2. Implement programs for visualizing time series data.

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

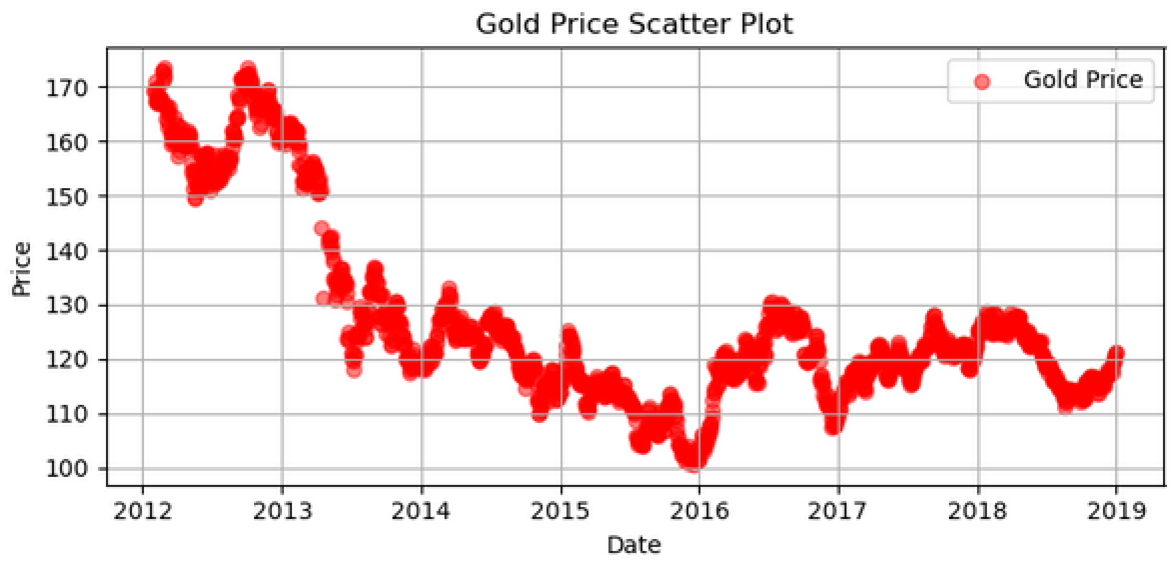
AIM:

Implement programs for visualizing time series data.

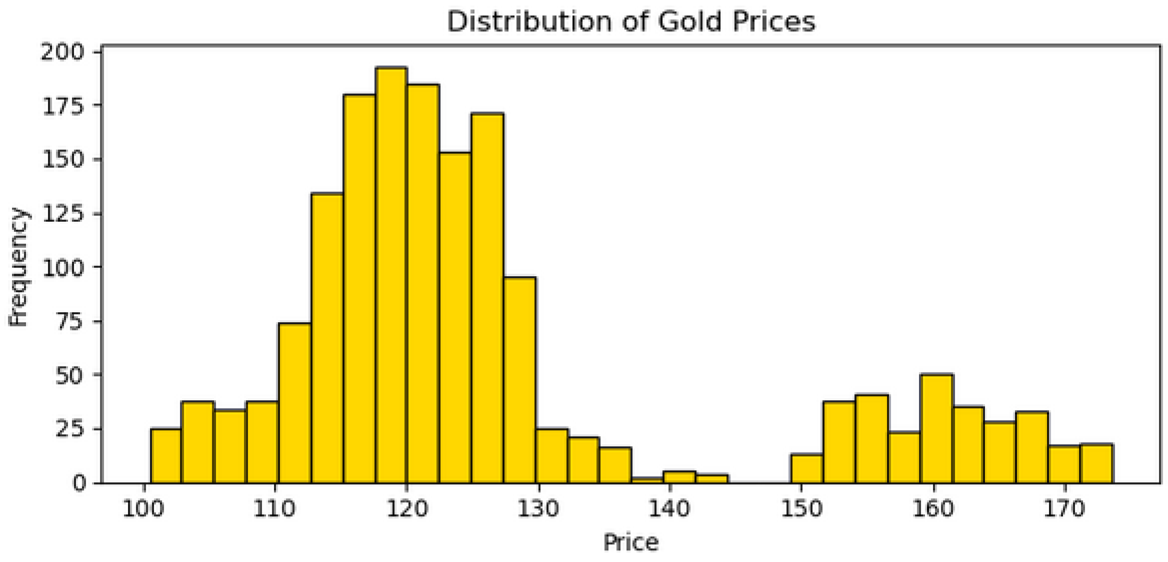
PROGRAM:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
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')  # Forward fill missing values
    data = data.fillna(method='bfill')  # Backward fill for remaining missing values
  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
def feature_engineering(data):
    data['SMA_7'] = data['Close'].rolling(window=7).mean()  # 7-day Simple Moving Average
    data['SMA_30'] = data['Close'].rolling(window=30).mean()  # 30-day Simple Moving Average
    data['Lag_1'] = data['Close'].shift(1)  # Previous day's price
    data['Lag_7'] = data['Close'].shift(7)  # Price a week ago
    data = data.dropna()
    return data
def visualize_data(data):

