Experiment No- 4

Aim :- Implement Time Series Analysis for rainfall in R/Python Programming.

Objective:- To understand the use of time series models for prediction.

Description:-

- Time series analysis is a specific way of analyzing a sequence of data points collected over an interval of time. In time series analysis, analysts record data points at consistent intervals over a set period of time rather than just recording the data points randomly.

A time-series data is a series of data points or observations recorded at different or regular time intervals. In general, a time series is a sequence of data points taken at equally spaced time intervals. The frequency of recorded data points may be hourly, daily, weekly, monthly, quarterly or annually.

Time-Series Forecasting is the process of using a statistical model to predict future values of a time-series based on past results.

Components of a Time-Series

- **Trend -** The trend shows a general direction of the time series data over a long period of time. A trend can be increasing(upward), decreasing(downward), or horizontal(stationary).
- **Seasonality -** The seasonality component exhibits a trend that repeats with respect to timing, direction, and magnitude. Some examples include an increase in water consumption in summer due to hot weather conditions.
- **Cyclical Component -** These are the trends with no set repetition over a particular period of time. A cycle refers to the period of ups and downs, booms and slums of a time series, mostly observed in business cycles. These cycles do not exhibit a seasonal variation but generally occur over a time period of 3 to 12 years depending on the nature of the time series.
- **Irregular Variation -** These are the fluctuations in the time series data which become evident when trend and cyclical variations are removed. These variations are unpredictable, erratic, and may or may not be random.

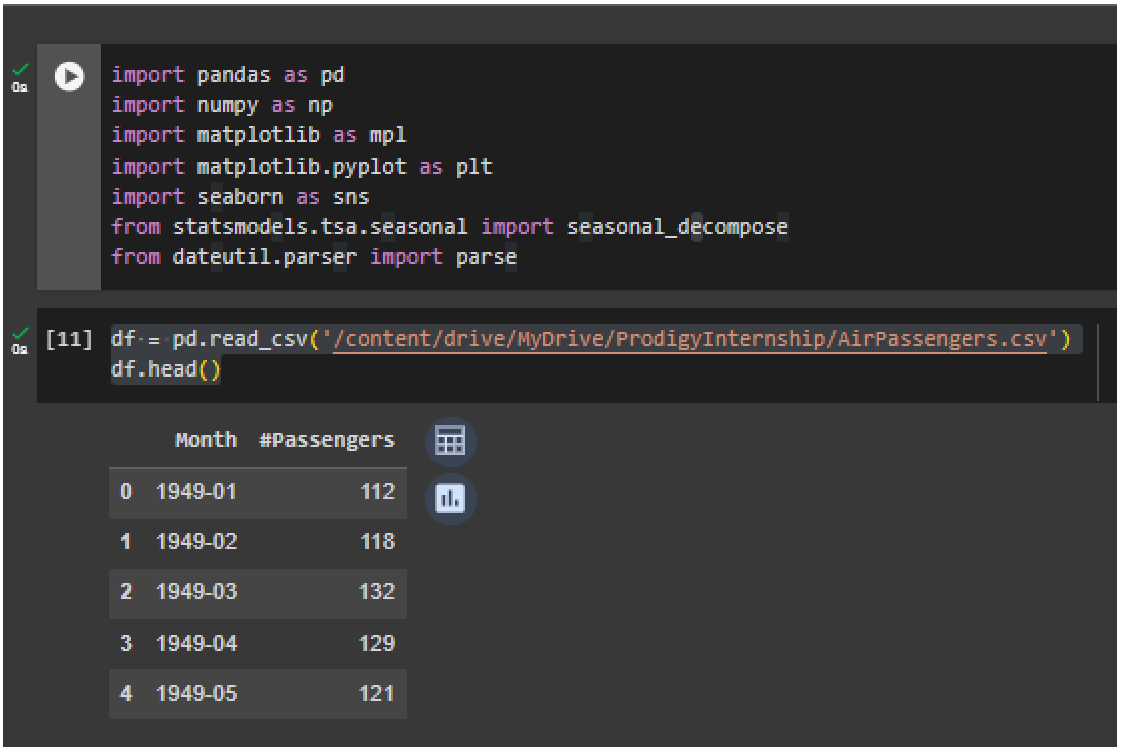
Different Time Intervals

The value of the frequency parameter in the ts() function decides the time intervals at which the data points are measured. A value of 12 indicates that the time series is for 12 months. Other values and its meaning is as below −

- frequency = 12 pegs the data points for every month of a year.
- frequency = 4 pegs the data points for every quarter of a year.

