**EX:No.5 221501028**

**27/03/25**

**IMPLEMENT PROGRAMS FOR ESTIMATING & ELIMINATING TREND IN TIME SERIES DATA- AGGREGATION, SMOOTHING**

**AIM:**

To develop a Python program that estimates and eliminates trends in the "Autism Screening for Toddlers" dataset by simulating a time series, using aggregation and smoothing techniques, 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) and clean the data.
3. Aggregate the data to a weekly level to estimate the trend.
4. Apply a moving average to smooth the data and further estimate the trend.
5. Detrend the data by subtracting the smoothed series and visualize the results.

**PROCESS:**

**i**mport pandas as pd

import numpy as np

import matplotlib.pyplot as plt

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

df = pd.read\_csv('Toddler Autism dataset July 2018.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)

**# Aggregate daily data to weekly data using mean**

weekly\_data = df['Total\_Score'].resample('W').mean()

**# Plot aggregated weekly data**

plt.figure(figsize=(12, 4))

weekly\_data.plot(title='Weekly Aggregated Total Autism Screening Scores')

plt.xlabel('Date')

plt.ylabel('Total Score')

plt.show()

**# Apply 7-day moving average for smoothing on the original daily data**

df['smoothed'] = df['Total\_Score'].rolling(window=7, center=True).mean()

**# Plot original and smoothed data**

plt.figure(figsize=(12, 4))

df[['Total\_Score', 'smoothed']].plot(title='Smoothed Total Autism Screening Scores (7-day MA)')

plt.xlabel('Date')

plt.ylabel('Total Score')

plt.show()

**# Detrend by subtracting the smoothed series from the original**

df['detrended'] = df['Total\_Score'] - df['smoothed']

**# Plot detrended data**

plt.figure(figsize=(12, 4))

df['detrended'].plot(title='Detrended Total Autism Screening Scores')

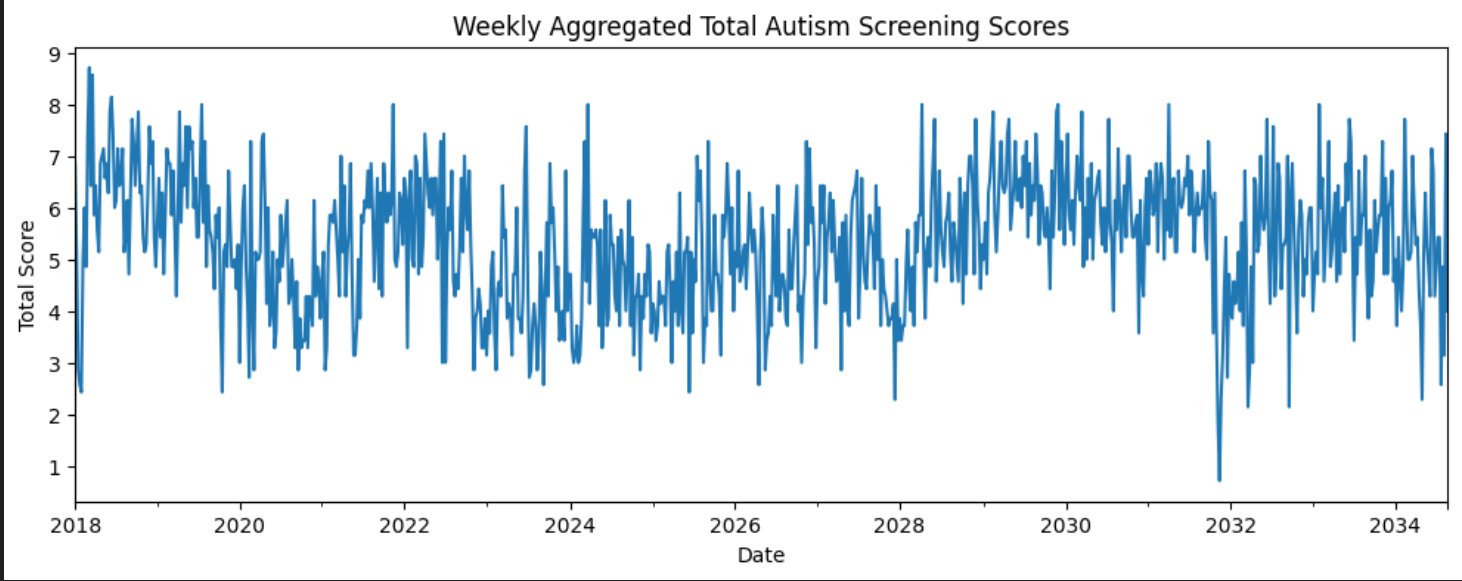
plt.axhline(0, color='gray', linestyle='--')

