WEATHER DATA ANALYSIS

SUBJECT: *PROGRAMMING FOR PROBLEM SOLVING USING PYTHON*

SUBMITTED TO: *DR. SAMEER FAROOQ*

COURSE CODE: *ETCCPP102*

MADE BY: *SONAM YADAV*

CLASS: *B.TECH CSE (AI ML)*

SECTION: *B*

ROLL NO: *2501730260*

INTRODUCTION:

This project focuses on analyzing real-world weather data using Python. Weather datasets usually contain daily records such as temperature, humidity, and rainfall. However, these datasets often include missing values, incorrect formats, and inconsistent data.

The goal of this project was to load the dataset, clean it, analyze important parameters, and create visualizations that help understand weather patterns. For this, I used Python libraries like **Pandas** for data handling, **NumPy** for calculations, and **Matplotlib** for plotting graphs.

This project helped me understand how data is processed in real data-science workflows and how Python is used in weather-based analysis.

OBJECTIVES:

- ✓ To load and explore the weather dataset using Pandas.
- ✓ To clean missing or invalid values and format the data properly
- ✓ To compute essential statistics such as mean, min, max, and standard deviation.
- ✓ To visualize weather trends using different charts and export the results.

WORKING AND DESCRIPTION

Below is the description of the main files and how the program works:

1. **weather_data.csv (Input File)**
   This file contains the raw Dweather dataset that includes fields such as:
   - **date**
   - **temperature**
   - **rainfall**
   - **humidity**

   The program reads this file and perform all analysis on this dataset.

2. **Data Loading & Initial Checks**
   The program begins by loading the CSV file using:

   data=pd.read_csv("weather_data.csv")

   If the file is missing, the program shows an error message and stops. After successful loading, the following details are displayed:
   - **First 5 rows of data**
   - **Dataset structure(info)**
   - **Statistical summary(describe())**

3. **Data Cleaning**
   This part prepare the raw data for analysis.

❖**Date Conversion**

The "date" column is converted into proper datetime format.

❖**Handling Missing Values**

Numeric columns such as:

- Temperature
- Rainfaill
- Humidity

are filled with their mean values.

❖**Column Selection**

Only the important columns are kept:

- Date
- Temperature
- Rainfall
- Humidity

❖**Month Extraction**

A new column month is created using:

data["month"]= data["date"].dt.month

4. Statistical Calculations

Numpy is used to calculate important temperature statistics:

- **Mean temperature**
- **Minimum temperature**
- **Maximum temperature**

- **Standard deviation**

These help understand the temperature variation in the dataset.

5. Data Visualization (Charts & Graphs)

The program generates several graphs using Matplotlib:

a. Daily Temperature Line Chart

Shows how temperature changes throughout the recorded days.

b. Monthly Rainfall Bar Chart

Displays total rainfall for each month.

c. Humidity vs Temperature Scatter Plot

Helps analyse the relationship between humidity and temperature.

d. Combined Plot (Subplots)

Includes temperature trend and humidity scatter together

All charts are saved as PNG files:

- daily_temperature.png
- monthly_rainfall.png
- temp_vs_humidity.png
- combined_plot.png

6. Monthly Summary Table

Using:

data.groupby("month").agg(...)

this program generates a summary that includes:

- mean, min, max temperature
- total rainfall
- average humidity

This summary helps identify weather patterns month-wise.

7. Exporting Results

The program produces two output files:

- **cleaned_weather_data.csv**

  Contains the cleaned and processed dataset.

- **summary_report.txt**

  Contains:

  - Monthly summary table
  - Key observations
  - Notes on temperature, rainfall, and humidity patterns

Finally, the program prints a message confirming completion.

OUTPUT SCREENSHOTS:

