



# **“Python Programming”**

## **Assignment-4**

**Topic** - Data Analysis and Visualisation with RealWorld  
Weather Data

**Submitted by** - Priyam Sharma

**Roll no** - 2501730184

**Course** - B. Tech CSE (AI & ML)

**Section** - B

**Faculty name**- Mr Sameer Farooq

## **Introduction**

Weather conditions play a vital role in agriculture, transportation, health, and environmental planning. With data analysis tools like Python, NumPy, and Pandas, as well as visualisation libraries, we can convert raw datasets into meaningful insights. This project focuses on weather data analysis using data cleaning, statistical computation, and visual representation.

## **Objectives**

- To clean and preprocess the raw weather dataset by handling missing values and formatting dates.
- To calculate important weather statistics such as mean, maximum, minimum temperature, total rainfall, and humidity patterns.
- To generate monthly weather summaries for comparative analysis.
- To visualise weather trends using line plots, bar charts, and scatter plots.
- To export analytical results into files for further interpretation (summary CSV and text report).

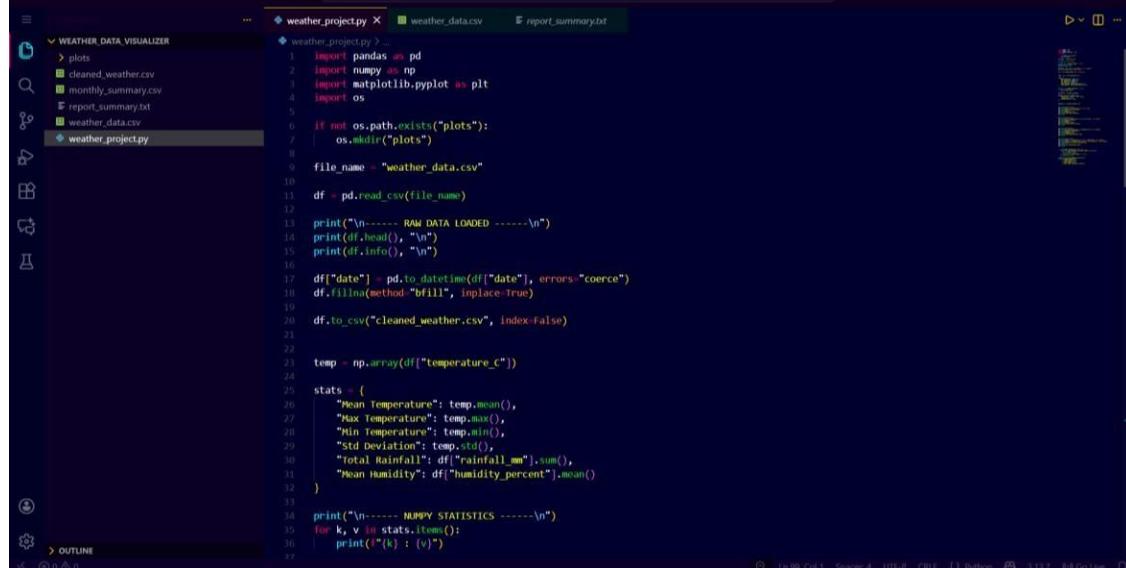
## **Program Description**

This project processes and analyses weather data stored in a CSV file using Python. It performs data cleaning, calculates essential statistical values, groups data every month, and visualises important weather patterns.

The script reads the dataset, converts date formats, handles missing values, and generates a cleaned file for further use. The

project also creates monthly summaries including temperature trends, rainfall distribution, humidity vs. temperature scatter, and combined weather plots. Results are exported as files like CSV summaries, graphs, and a final report.

