**EX:No.7 221501028**

**4/04/25**

**IMPLEMENT PROGRAM FOR DECOMPOSING TIME SERIES DATA INTO TREND AND SEASONALITY**

**AIM:**

To develop a Python program to decompose the time series data derived from the "Autism Screening for Toddlers" dataset into trend and seasonality components, simulating a time series, and visualizing the results.

**ALGORITHM:**

1. Load the dataset and simulate a time series by assigning synthetic timestamps.
2. Calculate the total score from screening variables (A1 to A10) as the time series variable.
3. Decompose the time series into trend, seasonality, and residual components using a seasonal decomposition method.
4. Visualize the original time series along with its trend, seasonality, and residual components.
5. Analyze the decomposition to understand the underlying patterns in the data.

**PROCESS:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from statsmodels.tsa.seasonal import seasonal\_decompose

**# Load the Autism Screening for Toddlers dataset**

df = pd.read\_csv('autism.csv') # Replace with actual file path

**# Simulate a time series by adding a timestamp (since the dataset lacks explicit time data)**

date\_rng = pd.date\_range(start='2018-01-01', periods=len(df), freq='D') # Daily frequency for demo

df['timestamp'] = date\_rng

df.set\_index('timestamp', inplace=True)

**# Calculate Total\_Score as the sum of A1 to A10 scores**

df['Total\_Score'] = df[['A1\_Score', 'A2\_Score', 'A3\_Score', 'A4\_Score', 'A5\_Score',

'A6\_Score', 'A7\_Score', 'A8\_Score', 'A9\_Score', 'A10\_Score']].sum(axis=1)

**# Clean data (handle any potential missing values)**

df['Total\_Score'] = df['Total\_Score'].fillna(method='ffill').fillna(method='bfill').fillna(0)

**# Ensure the time series has enough data for decomposition (minimum period for seasonality)**

if len(df) < 12: # Assuming monthly seasonality; adjust based on data

raise ValueError("Dataset too small for seasonal decomposition. Need more data points.")

**# Perform seasonal decomposition (assuming additive model and 7-day seasonality for demo)**

decomposition = seasonal\_decompose(df['Total\_Score'], model='additive', period=7)

**# Plot the decomposed components**

plt.figure(figsize=(12, 8))

plt.subplot(411)

plt.plot(df['Total\_Score'], label='Original Data', color='blue')

plt.legend(loc='upper left')

plt.title('Original Time Series')

plt.subplot(412)

plt.plot(decomposition.trend, label='Trend', color='red')

plt.legend(loc='upper left')

plt.title('Trend Component')

plt.subplot(413)

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

plt.legend(loc='upper left')

plt.title('Seasonal Component')

plt.subplot(414)

plt.plot(decomposition.resid, label='Residual', color='black')

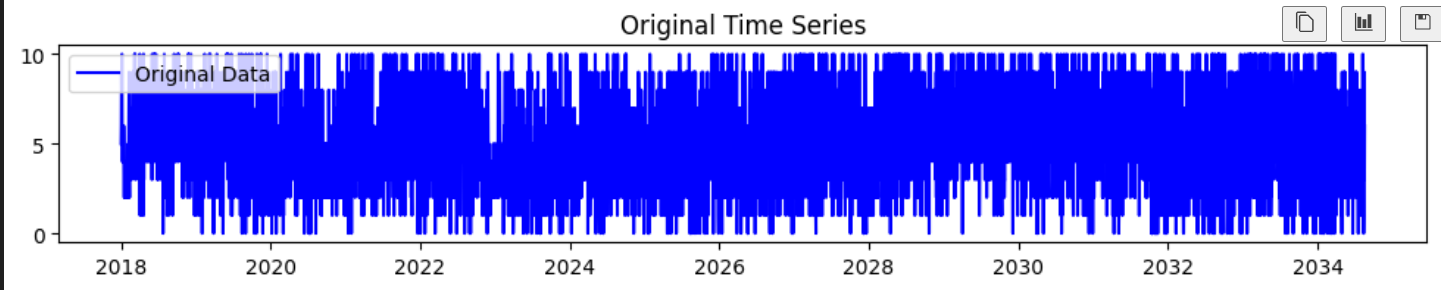
plt.legend(loc='upper left')

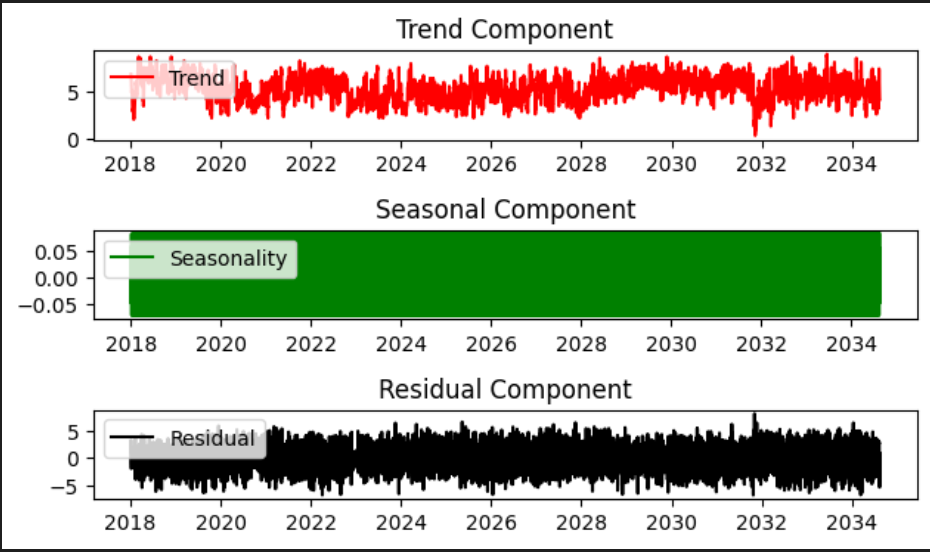
plt.title('Residual Component')

plt.tight\_layout()

plt.show()

**OUTPUT:**





**RESULT:**

The program successfully simulates a time series from the "Autism Screening for Toddlers" dataset, decomposes it into trend and seasonality components using seasonal decomposition, and visualizes the original data along with its trend, seasonality, and residual components.