```
from google.colab import files
uploaded = files.upload()
```

Choose Files weather_data.csv.zip

- **weather_data.csv.zip**(application/x-zip-compressed) - 44638390 bytes, last modified: 8/3/2025 - 100% done

```
Saving weather_data.csv.zip to weather_data.csv.zip
```

```
import pandas as pd

# Replace 'your_file.csv' with your actual filename
df = pd.read_csv('weather_data.csv.zip')

# Preview first few rows
df.head()
```

	Location	Date_Time	Temperature_C	Humidity_pct	Precipitation_mm	Wind_Speed_kmh
0	San Diego	2024-01-14 21:12:46	10.683001	41.195754	4.020119	8.233540
1	San Diego	2024-05-17 15:22:10	8.734140	58.319107	9.111623	27.715161
2	San Diego	2024-05-11 09:30:59	11.632436	38.820175	4.607511	28.732951
3	Philadelphia	2024-02-26 17:32:39	-8.628976	54.074474	3.183720	26.367303
4	San Antonio	2024-04-29 13:23:51	39.808213	72.899908	9.598282	29.898622

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Task

Analyze the provided CSV data using Python to perform comprehensive data analysis. The analysis should include importing necessary libraries, loading and previewing the data, handling missing values, generating descriptive statistics, creating various visualizations (bar charts, pie charts, line charts, correlation heatmap, histograms), identifying patterns, trends, and outliers, and extracting 5 key insights. The final output should be visually appealing and well-structured, incorporating multiple charts to showcase the data attractively.

Import libraries

Subtask:

Import necessary libraries (Pandas, NumPy, Matplotlib, Seaborn).

Reasoning: Import the necessary libraries for data analysis and visualization.

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Handle missing values

Subtask:

Check for missing values in the data nd.

Reasoning: Check for missing value

✓ Describe data
✓ Create visualizations
✓ Analyze patterns and trends
✓ Extract key insights
✓ Present analysis
✓ Finish task

What can I help you build?

Gemini can make mistakes so double-check it and use code with caution. Learn more

```
print(df.isnull().sum())
```

```
Location            0
Date_Time           0
Temperature_C       0
Humidity_pct        0
Precipitation_mm    0
Wind_Speed_kmh      0
dtype: int64
```

Describe data

Subtask:

Generate descriptive statistics for numerical columns and explore the unique values and counts for categorical columns.

Reasoning: Generate descriptive statistics for numerical columns and explore the unique values and counts for the categorical column 'Location'.

```
display(df.describe())
display(df['Location'].value_counts())
```

	Temperature_C	Humidity_pct	Precipitation_mm	Wind_Speed_kmh
count	1000000.000000	1000000.000000	1000000.000000	1000000.000000
mean	14.779705	60.021830	5.109639	14.997598
std	14.482558	17.324022	2.947997	8.663556
min	-19.969311	30.000009	0.000009	0.000051
25%	2.269631	45.008500	2.580694	7.490101
50%	14.778002	60.018708	5.109917	14.993777
75%	27.270489	75.043818	7.613750	22.514110
max	39.999801	89.999977	14.971583	29.999973

	count
Location	
Phoenix	100209
Chicago	100164
Philadelphia	100122
Houston	100076
New York	99972
San Antonio	99962
Dallas	99936
Los Angeles	99922
San Jose	99863
San Diego	99774

