Date : 27/03/25

EX .NO : 5

Implement programs for estimating & eliminating trend in time series data- aggregation, smoothing.

AIM:

To Implement programs for estimating & eliminating trend in time series data- aggregation, smoothing.

PROCEDURE:

1. Data Loading

First, load the dataset using **Pandas**. The dataset should contain columns like `Date` and `Close`, representing the stock prices.

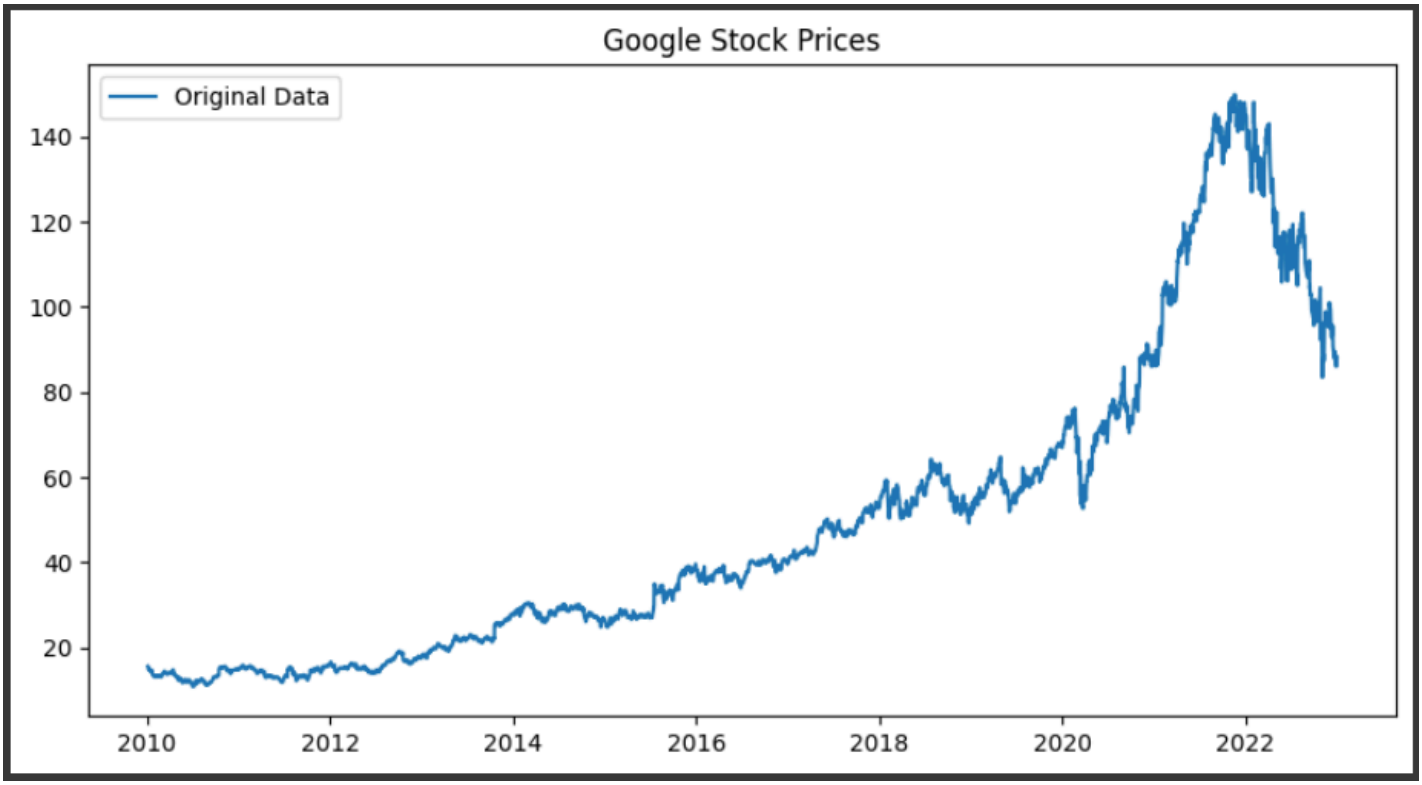
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
data = pd.read_csv('google_stock_prices.csv', parse_dates=['Date'],
index_col='Date')
print(data.head())
```

2. Visualization of Original Data

Plot the original stock prices to visualize the overall trend

