

Perform Exploratory Data Analysis on the provided dataset to extract meaningful insights using pandas, Matplotlib/plotly

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from google.colab import files
uploaded = files.upload()
```

Choose Files

weatherHistory.csv

- weatherHistory.csv**(text/csv) - 16294377 bytes, last modified: 6/30/2024 - 100% done

Saving weatherHistory.csv to weatherHistory (1).csv

```
file_name = list(uploaded.keys())[0]

# Load the dataset
data = pd.read_csv(file_name)
data.head()
```

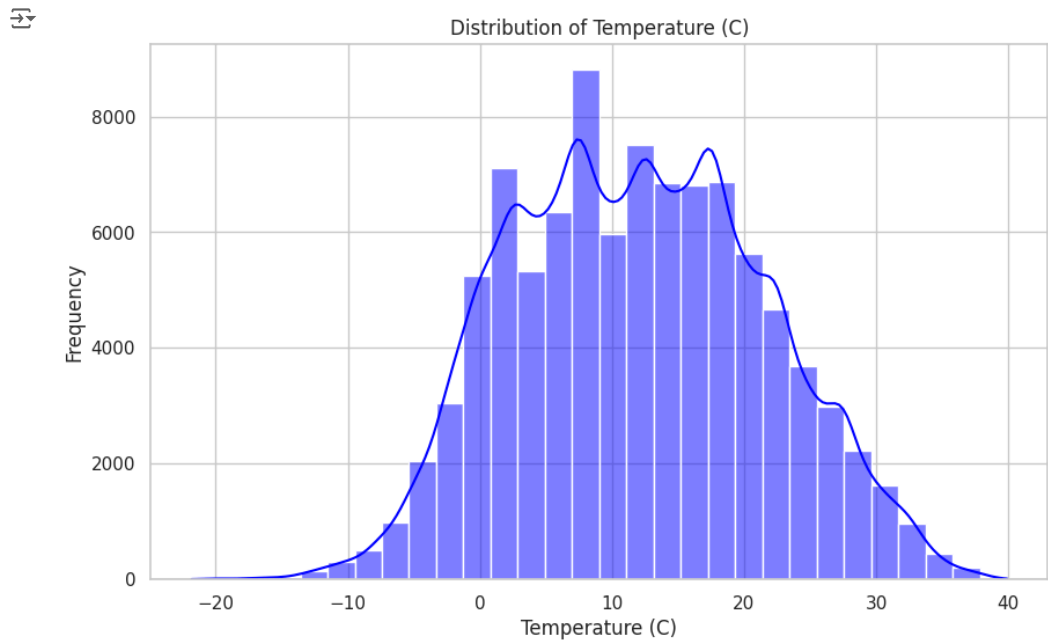
	Formatted Date	Summary	Precip Type	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibility (km)	Loud Cover (m	
0	2006-04-01 00:00:00.000+0200	Partly Cloudy	rain	9.472222	7.388889	0.89	14.1197	251.0	15.8263	0.0	
1	2006-04-01 01:00:00.000+0200	Partly Cloudy	rain	9.355556	7.227778	0.86	14.2646	259.0	15.8263	0.0	
2	2006-04-01 02:00:00.000+0200	Mostly Cloudy	rain	9.377778	9.377778	0.89	3.9284	204.0	14.9569	0.0	

Next steps: [Generate code with data](#) [View recommended plots](#)

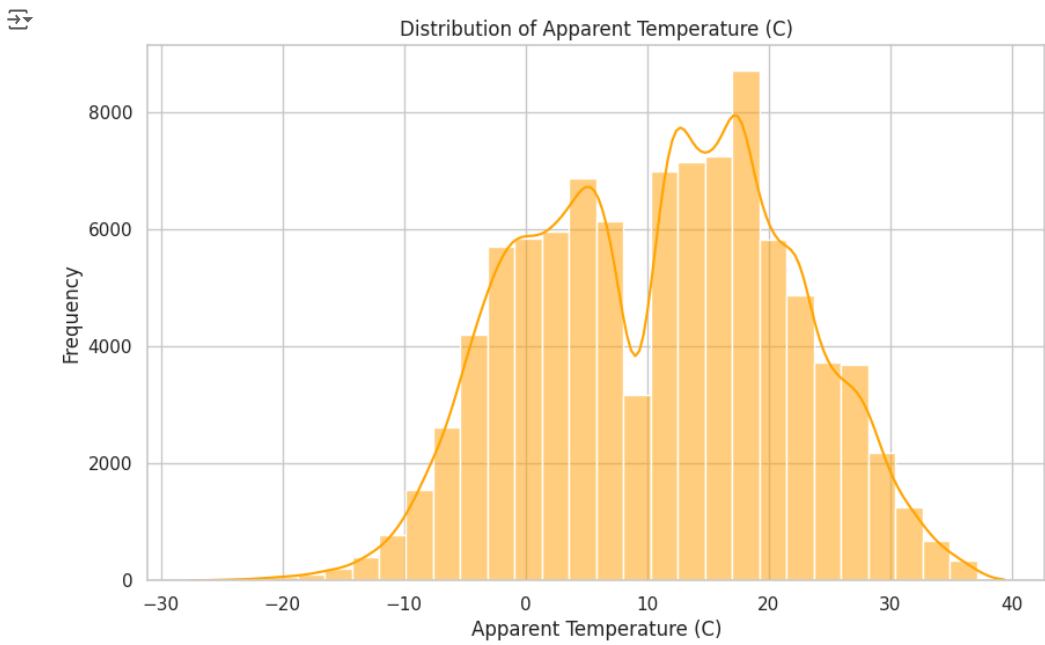
```
# Convert Formatted Date to datetime
data['Formatted Date'] = pd.to_datetime(data['Formatted Date'],utc=True)
```