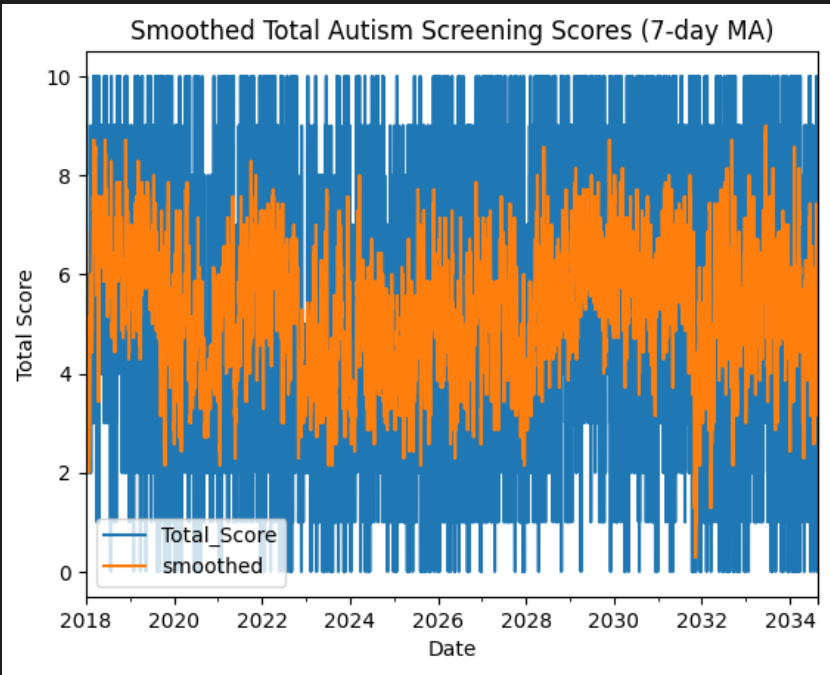
plt.xlabel('Date')

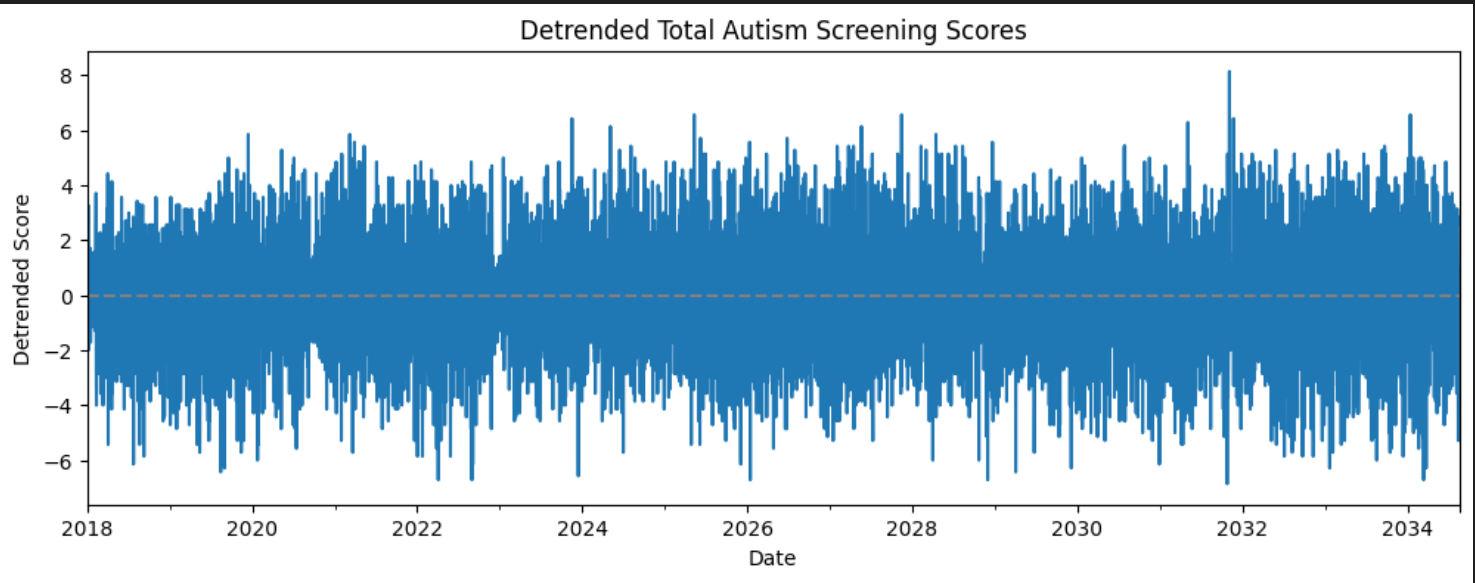
plt.ylabel('Detrended Score')

plt.show()

**OUTPUT:**

****

****

****

**RESULT:**

The program successfully simulates a time series from the "Autism Screening for Toddlers" dataset, estimates trends using aggregation and smoothing, and eliminates them through detrending.