# Implement Program for Decomposing Time Series Data into Trend and Seasonality

## Aim

To decompose a time series into its underlying components: trend, seasonality, and residuals. This aids in better understanding and modeling of the data.

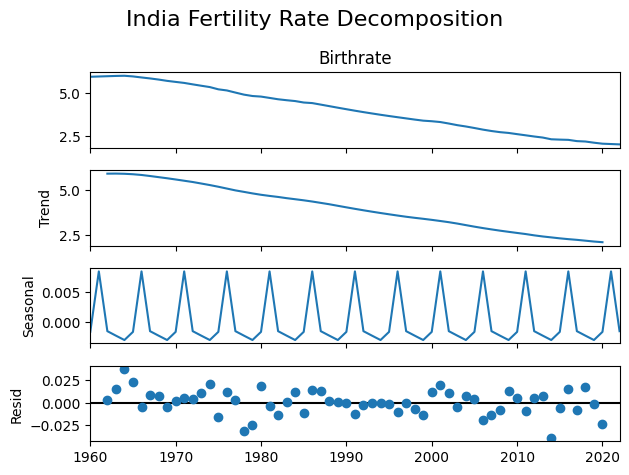
## Algorithm

1. Load the time series data.  
2. Visualize the raw time series data.  
3. Use a decomposition method (e.g., additive or multiplicative) to separate the data.  
4. Extract and visualize the trend component.  
5. Extract and visualize the seasonal component.  
6. Extract and visualize the residual component.

## Code

import pandas as pd  
import matplotlib.pyplot as plt  
from statsmodels.tsa.seasonal import seasonal\_decompose  
  
# Load data  
file\_path = r"C:\Users\22150\Downloads\Birthrate.csv"  
df = pd.read\_csv(file\_path, skiprows=4)  
  
# Filter for India  
country = 'India'  
country\_df = df[df['Country Name'] == country]  
  
# Extract year columns (columns that are digits)  
year\_columns = [col for col in country\_df.columns if col.isdigit()]  
ts = country\_df[year\_columns].T  
ts.columns = ['Birthrate']  
  
# Convert index to datetime  
ts.index = pd.to\_datetime(ts.index, format='%Y')  
  
# Ensure the 'Birthrate' column is numeric  
ts['Birthrate'] = pd.to\_numeric(ts['Birthrate'], errors='coerce')  
  
# Drop any rows with NaN values (this step is very important)  
ts = ts.dropna()  
  
# Print the first few rows before reindexing  
print("Before reindexing:")  
print(ts.head())  
  
# Reindex to make sure we have a continuous datetime index starting from '01-01'  
ts = ts.reindex(pd.date\_range(start='1960-01-01', end='2023-01-01', freq='YS'))  
  
# Print the first few rows after reindexing  
print("\nAfter reindexing:")  
print(ts.head())  
  
# Drop NaN values after reindexing  
ts = ts.dropna()  
  
# Double-check valid observations after cleaning  
print(f"Valid observations for {country} after cleaning: {len(ts)}")  
  
# Decompose the time series  
decomposition = seasonal\_decompose(ts['Birthrate'], model='additive', period=5)  
  
# Plot the decomposition  
decomposition.plot()  
plt.suptitle(f"{country} Fertility Rate Decomposition", fontsize=16)  
plt.tight\_layout()  
plt.show()

## Visualization



## Result

The decomposition revealed the underlying trend and seasonal patterns in the time series. This helps improve interpretability and is beneficial for further forecasting tasks.