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

**EX:No.5**

**DATE:29/03/25**

**AIM:**

To estimate and eliminate trends in AAPL stock time series data using aggregation (moving average) and smoothing (exponential smoothing).

**ALGORITHM:**

1. Load the Data: Read the AAPL CSV file and sort it by date.
2. Select Closing Price: Extract the closing price for trend analysis.
3. Apply Rolling Mean: Compute the moving average for trend estimation.
4. Apply Exponential Smoothing: Use Holt-Winters smoothing to refine the trend.
5. Detrend the Data: Subtract the estimated trend from the original data.
6. Decompose the Series: Perform seasonal decomposition to extract trend, seasonality, and residuals.
7. Visualize Results: Plot the original data, estimated trend, detrended series, and seasonal components.

**CODE:**

import pandas as pd

import matplotlib.pyplot as plt

from statsmodels.tsa.seasonal import seasonal\_decompose

from statsmodels.tsa.holtwinters import ExponentialSmoothing

# Load AAPL stock data (modify 'Date' column name if needed)

df = pd.read\_csv("/content/AAPL.csv", parse\_dates=["Date"], index\_col="Date")

# Ensure the dataset is sorted by date

df = df.sort\_index()

# Select relevant column (modify if needed)

df["Price"] = df["Close"] # Change if needed (e.g., "Adj Close")

# Aggregation using Rolling Mean (Moving Average)

window\_size = 20 # Adjust window size as needed

df["Rolling\_Mean"] = df["Price"].rolling(window=window\_size).mean()

# Exponential Smoothing for trend estimation

model = ExponentialSmoothing(df["Price"], trend="additive", seasonal=None)

fit = model.fit(smoothing\_level=0.2, optimized=True)

df["Exp\_Smooth"] = fit.fittedvalues

# Detrended Data (Original - Estimated Trend)

df["Detrended"] = df["Price"] - df["Exp\_Smooth"]

# Seasonal Decomposition

decomposition = seasonal\_decompose(df["Price"], model="additive", period=window\_size)

df["Trend"] = decomposition.trend

df["Seasonal"] = decomposition.seasonal

df["Residual"] = decomposition.resid

# Plot Results

plt.figure(figsize=(12, 8))

plt.subplot(3, 1, 1)

plt.plot(df.index, df["Price"], label="Original Price", alpha=0.7)

plt.plot(df.index, df["Rolling\_Mean"], label="Rolling Mean", color="red")

plt.plot(df.index, df["Exp\_Smooth"], label="Exp. Smoothing", color="green", linestyle="dashed")

plt.legend()

plt.title("AAPL Stock Price with Estimated Trend")

plt.subplot(3, 1, 2)

plt.plot(df.index, df["Detrended"], label="Detrended Price", color="purple")

plt.legend()

plt.title("Detrended Stock Price")

plt.subplot(3, 1, 3)

plt.plot(df.index, df["Seasonal"], label="Extracted Seasonality", color="orange")

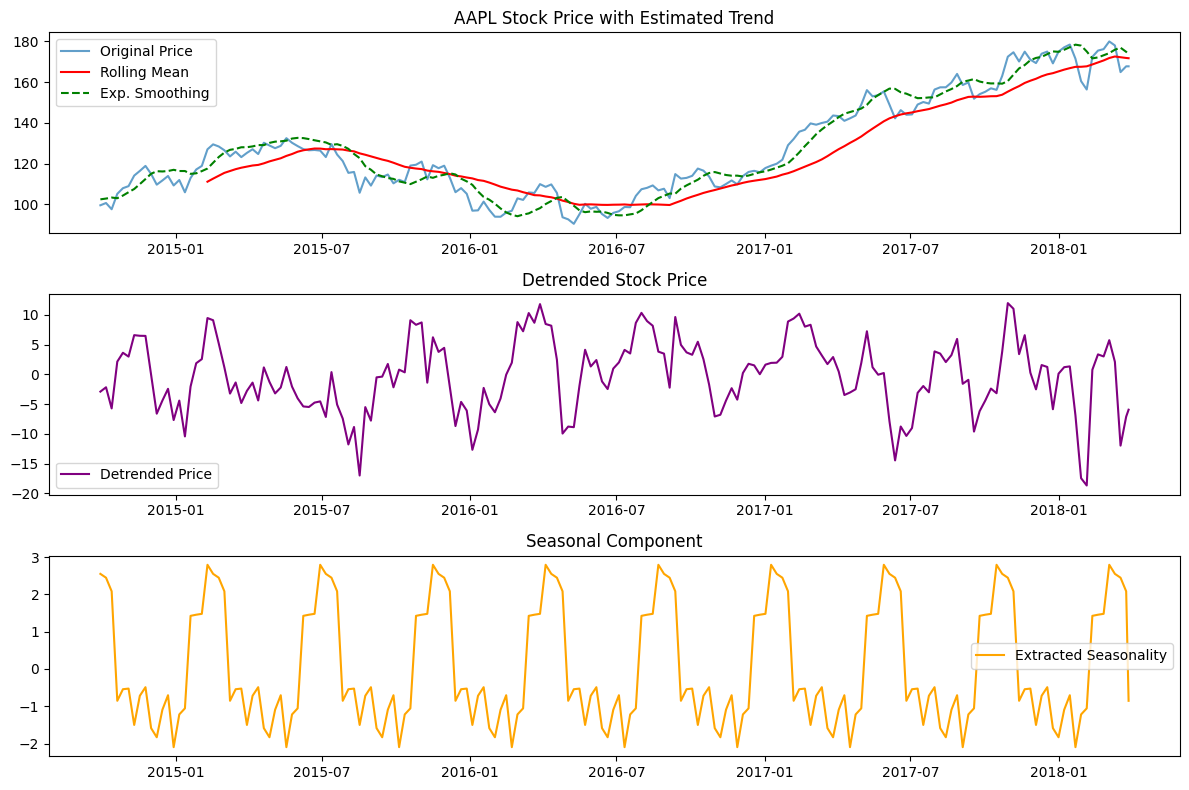
plt.legend()

plt.title("Seasonal Component")

plt.tight\_layout()

plt.show()

**OUTPUT:**



**RESULT:**

Thus the program has been completed and verified successfully.