```
#import necessary libraries
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

weather_data = pd.read_csv("weather.csv")

weather_data.head()
```

```
   MinTemp  MaxTemp  Rainfall  Evaporation  Sunshine WindGustDir  \
0      8.0     24.3       0.0          3.4       6.3          NW
1     14.0     26.9       3.6          4.4       9.7         ENE
2     13.7     23.4       3.6          5.8       3.3          NW
3     13.3     15.5      39.8          7.2       9.1          NW
4      7.6     16.1       2.8          5.6      10.6         SSE

   WindGustSpeed WindDir9am WindDir3pm  WindSpeed9am  ...  Humidity3pm
\
0           30.0         SW         NW           6.0  ...           29

1           39.0          E          W           4.0  ...           36

2           85.0          N        NNE           6.0  ...           69

3           54.0        WNW          W          30.0  ...           56

4           50.0        SSE        ESE          20.0  ...           49


   Pressure9am  Pressure3pm  Cloud9am  Cloud3pm  Temp9am  Temp3pm
RainToday  \
0       1019.7       1015.0         7         7     14.4     23.6
No
1       1012.4       1008.4         5         3     17.5     25.7
Yes
2       1009.5       1007.2         8         7     15.4     20.2
Yes
3       1005.5       1007.0         2         7     13.5     14.1
Yes
4       1018.3       1018.5         7         7     11.1     15.4
Yes

   RISK_MM RainTomorrow
0      3.6          Yes
1      3.6          Yes
2     39.8          Yes
3      2.8          Yes
4      0.0           No
```

```
[5 rows x 22 columns]
```

```
weather_data.shape
```

```
(366, 22)
```

Data Cleaning: Checking and removing Duplicate values and Missing values

```
#Data Cleaning
duplicate_rows = weather_data[weather_data.duplicated()]

print(duplicate_rows)
```

```
Empty DataFrame
Columns: [MinTemp, MaxTemp, Rainfall, Evaporation, Sunshine,
WindGustDir, WindGustSpeed, WindDir9am, WindDir3pm, WindSpeed9am,
WindSpeed3pm, Humidity9am, Humidity3pm, Pressure9am, Pressure3pm,
Cloud9am, Cloud3pm, Temp9am, Temp3pm, RainToday, RISK_MM,
RainTomorrow]
Index: []

[0 rows x 22 columns]
```

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

```
MinTemp           0
MaxTemp           0
Rainfall          0
Evaporation       0
Sunshine          3
WindGustDir       3
WindGustSpeed     2
WindDir9am       31
WindDir3pm        1
WindSpeed9am      7
WindSpeed3pm      0
Humidity9am       0
Humidity3pm       0
Pressure9am       0
Pressure3pm       0
Cloud9am          0
Cloud3pm          0
Temp9am           0
Temp3pm           0
RainToday         0
RISK_MM           0
RainTomorrow      0
dtype: int64
```

As you can see there are missing values in the data set

```
data_clean = weather_data.dropna()

data_clean
```

```
     MinTemp  MaxTemp  Rainfall  Evaporation  Sunshine WindGustDir  \
0        8.0     24.3       0.0          3.4       6.3          NW
1       14.0     26.9       3.6          4.4       9.7         ENE
2       13.7     23.4       3.6          5.8       3.3          NW
3       13.3     15.5      39.8          7.2       9.1          NW
4        7.6     16.1       2.8          5.6      10.6         SSE
..       ...      ...       ...          ...       ...         ...
361      9.0     30.7       0.0          7.6      12.1         NNW
362      7.1     28.4       0.0         11.6      12.7           N
363     12.5     19.9       0.0          8.4       5.3         ESE
364     12.5     26.9       0.0          5.0       7.1          NW
365     12.3     30.2       0.0          6.0      12.6          NW

     WindGustSpeed WindDir9am WindDir3pm  WindSpeed9am  ...
Humidity3pm  \
0             30.0         SW         NW           6.0  ...
29
1             39.0          E          W           4.0  ...
36
2             85.0          N        NNE           6.0  ...
69
3             54.0        WNW          W          30.0  ...
56
4             50.0        SSE        ESE          20.0  ...
49
..             ...        ...        ...           ...  ...          .
..
361           76.0        SSE         NW           7.0  ...
15
362           48.0        NNW        NNW           2.0  ...
22
363           43.0        ENE        ENE          11.0  ...
47
364           46.0        SSW        WNW           6.0  ...
39
365           78.0         NW        WNW          31.0  ...
13

     Pressure9am  Pressure3pm  Cloud9am  Cloud3pm  Temp9am  Temp3pm  \
0         1019.7       1015.0         7         7     14.4     23.6
1         1012.4       1008.4         5         3     17.5     25.7
2         1009.5       1007.2         8         7     15.4     20.2
3         1005.5       1007.0         2         7     13.5     14.1
4         1018.3       1018.5         7         7     11.1     15.4
..           ...          ...       ...       ...      ...      ...
361       1016.1       1010.8         1         3     20.4     30.0
```

```
362       1020.0       1016.9         0         1     17.2     28.2
363       1024.0       1022.8         3         2     14.5     18.3
364       1021.0       1016.2         6         7     15.8     25.9
365       1009.6       1009.2         1         1     23.8     28.6

     RainToday  RISK_MM RainTomorrow
0           No      3.6          Yes
1          Yes      3.6          Yes
2          Yes     39.8          Yes
3          Yes      2.8          Yes
4          Yes      0.0           No
..         ...      ...          ...
361         No      0.0           No
362         No      0.0           No
363         No      0.0           No
364         No      0.0           No
365         No      0.0           No

[328 rows x 22 columns]

data_clean.info()

<class 'pandas.core.frame.DataFrame'>
Index: 328 entries, 0 to 365
Data columns (total 22 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   MinTemp        328 non-null    float64
 1   MaxTemp        328 non-null    float64
 2   Rainfall       328 non-null    float64
 3   Evaporation    328 non-null    float64
 4   Sunshine       328 non-null    float64
 5   WindGustDir    328 non-null    object
 6   WindGustSpeed  328 non-null    float64
 7   WindDir9am     328 non-null    object
 8   WindDir3pm     328 non-null    object
 9   WindSpeed9am   328 non-null    float64
 10  WindSpeed3pm   328 non-null    int64
 11  Humidity9am    328 non-null    int64
 12  Humidity3pm    328 non-null    int64
 13  Pressure9am    328 non-null    float64
 14  Pressure3pm    328 non-null    float64
 15  Cloud9am       328 non-null    int64
 16  Cloud3pm       328 non-null    int64
 17  Temp9am        328 non-null    float64
 18  Temp3pm        328 non-null    float64
 19  RainToday      328 non-null    object
 20  RISK_MM        328 non-null    float64
 21  RainTomorrow   328 non-null    object
```

```
dtypes: float64(12), int64(5), object(5)
memory usage: 58.9+ KB
```

```
cleaned_df[["WindSpeed3pm", "Humidity9am", "Humidity3pm", "Cloud9am",
"Cloud3pm"]] = cleaned_df[["WindSpeed3pm", "Humidity9am",
"Humidity3pm", "Cloud9am", "Cloud3pm"]].astype(float)
data_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 328 entries, 0 to 365
Data columns (total 22 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   MinTemp        328 non-null    float64
 1   MaxTemp        328 non-null    float64
 2   Rainfall       328 non-null    float64
 3   Evaporation    328 non-null    float64
 4   Sunshine       328 non-null    float64
 5   WindGustDir    328 non-null    object
 6   WindGustSpeed  328 non-null    float64
 7   WindDir9am     328 non-null    object
 8   WindDir3pm     328 non-null    object
 9   WindSpeed9am   328 non-null    float64
 10  WindSpeed3pm   328 non-null    float64
 11  Humidity9am    328 non-null    float64
 12  Humidity3pm    328 non-null    float64
 13  Pressure9am    328 non-null    float64
 14  Pressure3pm    328 non-null    float64
 15  Cloud9am       328 non-null    float64
 16  Cloud3pm       328 non-null    float64
 17  Temp9am        328 non-null    float64
 18  Temp3pm        328 non-null    float64
 19  RainToday      328 non-null    object
 20  RISK_MM        328 non-null    float64
 21  RainTomorrow   328 non-null    object
dtypes: float64(17), object(5)
memory usage: 58.9+ KB
```

