amazon STOCK PRICE (ALL-TIME)

Data Analysis Project

Using

python™

I have done this project to understand what is DATA SCIENCE AND how we use python for data analysis

OVERVIEW

This project involves analyzing the historical stock price data of Amazon (AMZN) to uncover trends, patterns, and insights. Using various Python libraries, we performed a comprehensive analysis of the stock prices from the company's initial public offering (IPO) in 1997 onwards.

OBJECTIVES

1. **Identify trends and patterns in Amazon's stock prices over time.**
2. **Understand the volatility and performance of the stock.**
3. **Analyze the distribution of trading volumes and price changes.**
4. **Detect seasonal trends and long-term growth.**
5. **Visualize the data using different types of charts and graphs.**
6. **Understanding of data analysis using Python**

DATASET

Use Dataset of Amazon Stock Price (All-Time) from Kaggle

https://www.kaggle.com/datasets/kannan1314/amazon-stock-price-all-time

Detail of Dataset

- The dataset consists of recodes 6155
- It consists of 7 Attributes named Date, Open, High, Low, Close Adj, and Close Volume)

STEPS OFDATA ANALYSIS

- Uploading the CVS file using Python Panda
- Understand the dataset attributes by using python
- Data Cleaning
    1. Delete Null values
    2. Delete blank fields
- Exploratory Data Analysis

QUESTIONS

1. What is Amazon's closing price trend over the given period?
2. How does the opening price compare to the closing price over time?
3. What is the distribution of daily trading volumes?
4. What are the highest and lowest closing prices in the dataset?
5. What is the average monthly closing price?
6. How does the stock's volatility change over time?
7. What are the significant trends in the adjusted closing prices?
8. Which day had the highest trading volume and what was the price trend on that day?
9. What is the correlation between trading volume and closing price?
10. How do the daily returns (percentage change in closing price) vary over time?

QUESTIONS

11. What is the moving average of the closing prices?
12. How did Amazon's stock perform every week?
13. Which days experienced the highest single-day price increases and decreases?
14. What is the trend of the opening prices over time?
15. How do the high and low prices compare over the dataset period?
16. What is the rolling 7-day average of the closing prices?
17. How does the stock price react after large trading volume days?
18. What is the cumulative return of Amazon's stock over the dataset period?
19. How do the monthly high and low prices compare over the dataset period?
20. What are the seasonal trends in Amazon's stock prices?

PYHTON LIBRARIES USED

1. Pandas

- Purpose: Data manipulation and analysis
- Usage: Loading datasets, handling data frames, performing time series analysis, and data preprocessing

2. NumPy

- Purpose: Numerical computing
- Usage: Handling numerical operations and creating arrays for efficient computation

3. Matplotlib

- Purpose: Data visualization
- Usage: Plotting basic charts, including line plots, bar charts, area charts, and customizing visualizations

PYHTON LIBRARIES USED

4. Seaborn

- Purpose: Statistical data visualization
- Usage: Creating advanced plots, including histograms, scatter plots, and enhancing visual appeal

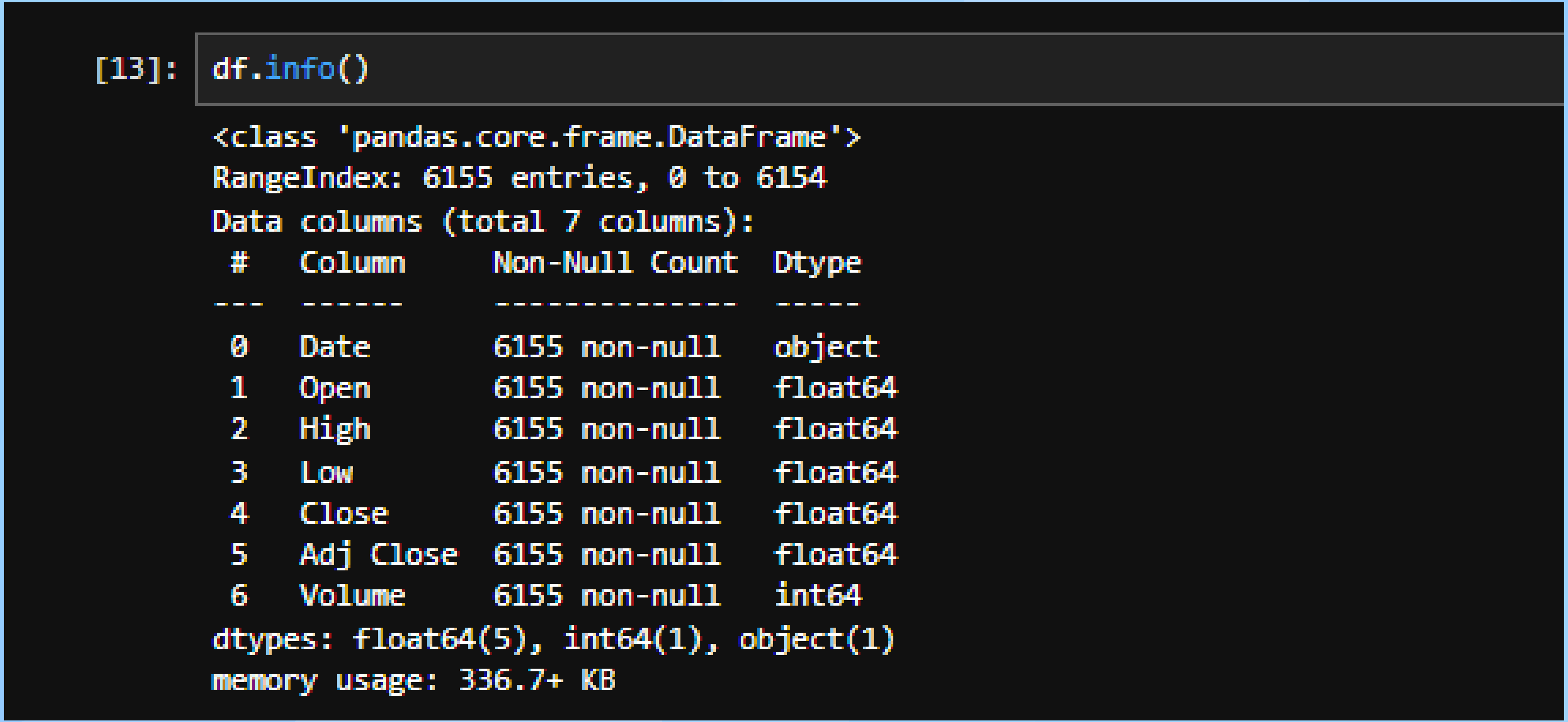
5. Statsmodels

- Purpose: Statistical modeling
- Usage: Performing seasonal decomposition of time series data to identify trends, seasonality, and residuals

6. Warnings

- Purpose: Managing warnings
- Usage: Controlling and filtering warning messages for a cleaner output during analysis

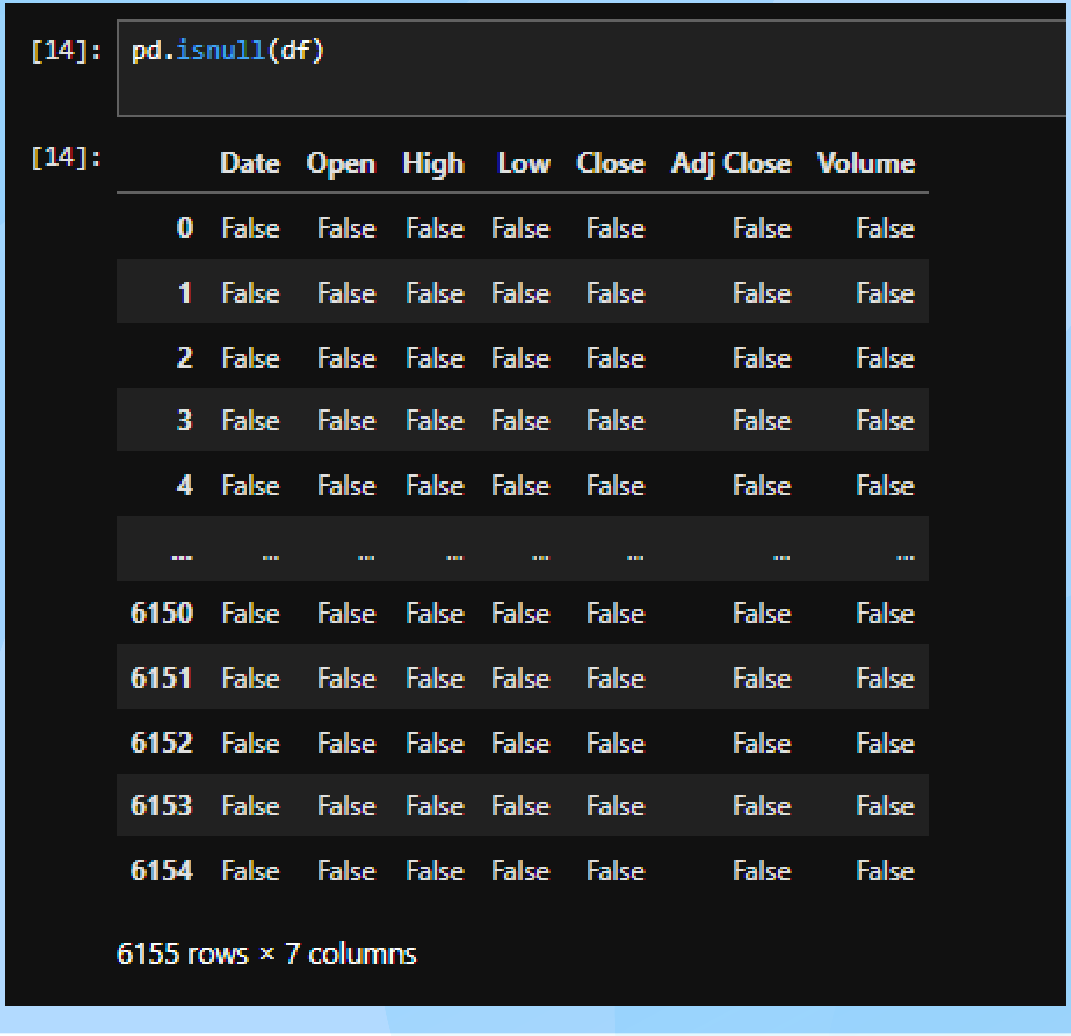
DATA CLEANING



```
[13]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6155 entries, 0 to 6154
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Date       6155 non-null   object 
 1   Open       6155 non-null   float64
 2   High       6155 non-null   float64
 3   Low        6155 non-null   float64
 4   Close      6155 non-null   float64
 5   Adj Close  6155 non-null   float64
 6   Volume     6155 non-null   int64  
dtypes: float64(5), int64(1), object(1)
memory usage: 336.7+ KB
```

DATA CLEANING



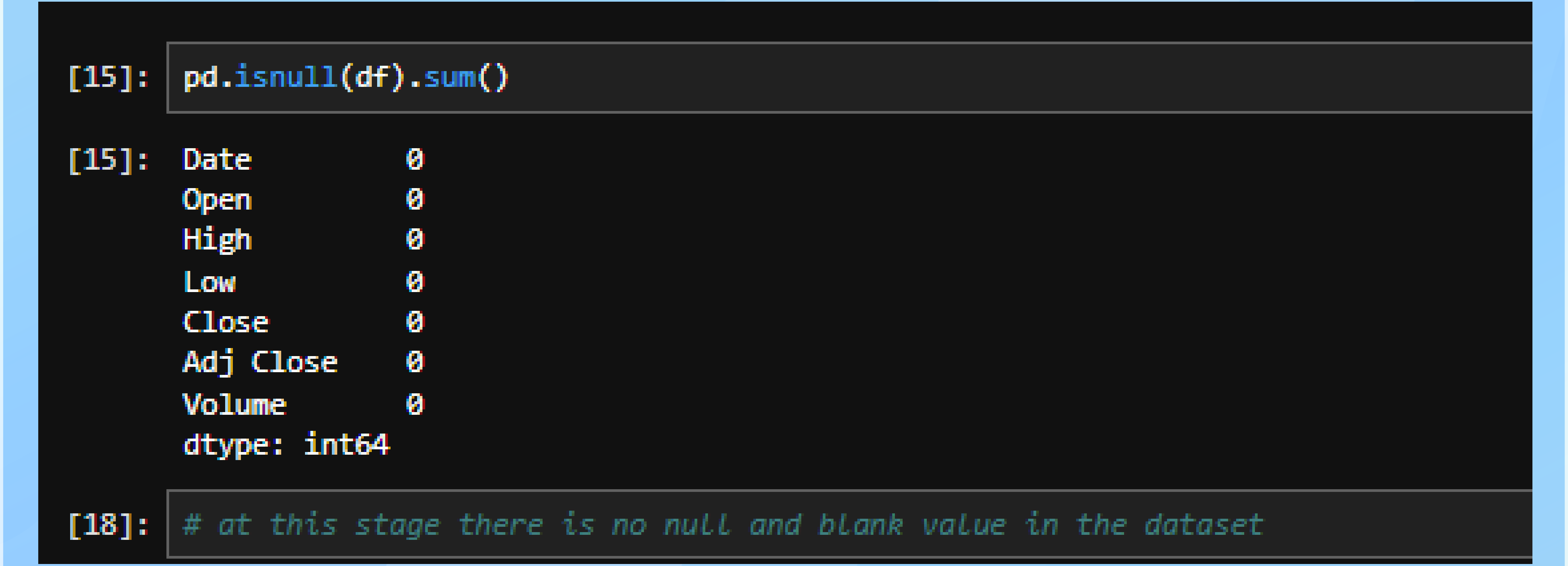
```
[14]: pd.isnull(df)
```

[14]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...
6150	False	False	False	False	False	False	False
6151	False	False	False	False	False	False	False
6152	False	False	False	False	False	False	False
6153	False	False	False	False	False	False	False
6154	False	False	False	False	False	False	False