1. **RAW DATA HEAD**

Python Lab 4 > weather_data.csv > data

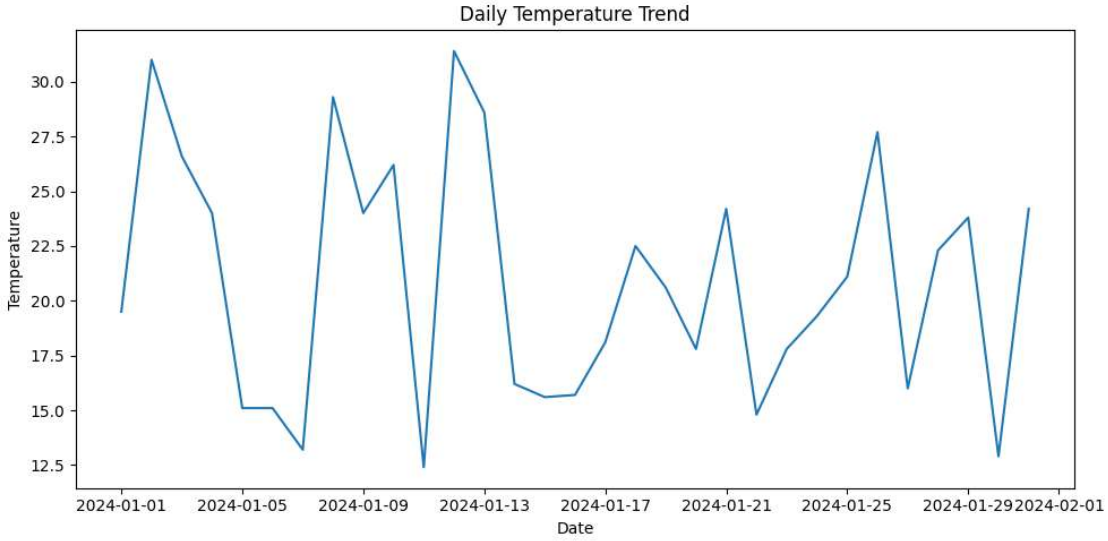
```
date,temperature,rainfall,humidity
2024-01-01,19.5,3.4,81.4
2024-01-02,31.0,1.3,57.8
2024-01-03,26.6,19.0,54.0
2024-01-04,24.0,19.3,67.1
2024-01-05,15.1,16.2,47.0
2024-01-06,15.1,6.1,80.1
2024-01-07,13.2,2.0,43.7
2024-01-08,29.3,13.7,89.3
2024-01-09,24.0,8.8,78.6
2024-01-10,26.2,2.4,49.9
2024-01-11,12.4,9.9,40.3
2024-01-12,31.4,0.7,80.8
2024-01-13,28.6,18.2,75.3
2024-01-14,16.2,5.2,76.5
2024-01-15,15.6,13.3,78.6
2024-01-16,15.7,6.2,43.7
2024-01-17,18.1,10.4,57.9
2024-01-18,22.5,10.9,45.8
2024-01-19,20.6,3.7,83.2
2024-01-20,17.8,19.4,71.2
2024-01-21,24.2,15.5,56.5
2024-01-22,14.8,18.8,43.2
2024-01-23,17.8,17.9,55.5
2024-01-24,19.3,12.0,56.3
2024-01-25,21.1,18.4,76.5
2024-01-26,27.7,1.8,71.9
2024-01-27,16.0,3.9,84.4
2024-01-28,22.3,0.9,63.6
2024-01-29,23.8,6.5,46.0
2024-01-30,12.9,7.8,75.7
2024-01-31,24.2,5.4,78.0
```