Adding Date column to the dataset as we need it in the further process As there is 366 rows we can consider it as a Leap year and start the data with "2020-01-01"

```
from datetime import timedelta

start_date = pd.to_datetime('2020-01-01')
interval = timedelta(days = 1)
num_records = 366

date_range = pd.date_range(start=start_date, periods=num_records,
freq=interval)
Weather_data = pd.DataFrame({'Date': date_range})
```

```
merged_data = pd.concat([Weather_data, weather_data], axis=1)

print(merged_data)
```

```
          Date  MinTemp  MaxTemp  Rainfall  Evaporation  Sunshine
WindGustDir  \
0   2020-01-01      8.0     24.3       0.0          3.4       6.3
NW
1   2020-01-02     14.0     26.9       3.6          4.4       9.7
ENE
2   2020-01-03     13.7     23.4       3.6          5.8       3.3
NW
3   2020-01-04     13.3     15.5      39.8          7.2       9.1
NW
4   2020-01-05      7.6     16.1       2.8          5.6      10.6
SSE
..         ...      ...      ...       ...          ...       ...
...
361 2020-12-27      9.0     30.7       0.0          7.6      12.1
NNW
362 2020-12-28      7.1     28.4       0.0         11.6      12.7
N
363 2020-12-29     12.5     19.9       0.0          8.4       5.3
ESE
364 2020-12-30     12.5     26.9       0.0          5.0       7.1
NW
365 2020-12-31     12.3     30.2       0.0          6.0      12.6
NW

     WindGustSpeed WindDir9am WindDir3pm  ...  Humidity3pm
Pressure9am  \
0             30.0         SW         NW  ...           29
1019.7
1             39.0          E          W  ...           36
1012.4
2             85.0          N        NNE  ...           69
1009.5
3             54.0        WNW          W  ...           56
1005.5
4             50.0        SSE        ESE  ...           49
1018.3
..             ...        ...        ...  ...          ...          ..
.
361           76.0        SSE         NW  ...           15
1016.1
362           48.0        NNW        NNW  ...           22
1020.0
363           43.0        ENE        ENE  ...           47
1024.0
```

```
364           46.0         SSW         WNW  ...           39
1021.0
365           78.0          NW         WNW  ...           13
1009.6

     Pressure3pm  Cloud9am  Cloud3pm  Temp9am  Temp3pm  RainToday
RISK_MM  \
0         1015.0         7         7     14.4     23.6         No
3.6
1         1008.4         5         3     17.5     25.7        Yes
3.6
2         1007.2         8         7     15.4     20.2        Yes
39.8
3         1007.0         2         7     13.5     14.1        Yes
2.8
4         1018.5         7         7     11.1     15.4        Yes
0.0
..           ...       ...       ...      ...      ...        ...
...
361       1010.8         1         3     20.4     30.0         No
0.0
362       1016.9         0         1     17.2     28.2         No
0.0
363       1022.8         3         2     14.5     18.3         No
0.0
364       1016.2         6         7     15.8     25.9         No
0.0
365       1009.2         1         1     23.8     28.6         No
0.0

     RainTomorrow
0             Yes
1             Yes
2             Yes
3             Yes
4              No
..            ...
361            No
362            No
363            No
364            No
365            No

[366 rows x 23 columns]
```

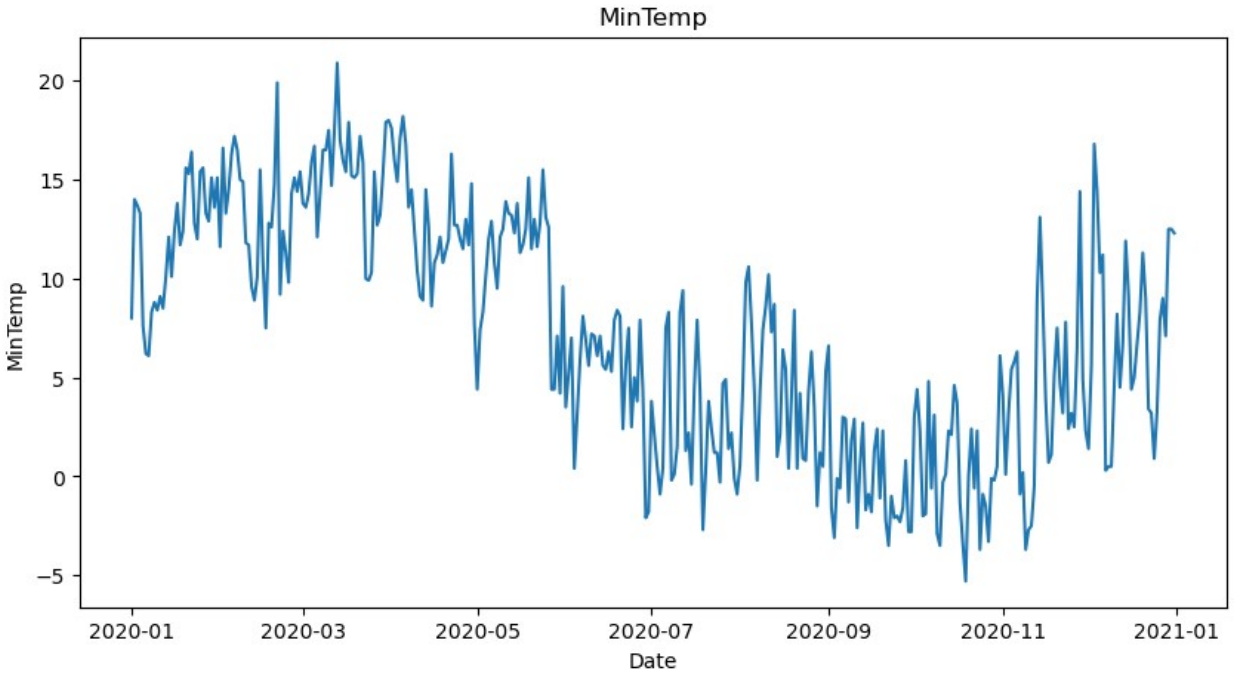
1. Descriptive Statistics

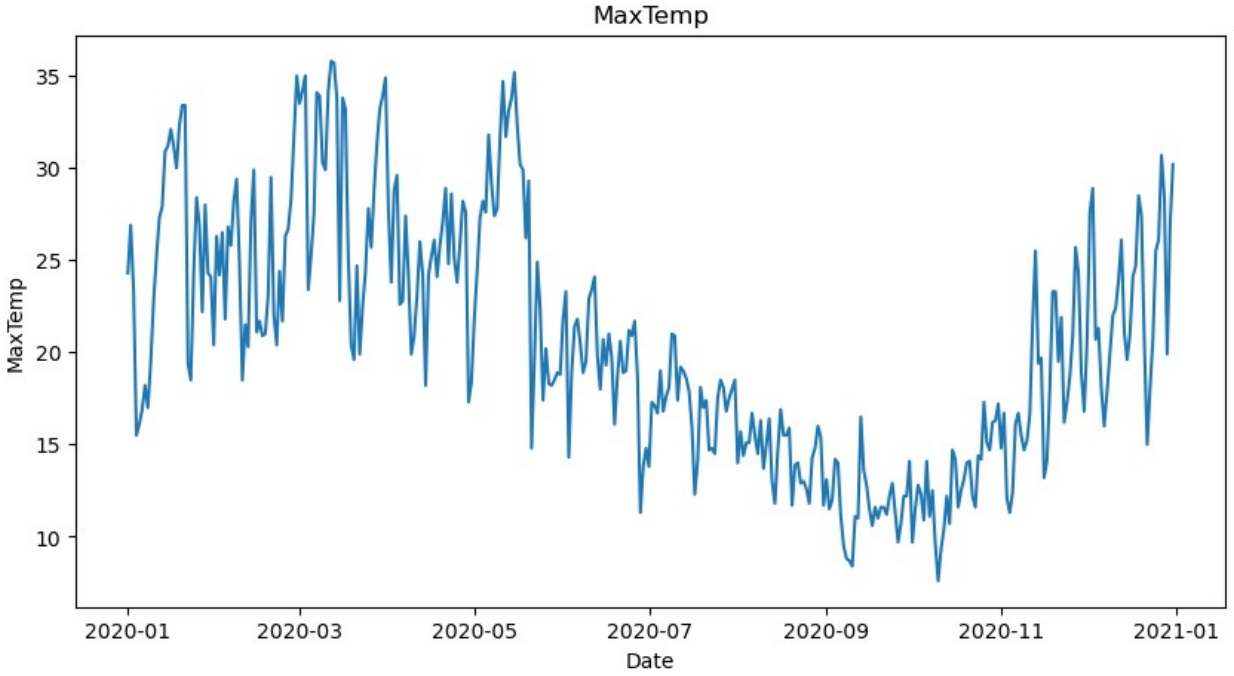
```
data_clean[["MinTemp", "MaxTemp", "Rainfall",
"Evaporation"]].describe().T
```

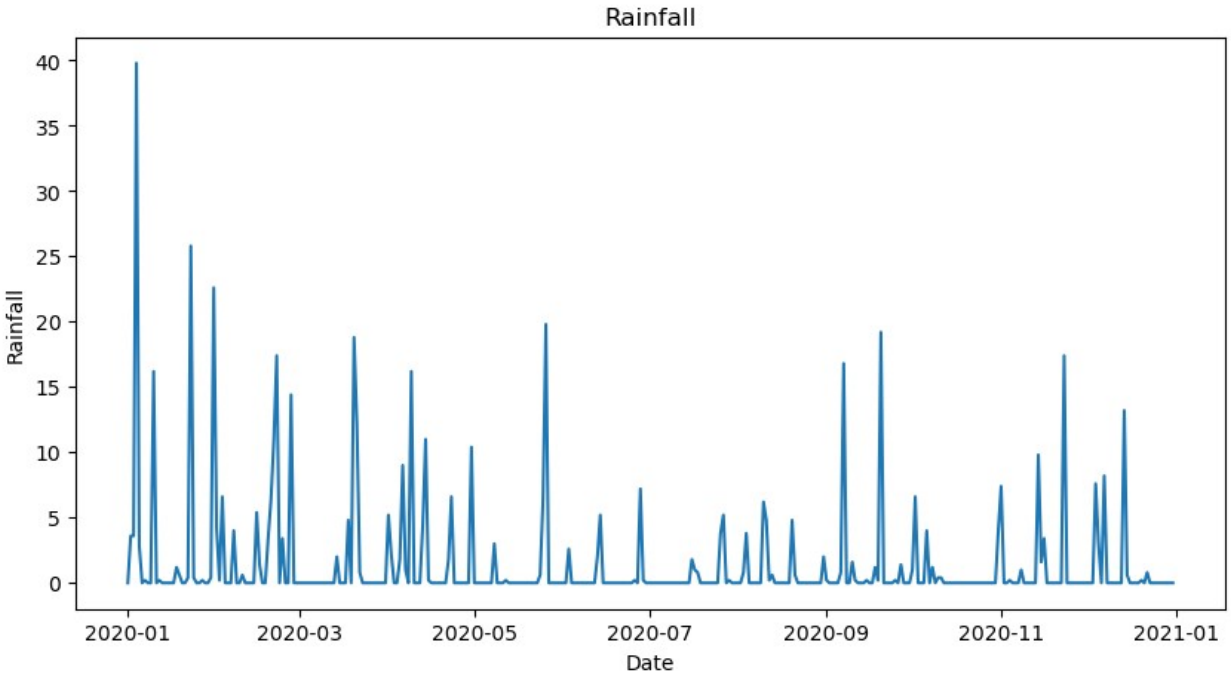
	count	mean	std	min	25%	50%	75%	max
MinTemp	328.0	7.742988	5.945199	-5.3	2.85	7.9	12.8	20.9
MaxTemp	328.0	20.897561	6.707310	7.6	15.50	20.4	25.8	35.8
Rainfall	328.0	1.440854	4.289427	0.0	0.00	0.0	0.2	39.8
Evaporation	328.0	4.702439	2.681183	0.2	2.55	4.4	6.6	13.8