6155 rows × 7 columns

DATA CLEANING



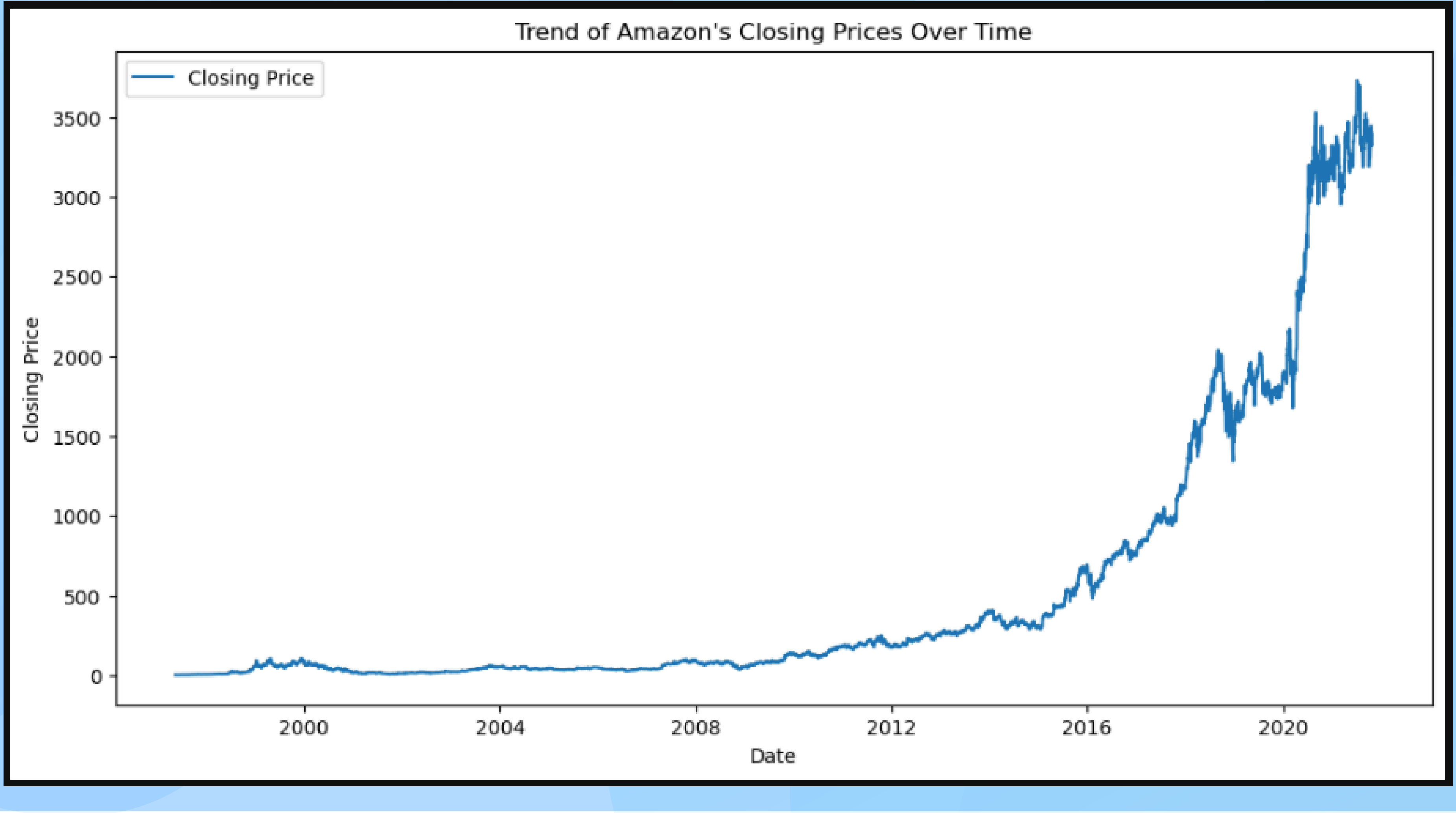
```
[15]: pd.isnull(df).sum()

[15]: Date         0
      Open         0
      High         0
      Low          0
      Close        0
      Adj Close    0
      Volume       0
      dtype: int64

[18]: # at this stage there is no null and blank value in the dataset
```

CODE OF QUESTION # 1

1. What is Amazon's closing price trend over the given period?

```
plt.figure(figsize=(12, 6))
plt.plot(df['Close'], label='Closing Price')
plt.title('Trend of Amazon\'s Closing Prices Over Time')
plt.xlabel('Date')
plt.ylabel('Closing Price')
plt.legend()
plt.show()
```

ANSWER OF QUESTION # 1

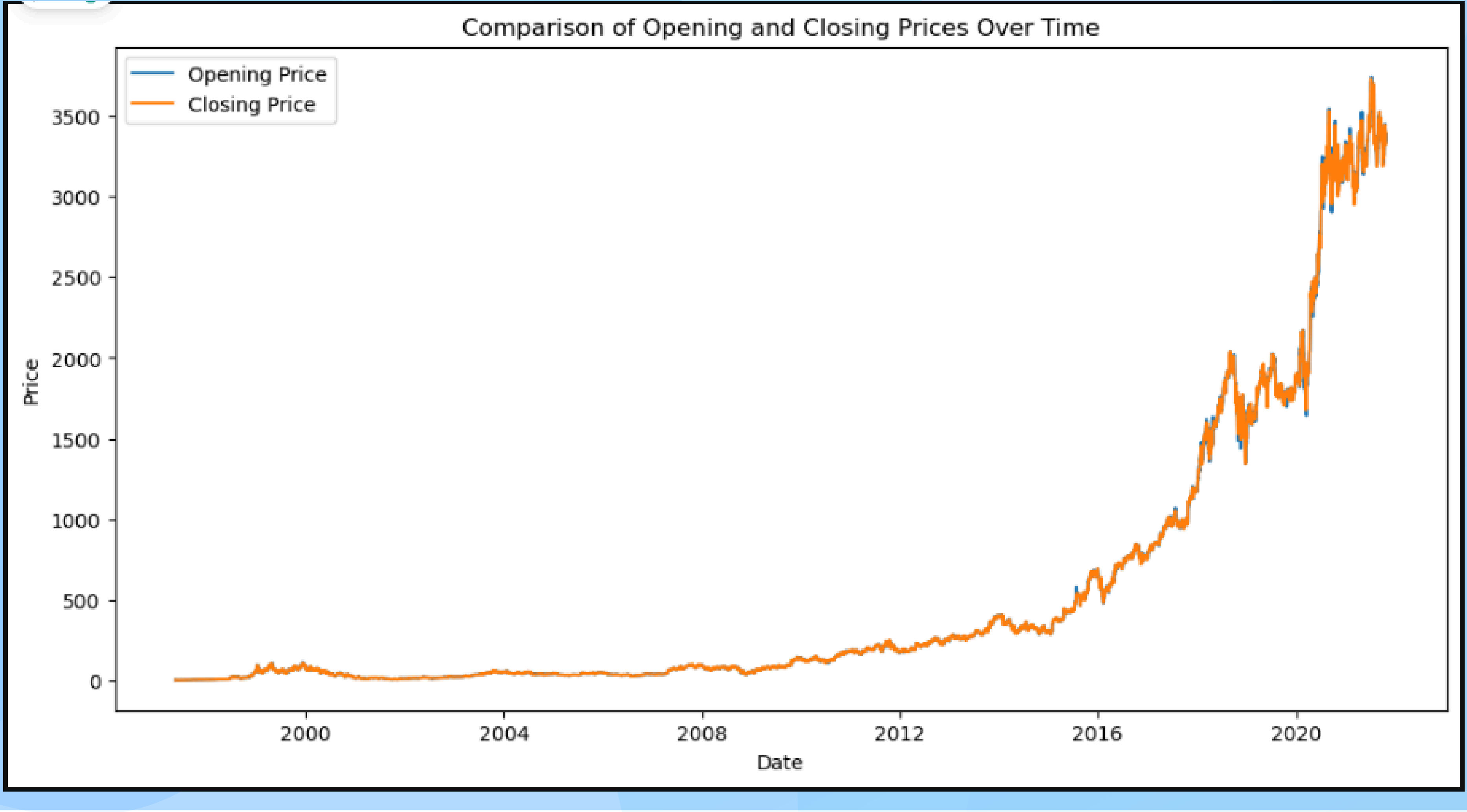


CODE OF QUESTION # 2

2. How does the opening price compare to the closing price over time?

```
plt.figure(figsize=(12, 6))
plt.plot(df['Open'], label='Opening Price')
plt.plot(df['Close'], label='Closing Price')
plt.title('Comparison of Opening and Closing Prices Over Time')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()
```

ANSWER OF QUESTION # 2

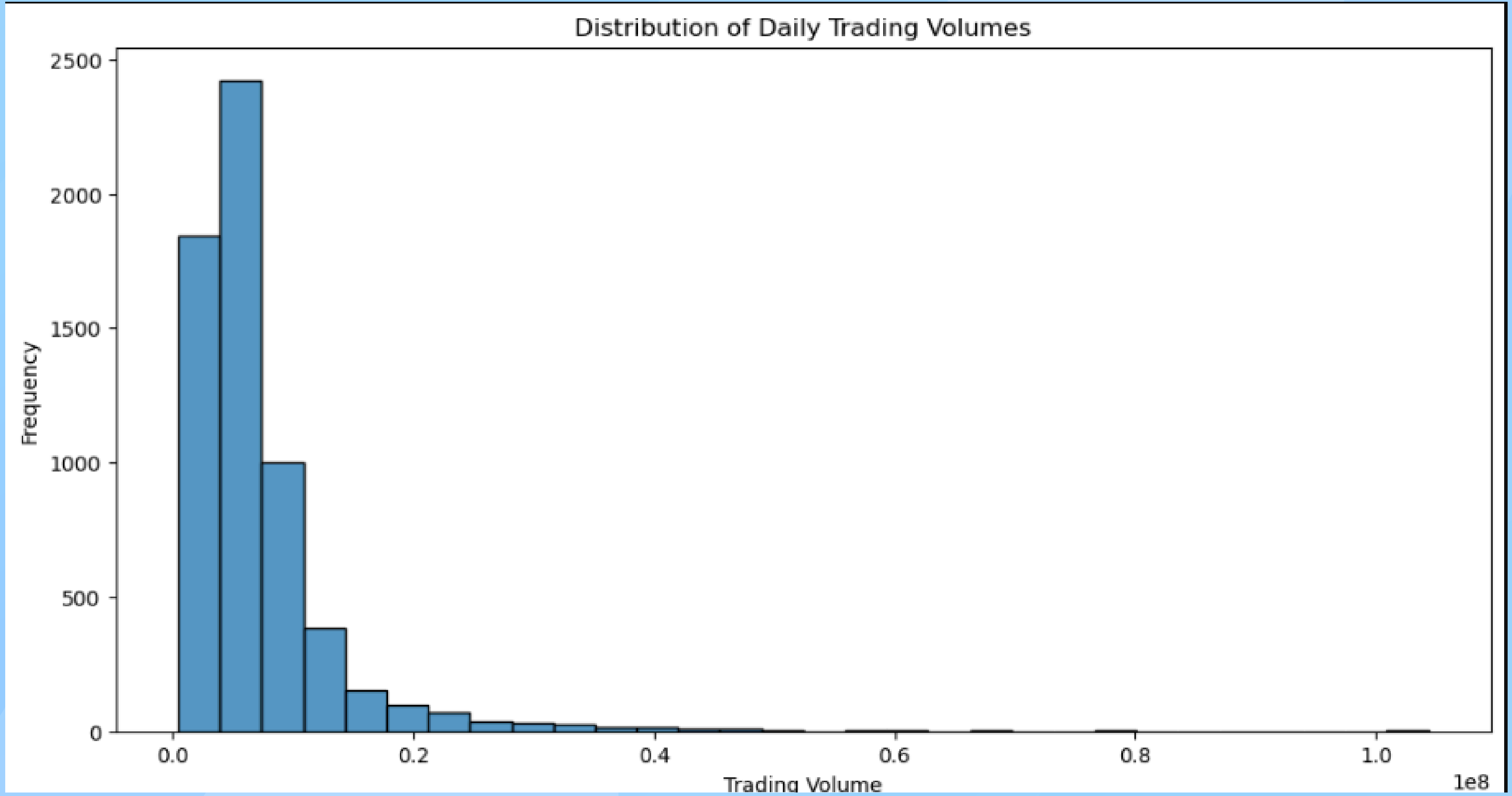


CODE OF QUESTION # 3

3. What is the distribution of daily trading volumes?

```
plt.figure(figsize=(12, 6))
sns.histplot(df['Volume'], bins=30)
plt.title('Distribution of Daily Trading Volumes')
plt.xlabel('Trading Volume')
plt.ylabel('Frequency')
plt.show()
```