- frequency = 6 pegs the data points for every 10 minutes of an hour.
- frequency = 24*6 pegs the data points for every 10 minutes of a day.

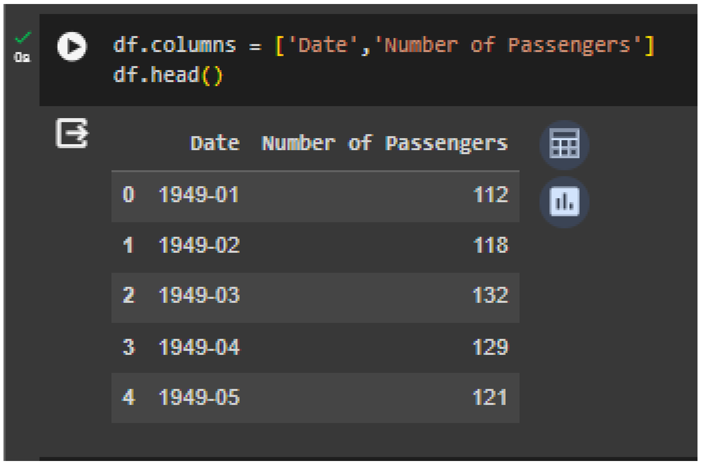
Program(Code) and Output:-



```
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.tsa.seasonal import seasonal_decompose
from dateutil.parser import parse
```

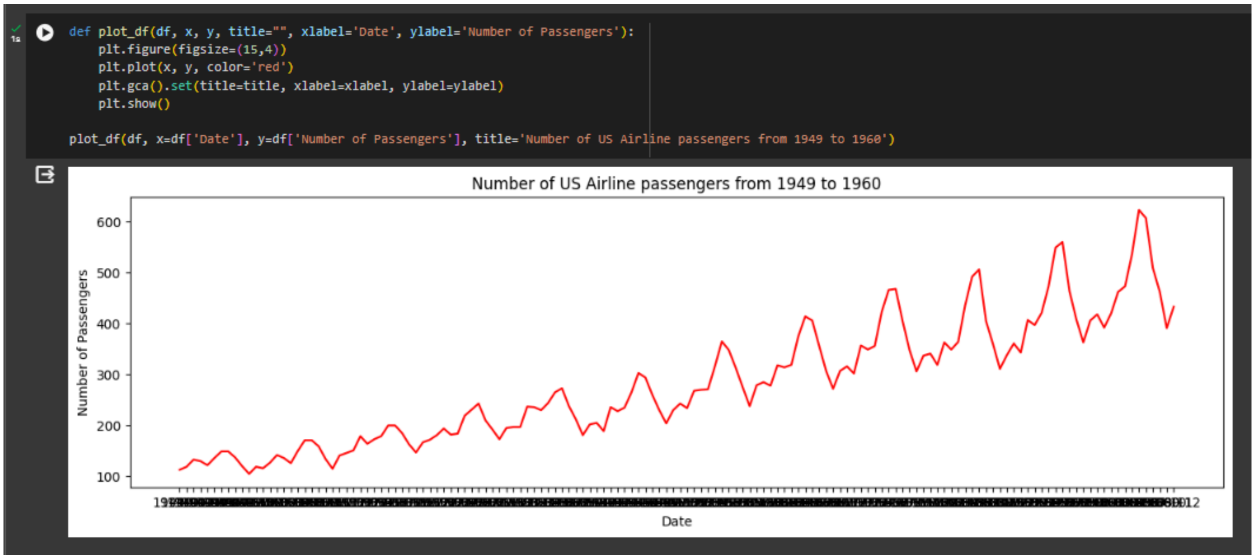
```
df = pd.read_csv('/content/drive/MyDrive/ProdigyInternship/AirPassengers.csv')
df.head()
```

