EX:No.2 DATE: 25/01/25	Implement programs for visualizing time series data.

AIM:

Write a program to implement time series data for import library, load data, Preprocessing and visualising.

ALGORITHM:

Step 1: Install required libraries (if not already installed).

Step 2: Import necessary libraries (pandas, numpy, matplotlib).

Step 3: Load oil prediction data, parse dates, and set 'date' as the index.

Step 4: Remove duplicate timestamps and fill missing values.

Step 5: Select the 'price_today' column.

Step 6: Remove outliers using the IQR method.

Step 7: Ensure daily data frequency.

Step 8: Resample to weekly average (optional, not used in the plot).

Step 9: Create a figure and plot daily price levels as a line graph.

Step 10: Set labels, title, and legend for the plot.

Step 11: Show the plot.

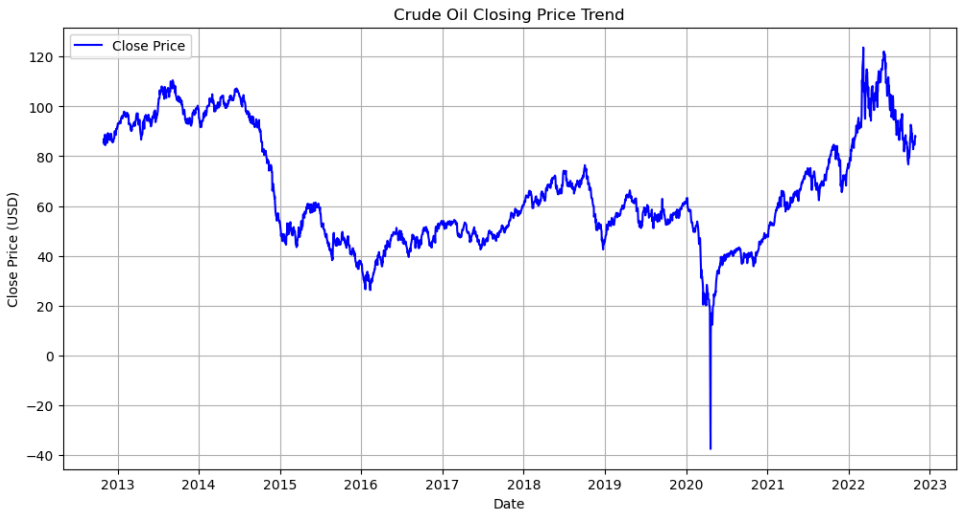
CODE:

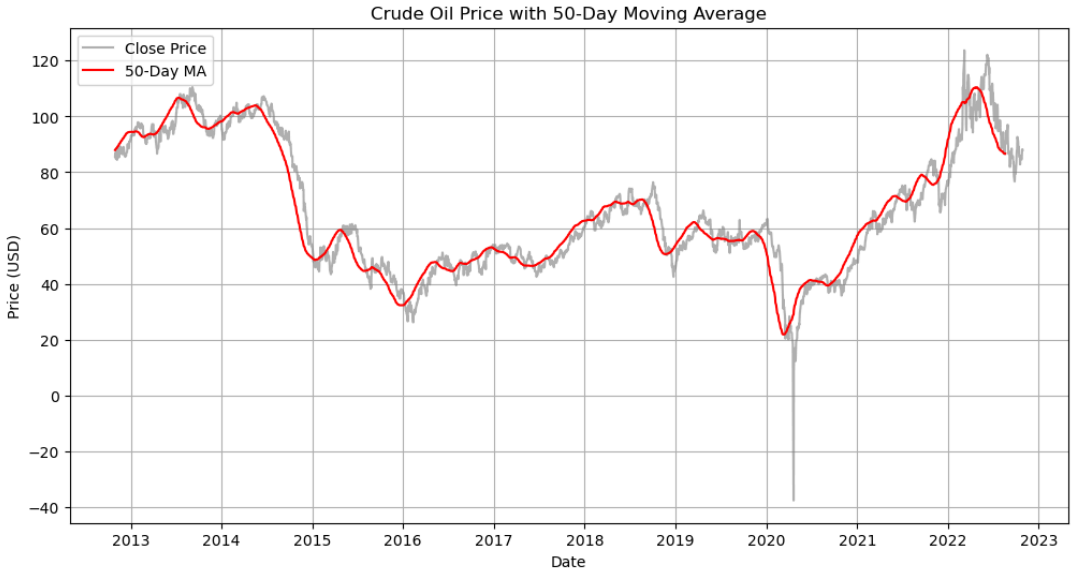
```
import os

file_path = r"D:\Abdul\ex-2\Crude oil.csv"

if os.path.exists(file_path):
    print("File exists!")
else:
    print("File not found. Check the path.")
import pandas as pd

file_path = r"D:\Abdul\ex-2\Crude oil.csv"

# Read CSV without parsing dates
```

