EX:No.2 221501039

25/01/25

Program to analyze and visualize stock trends using time series plots, moving averages, volume analysis, and daily returns.

Aim:

Write a program to analyze and visualize stock trends using time series plots, moving averages, volume analysis, and daily returns.

Algorithm:

Step 1: Import Required Libraries

- Import `pandas` for data handling, `numpy` for numerical computations, `matplotlib.pyplot` and `seaborn` for data visualization.

Step 2: Upload and Load the Dataset

- Load the dataset using `pd.read_csv()`.

Step 3: Data Preprocessing

Step 4: Set Up Visualization Styling

- Convert the date column into a `datetime` object using `pd.to_datetime()`.
- Sort the dataset by date.
- Set the date column as the index for time series analysis.
- Compute daily returns of house prices using `pct_change()`..

Step 5: Generate Visualizations

Histogram of House Prices – To understand price distribution.

Line Plot of House Prices Over Time – To analyze price trends.

Scatter Plot of Daily Returns vs Price – To study price volatility.

Box Plot of House Prices – To observe spread and detect outliers.

Correlation Heatmap – To evaluate relationships between numerical variables.

Step 6: Display Results and Interpret Insights

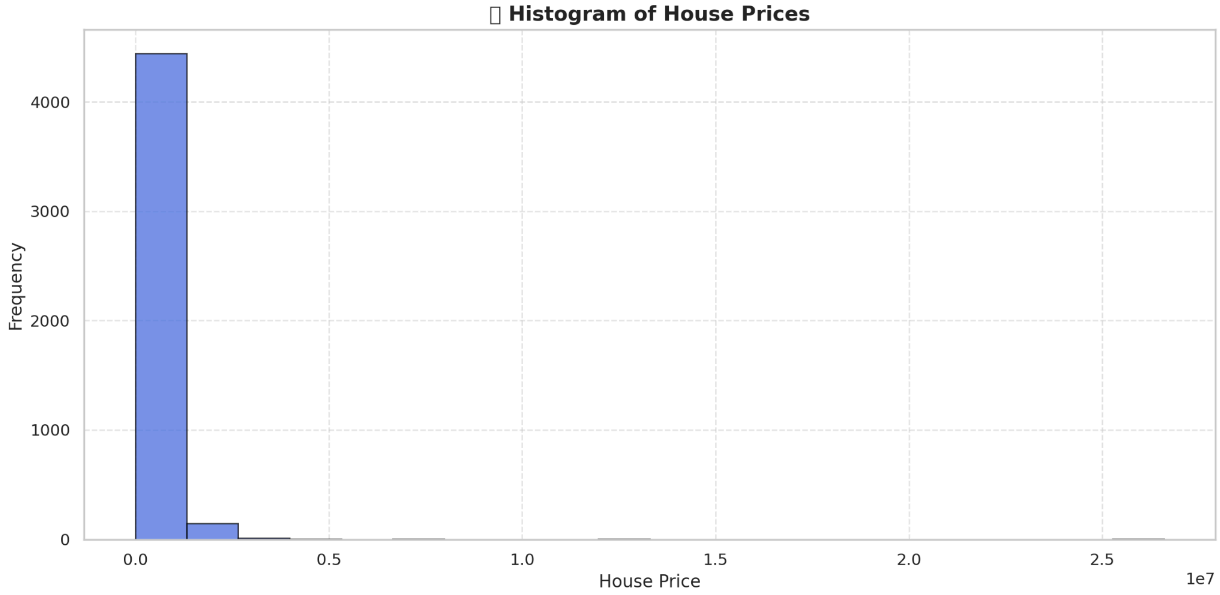
- Analyze the patterns, trends, and relationships among the features.