```
plt.figure(figsize=(10, 5))
plt.plot(data['Close'], label='Original Data')
plt.title('Google Stock Prices')
plt.xlabel('Date')
plt.ylabel('Closing Price')
plt.legend()
plt.grid()
plt.show()
```

This plot provides a clear view of the general movement of stock prices.

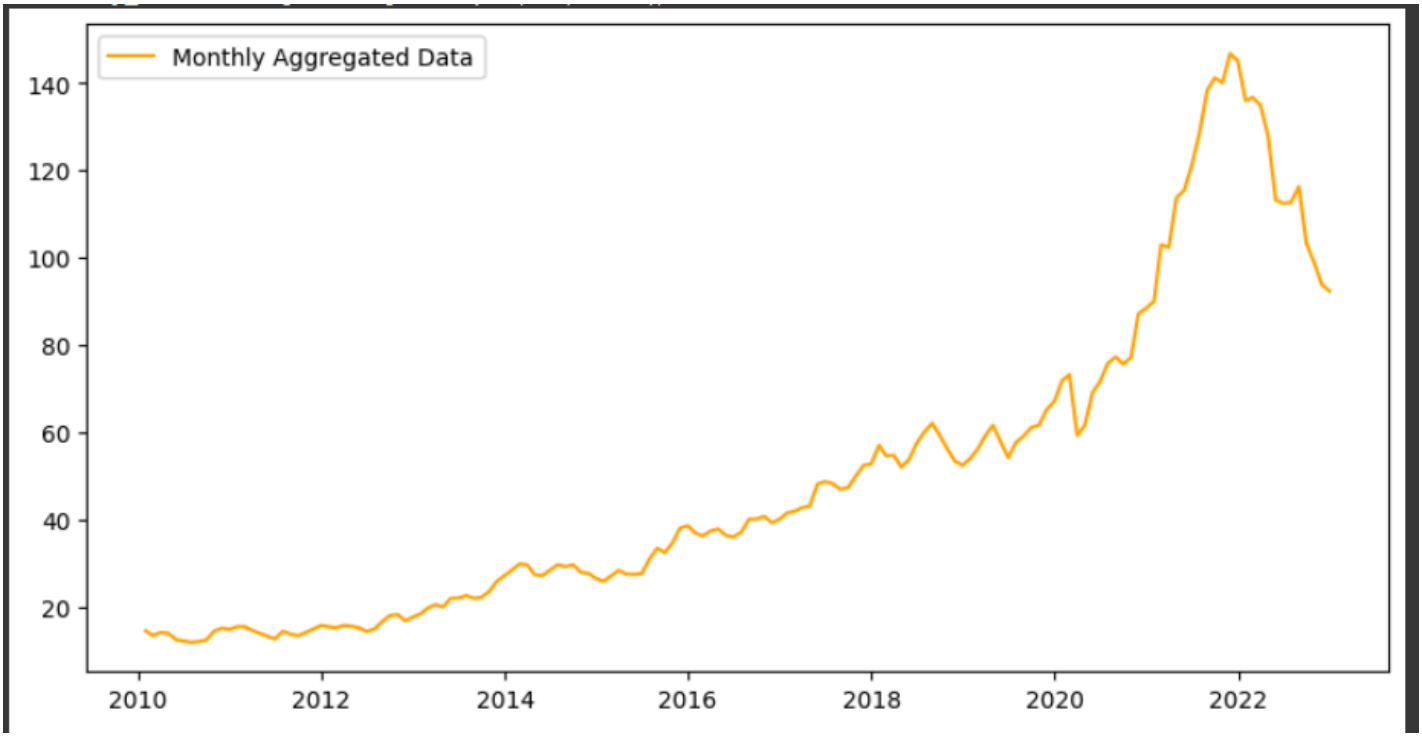


3. Aggregation using Monthly Mean

Perform aggregation using the **monthly mean** to analyze the overall trend.

