**EXP-5 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.

**PROCEDURE:**

1. Import the necessary libraries

import matplotlib.pyplot as plt

import numpy as np

from statsmodels.tsa.api import SimpleExpSmoothing

import pandas as pd

1. Load the time series dataset.

df = pd.read\_csv("/content/PRICE\_AND\_DEMAND\_201801\_NSW1.csv")

1. Convert SETTLEMENTDATE to datetime and set as index

f['SETTLEMENTDATE'] = pd.to\_datetime(df['SETTLEMENTDATE'])

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

1. Select only TOTALDEMAND column

df = df[['TOTALDEMAND']]

1. Plot Original Time Series Data

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

plt.plot(df, label='Original Demand Data', color='blue', alpha=0.7)

plt.title('Electricity Demand Time Series')

plt.xlabel('Time')

plt.ylabel('Total Demand')

plt.legend()

plt.show()

1. Aggregation (Resampling to Daily Mean)

df\_daily = df.resample('D').mean()

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

plt.plot(df, label='Original Data', alpha=0.5)

plt.plot(df\_daily, label='Aggregated (Daily Mean)', linewidth=2, color='red')

plt.title('Aggregation (Daily Mean)')

plt.legend()

plt.show()

1. Moving Average Smoothing (7-day window)

df['MA\_rolling'] = df['TOTALDEMAND'].rolling(window=7).mean()

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

plt.plot(df['TOTALDEMAND'], label='Original Data', alpha=0.5)

plt.plot(df['MA\_rolling'], label='Moving Average (7-day)', linewidth=2, color='orange')

plt.title('Moving Average Smoothing')

plt.legend()

plt.show()

1. Exponential Smoothing (alpha=0.2)

model=SimpleExpSmoothing(df['TOTALDEMAND'].dropna()).fit(smoothing\_level=0.2, optimized=False)

df['Exp\_Smoothing'] = model.fittedvalues

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

plt.plot(df['TOTALDEMAND'], label='Original Data', alpha=0.5)

plt.plot(df['Exp\_Smoothing'], label='Exponential Smoothing (α=0.2)', linewidth=2, color='green')

plt.title('Exponential Smoothing')

plt.legend()

plt.show()

1. Differencing to Remove Trend

df['Differenced'] = df['TOTALDEMAND'].diff()

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

plt.plot(df['Differenced'], label='Differenced Data', color='purple')

plt.title('Trend Elimination using Differencing')

plt.legend()

plt.show()

1. Display processed data head

df.head()

**RESULT:**

Thus the program for estimating and eliminating the trend in time series data- aggregation, smoothing has been implemented successfully and verified.