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

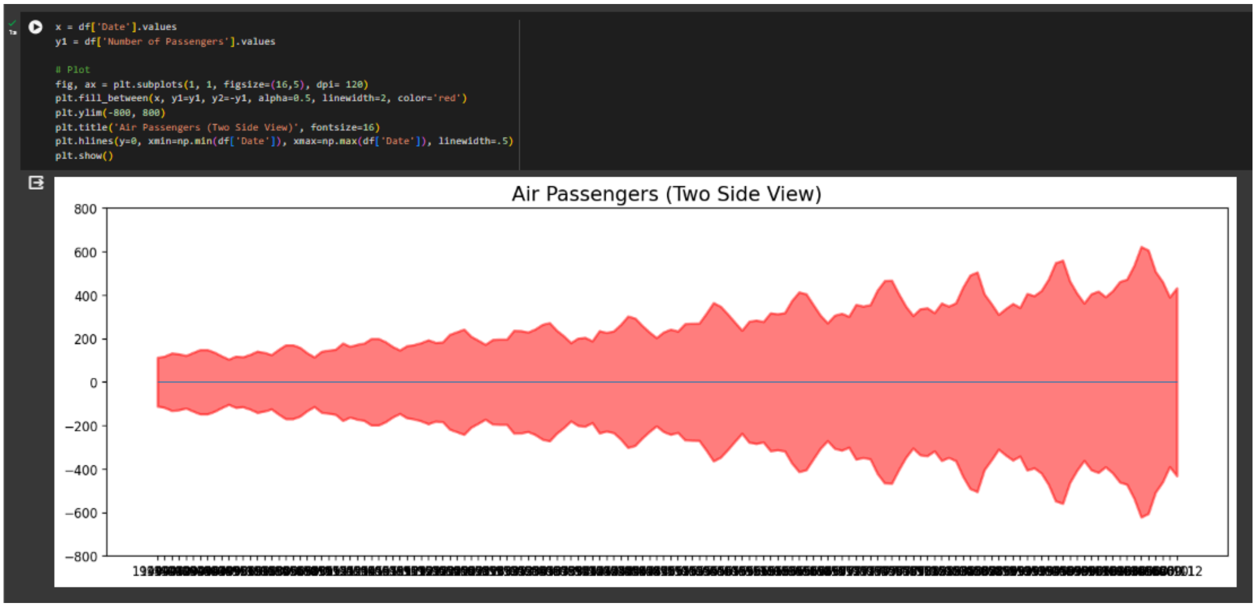


```
df.columns = ['Date','Number of Passengers']
df.head()
```

