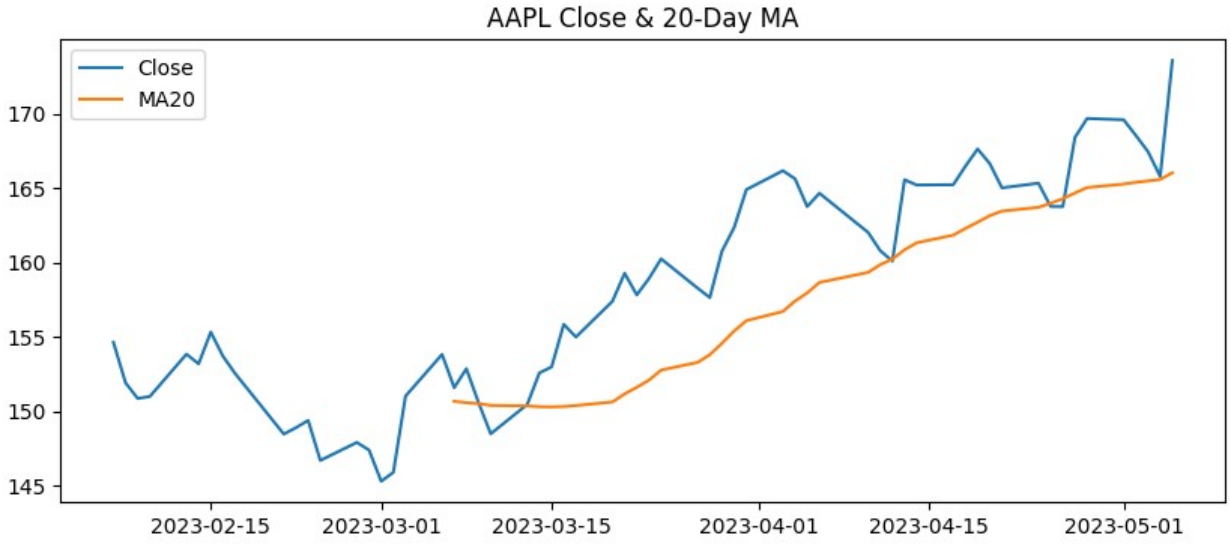
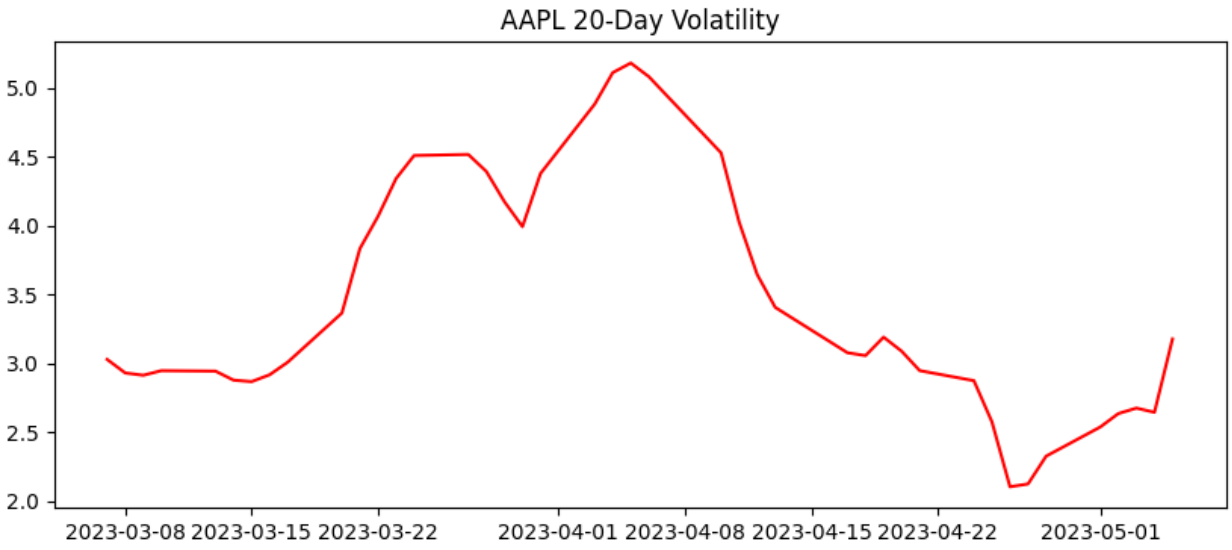
