

```
import pandas as pd
```

```
# Load the weather dataset (assuming it's in CSV format)
```

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

```
# Explore the dataset
```

```
print(weather_data)
```

```

3650      0.22      1.68      0.2
3651      0.07      1.54      0.0
3652      0.17      0.57      0.1
3653      0.08      0.56      0.0

   BASEL_temp_mean  BASEL_temp_min  ...  STOCKHOLM_temp_min  \
0                2.9             1.6  ...                -9.3
1                3.6             2.7  ...                 0.5
2                2.2             0.1  ...                -1.0
3                3.9             0.5  ...                 2.5
4                6.0             3.8  ...                -1.8
...             ...             ...  ...             ...
3649             3.2             1.0  ...                -2.7
3650             4.5             2.4  ...                -9.5
3651             8.5             7.5  ...               -12.5
3652             6.6             4.3  ...                -9.3
3653             2.9            -0.2  ...                -8.8

   STOCKHOLM_temp_max  TOURS_wind_speed  TOURS_humidity  TOURS_pressure  \
0                   0.7                1.6             0.97           1.0275
1                   2.0                2.0             0.99           1.0293
2                   2.8                3.4             0.91           1.0267
3                   4.6                4.9             0.95           1.0222
4                   2.9                3.6             0.95           1.0209
...             ...             ...             ...             ...
3649             2.4                3.7             0.95           1.0011
3650             0.8                5.3             0.89           0.9966
3651            -7.4                3.8             0.88           0.9939
3652            -6.5                4.2             0.88           0.9933
3653            -7.0                3.4             0.86           1.0040

   TOURS_global_radiation  TOURS_precipitation  TOURS_temp_mean  \
0                   0.25                0.04                8.5
1                   0.17                0.16                7.9
2                   0.27                0.00                8.1
3                   0.11                0.44                8.6
4                   0.39                0.04                8.0
...             ...             ...             ...
3649             0.22                1.50                6.2
3650             0.24                0.40               10.4
3651             0.24                1.00               10.0
3652             0.58                0.02                8.5
3653             0.11                0.00                0.5

   TOURS_temp_min  TOURS_temp_max
0              7.2              9.8
1              6.6              9.2
2              6.6              9.6
3              6.4             10.8
4              6.4              9.5
...             ...             ...
3649             1.8             10.6
3650             6.2             14.5
3651             8.7             11.3
3652             6.2             10.9
3653            -0.7              1.8

```

```
[3654 rows x 165 columns]
```

```
import numpy as np
import pandas as pd

# Generate synthetic weather data
np.random.seed(0) # For reproducibility

# Define the number of samples
num_samples = 1000


# Generate random values for temperature, humidity, and wind speed
temperature = np.random.uniform(low=0, high=40, size=num_samples)
humidity = np.random.uniform(low=0, high=100, size=num_samples)
wind_speed = np.random.uniform(low=0, high=30, size=num_samples)



# Generate target variable (e.g., rainfall)
# Let's assume a simple linear relationship with temperature and humidity
rainfall = temperature * 0.5 + humidity * 0.3 + np.random.normal(loc=0, scale=5, size=num_samples)


# Create a DataFrame to store the data
weather_data = pd.DataFrame({
    'Temperature': temperature,
    'Humidity': humidity,
    'Wind_Speed': wind_speed,
    'Rainfall': rainfall
})

# Save the dataset to a CSV file
weather_data.to_csv('weather_prediction_dataset.csv', index=False)
```

```
weather_data.head()
```




	Temperature	Humidity	Wind_Speed	Rainfall	
0	21.952540	59.288027	24.345554	37.764992	
1	28.607575	1.006370	14.282520	8.630179	
2	24.110535	47.582620	15.694680	28.912591	
3	21.795327	70.877039	7.515618	34.207250	
4	16.946192	4.397543	18.151291	7.696949	

Next steps:  [View recommended plots](#)

```
import pandas as pd

# Load the weather dataset (assuming it's in CSV format)
weather_data = pd.read_csv('weather_prediction_bbq_labels.csv')

# Explore the dataset
print(weather_data)
```



	DATE	BASEL_BBQ_weather	BUDAPEST_BBQ_weather	DE_BBQ_weather	\
0	20000101	False	False	False	
1	20000102	False	False	False	
2	20000103	False	False	False	
3	20000104	False	False	False	
4	20000105	False	False	False	
...	
3649	20091228	False	False	False	
3650	20091229	False	False	False	
3651	20091230	False	False	False	
3652	20091231	False	False	False	
3653	20100101	False	False	False	
	DRESDEN_BBQ_weather	DUSSELDORF_BBQ_weather	HEATHROW_BBQ_weather	\	
0	False	False	False		
1	False	False	False		
2	False	False	False		

```

3          False          False          False
4          False          False          False
...          ...          ...          ...
3649       False          False          False
3650       False          False          False
3651       False          False          False
3652       False          False          False
3653       False          False          False