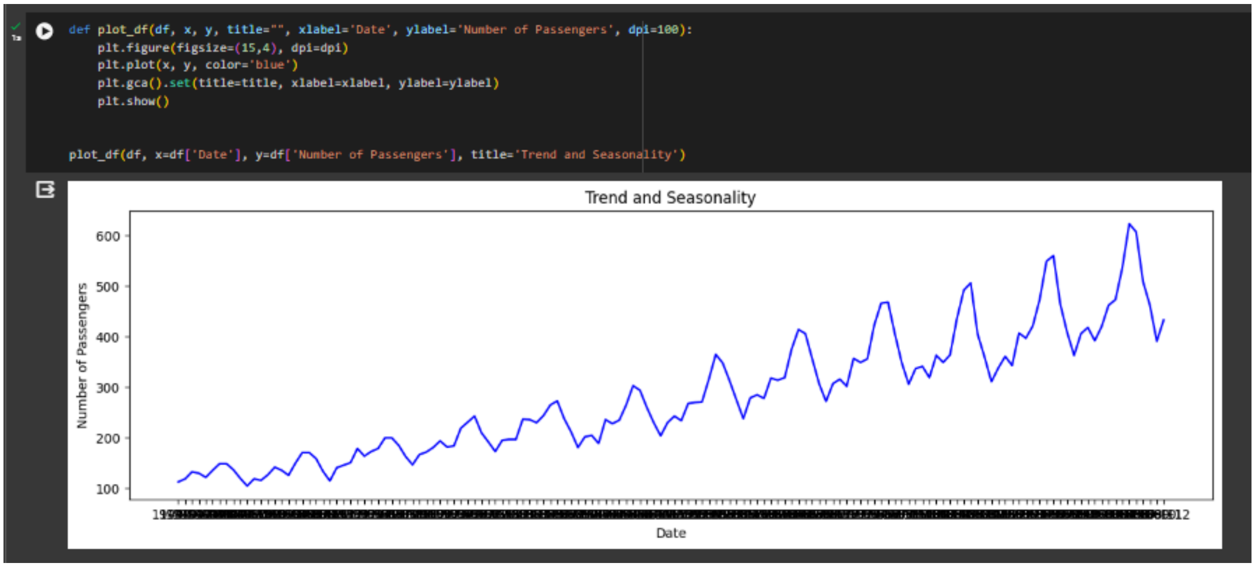
```
      Date  Number of Passengers
0  1949-01                   112
1  1949-02                   118
2  1949-03                   132
3  1949-04                   129
4  1949-05                   121
```



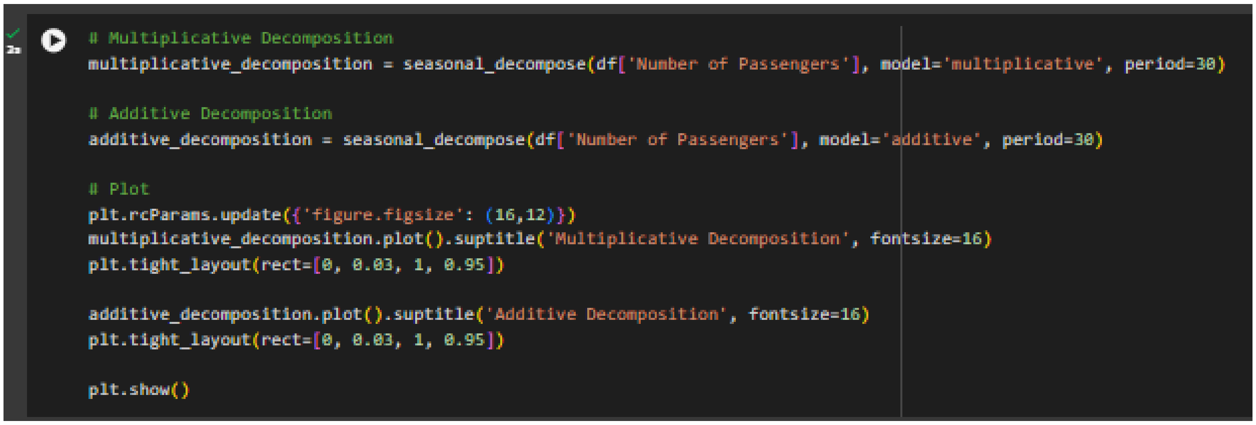
```
def plot_df(df, x, y, title="", xlabel='Date', ylabel='Number of Passengers'):
    plt.figure(figsize=(15,4))
    plt.plot(x, y, color='red')
    plt.gca().set(title=title, xlabel=xlabel, ylabel=ylabel)
    plt.show()

plot_df(df, x=df['Date'], y=df['Number of Passengers'], title='Number of US Airline passengers from 1949 to 1960')
```



