EXPERIMENT - 5

AIM: To implement programs for estimating & eliminating trend in time series data- aggregation, smoothing

PROCEDURE AND CODE:

Steps 1: Importing the necessary packages.

```
from statsmodels.tsa.holtwinters import SimpleExpSmoothing, ExponentialSmoothing
```

Step 2: Applying Simple Exponential Smoothing.(taking Volume as variable)

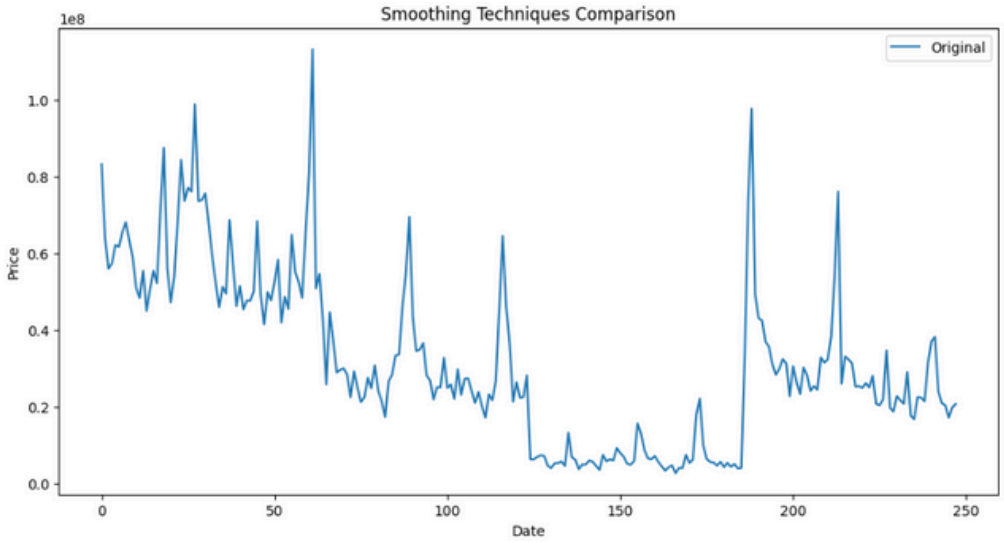
```
df['SES'] = SimpleExpSmoothing(df['seasional_first_difference']).fit(smoothing_level=0.5, optimized=False).fittedvalues
```

Step 3: Applying Exponential Smoothing.

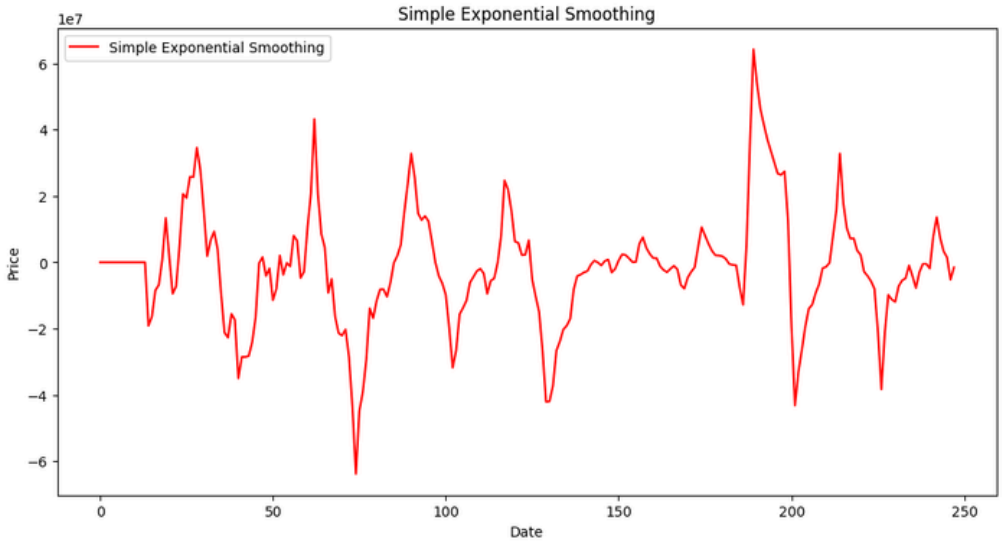
```
df['HWES'] = ExponentialSmoothing(df['seasional_first_difference'], trend='add', seasonal='add', seasonal_periods=7).fit().fittedvalues
```

