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


data=pd.read_csv('stocks.csv')


data.head()
data.describe()
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 248 entries, 0 to 247
Data columns (total 8 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Ticker     248 non-null    object 
 1   Date       248 non-null    object 
 2   Open       248 non-null    float64
 3   High       248 non-null    float64
 4   Low        248 non-null    float64
 5   Close      248 non-null    float64
 6   Adj Close  248 non-null    float64
 7   Volume     248 non-null    int64  
dtypes: float64(5), int64(1), object(2)
memory usage: 15.6+ KB
```

```
print(data.isnull().sum())


data.fillna(method='ffill', inplace=True)
```

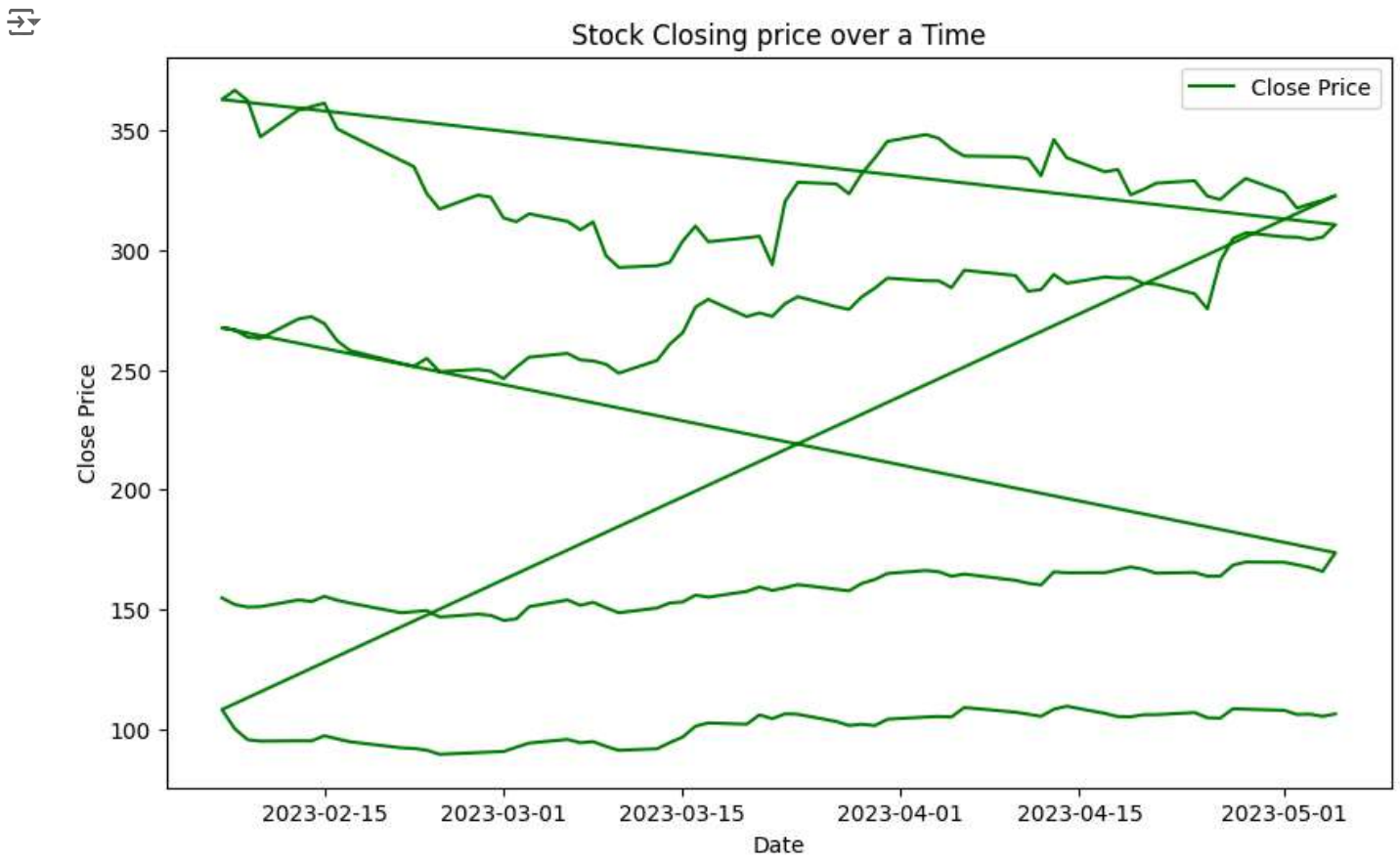
```
Ticker       0
Date         0
Open         0
High         0
Low          0
Close        0
Adj Close    0
Volume       0
dtype: int64
<ipython-input-5-6a5c96c7883b>:4: FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise
  data.fillna(method='ffill', inplace=True)
```

