Weather Data Analysis Report

Project Title: Weather Data Analysis

Student Name: Saumya Dubey

Roll No.: 01

Branch: CSE AI

AKTU ROLL NO.: 202401100300221

Date: 11/03/2025

Introduction

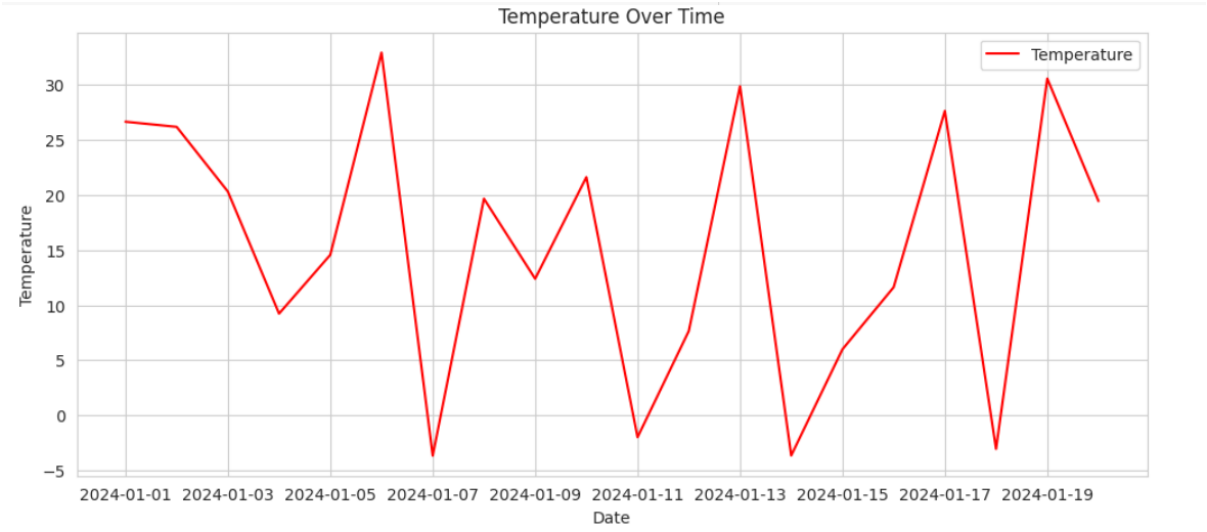
Problem Statement

Weather analysis plays a crucial role in understanding climate patterns, predicting extreme weather conditions, and making informed decisions in various sectors such as agriculture, transportation, and urban planning. The goal of this project is to analyze weather data, explore temperature trends, visualize relationships between different weather parameters, and detect anomalies.

Objectives

- Load and preprocess weather data.
- Perform exploratory data analysis (EDA) to identify trends and patterns.
- Visualize temperature variations over time.
- Detect correlations between different weather parameters.
- Identify anomalies using statistical methods.

Below is an example of *w*eather trend image:



Methodology

The following steps were followed to perform weather data analysis:

1. **Dataset Upload:** The dataset was uploaded in CSV format using Google Colab.
2. **Data Preprocessing:** The dataset was cleaned by handling missing values and converting date columns to proper formats.
3. **Exploratory Data Analysis (EDA):** Basic statistical information was extracted, including mean, median, and standard deviation of temperature values.
4. **Data Visualization:** Various visualizations such as line charts, histograms, and heatmaps were used to analyze trends.
5. **Outlier Detection:** Boxplots were used to identify temperature anomalies.

