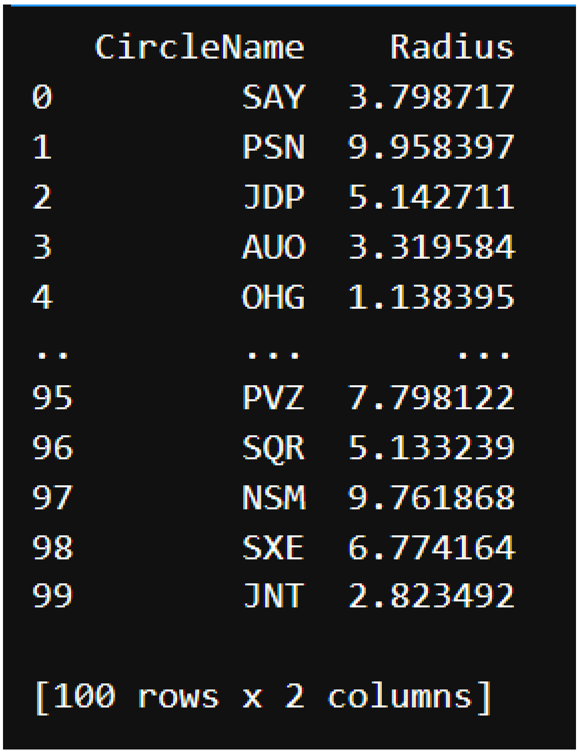
Question 1: Extract data from the given url and do the task mentioned below Link *Add a calculated area of circle column.*

Solution:

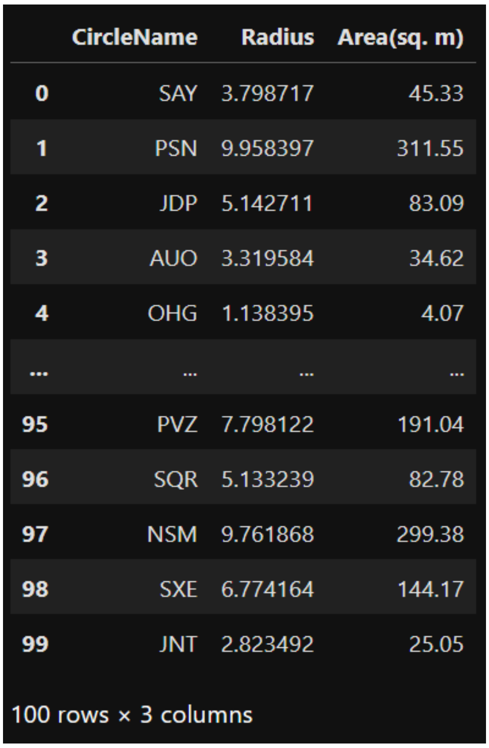


```python
# Import the necessary libraries
import pandas as pd               # For data manipulation and analysis
from scipy.constants import pi     # Import pi constant from SciPy for area calculation

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

# Load the CSV data into a DataFrame
df = pd.read_csv(url)

# Print the loaded DataFrame to check the contents
print(df)

# Calculate the area of a circle for each radius value, add the result as a new column
# Formula for area: Area = π * radius^2; rounding to 2 decimal places
df["Area(sq. m)"] = (pi * df["Radius"]**2).round(2)

# Save the updated DataFrame with the area calculation to a new CSV file
df.to_csv("UpdatedArea.csv", index=False)  # Set index=False to avoid writing row indices

# Display the updated DataFrame with the new area column
df
```

Output:



```
   CircleName    Radius
0         SAY  3.798717
1         PSN  9.958397
2         JDP  5.142711
3         AUO  3.319584
4         OHG  1.138395
..        ...       ...
95        PVZ  7.798122
96        SQR  5.133239
97        NSM  9.761868
98        SXE  6.774164
99        JNT  2.823492

[100 rows x 2 columns]
```