```

```

      KASSEL_BBQ_weather  LJUBLJANA_BBQ_weather  MAASTRICHT_BBQ_weather  \
0          False          False          False
1          False          False          False
2          False          False          False
3          False          False          False
4          False          False          False
...          ...          ...          ...
3649       False          False          False
3650       False          False          False
3651       False          False          False
3652       False          False          False
3653       False          False          False

```

```

      MALMO_BBQ_weather  MONTELIBMAR_BBQ_weather  MUENCHEN_BBQ_weather  \
0          False          False          False
1          False          False          False
2          False          False          False
3          False          False          False
4          False          False          False
...          ...          ...          ...
3649       False          False          False
3650       False          False          False
3651       False          False          False
3652       False          False          False
3653       False          False          False

```

```

      OSLO_BBQ_weather  PERPIGNAN_BBQ_weather  SONNBLICK_BBQ_weather  \
0          False          False          False
1          False          False          False
2          False          False          False
3          False          False          False
4          False          False          False

```

```
import pandas as pd
```

```
# Assuming bbq_labels is your DataFrame containing your data
```

```
# Load or create bbq_labels DataFrame
```

```
bbq_labels = pd.read_csv('weather_prediction_bbq_labels.csv') # Load your dataset
```

```
# Get the first few rows of bbq_labels DataFrame
```

```
df = bbq_labels.head()
```

```
# Print the first few rows
```

```
print(df)
```

```

→      DATE  BASEL_BBQ_weather  BUDAPEST_BBQ_weather  DE_BBQ_weather  \
0  20000101          False          False          False
1  20000102          False          False          False
2  20000103          False          False          False
3  20000104          False          False          False
4  20000105          False          False          False

```

```

      DRESDEN_BBQ_weather  DUSSELDORF_BBQ_weather  HEATHROW_BBQ_weather  \
0          False          False          False
1          False          False          False
2          False          False          False
3          False          False          False
4          False          False          False

```

```

      KASSEL_BBQ_weather  LJUBLJANA_BBQ_weather  MAASTRICHT_BBQ_weather  \
0          False          False          False
1          False          False          False
2          False          False          False
3          False          False          False
4          False          False          False

```

```

      MALMO_BBQ_weather  MONTELIBMAR_BBQ_weather  MUENCHEN_BBQ_weather  \

```

0	False	False	False
1	False	False	False
2	False	False	False
3	False	False	False
4	False	False	False

	OSLO_BBQ_weather	PERPIGNAN_BBQ_weather	SONNBLICK_BBQ_weather	\
0	False	False	False	
1	False	False	False	
2	False	False	False	
3	False	False	False	
4	False	False	False	

	STOCKHOLM_BBQ_weather	TOURS_BBQ_weather
0	False	False
1	False	False
2	False	False
3	False	False
4	False	False

```
merged_data = pd.merge(weather_data, bbq_labels, on='DATE')
merged_data .head()
```

	DATE	BASEL_BBQ_weather_x	BUDAPEST_BBQ_weather_x	DE_BBQ_weather_x	DRESDEN_BBQ_weather_x	DUSSELDORF_BBQ_
0	20000101	False	False	False	False	
1	20000102	False	False	False	False	
2	20000103	False	False	False	False	
3	20000104	False	False	False	False	
4	20000105	False	False	False	False	

5 rows × 35 columns

```
missing_counts = merged_data.isna().sum().sum()
missing_counts
```

0

```
merged_data.head()
```

	DATE	BASEL_BBQ_weather_x	BUDAPEST_BBQ_weather_x	DE_BBQ_weather_x	DRESDEN_
0	20000101	False	False	False	
1	20000102	False	False	False	
2	20000103	False	False	False	
3	20000104	False	False	False	
4	20000105	False	False	False	