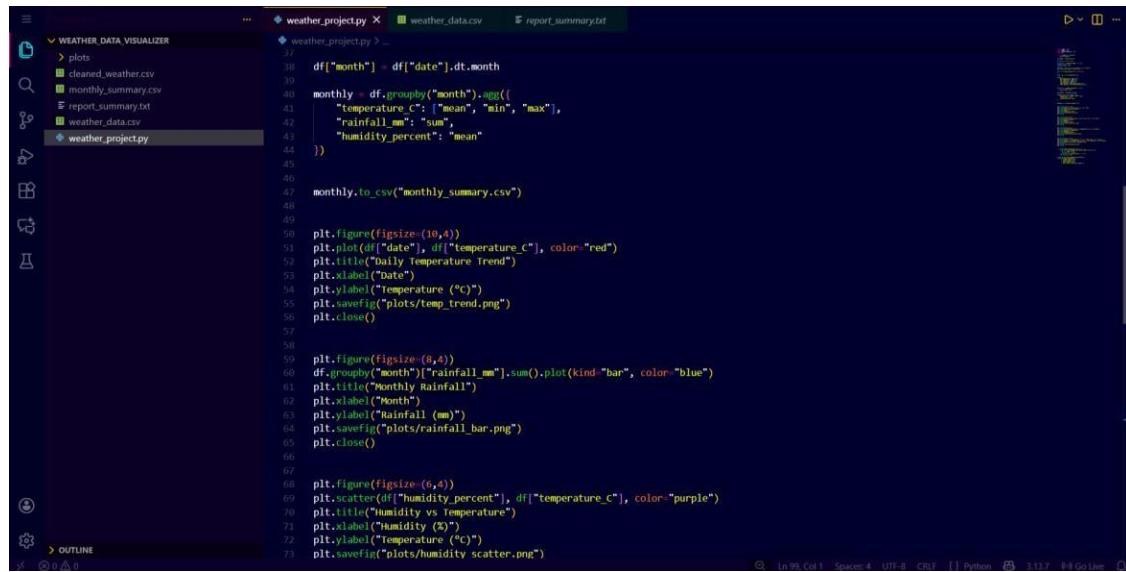
## Program Code



```

weather_project.py > ...
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import os
5
6 if not os.path.exists("plots"):
7     os.mkdir("plots")
8
9 file_name = "weather_data.csv"
10
11 df = pd.read_csv(file_name)
12
13 print("\n----- RAW DATA LOADED -----")
14 print(df.head(), "\n")
15 print(df.info(), "\n")
16
17 df["date"] = pd.to_datetime(df["date"], errors="coerce")
18 df.fillna(method="ffill", inplace=True)
19
20 df.to_csv("cleaned_weather.csv", index=False)
21
22
23 temp = np.array(df["temperature_c"])
24
25 stats = {
26     "Mean Temperature": temp.mean(),
27     "Max Temperature": temp.max(),
28     "Min Temperature": temp.min(),
29     "Std Deviation": temp.std(),
30     "Total Rainfall": df["rainfall_mm"].sum(),
31     "Mean Humidity": df["humidity_percent"].mean()
32 }
33
34 print("\n----- NUMPY STATISTICS -----")
35 for k, v in stats.items():
36     print(f"(k) : {v}")
37

```



```

weather_project.py > ...
38 df["month"] = df["date"].dt.month
39
40 monthly = df.groupby("month").agg([
41     {"temperature_c": ["mean", "min", "max"],
42      "rainfall_mm": "sum",
43      "humidity_percent": "mean"})
44
45
46 monthly.to_csv("monthly_summary.csv")
47
48
49 plt.figure(figsize=(10,4))
50 plt.plot(df["date"], df["temperature_c"], color="red")
51 plt.title("Daily Temperature Trend")
52 plt.xlabel("Date")
53 plt.ylabel("Temperature (°C)")
54 plt.savefig("plots/temp_trend.png")
55 plt.close()
56
57
58 plt.figure(figsize=(8,4))
59 df.groupby("month")["rainfall_mm"].sum().plot(kind="bar", color="blue")
60 plt.title("Monthly Rainfall")
61 plt.xlabel("Month")
62 plt.ylabel("Rainfall (mm)")
63 plt.savefig("plots/rainfall_bar.png")
64 plt.close()
65
66
67 plt.figure(figsize=(6,4))
68 plt.scatter(df["humidity_percent"], df["temperature_c"], color="purple")
69 plt.title("Humidity vs Temperature")
70 plt.xlabel("Humidity (%)")
71 plt.ylabel("Temperature (°C)")
72 plt.savefig("plots/humidity_scatter.png")
73 plt.close()

```

The screenshot shows a Jupyter Notebook environment with several open cells. The current cell is active and displays Python code for weather data analysis. The code includes plotting rainfall and temperature, generating a combined plot, and writing a summary report to a file. The notebook also lists files like cleaned\_weather.csv and monthly\_summary.csv.

```
weather_project.py
```

```
# weather_project.py
# weather project code

# imports
import pandas as pd
import matplotlib.pyplot as plt
from datetime import datetime
import numpy as np
import os

# reading data
df = pd.read_csv('weather_data.csv')
stats = df.describe()

# cleaning data
cleaned = df.dropna()
cleaned['date'] = pd.to_datetime(cleaned['date'])

# monthly summary
monthly_summary = cleaned.groupby([pd.Grouper(freq='M'), 'city']).mean().reset_index()

# report summary
report_summary = monthly_summary[['city', 'date', 'temperature_min', 'temperature_max', 'humidity_min', 'humidity_max', 'rainfall_mm']].groupby('city').mean().reset_index()

# saving data
cleaned.to_csv('cleaned_weather.csv', index=False)
monthly_summary.to_csv('monthly_summary.csv', index=False)
report_summary.to_csv('report_summary.txt', index=False)

# plots
plt.scatter(df['humidity_percent'], df['temperature_c'], color='purple')
plt.title("Humidity vs Temperature")
plt.xlabel("Humidity (%)")
plt.ylabel("Temperature (°C)")
plt.savefig('plots/humidity_scatter.png')
plt.close()

plt.figure(figsize=(10,5))
plt.plot(df['date'], df['temperature_c'], label="temperature", color="green")
plt.plot(df['date'], df['rainfall_mm'], alpha=0.4, label="rainfall", color="orange")
plt.title("Temperature & Rainfall Combined Plot")
plt.legend()
plt.savefig('plots/combined_plot.png')
plt.close()

with open("report_summary.txt", "w") as f:
    f.write("===== WEATHER DATA ANALYSIS REPORT =====\n")
    for k, v in stats.items():
        f.write(f"\t{k} : {v}\n")
    f.write("\n----- Monthly Summary ----- \n")
    f.write(str(monthly_summary))

print("All tasks completed successfully!")
print("Outputs generated:")
print("- cleaned_weather.csv\n"
      "- monthly_summary.csv\n"
      "- report_summary.txt\n"
      "- plots folder containing 4 graphs\n")
```

