Name: Shivam Pandey
Roll no:28
Experiment No. 7
Implement time series decomposition and moving averages method of trend estimation.
Date of Performance:19/03/2024
Date of Submission:04/04/2024

Academic Year: 2022-23 **Semester: VIII**
Class / Branch: BE Computer **Subject: Applied Data Science Lab**
Date of Performance: 19/03/2024 **Date of Completion:04/04/2024**

Experiment No. 7

Aim: Implement time series decomposition and moving averages method of trend estimation.

Dataset: Airline passenger dataset is used in the experiment. This dataset provides monthly totals of a US airline passengers from 1949 to 1960.

Software used: Google Colaboratory / Jupyter Notebook

Theory: -

A given time series is thought to consist of three systematic components including cycle, trend, seasonality, and one non-systematic component called noise.
In an additive decomposition, the components are decomposed in such a way that when they are added together, the original time series can be obtained.
Time series=Trend + Seasonality + Noise
In the case of multiplicative decomposition, the components are decomposed in the such a way that when they are multiplied together, the original time series can be derived back.
Time series=Trend * Seasonality * Noise
Both additive and multiplicative time series decomposition can be represented by these equations:
where Tt, St, and Et are trend, seasonal, and error components respectively.
The first step in a classical decomposition is to use a moving average method to estimate the trend-cycle

1. Moving Averages Method gives a trend with a fair degree of accuracy. In this method, we take arithmetic mean of the values for a certain time span. The time span can be three-years, four - years, five- years and so on depending on the data set and our interest. We will see the working procedure of this method.

Program:

[1]:
```
import pandas as  pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.seasonal import seasonal_decompose
```

[2]:
```
airline = pd.read_csv('airlinepassenger.csv')
```

[3]:
```
airline.head()
```

[3]:

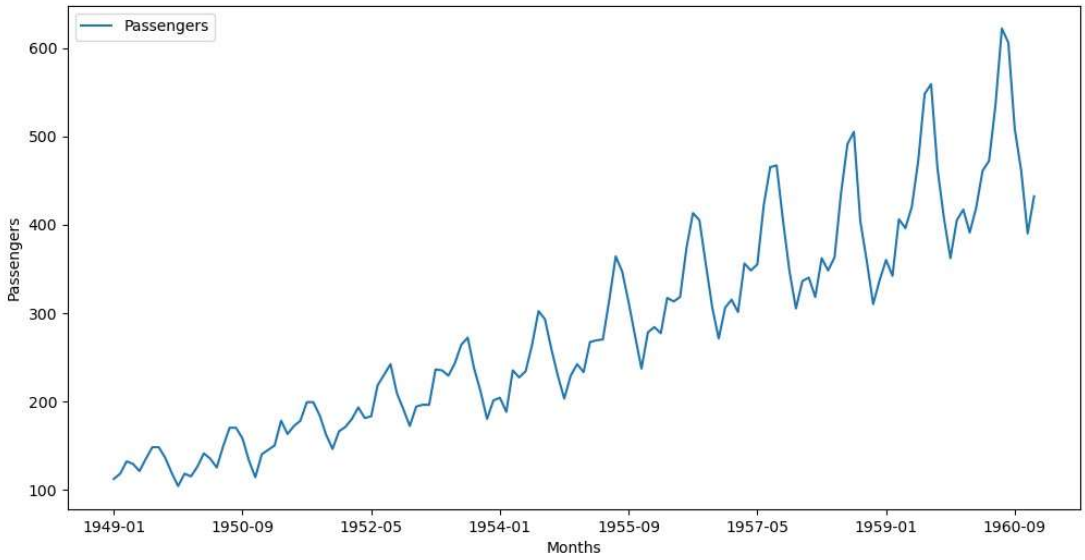
	Month	Passengers
0	1949-01	112
1	1949-02	118
2	1949-03	132
3	1949-04	129
4	1949-05	121

