

Weather Data Analysis and Temperature Trends in New York

Project review

In this project, we will analyze historical weather data to identify temperature trends and patterns. The analysis will include loading the dataset, visualizing temperature changes over time, and calculating basic statistics like mean, maximum, and minimum temperatures. This project will help you understand the fundamentals of data cleaning, visualization, and analysis in a simple yet meaningful way.

Dataset

Source: Global Weather Data (available on Kaggle)

Format: CSV file containing columns such as:

- City
- Date
- Temperature (in Celsius)
- Humidity
- Weather Condition

Project Steps

- Load the dataset.
- Clean and preprocess the data.
- Plot temperature trends over time.
- Calculate basic statistics (mean, max, min temperatures).
- Visualize patterns with line and bar charts.

Tools Required:

Python Libraries:

- pandas for data manipulation.
- matplotlib for data visualization.
- Seaborn for enhanced plots.

Step-by-Step Implementation

1. Load the Dataset

First, let's import the necessary libraries and load the dataset.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Load the dataset

```
weather_data = pd.read_csv('weather_data.csv')
```

Display the first few rows

```
print(weather_data.head())
```

2. Clean and Preprocess the Data

Ensure the dataset is clean and ready for analysis.

Check for missing values

```
print(weather_data.isnull().sum())
```

Drop rows with missing values

```
weather_data.dropna(inplace=True)
```

Convert 'Date' to datetime format

```
weather_data['Date'] = pd.to_datetime(weather_data['Date'])
```

Verify the data types

```
print(weather_data.dtypes)
```

3. Plot Temperature Trends Over Time

Let's visualize temperature changes over time for a specific city, such as New York.

Filter data for New York

```
nyc_data = weather_data[weather_data['City'] == 'New York']
```

Plot temperature trends over time

```
plt.figure(figsize=(12, 6))
```

```
plt.plot(nyc_data['Date'], nyc_data['Temperature(°C)'], color='blue', marker='o', linestyle='-')
```

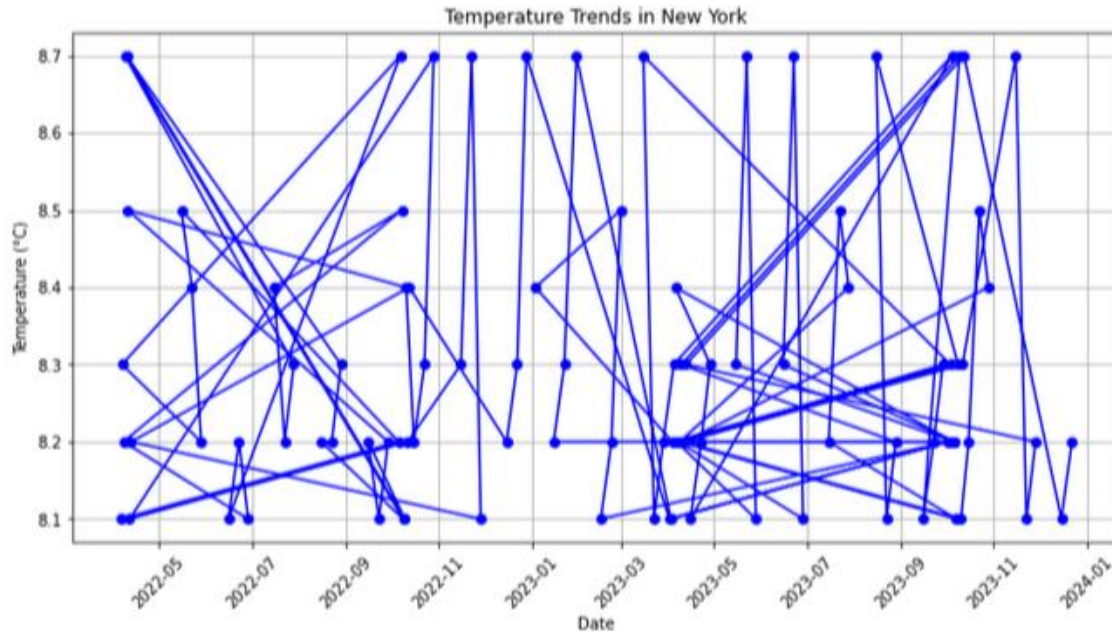
```
plt.title("Temperature Trends in New York")
```

```
plt.xlabel('Date')
```

```
plt.ylabel('Temperature (°C)')
```

```
plt.grid(True)
```

```
plt.xticks(rotation=45) plt.show()
```