```
data.head(5)
```

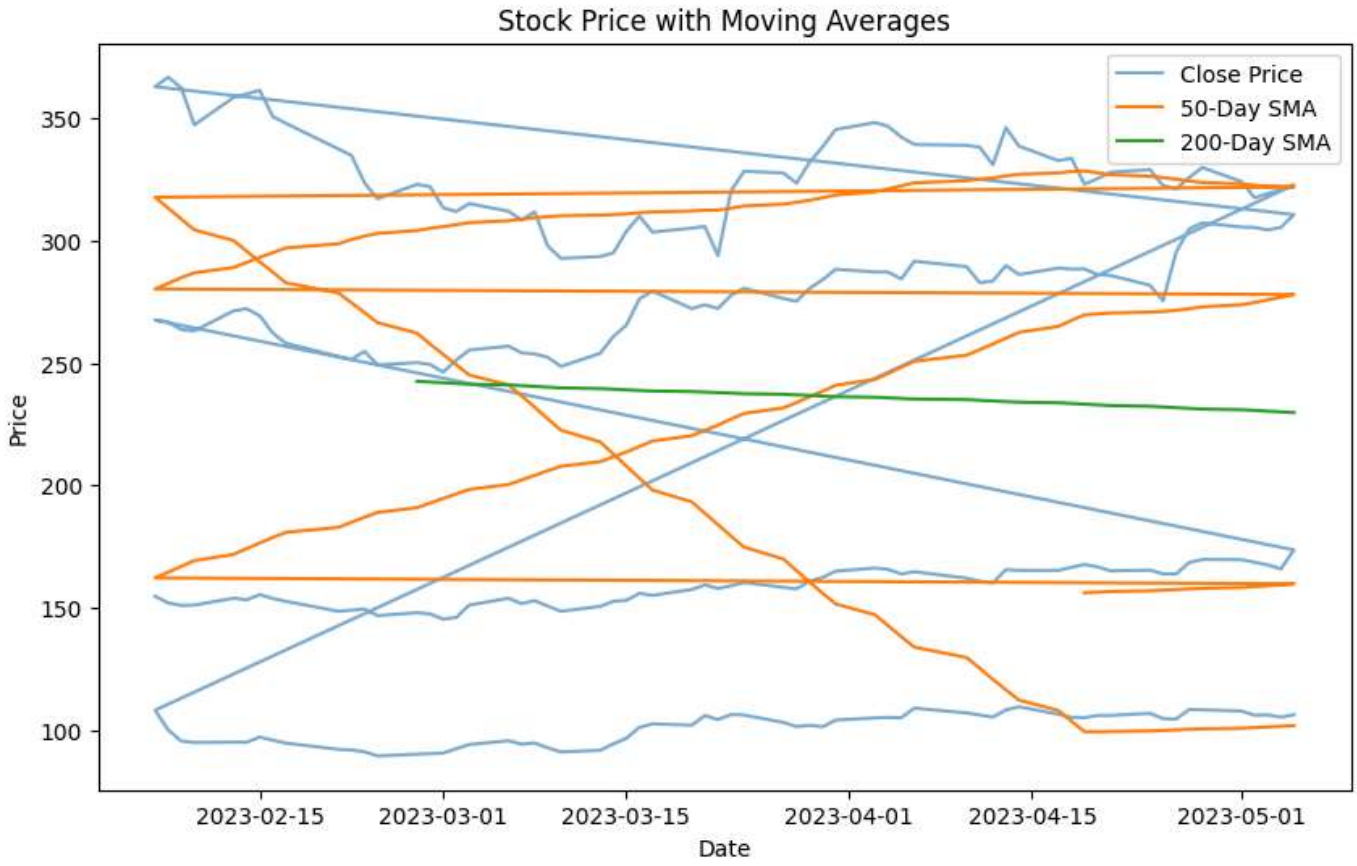
	Ticker	Date	Open	High	Low	Close	Adj Close	Volume
0	AAPL	2023-02-07	150.639999	155.229996	150.639999	154.649994	154.414230	83322600
1	AAPL	2023-02-08	153.880005	154.580002	151.169998	151.919998	151.688400	64120100
2	AAPL	2023-02-09	153.779999	154.330002	150.419998	150.869995	150.639999	56007100