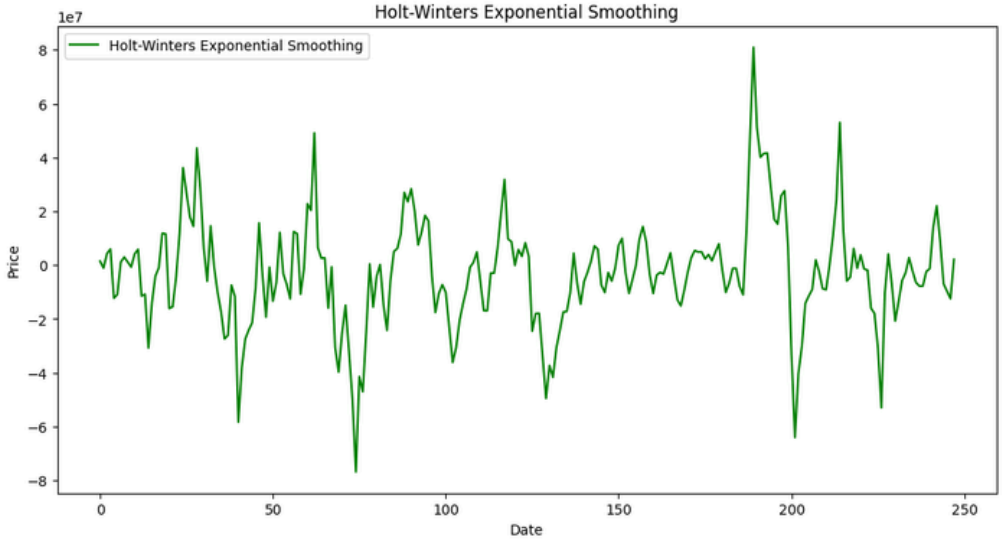
Step 4: Plotting the graph for Volume. Original vs Simple vs Exponential Smooting.

```
plt.figure(figsize=(12, 6))
plt.plot(df['Volume'], label='Original')
plt.legend()
plt.title('Smoothing Techniques Comparison')
plt.xlabel('Date')
plt.ylabel('Price')
plt.show()
```




```
plt.figure(figsize=(12, 6))
plt.plot(df['SES'], label='Simple Exponential Smoothing', color='red')
plt.legend()
plt.title('Simple Exponential Smoothing')
plt.xlabel('Date')
plt.ylabel('Price')
plt.show()
```





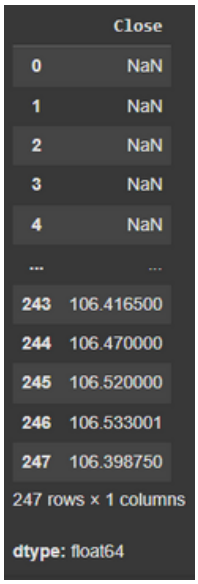
Steps 5: Doing Data aggregation techniques which includes mean, avg, volume etc.

```
weekly_data = df.resample('W', on='Date').mean(numeric_only=True)
weekly_data
average_close_by_ticker = df.groupby('Ticker')['Close'].mean()
average_close_by_ticker
```

Ticker	Close
AAPL	158.333934
GOOG	100.631532
MSFT	275.039839
NFLX	327.614677

dtype: float64

```
rolling_mean_20 = df['Close'].rolling(window=20).mean()
rolling_mean_20
```



	Close
0	NaN
1	NaN
2	NaN
3	NaN
4	NaN
...	...
243	106.416500
244	106.470000
245	106.520000
246	106.533001
247	106.398750

247 rows × 1 columns

dtype: float64

Result: The program to implement programs for estimating & eliminating trend in time series data- aggregation, smoothing