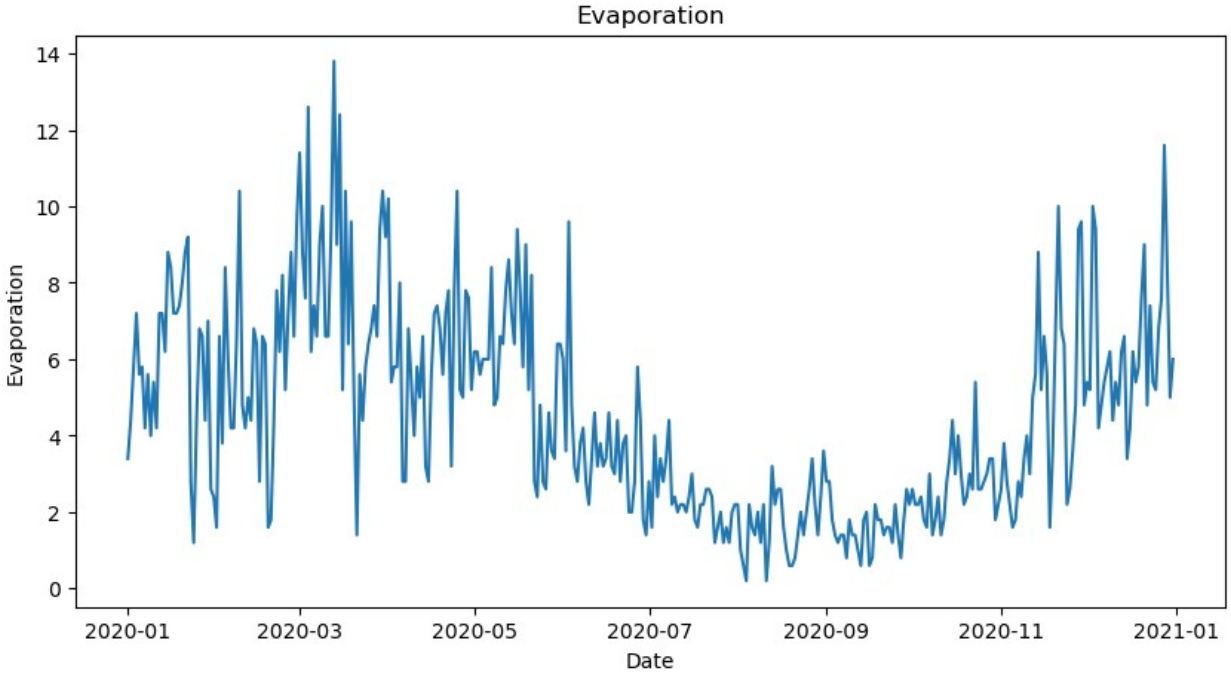
2. Time Series Visualization

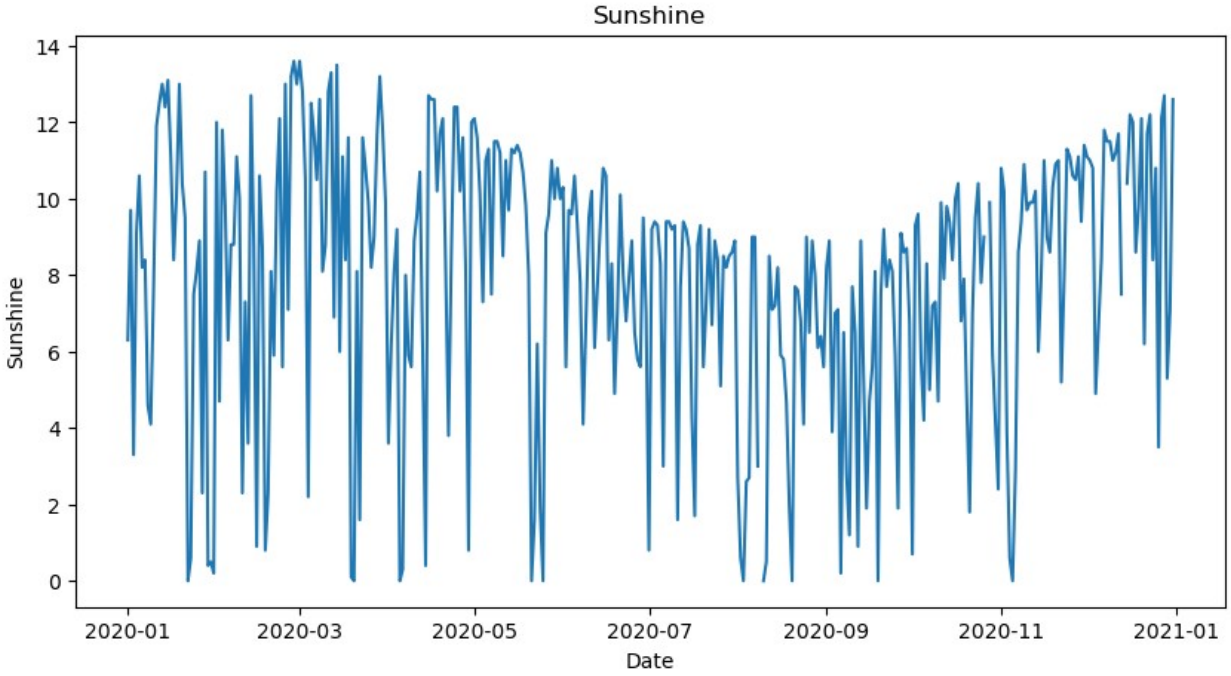
```
dates = merged_data["Date"]

columns_to_plot = ['MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation',
'Sunshine',
                   'WindGustSpeed', 'WindSpeed9am', 'WindSpeed3pm',
'Humidity9am',
                   'Humidity3pm', 'Pressure9am', 'Pressure3pm',
'Cloud9am', 'Cloud3pm',
                   'Temp9am', 'Temp3pm', 'RISK_MM']

for column in columns_to_plot:
    plt.figure(figsize=(10, 5))
    plt.plot(dates, merged_data[column])
    plt.title(column)
    plt.xlabel('Date')
    plt.ylabel(column)
    plt.show()
```

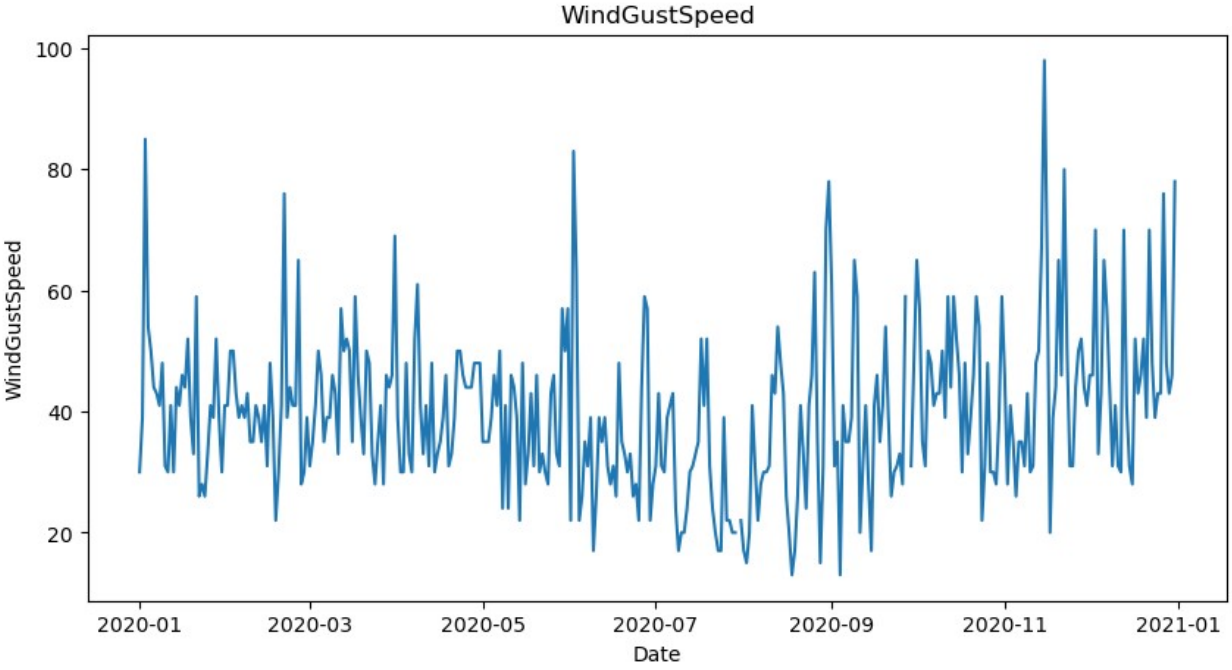

MinTemp
MinTemp
20
15
10
5
0
−5
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01
Date