3	AAPL	2023-02-10	149.460007	151.339996	149.220001	151.009995	151.009995	57450700
4	AAPL	2023-02-13	150.949997	154.259995	150.919998	153.850006	153.850006	62199000

Next steps: Generate code with data | View recommended plots | New interactive sheet

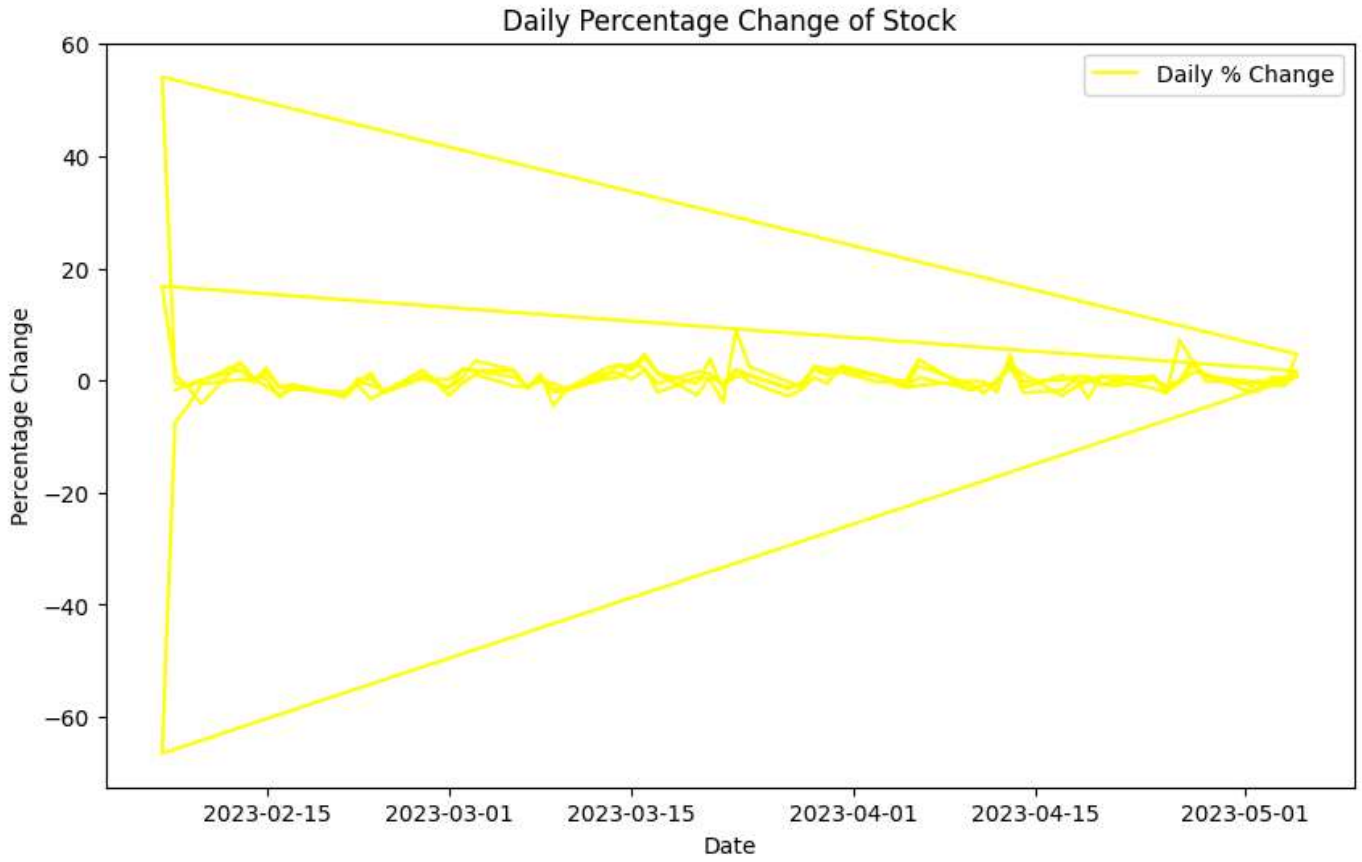
```
data['Date'] = pd.to_datetime(data['Date'])
data.set_index('Date', inplace=True)  # Set the Date as the index