dtype: int64

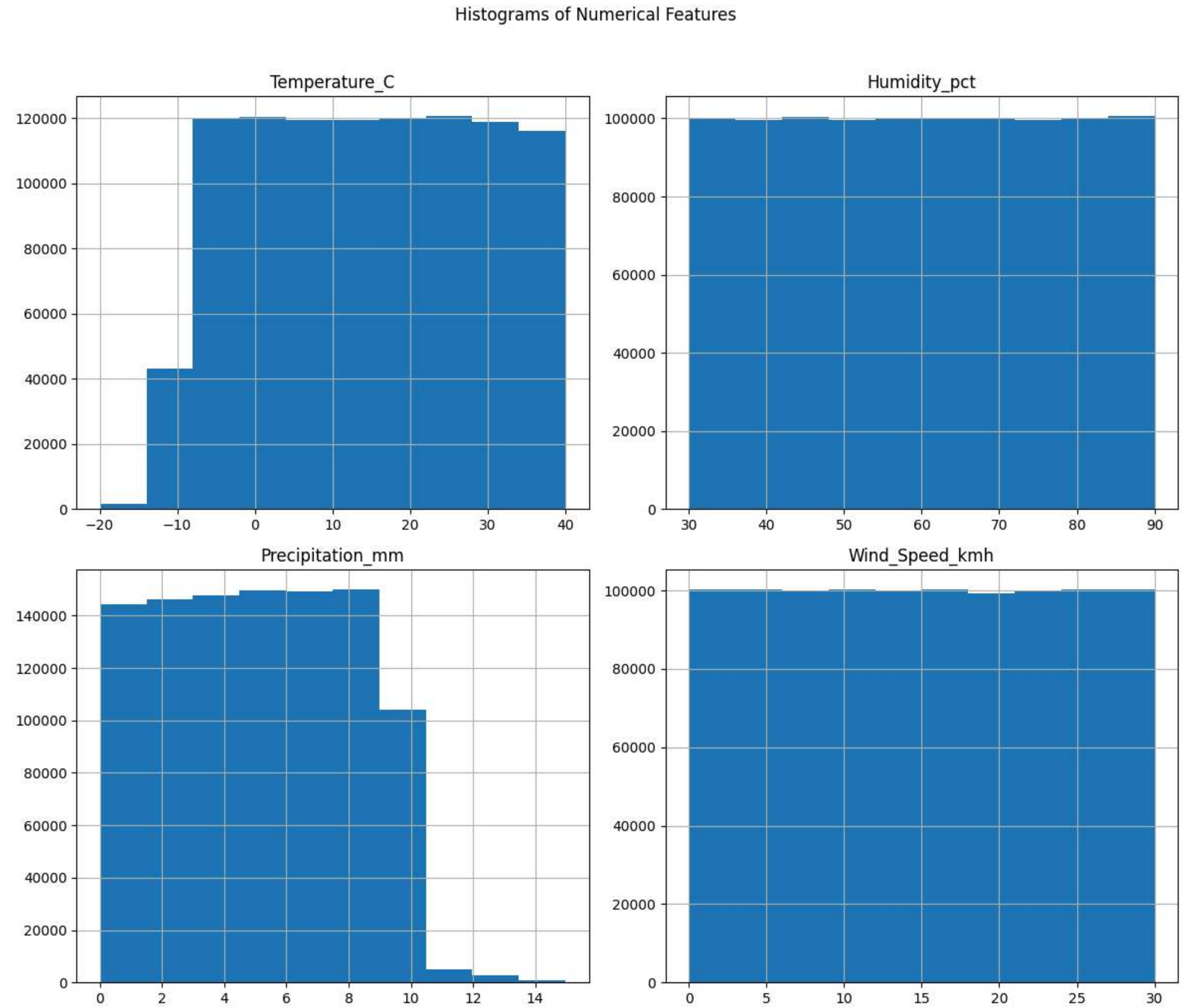
Create visualizations

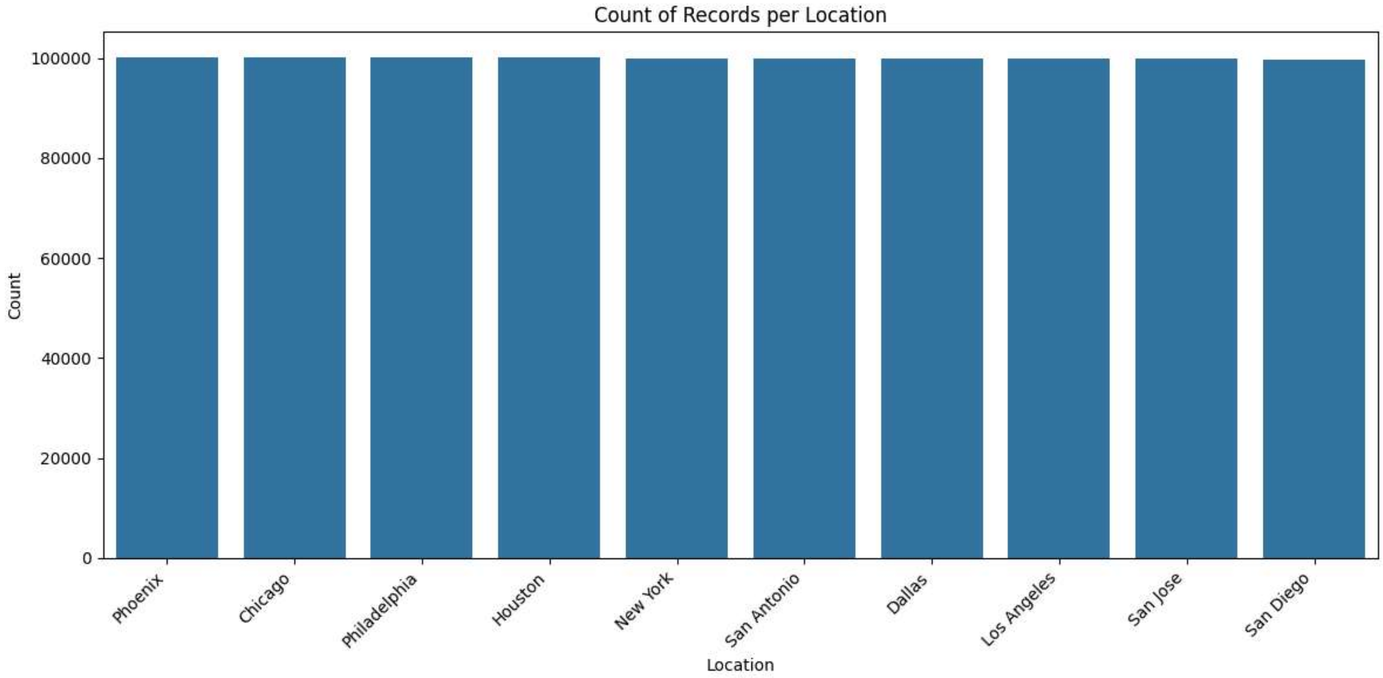
Subtask:

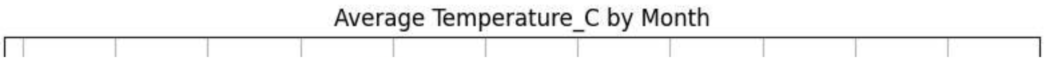
Generate various plots to visualize the data distribution, relationships between variables, and trends. This includes