2. Cleaned Data Preview

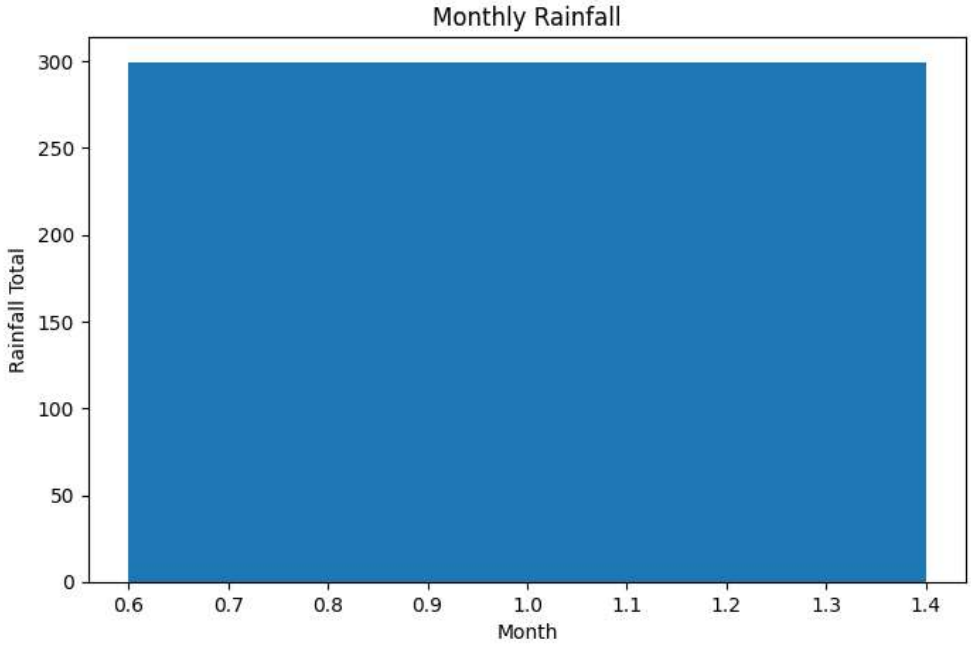
Python Lab 4 > cleaned_weather_data.csv > data

```
date,temperature,rainfall,humidity,month
2024-01-01,19.5,3.4,81.4,1
2024-01-02,31.0,1.3,57.8,1
2024-01-03,26.6,19.0,54.0,1
2024-01-04,24.0,19.3,67.1,1
2024-01-05,15.1,16.2,47.0,1
2024-01-06,15.1,6.1,80.1,1
2024-01-07,13.2,2.0,43.7,1
2024-01-08,29.3,13.7,89.3,1
2024-01-09,24.0,8.8,78.6,1
2024-01-10,26.2,2.4,49.9,1
2024-01-11,12.4,9.9,40.3,1
2024-01-12,31.4,0.7,80.8,1
2024-01-13,28.6,18.2,75.3,1
2024-01-14,16.2,5.2,76.5,1
2024-01-15,15.6,13.3,78.6,1
2024-01-16,15.7,6.2,43.7,1
2024-01-17,18.1,10.4,57.9,1
2024-01-18,22.5,10.9,45.8,1
2024-01-19,20.6,3.7,83.2,1
2024-01-20,17.8,19.4,71.2,1
2024-01-21,24.2,15.5,56.5,1
2024-01-22,14.8,18.8,43.2,1
2024-01-23,17.8,17.9,55.5,1
2024-01-24,19.3,12.0,56.3,1
2024-01-25,21.1,18.4,76.5,1
2024-01-26,27.7,1.8,71.9,1
2024-01-27,16.0,3.9,84.4,1
2024-01-28,22.3,0.9,63.6,1
2024-01-29,23.8,6.5,46.0,1
2024-01-30,12.9,7.8,75.7,1
2024-01-31,24.2,5.4,78.0,1
```