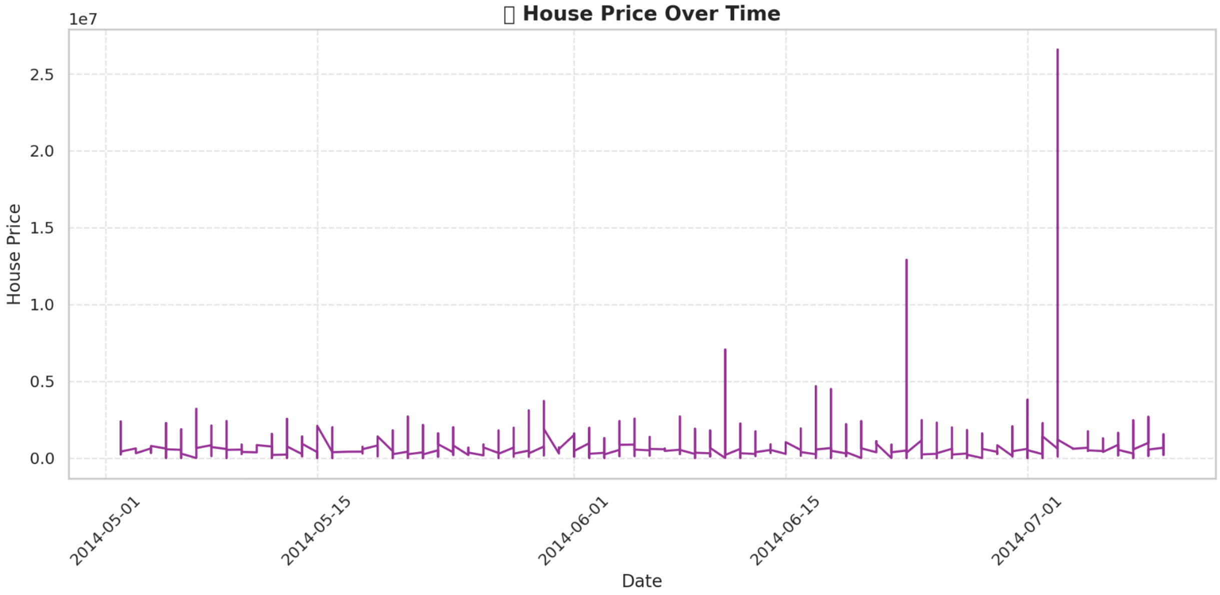
Code:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
# Load Dataset
df = pd.read_csv('/mnt/data/data.csv')
# Preprocessing
df['date'] = pd.to_datetime(df['date'])
df = df.sort_values('date')
df.set_index('date', inplace=True)
df['price'] = df['price'].interpolate(method='linear')
```

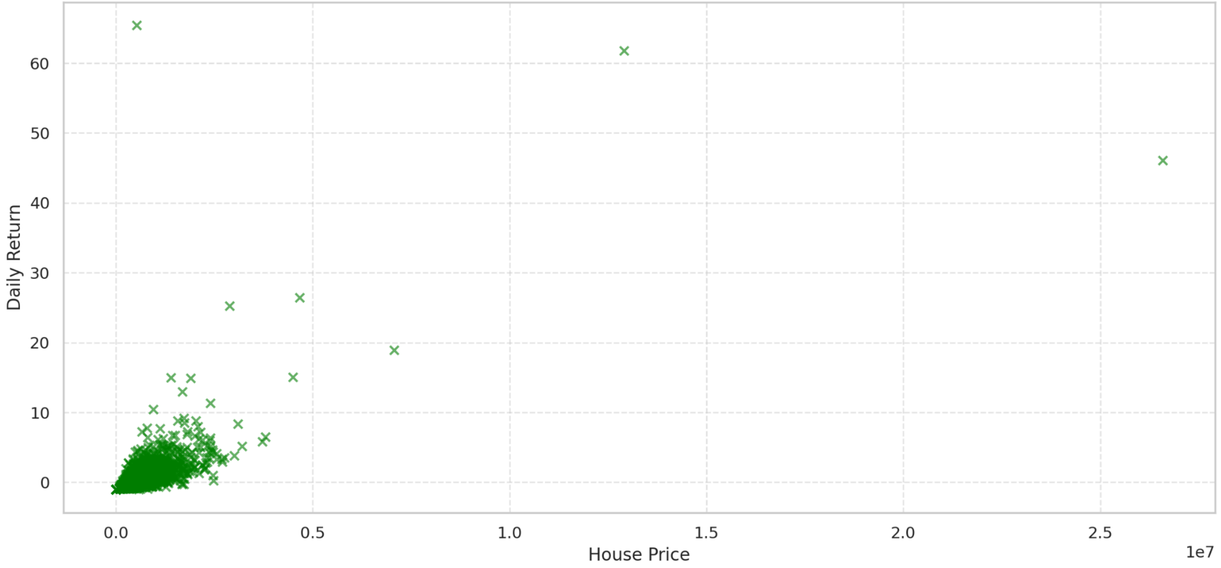
```
df['daily_return'] = df['price'].pct_change()
# Visualization Styling
sns.set_style("whitegrid")
# 1. Histogram of House Prices
plt.figure(figsize=(12, 6))
plt.hist(df['price'], bins=20, color="royalblue", alpha=0.7, edgecolor="black")
plt.xlabel("House Price", fontsize=12)
plt.ylabel("Frequency", fontsize=12)
plt.title("📊 Histogram of House Prices", fontsize=14, fontweight="bold")
plt.grid(True, linestyle="--", alpha=0.5)
plt.show()