histograms, bar charts, line charts, and a correlation heatmap.

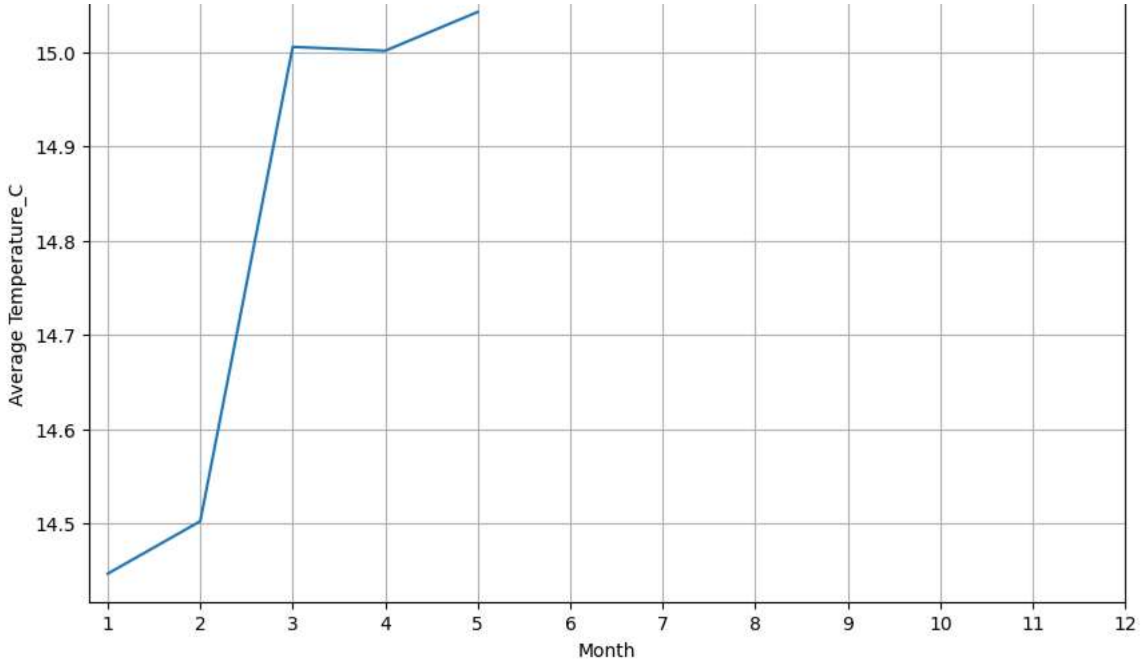
Reasoning: Generate various plots to visualize the data distribution, relationships between variables, and trends, including histograms, bar charts, line charts, and a correlation heatmap.

