```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Load dataset
df = pd.read_csv("/content/weather.csv")

# Display basic info
print(df.head())
print(df.info())

# Convert Date column to datetime
df['Date'] = pd.to_datetime(df['Date'])

# Extract Month and Year
df['Month'] = df['Date'].dt.month
df['Year'] = df['Date'].dt.year

# ----------------------------------------
# 1️⃣ Temperature Trend Analysis
# ----------------------------------------
plt.figure(figsize=(10,5))
plt.plot(df['Date'], df['Temp'], label='Temperature')
plt.title("Temperature Trend Over Time")
plt.xlabel("Date")
plt.ylabel("Temperature (°C)")
plt.legend()
plt.show()

# ----------------------------------------
# 2️⃣ Rainfall Analysis
# ----------------------------------------
plt.figure(figsize=(8,5))
plt.bar(df['Date'], df['Rainfall'])
plt.title("Rainfall Over Time")
plt.xlabel("Date")
plt.ylabel("Rainfall (mm)")
plt.show()

# ----------------------------------------
# 3️⃣ Humidity Analysis
# ----------------------------------------
plt.figure(figsize=(8,5))
plt.plot(df['Date'], df['Humidity'], color='green')
plt.title("Humidity Trend Over Time")
plt.xlabel("Date")
plt.ylabel("Humidity (%)")
plt.show()

# ----------------------------------------
# 4️⃣ Monthly Seasonal Comparison
# ----------------------------------------
```

```python
monthly_avg = df.groupby('Month').mean(numeric_only=True)

plt.figure(figsize=(8,5))
plt.plot(monthly_avg.index, monthly_avg['Temp'], marker='o')
plt.title("Average Monthly Temperature")
plt.xlabel("Month")
plt.ylabel("Temperature (°C)")
plt.show()

plt.figure(figsize=(8,5))
plt.plot(monthly_avg.index, monthly_avg['Rainfall'], marker='o')
plt.title("Average Monthly Rainfall")
plt.xlabel("Month")
plt.ylabel("Rainfall (mm)")
plt.show()

plt.figure(figsize=(8,5))
plt.plot(monthly_avg.index, monthly_avg['Humidity'], marker='o')
plt.title("Average Monthly Humidity")
plt.xlabel("Month")
plt.ylabel("Humidity (%)")
plt.show()

# ----------------------------------------
# 5 Summary Statistics
# ----------------------------------------
print("\nSummary Statistics:")
print(df[['Temp', 'Rainfall', 'Humidity']].describe())
```