Code

```
# Import necessary libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from google.colab import files


# Step 1: Upload the dataset

print("Please upload your weather dataset (CSV file).")

uploaded = files.upload()


# Get the filename from uploaded dictionary

filename = list(uploaded.keys())[0]


# Step 2: Load the dataset into a Pandas DataFrame

df = pd.read_csv(filename)


# Display the first few rows of the dataset

print("Dataset Preview:")
```

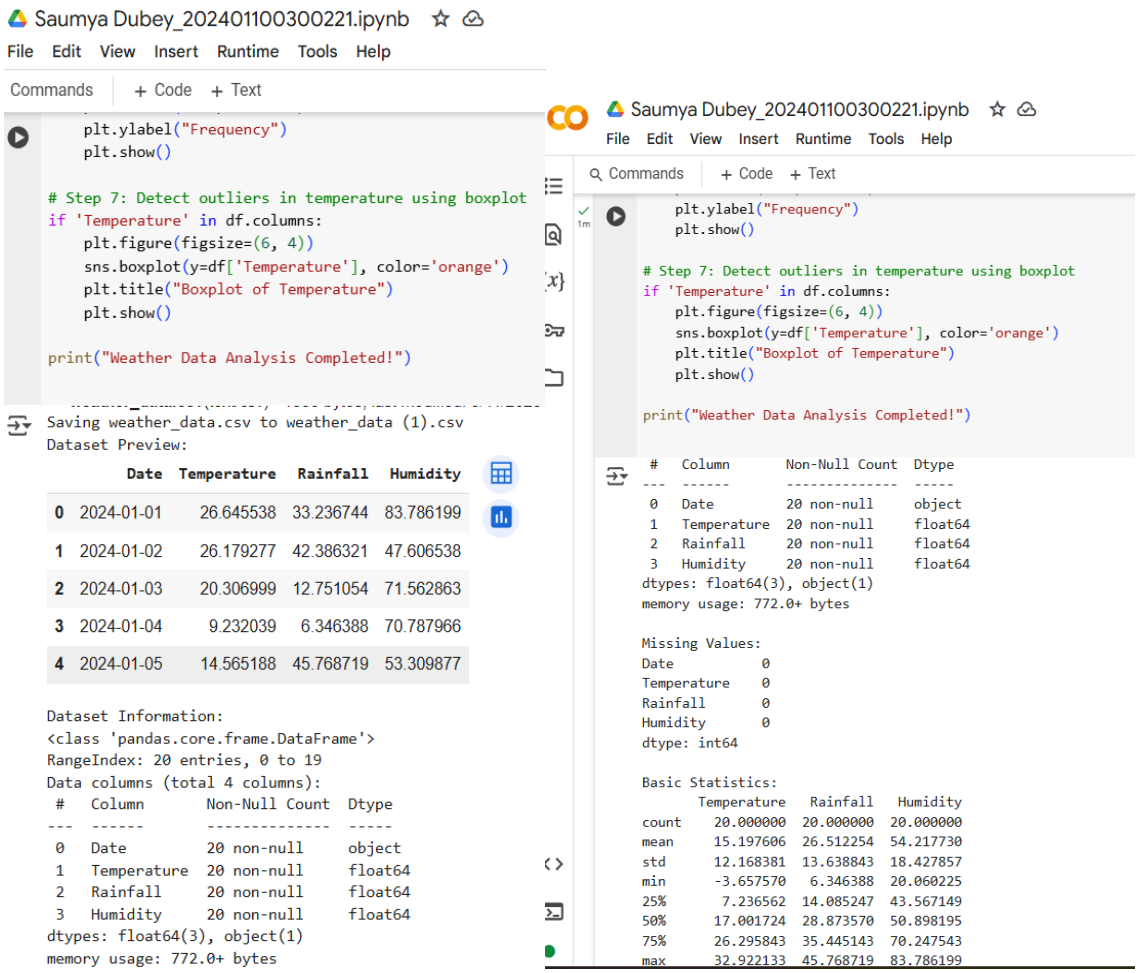
```
display(df.head())

# Step 3: Display basic information about the dataset
print("\nDataset Information:")
df.info()

# Step 4: Check for missing values
print("\nMissing Values:")
print(df.isnull().sum())

# Step 5: Perform basic statistics on numerical columns
print("\nBasic Statistics:")
print(df.describe())

# Step 6: Data Visualization
sns.set_style("whitegrid")

# Line plot of temperature over time (if date column is available)
if 'Date' in df.columns:
    df['Date'] = pd.to_datetime(df['Date'])  # Convert to datetime format
    df.set_index('Date', inplace=True)
    plt.figure(figsize=(12, 5))
    plt.plot(df.index, df['Temperature'], label='Temperature', color='red')
    plt.xlabel("Date")
    plt.ylabel("Temperature")
    plt.title("Temperature Over Time")
    plt.legend()
    plt.show()
```

```
# Correlation heatmap
plt.figure(figsize=(10,6))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt='.2f')
plt.title("Correlation Heatmap")
plt.show()

# Distribution plot for temperature
if 'Temperature' in df.columns:
    plt.figure(figsize=(8, 5))
    sns.histplot(df['Temperature'], bins=30, kde=True, color='blue')
    plt.title("Temperature Distribution")
    plt.xlabel("Temperature")
    plt.ylabel("Frequency")
    plt.show()

# Step 7: Detect outliers in temperature using boxplot
if 'Temperature' in df.columns:
    plt.figure(figsize=(6, 4))
    sns.boxplot(y=df['Temperature'], color='orange')
    plt.title("Boxplot of Temperature")
    plt.show()

print("Weather Data Analysis Completed!")
```