5 rows × 35 columns

```

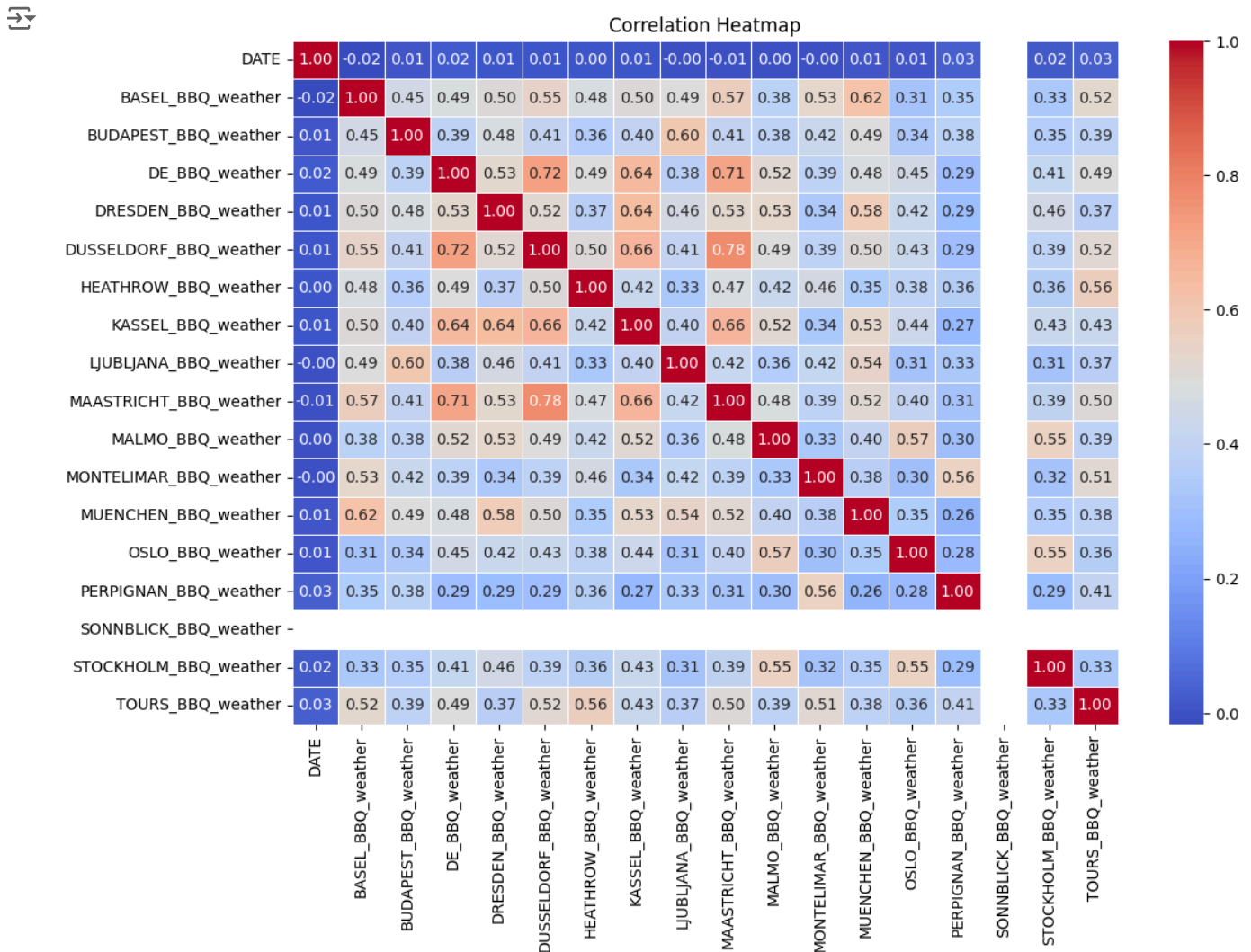
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming bbq_labels is your DataFrame containing your data
# Load or create bbq_labels DataFrame
bbq_labels = pd.read_csv('weather_prediction_bbq_labels.csv') # Load your dataset

# Calculate the correlation matrix
correlation_matrix = bbq_labels.corr()

# Plot the heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
plt.title('Correlation Heatmap')
plt.show()

