EXP:6 27/03/2025	MOVING AVERAGE SMOOTHING IN TIME SERIES DATA.

AIM:

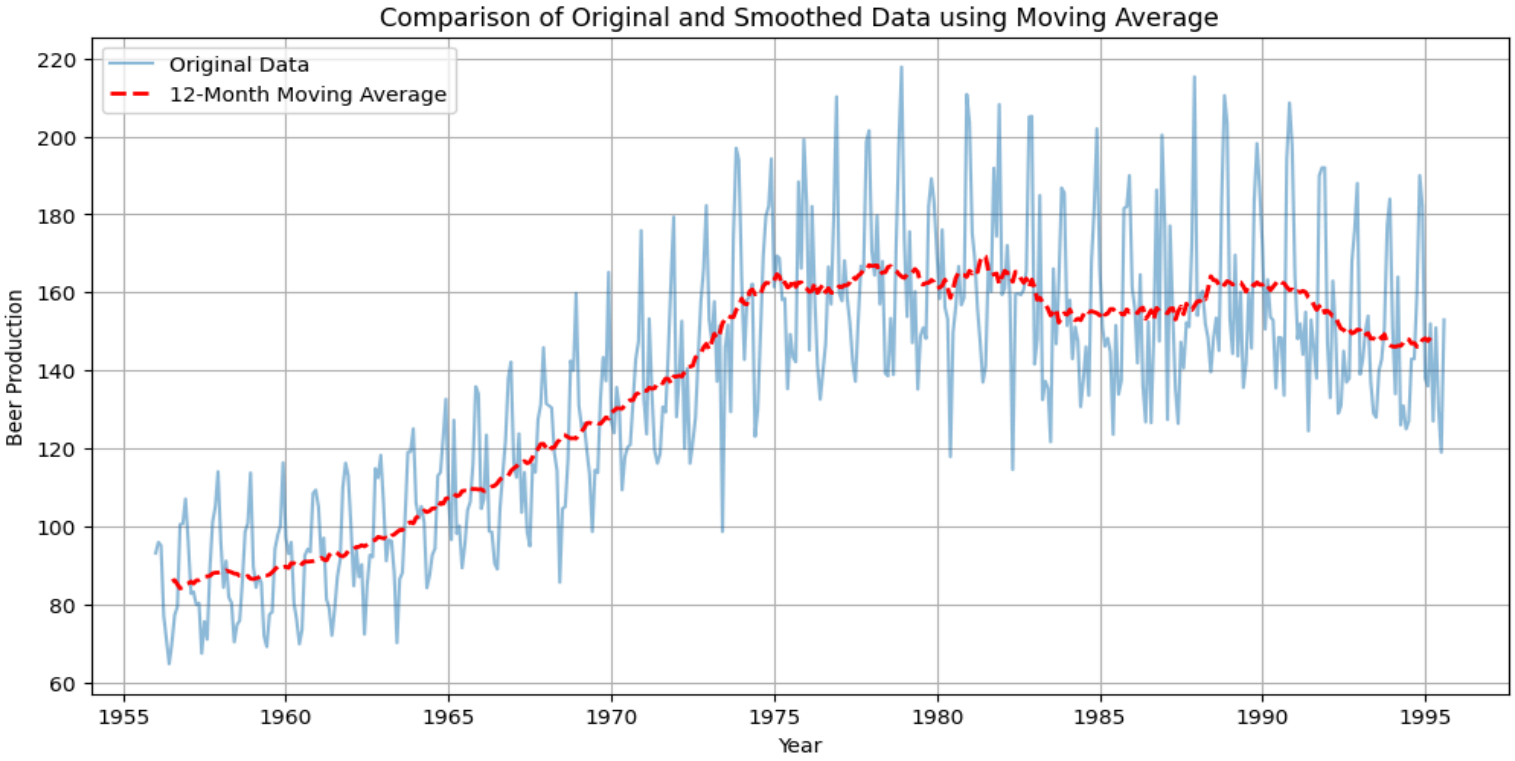
To Implement a program to apply moving average smoothing for data preparation and time series forecasting.