# 2. Line Plot of House Prices Over Time
plt.figure(figsize=(12, 6))
plt.plot(df.index, df['price'], alpha=0.8, color="purple")
plt.xlabel("Date", fontsize=12)
plt.ylabel("House Price", fontsize=12)
plt.title("📈 House Price Over Time", fontsize=14, fontweight="bold")
plt.xticks(rotation=45)
plt.grid(True, linestyle="--", alpha=0.5)
plt.show()

# 3. Scatter Plot of Daily Returns vs House Price
plt.figure(figsize=(12, 6))
plt.scatter(df['price'], df['daily_return'], alpha=0.6, color="green")
plt.xlabel("House Price", fontsize=12)
plt.ylabel("Daily Return", fontsize=12)
plt.title("🔁 Scatter Plot of Daily Returns vs House Price", fontsize=14, fontweight="bold")
plt.grid(True, linestyle="--", alpha=0.5)
plt.show()

# 4. Box Plot of House Prices
plt.figure(figsize=(8, 6))
sns.boxplot(y=df['price'], color="lightblue")
plt.ylabel("House Price", fontsize=12)
plt.title("📦 Box Plot of House Prices", fontsize=14, fontweight="bold")
plt.grid(True, linestyle="--", alpha=0.5)
plt.show()

# 5. Correlation Heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap="coolwarm", fmt=".2f", linewidths=0.5)
plt.title("🔥 Correlation Heatmap", fontsize=14, fontweight="bold")
plt.show()
```

Output:

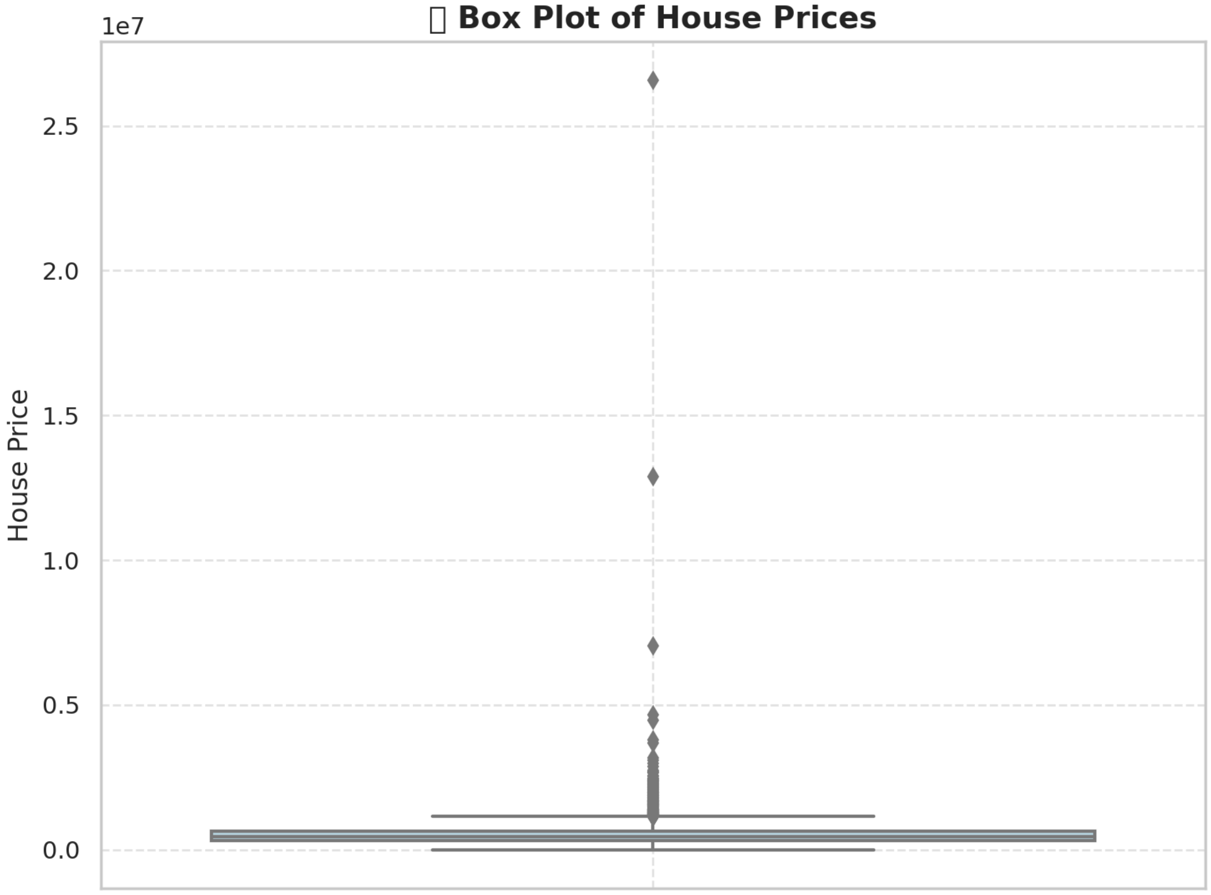

□ Histogram of House Prices
Frequency
0
1000
2000
3000
4000
0.0
0.5
1.0
1.5
2.0
2.5
1e7
House Price

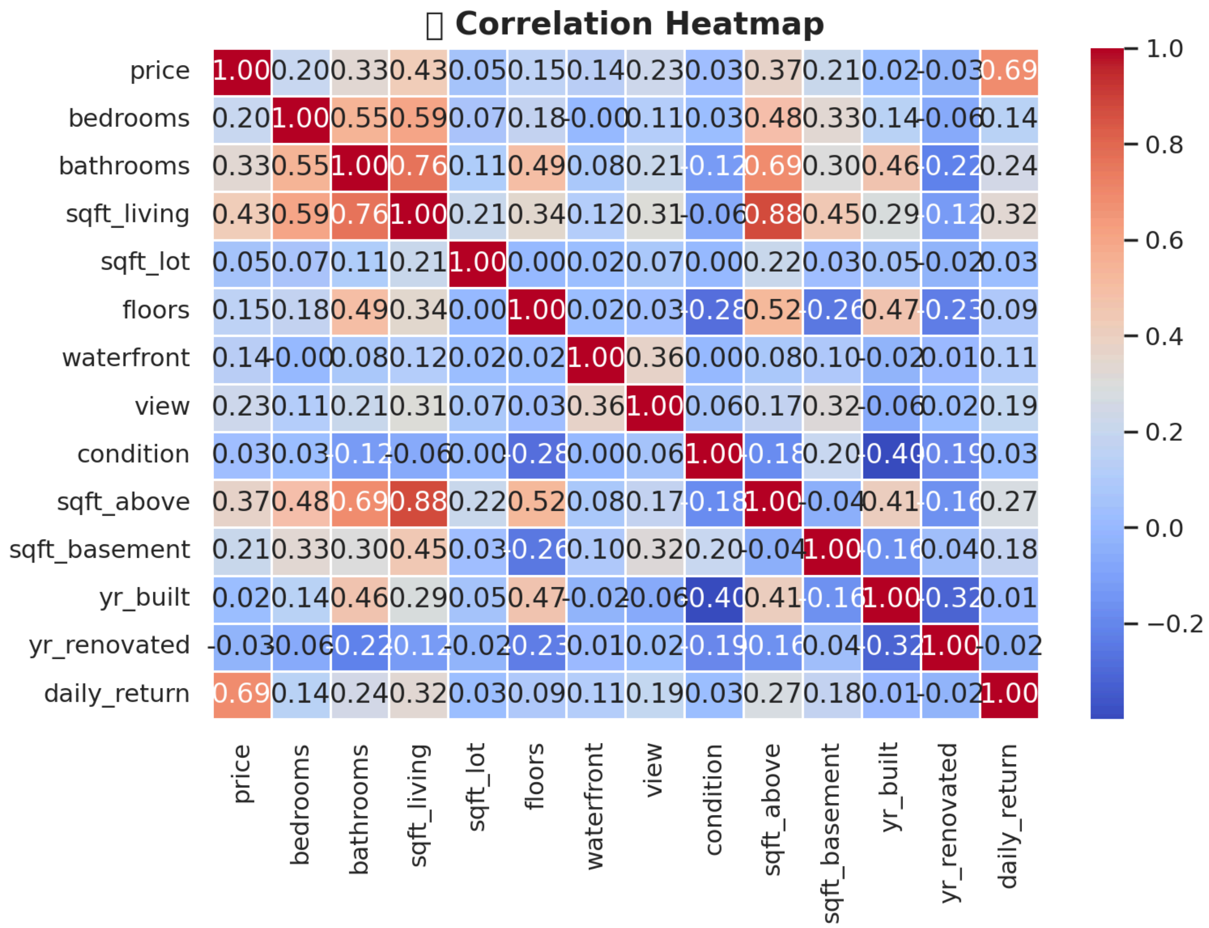

□ House Price Over Time
1e7
House Price
0.0
0.5
1.0
1.5
2.0
2.5
2014-05-01
2014-05-15
2014-06-01
2014-06-15
2014-07-01
Date

□ Scatter Plot of Daily Returns vs House Price



□ Box Plot of House Prices





Result:

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