[4]:
```
airline.isnull().sum()
```

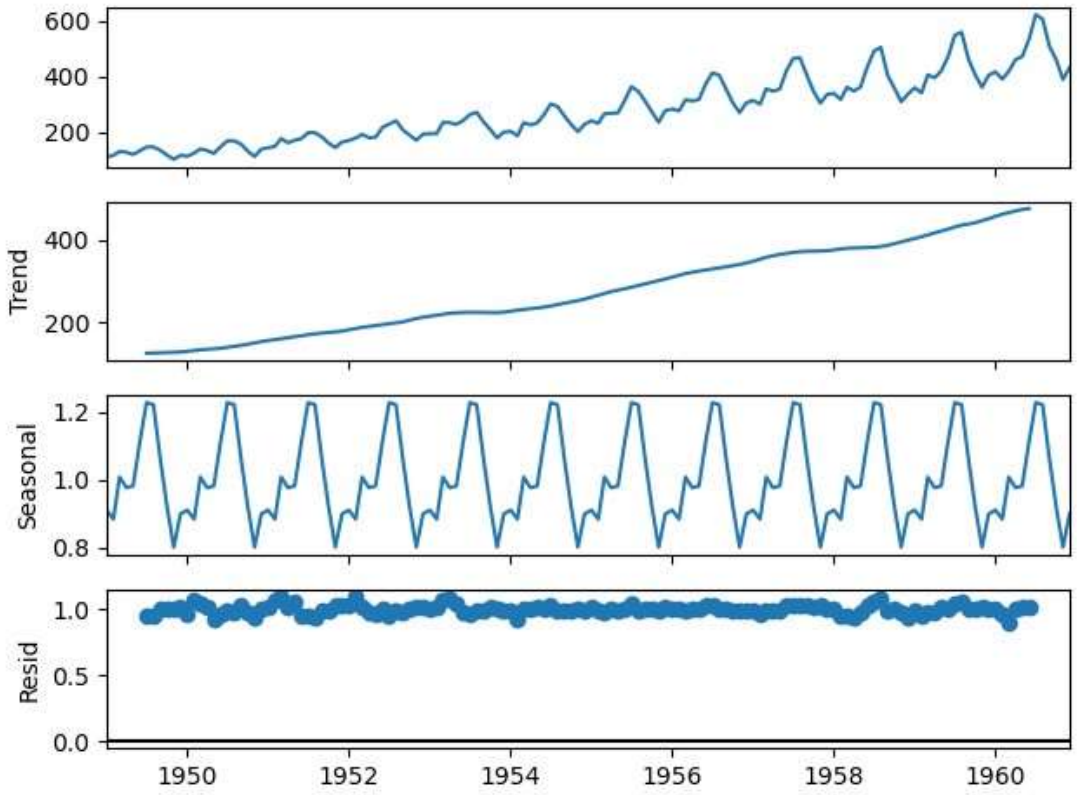
[4]:
```
Month        0
Passengers  0 dtype: int64
```

[5]:
```
airline.plot(x='Month',y='Passengers',figsize=(12,6))
plt.xlabel('Months')
plt.ylabel('Passengers')
```

[5]:
```
Text(0, 0.5, 'Passengers')
```



[6]:
```
airline['Month'] = pd.to_datetime(airline['Month'])
airline = airline.set_index('Month')
result = seasonal_decompose(airline,model='multiplicative')
result.plot()
plt.show()
```



[7]:
```
#Create Dummy Data
product = {
            'Month':[1,2,3,4,5,6,7,8,9,10,11,12],
            'Passenger':[290,260,288,300,310,303,329,340,316,330,308,310]
          }

product_df = pd.DataFrame(product)
```

[8]:
```
product_df.head()
```

[8]:

	Month	Passenger
0	1	290
1	2	260
2	3	288
3	4	300
4	5	310

[9]:
```
product_df['SMA_3'] =
product_df['Passenger'].rolling(window=3).mean()
product_df['SMA_4'] =
product_df['Passenger'].rolling(window=4).mean()
```

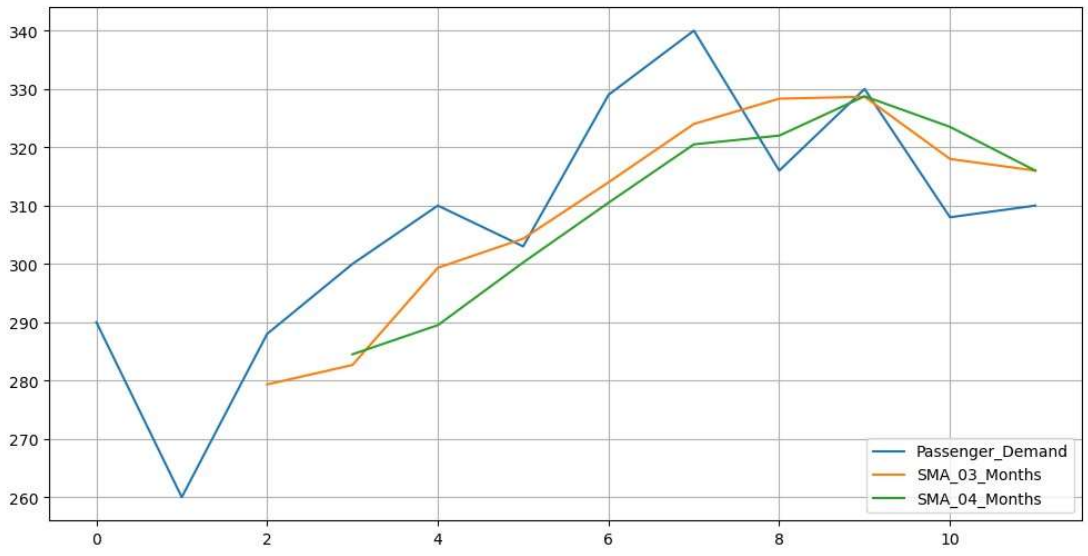
[10]:
```
product_df.head()
```

[10]:

	Month	Passenger	SMA_3	SMA_4
0	1	290	NaN	NaN
1	2	260	NaN	NaN
2	3	288	279.333333	NaN
3	4	300	282.666667	284.5
4	5	310	299.333333	289.5

[11]:
```
fig,ax= plt.subplots(figsize=(12,6))
plt.grid(True)
plt.plot(product_df['Passenger'],label='Passenger_Demand')
plt.plot(product_df['SMA_3'],label='SMA_03_Months')
plt.plot(product_df['SMA_4'],label='SMA_04_Months')
plt.legend(loc=4)
```

[11]: <matplotlib.legend.Legend at 0x2d885e6e290>



:

Conclusion: - Time series decomposition is one of the best ways to understand how a time series behaves. The statsmodels library provides an implementation of the naive, or classical, decomposition method in a function called seasonal_decompose().