|  |  |
| --- | --- |
| **EXP:6**  **27/03/2025** | **Implementing Moving Average Smoothing for Data Preparation and Time Series Forecasting** |

**AIM:**

To apply moving average smoothing techniques to time series data in order to reduce short-term fluctuations, reveal underlying trends, and enhance the accuracy of forecasts.

**PROCEDURE:**

**1) Import Necessary Libraries**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

**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) Apply Moving Average Smoothing**

window\_size = 5

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

### **7) Visualize Original Data and Moving Average**

plt.figure(figsize=(12, 6))

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

plt.plot(time, df['Moving\_Avg'], label=f'{window\_size}-Day Moving Average', color='red')

plt.xlabel('Time')

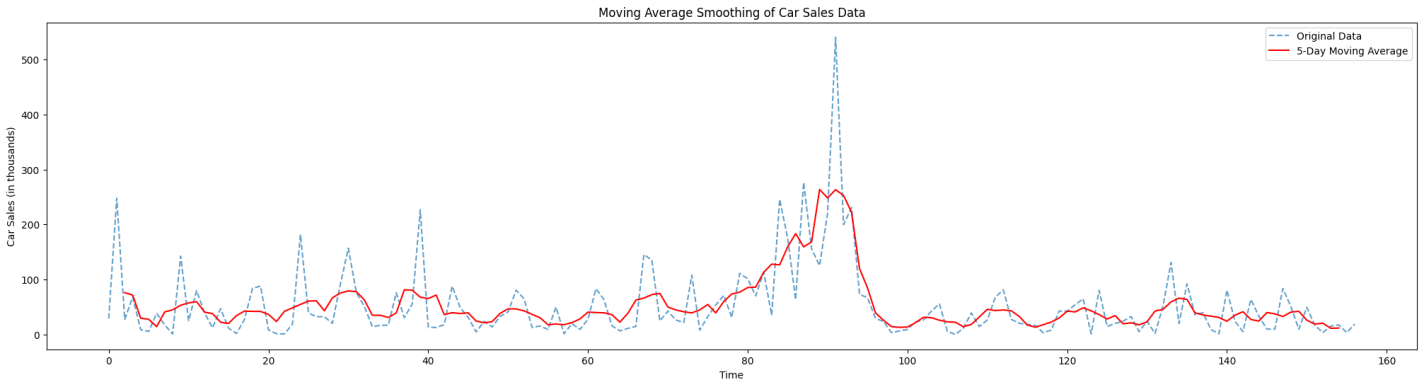
plt.ylabel('Car Sales (in thousands)')

plt.legend()

plt.title('Moving Average Smoothing of Car Sales Data')

plt.show()

**OUTPUT**

****

**RESULT:**

Thus the program has been executed successfully