plt.figure(figsize=(10, 6))
plt.plot(data['Close'], label='Close Price',color='green')
plt.title('Stock Closing price over a Time')
plt.xlabel('Date')
plt.ylabel('Close Price')
plt.legend()
plt.show()
```



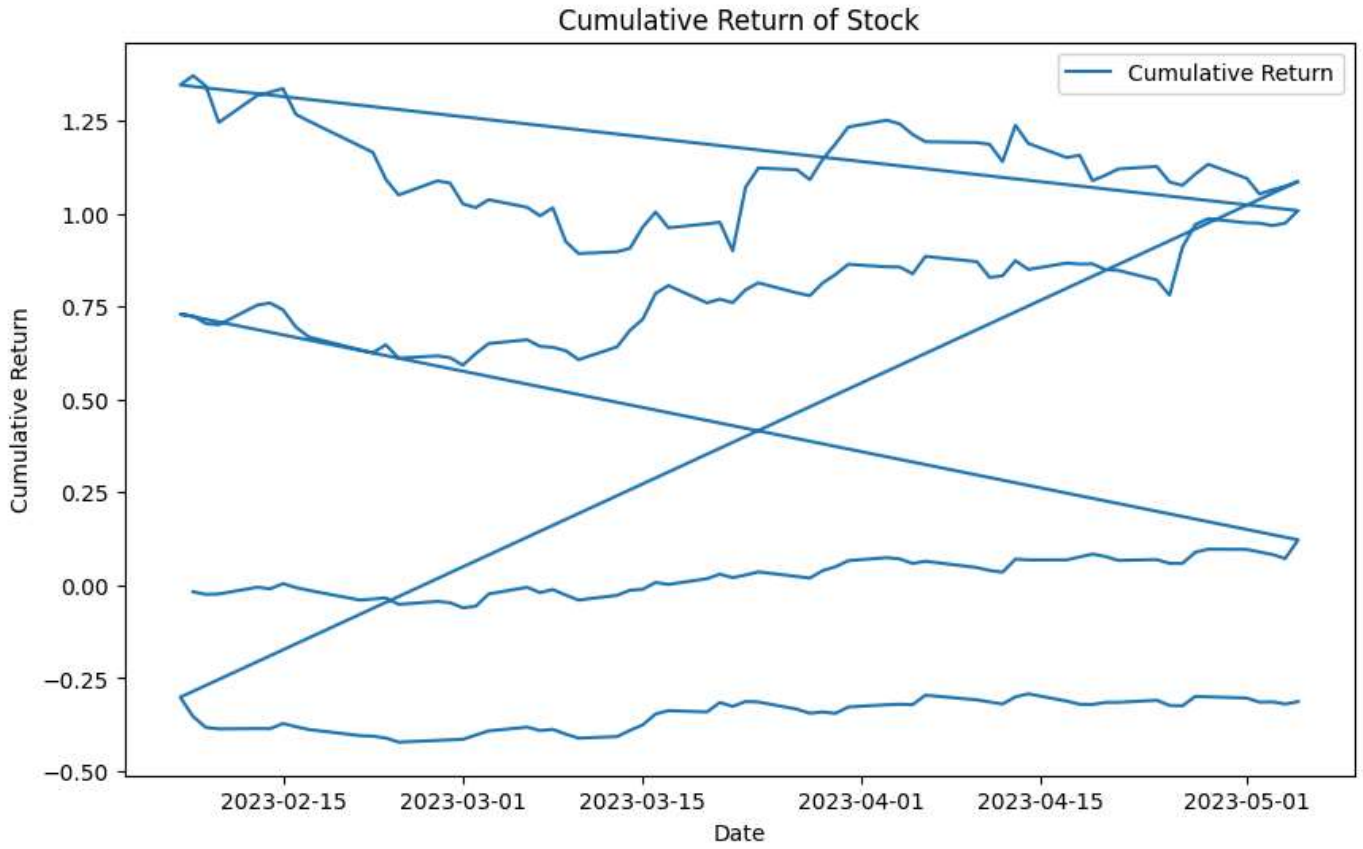