4. Calculate Basic Statistics

Compute the mean, maximum, and minimum temperatures for New York.

```
mean_temp = nyc_data['Temperature(°C)'].mean()
```

```
max_temp = nyc_data['Temperature(°C)'].max()
```

```
min_temp = nyc_data['Temperature(°C)'].min()
```

```
print(f'Mean Temperature: {mean_temp:.2f} °C')
```

```
print(f'Maximum Temperature: {max_temp:.2f} °C')
```

```
print(f'Minimum Temperature: {min_temp:.2f} °C')
```

5. Visualize Patterns with Line and Bar Charts Line Plot of Daily Temperatures

```
plt.figure(figsize=(12, 6))
```

```
sns.lineplot(data=nyc_data, x='Date', y='Temperature(°C)', color='green')
```

```
plt.title('Daily Temperature Trends in New York')
```

```
plt.xlabel('Date')
```

```
plt.ylabel('Temperature (°C)')
```

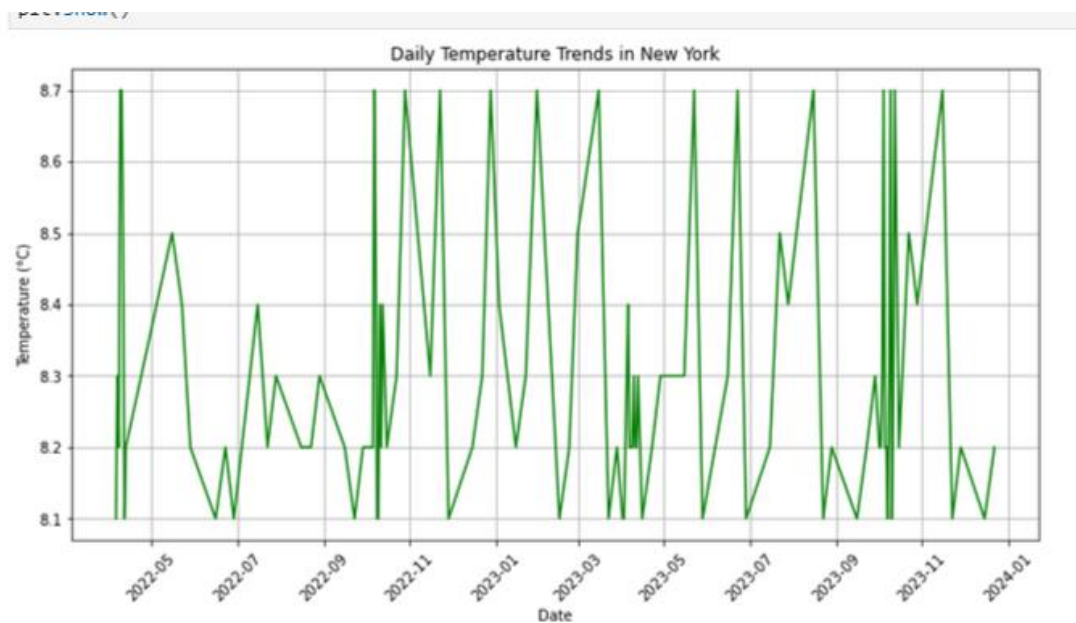
```
plt.xticks(rotation=45)
```

```
plt.grid(True)
```

```
plt.show()
```

Bar Plot Showing Average Monthly Temperature

Now, let's aggregate the data by month to see monthly trends.



