**Exp no: 7 Program for decomposing time series data into**

**trend and seasonality.**

**Date: 8/4/25**

**Aim:**

The aim of this project is to analyze and decompose the AirPassengers time series dataset into its fundamental components—trend, seasonality, and residuals—using time series decomposition techniques in Python.

**Objectives:**

The objectives of the project include understanding the structure and behavior of time series data through decomposition, implementing the decomposition using Python libraries like statsmodels, extracting and visualizing the trend, seasonal, and residual components separately, and analyzing how these components contribute to the overall time series. Additionally, the project seeks to provide insights that can assist in forecasting and anomaly detection tasks, which are valuable in time series analysis.

**Background/Scope:**

Time series data often exhibit various patterns, such as trends, seasonality, and irregularities, which can be better understood when broken down into separate components. The AirPassengers dataset, which records monthly totals of international airline passengers from 1949 to 1960, is an ideal example for studying such patterns. Decomposing the data allows for the identification of long-term trends in air travel, seasonal fluctuations, and random noise or irregularities.

**Steps for Time Series Sales Data Preprocessing:**

**Step 1: Import Libraries and Load Dataset**

The first step is to load the Air Passenger dataset into a pandas DataFrame. The dataset contains monthly international airline passenger data indexed by date.

import pandas as pd

import matplotlib.pyplot as plt

url = 'https://raw.githubusercontent.com/jbrownlee/Datasets/master/airline-passengers.csv'

data = pd.read\_csv(url, parse\_dates=['Month'], index\_col='Month')

data.columns = ['Passengers']

**Step 2: Perform Time Series Decomposition**

We use the seasonal\_decompose() function from statsmodels to decompose the data into trend, seasonal, and residual components.

result = seasonal\_decompose(data['Passengers'], model='multiplicative', period=12)

**Step 3: Plot the Original (Observed) Time Series**

This step plots the original dataset to visualize the raw number of airline passengers over time.

plt.figure(figsize=(10, 4))

plt.plot(data['Passengers'], label='Observed')

plt.title('Original Time Series (Observed)')

plt.xlabel('Date')

plt.ylabel('Number of Passengers')

plt.legend()

plt.tight\_layout()

plt.show()

A graph showing the growth of the company's growth

AI-generated content may be incorrect.

**Step 4: Plot the Trend Component**

The trend component shows the long-term movement in the data, smoothed over a 12-month period.

plt.figure(figsize=(10, 4))

plt.plot(result.trend, label='Trend', color='orange')

plt.title('Trend Component')

plt.xlabel('Date')

plt.ylabel('Trend')

plt.legend()

plt.tight\_layout()

plt.show()

A graph with a line

AI-generated content may be incorrect.

**Step 5: Plot the Seasonal Component**

This step reveals repeating patterns over each year, which in this case shows strong seasonality in air travel.

plt.figure(figsize=(10, 4))

plt.plot(result.seasonal, label='Seasonality', color='green')

plt.title('Seasonal Component')

plt.xlabel('Date')

plt.ylabel('Seasonality')

plt.legend()

plt.tight\_layout()

plt.show()

A green line graph with numbers

AI-generated content may be incorrect.

**Step 6: Plot the Residual Component**

Finally, we extract and plot the residuals, which are the random variations left after removing trend and seasonality.

plt.figure(figsize=(10, 4))

plt.plot(result.resid, label='Residuals', color='red')

plt.title('Residual Component')

plt.xlabel('Date')

plt.ylabel('Residuals')

plt.legend()

plt.tight\_layout()

plt.show()

A graph showing a graph of a graph

AI-generated content may be incorrect.

**Result:**

Thus, the program to estimate and eliminate the trend in the AirPassengers time series data using aggregation and smoothing techniques has been implemented successfully.