

## 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
```

2. Load the time series dataset.

```
df = pd.read_csv("/content/PRICE_AND_DEMAND_201801_NSW1.csv")
```

3. Convert SETTLEMENTDATE to datetime and set as index

```
f['SETTLEMENTDATE'] = pd.to_datetime(df['SETTLEMENTDATE'])
df.set_index('SETTLEMENTDATE', inplace=True)
```

4. Select only TOTALDEMAND column

```
df = df[['TOTALDEMAND']]
```

5. 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()
```

6. 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()
```

#### 7. 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()
```

#### 8. 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 ( $\alpha=0.2$ )',
         linewidth=2, color='green')
plt.title('Exponential Smoothing')
plt.legend()
plt.show()
```

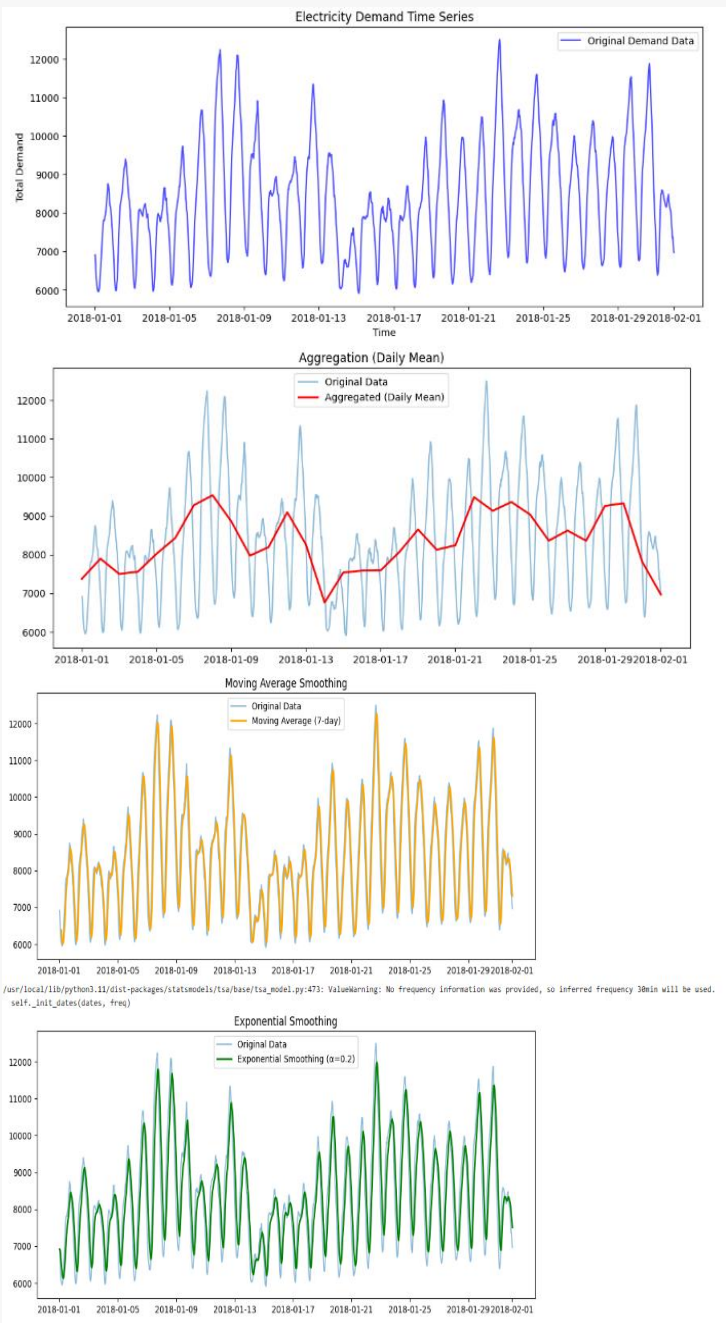
#### 9. Differencing to Remove Trend

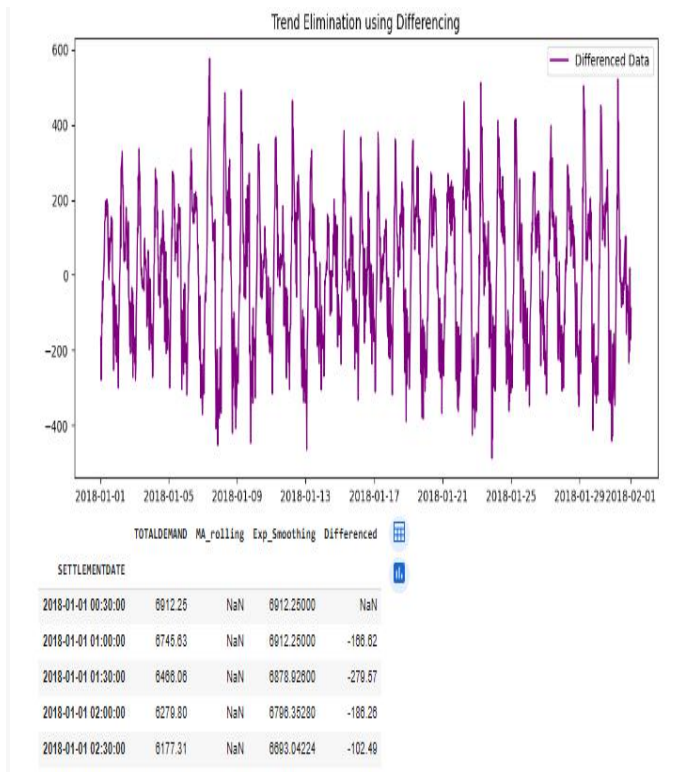
```
df['Differenced'] = df['TOTALDEMAND'].diff()
plt.figure(figsize=(12, 5))
plt.plot(df['Differenced'], label='Differenced Data',
         color='purple')
plt.title('Trend Elimination using Differencing')
plt.legend()
plt.show()
```

#### 10. Display processed data head

```
df.head()
```

OUTPUT:





## RESULT:

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