ANSWER OF QUESTION # 3

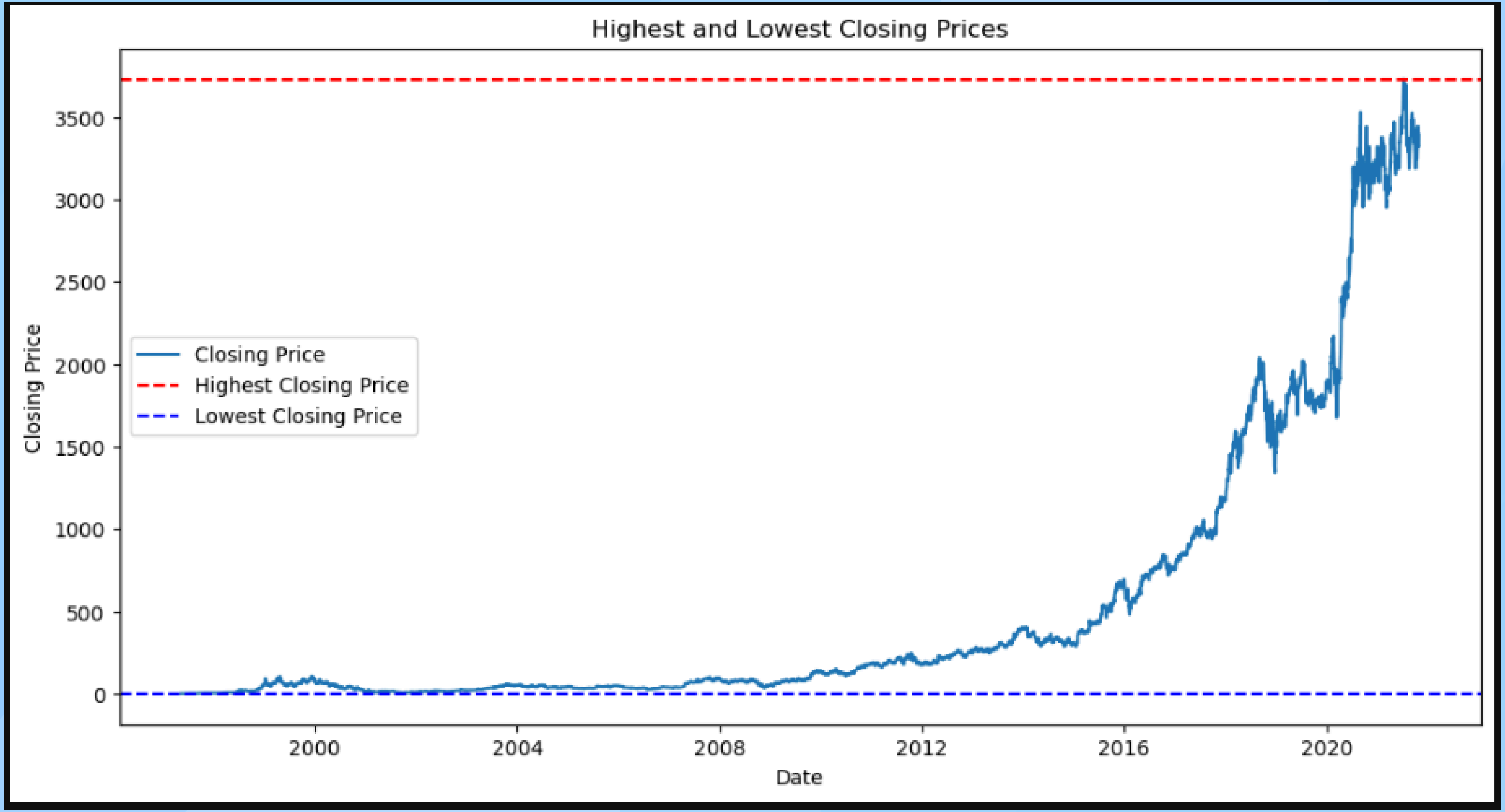


CODE OF QUESTION # 4

4. What are the highest and lowest closing prices in the dataset?

```
highest_closing_price = df['Close'].max()
lowest_closing_price = df['Close'].min()
print(f'Highest Closing Price: {highest_closing_price}')
print(f'Lowest Closing Price: {lowest_closing_price}')

plt.figure(figsize=(12, 6))
plt.plot(df['Close'], label='Closing Price')
plt.axhline(highest_closing_price, color='r', linestyle='--', label='Highest Closing Price')
plt.axhline(lowest_closing_price, color='b', linestyle='--', label='Lowest Closing Price')
plt.title('Highest and Lowest Closing Prices')
plt.xlabel('Date')
plt.ylabel('Closing Price')
plt.legend()
plt.show()
```

ANSWER OF QUESTION # 4



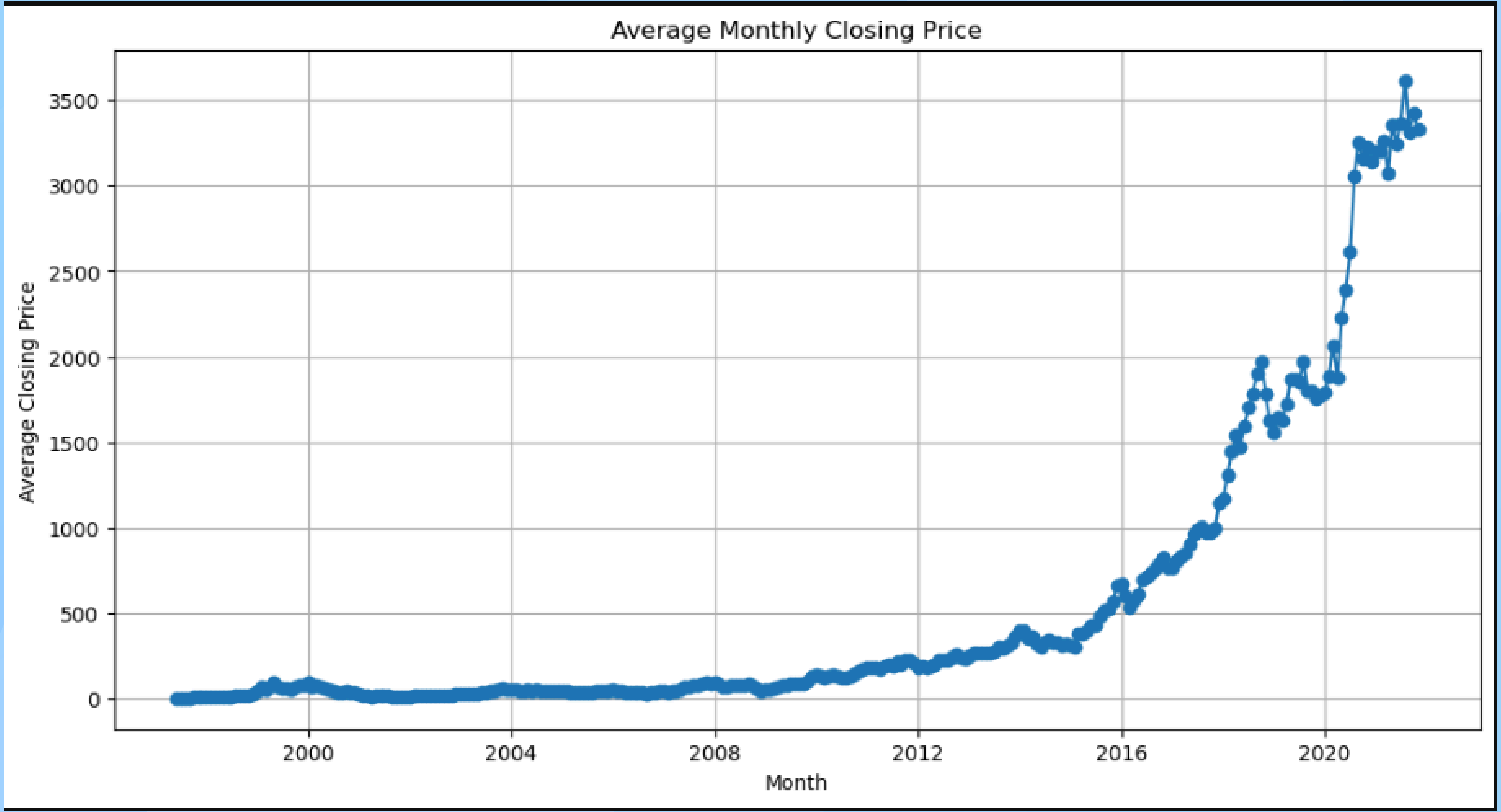
CODE OF QUESTION # 5

5. What is the average monthly closing price?

```
monthly_avg_closing_price = df['Close'].resample('M').mean()

# Plot the average monthly closing price using a line chart with markers
plt.figure(figsize=(12, 6))
plt.plot(monthly_avg_closing_price, marker='o')
plt.title('Average Monthly Closing Price')
plt.xlabel('Month')
plt.ylabel('Average Closing Price')
plt.grid(True)
plt.show()
```

ANSWER OF QUESTION # 5

5. What is the average monthly closing price?

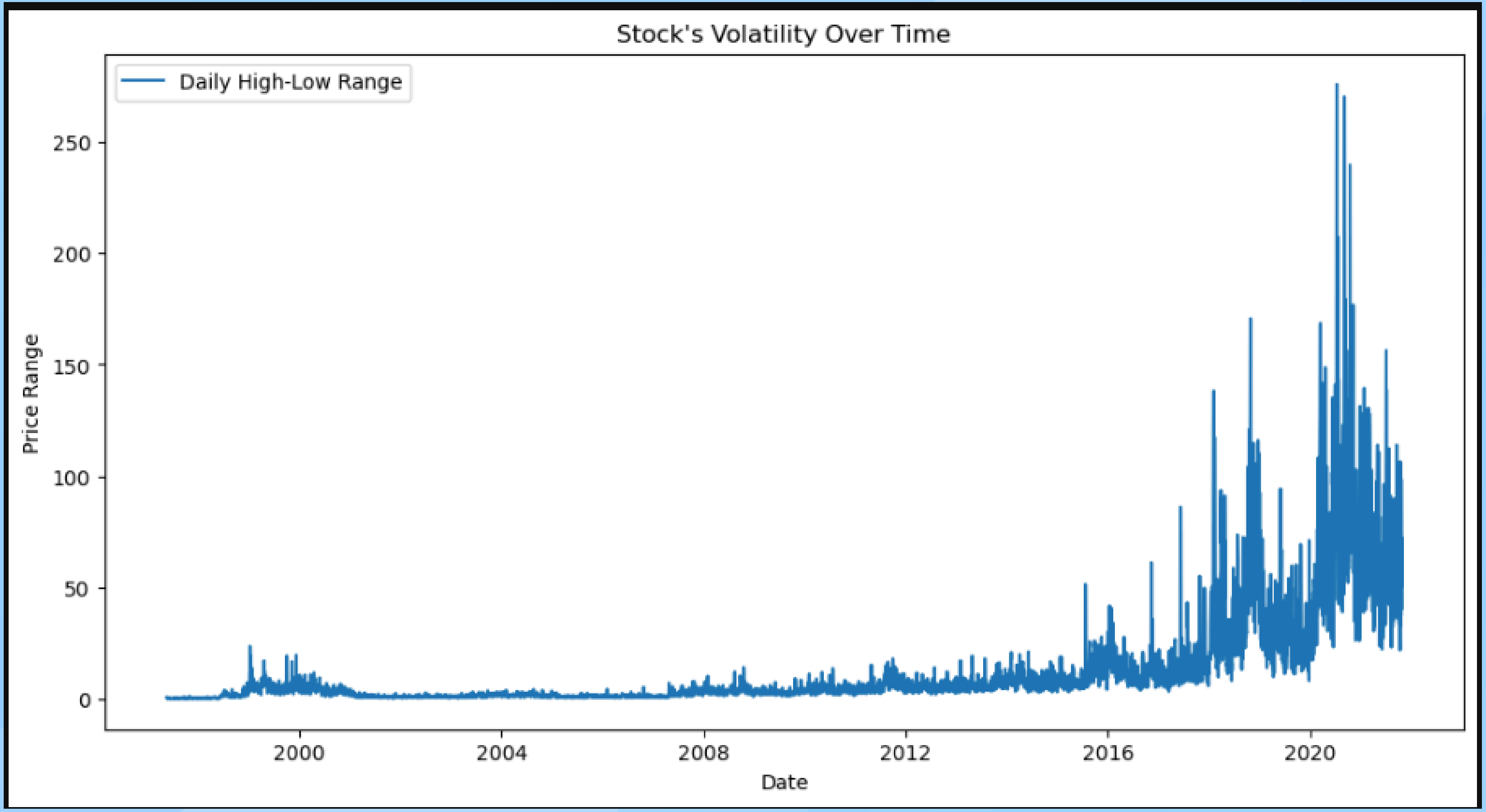


CODE OF QUESTION # 6

6. How does the stock's volatility change over time?

```
daily_high_low_range = df['High'] - df['Low']

plt.figure(figsize=(12, 6))
plt.plot(daily_high_low_range, label='Daily High-Low Range')
plt.title('Stock\'s Volatility Over Time')
plt.xlabel('Date')
plt.ylabel('Price Range')
plt.legend()
plt.show()
print ("Stock's Volatility Over Time \n" , daily_high_low_range)
```