Output

```
  Date  Temp  Rainfall  Humidity
0  2023-01-01    25      12        60
1  2023-01-02    26       0        61
2  2023-01-03    27       0        62
3  2023-01-04    28       0        63
4  2023-01-05    29       0        64
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Date      365 non-null    object
 1   Temp      365 non-null    int64
 2   Rainfall  365 non-null    int64
 3   Humidity  365 non-null    int64
dtypes: int64(3), object(1)
memory usage: 11.5+ KB
None
```
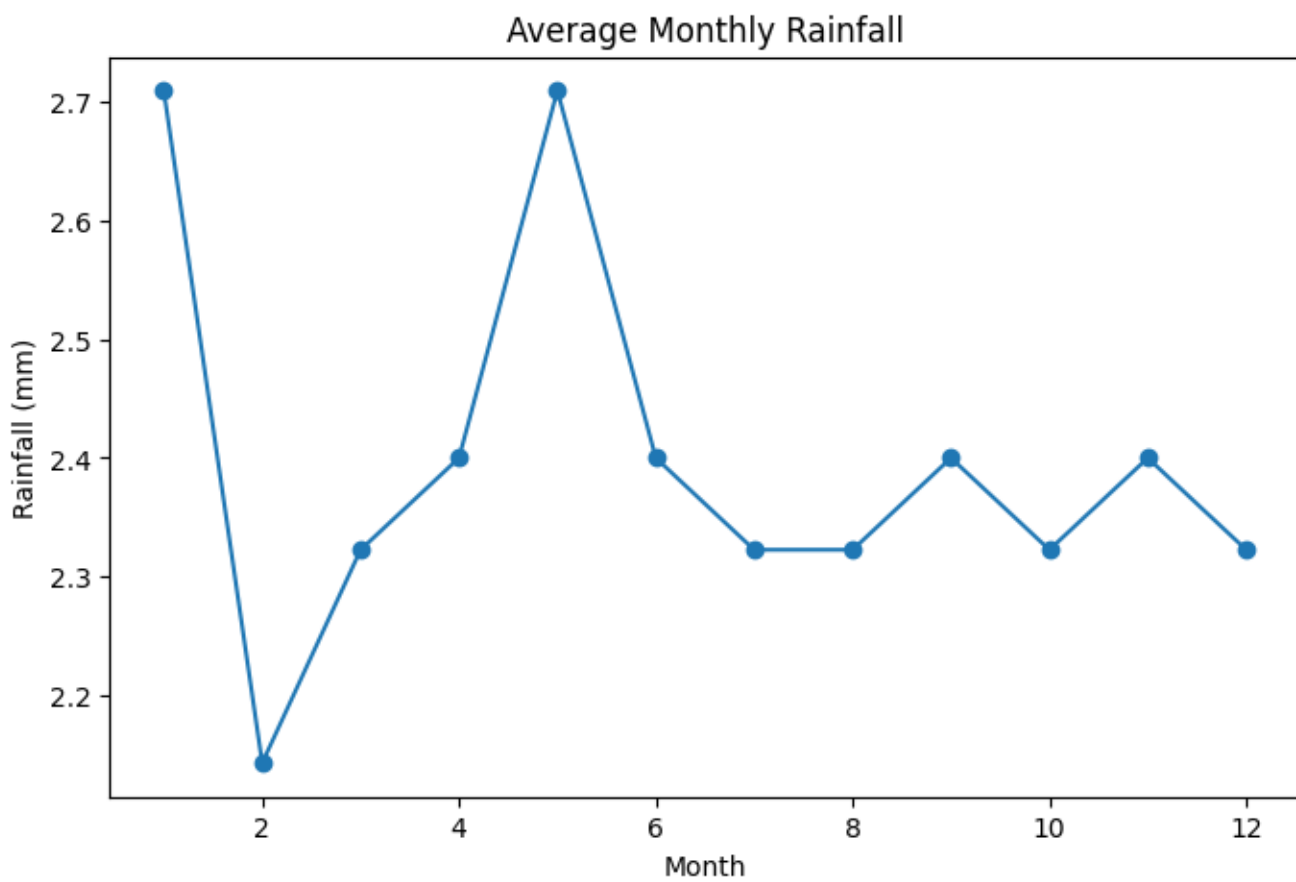