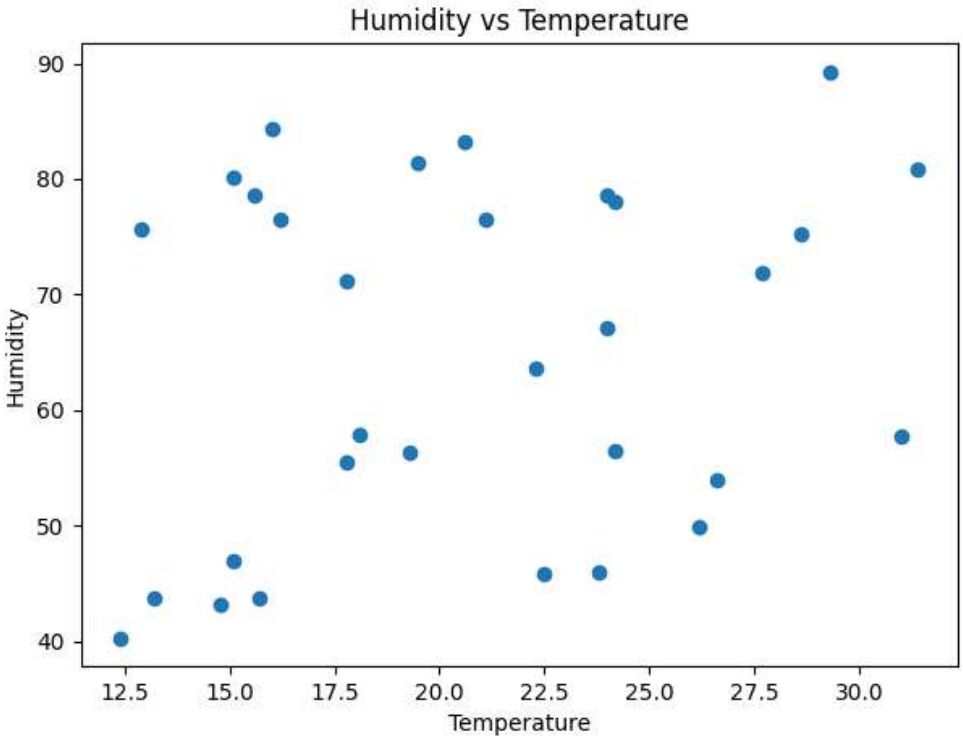
3. Daily Temperature Plot



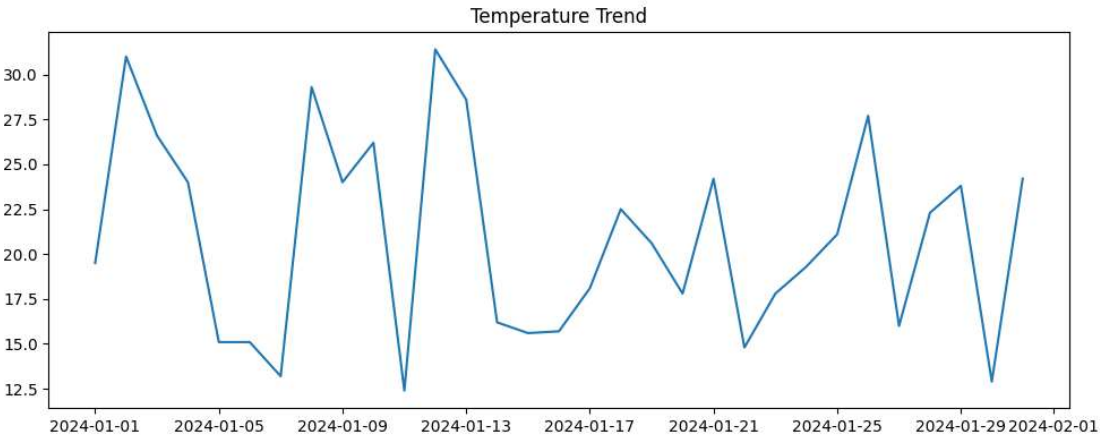
4. Monthly Rainfall Bar Chart

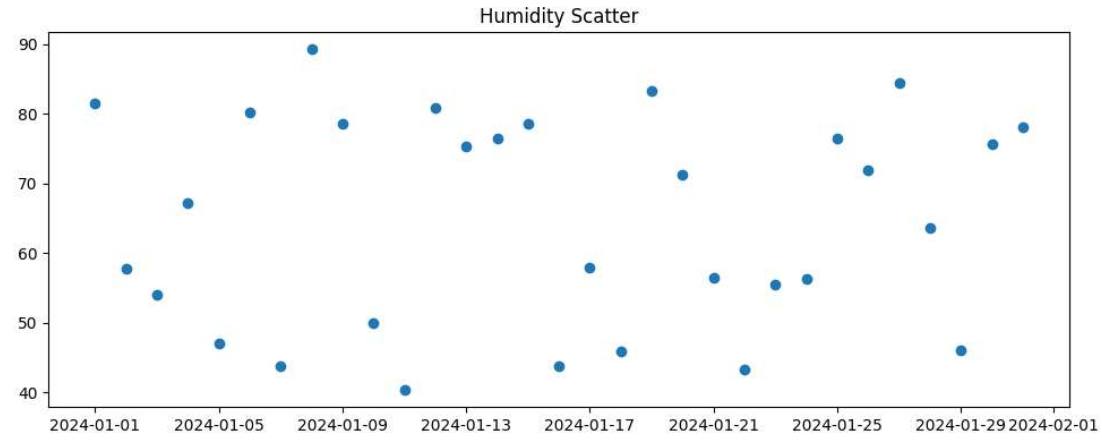


5. Humidity vs Temperature Scatter Plot



6. Combined Plot





7. Cleaned CSV & Summary File Generated

Python Lab 4 > summary_report.txt

```
Weather Data Summary Report
--------------------------
      temperature             rainfall   humidity
             mean   min   max      sum       mean
month
1       20.870968  12.4  31.4    299.0  64.832258

Key Observations:
- Temperature varies across months.
- Rainfall data shows seasonal patterns.
- Humidity relation can be checked in scatter plot.

```

REFLECTION:

Working on this project was a good learning experience for me. Before this, I mostly worked with simple Python programs, but here I got to use data-science libraries in a practical way. At first, I faced issues with missing values and date conversion, but after trying different methods,

I understood how important data cleaning is. I also realized how helpful graphs are in understanding patterns that are not easily visible in raw numbers. This project has encouraged me to explore more topics in data analysis and maybe even try machine-learning-based predictions in the future.

CONCLUSION:

This project gave me hands-on experience in data cleaning and weather analysis. I learned how missing values and inconsistent formats affect results and how to fix them using Pandas. I also understood how to compute statistical measures and visualize data using Matplotlib.

The final output clearly shows seasonal variations in temperature, rainfall, and humidity. Such analysis can be used in agricultural planning, environmental studies, and weather prediction projects. Overall, this project improved my skills in handling real-world datasets.

