Experiment 7: Time Series Decomposition into Trend and Seasonality

Aim:

To implement a program for decomposing time series data into trend, seasonality, and residuals using Microsoft stock data.

---

1. Importing Required Libraries

python
CopyEdit

```
import yfinance as yf
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.seasonal import seasonal_decompose
```

---

2. Loading the Dataset

python
CopyEdit

```
msft = yf.download("MSFT", start="2015-01-01", end="2024-12-31")
data = msft[['Close']].dropna()
```

---

3. Resampling the Data

- Since daily data can be noisy, we resample to **monthly average** for clearer patterns.

python
CopyEdit

```
monthly_data = data['Close'].resample('M').mean()
```

---

4. Time Series Decomposition

python
CopyEdit

```
decomposition = seasonal_decompose(monthly_data, model='additive',
period=12)
```

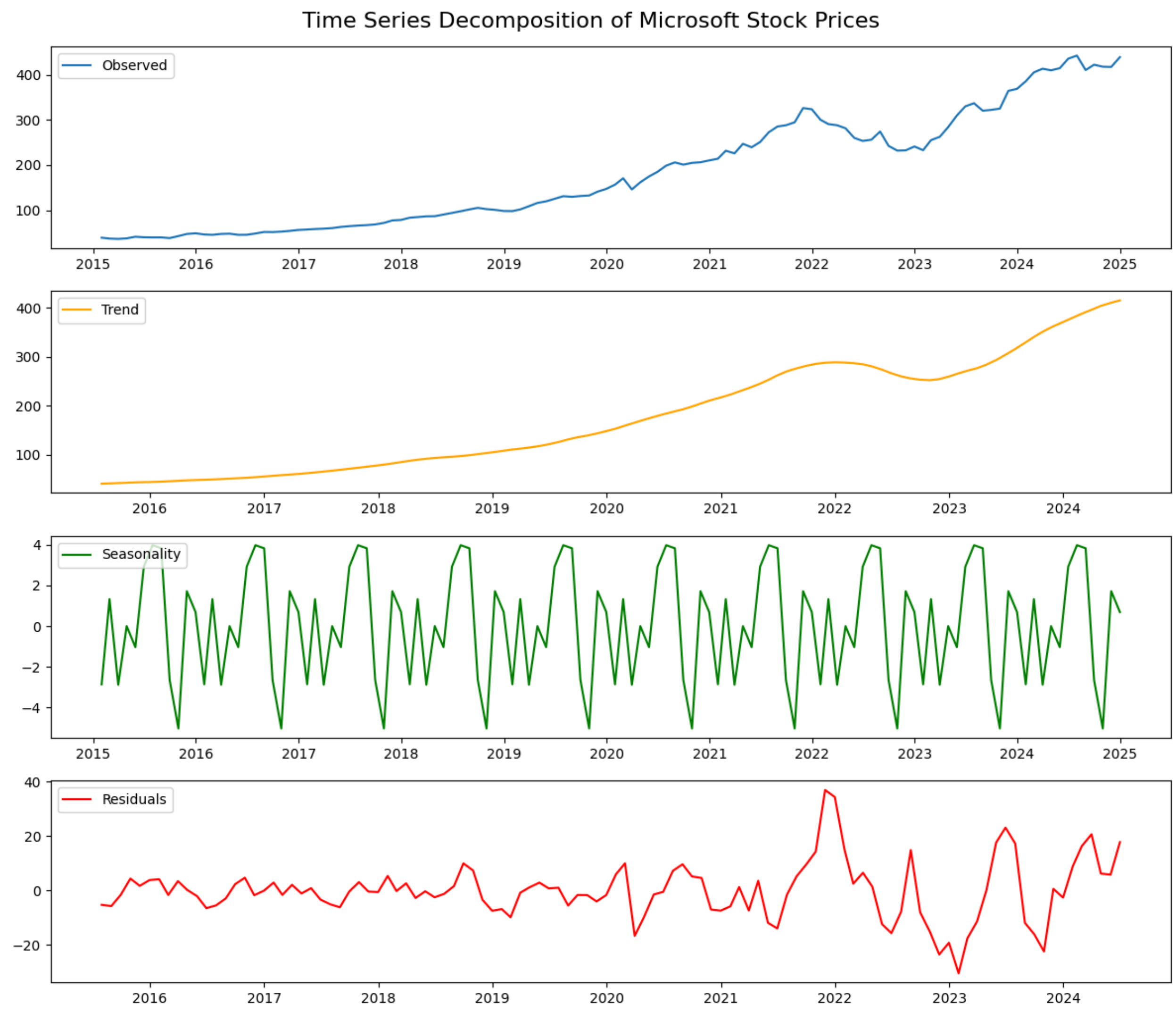
- `model='additive'` assumes time series is the sum of trend, seasonality, and residual.
- `period=12` implies **yearly seasonality** for monthly data.

---

5. Visualization

python
CopyEdit

```
plt.figure(figsize=(12, 10))

plt.subplot(411)
plt.plot(decomposition.observed, label='Observed')
plt.legend(loc='upper left')

plt.subplot(412)
plt.plot(decomposition.trend, label='Trend', color='orange')
plt.legend(loc='upper left')

plt.subplot(413)
plt.plot(decomposition.seasonal, label='Seasonality', color='green')
plt.legend(loc='upper left')

plt.subplot(414)
plt.plot(decomposition.resid, label='Residuals', color='red')
plt.legend(loc='upper left')

plt.tight_layout()
plt.suptitle("Time Series Decomposition of Microsoft Stock Prices",
fontsize=16, y=1.02)
plt.show()
```



Result:

Thus, the time series data for Microsoft stock prices was successfully decomposed into its **trend**, **seasonal**, and **residual** components.