EX.N0 : 2	**Implement programs for visualization time series data.**
DATE : 01/02/2025	

AIM:

Implement programs for visualizing time series data.

PROGRAM:

IMPORT LIBRARIES

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

LOAD DATASET

```
def load_data(filepath):
    try:
        data = pd.read_csv(filepath, parse_dates=True, index_col='Date')
        print("Dataset loaded successfully.")
        return data
    except Exception as e:
        print(f"Error loading dataset: {e}")
        return None
def clean_data(data):
    print("Initial dataset shape:", data.shape)
    data = data.drop_duplicates()
    data = data.fillna(method='ffill')
    data = data.fillna(method='bfill')
    data = data.dropna()
print("Dataset shape after cleaning:", data.shape)
    return data
def preprocess_time_series(data):
    print("Index type:", type(data.index))
    if not isinstance(data.index, pd.DatetimeIndex):
        data.index = pd.to_datetime(data.index)
    data = data.sort_index()
    return data
```

FEATURE ENGINEERING

```python
def feature_engineering(data):
    data['SMA_7'] = data['Close'].rolling(window=7).mean()
    data['SMA_30'] = data['Close'].rolling(window=30).mean()Average
data['Lag_1'] = data['Close'].shift(1)
data['Lag_7'] = data['Close'].shift(7)
    data = data.dropna()
    return data
```