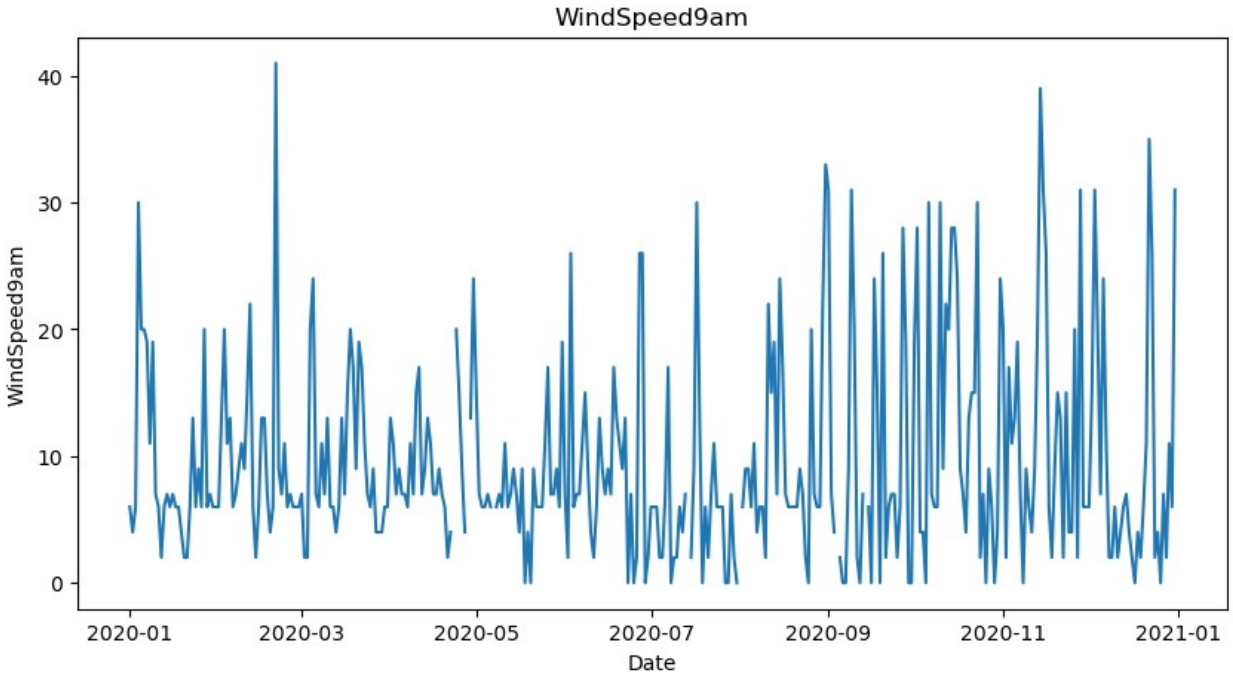

MaxTemp
MaxTemp
35
30
25
20
15
10
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01
Date

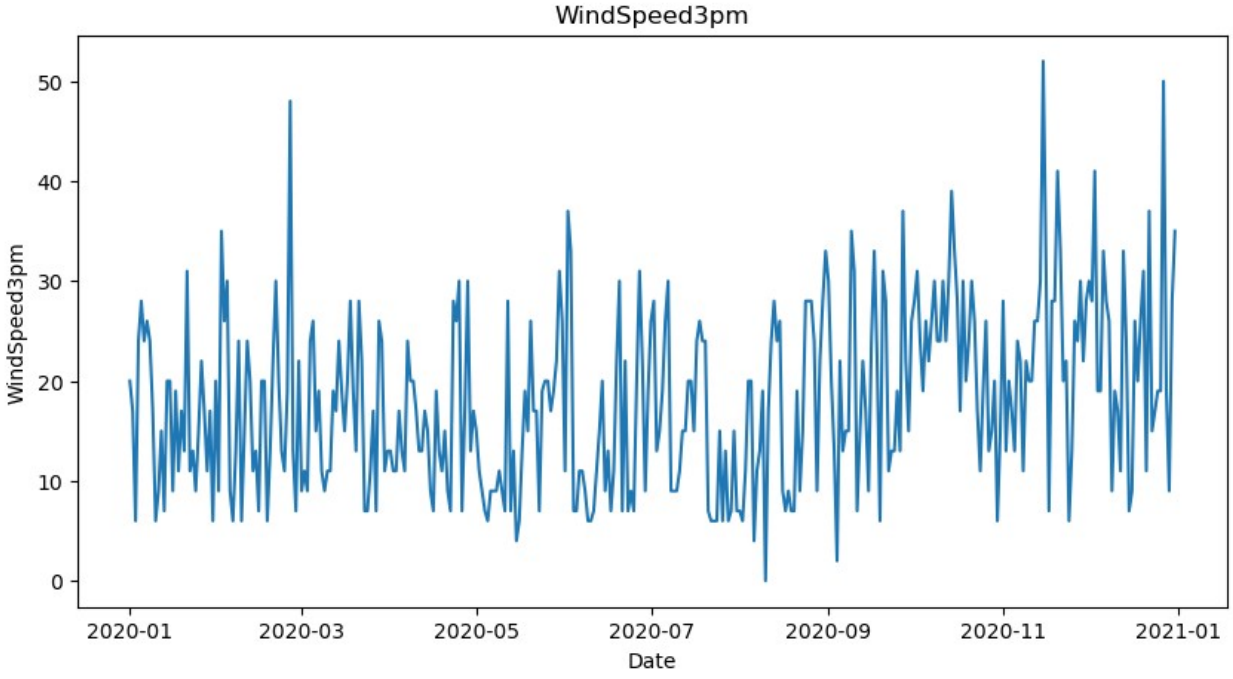

Rainfall
Rainfall
40
35
30
25
20
15
10
5
0
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01
Date

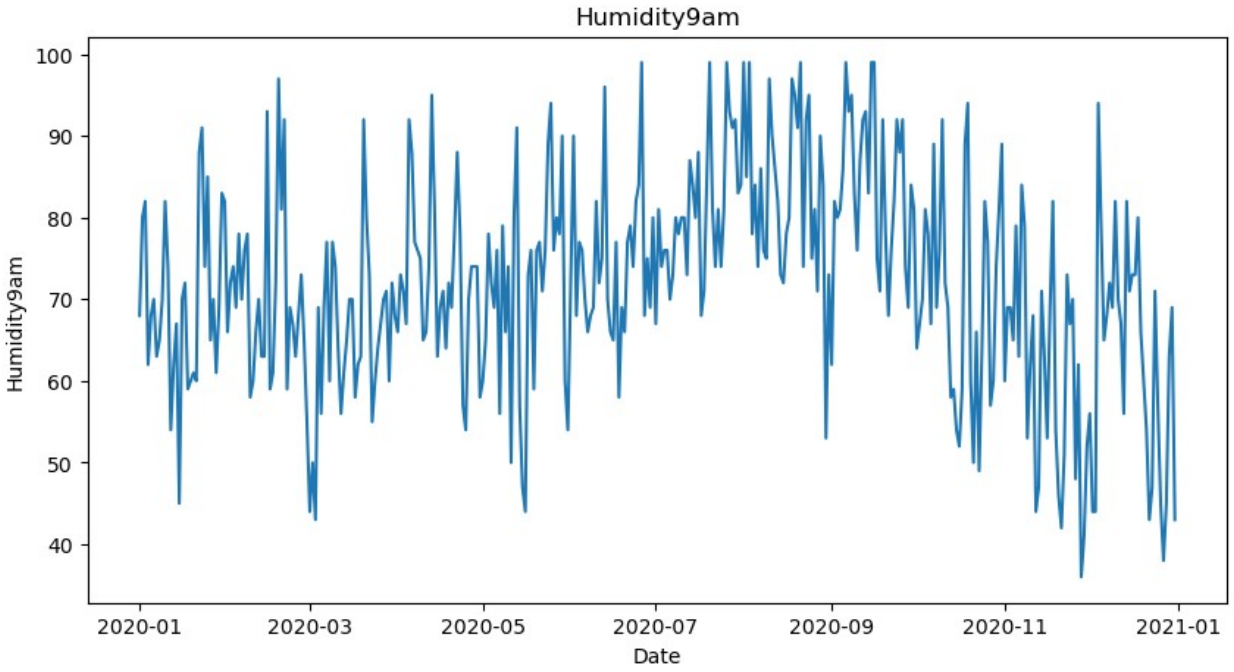

Evaporation
Evaporation
14
12
10
8
6
4
2
0
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01
Date