Extract month and year from the Date

```
nyc_data['Month'] = nyc_data['Date'].dt.month
```

```
nyc_data['Year'] = nyc_data['Date'].dt.year
```

Group by year and month to calculate the average monthly temperature

```
monthly_avg_temp = nyc_data.groupby(['Year', 'Month'])['Temperature'].mean().reset_index()
```

Create a new 'Month-Year' column for plotting

```
monthly_avg_temp['Month-Year'] = monthly_avg_temp['Year'].astype(str) + '-' +  
monthly_avg_temp['Month'].astype(str).str.zfill(2)
```

Plot the average monthly temperature

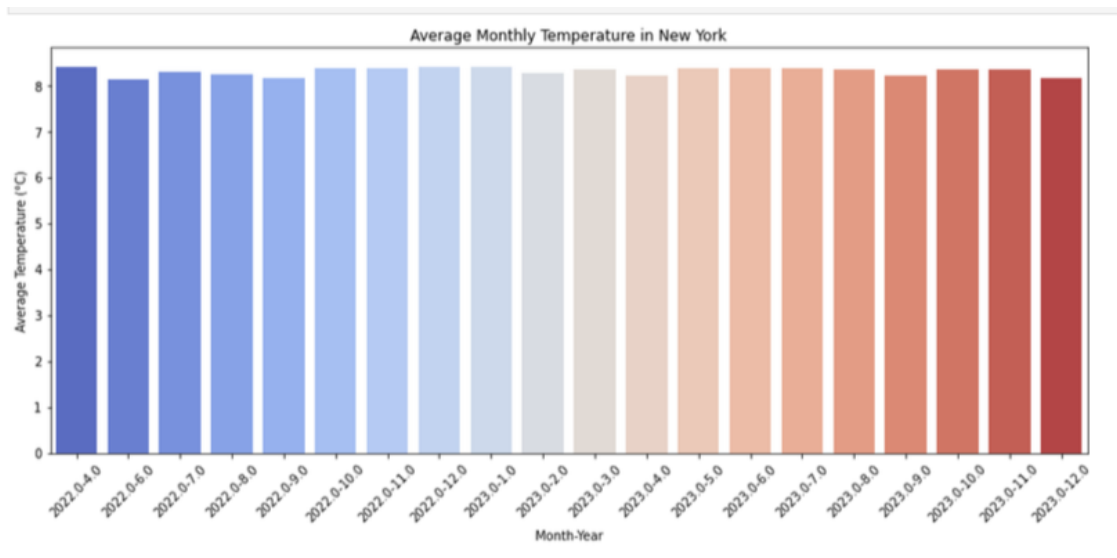
```
plt.figure(figsize=(15, 6))
```

```
sns.barplot(data=monthly_avg_temp, x='Month-Year', y='Temperature', palette='coolwarm')  
plt.title('Average Monthly Temperature in New York')
```

```
plt.xlabel('Month-Year')
```

```
plt.ylabel('Average Temperature (°C)')
```

```
plt.xticks(rotation=45) plt.show()
```



Results and Interpretation

Temperature Trends:

The line chart shows how the temperature fluctuates over different days.

Noticeable dips in temperature may indicate winter months, while peaks represent summer.

Basic Statistics:

The mean, maximum, and minimum temperatures give an idea of the overall climate in New York.

Monthly Patterns:

The bar chart helps visualize how the temperature changes month by month over the years.

Conclusion

In this project, we analyzed weather data for New York by:

Loading and cleaning the dataset. Visualizing temperature trends over time. Calculating key statistics like mean, max, and min temperatures. Plotting monthly temperature trends. This analysis can be extended to other cities or include additional factors like humidity, wind speed, or precipitation.