

Weather Data Analysis Report

Problem Statement: Visualize temperature trends, rainfall, and seasonal patterns from a small weather dataset.

Name: Divyanshi Gautam

Roll No.: 20240110040087

Instructor's Name: Mr. Abhishek Shukla

Course: AI

Date: 10- 03- 2025

Introduction

Weather plays a crucial role in various aspects of life, from agriculture to urban planning. Analyzing weather data helps identify trends in temperature, rainfall, and seasonal variations, which can be useful for making informed decisions. This report presents an analysis of a small weather dataset to visualize temperature trends, rainfall distribution, and seasonal patterns.

Methodology

1. **Dataset Selection:** A small weather dataset containing temperature, rainfall, and date attributes was used.
2. **Data Preprocessing:** Missing values were handled, and data was cleaned for consistency.
3. **Visualization Techniques:**
 - Line charts for temperature trends.
 - Bar graphs for rainfall analysis.
 - Seasonal patterns explored through grouped visualizations.
4. **Tools Used:**
 - Python programming language.
 - Libraries: Pandas, Matplotlib.

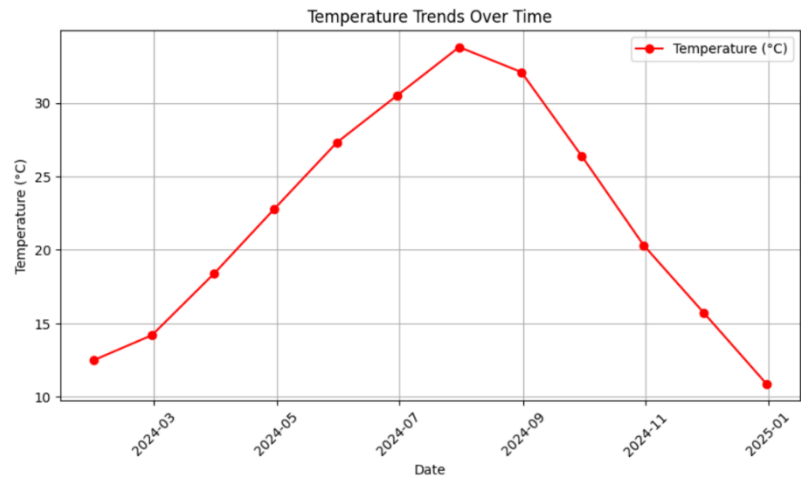
CODE

```
5. import pandas as pd
6. import matplotlib.pyplot as plt
7.
8. # Create the weather dataset
9. data = {
10.     "Date": pd.date_range(start="2024-01-05", periods=12,
        freq='M'),
11.     "Temperature (°C)": [12.5, 14.2, 18.4, 22.8, 27.3, 30.5,
        33.8, 32.1, 26.4, 20.3, 15.7, 10.9],
12.     "Rainfall (mm)": [5.2, 2.8, 1.5, 0.0, 3.1, 10.4, 15.2,
        12.7, 8.3, 6.1, 4.2, 7.5],
13.     "Humidity (%)": [75, 72, 68, 60, 55, 50, 52, 53, 58, 65,
        70, 74],
14.     "Wind Speed (km/h)": [10, 12, 14, 16, 18, 20, 22, 21,
        19, 15, 13, 11],
15.     "Season": ["Winter", "Winter", "Spring", "Spring",
        "Summer", "Summer", "Summer", "Summer", "Fall", "Fall", "Fall",
        "Winter"]
16. }
17.
18. # Convert to DataFrame
19. df = pd.DataFrame(data)
20.
21. # Convert Date column to datetime format
22. df["Date"] = pd.to_datetime(df["Date"])
23.
24. # Set Date as the index for plotting
25. df.set_index("Date", inplace=True)
26.
27. # Plot 1: Temperature Trend Over Time
28. plt.figure(figsize=(10, 5))
29. plt.plot(df.index, df["Temperature (°C)"], marker='o',
        linestyle='-', color='r', label="Temperature (°C)")
30. plt.xlabel("Date")
31. plt.ylabel("Temperature (°C)")
32. plt.title("Temperature Trends Over Time")
33. plt.xticks(rotation=45)
34. plt.legend()
35. plt.grid()
36. plt.show()
37.
38. # Plot 2: Rainfall Distribution
39. plt.figure(figsize=(10, 5))
```

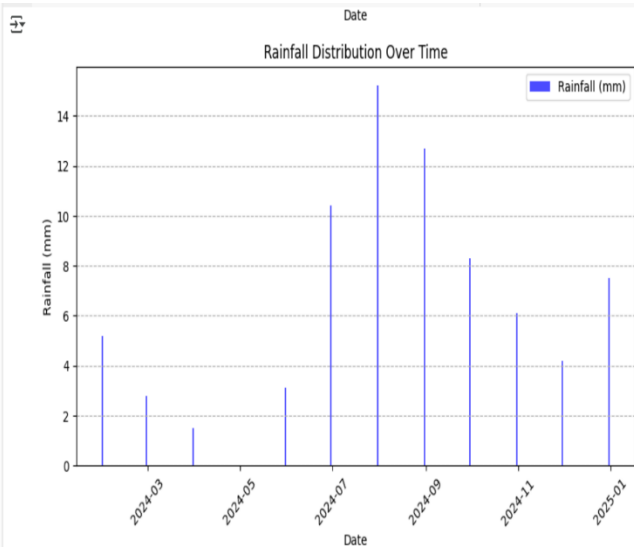
```
40.     plt.bar(df.index, df["Rainfall (mm)"], color='blue',
    alpha=0.7, label="Rainfall (mm)")
41.     plt.xlabel("Date")
42.     plt.ylabel("Rainfall (mm)")
43.     plt.title("Rainfall Distribution Over Time")
44.     plt.xticks(rotation=45)
45.     plt.legend()
46.     plt.grid(axis='y', linestyle='--')
47.     plt.show()
48.
49.     # Plot 3: Seasonal Temperature Variations (Box Plot)
50.     plt.figure(figsize=(8, 5))
51.     df.boxplot(column="Temperature (°C)", by="Season",
    grid=False, patch_artist=True)
52.     plt.title("Seasonal Temperature Variations")
53.     plt.suptitle("") # Remove default title
54.     plt.xlabel("Season")
55.     plt.ylabel("Temperature (°C)")
56.     plt.show()
57.
```

Output

```
<ipython-input-1-0c0d9f4bb0ed>:6: FutureWarning: 'M' is deprecated and will be removed in a future version, please use 'ME' instead.  
"Date": pd.date_range(start="2024-01-05", periods=12, freq='M'),
```



✓ 1s completed at 2:15PM



References/Credits

- **Dataset Source: chatgpt**
 - **Python Libraries Used: Pandas, Matplotlib**
 - **Images/Graphs: Generated using Matplotlib**
-