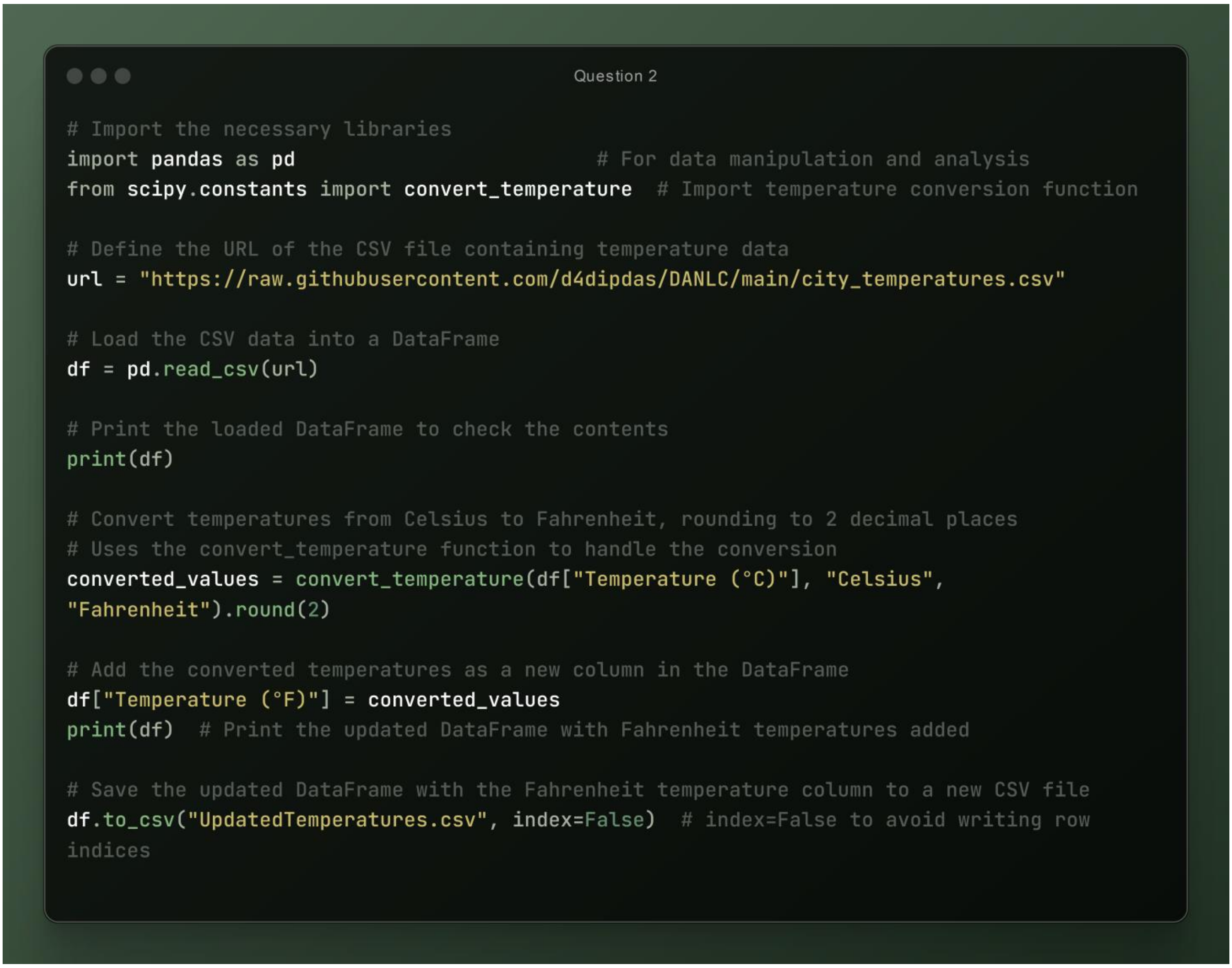


	CircleName	Radius	Area(sq. m)
0	SAY	3.798717	45.33
1	PSN	9.958397	311.55
2	JDP	5.142711	83.09
3	AUO	3.319584	34.62
4	OHG	1.138395	4.07
...	...	...	...
95	PVZ	7.798122	191.04
96	SQR	5.133239	82.78
97	NSM	9.761868	299.38
98	SXE	6.774164	144.17
99	JNT	2.823492	25.05

