EX:No.1 **DATE: 25/01/2025**	**Implement Programs For Time Series Data Cleaning, Loading, And Handling Time Series Data And Pre-Processing Techniques**

AIM: Write a program to implement time series data for import library, load data, Preprocessing and visualising.

ALGORITHM:

Step 1: Install required libraries (if not already installed).

Step 2: Import necessary libraries (pandas, numpy, matplotlib).

Step 3: Load air pollution data, parse dates, and set 'date' as the index.

Step 4: Remove duplicate timestamps and fill missing values.

Step 5: Select the 'pollution_today' column.

Step 6: Remove outliers using the IQR method.

Step 7: Ensure daily data frequency.

Step 8: Resample to weekly average (optional, not used in the plot).

Step 9: Create a figure and plot daily pollution levels as a line graph.

Step 10: Set labels, title, and legend for the plot.

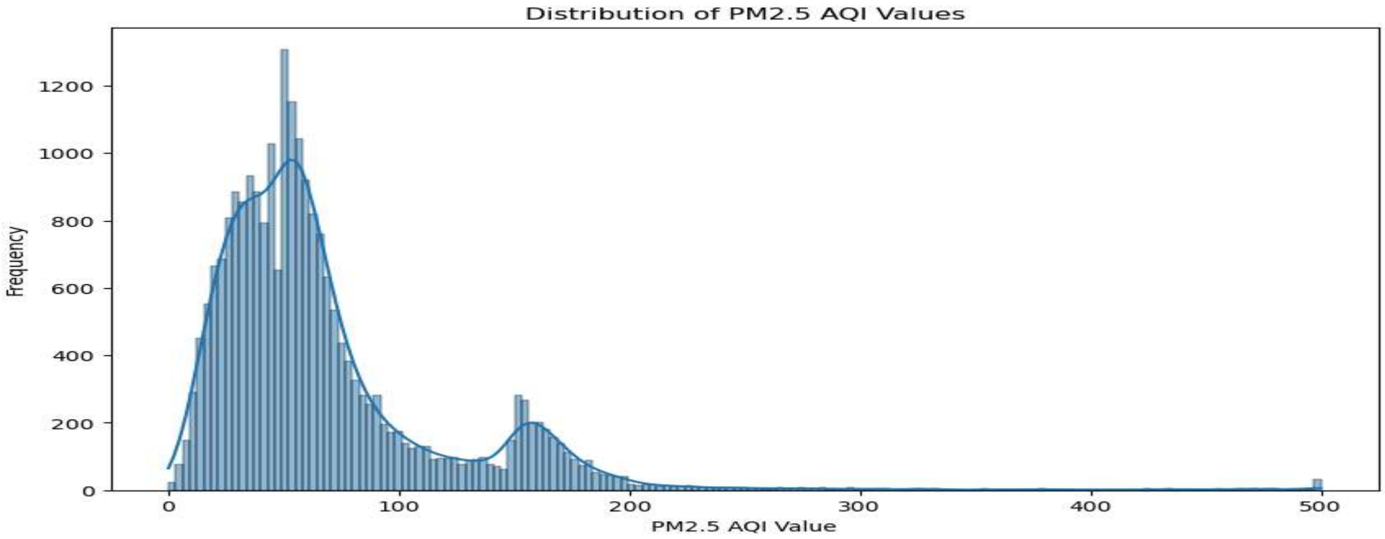
Step 11: Show the plot.

CODE:

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

df = pd.read_csv("/content/air_pollution.csv", parse_dates=["date"], index_col="date")

df = df[~df.index.duplicated(keep='first')]

df.fillna(method="ffill", inplace=True)

df.fillna(method="bfill", inplace=True)
```

```
df = df[['pollution_today']]

Q1 = df.quantile(0.25)

Q3 = df.quantile(0.75)

IQR = Q3 - Q1

df = df[~((df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 * IQR))).any(axis=1)]

df = df.asfreq('D')

df_weekly = df.resample('W').mean()

plt.figure(figsize=(12,5))

plt.plot(df, label="Daily Pollution Level", color="blue", alpha=0.6)

plt.xlabel("Date")

plt.ylabel("Pollution Level")

plt.title("Air Pollution Over Time (Cleaned)")

plt.legend()

plt.show()
```

OUTPUT:



RESULT:

Thus, the program using the time series data implementation has been done successfully.