Output/Result

Below are the screenshots of the program execution and visualizations:

Screenshot 1: Dataset Upload and Display



Saumya Dubey_202401100300221.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

```
    plt.ylabel("Frequency")
    plt.show()

# Step 7: Detect outliers in temperature using boxplot
if 'Temperature' in df.columns:
    plt.figure(figsize=(6, 4))
    sns.boxplot(y=df['Temperature'], color='orange')
    plt.title("Boxplot of Temperature")
    plt.show()

print("Weather Data Analysis Completed!")
```

```
Saving weather_data.csv to weather_data (1).csv
Dataset Preview:
```

	Date	Temperature	Rainfall	Humidity
0	2024-01-01	26.645538	33.236744	83.786199
1	2024-01-02	26.179277	42.386321	47.606538
2	2024-01-03	20.306999	12.751054	71.562863
3	2024-01-04	9.232039	6.346388	70.787966
4	2024-01-05	14.565188	45.768719	53.309877

```
Dataset Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 4 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Date         20 non-null     object
 1   Temperature  20 non-null     float64
 2   Rainfall     20 non-null     float64
 3   Humidity     20 non-null     float64
dtypes: float64(3), object(1)
memory usage: 772.0+ bytes
```

Saumya Dubey_202401100300221.ipynb

File Edit View Insert Runtime Tools Help

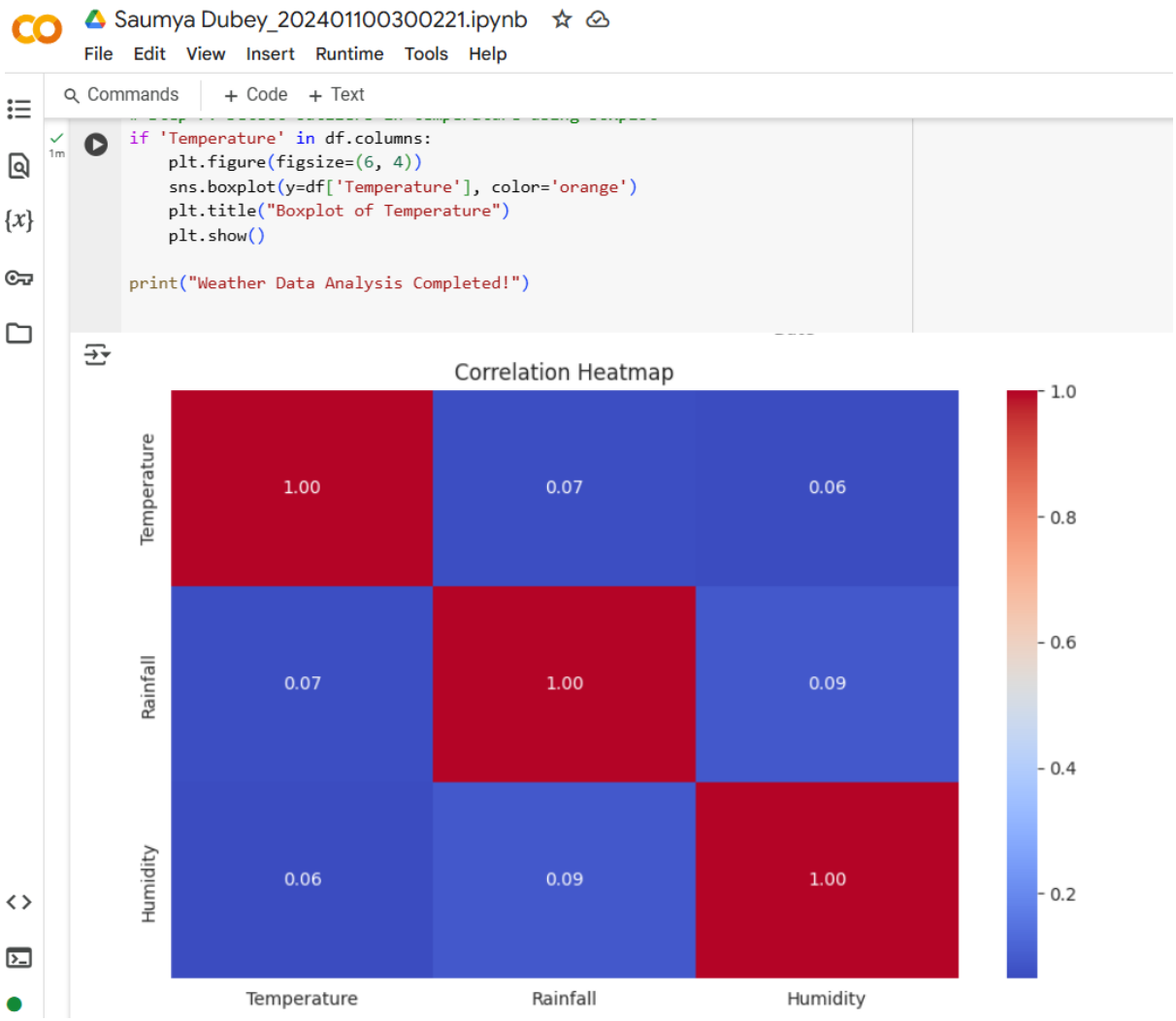
Commands + Code + Text

```
    plt.ylabel("Frequency")
    plt.show()

# Step 7: Detect outliers in temperature using boxplot
if 'Temperature' in df.columns:
    plt.figure(figsize=(6, 4))
    sns.boxplot(y=df['Temperature'], color='orange')
    plt.title("Boxplot of Temperature")
    plt.show()

print("Weather Data Analysis Completed!")
```