```
data['SMA_50'] = data['Close'].rolling(window=50).mean()
data['SMA_200'] = data['Close'].rolling(window=200).mean()

plt.figure(figsize=(10, 6))
plt.plot(data['Close'], label='Close Price', alpha=0.6)
plt.plot(data['SMA_50'], label='50-Day SMA')
plt.plot(data['SMA_200'], label='200-Day SMA')
plt.title('Stock Price with Moving Averages')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()
```




```
data['Daily Change'] = data['Close'].pct_change() * 100

plt.figure(figsize=(10, 6))
plt.plot(data['Daily Change'], label='Daily % Change',color='yellow')
plt.title('Daily Percentage Change of Stock')
plt.xlabel('Date')
plt.ylabel('Percentage Change')
plt.legend()
plt.show()
```




```
data['Cumulative Return'] = (1 + data['Daily Change'] / 100).cumprod() - 1

plt.figure(figsize=(10, 6))
plt.plot(data['Cumulative Return'], label='Cumulative Return')
plt.title('Cumulative Return of Stock')
plt.xlabel('Date')
plt.ylabel('Cumulative Return')
plt.legend()
plt.show()
```




```
plt.figure(figsize=(10, 6))
plt.bar(data.index, data['Volume'], color='green', label='Volume')
plt.title('Stock Trading Volume Over Time')
plt.xlabel('Date')
plt.ylabel('Volume')
plt.legend()
plt.show()
```


```
correlation_matrix = data[['Open', 'High', 'Low', 'Close', 'Volume']].corr()
print(correlation_matrix)
```

```
            Open      High       Low     Close    Volume
Open    1.000000  0.999626  0.999650  0.999176 -0.547741
High    0.999626  1.000000  0.999654  0.999644 -0.546175
Low     0.999650  0.999654  1.000000  0.999663 -0.544590
Close   0.999176  0.999644  0.999663  1.000000 -0.544194
Volume -0.547741 -0.546175 -0.544590 -0.544194  1.000000
```

```
data.to_csv('FINAL_stock_data.csv')
```

```
pd.read_csv('FINAL_stock_data.csv')
```

	Date	Ticker	Open	High	Low	Close	Adj Close	Volume	SMA_50	SMA_200	
0	2023-02-07	AAPL	150.639999	155.229996	150.639999	154.649994	154.414230	83322600	NaN	NaN	
1	2023-02-08	AAPL	153.880005	154.580002	151.169998	151.919998	151.688400	64120100	NaN	NaN	-
2	2023-02-09	AAPL	153.779999	154.330002	150.419998	150.869995	150.639999	56007100	NaN	NaN	-
3	2023-02-10	AAPL	149.460007	151.339996	149.220001	151.009995	151.009995	57450700	NaN	NaN	
	2023										