

EX:No.4

DATE:1/02/25

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

AIM:

To implement programs for estimating & eliminating trends in time series data – aggregation, smoothing, etc.

OBJECTIVE:

To estimate and remove trends in time-series gold price data using aggregation and smoothing techniques.

BACKGROUND:

- Time series data often has trends that affect analysis.
- Aggregation (e.g., monthly/yearly averaging) helps identify patterns.
- Smoothing (e.g., moving average, exponential smoothing) removes fluctuations.
- Trend elimination improves forecasting and stationarity.

SCOPE OF THE PROGRAM:

- Load and gold price (2012-2021).
- Apply aggregation (monthly/yearly averages) to estimate trends.
- Use moving average smoothing to reduce noise.
- Apply exponential smoothing to highlight trends

CODE:

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

df = pd.read_csv("/content/gold_data.csv")

df['Date'] = pd.to_datetime(df['Date'], errors='coerce')

gold_price_col = "Price"

df = df[(df['Date'].dt.year >= 2012) & (df['Date'].dt.year <= 2021)]

df.set_index('Date', inplace=True)

df_monthly = df[gold_price_col].resample('M').mean()
df_yearly = df[gold_price_col].resample('Y').mean()

df['Moving_Avg'] = df[gold_price_col].rolling(window=12).mean()

df['Exp_Smooth'] = df[gold_price_col].ewm(span=12, adjust=False).mean()
```

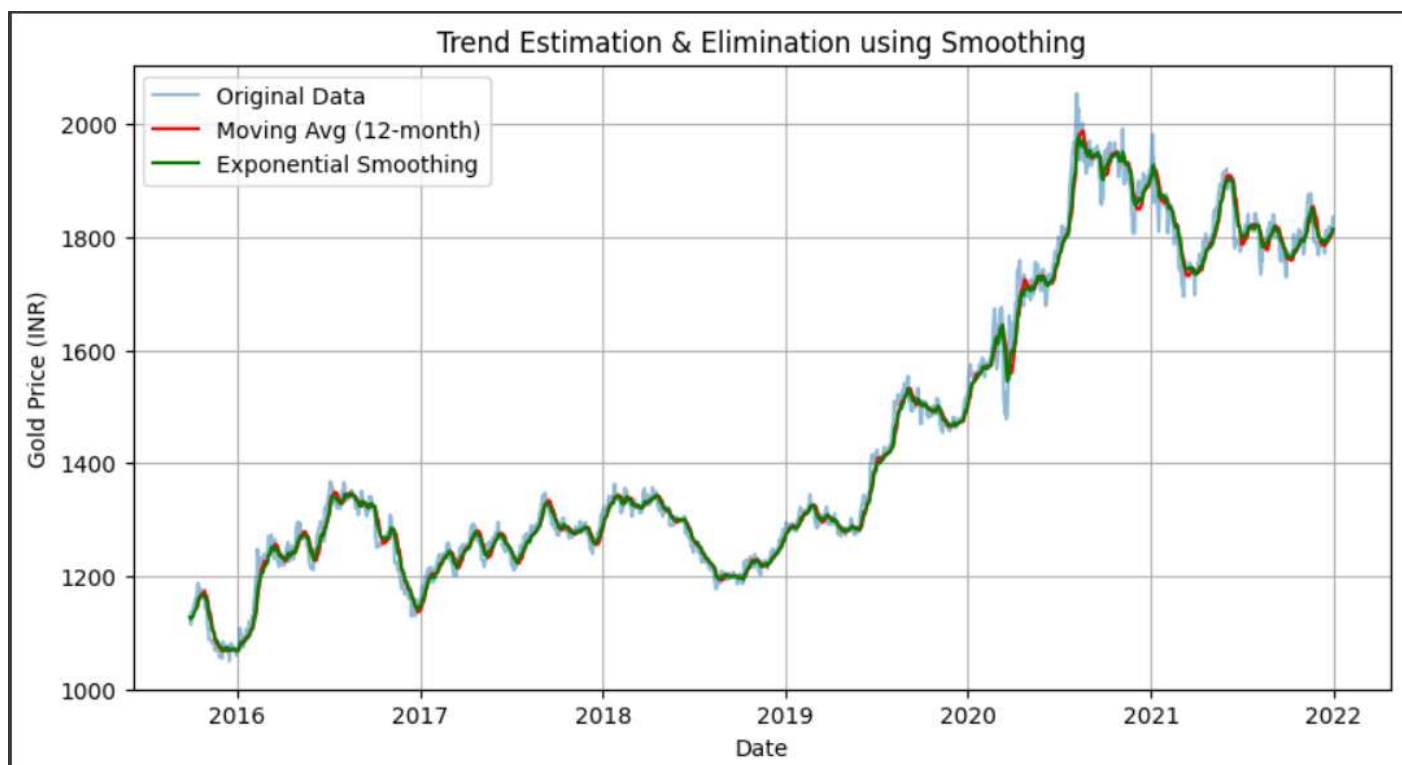
```

plt.figure(figsize=(10, 5))
plt.plot(df[gold_price_col], label="Original Data", alpha=0.5)
plt.plot(df['Moving_Avg'], label="Moving Avg (12-month)", color='red')
plt.plot(df['Exp_Smooth'], label="Exponential Smoothing", color='green')

plt.xlabel("Date")
plt.ylabel("Gold Price (INR)")
plt.title("Trend Estimation & Elimination using Smoothing")
plt.legend()
plt.grid()
plt.show()

```

OUTPUT:



RESULT:

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