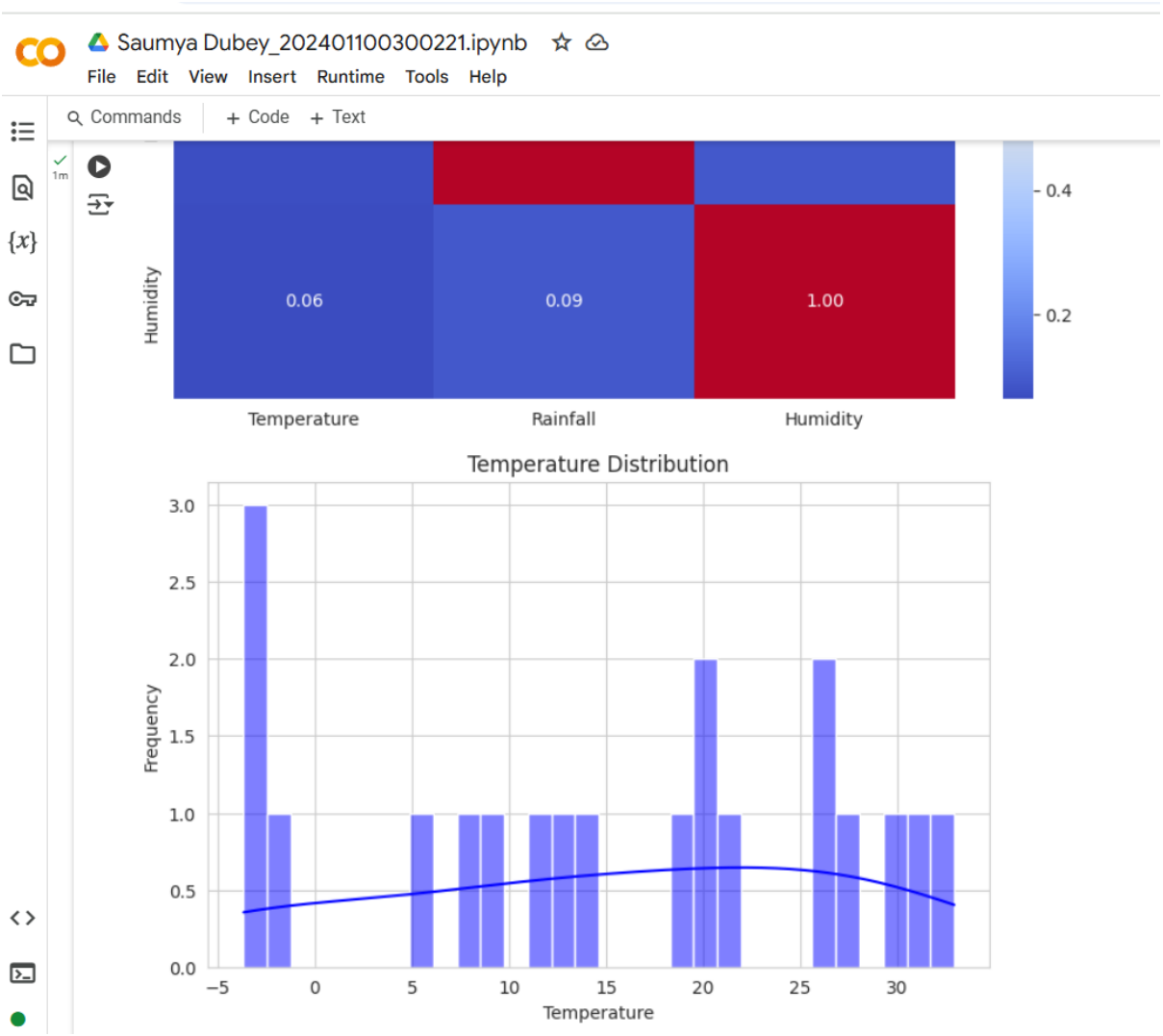
```
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Date         20 non-null     object
 1   Temperature  20 non-null     float64
 2   Rainfall     20 non-null     float64
 3   Humidity     20 non-null     float64
dtypes: float64(3), object(1)
memory usage: 772.0+ bytes

Missing Values:
Date           0
Temperature    0
Rainfall       0
Humidity       0
dtype: int64

Basic Statistics:
       Temperature   Rainfall   Humidity
count    20.000000  20.000000  20.000000
mean     15.197606  26.512254  54.217730
std      12.168381  13.638843  18.427857
min      -3.657570   6.346388  20.060225
25%       7.236562  14.085247  43.567149
50%      17.001724  28.873570  50.898195
75%      26.295843  35.445143  70.247543
max      32.922133  45.768719  83.786199
```