```



```

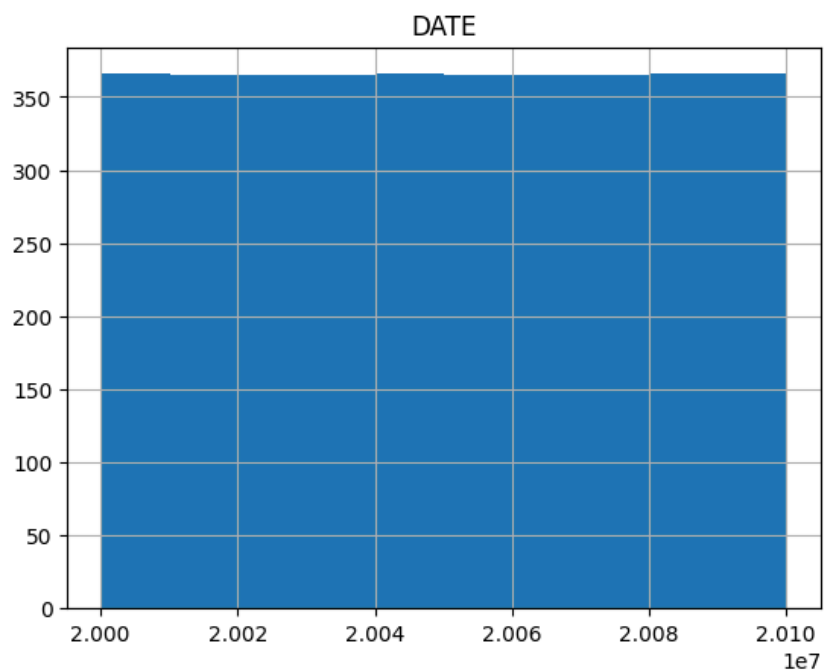
-----
NameError                                Traceback (most recent call last)
<ipython-input-23-5cbd3fd13044> in <cell line: 4>()
      2
      3 # Calculate Pearson correlation coefficients
----> 4 correlations = X.corrwith(y)
      5
      6 # Absolute values of correlations for better interpretation

NameError: name 'X' is not defined

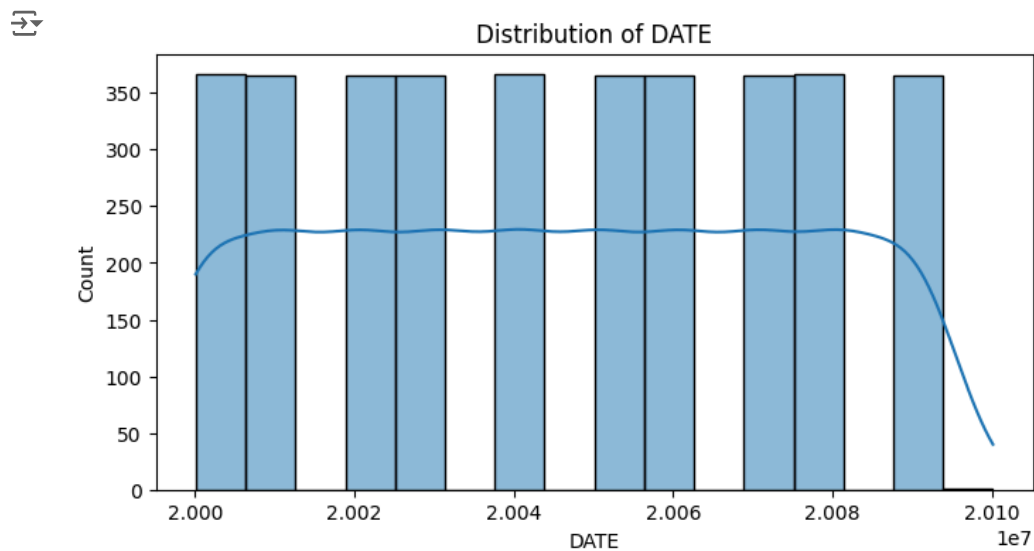
```

```
bbq_labels.hist()
```