ANSWER OF QUESTION # 6

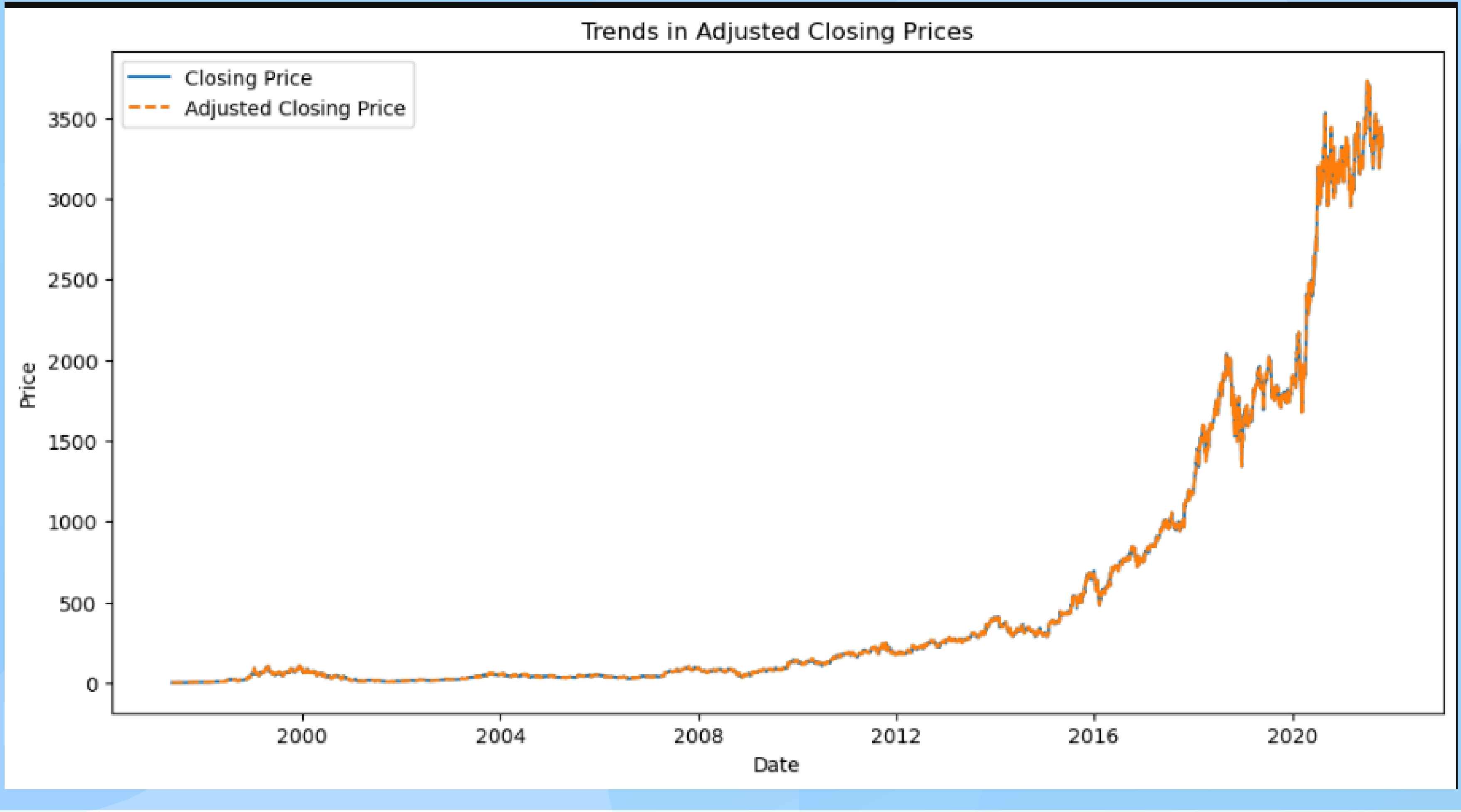


CODE OF QUESTION # 7

7. What are the significant trends in the adjusted closing prices?

```
plt.figure(figsize=(12, 6))
plt.plot(df['Close'], label='Closing Price')
plt.plot(df['Adj Close'], label='Adjusted Closing Price', linestyle='--')
plt.title('Trends in Adjusted Closing Prices')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()
```

ANSWER OF QUESTION # 7

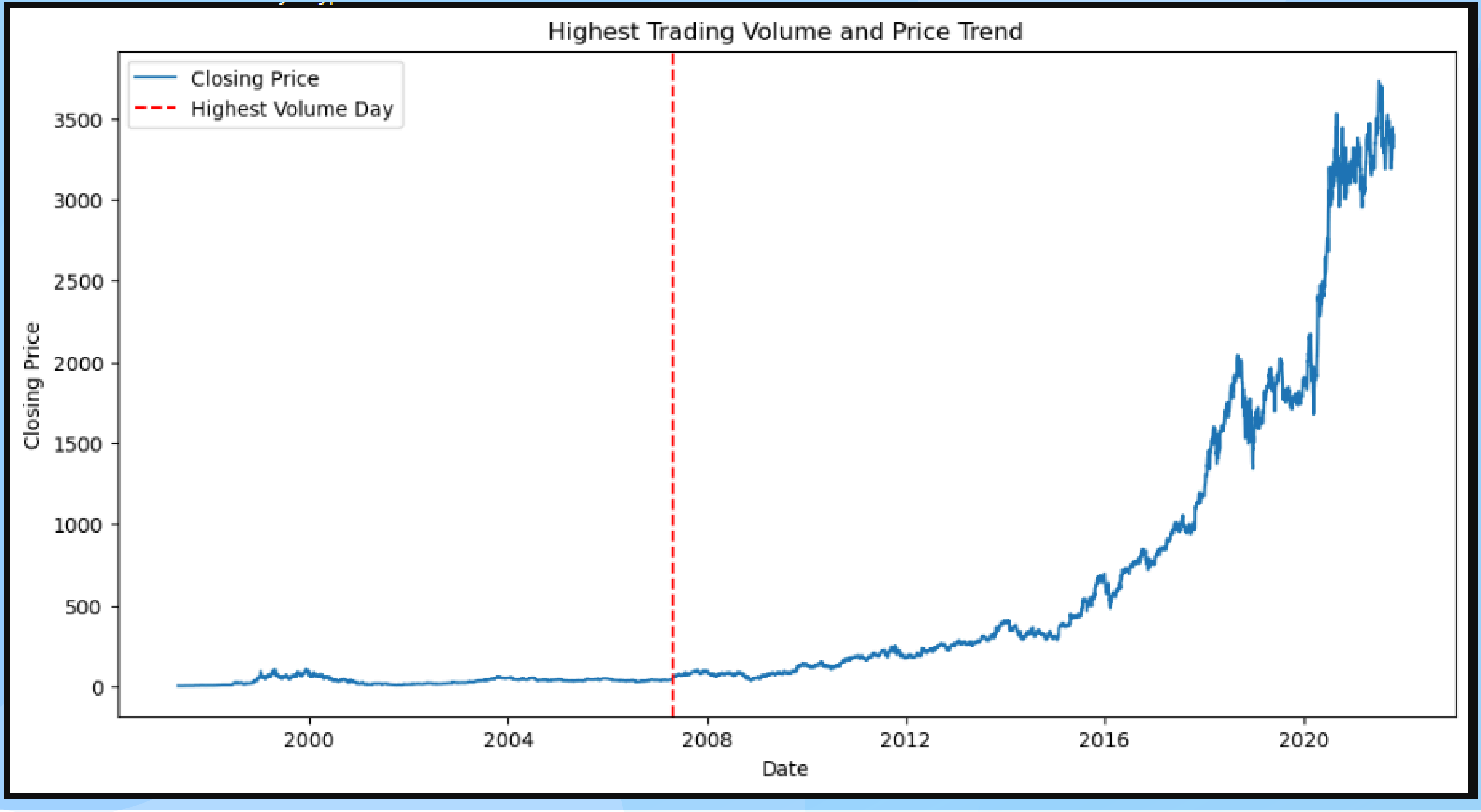


CODE OF QUESTION # 8

8. Which day had the highest trading volume and what was the price trend on that day?

```
highest_volume_day = df['Volume'].idxmax()
highest_volume_data = df.loc[highest_volume_day]
print(f'Highest Trading Volume Day: {highest_volume_day}')
print(highest_volume_data)

plt.figure(figsize=(12, 6))
plt.plot(df['Close'], label='Closing Price')
plt.axvline(highest_volume_day, color='r', linestyle='--', label='Highest Volume Day')
plt.title('Highest Trading Volume and Price Trend')
plt.xlabel('Date')
plt.ylabel('Closing Price')
plt.legend()
plt.show()
```

ANSWER OF QUESTION # 8

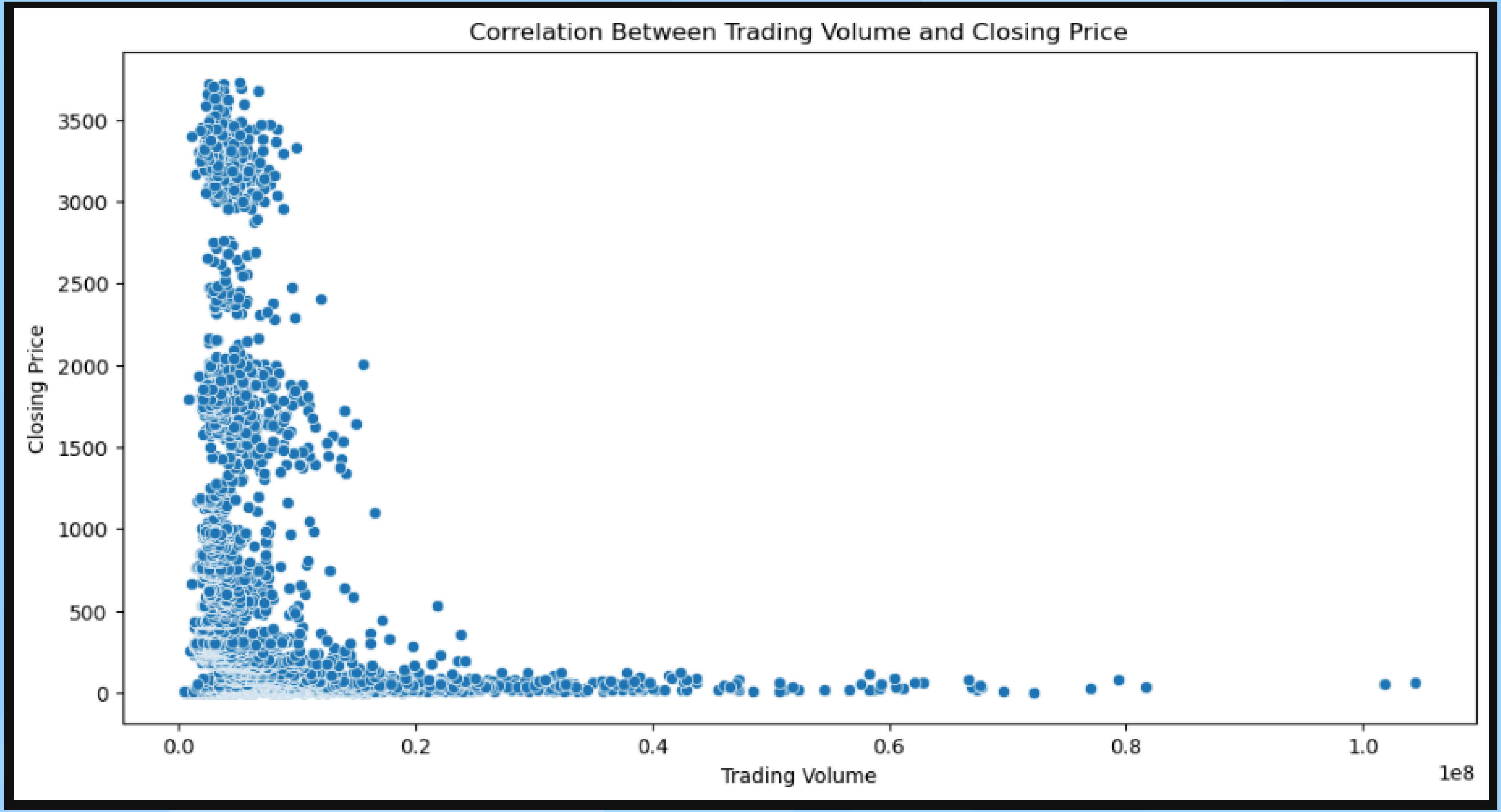


CODE OF QUESTION # 9

9. What is the correlation between trading volume and closing price?

```
plt.figure(figsize=(12, 6))
sns.scatterplot(x=df['Volume'], y=df['Close'])
plt.title('Correlation Between Trading Volume and Closing Price')
plt.xlabel('Trading Volume')
plt.ylabel('Closing Price')
plt.show()
```