Screenshot 2: Temperature Over Time Plot

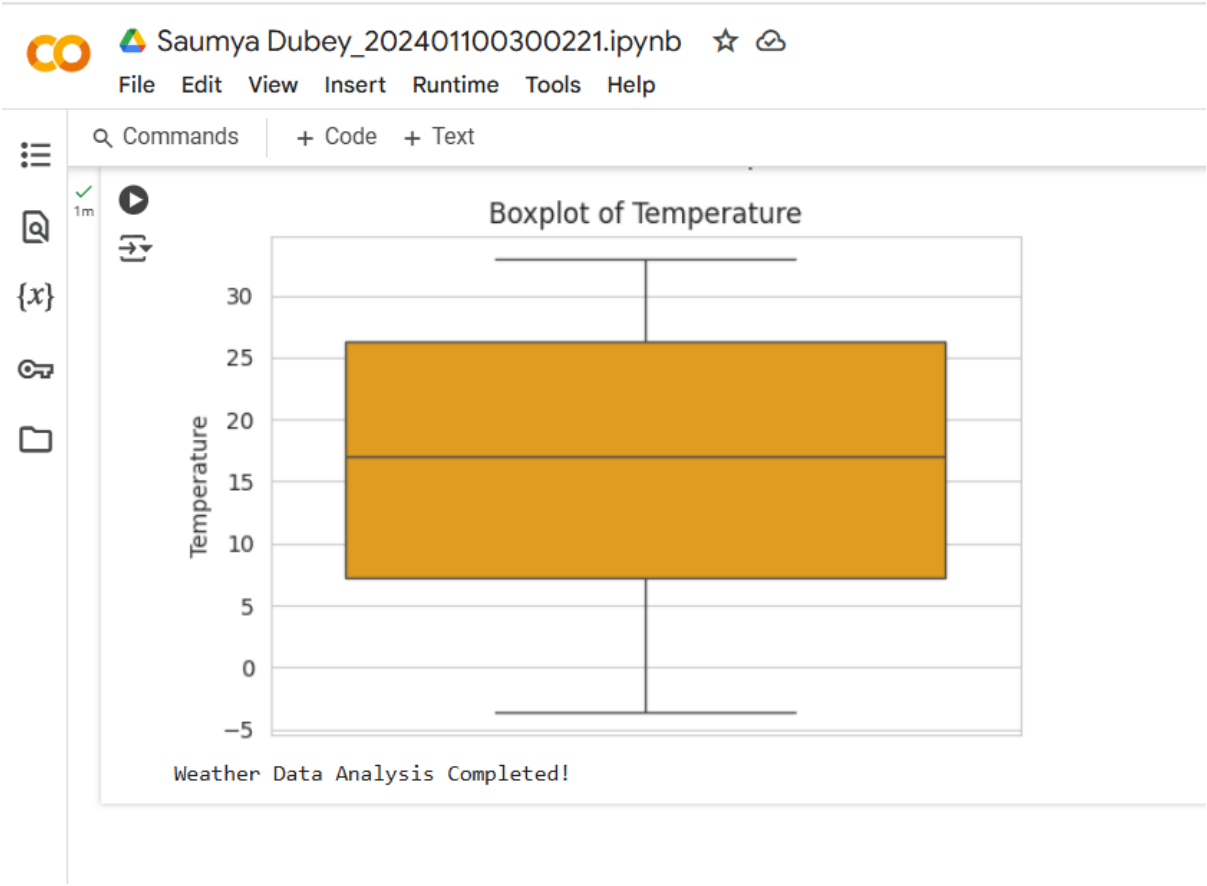
Screenshot 3: Correlation Heatmap



Screenshot 4: Temperature Distribution Plot



Screenshot 5: Boxplot for Temperature Outliers



References/Credits

- Dataset Source: [Mention dataset source, e.g., Kaggle, NOAA, etc.]
- Seaborn Documentation: https://seaborn.pydata.org/
- Matplotlib Documentation: https://matplotlib.org/
- Google Colab Documentation: https://colab.research.google.com/

End of Report