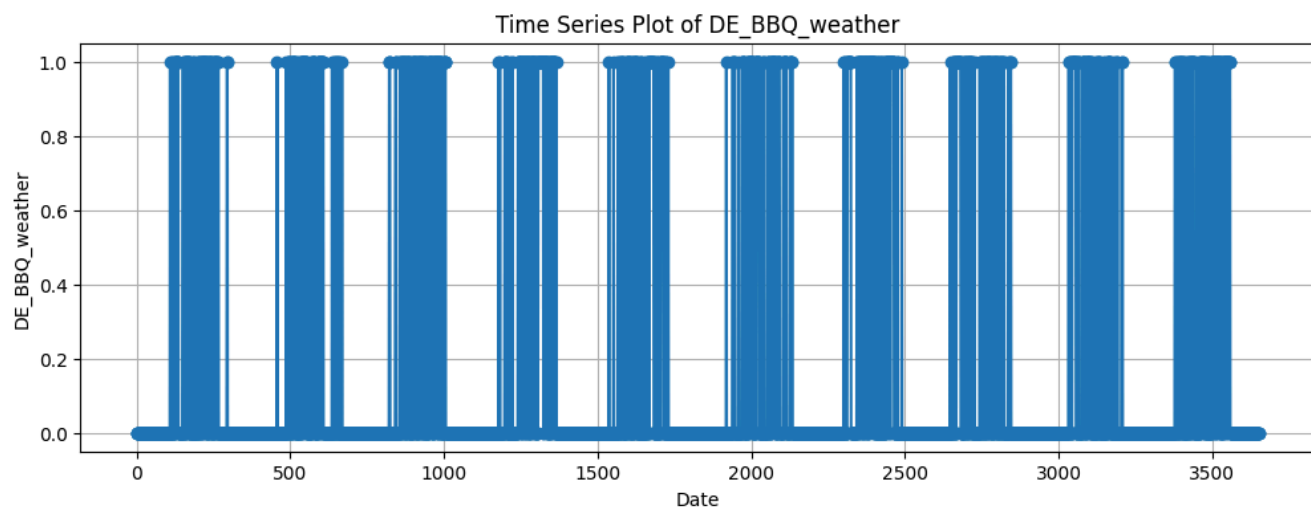
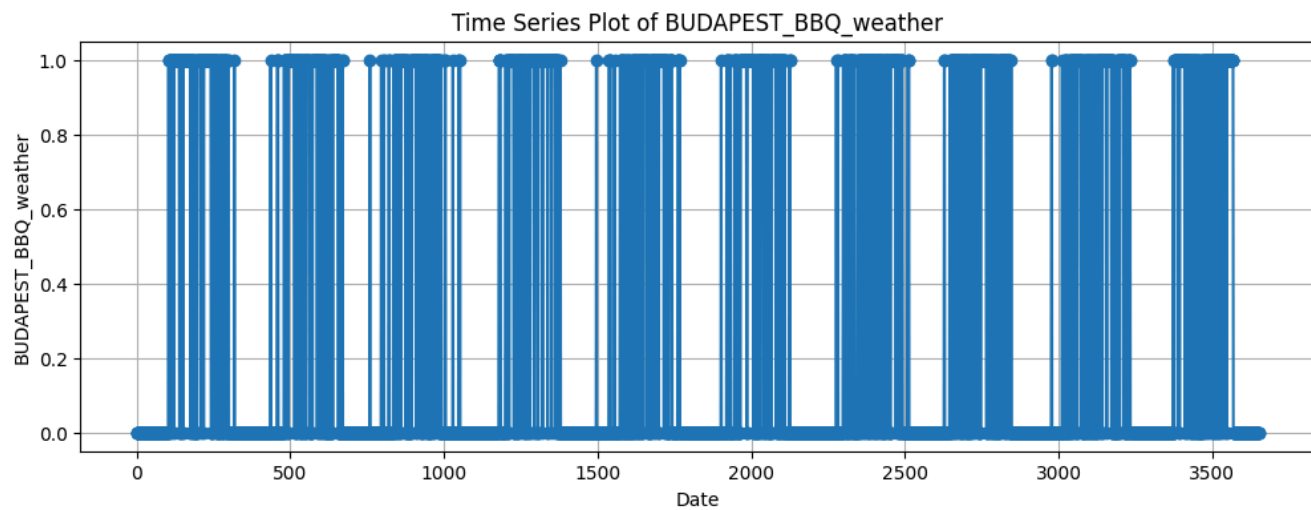
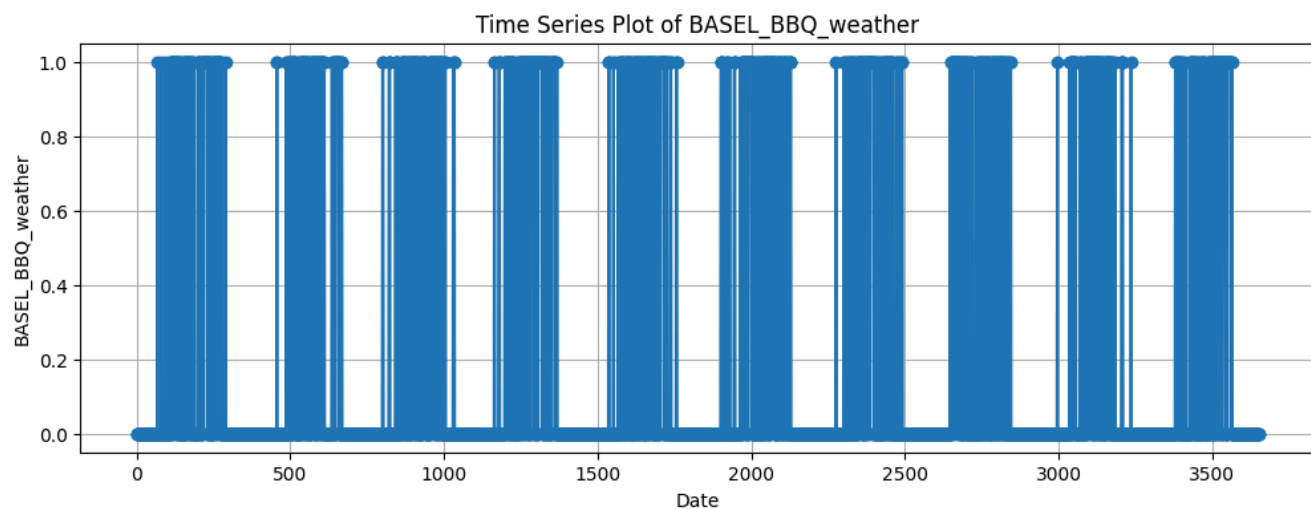
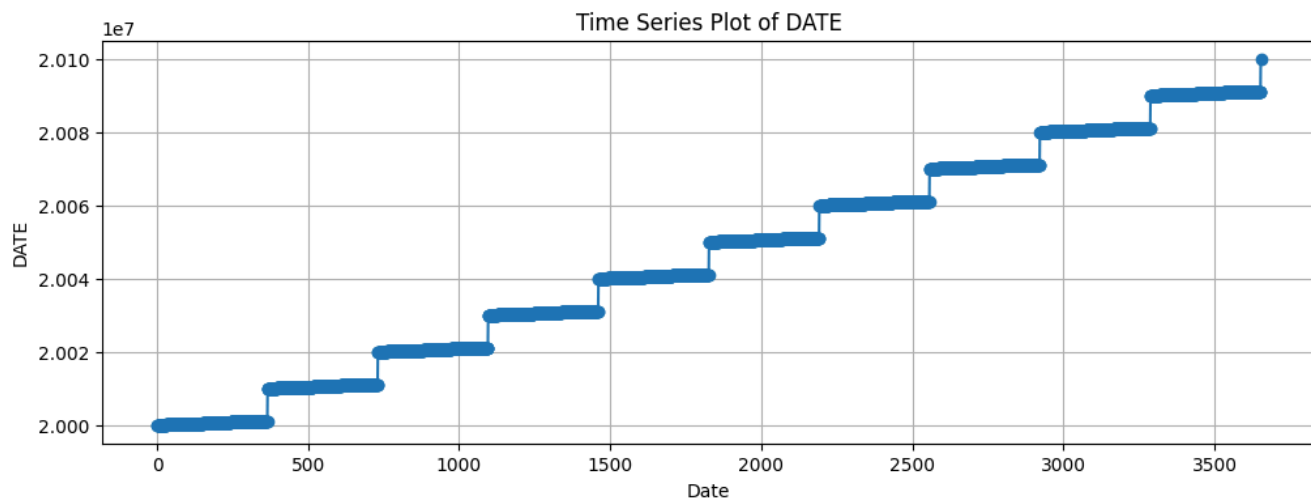
```
array([[<Axes: title={'center': 'DATE'}>]], dtype=object)
```

