EX:No.7 221501062

**Implement program for decomposing time series data into trend and seasonality**

**Aim:**

Write a program for decomposing time series data into trend and seasonality

**Algorithm:**

**1.Import Required Libraries**

* **Use pandas for data handling.**
* **Use matplotlib.pyplot for plotting.**
* **Use seasonal\_decompose from statsmodels.**

**2.Load the Dataset**

* **Read the CSV file (weather\_data.csv) using pandas.read\_csv.**
* **Convert the 'Date' column to datetime and set it as the index.**

**3.Select the Target Time Series Column**

* **Choose a column like 'Temperature' for decomposition.**

**4.Apply Seasonal Decomposition**

* **Use seasonal\_decompose() with model type:**
  + **'additive' (if variations are roughly constant over time).**
  + **Set period=365 for yearly seasonality (adjust based on data granularity).**
* **This will return:**
  + **observed: the original data**
  + **trend: the long-term movement**
  + **seasonal: recurring pattern**
  + **resid: random noise**

**5.Plot the Components**

* **Create subplots to display:**
  + **Observed data**
  + **Trend component**
  + **Seasonal component**
  + **Residual component**

**6.Display the Plot**

* **Use plt.tight\_layout() and plt.show() for a clean, readable output.**

**Code:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from statsmodels.tsa.seasonal import seasonal\_decompose

df = pd.read\_csv(r"C:\Users\kishore\Downloads\pune\_precipitaion\_data.csv", parse\_dates=['date'], index\_col='date')

time\_series = df['p']

decomposition = seasonal\_decompose(time\_series, model='additive', period=365) # adjust 'period' as per your data frequency

plt.figure(figsize=(14, 10))

plt.subplot(411)

plt.plot(decomposition.observed, label='Observed', color='blue')

plt.legend(loc='upper left')

plt.subplot(412)

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

plt.legend(loc='upper left')

plt.subplot(413)

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

plt.legend(loc='upper left')

plt.subplot(414)

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

plt.legend(loc='upper left')

plt.tight\_layout()

plt.show()

**Output:**

**A screenshot of a graph

AI-generated content may be incorrect.**

**Result:**

Thus, the program for decomposing time series data into trend and seasonality was done.