```
monthly_data = data['Close'].resample('M').mean()
plt.figure(figsize=(10, 5))
plt.plot(monthly_data, label='Monthly Aggregated Data', color='orange')
plt.title('Monthly Aggregated Google Stock Prices')
plt.xlabel('Date')
plt.ylabel('Average Closing Price')
plt.legend()
plt.grid()
plt.show()
```

`.resample('M')` means the data is resampled to a monthly frequency.
`.mean()` computes the average stock price for each month.

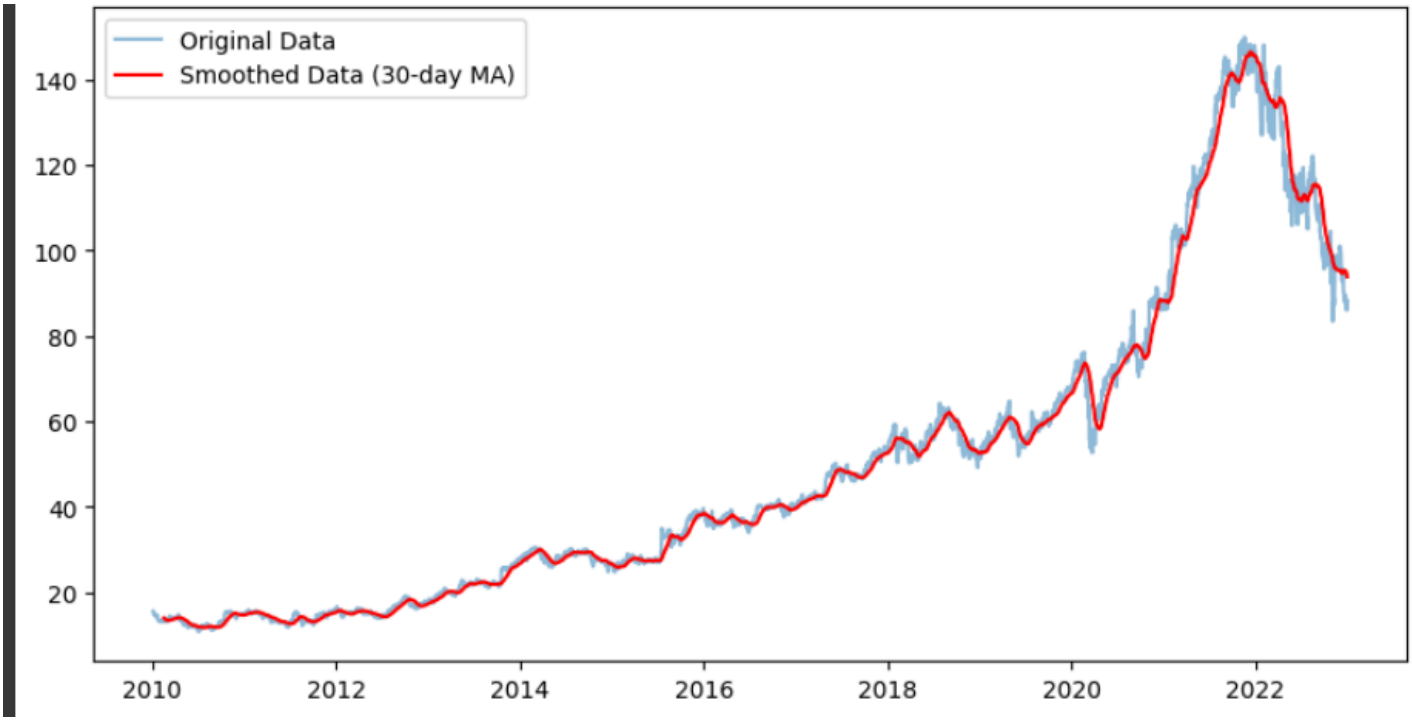


4. Smoothing using Moving Average

Apply a **30-day moving average** for smoothing to reduce noise.

