|  |  |
| --- | --- |
| **EXP:5**  **27/03/2025** | **Trend Estimation & Elimination in Time Series Data: Aggregation & Smoothing Techniques** |

**AIM:**

To implement programs for estimating and eliminating trends in time series data using aggregation and smoothing techniques, helping to identify underlying patterns and reduce fluctuations in the dataset.

**PROCEDURE:**

**1) Import Necessary Libraries**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from statsmodels.tsa.api import SimpleExpSmoothing

**2)** **Load and Inspect the Dataset**

**file\_path = "/content/sample\_data/Car\_sales.csv"**

**df = pd.read\_csv(file\_path)**

**3) Convert 'Latest\_Launch' to Datetime and Sort Data**

df['Latest\_Launch'] = pd.to\_datetime(df['Latest\_Launch'], errors='coerce')

df = df.sort\_values(by='Latest\_Launch').reset\_index(drop=True)

**4)** **Create Numerical Time Index**

df['Time'] = np.arange(len(df))

**5) Select Relevant Columns for Analysis**

time = df['Time']

values = df['Sales\_in\_thousands']

**6) Estimate Trend Using Moving Average**

window\_size = 5

df['Moving\_Avg'] = values.rolling(window=window\_size, center=True).mean()

### **7) Eliminate Trend Using Exponential Smoothing**

alpha = 0.2

ses\_model = SimpleExpSmoothing(values).fit(smoothing\_level=alpha)

df['Smoothed'] = ses\_model.fittedvalues

**8) Visualize Original Data, Moving Average, and Smoothed Data**

plt.figure(figsize=(25, 6))

plt.plot(time, values, label='Original Data', linestyle='dashed', alpha=0.7)

plt.plot(time, df['Moving\_Avg'], label='Moving Average (Trend Estimation)', color='red')

plt.plot(time, df['Smoothed'], label='Exponential Smoothing (Trend Elimination)', color='green')

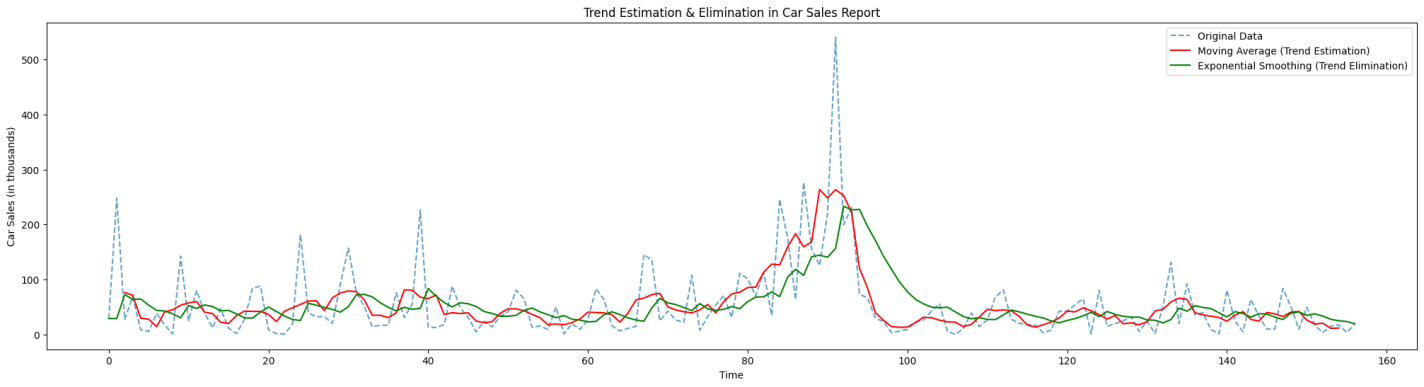
plt.xlabel('Time')

plt.ylabel('Car Sales (in thousands)')

plt.legend()

plt.title('Trend Estimation & Elimination in Car Sales Report')

plt.show()

**OUTPUT**

**RESULT:**

Thus the program has been executed successfully