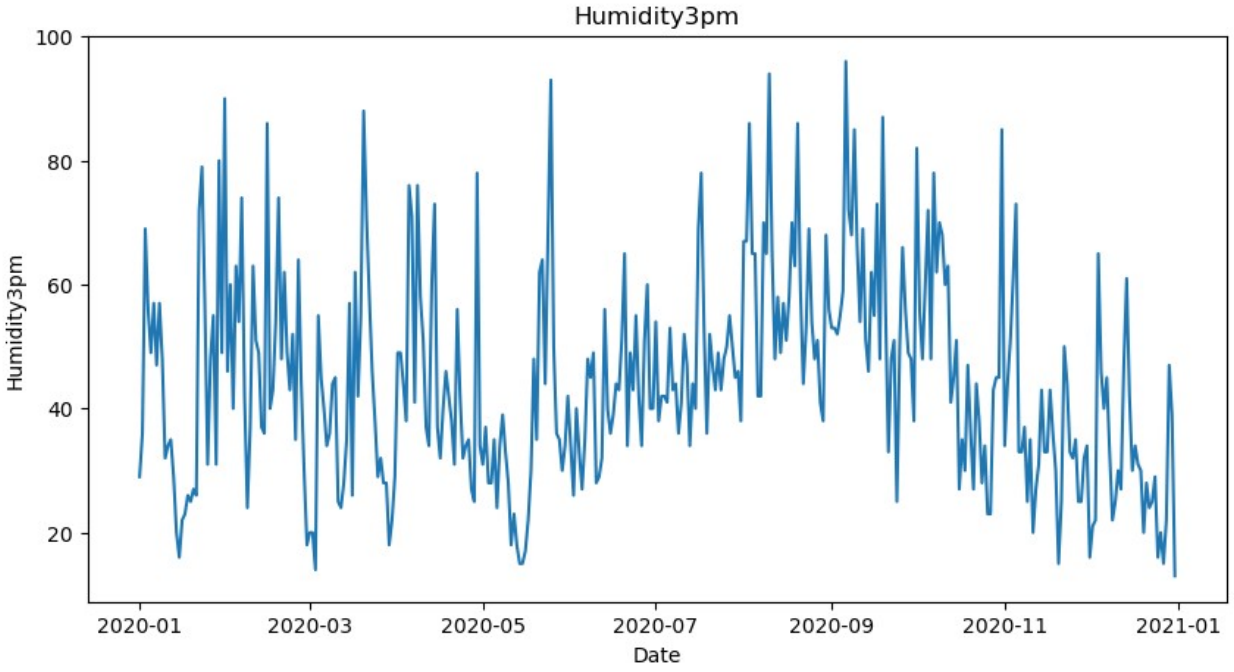

Sunshine
Sunshine
Date
0
2
4
6
8
10
12
14
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01

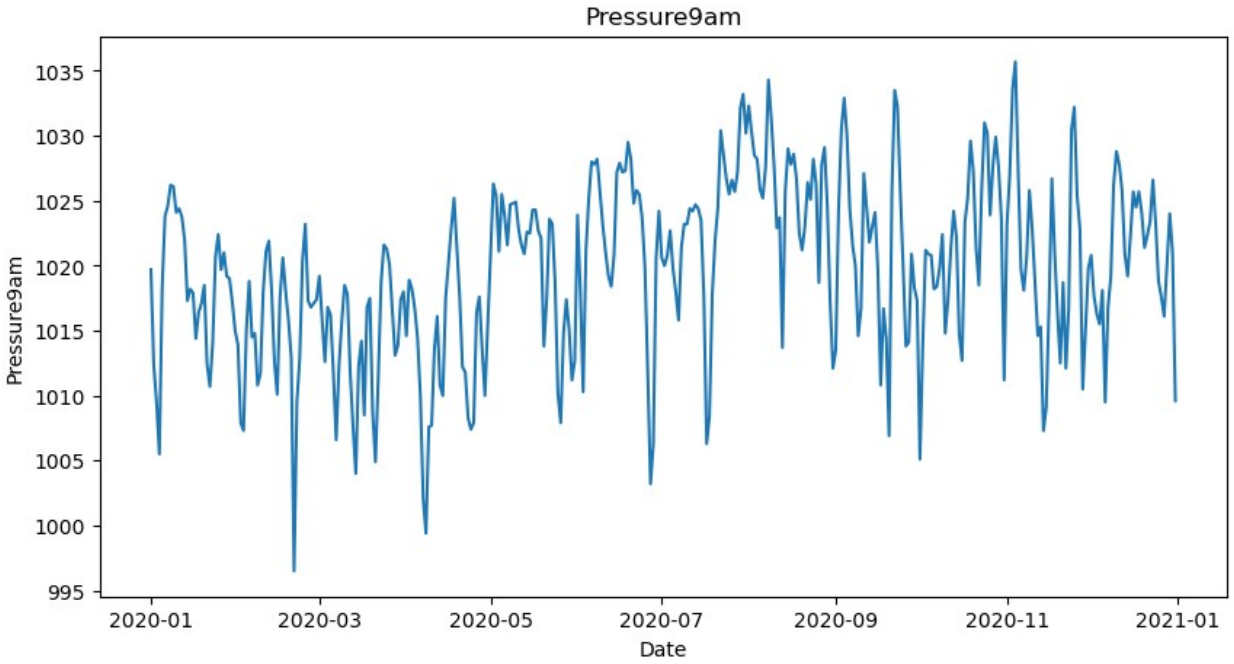

WindGustSpeed
WindGustSpeed
Date
20
40
60
80
100
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01

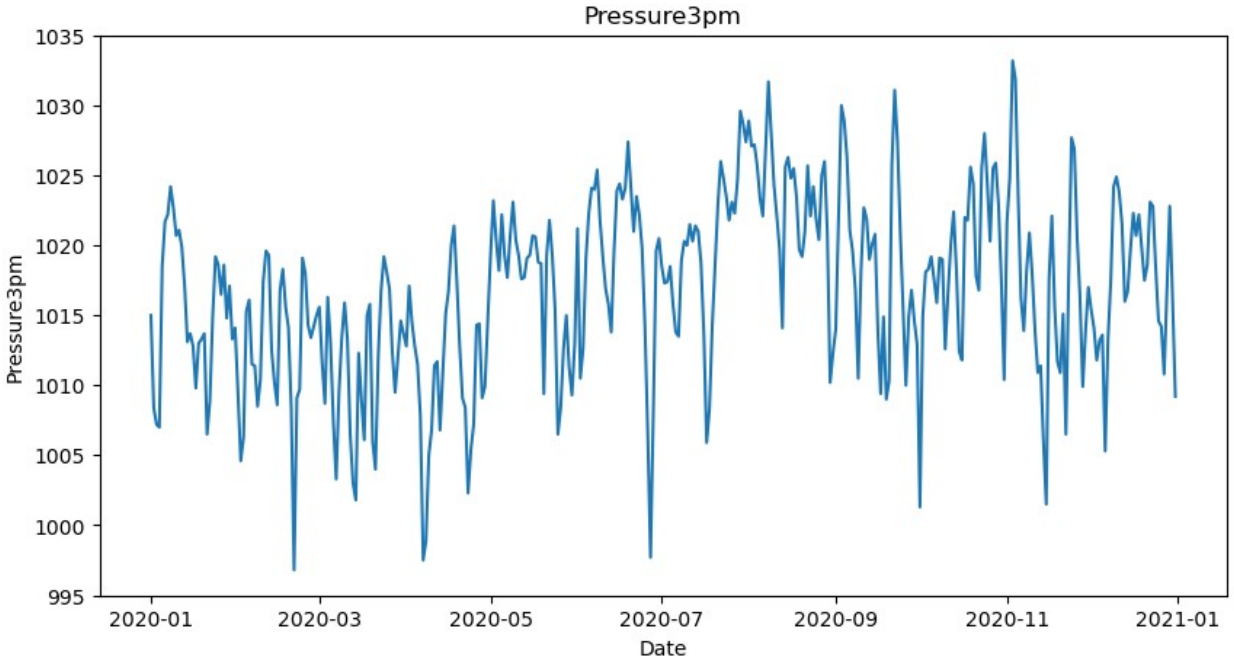

WindSpeed9am
WindSpeed9am
0
10
20
30
40
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01
Date

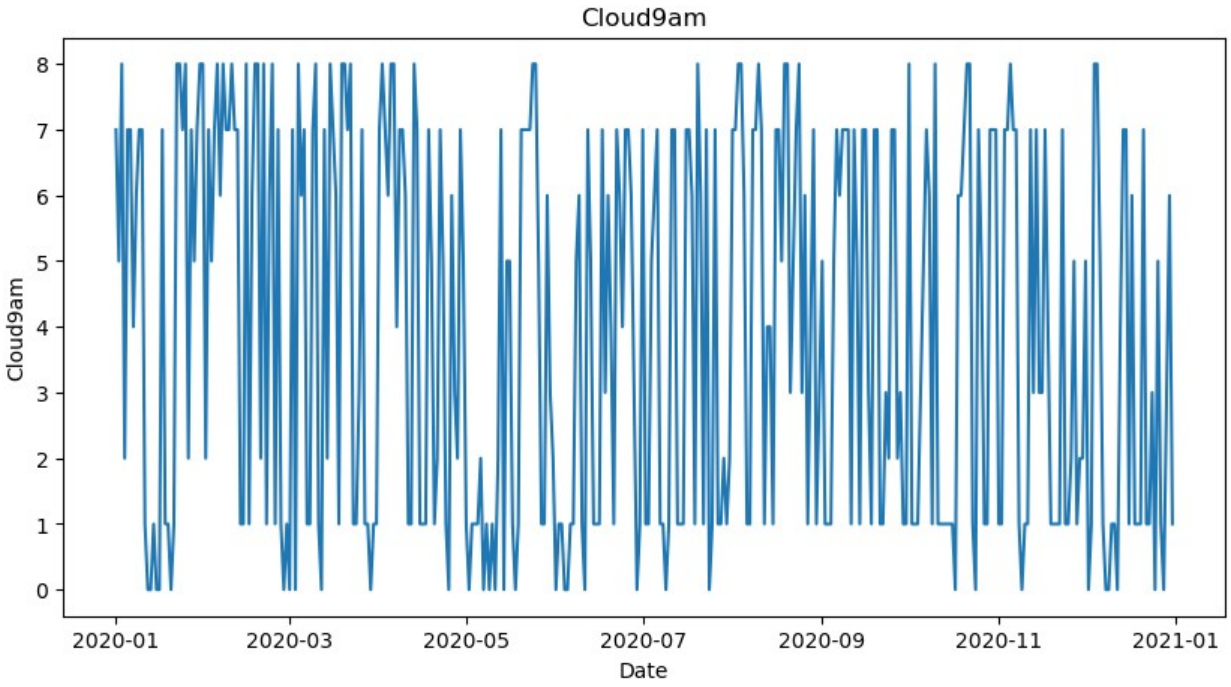

WindSpeed3pm
WindSpeed3pm
0
10
20
30
40
50
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01
Date