```
window_size = 30
smoothed_data = data['Close'].rolling(window=window_size).mean()
plt.figure(figsize=(10, 5))
plt.plot(data['Close'], alpha=0.5, label='Original Data')
plt.plot(smoothed_data, label='Smoothed Data (30-day MA)', color='red')
plt.title('Moving Average Smoothing')
plt.xlabel('Date')
plt.ylabel('Closing Price')
plt.legend()
plt.grid()
plt.show()
```

- `.rolling(window=30)` creates a 30-day window for averaging.
- This removes short-term fluctuations and highlights the trend.

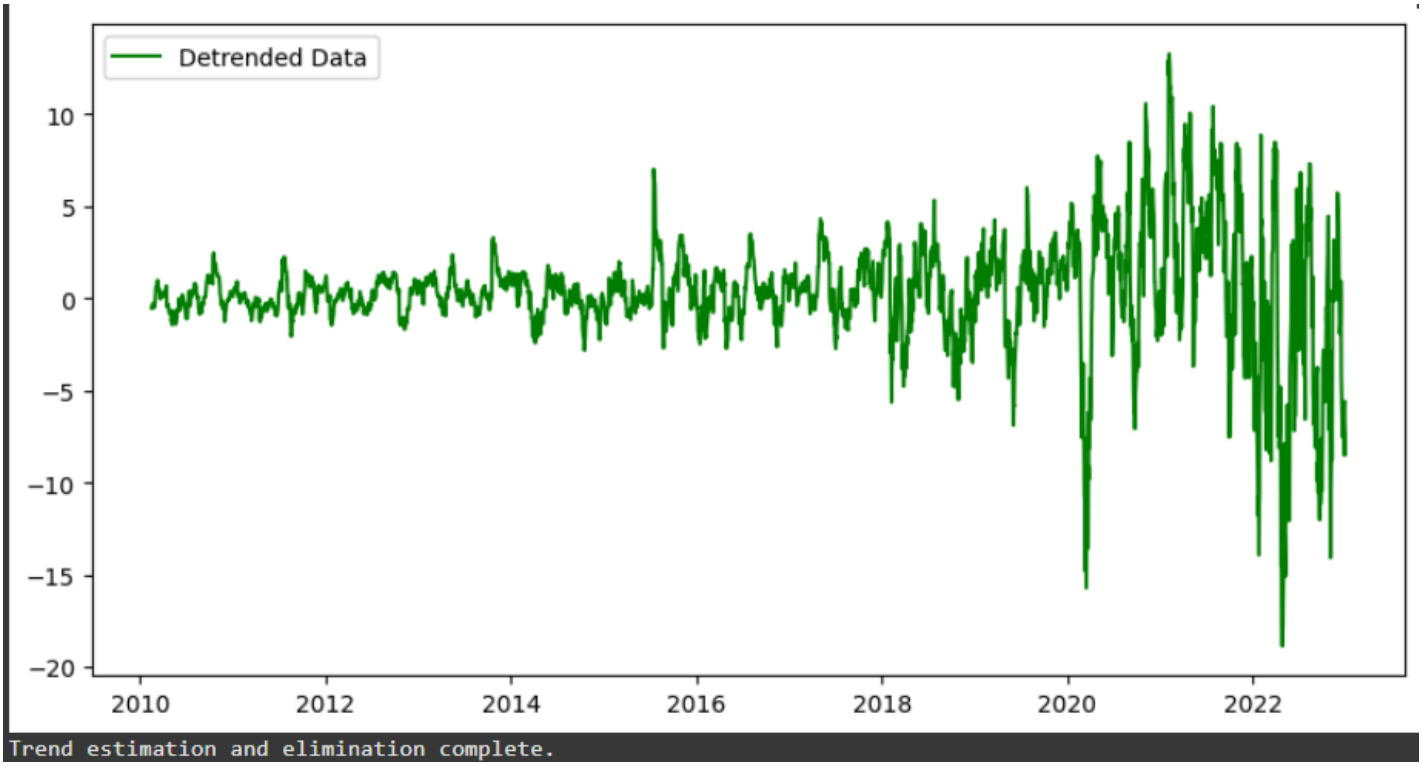


5. Trend Elimination using Detrending

Perform detrending by subtracting the moving average from the original data.

```
detrended_data = data['Close'] - smoothed_data
plt.figure(figsize=(10, 5))
plt.plot(detrended_data, label='Detrended Data', color='green')
plt.title('Detrended Google Stock Prices')
plt.xlabel('Date')
plt.ylabel('Price Difference (Original - Trend)')
plt.legend()
plt.grid()
plt.show()
```

Detrending helps analyze cyclical and seasonal components by removing the trend.



RESULT :

Thus the Implementation programs for estimating & eliminating trend in time series data- aggregation, smoothing is done successfully.