EX:No.2 221501046

Program to implement visualization techniques for time series data

Aim:

Write a program to visualize time series data using line plots, moving averages, seasonal patterns, and decomposition to understand trends, seasonality, and patterns in the data.

Algorithm:

Import Libraries: Load necessary libraries such as pandas, matplotlib, seaborn, and statsmodels.

1. Load Dataset: Upload and load the AirPassengers dataset into a DataFrame.
2. Convert & Set Index: Convert the 'Month' column to datetime and set it as the index.
3. Set Frequency: Ensure consistent time series frequency (Monthly Start).
4. Line Plot: Plot the original time series to observe overall trends.
5. Rolling Averages: Plot 12-month and 6-month moving averages to smooth the series.
6. Seasonal Decomposition: Use statsmodels to decompose the series into trend, seasonal, and residual components.
7. Subplot Visualization: Display all plots to visually analyze seasonality and variation in the dataset

Code:

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from google.colab import files

from statsmodels.tsa.seasonal import seasonal_decompose

uploaded = files.upload()

df = pd.read_csv('AirPassengers.csv')
```

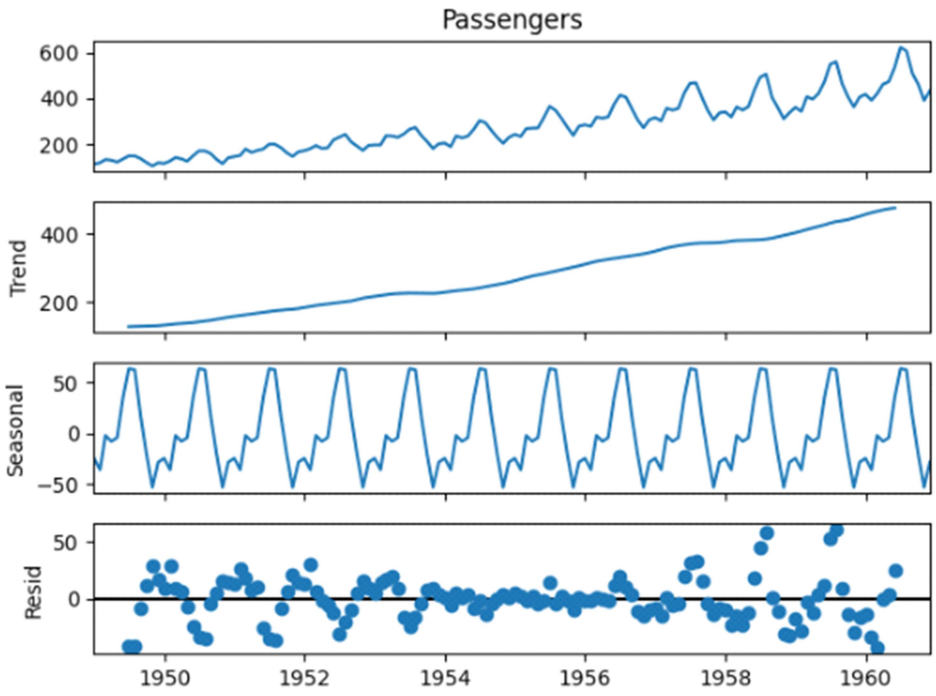
```
df.columns = ['Month', 'Passengers']
df['Month'] = pd.to_datetime(df['Month'])
df.set_index('Month', inplace=True)
df = df.asfreq('MS')
plt.figure(figsize=(10, 5))
plt.plot(df['Passengers'], label='Monthly Passengers')
plt.title('Monthly Air Passengers Over Time')
plt.xlabel('Date')
plt.ylabel('Passengers')
plt.legend()
plt.grid(True)
plt.show()
df['6_MA'] = df['Passengers'].rolling(window=6).mean()
df['12_MA'] = df['Passengers'].rolling(window=12).mean()
plt.figure(figsize=(10, 5))
plt.plot(df['Passengers'], label='Original')
plt.plot(df['6_MA'], label='6-Month MA')
plt.plot(df['12_MA'], label='12-Month MA')
plt.title('Moving Average Smoothing')
plt.xlabel('Date')
plt.ylabel('Passengers')
plt.legend()
plt.grid(True)
plt.show()
decomposition = seasonal_decompose(df['Passengers'], model='additive')
```

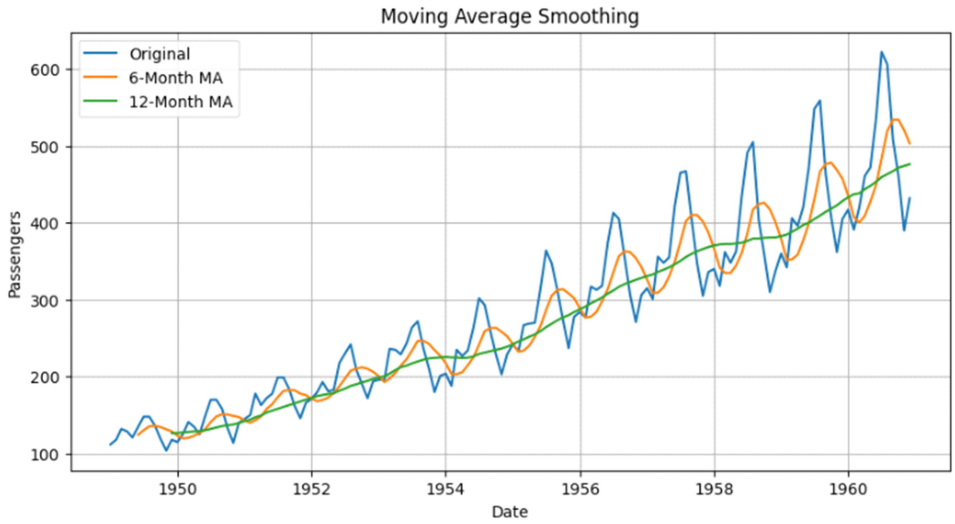
```
decomposition.plot()

plt.tight_layout()

plt.show()
```

Output:





Result:

Thus, the program using the time series data implementation has been done successfully.