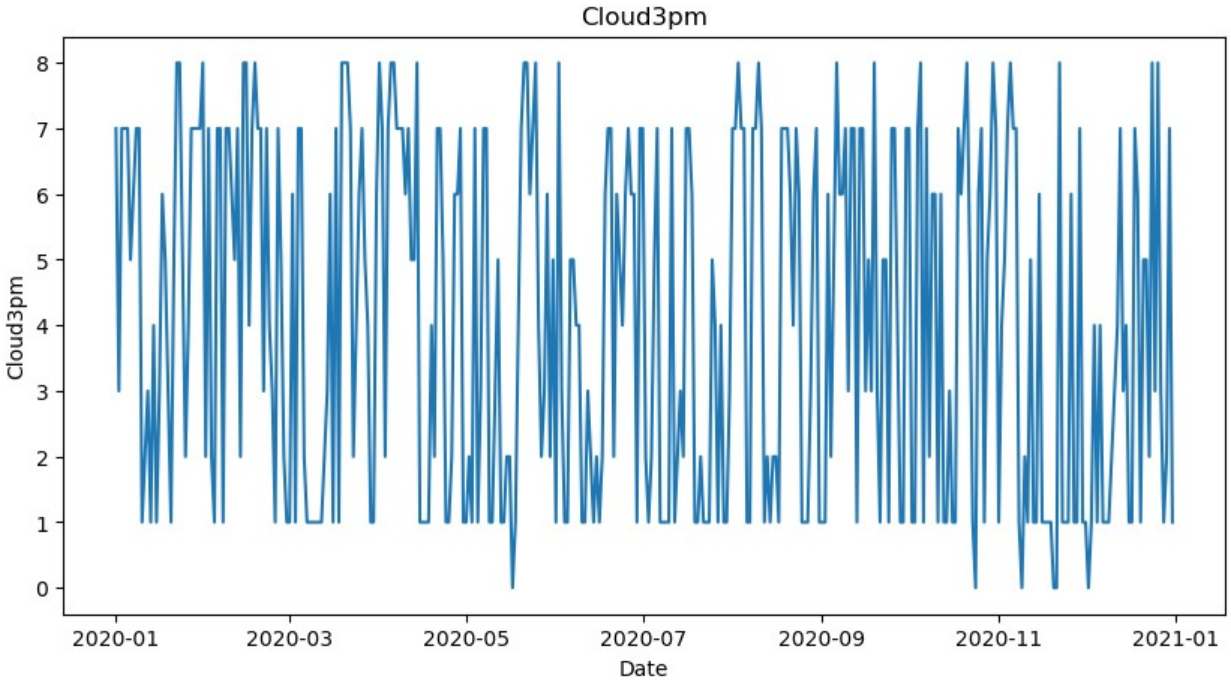

Humidity9am
Humidity9am
100
90
80
70
60
50
40
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01
Date

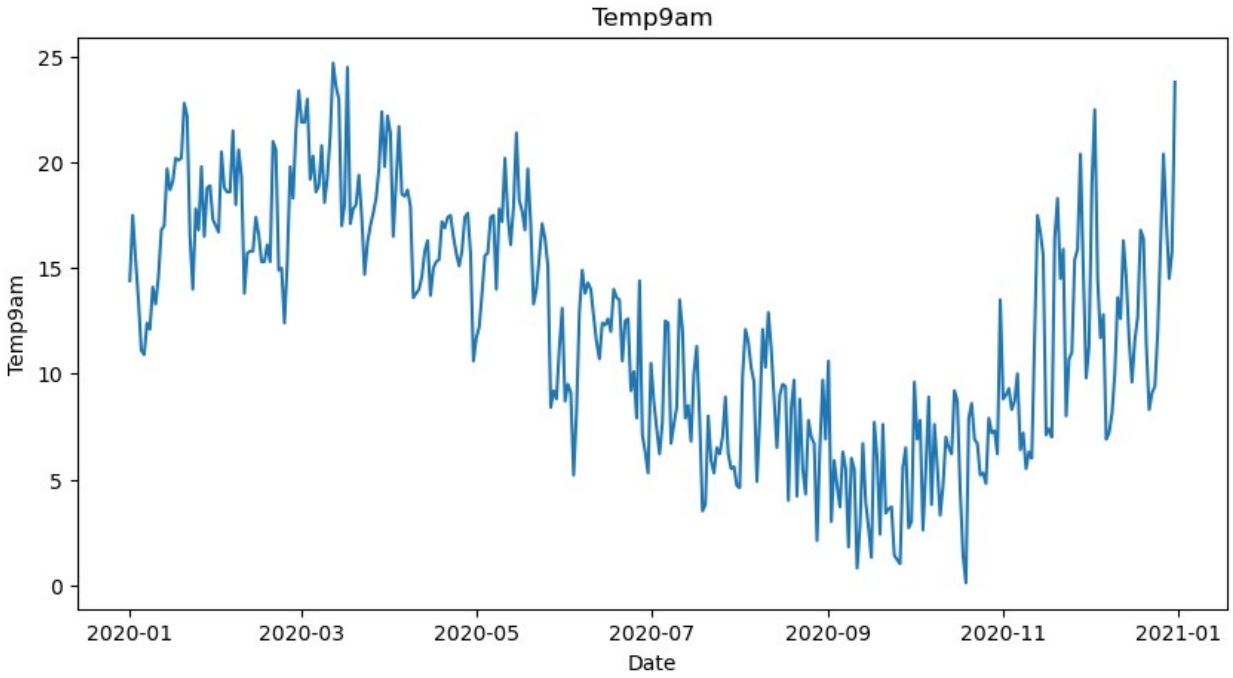

Humidity3pm
Humidity3pm
100
80
60
40
20
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01
Date

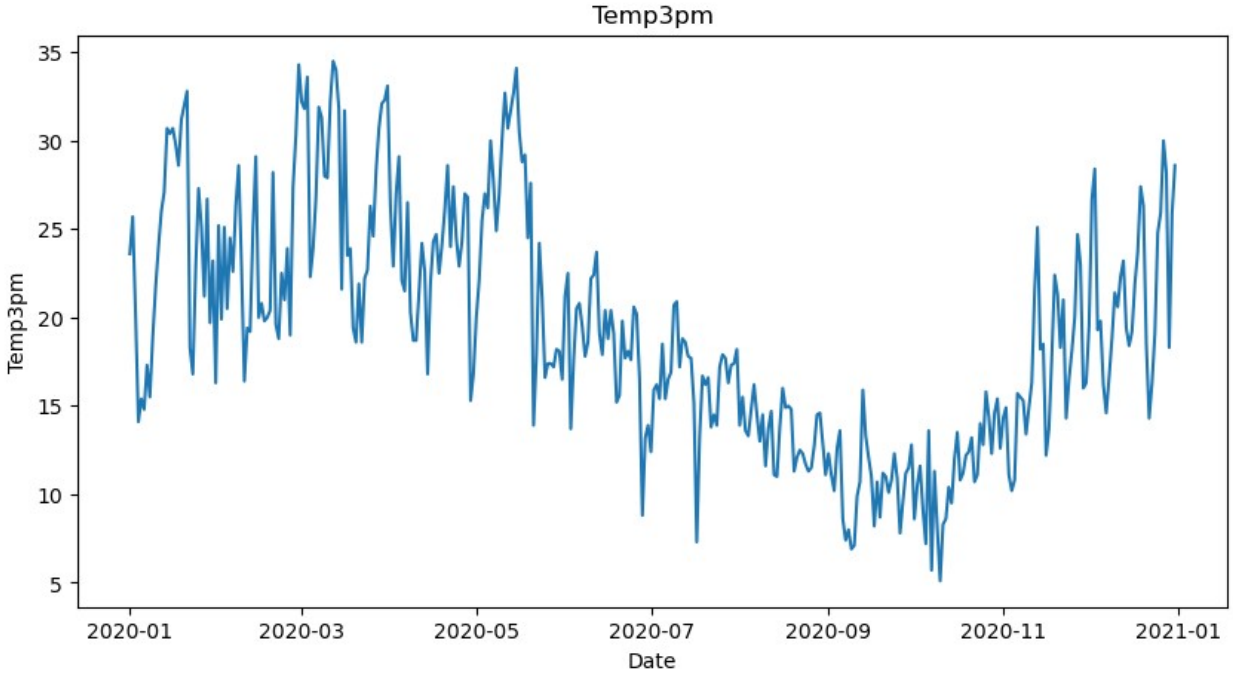

Pressure9am
Pressure9am
1035
1030
1025
1020
1015
1010
1005
1000
995
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01
Date