The screenshot shows a Jupyter Notebook interface with the following details:

- Left Sidebar:** Contains icons for file operations (New, Open, Save, etc.), a search bar, and a "WEATHER\_DATA\_VISUALIZER" section with files: plots, cleaned\_weather.csv, monthly\_summary.csv, report\_summary.txt, weather\_data.csv, and weather\_project.py.
- Top Bar:** Shows tabs for weather\_project.py, weather\_data.csv, and report\_summary.txt.
- Code Cell:** Displays the content of the weather\_data.csv file, which contains 177 rows of weather data with columns: date, temperature\_C, rainfall\_mm, and humidity\_percent. The data spans from January 1, 2024, to December 31, 2024.
- Bottom Bar:** Includes buttons for Run, Kernel, Help, and a Go Live button.

## Sample Output

▼ WEATHER\_DATA\_VISUALIZER

- plots
- cleaned\_weather.csv
- monthly\_summary.csv
- report\_summary.txt
- weather\_data.csv
- weather\_project.py

PS C:\Users\vip\Desktop\python programming\weather\_data\_visualizer> python weather\_project.py

----- RAW DATA LOADED -----

	date	temperature_C	rainfall_mm	humidity_percent
0	2024-01-01	14	2.5	73
1	2024-01-02	15	0.0	69
2	2024-01-03	18	3.2	73
3	2024-01-04	13	0.0	75
4	2024-01-05	12	4.3	76

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 51 entries, 0 to 50

Data columns (total 4 columns):

#	Column	Non-Null Count	Dtype	
0	date	51	non-null	object
1	temperature_C	51	non-null	int64
2	rainfall_mm	51	non-null	float64
3	humidity_percent	51	non-null	int64

dtypes: float64(1), int64(2), object(1)

memory usage: 1.74 KB

None

C:\Users\vip\Desktop\python programming\weather\_data\_visualizer>weather\_project.py:31: FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a future version. use obj.fff1ll() or obj.bfill() instead.  
df.fillna(method="bf1ll", inplace=True) # handle missing values

----- NUMPY STATISTICS -----

Mean Temperature : 23.80392156862745  
Max Temperature : 34  
Min Temperature : 12  
Std Deviation : 5.688012182705683  
Total Rainfall : 37.8  
Mean Humidity : 51.64705882352941

✓ All tasks completed successfully!

```

>>> df.fillna(method="bfill", inplace=True)                                # handle missing values
----- Numpy Statistics -----
Mean Temperature : 23.80392156862745
Max Temperature : 34
Min Temperature : 12
Std Deviation : 5.688032182705683
Total Rainfall : 37.8
Mean Humidity : 51.64705882352941

✓ All tasks completed successfully!
↳ Outputs generated:
- cleaned_weather.csv
- monthly_summary.csv
- report_summary.txt
- plots folder containing 4 graphs

PS C:\Users\hp\Desktop\python programming\weather_data_visualizer>

```

```

=====
WEATHER DATA ANALYSIS REPORT =====
1
2
3 Mean Temperature : 23.80392156862745
4 Max Temperature : 34
5 Min Temperature : 12
6 Std Deviation : 5.688032182705683
7 Total Rainfall : 37.8
8 Mean Humidity : 51.64705882352941
9
10 ----- Monthly Summary -----
11   temperature_c      rainfall_mm humidity_percent
12   mean min max      sum      mean
13   month
14   1     21.354839 12 31    26.7    54.322581
15   2     27.600000 20 34    11.1    47.500000

```

## Conclusion

This project successfully analyses weather data by cleaning the dataset, calculating key statistics, and visualising temperature, rainfall, and humidity trends. It shows how Python can effectively convert raw climate data into useful insights for weather monitoring and decision-making.