DATA VISUALIZATION

```python
def visualize_data(data):
    plt.figure(figsize=(12, 6))
    plt.plot(data['Close'], label='Gold Price')
    plt.plot(data['SMA_7'], label='7-Day SMA', linestyle='--')
    plt.plot(data['SMA_30'], label='30-Day SMA', linestyle='--')
    plt.title('Averages')
    plt.xlabel('Date')
    plt.ylabel('Price')
    plt.legend()
    plt.grid()
    plt.show()
def visualize_time_series(data):
    plt.figure(figsize=(14, 7))
    plt.subplot(2, 1, 1)
    plt.plot(data['Close'], label='Gold Price', color='blue')
    plt.title('Gold Price Over Time')
    plt.xlabel('Date')
    plt.ylabel('Price')
    plt.legend()
    plt.grid()
    plt.subplot(2, 1, 2)
    plt.hist(data['Close'], bins=30, color='gold', edgecolor='black')
    plt.title()
    plt.xlabel()
    plt.ylabel()
    plt.tight_layout()
    plt.show()
```
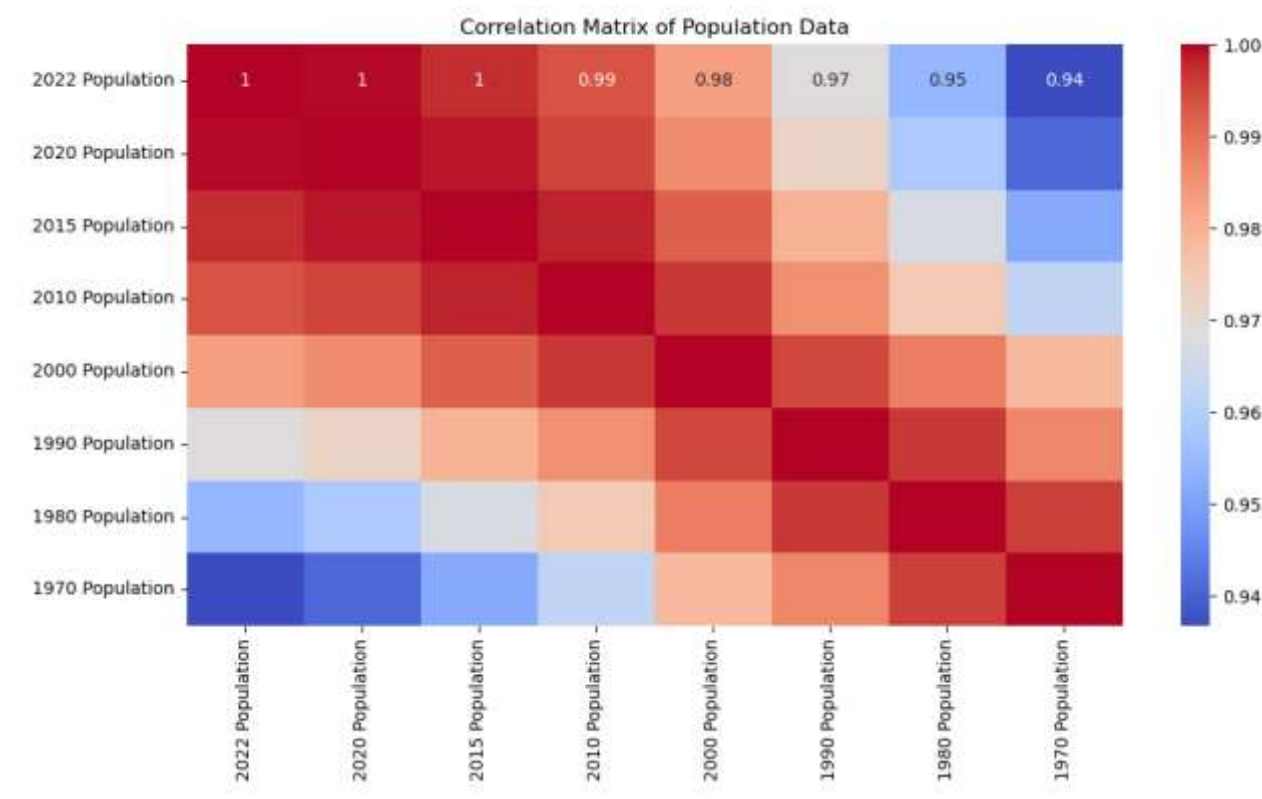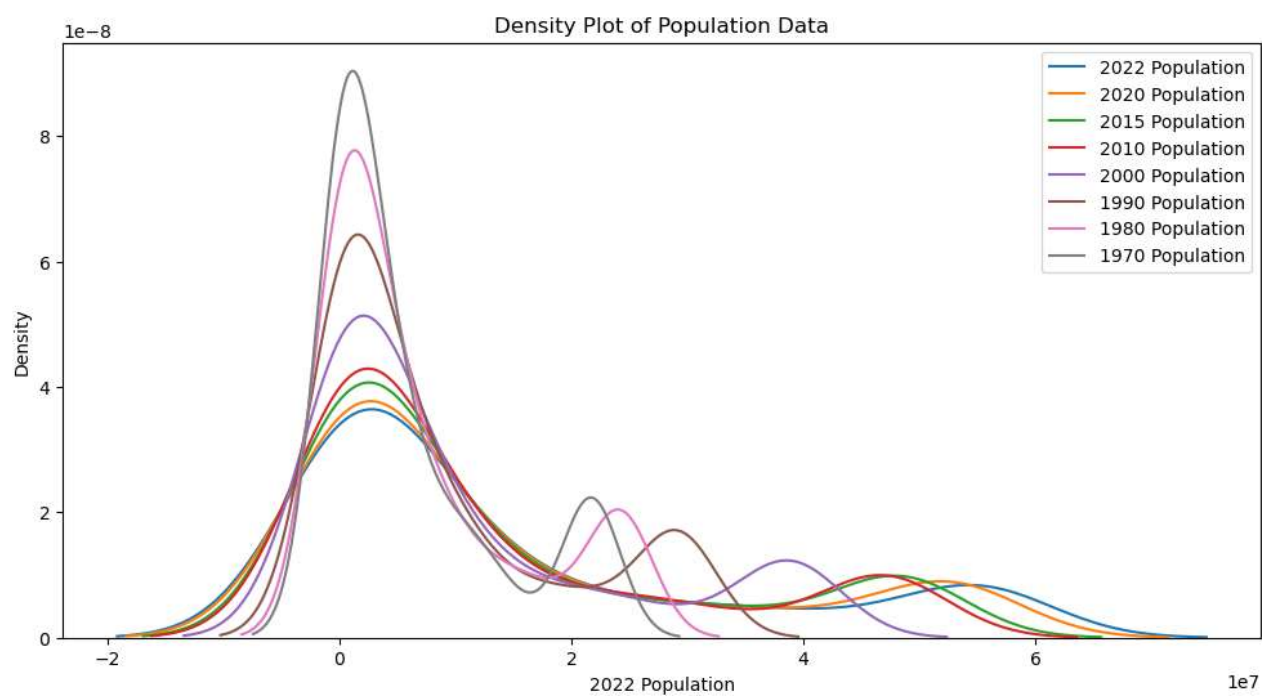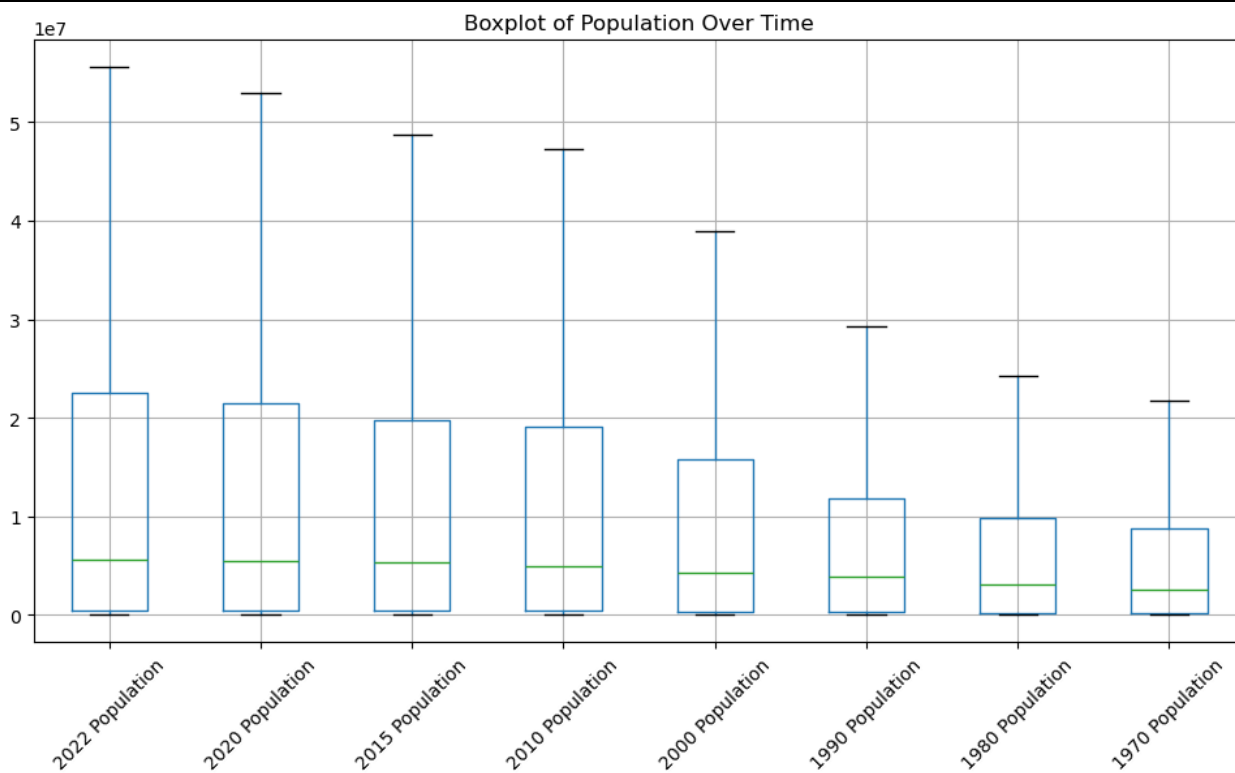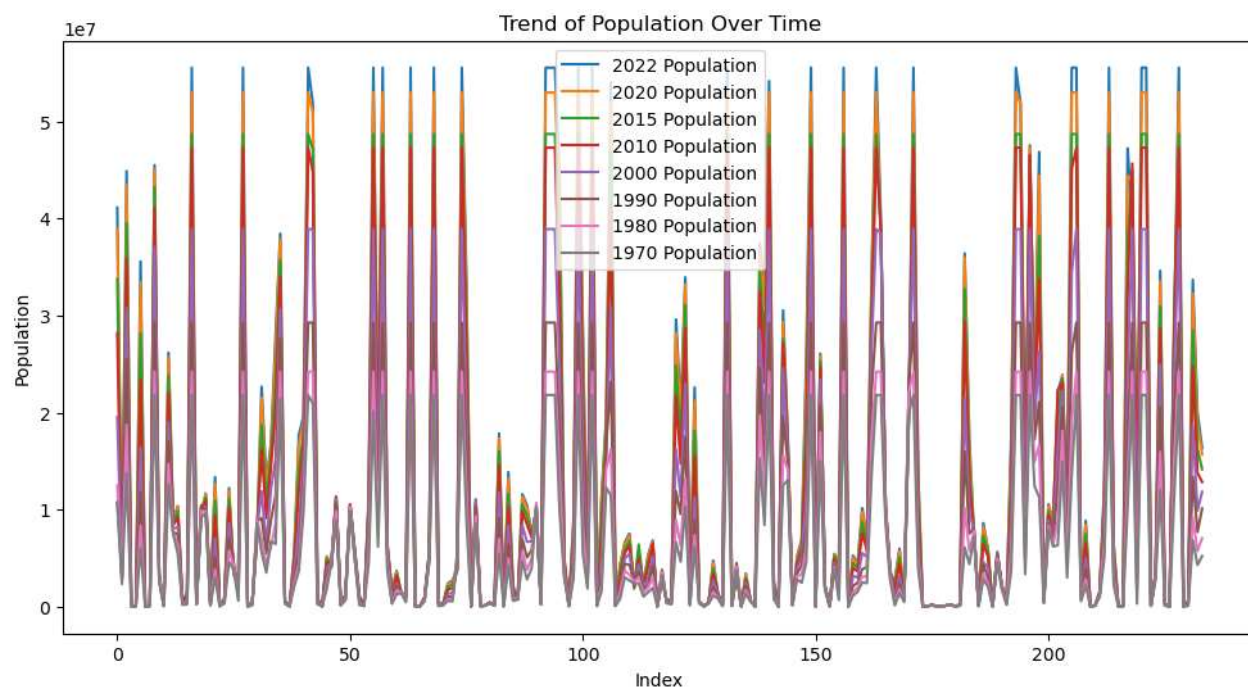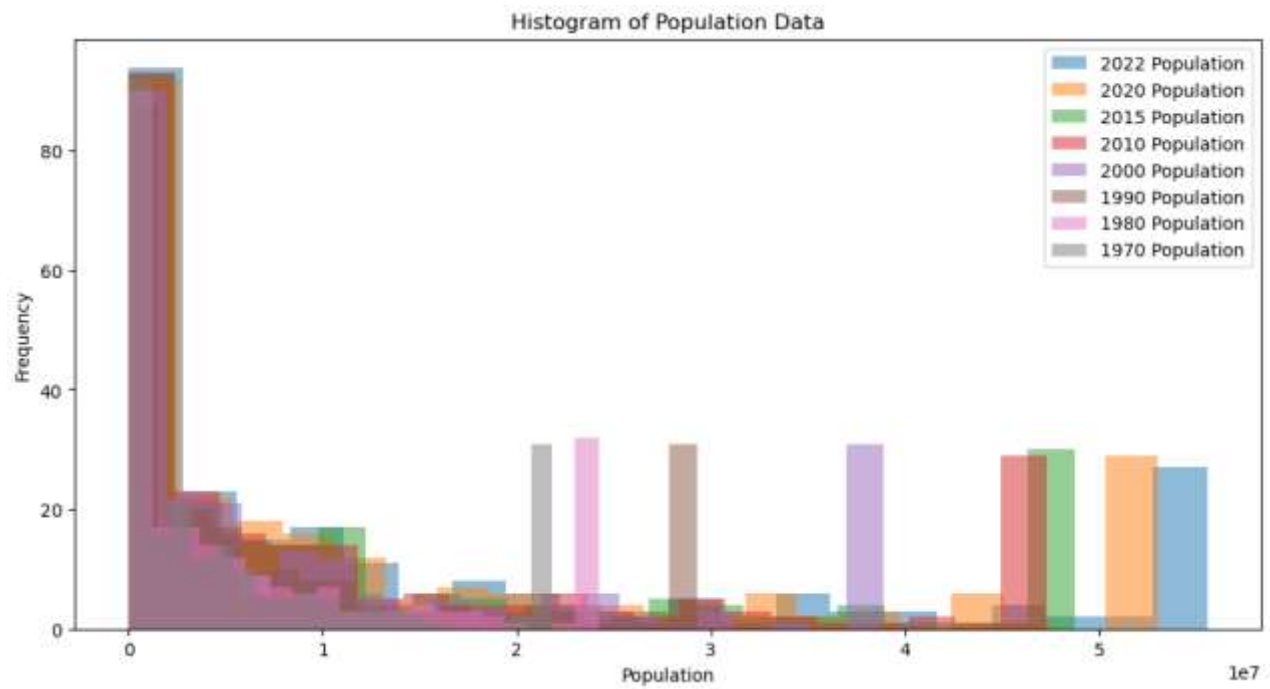
MAIN FUNCTION

```
def main():
    filepath = "C:\\Users\\Downloads\\archive (1) (1)\\World_population.csv"
    data = load_data(filepath)
    if data is None:
        return
    data = clean_data(data)
    data = preprocess_time_series(data)
data = feature_engineering(data)
    visualize_data(data)
    visualize_time_series(data)
    print("Processed dataset preview:\n", data.head())
if   name   == "  main  ":
    main()
```

Density Plot of Population Data

Correlation Matrix of Population Data

Trend of Population Over Time

Boxplot of Population Over Time

Histogram of Population Data

RESULT:

Thus, the program for Implement programs for visualizing time series data is executed successfully.

\