100 rows × 3 columns

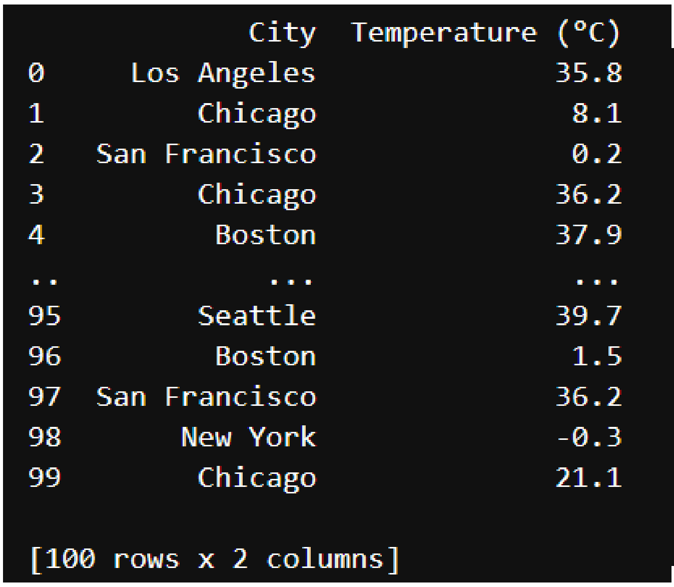
Question 2: Extract data from the given url and do the task mentioned below Link *Add a column with temperatures converted to Fahrenheit*

Solution:

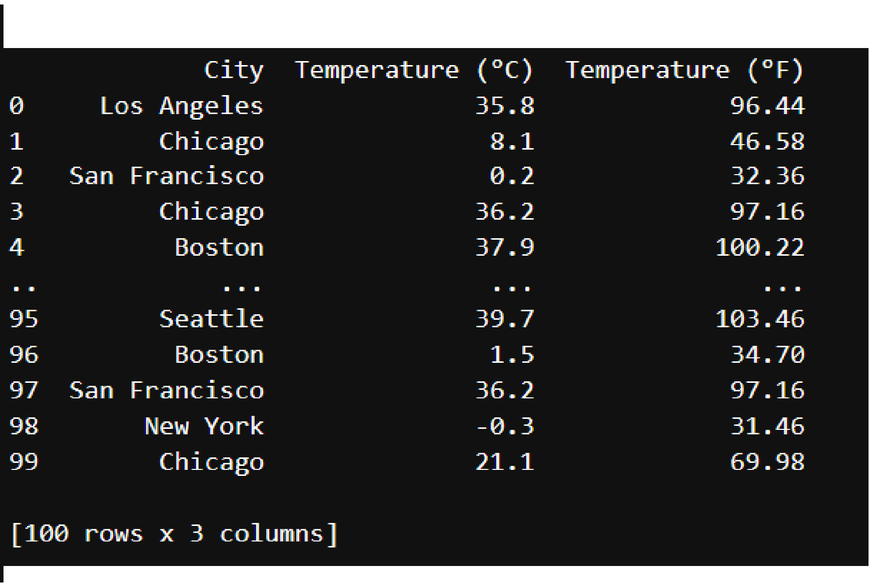


```
# Import the necessary libraries
import pandas as pd                          # For data manipulation and analysis
from scipy.constants import convert_temperature  # Import temperature conversion function

# Define the URL of the CSV file containing temperature data
url = "https://raw.githubusercontent.com/d4dipdas/DANLC/main/city_temperatures.csv"

# Load the CSV data into a DataFrame
df = pd.read_csv(url)

# Print the loaded DataFrame to check the contents
print(df)

# Convert temperatures from Celsius to Fahrenheit, rounding to 2 decimal places
# Uses the convert_temperature function to handle the conversion
converted_values = convert_temperature(df["Temperature (°C)"], "Celsius",
"Fahrenheit").round(2)

# Add the converted temperatures as a new column in the DataFrame
df["Temperature (°F)"] = converted_values
print(df)  # Print the updated DataFrame with Fahrenheit temperatures added

# Save the updated DataFrame with the Fahrenheit temperature column to a new CSV file
df.to_csv("UpdatedTemperatures.csv", index=False)  # index=False to avoid writing row
indices
```

Output:



```
             City  Temperature (°C)
0     Los Angeles              35.8
1         Chicago               8.1
2   San Francisco               0.2
3         Chicago              36.2
4          Boston              37.9
..            ...               ...
95        Seattle              39.7
96         Boston               1.5
97  San Francisco              36.2
98       New York              -0.3
99        Chicago              21.1

[100 rows x 2 columns]
```



```
             City  Temperature (°C)  Temperature (°F)
0     Los Angeles              35.8             96.44
1         Chicago               8.1             46.58
2   San Francisco               0.2             32.36
3         Chicago              36.2             97.16
4          Boston              37.9            100.22
..            ...               ...               ...
95        Seattle              39.7            103.46
96         Boston               1.5             34.70
97  San Francisco              36.2             97.16
98       New York              -0.3             31.46
99        Chicago              21.1             69.98

[100 rows x 3 columns]
```