**Implement a program for time series data cleaning, loading and handling time seriesd data and preprocessing techniques**

**AIM:** To implement a program for time series data cleaning, loading and handling time series data and preprocessing techniques using weather data.

**PROCEDURE:**

**Step 1:** Install and import all necessary libraries.

import os

import numpy as np

!pip install seaborn

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

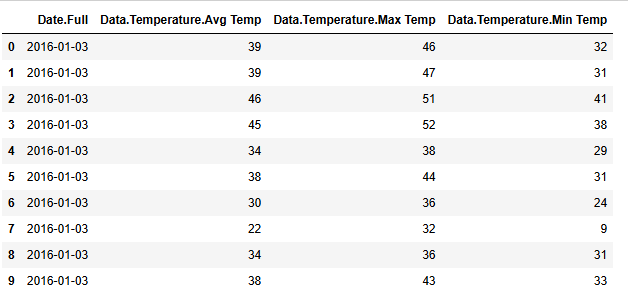
**Step 2:** Load and display the weather dataset.

data=pd.read\_csv(r"D:\weather.csv")

print(data.shape);

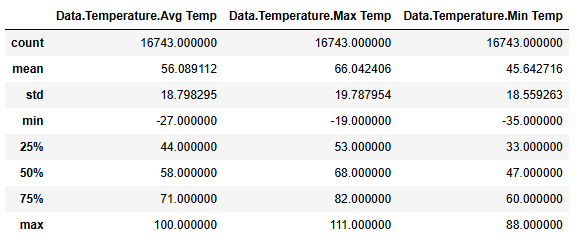
temp\_df = data[["Date.Full","Data.Temperature.Avg Temp","Data.Temperature.Max Temp","Data.Temperature.Min Temp"]]

temp\_df.head(10)



print(temp\_df.shape)

temp\_df.describe()



**Step 3:** Check for null and duplicate values.

data.isnull().sum()

Data.Precipitation 0

Date.Full 0

Date.Month 0

Date.Week of 0

Date.Year 0

Station.City 0

Station.Code 0

Station.Location 0

Station.State 0

Data.Temperature.Avg Temp 0

Data.Temperature.Max Temp 0

Data.Temperature.Min Temp 0

Data.Wind.Direction 0

Data.Wind.Speed 0

temp\_df.duplicated(['Data.Temperature.Avg Temp','Data.Temperature.Max Temp','Data.Temperature.Min Temp'])

0 False

1 False

2 False

3 False

4 False

...

16738 False

16739 True

16740 False

16741 False

16742 True

data['Date.Full'].head()

0 2016-01-03

1 2016-01-03

2 2016-01-03

3 2016-01-03

4 2016-01-03

Name: Date.Full, dtype: object

data['Date.Full']=pd.to\_datetime(data['Date.Full'])

data['Date.Full'].head()

0 2016-01-03

1 2016-01-03

2 2016-01-03

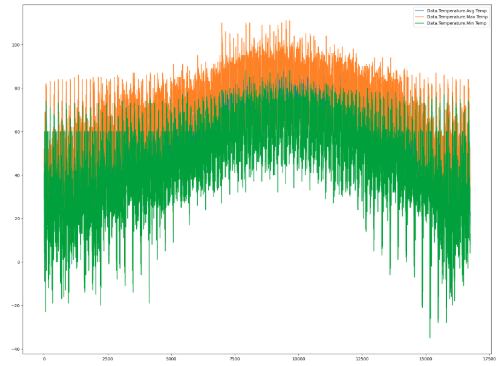
3 2016-01-03

4 2016-01-03

Name: Date.Full, dtype: datetime64[ns]

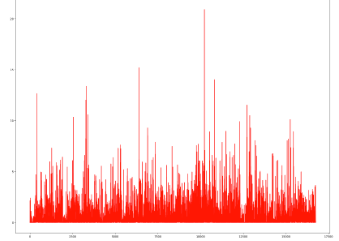
**Step 4:**Visualize the dataset and check for outliers.

temp\_df.plot(figsize=(20,15))

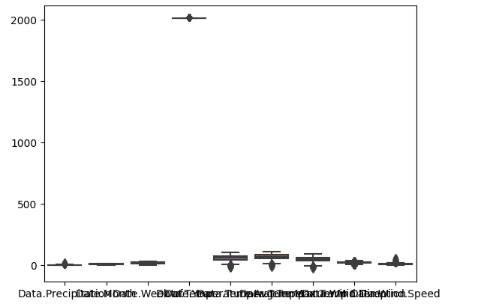


plt.figure(figsize=(20,15))

plt.plot(data.index,data['Data.Precipitation'],label='precipation',color='red')



sns.boxplot(df)



def removal\_box\_plot(df, column, threshold):

sns.boxplot(df[column])

plt.title(f'Original Box Plot of {column}')

plt.show()

removed\_outliers = df[df[column] <= threshold]

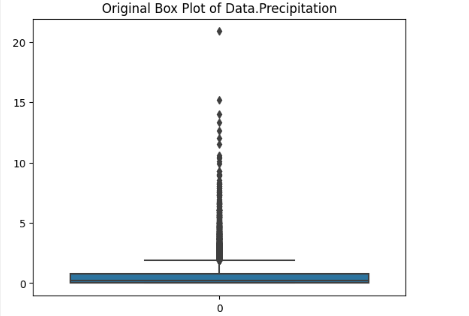
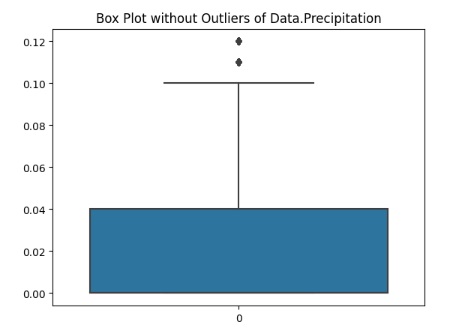
sns.boxplot(removed\_outliers[column])

plt.title(f'Box Plot without Outliers of {column}')

plt.show()

return removed\_outliers

threshold\_value = 0.12

no\_outliers = removal\_box\_plot(df, 'Data.Precipitation', threshold\_value)  
   


.

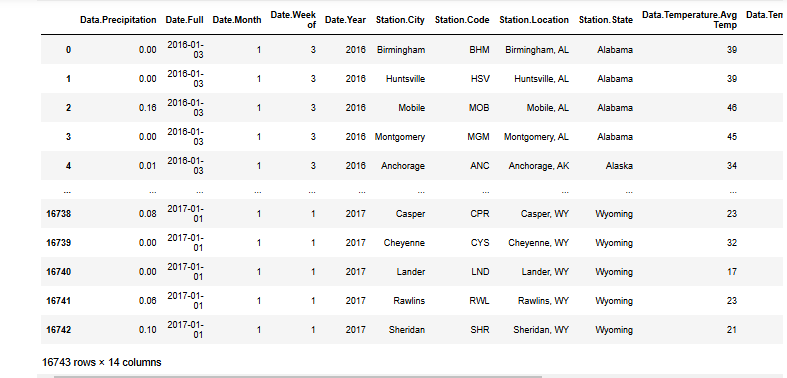
**Step 5:**Data preprocessing and cleaning.  
df = pd.DataFrame(data)

display(df)

# Removing duplicates

unique\_df = df.drop\_duplicates()

display(unique\_df)



**RESULT:**The program for time series data cleaning, loading, handling and preprocessing technique has been successfully completed.