**EX:No.5 221501012**

**25/03/25**

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

**AIM:**

To Implement programs for estimating & eliminating trend in time series data- aggregation, smoothing.

**ALGORITHM:**

1. **Import Libraries**: Load required Python libraries (pandas, numpy, matplotlib, seaborn).
2. **Load Dataset**: Read the Electric\_Production.csv file using pandas.read\_csv().
3. **Preprocess Data**:
   * Strip leading/trailing spaces from column names.
   * Check if the required column (IPG2211A2N) exists.
   * Convert the DATE column to datetime format and set it as the index.
   * Handle missing values by replacing them with the mean.
4. **Aggregation**:
   * Resample data to compute **monthly mean** (M).
   * Resample data to compute **yearly mean** (Y).
5. **Smoothing**:
   * Compute **3-month rolling mean** to reduce short-term fluctuations.
   * Compute **12-month rolling mean** for long-term trend analysis.
6. **Visualization**:
   * Plot **original vs. aggregated data** (monthly, yearly).
   * Plot **original vs. smoothed data** (3-month, 12-month rolling means).
7. **Output Result**: Display a confirmation message.

**PROCESS:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

**# Load the stock data**

file\_path = r'Electric\_Production.csv'

data = pd.read\_csv(file\_path)

**# Ensure column names have no leading/trailing spaces**

data.columns = data.columns.str.strip()

**# Check if the relevant column exists**

if 'IPG2211A2N' not in data.columns:

raise KeyError("'IPG2211A2N' column not found in the dataset. Please check the column name.")

**# Convert 'DATE' column to datetime format**

data['DATE'] = pd.to\_datetime(data['DATE'])

data.set\_index('DATE', inplace=True)

**# Handling Missing Values**

data['IPG2211A2N'].fillna(data['IPG2211A2N'].mean(), inplace=True)

**# Aggregation: Resampling data to monthly and yearly means**

monthly\_data = data['IPG2211A2N'].resample('M').mean()

yearly\_data = data['IPG2211A2N'].resample('Y').mean()

**# Smoothing: Using rolling mean to remove short-term fluctuations**

rolling\_mean\_3 = data['IPG2211A2N'].rolling(window=3).mean()

rolling\_mean\_12 = data['IPG2211A2N'].rolling(window=12).mean()

**# Plotting original data vs aggregated and smoothed data**

plt.figure(figsize=(12, 6))

plt.plot(data['IPG2211A2N'], label='Original Data', alpha=0.5)

plt.plot(monthly\_data, label='Monthly Aggregated', linestyle='dashed')

plt.plot(yearly\_data, label='Yearly Aggregated', linestyle='dotted')

plt.title('Electric Production: Aggregated Data')

plt.xlabel('Time')

plt.ylabel('Electric Production')

plt.legend()

plt.grid(True)

plt.show()

plt.figure(figsize=(12, 6))

plt.plot(data['IPG2211A2N'], label='Original Data', alpha=0.5)

plt.plot(rolling\_mean\_3, label='3-month Rolling Mean', linestyle='dashed')

plt.plot(rolling\_mean\_12, label='12-month Rolling Mean', linestyle='dotted')

plt.title('Electric Production: Smoothing Techniques')

plt.xlabel('Time')

plt.ylabel('Electric Production')

plt.legend()

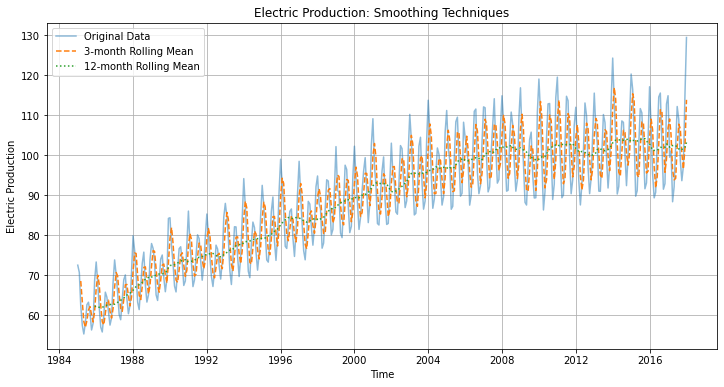
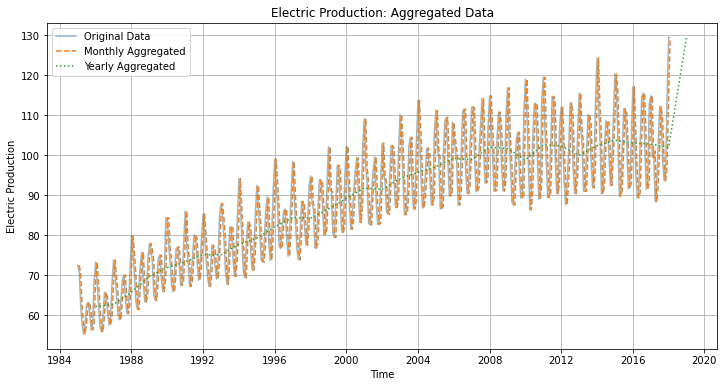
plt.grid(True)

plt.show()

**# Result:**

print("Aggregation and smoothing techniques applied successfully on the time series dataset.")

**OUTPUT:**



**RESULT:**

The program to Implement estimating & eliminating trend in time series data- aggregation, smoothing is created and executed successfully.