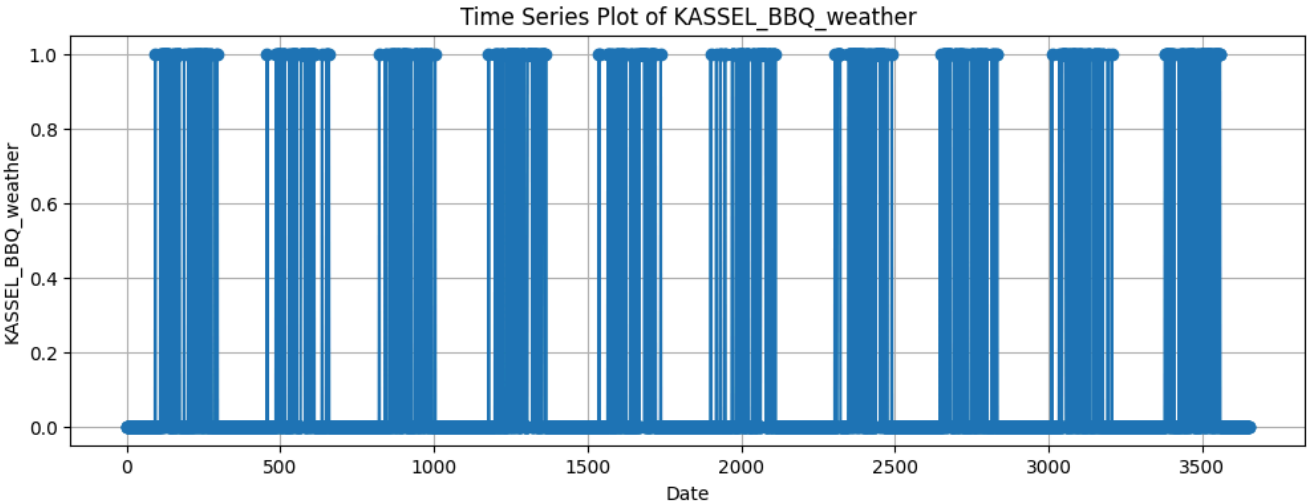
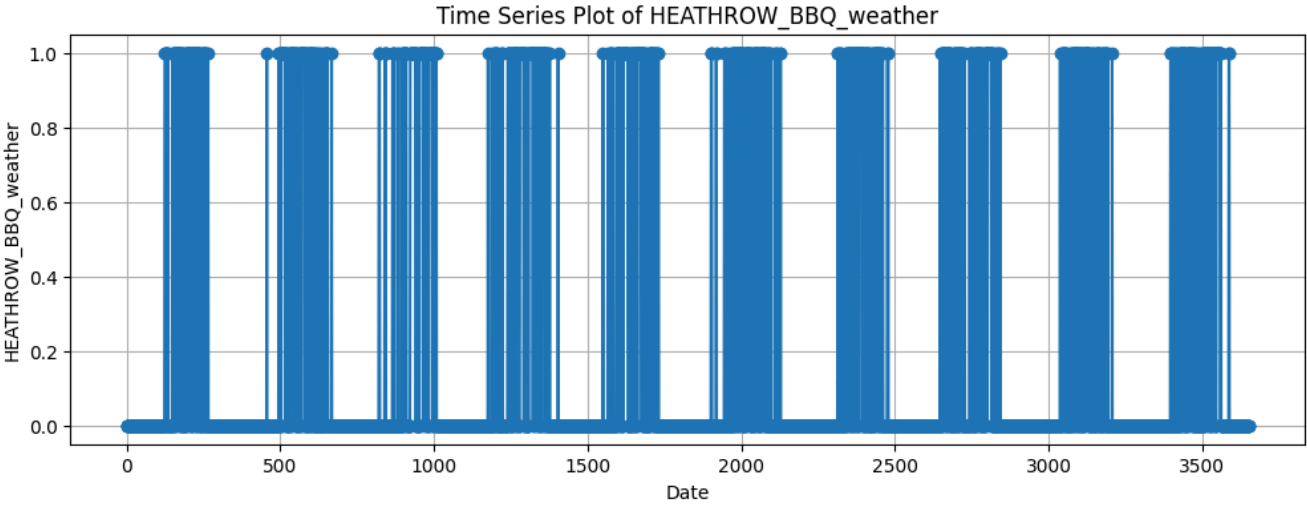
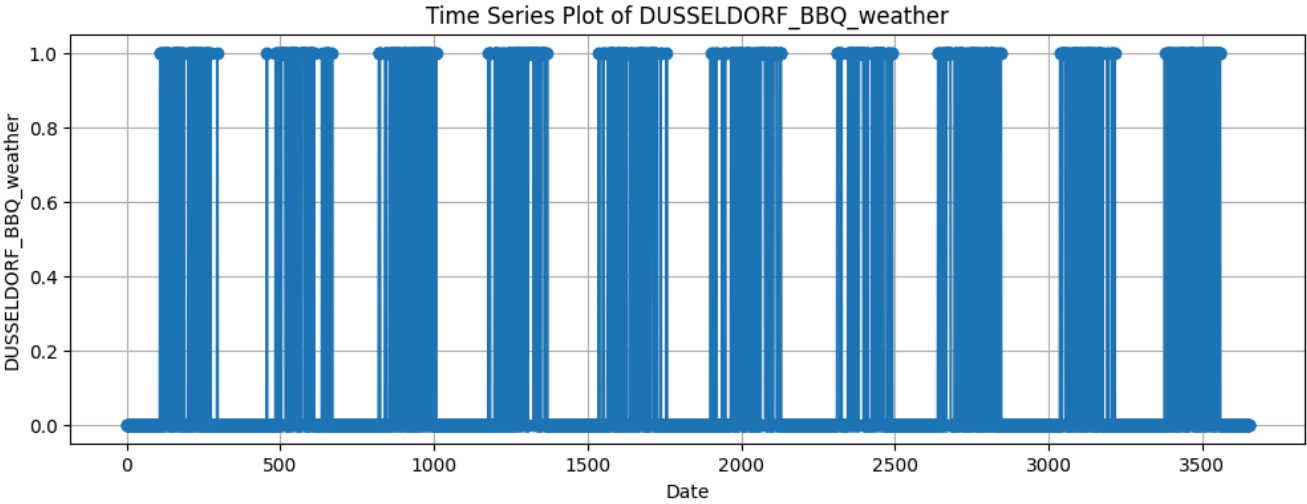
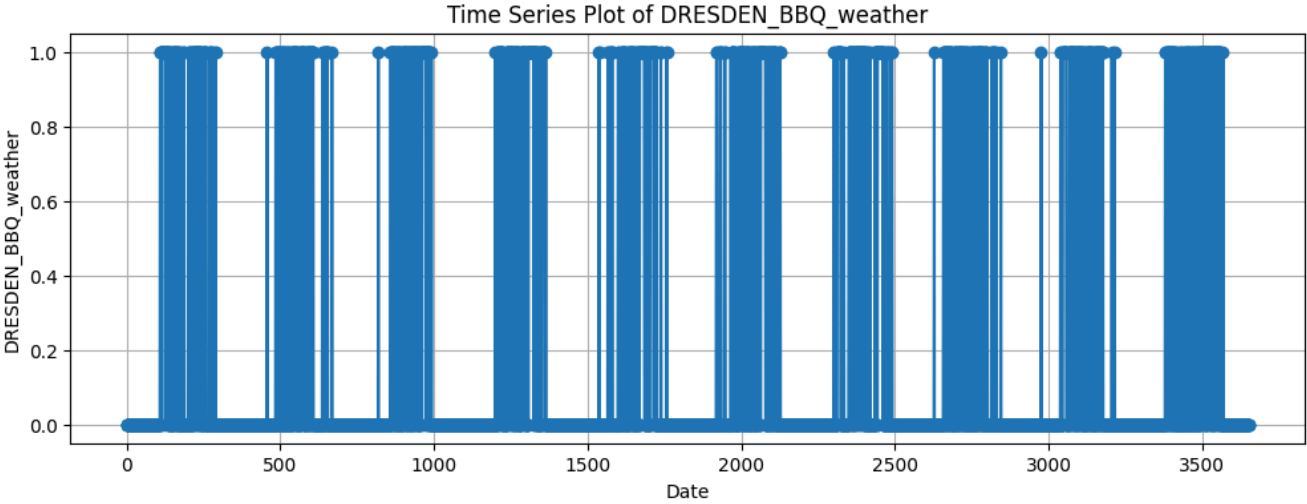