ANSWER OF QUESTION # 9

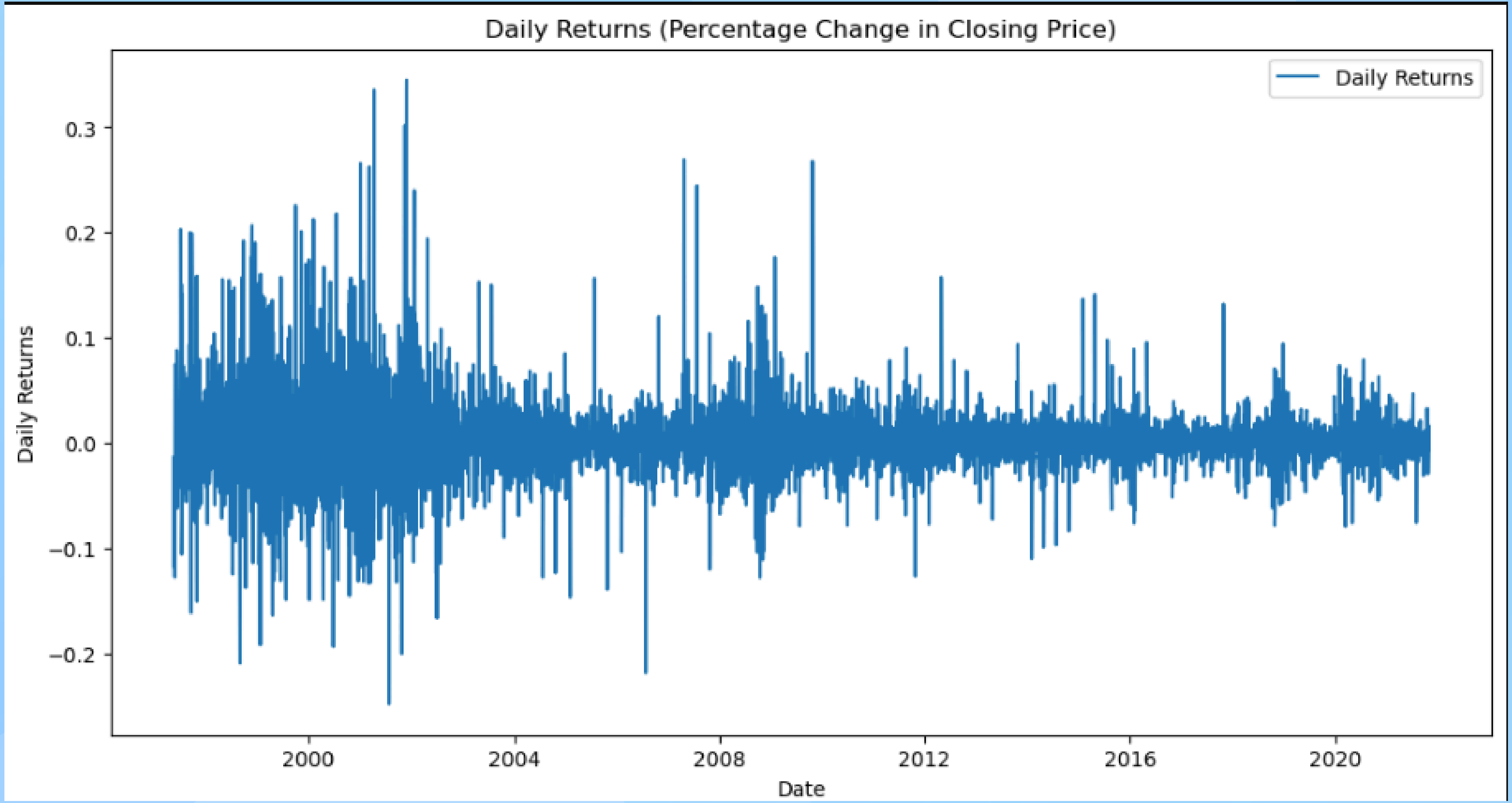


CODE OF QUESTION # 10

10. How do the daily returns (percentage change in closing price) vary over time?

```
df['Daily Returns'] = df['Close'].pct_change()

plt.figure(figsize=(12, 6))
plt.plot(df['Daily Returns'], label='Daily Returns')
plt.title('Daily Returns (Percentage Change in Closing Price)')
plt.xlabel('Date')
plt.ylabel('Daily Returns')
plt.legend()
plt.show()
```

ANSWER OF QUESTION # 10

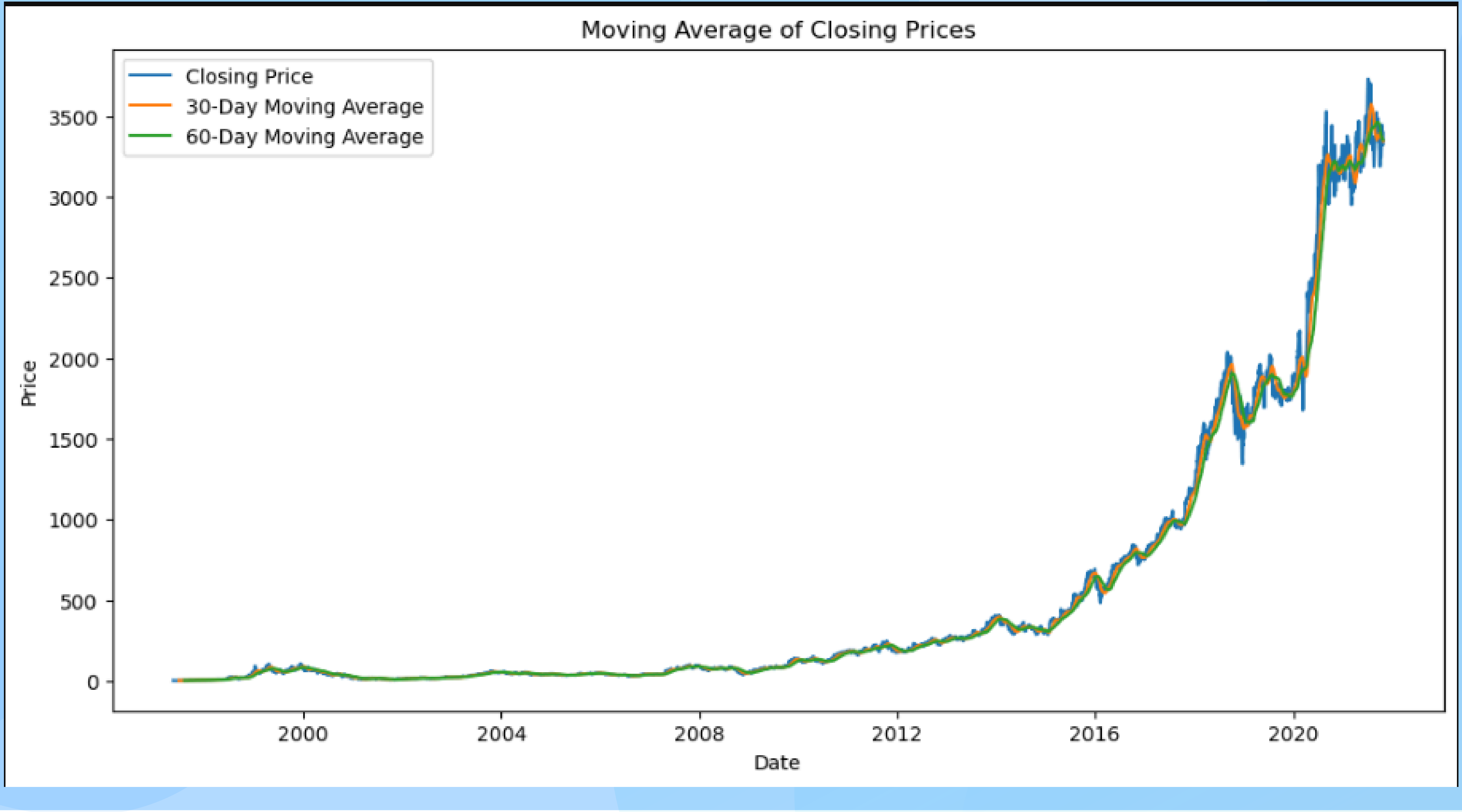


CODE OF QUESTION # 11

11. What is the moving average of the closing prices?

```
df['30-Day MA'] = df['Close'].rolling(window=30).mean()
df['60-Day MA'] = df['Close'].rolling(window=60).mean()

plt.figure(figsize=(12, 6))
plt.plot(df['Close'], label='Closing Price')
plt.plot(df['30-Day MA'], label='30-Day Moving Average')
plt.plot(df['60-Day MA'], label='60-Day Moving Average')
plt.title('Moving Average of Closing Prices')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()
```

ANSWER OF QUESTION # 11

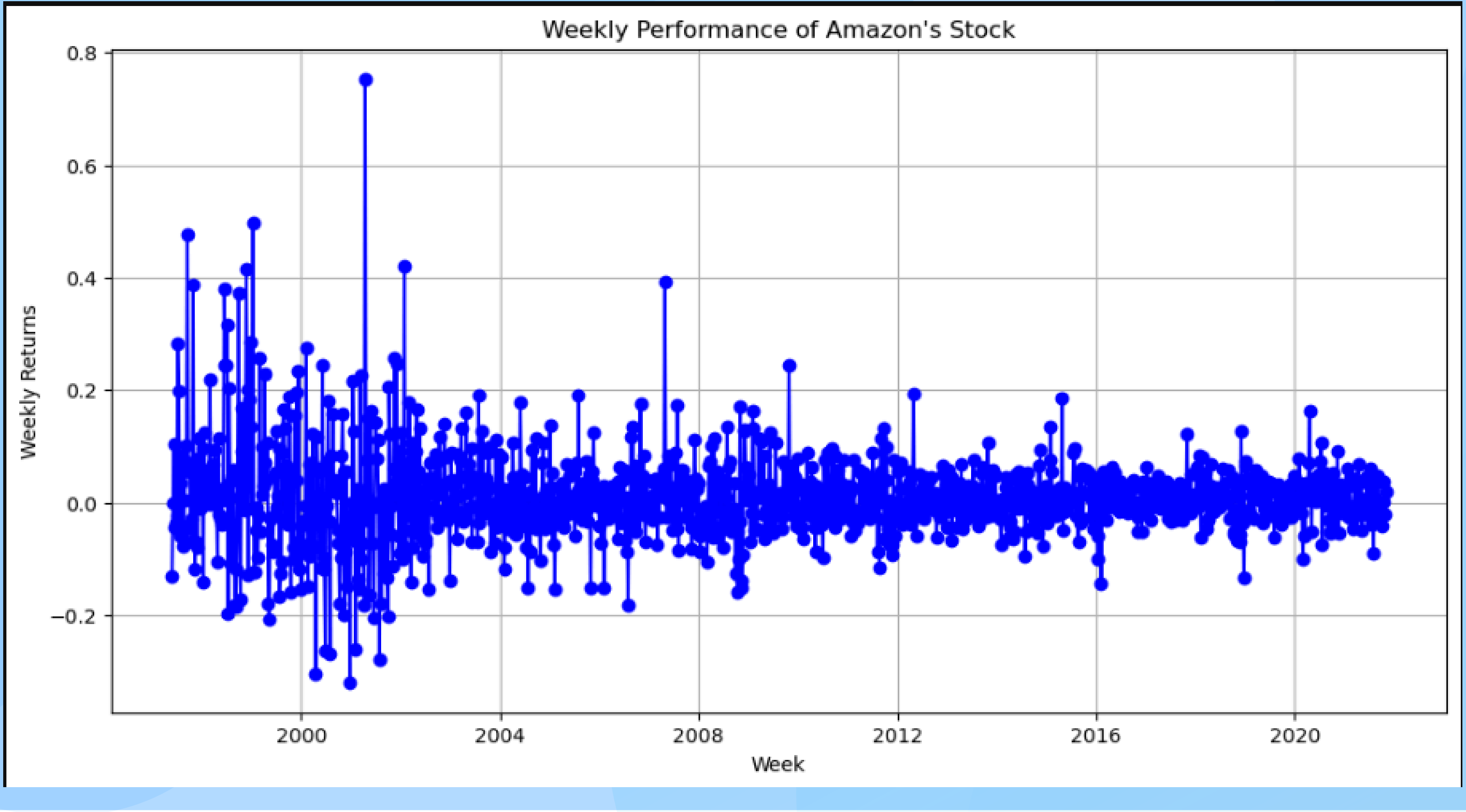


CODE OF QUESTION # 12

12. How did Amazon's stock perform every week?

```
# Calculate weekly returns
weekly_returns = df['Close'].resample('W').ffill().pct_change()

# Plot the weekly returns using a line chart with markers
plt.figure(figsize=(12, 6))
plt.plot(weekly_returns, marker='o', linestyle='-', color='b')
plt.title('Weekly Performance of Amazon\'s Stock')
plt.xlabel('Week')
plt.ylabel('Weekly Returns')
plt.grid(True)
plt.show()
```