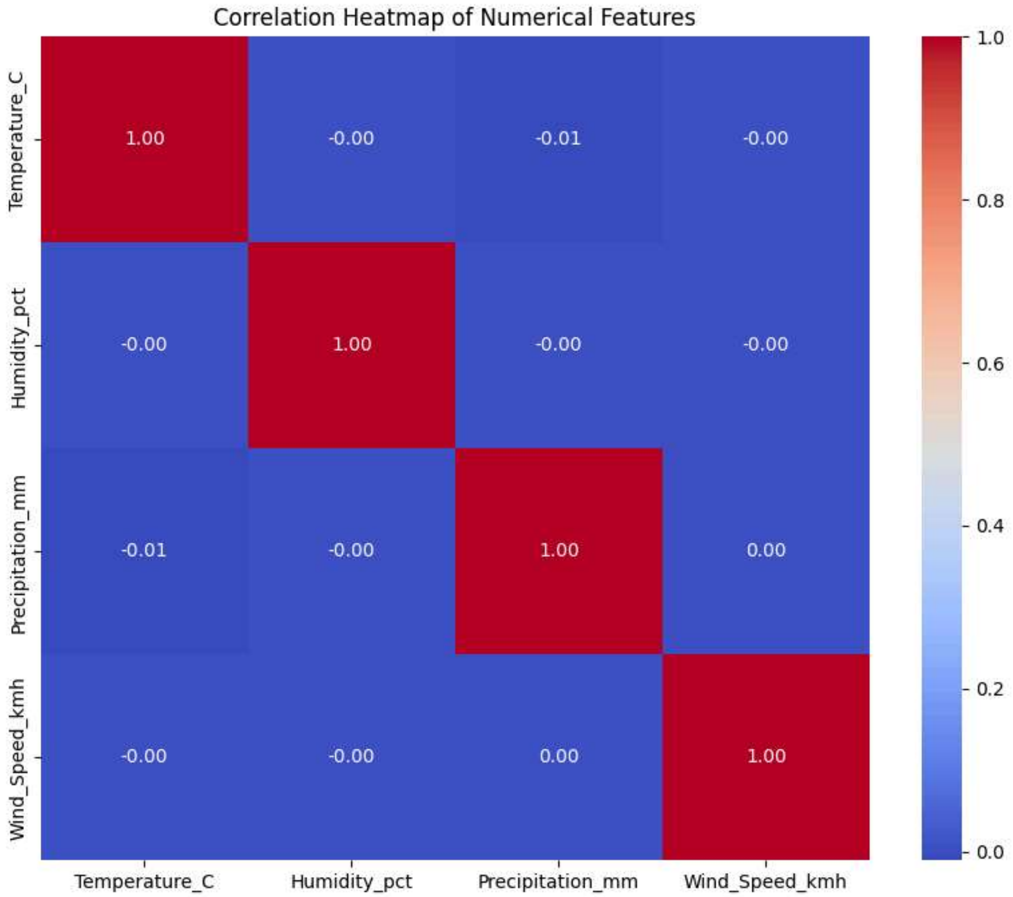
```
# 1. Histograms for numerical columns
numerical_cols = ['Temperature_C', 'Humidity_pct', 'Precipitation_mm', 'Wind_Speed_kmh']
df[numerical_cols].hist(figsize=(12, 10))
plt.suptitle('Histograms of Numerical Features', y=1.02)
plt.tight_layout()
plt.show()

# 2. Bar chart for 'Location'
plt.figure(figsize=(12, 6))
sns.countplot(data=df, x='Location', order=df['Location'].value_counts().index)
plt.title('Count of Records per Location')
plt.xlabel('Location')
plt.ylabel('Count')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

# 3. Convert 'Date_Time' to datetime objects
df['Date_Time'] = pd.to_datetime(df['Date_Time'])

# 4. Extract month
df['Month'] = df['Date_Time'].dt.month

# 5. Line plot for average 'Temperature_C' over 'Month'
avg_temp_by_month = df.groupby('Month')['Temperature_C'].mean().reset_index()
plt.figure(figsize=(10, 6))
sns.lineplot(data=avg_temp_by_month, x='Month', y='Temperature_C')
plt.title('Average Temperature_C by Month')
plt.xlabel('Month')
plt.ylabel('Average Temperature_C')
plt.xticks(range(1, 13))
plt.grid(True)
plt.show()

# 6. Correlation heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(df[numerical_cols].corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap of Numerical Features')
plt.show()
```