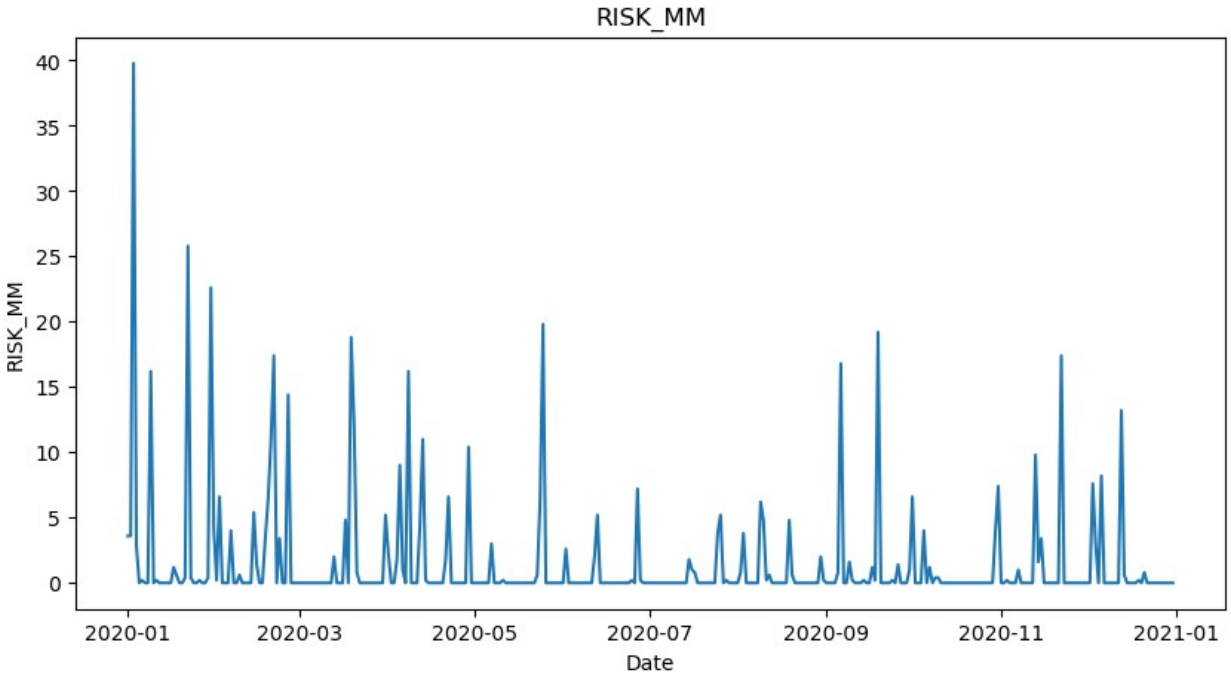

Pressure3pm
Pressure3pm
1035
1030
1025
1020
1015
1010
1005
1000
995
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01
Date


Cloud9am
Cloud9am
Date
0
1
2
3
4
5
6
7
8
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01


Cloud3pm
Cloud3pm
Date
0
1
2
3
4
5
6
7
8
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01


Temp9am
Temp9am
25
20
15
10
5
0
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01
Date

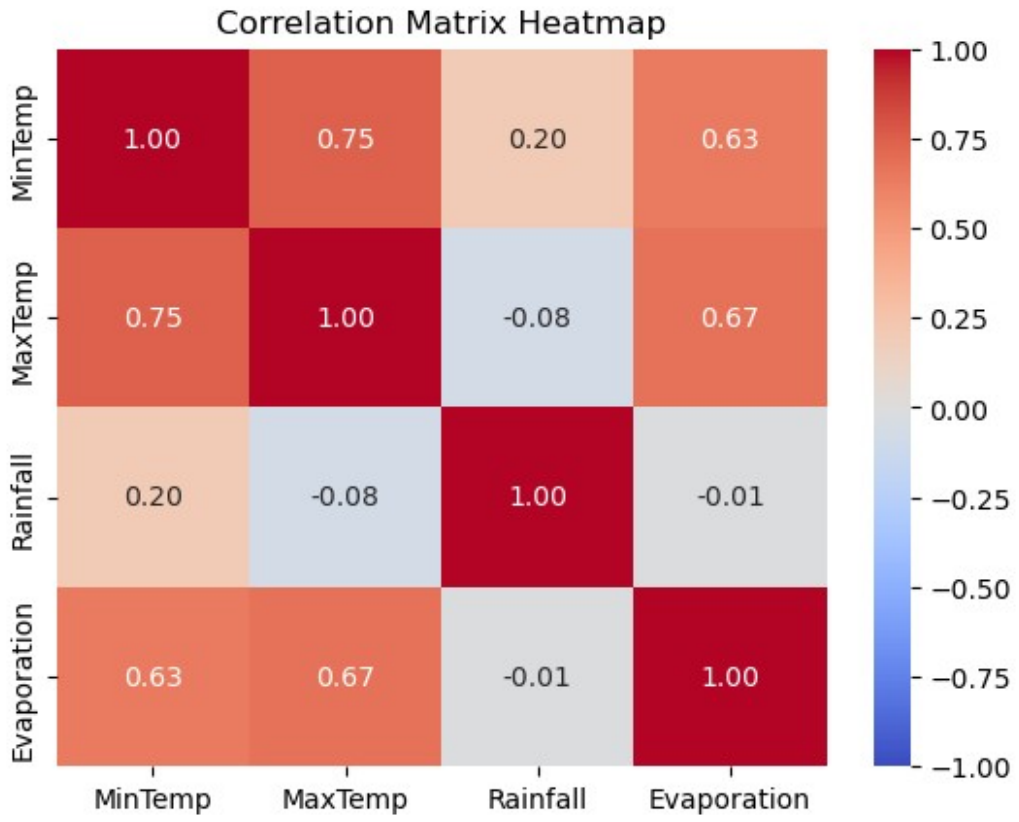

Temp3pm
Temp3pm
35
30
25
20
15
10
5
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01
Date



3. Correlation Analysis

```
correlation_matrix = data_clean[['MinTemp', 'MaxTemp', 'Rainfall',
'Evaporation']].corr()
print(correlation_matrix)
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
fmt=".2f", vmin=-1,vmax=1)
plt.title('Correlation Matrix Heatmap')
plt.show()
```

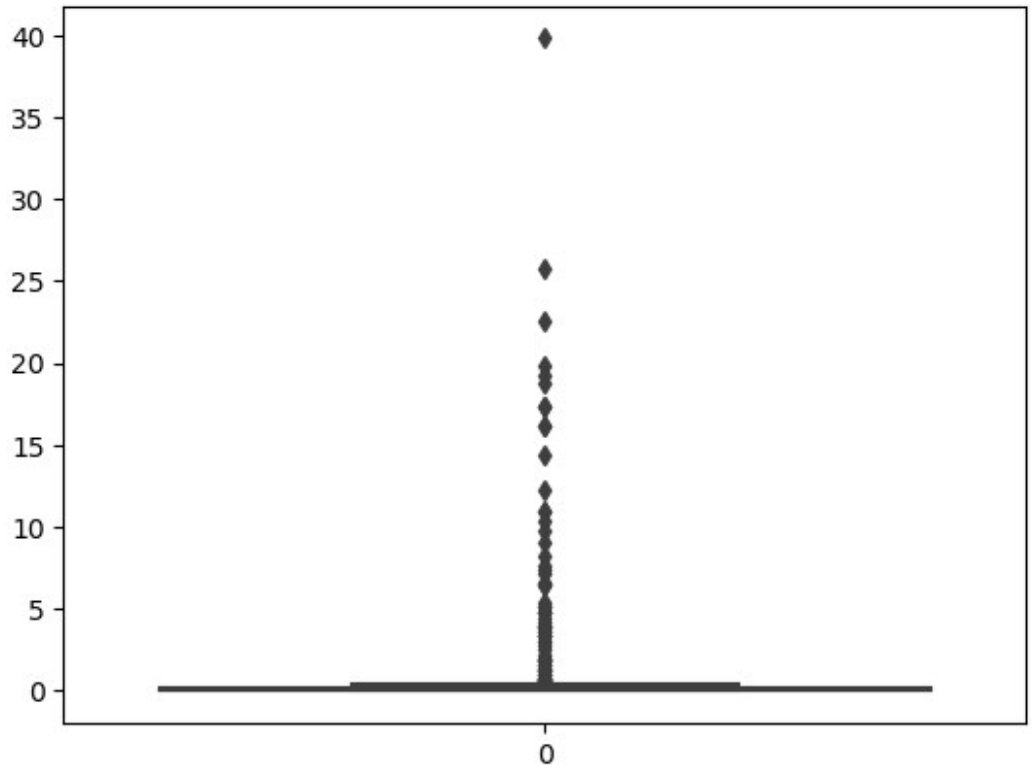
```
               MinTemp   MaxTemp  Rainfall  Evaporation
MinTemp       1.000000  0.745911  0.197339     0.634720
MaxTemp       0.745911  1.000000 -0.077263     0.673162
Rainfall      0.197339 -0.077263  1.000000    -0.011767
Evaporation   0.634720  0.673162 -0.011767     1.000000
```