  plt.figure(figsize=(14, 10)
    plt.subplot(3, 2, 1)
    plt.scatter(data.index, data['Close'], label='Gold Price', color='red', alpha=0.5)
    plt.title('Gold Price Scatter Plot')
    plt.xlabel('Date')
    plt.ylabel('Price')
    plt.legend()
    plt.grid()

    plt.subplot(3, 2, 2)
    plt.hist(data['Close'], bins=30, color='gold', edgecolor='black')
    plt.title('Distribution of Gold Prices')
    plt.xlabel('Price')
    plt.ylabel('Frequency')


    plt.subplot(3, 2, 3)
    plt.boxplot(data['Close'], vert=False)
    plt.title('Box Plot of Gold Prices')

    plt.subplot(3, 2, 4)
    plt.bar(data.index, data['SMA_7'], label='7-Day SMA', color='green', alpha=0.5)
    plt.bar(data.index, data['SMA_30'], label='30-Day SMA', color='purple', alpha=0.5)
    plt.title('Moving Averages')
    plt.xlabel('Date')
    plt.ylabel('Price')
    plt.legend()
    plt.grid()

    plt.subplot(3, 2, 5)
    plt.violinplot(data['Close'])
    plt.title('Violin Plot of Gold Prices')

    plt.tight_layout()
    plt.show()

def main():
    filepath = "C:\\Users\\HDC0422251\\Downloads\\archive (1)\\FINAL_USO.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)

    print("Processed dataset preview:\n", data.head())