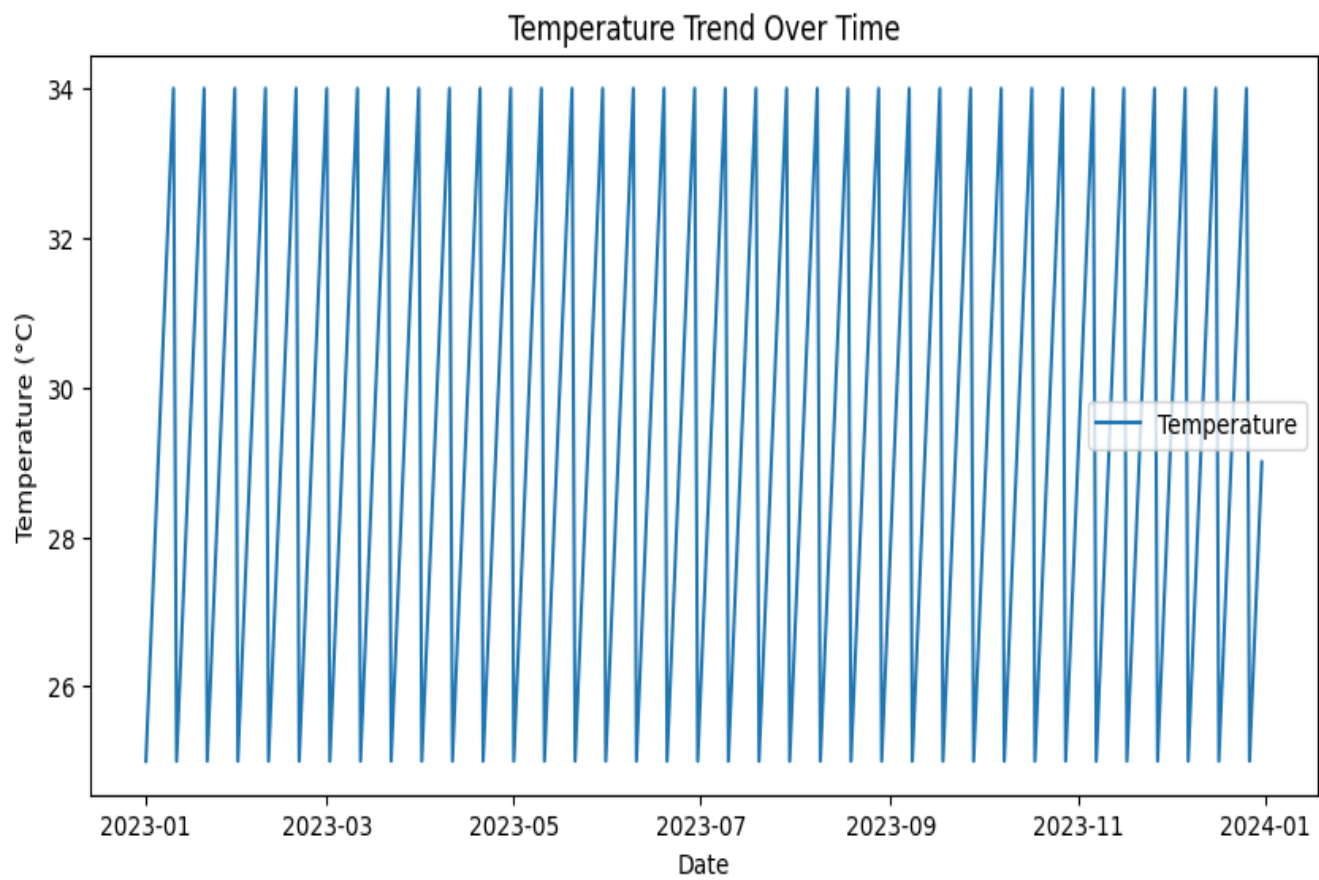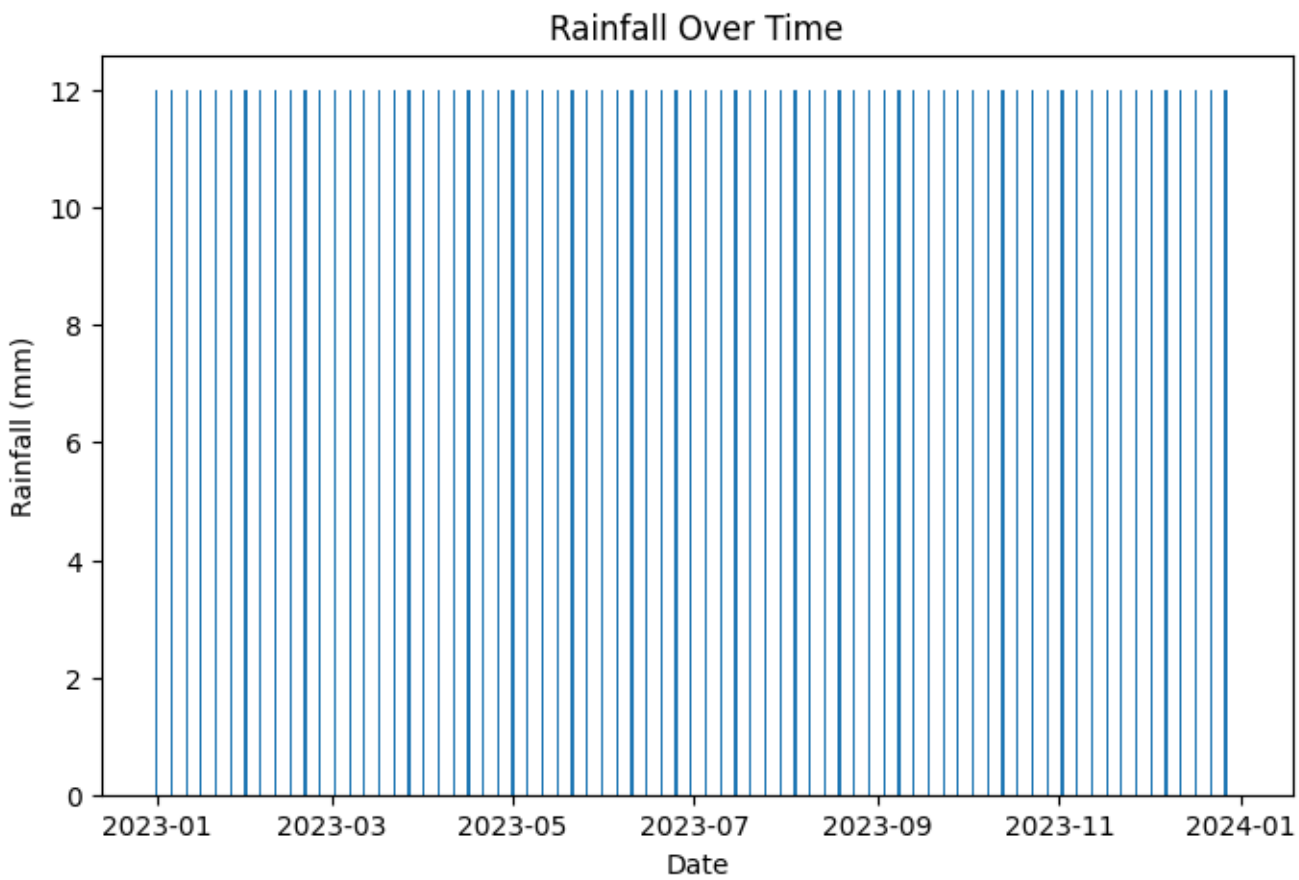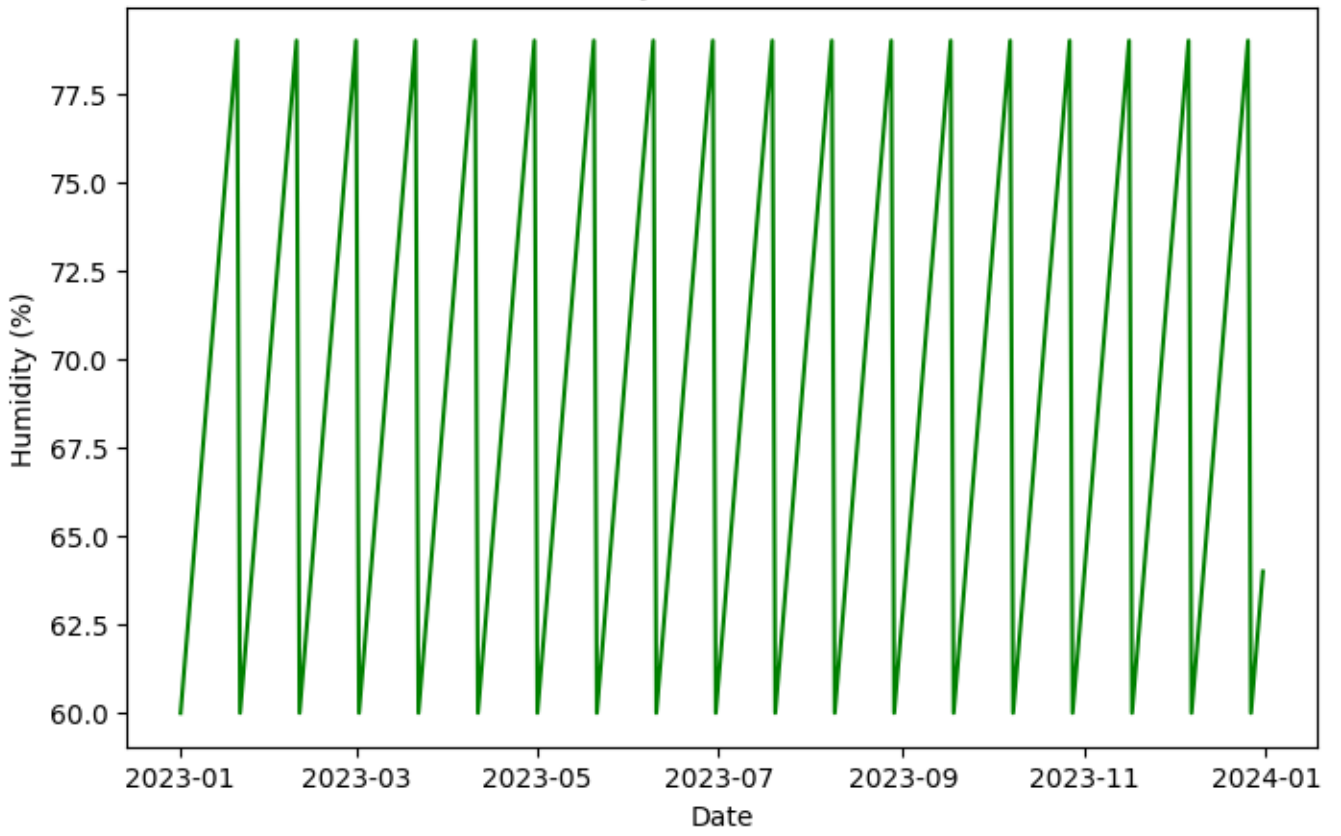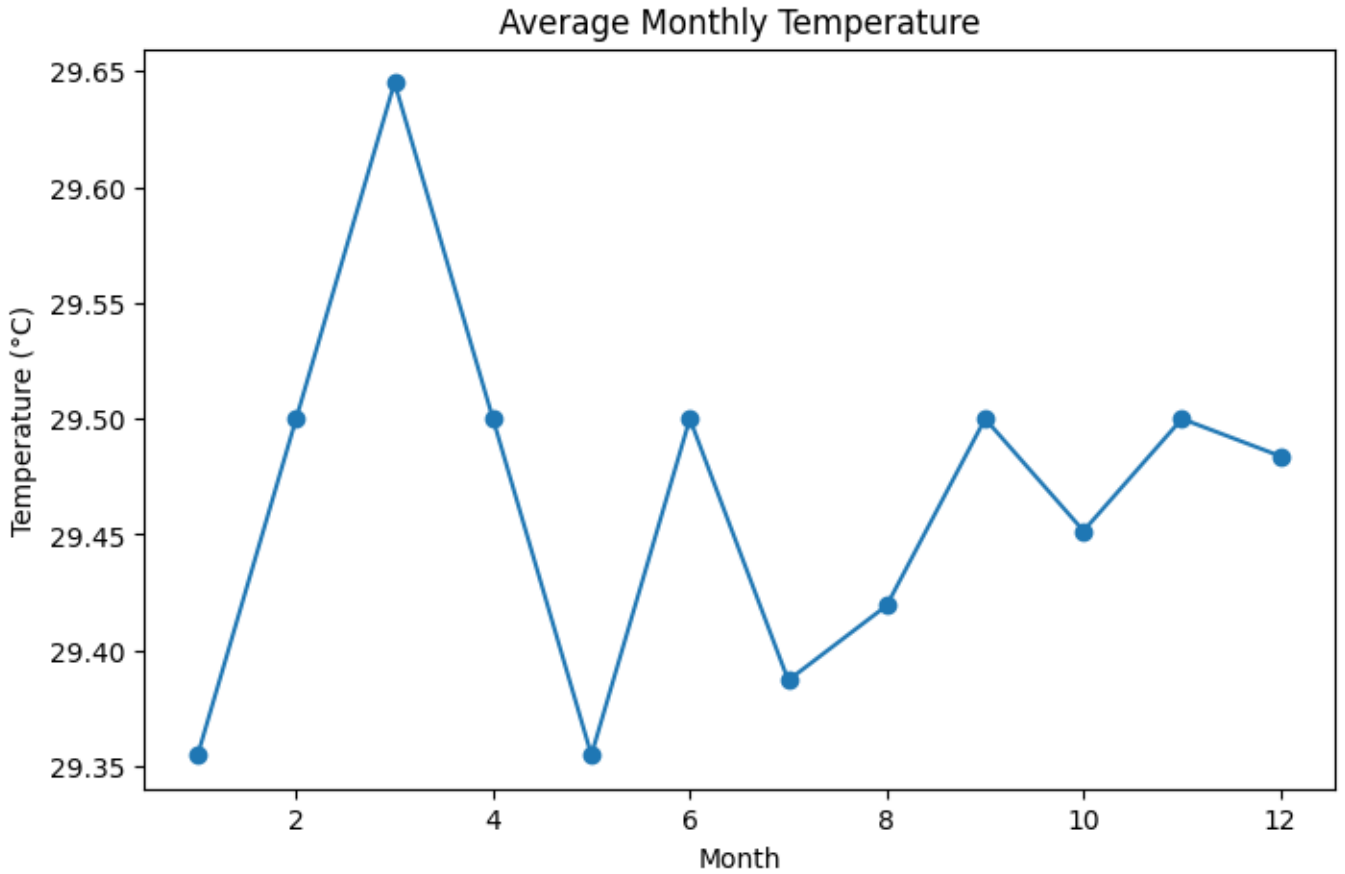
Average Monthly Rainfall