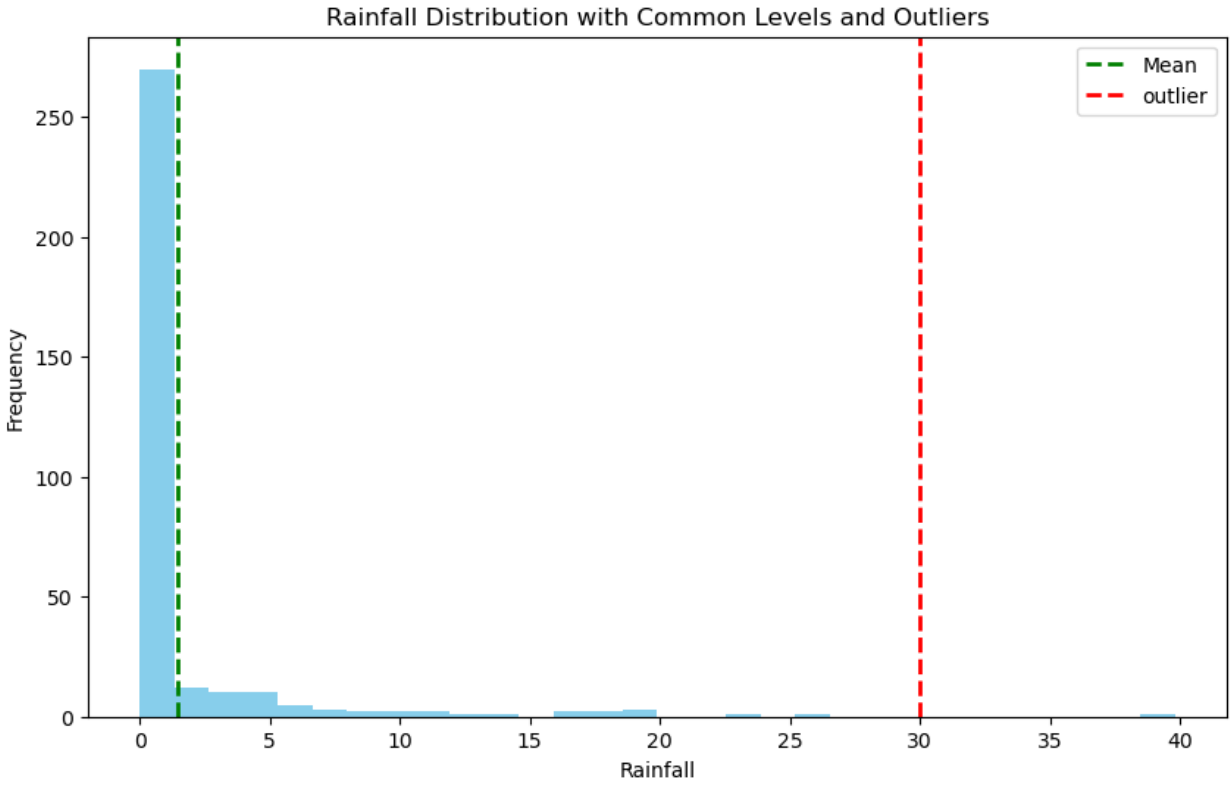
4. Rainfall Distribution

```
#checking for outlier
sns.boxplot(data_clean['Rainfall'])
```

```
<Axes: >
```




```
#from above box plot 40 is the outlier
plt.figure(figsize=(10, 6))
plt.hist(data_clean['Rainfall'], color='skyblue', bins=30)
plt.title('Rainfall Distribution with Common Levels and Outliers')
plt.xlabel('Rainfall')
plt.ylabel('Frequency')

#common level
plt.axvline(data_clean['Rainfall'].mean(), color='green',
linestyle='dashed', linewidth=2, label='Mean')

#outliers
outlier_threshold = 30
plt.axvline(outlier_threshold, color='red', linestyle='dashed',
linewidth=2, label='outlier')

plt.legend()
plt.show()
```



Creating a 'Season' column

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

def get_season(month):
    if 3 <= month <= 5:
        return 'Spring'
    elif 6 <= month <= 8:
        return 'Summer'
    elif 9 <= month <= 11:
        return 'Autumn'
    else:
        return 'Winter'

merged_data['Season'] = merged_data['Date'].dt.month.apply(get_season)

merged_data.columns
```

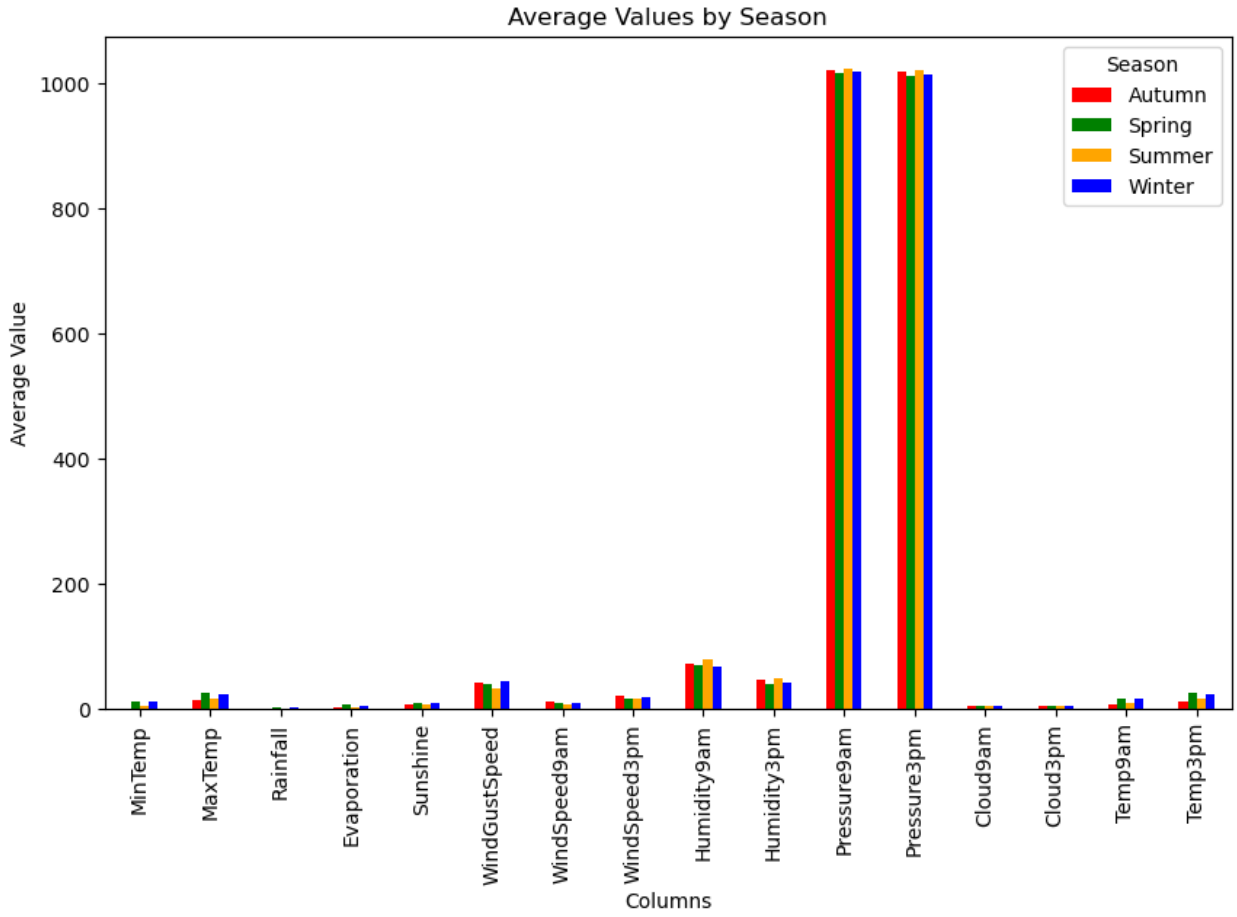
```
Index(['Date', 'MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation',
'Sunshine',
       'WindGustDir', 'WindGustSpeed', 'WindDir9am', 'WindDir3pm',
       'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am', 'Humidity3pm',
       'Pressure9am', 'Pressure3pm', 'Cloud9am', 'Cloud3pm',
'Temp9am',
       'Temp3pm', 'RainToday', 'RISK_MM', 'RainTomorrow', 'Season'],
      dtype='object')
```

5. Seasonal Analysis

```
columns_average = ['MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation',
'Sunshine',
                   'WindGustSpeed', 'WindSpeed9am', 'WindSpeed3pm',
'Humidity9am',
                   'Humidity3pm', 'Pressure9am', 'Pressure3pm',
'Cloud9am',
                   'Cloud3pm', 'Temp9am', 'Temp3pm']

means = {}
for column in columns_average:
    means[column] = merged_data[column].mean()

for column, mean_value in means.items():
    print(f"Mean of {column}: {mean_value:.2f}")
```

```
Mean of MinTemp: 7.27
Mean of MaxTemp: 20.55
Mean of Rainfall: 1.43
Mean of Evaporation: 4.52
Mean of Sunshine: 7.91
Mean of WindGustSpeed: 39.84
Mean of WindSpeed9am: 9.65
Mean of WindSpeed3pm: 17.99
Mean of Humidity9am: 72.04
Mean of Humidity3pm: 44.52
Mean of Pressure9am: 1019.71
Mean of Pressure3pm: 1016.81
Mean of Cloud9am: 3.89
Mean of Cloud3pm: 4.02
Mean of Temp9am: 12.36
Mean of Temp3pm: 19.23
```

```
means_by_season = merged_data.groupby('Season')
[columns_average].mean()

season_colors = {'Winter': 'blue', 'Spring': 'green', 'Summer':
'orange', 'Autumn': 'red'}

fig, ax = plt.subplots(figsize=(10, 6))
means_by_season.T.plot(kind='bar', ax=ax, color=[season_colors[season]
for season in means_by_season.index])
plt.title('Average Values by Season')
plt.ylabel('Average Value')
plt.xlabel('Columns')
plt.legend(title='Season', loc='upper right')
plt.show()
```


Average Values by Season
Season
Autumn
Spring
Summer
Winter
Average Value
0
200
400
600
800
1000
MinTemp
MaxTemp
Rainfall
Evaporation
Sunshine
WindGustSpeed
WindSpeed9am
WindSpeed3pm
Humidity9am
Humidity3pm
Pressure9am
Pressure3pm
Cloud9am
Cloud3pm
Temp9am
Temp3pm
Columns