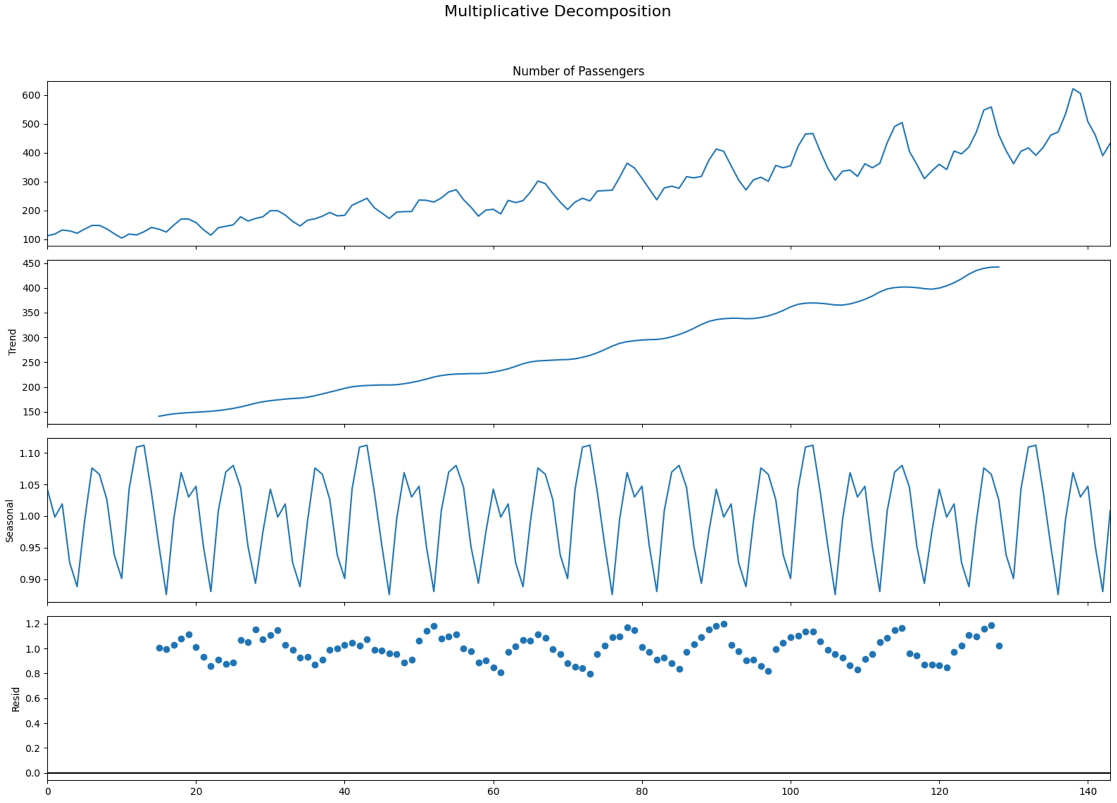
```
x = df['Date'].values
y1 = df['Number of Passengers'].values

# Plot
fig, ax = plt.subplots(1, 1, figsize=(16,5), dpi= 120)
plt.fill_between(x, y1=y1, y2=-y1, alpha=0.5, linewidth=2, color='red')
plt.ylim(-800, 800)
plt.title('Air Passengers (Two Side View)', fontsize=16)
plt.hlines(y=0, xmin=np.min(df['Date']), xmax=np.max(df['Date']), linewidth=.5)
plt.show()
```