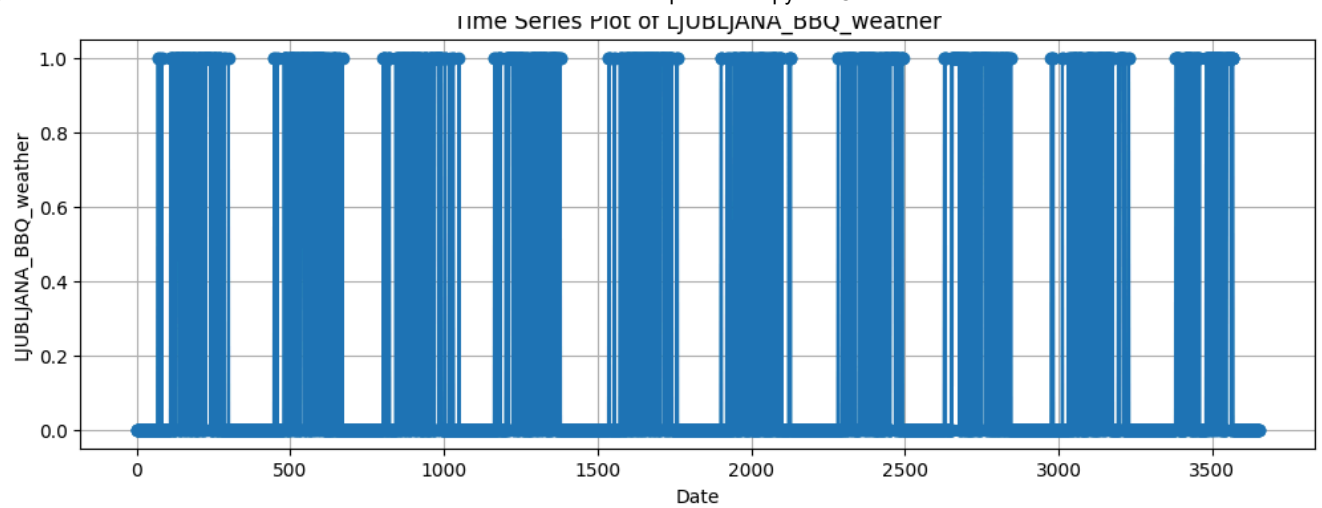
```
import matplotlib.pyplot as plt
# Plot distribution of each numerical feature
for column in bbq_labels.select_dtypes(include=['float64', 'int64']).columns:
    plt.figure(figsize=(8, 4))
    sns.histplot(bbq_labels[column].dropna(), kde=True)
    plt.title(f'Distribution of {column}')
    plt.show()
```



