Decomposing Time Series Data Into Trend And Seasonality

Aim:

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

Algorithm:

1. Import libraries for numerical computation, data manipulation, plotting, and time series decomposition.
2. Load the time series dataset, convert the date column to datetime, and set it as the DataFrame index.
3. Extract the target variable (e.g., "Passengers") from the dataset as the time series.
4. Apply the seasonal_decompose function with an additive (or multiplicative) model and specify the seasonal period (e.g., 12 for monthly data).
5. Extract the trend, seasonal, and residual components from the decomposition output.
6. Plot the original time series alongside its trend, seasonal, and residual components to analyze the underlying patterns.
7. End the decomposition and visualization process.

Program Code:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.holtwinters import ExponentialSmoothing

%matplotlib inline

url = 'https://raw.githubusercontent.com/jbrownlee/Datasets/master/airline-passengers.csv'
df = pd.read_csv(url, parse_dates=['Month'])
```

```
df.set_index('Month', inplace=True)
ts = df['Passengers']

ts.head()
```

```
           Passengers
Month
1949-01-01     112
1949-02-01     118
1949-03-01     132
1949-04-01     129
1949-05-01     121

dtype: int64
```

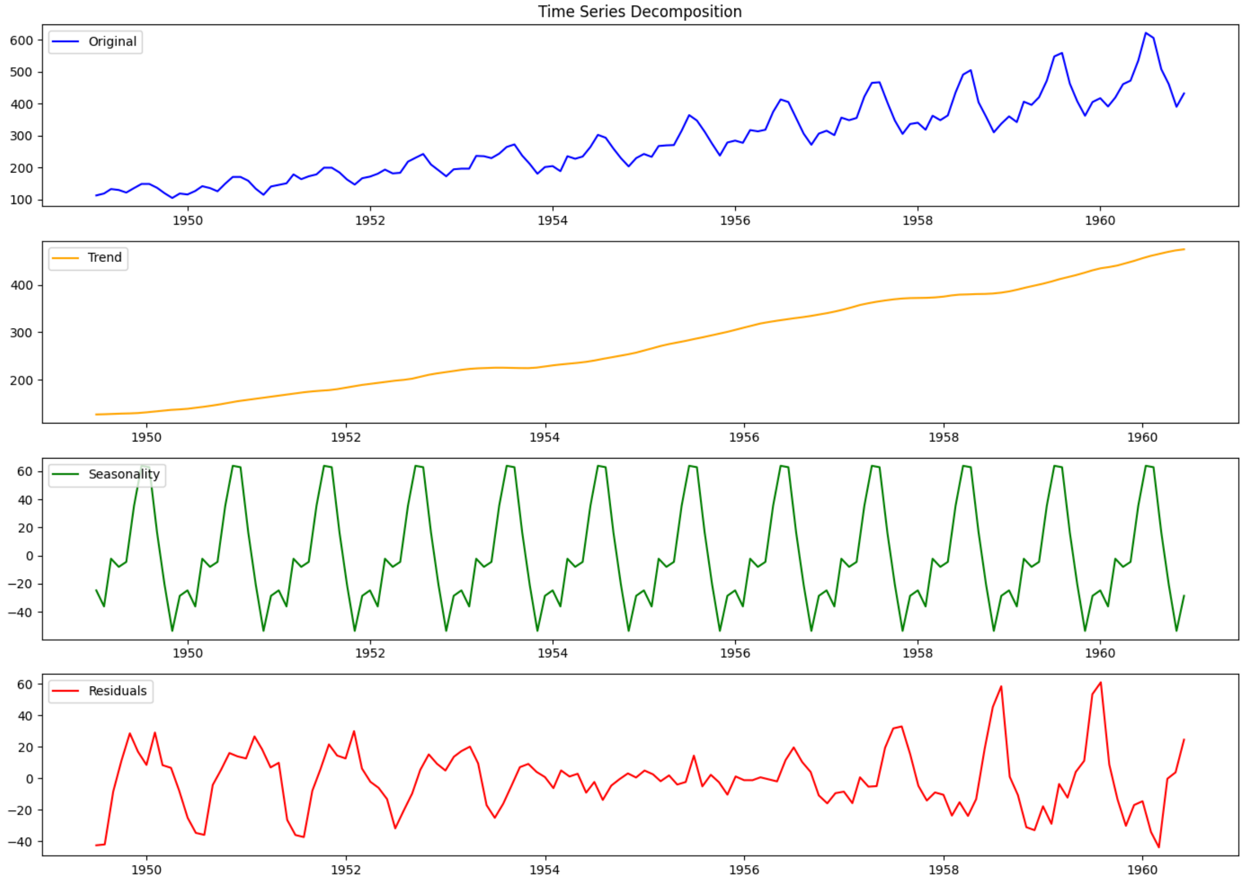
```
decomposition = seasonal_decompose(ts, model='additive', period=12)

trend = decomposition.trend
seasonal = decomposition.seasonal
residual = decomposition.resid

print("Trend Component:")
print(trend.head(15))
print("\nSeasonal Component:")
print(seasonal.head(15))
print("\nResidual Component:")
print(residual.head(15))
```

```
Trend Component:
Month
1949-01-01          NaN
1949-02-01          NaN
1949-03-01          NaN
1949-04-01          NaN
1949-05-01          NaN
1949-06-01          NaN
```

```
1949-07-01    126.791667
1949-08-01    127.250000
1949-09-01    127.958333
1949-10-01    128.583333
1949-11-01    129.000000
1949-12-01    129.750000
1950-01-01    131.250000
1950-02-01    133.083333
1950-03-01    134.916667
Name: trend, dtype: float64

Seasonal Component:
Month
1949-01-01   -24.748737
1949-02-01   -36.188131
1949-03-01    -2.241162
1949-04-01    -8.036616
1949-05-01    -4.506313
1949-06-01    35.402778
1949-07-01    63.830808
1949-08-01    62.823232
1949-09-01    16.520202
1949-10-01   -20.642677
1949-11-01   -53.593434
1949-12-01   -28.619949
1950-01-01   -24.748737
1950-02-01   -36.188131
1950-03-01    -2.241162
Name: seasonal, dtype: float64

Residual Component:
Month
1949-01-01          NaN
1949-02-01          NaN
1949-03-01          NaN
1949-04-01          NaN
```

```
1949-05-01          NaN
1949-06-01          NaN
1949-07-01   -42.622475
1949-08-01   -42.073232
1949-09-01    -8.478535
1949-10-01    11.059343
1949-11-01    28.593434
1949-12-01    16.869949
1950-01-01     8.498737
1950-02-01    29.104798
1950-03-01     8.324495
Name: resid, dtype: float64
```

```
plt.figure(figsize=(14, 10))

plt.subplot(411)
plt.plot(ts, label='Original', color='blue')
plt.legend(loc='upper left')
plt.title('Time Series Decomposition')

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

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

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

plt.tight_layout()
plt.show()
```



RESULTS:

The program has been created and implemented successfully for decomposing time series data into trend and seasonality.