# TIME SERIES DATA VISUALIZATION TECHNIQUES

## AIM:

To implement programs for visualizing time series data using various plotting techniques.

## PROCEDURE:

• Load the data - Read the dataset using Pandas.

• Clean the column names - Remove unwanted characters and spaces.

• Handle missing values - Convert non-numeric values to NaN and drop missing data.

• Reshape data for time series - Convert wide format data into long format.

• Convert data types - Ensure numerical columns are properly formatted.

• Visualize the data - Use different types of plots to analyze trends.

## CODE:

import pandas as pd  
import matplotlib.pyplot as plt  
  
# Load the dataset  
file\_path = r"C:\Users\admin\Downloads\API\_SP.DYN.TFRT.IN\_DS2\_EN\_csv\_v2\_162\API\_SP.DYN.TFRT.IN\_DS2\_EN\_csv\_v2\_162.csv"  
df = pd.read\_csv(file\_path, delimiter=',', skiprows=4, on\_bad\_lines='skip')  
  
# Clean column names  
df.columns = df.columns.str.replace('"', '').str.strip()  
  
# Reshape data  
df\_long = pd.melt(df[['Country Name'] + [str(year) for year in range(1960, 2024)]],  
 id\_vars=["Country Name"], var\_name="Year", value\_name="Birth Rate")  
df\_long['Year'] = pd.to\_numeric(df\_long['Year'])  
df\_long['Birth Rate'] = pd.to\_numeric(df\_long['Birth Rate'], errors='coerce')  
df\_long.dropna(subset=['Birth Rate'], inplace=True)  
  
df\_country = df\_long[df\_long['Country Name'] == 'India']  
  
# Line Plot  
plt.figure(figsize=(10, 6))  
plt.plot(df\_country['Year'], df\_country['Birth Rate'], marker='o', color='b')  
plt.title('Birth Rate Over Time for India (Line Plot)')  
plt.xlabel('Year')  
plt.ylabel('Birth Rate (per 1000 people)')  
plt.grid(True)  
plt.xticks(rotation=45)  
plt.tight\_layout()  
plt.show()  
  
# Bar Plot  
plt.figure(figsize=(10, 6))  
plt.bar(df\_country['Year'], df\_country['Birth Rate'], color='orange')  
plt.title('Birth Rate Over Time for India (Bar Plot)')  
plt.xlabel('Year')  
plt.ylabel('Birth Rate (per 1000 people)')  
plt.xticks(rotation=45)  
plt.tight\_layout()  
plt.show()  
  
# Box Plot  
plt.figure(figsize=(10, 6))  
plt.boxplot(df\_country['Birth Rate'], vert=False)  
plt.title('Birth Rate Distribution for India (Box Plot)')  
plt.xlabel('Birth Rate (per 1000 people)')  
plt.tight\_layout()  
plt.show()  
  
# Histogram  
plt.figure(figsize=(10, 6))  
plt.hist(df\_country['Birth Rate'], bins=20, color='purple', edgecolor='black')  
plt.title('Distribution of Birth Rate for India (Histogram)')  
plt.xlabel('Birth Rate (per 1000 people)')  
plt.ylabel('Frequency')  
plt.tight\_layout()  
plt.show()  
  
# Rolling Average Plot  
df\_country.loc[:, 'Rolling Average'] = df\_country['Birth Rate'].rolling(window=5).mean()  
plt.figure(figsize=(10, 6))  
plt.plot(df\_country['Year'], df\_country['Rolling Average'], marker='x', color='green', label='Rolling Average (5 years)')  
plt.plot(df\_country['Year'], df\_country['Birth Rate'], marker='o', color='b', alpha=0.5, label='Original Data')  
plt.title('Rolling Average Birth Rate for India (Rolling Plot)')  
plt.xlabel('Year')  
plt.ylabel('Birth Rate (per 1000 people)')  
plt.legend()  
plt.grid(True)  
plt.xticks(rotation=45)  
plt.tight\_layout()  
plt.show()

## RESULT:

The program successfully visualizes time series data using multiple plotting techniques, including line plots, bar plots, box plots, histograms, and rolling average plots.