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

**Aim:**

To apply **time series decomposition** on using COVID 19 data set.

**Procedure:**

1. **Import Libraries**:
   * Import the required libraries: numpy, pandas, matplotlib.pyplot, and seasonal\_decompose from statsmodels.
2. **Define Functions**:
   * **Moving Average Smoothing**: Create a function to apply moving average smoothing to a given dataset using a specified window size.
   * **Decompose Time Series**: Create a function to decompose the time series data into trend, seasonality, and residuals.
3. **Load the COVID-19 Dataset**:
   * Load the dataset using pandas.read\_csv() and clean column names by stripping any spaces.
4. **Preprocess the Data**:
   * Convert the Date column to datetime format and set it as the index.
   * Sort the data in chronological order.
5. **Aggregate the Daily Confirmed Cases**:
   * Group the dataset by Date and sum the confirmed cases.
   * Calculate the daily new cases by finding the difference between consecutive days.
6. **Apply Moving Average Smoothing**:
   * Apply a moving average smoothing technique with a window size of 5 to the daily\_cases data.
7. **Decompose the Time Series**:
   * Use the seasonal\_decompose method to decompose the daily\_cases data into trend, seasonal, and residual components, with a period of 7 (weekly seasonality).
8. **Plot the Original vs Smoothed Data**:
   * Plot the original daily cases and the smoothed data on the same graph, using dashed lines for the original and solid lines for the smoothed data.
9. **Plot the Decomposition Results**:
   * Plot the trend, seasonal, and residual components of the decomposition in separate subplots.
10. **Display the Plots**:
    * The first compares the original and smoothed daily cases.
    * The second set shows the decomposed components (trend, seasonality, and residuals).

**Code:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from statsmodels.tsa.seasonal import seasonal\_decompose

def moving\_average\_smoothing(data, window\_size):

“ “ “Apply moving average smoothing to the data. “ “ “

return data.rolling(window=window\_size, min\_periods=1).mean()

def decompose\_time\_series(data, period):

“””Decompose time series data into trend, seasonality, and residuals.””””

return seasonal\_decompose(data, period=period, model='additive')

df = pd.read\_csv("/content/time-series-19-covid-combined.csv")

df.columns = df.columns.str.strip()

df["Date"] = pd.to\_datetime(df["Date"])

df.set\_index("Date", inplace=True)

df.sort\_index(inplace=True)

daily\_df = df.groupby(df.index).sum()

daily\_df["daily\_cases"] = daily\_df["Confirmed"].diff()

daily\_df = daily\_df.dropna(subset=["daily\_cases"]) # Drop initial NaN after diff

time\_series\_data = daily\_df["daily\_cases"]

window\_size = 5

smoothed\_data = moving\_average\_smoothing(time\_series\_data, window\_size)

period = 7

decomposition = decompose\_time\_series(time\_series\_data, period)

plt.figure(figsize=(12, 5))

plt.plot(time\_series\_data, label='Original Daily Cases', linestyle='dashed', alpha=0.6)

plt.plot(smoothed\_data, label=f'Smoothed (Window={window\_size})', linewidth=2, color='orange')

plt.legend()

plt.title('Moving Average Smoothing of Daily COVID-19 Cases')

plt.xlabel('Date')

plt.ylabel('Cases')

plt.tight\_layout()

plt.show()

fig, axes = plt.subplots(3, 1, figsize=(12, 8), sharex=True)

decomposition.trend.plot(ax=axes[0], title='Trend')

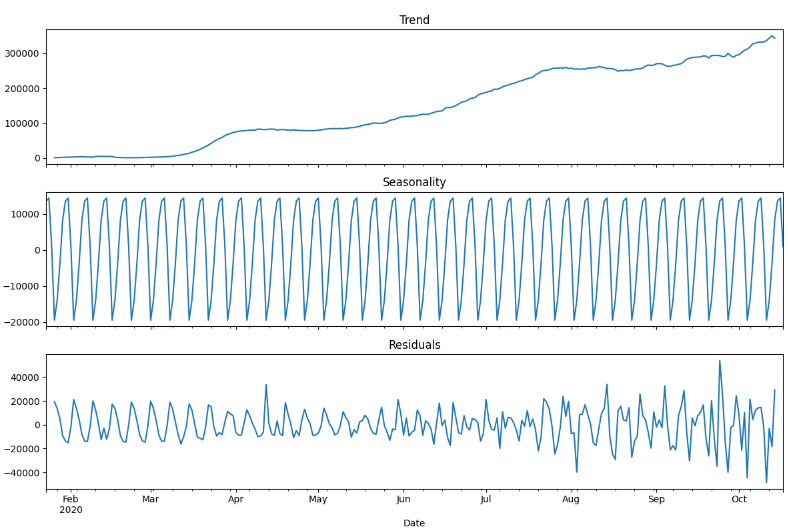
decomposition.seasonal.plot(ax=axes[1], title='Seasonality')

decomposition.resid.plot(ax=axes[2], title='Residuals')

plt.xlabel("Date")

plt.tight\_layout()

# output:



**Result:**

The program to decompose the covid 19 dataset has been successfully implemented and verified