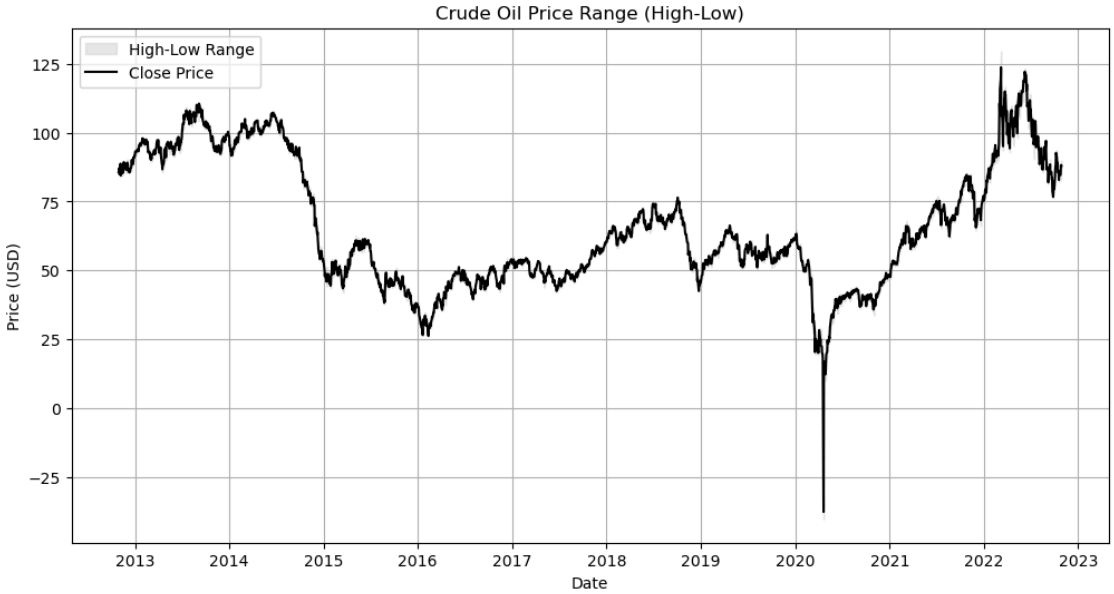
```
df = pd.read_csv(file_path)

# Print the first few rows and column names
print(df.head())
print("\nColumn names:", df.columns.tolist())
import pandas as pd

file_path = r"D:\Abdul\ex-2\Crude oil.csv"

# Read CSV with correct date parsing
df = pd.read_csv(file_path, parse_dates=['Date'])

# Set 'Date' as the index
df.set_index('Date', inplace=True)

print(df.head())  # Verify output

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load dataset
file_path = r"D:\Abdul\ex-2\Crude oil.csv"
df = pd.read_csv(file_path, parse_dates=['Date'])
df.set_index('Date', inplace=True)

# Convert 'Close/Last' to numeric
df.rename(columns={'Close/Last': 'Close'}, inplace=True)
df['Close'] = pd.to_numeric(df['Close'], errors='coerce')

# 1️⃣ Line Chart - Price Trend Over Time
plt.figure(figsize=(12, 6))
plt.plot(df.index, df['Close'], label='Close Price', color='blue')
plt.xlabel("Date")
plt.ylabel("Close Price (USD)")
plt.title("Crude Oil Closing Price Trend")
plt.legend()
plt.grid()
plt.show()

# 2️⃣ Moving Average (50-day)
df['50_MA'] = df['Close'].rolling(window=50).mean()

plt.figure(figsize=(12, 6))
plt.plot(df.index, df['Close'], label='Close Price', color='gray', alpha=0.6)
plt.plot(df.index, df['50_MA'], label='50-Day MA', color='red')
plt.xlabel("Date")
plt.ylabel("Price (USD)")
```