ANSWER OF QUESTION # 12

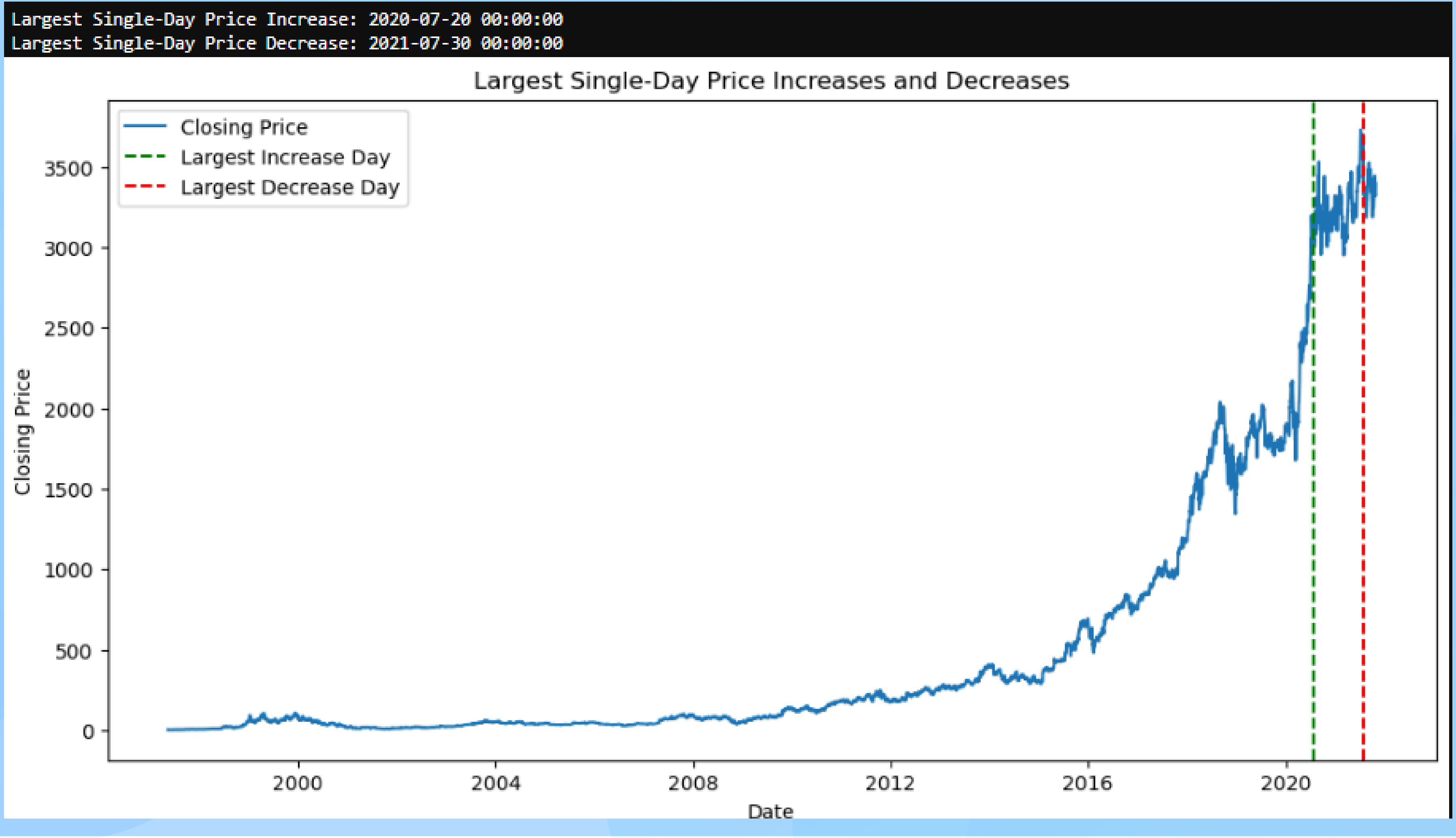


CODE OF QUESTION # 13

13. Which days experienced the highest single-day price increases and decreases?

```
daily_price_change = df['Close'].diff()
largest_increase_day = daily_price_change.idxmax()
largest_decrease_day = daily_price_change.idxmin()

print(f'Largest Single-Day Price Increase: {largest_increase_day}')
print(f'Largest Single-Day Price Decrease: {largest_decrease_day}')

plt.figure(figsize=(12, 6))
plt.plot(df['Close'], label='Closing Price')
plt.axvline(largest_increase_day, color='g', linestyle='--', label='Largest Increase Day')
plt.axvline(largest_decrease_day, color='r', linestyle='--', label='Largest Decrease Day')
plt.title('Largest Single-Day Price Increases and Decreases')
plt.xlabel('Date')
plt.ylabel('Closing Price')
plt.legend()
plt.show()
```

ANSWER OF QUESTION # 13



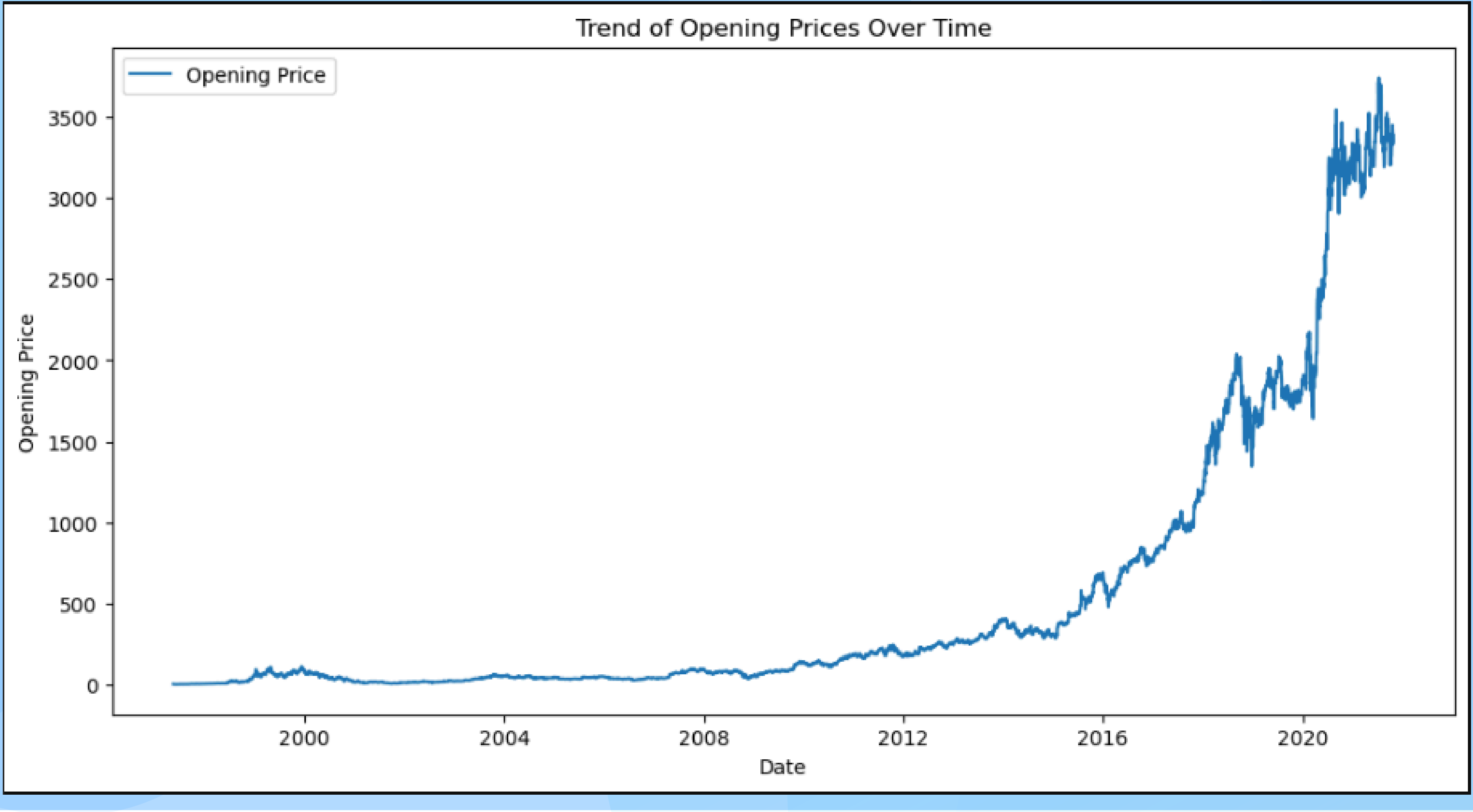
```
Largest Single-Day Price Increase: 2020-07-20 00:00:00
Largest Single-Day Price Decrease: 2021-07-30 00:00:00
```

CODE OF QUESTION # 14

14. What is the trend of the opening prices over time?

```
plt.figure(figsize=(12, 6))
plt.plot(df['Open'], label='Opening Price')
plt.title('Trend of Opening Prices Over Time')
plt.xlabel('Date')
plt.ylabel('Opening Price')
plt.legend()
plt.show()
```

ANSWER OF QUESTION # 14

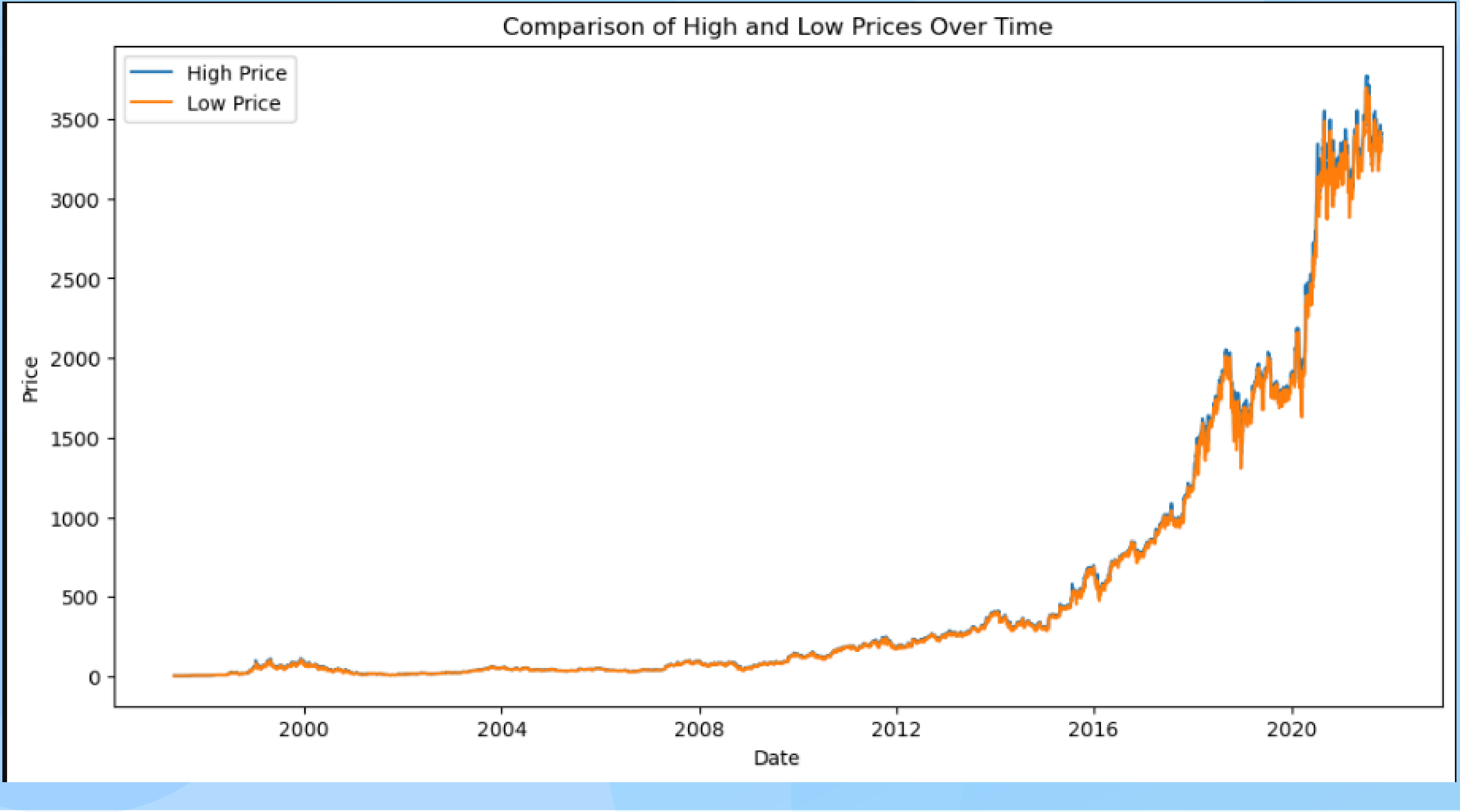


CODE OF QUESTION # 15

15. How do the high and low prices compare over the dataset period?

```
plt.figure(figsize=(12, 6))
plt.plot(df['High'], label='High Price')
plt.plot(df['Low'], label='Low Price')
plt.title('Comparison of High and Low Prices Over Time')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()
```