Histograms of Numerical Features
Temperature_C
0
20000
40000
60000
80000
100000
120000
−20
−10
0
10
20
30
40
Humidity_pct
0
20000
40000
60000
80000
100000
30
40
50
60
70
80
90
Precipitation_mm
0
20000
40000
60000
80000
100000
120000
140000
0
2
4
6
8
10
12
14
Wind_Speed_kmh
0
20000
40000
60000
80000
100000
0
5
10
15
20
25
30


Count of Records per Location
Count
0
20000
40000
60000
80000
100000
Phoenix
Chicago
Philadelphia
Houston
New York
San Antonio
Dallas
Los Angeles
San Jose
San Diego
Location


Average Temperature_C by Month


Average Temperature_C
15.0
14.9
14.8
14.7
14.6
14.5
1
2
3
4
5
6
7
8
9
10
11
12
Month


Correlation Heatmap of Numerical Features
Temperature_C
Humidity_pct
Precipitation_mm
Wind_Speed_kmh
1.00
-0.00
-0.01
-0.00
-0.00
1.00
-0.00
-0.00
-0.01
-0.00
1.00
0.00
-0.00
-0.00
0.00
1.00
Temperature_C
Humidity_pct
Precipitation_mm
Wind_Speed_kmh
1.0
0.8
0.6
0.4
0.2
0.0

Analyze patterns and trends

Subtask:

Based on the descriptive statistics and visualizations, identify patterns, trends, and potential outliers in the data.

Extract key insights

Subtask:

Summarize the findings from the analysis into 5 key insights about the weather data.

Present analysis

Subtask:

Present the analysis in a structured and visually appealing manner, including the code, outputs, visualizations, and insights.

Reasoning: Add markdown cells to structure the notebook and include the key insights.

```
# This is a placeholder cell to maintain the flow before adding markdown cells.
# The actual content will be added in markdown cells.
```

Reasoning: Add a markdown cell for the introduction and data loading section.

```
%%markdown
# Comprehensive Weather Data Analysis

This notebook presents a comprehensive analysis of the provided weather data. The analysis includes data loading, handling missing values, g

## Data Loading and Preview

The weather data is loaded into a pandas DataFrame and the first few rows are displayed to get an initial look at the data structure and cor
```

Comprehensive Weather Data Analysis

This notebook presents a comprehensive analysis of the provided weather data. The analysis includes data loading, handling missing values, generating descriptive statistics, creating various visualizations, and extracting key insights from the dataset.

Data Loading and Preview

The weather data is loaded into a pandas DataFrame and the first few rows are displayed to get an initial look at the data structure and content.

Reasoning: Add a markdown cell for the descriptive statistics section.

```
%%markdown
## Descriptive Statistics

Descriptive statistics for the numerical columns are generated to understand the central tendency, dispersion, and shape of the data distrib
```

Descriptive Statistics

Descriptive statistics for the numerical columns are generated to understand the central tendency, dispersion, and shape of the data distribution. The unique values and their counts for the 'Location' column are also explored.

Reasoning: Add a markdown cell for the visualization section.

```
%%markdown
## Visualizations

Various visualizations, including histograms, bar charts, line charts, and a correlation heatmap, are generated to explore the data distribu
```