```
# Set the style for the plots
sns.set(style="whitegrid")
```

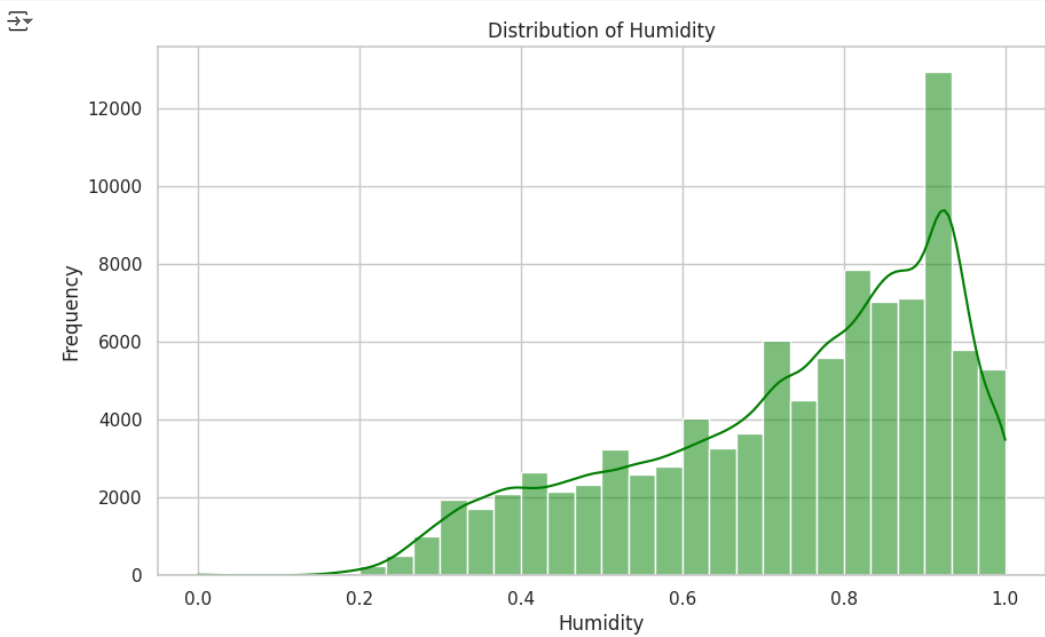
```
# Histogram for Temperature
plt.figure(figsize=(10, 6))
sns.histplot(data['Temperature (C)'], bins=30, kde=True, color='blue')
plt.title('Distribution of Temperature (C)')
plt.xlabel('Temperature (C)')
plt.ylabel('Frequency')
plt.show()
```



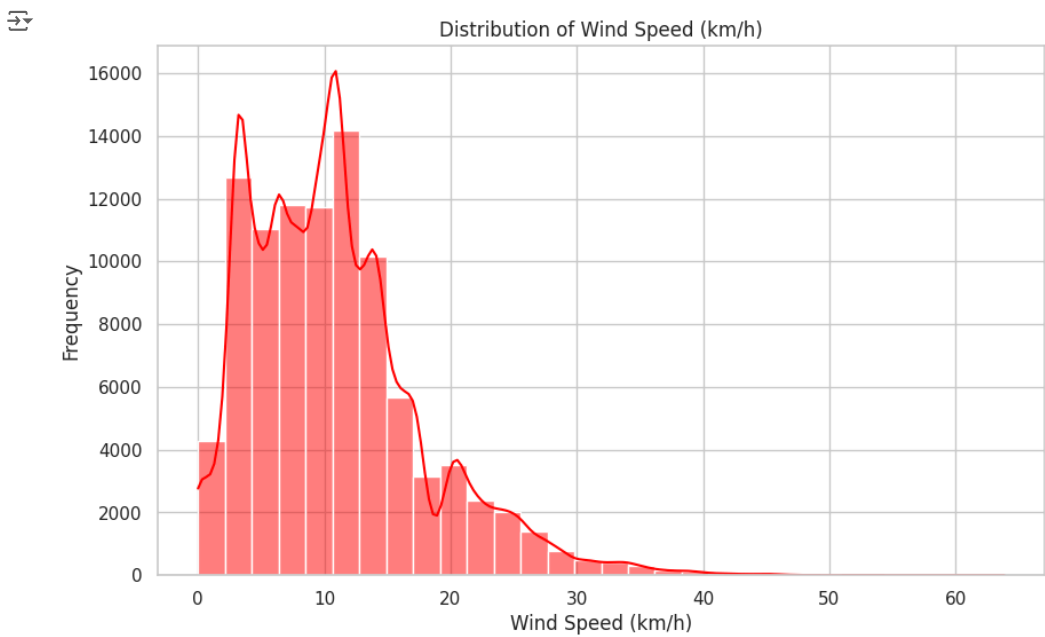
```
# Histogram for Apparent Temperature
plt.figure(figsize=(10, 6))
sns.histplot(data['Apparent Temperature (C)'], bins=30, kde=True, color='orange')
plt.title('Distribution of Apparent Temperature (C)')
plt.xlabel('Apparent Temperature (C)')
plt.ylabel('Frequency')
plt.show()
```



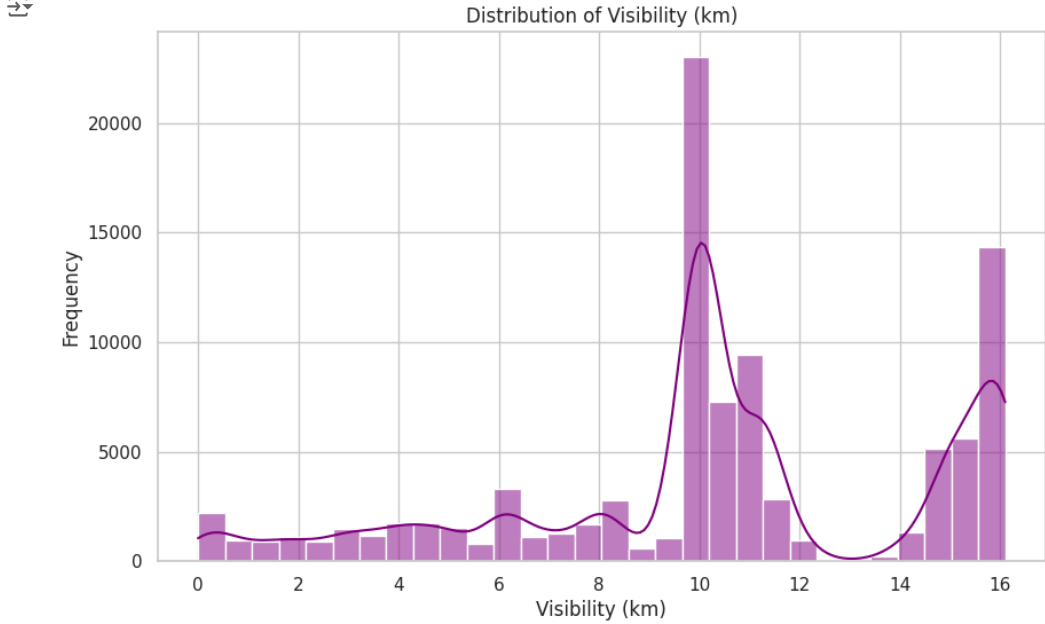
```