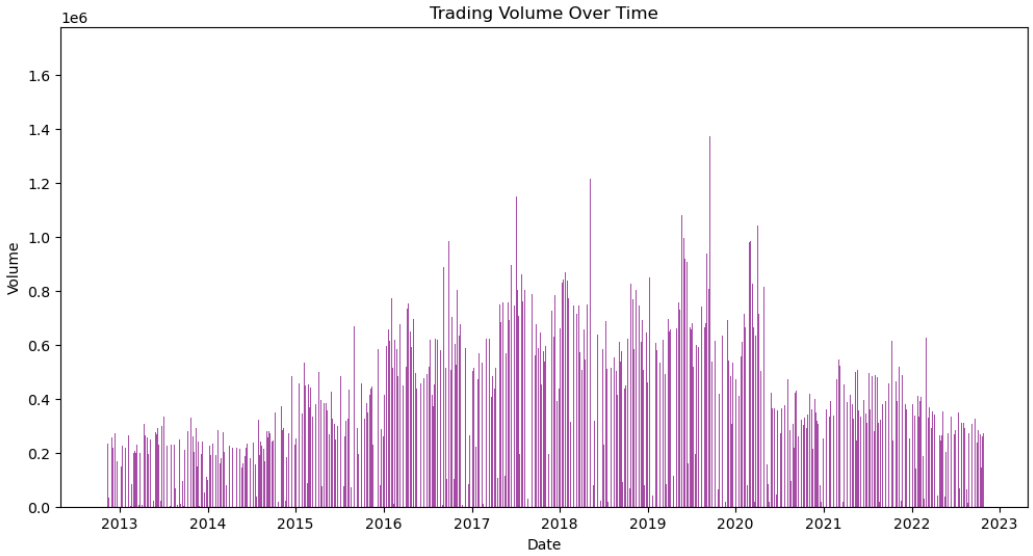
```
plt.title("Crude Oil Price with 50-Day Moving Average")
plt.legend()
plt.grid()
plt.show()

# 3️⃣ Bar Chart - Trading Volume
plt.figure(figsize=(12, 6))
plt.bar(df.index, df['Volume'], color='purple', alpha=0.7)
plt.xlabel("Date")
plt.ylabel("Volume")
plt.title("Trading Volume Over Time")
plt.show()

# 4️⃣ High-Low Range (Shaded Area)
plt.figure(figsize=(12, 6))
plt.fill_between(df.index, df['Low'], df['High'], color='lightgray', alpha=0.5, label='High-Low Range')
plt.plot(df.index, df['Close'], label='Close Price', color='black')
plt.xlabel("Date")
plt.ylabel("Price (USD)")
plt.title("Crude Oil Price Range (High-Low)")
plt.legend()
plt.grid()
plt.show()

# 5️⃣ Box Plot - Price Distribution
plt.figure(figsize=(8, 6))
sns.boxplot(y=df['Close'], color='cyan')
plt.ylabel("Price (USD)")
plt.title("Crude Oil Price Distribution")
plt.grid()
plt.show()
```

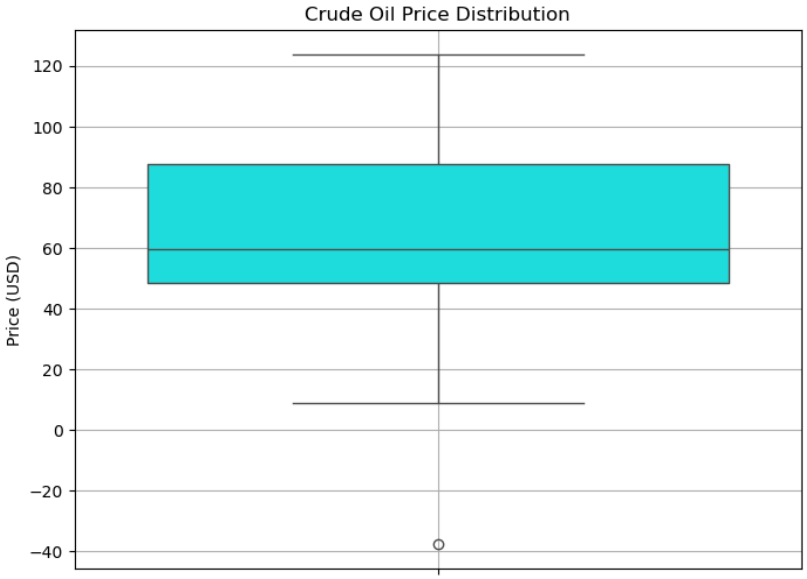
OUTPUT:




Crude Oil Price with 50-Day Moving Average
Close Price
50-Day MA
Price (USD)
Date
120
100
80
60
40
20
0
−20
−40
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023


Crude Oil Price Range (High-Low)
High-Low Range
Close Price
Price (USD)
Date
125
100
75
50
25
0
−25
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023





RESULT:

Thus, the program using the time series data implementation has been done successfully.