EXP No:7 Implement program for decomposing time series data into trend and seasonality

Aim:

To analyze and decompose electric production time series data into its trend, seasonal, and residual components using statistical methods.

Objectives:

- To load and preprocess electric production time series data.
- To apply seasonal decomposition using the additive model.
- To visualize the observed, trend, seasonal, and residual components.
- To understand the underlying patterns for forecasting and analysis.

Background:

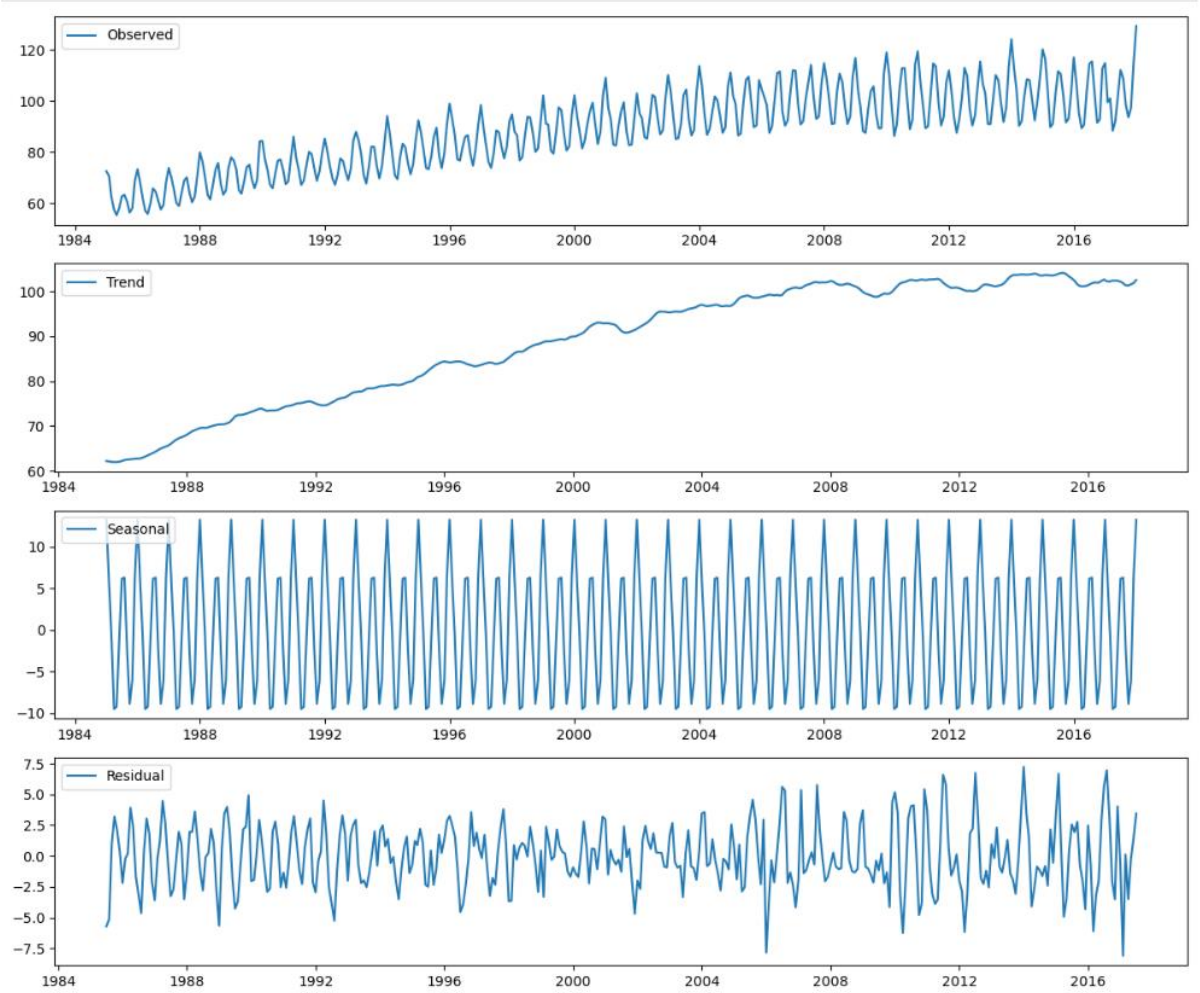
Time series decomposition is a key technique in time series analysis that separates data into three components: trend (long-term direction), seasonality (repeating patterns), and residuals (random noise). By breaking down the electric production data, we can better understand how energy usage fluctuates over time and identify useful patterns for forecasting and planning.

Code:

```
import pandas as pd

import matplotlib.pyplot as plt

from statsmodels.tsa.seasonal import seasonal_decompose

# Load dataset

df = pd.read_csv(r"C:\Users\Lenovo\Downloads\Electric_Production.csv")

# Convert 'DATE' column to datetime

df['DATE'] = pd.to_datetime(df['DATE'])



# Set 'DATE' as the index

df.set_index('DATE', inplace=True)
```

```
# Rename the value column for convenience
df.rename(columns={'IPG2211A2N': 'Electric_Production'}, inplace=True)


# Decompose the time series using additive model (suitable for linear seasonality)
# Set period=12 for monthly data (12 months in a year)
decomposition = seasonal_decompose(df['Electric_Production'], model='additive', period=12)
# Plotting the decomposed components
plt.figure(figsize=(12, 10))
plt.subplot(411)
plt.plot(decomposition.observed, label='Observed')
plt.legend(loc='upper left')
plt.subplot(412)
plt.plot(decomposition.trend, label='Trend')
plt.legend(loc='upper left')
plt.subplot(413)
plt.plot(decomposition.seasonal, label='Seasonal')
plt.legend(loc='upper left')
plt.subplot(414)
plt.plot(decomposition.resid, label='Residual')
plt.legend(loc='upper left')
plt.tight_layout()
plt.show()
```

Output:



Result:

Thus the program implement program for decomposing time series data into trend and seasonality.