if __name__ == "__main__":
    main()
-
```

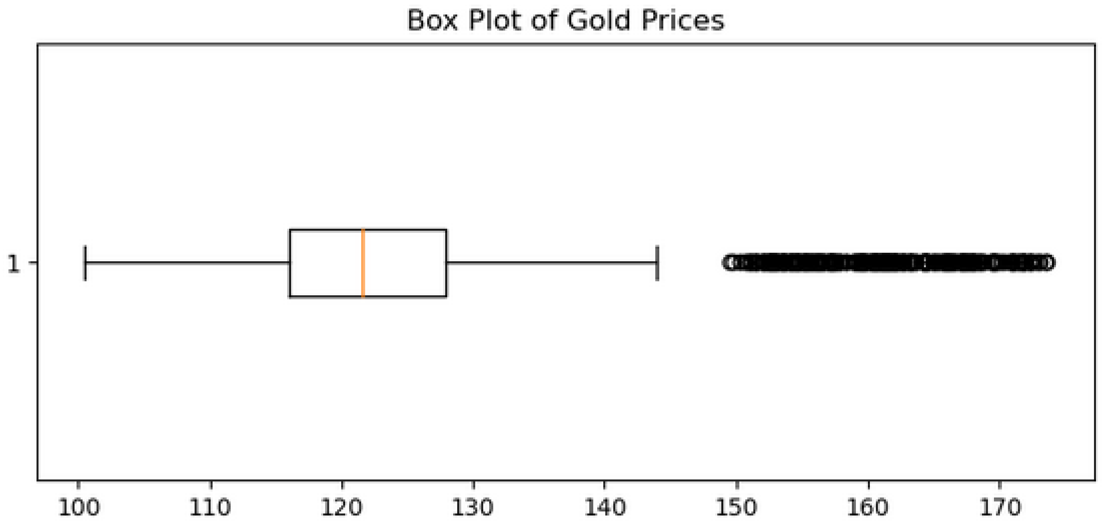
OUTPUT:



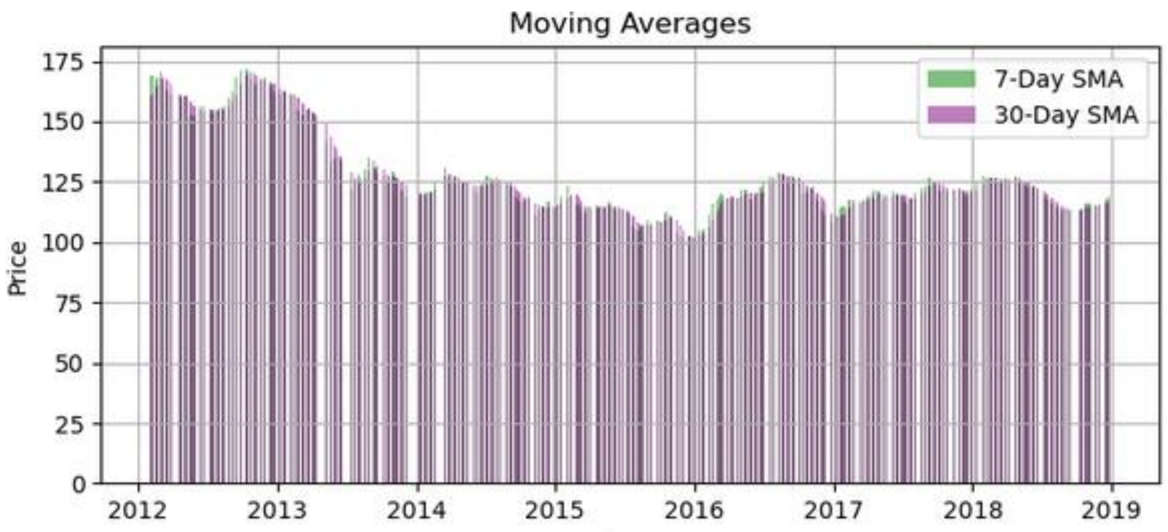
Gold Price Scatter Plot
Gold Price
Price
Date
170
160
150
140
130
120
110
100
2012
2013
2014
2015
2016
2017
2018
2019

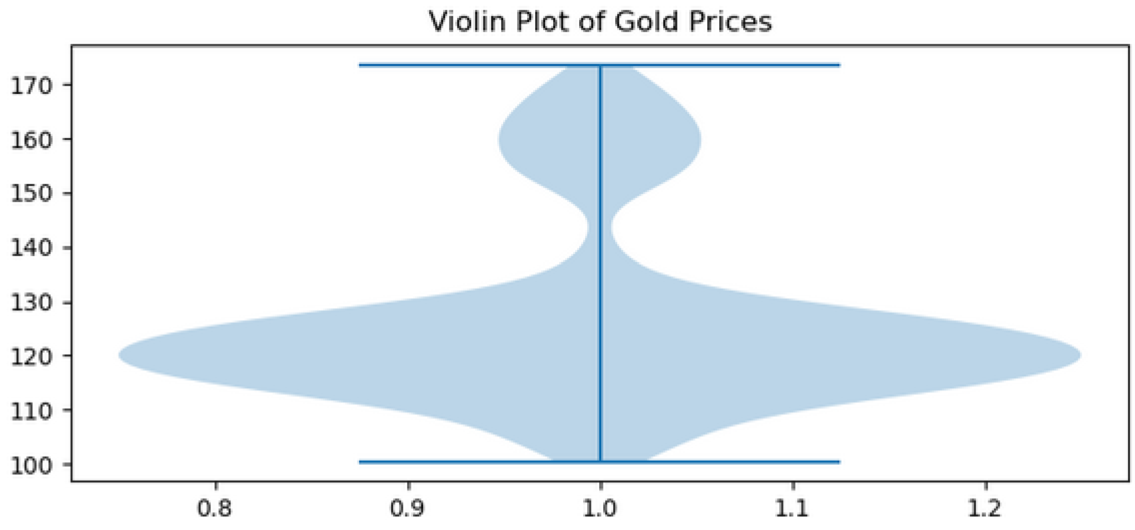


Distribution of Gold Prices
Frequency
Price
200
175
150
125
100
75
50
25
0
100
110
120
130
140
150
160
170



Box Plot of Gold Prices
1
100
110
120
130
140
150
160
170





RESULT:

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