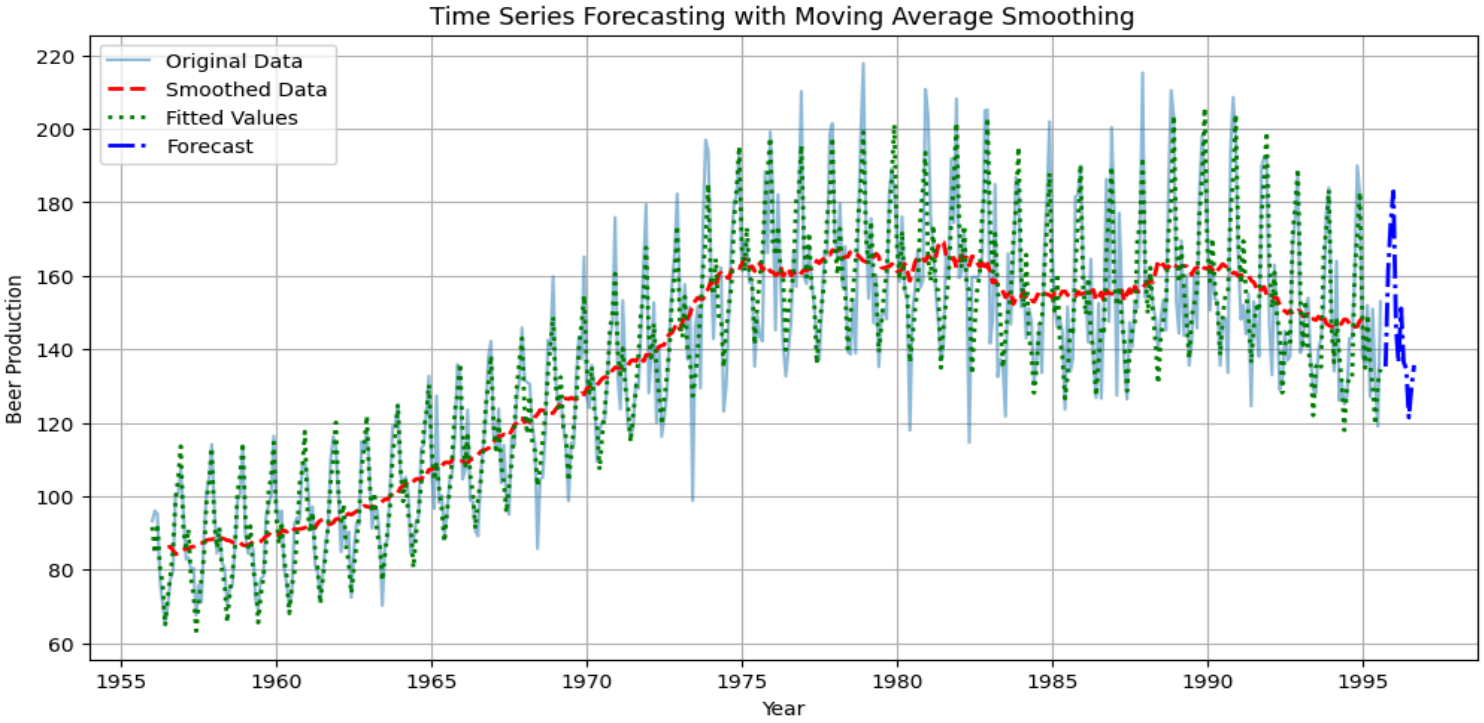
PROCEDURE:

```
1) Load the dataset
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
from statsmodels.tsa.holtwinters import ExponentialSmoothing

file_path = "monthly-beer.csv"
df = pd.read_csv(file_path)

2) Convert 'Month' to datetime format
df['Month'] = pd.to_datetime(df['Month'])

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

df['Moving_Avg_12'] = df['Monthly beer production'].rolling(window=12,
center=True).mean()

3) Plot original data vs. smoothed data
plt.figure(figsize=(12, 6))
plt.plot(df.index, df['Monthly beer production'], label="Original Data",
alpha=0.5)
plt.plot(df.index, df['Moving_Avg_12'], label="12-Month Moving Average",
linewidth=2, linestyle="dashed", color='red')
plt.legend()
plt.title("Comparison of Original and Smoothed Data using Moving Average")
plt.xlabel("Year")
plt.ylabel("Beer Production")
plt.grid()
```

```
plt.show()

4) Apply exponential smoothing for time series forecasting
model = ExponentialSmoothing(df['Monthly beer production'], trend='add',
seasonal='add', seasonal_periods=12).fit()
df['Forecast'] = model.fittedvalues

future_steps = 12
future_index = pd.date_range(start=df.index[-1], periods=future_steps + 1,
freq='M')[1:]
future_forecast = model.forecast(future_steps)

5) Plot original data, smoothed data, and forecast
plt.figure(figsize=(12, 6))
plt.plot(df.index, df['Monthly beer production'], label="Original Data",
alpha=0.5)
plt.plot(df.index, df['Moving_Avg_12'], label="Smoothed Data",
linewidth=2, linestyle="dashed", color='red')
plt.plot(df.index, df['Forecast'], label="Fitted Values", linewidth=2,
linestyle="dotted", color='green')
plt.plot(future_index, future_forecast, label="Forecast", linewidth=2,
linestyle="dashdot", color='blue')
plt.legend()
plt.title("Time Series Forecasting with Moving Average Smoothing")
plt.xlabel("Year")
plt.ylabel("Beer Production")
plt.grid()
plt.show()
```

OUTPUT:





RESULT :

Thus the Program has been Implemented and executed successfully.