

**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 gold price 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 gold price during various years.
- Smoothed data supports better forecasting and helps in making informed environmental policy decisions.

### **CODE:**

```
#Import Libraries
import pandas as pd
import matplotlib.pyplot as plt

#Load the dataset
df = pd.read_csv('gold_data (1).csv')

#Convert 'Date' to datetime and sort
df['Date'] = pd.to_datetime(df['Date'], dayfirst=True)
df = df.sort_values('Date')

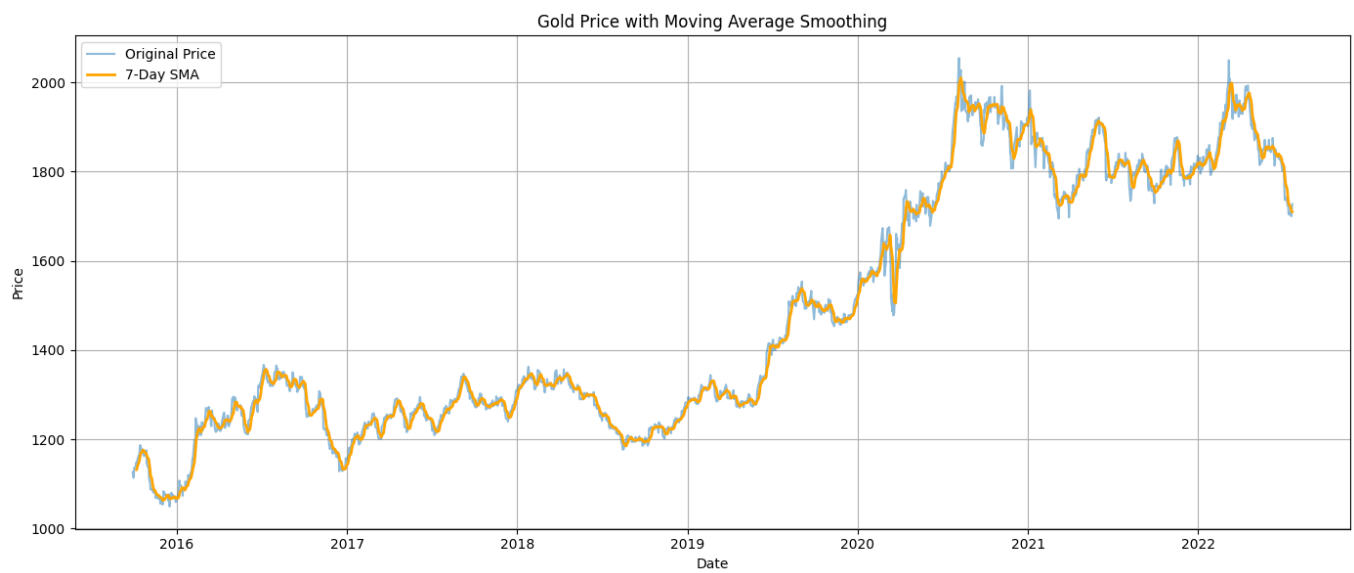
#Set 'Date' as index for time series operations
df.set_index('Date', inplace=True)

#Apply Moving Average Smoothing
window_size = 7 # You can change this to experiment
df['SMA_Price'] = df['Price'].rolling(window=window_size).mean()

#plot
plt.figure(figsize=(14, 6))
plt.plot(df['Price'], label='Original Price', alpha=0.5)
```

```
plt.plot(df['SMA_Price'], label=f'{window_size}-year SMA', linewidth=2,
color='orange')
plt.title('Gold Price with Moving Average Smoothing')
plt.xlabel('Date')
plt.ylabel('Price')
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.