ANSWER OF QUESTION # 15

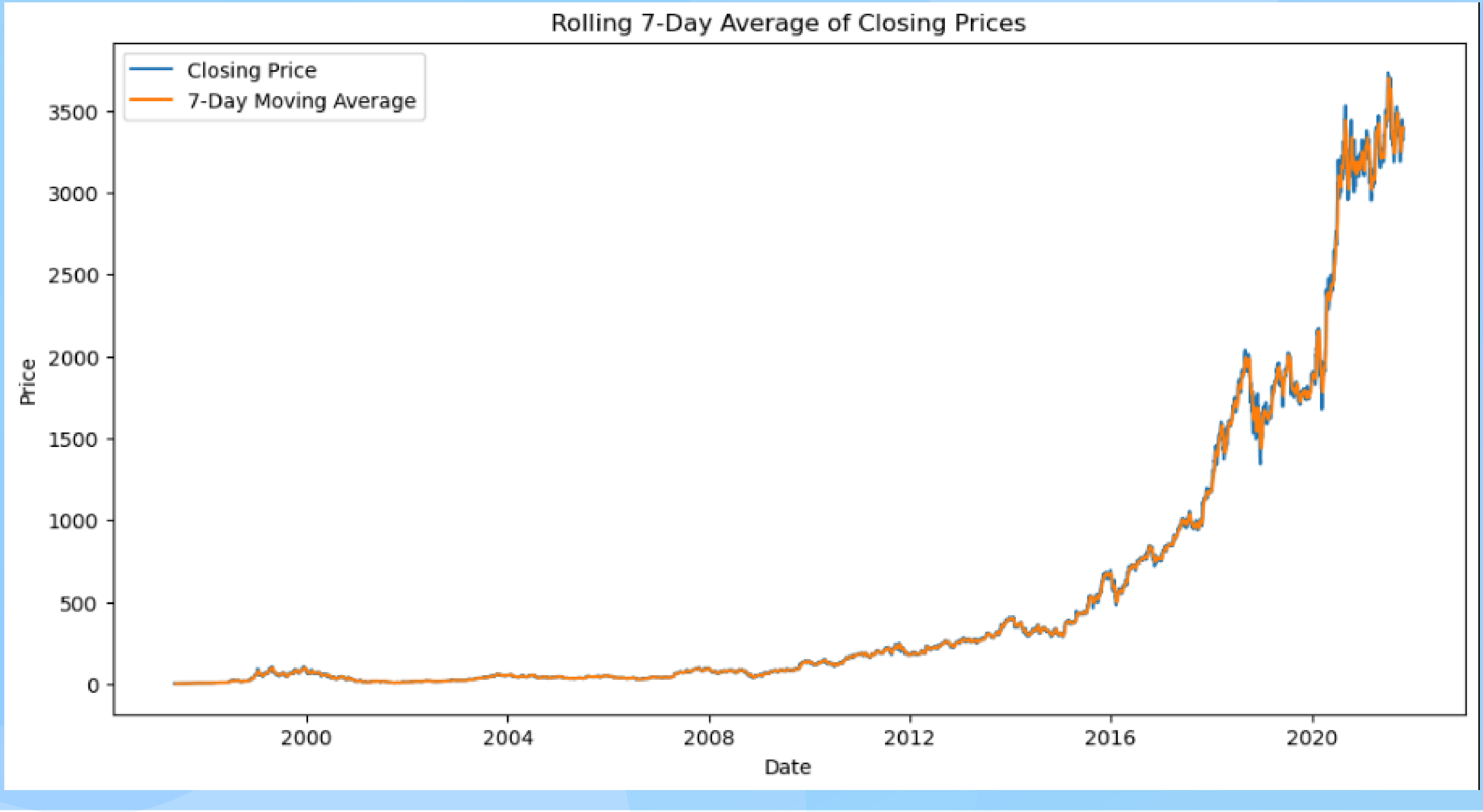


CODE OF QUESTION # 16

16. What is the rolling 7-day average of the closing prices

```
df['7-Day MA'] = df['Close'].rolling(window=7).mean()

plt.figure(figsize=(12, 6))
plt.plot(df['Close'], label='Closing Price')
plt.plot(df['7-Day MA'], label='7-Day Moving Average')
plt.title('Rolling 7-Day Average of Closing Prices')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()
```

ANSWER OF QUESTION # 16

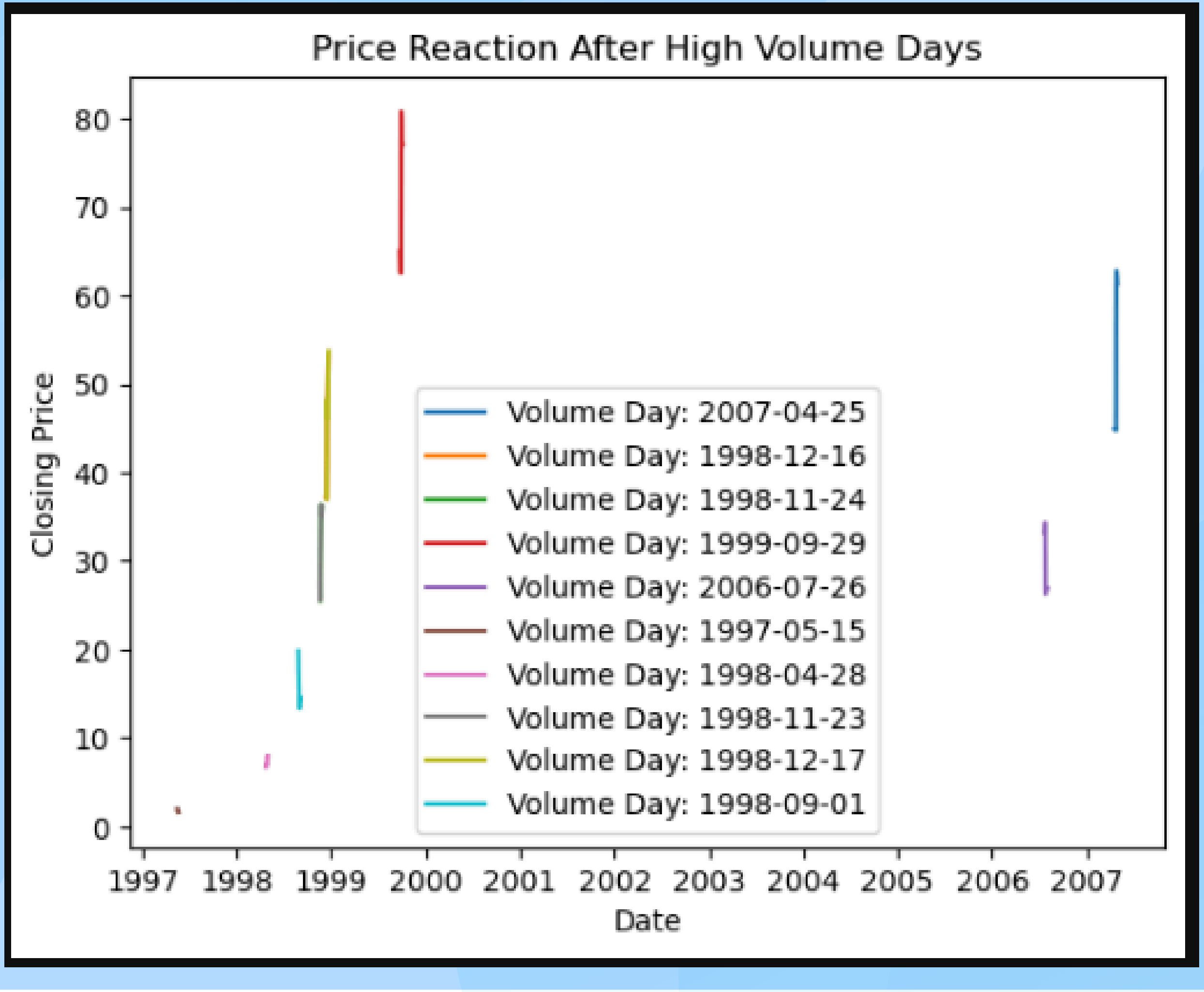


CODE OF QUESTION # 17

17. How does the stock price react after large trading volume days?

```
top_10_volume_days = df.nlargest(10, 'Volume').index

for day in top_10_volume_days:
    start_day = day - pd.Timedelta(days=5)
    end_day = day + pd.Timedelta(days=5)
    price_reaction = df.loc[start_day:end_day, 'Close']
    plt.plot(price_reaction, label=f'Volume Day: {day.date()}')

plt.title('Price Reaction After High Volume Days')
plt.xlabel('Date')
plt.ylabel('Closing Price')
plt.legend()
plt.show()
```

ANSWER OF QUESTION # 17

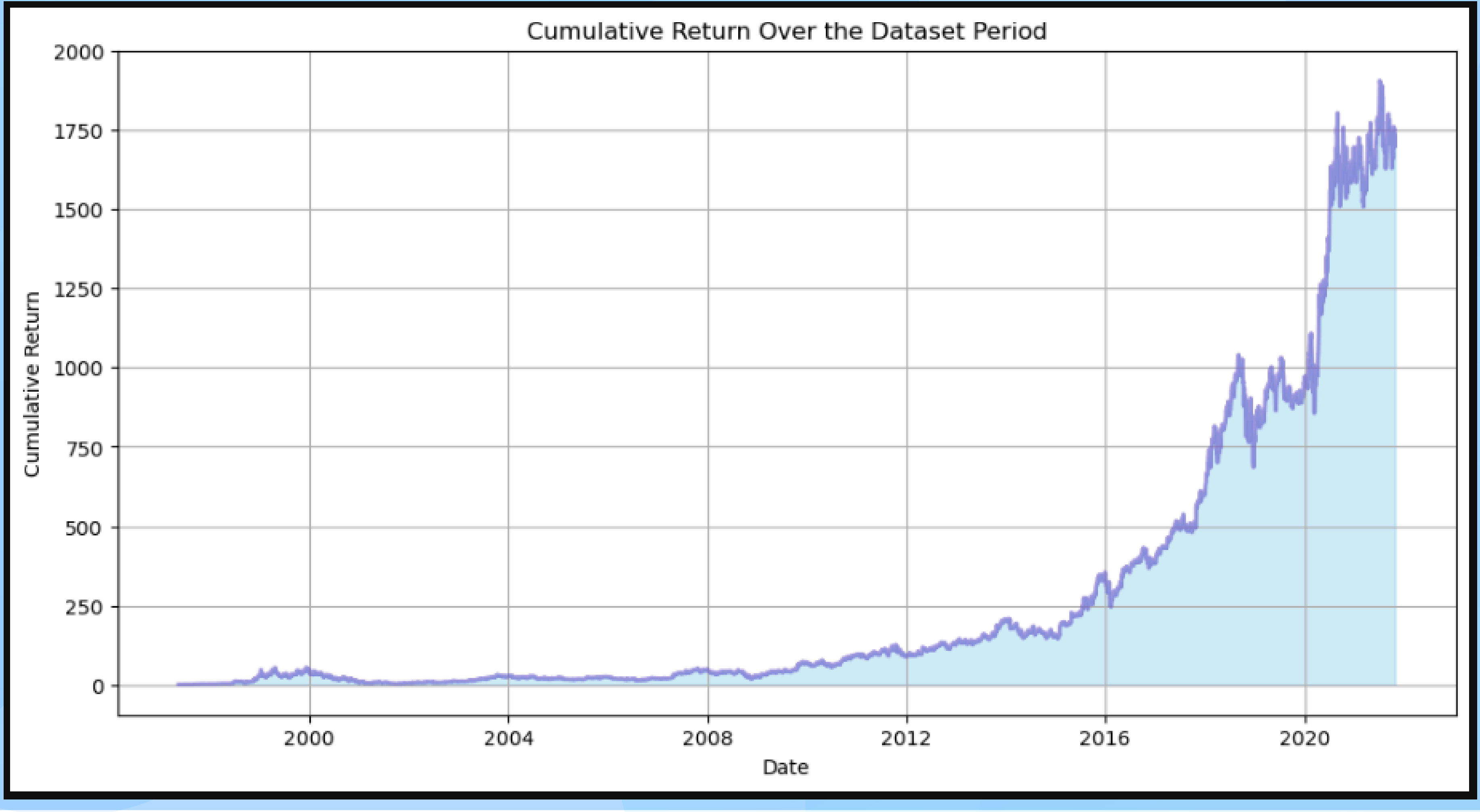


CODE OF QUESTION # 18

18. What is the cumulative return of Amazon's stock over the dataset period?

```
# Calculate daily returns
df['Daily Returns'] = df['Close'].pct_change()

# Calculate cumulative return
df['Cumulative Return'] = (1 + df['Daily Returns']).cumprod()

# Plot the cumulative return using an area chart
plt.figure(figsize=(12, 6))
plt.fill_between(df.index, df['Cumulative Return'], color='skyblue', alpha=0.4)
plt.plot(df['Cumulative Return'], color='Slateblue', alpha=0.6, linewidth=2)
plt.title('Cumulative Return Over the Dataset Period')
plt.xlabel('Date')
plt.ylabel('Cumulative Return')
plt.grid(True)
plt.show()
```