Temperature Trend Over Time

Rainfall Over Time

Humidity Trend Over Time

Average Monthly Temperature

The dataset contains **365 records** with the following columns:

Column Name	Description
Date	Date of observation
Temp	Temperature (°C)
Rainfall	Rainfall (mm)
Humidity	Humidity (%)

Weather Data Analysis – Mini Project Description

Project Title: Weather Data Analysis using Python

Objective:
The main objective of this project is to analyze historical weather data to understand patterns and

trends in **temperature, rainfall, and humidity**. The project helps in identifying seasonal variations and observing how weather parameters change over time.

Dataset:

The dataset used is a weather CSV file containing daily records with attributes such as **Date, Temperature, Rainfall, and Humidity**. The data is suitable for basic statistical analysis and visualization.

Tools & Technologies:

- Python
- pandas (data manipulation)
- matplotlib (data visualization)

Methodology:

1. Load the weather dataset using pandas.
2. Perform data preprocessing (date conversion, feature extraction).
3. Analyze temperature, rainfall, and humidity trends.
4. Compare seasonal variations using monthly averages.
5. Visualize results using line charts and bar graphs.

Output:

- Temperature trend graph
- Rainfall analysis chart
- Humidity variation graph
- Monthly seasonal comparison plots
- Summary statistics of weather parameters

Conclusion:

The project successfully demonstrates how Python can be used for weather data analysis. It reveals clear seasonal patterns in temperature, rainfall, and humidity, making it useful for academic learning and introductory data analytics practice.