# Histogram for Humidity
plt.figure(figsize=(10, 6))
sns.histplot(data['Humidity'], bins=30, kde=True, color='green')
plt.title('Distribution of Humidity')
plt.xlabel('Humidity')
plt.ylabel('Frequency')
plt.show()
```



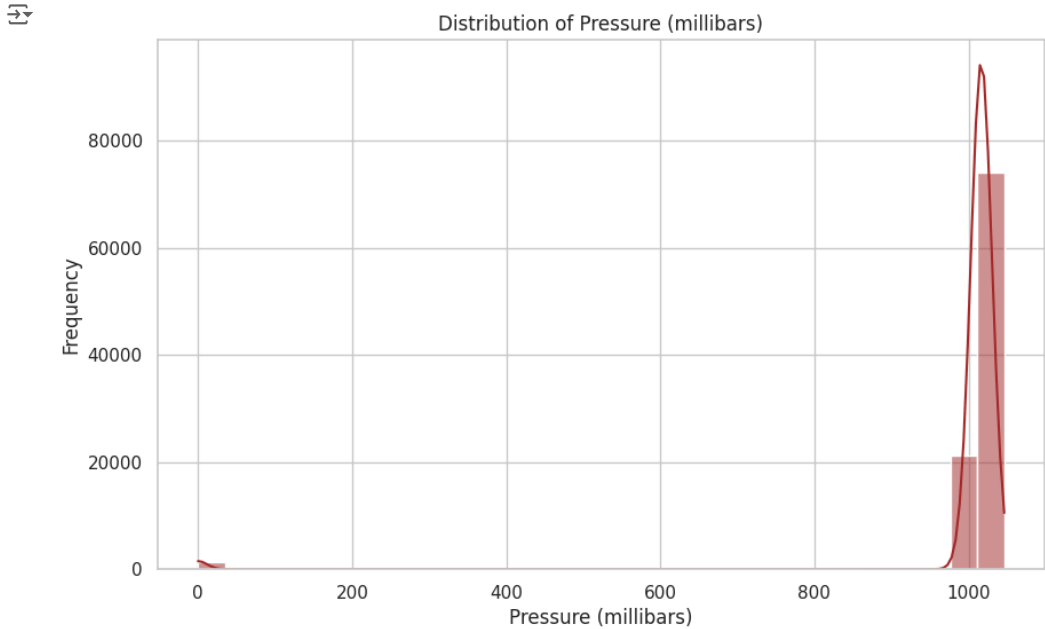
```
# Histogram for Wind Speed
plt.figure(figsize=(10, 6))
sns.histplot(data['Wind Speed (km/h)'], bins=30, kde=True, color='red')
plt.title('Distribution of Wind Speed (km/h)')
plt.xlabel('Wind Speed (km/h)')
plt.ylabel('Frequency')
plt.show()
```



```
# Histogram for Visibility
plt.figure(figsize=(10, 6))
sns.histplot(data['Visibility (km)'], bins=30, kde=True, color='purple')
plt.title('Distribution of Visibility (km)')
plt.xlabel('Visibility (km)')
plt.ylabel('Frequency')
plt.show()
```



```
# Histogram for Pressure (millibars)
plt.figure(figsize=(10, 6))
sns.histplot(data['Pressure (millibars)'], bins=30, kde=True, color='brown')
plt.title('Distribution of Pressure (millibars)')
plt.xlabel('Pressure (millibars)')
plt.ylabel('Frequency')
plt.show()
```



```
# Bar plot for Summary
plt.figure(figsize=(12, 6))
summary_counts = data['Summary'].value_counts().head(20) # Show top 20 most frequent summaries
sns.barplot(x=summary_counts, y=summary_counts.index, palette="viridis")
plt.title('Frequency of Different Weather Summaries')
plt.xlabel('Count')
plt.ylabel('Weather Summary')
plt.show()
```

<ipython-input-39-1bc56cfae52c>:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