ANSWER OF QUESTION # 18

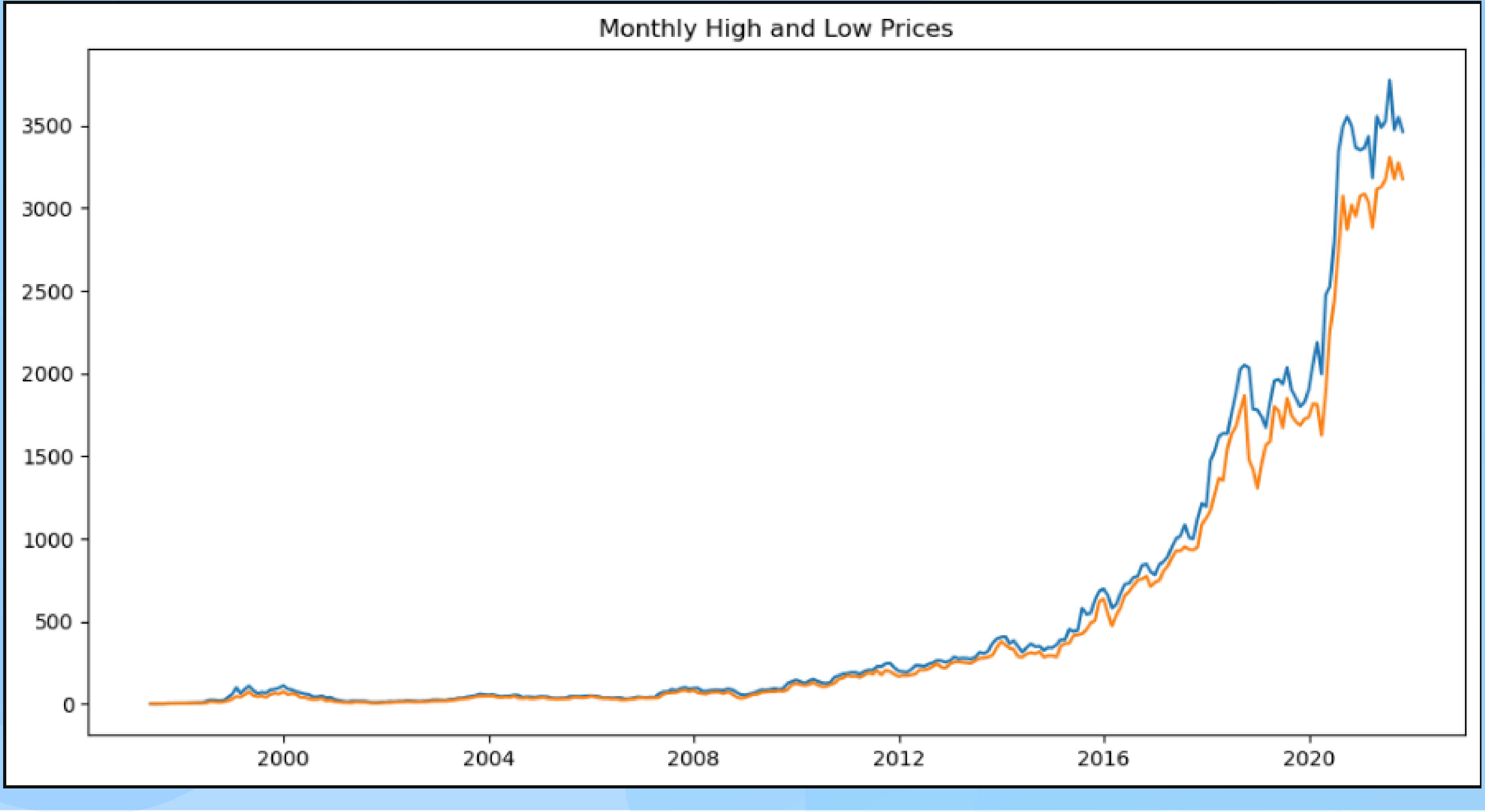


CODE OF QUESTION # 19

19. How do the monthly high and low prices compare over the dataset period?

```
monthly_high = df['High'].resample('M').max()
monthly_low = df['Low'].resample('M').min()

plt.figure(figsize=(12, 6))
plt.plot(monthly_high, label='Monthly High')
plt.plot(monthly_low, label='Monthly Low')
plt.title('Monthly High and Low Prices')
```

ANSWER OF QUESTION # 19

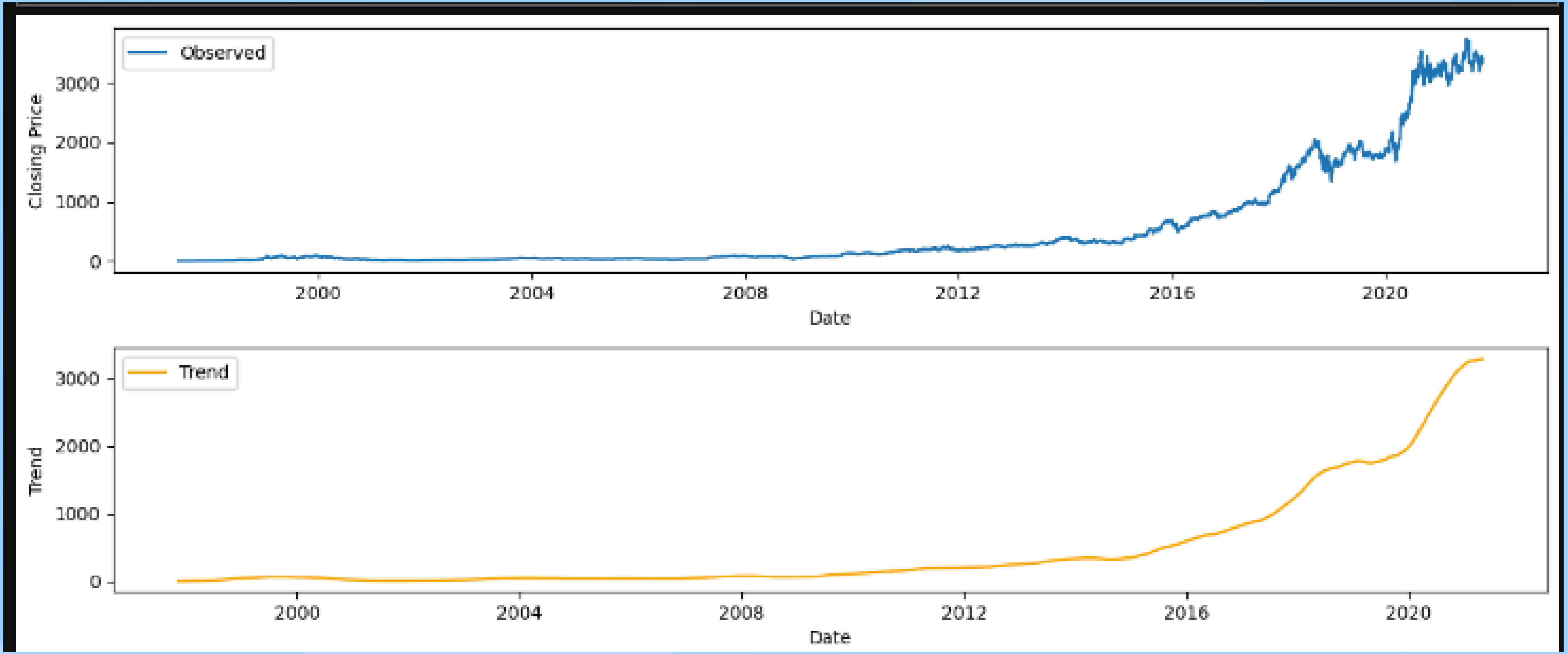


CODE OF QUESTION # 20

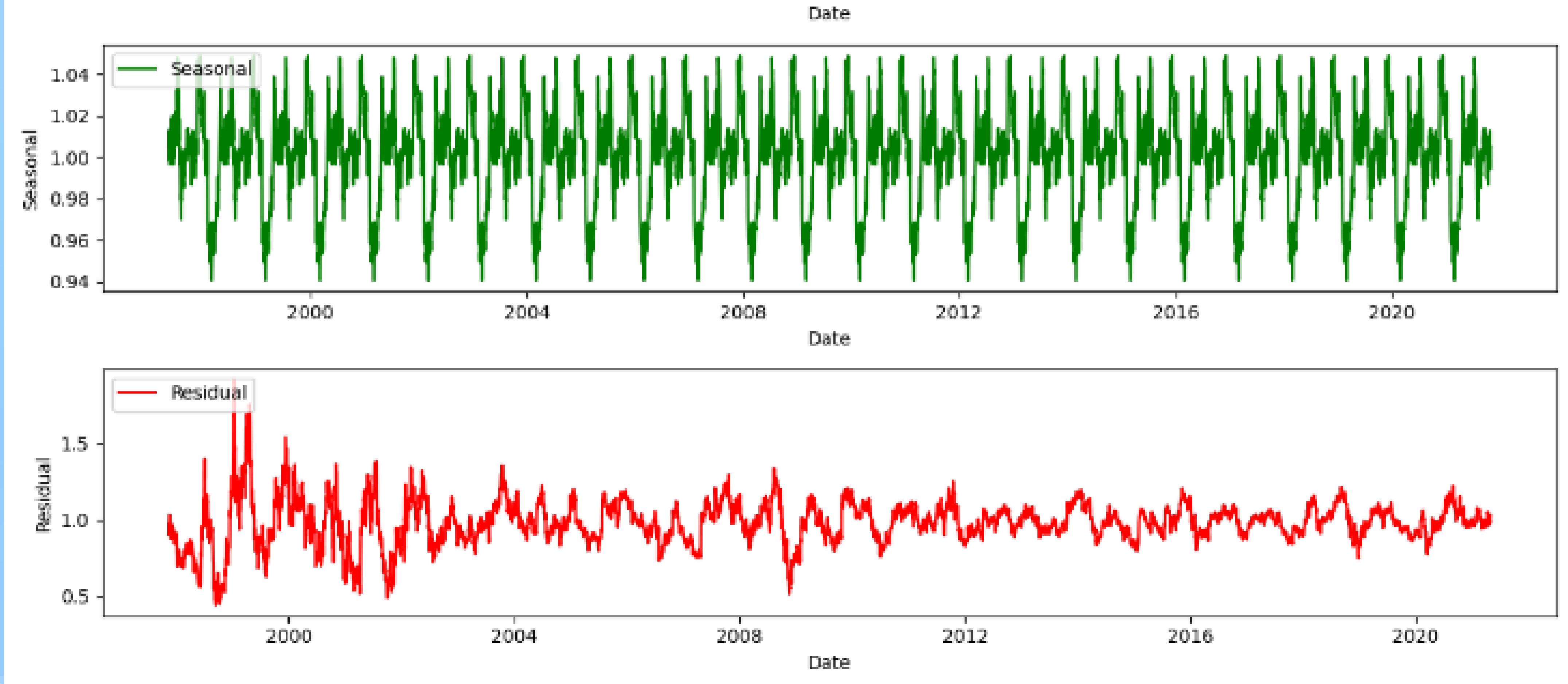
20. What are the seasonal trends in Amazon's stock prices?

```
# Ensure the data is at a regular frequency (e.g., daily)
df = df.asfreq('D')

# Fill any missing values (if necessary)
df['Close'] = df['Close'].ffill()

# Perform seasonal decomposition
result = seasonal_decompose(df['Close'],
model='multiplicative', period=365)

# Plot the seasonal decomposition
plt.figure(figsize=(12, 10))

plt.subplot(411)
plt.plot(result.observed, label='Observed')
plt.legend(loc='upper left')
plt.xlabel('Date')
plt.ylabel('Closing Price')

plt.subplot(412)
plt.plot(result.trend, label='Trend', color='orange')
plt.legend(loc='upper left')
plt.xlabel('Date')
plt.ylabel('Trend')

plt.subplot(413)
plt.plot(result.seasonal, label='Seasonal', color='green')
plt.legend(loc='upper left')
plt.xlabel('Date')
plt.ylabel('Seasonal')

plt.subplot(414)
plt.plot(result.resid, label='Residual', color='red')
plt.legend(loc='upper left')
plt.xlabel('Date')
plt.ylabel('Residual')

plt.tight_layout()
plt.show()
```

ANSWER OF QUESTION # 20



ANSWER OF QUESTION # 20



Usama

Computer Engineer

usama200101010@gmail.com
03099604550

Exploring the field of data science

Skills

- OOP
- Data Structure
- Database
- Python
- Data Science
- MySQL
- JAVA
- C++
- C
- Blockchain
- Solidity