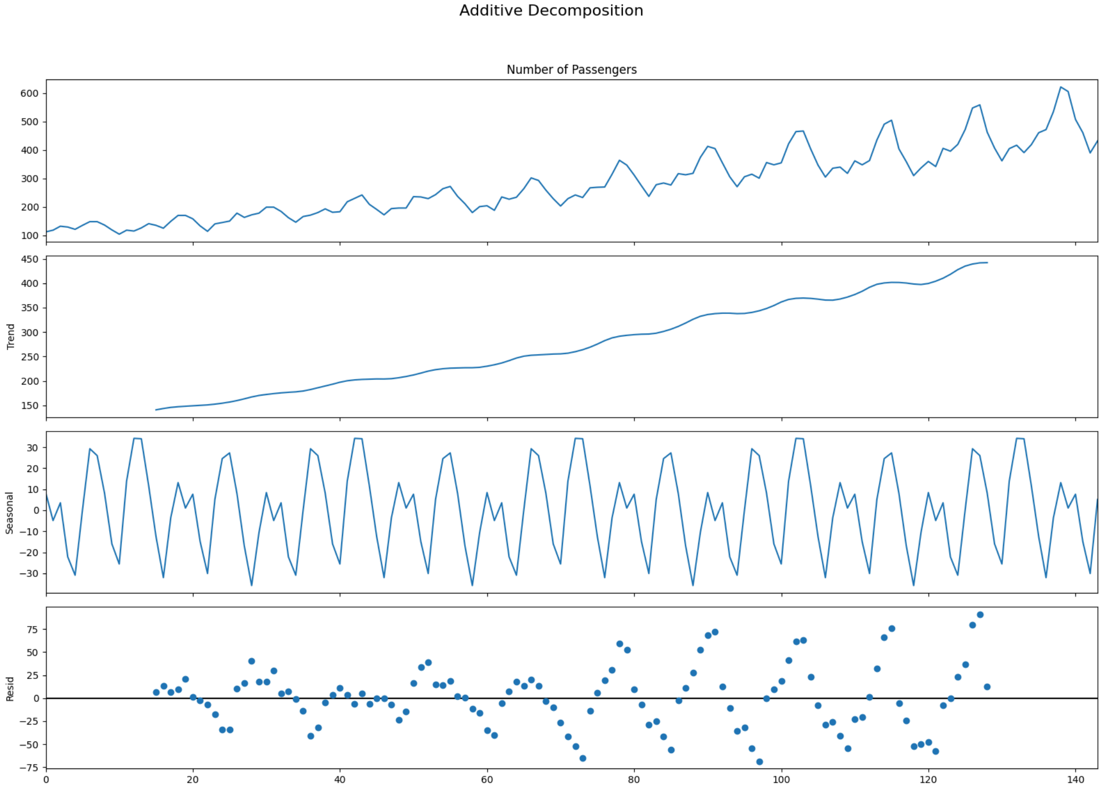


```python
def plot_df(df, x, y, title="", xlabel='Date', ylabel='Number of Passengers', dpi=100):
    plt.figure(figsize=(15,4), dpi=dpi)
    plt.plot(x, y, color='blue')
    plt.gca().set(title=title, xlabel=xlabel, ylabel=ylabel)
    plt.show()


plot_df(df, x=df['Date'], y=df['Number of Passengers'], title='Trend and Seasonality')
```



```python
# Multiplicative Decomposition
multiplicative_decomposition = seasonal_decompose(df['Number of Passengers'], model='multiplicative', period=30)

# Additive Decomposition
additive_decomposition = seasonal_decompose(df['Number of Passengers'], model='additive', period=30)

# Plot
plt.rcParams.update({'figure.figsize': (16,12)})
multiplicative_decomposition.plot().suptitle('Multiplicative Decomposition', fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.95])

additive_decomposition.plot().suptitle('Additive Decomposition', fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.95])

plt.show()
```


Multiplicative Decomposition
Number of Passengers
100
200
300
400
500
600
Trend
150
200
250
300
350
400
450
Seasonal
0.90
0.95
1.00
1.05
1.10
Resid
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
20
40
60
80
100
120
140


Additive Decomposition
Number of Passengers
100
200
300
400
500
600
Trend
150
200
250
300
350
400
450
Seasonal
−30
−20
−10
0
10
20
30
Resid
−75
−50
−25
0
25
50
75
0
20
40
60
80
100
120
140

Conclusion :-

1. An orderly set of data arranged in accordance with their time of occurrence is called **“time series”**

   —--

2. The graph of time series is called **“time series plot”**

   —--

3. Use of plt.fill_between()- **“fill the area between two horizontal curves”**

   —--