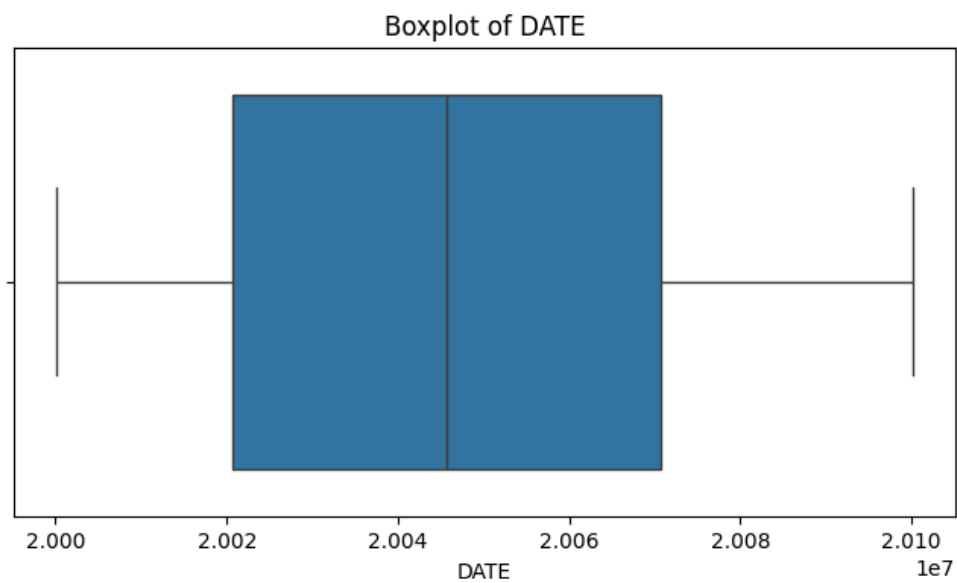
```
for column in bbq_labels.columns:
    plt.figure(figsize=(12, 4))
    plt.plot(bbq_labels.index, bbq_labels[column], marker='o', linestyle='--')
    plt.title(f'Time Series Plot of {column}')
    plt.xlabel('Date')
    plt.ylabel(column)
    plt.grid(True)
    plt.show()
```







```
# Plot boxplots for each numerical feature
for column in bbq_labels.select_dtypes(include=['float64', 'int64']).columns:
    plt.figure(figsize=(8, 4))
    sns.boxplot(x=bbq_labels[column])
    plt.title(f'Boxplot of {column}')
    plt.show()
```



```
df.columns
```