

**EX:No.6**

**DATE:22/03/25**

**Implement program to apply moving average smoothing for data preparation and time series forecasting.**

**AIM:**

Implement program to apply moving average smoothing for data preparation and time series forecasting.

**OBJECTIVE:**

To apply moving average smoothing for effective time series data preparation and forecasting of air pollution trends.

**BACKGROUND:**

- Time series data often contains random fluctuations that can obscure meaningful patterns.
- Moving average smoothing reduces noise, enabling clearer visualization of long-term trends.

**SCOPE OF THE PROGRAM:**

- The technique is applicable to various air pollutants like PM2.5, CO, NO<sub>2</sub>, SO<sub>2</sub>, and O<sub>3</sub>.
- Smoothed data supports better forecasting and helps in making informed environmental policy decisions.

**CODE:**

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset
file_path = '/content/us_air_pollution_2012_2021.csv'
df = pd.read_csv(file_path)

# Convert 'Date' column to datetime format and set as index
df['Date'] = pd.to_datetime(df['Date'])
df.set_index('Date', inplace=True)

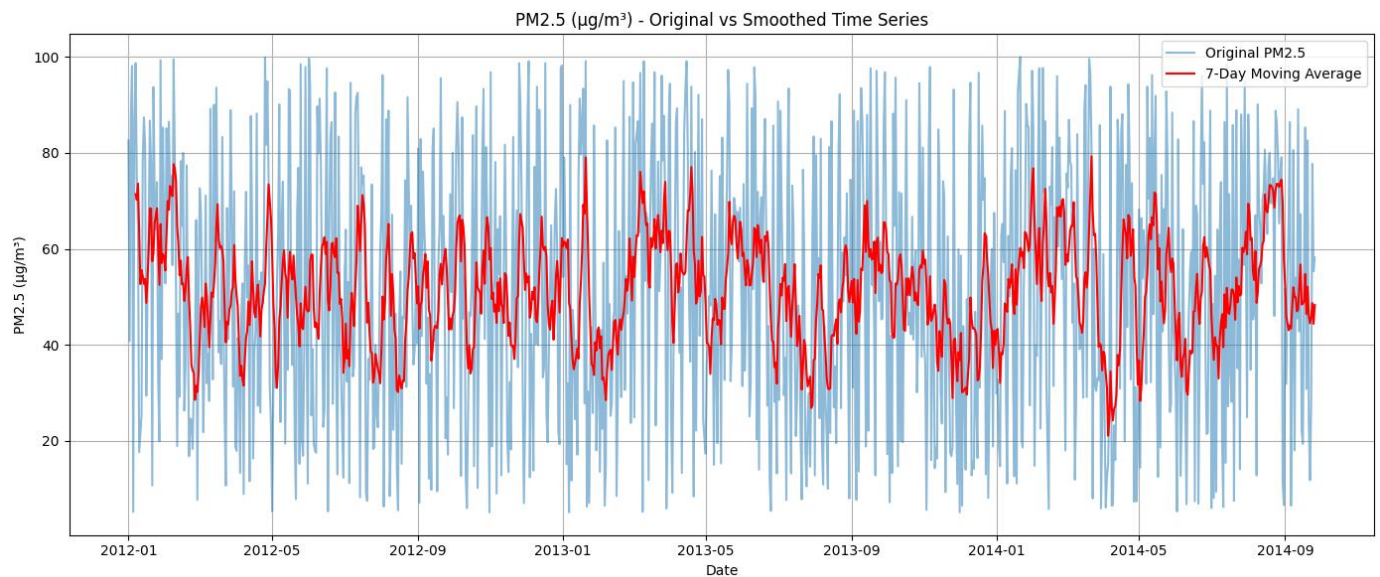
# Select the 'PM2.5 (µg/m³)' column and apply moving average smoothing
pollutant_col = 'PM2.5 (µg/m³)'
ts = df[pollutant_col].dropna()

# Apply a 7-day moving average
window_size = 7
smoothed_ts = ts.rolling(window=window_size).mean()

# Plot original vs smoothed time series
plt.figure(figsize=(14, 6))
plt.plot(ts, label='Original PM2.5', alpha=0.5)
plt.plot(smoothed_ts, label='7-Day Moving Average', color='red')
plt.title('PM2.5 (µg/m³) - Original vs Smoothed Time Series')
```

```
plt.xlabel('Date')
plt.ylabel('PM2.5 (µg/m³)')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

## OUTPUT:



## RESULT:

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