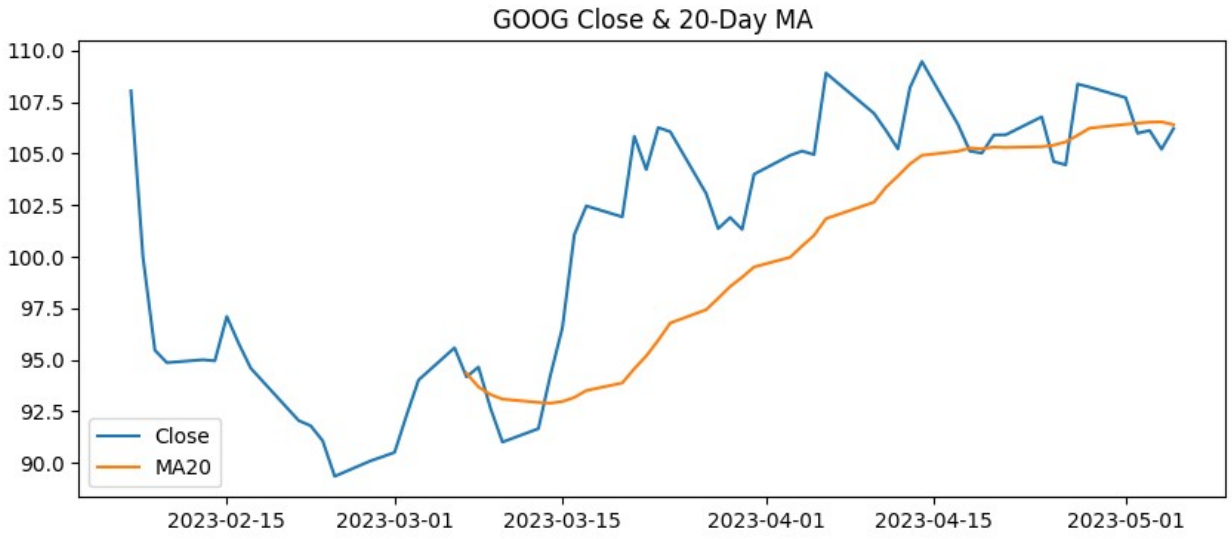
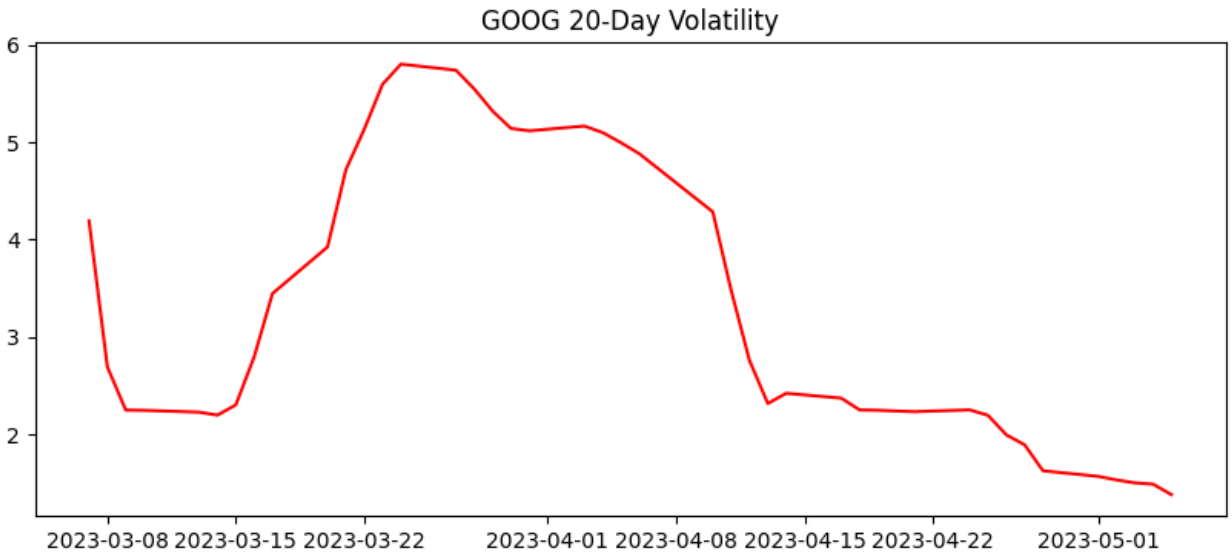
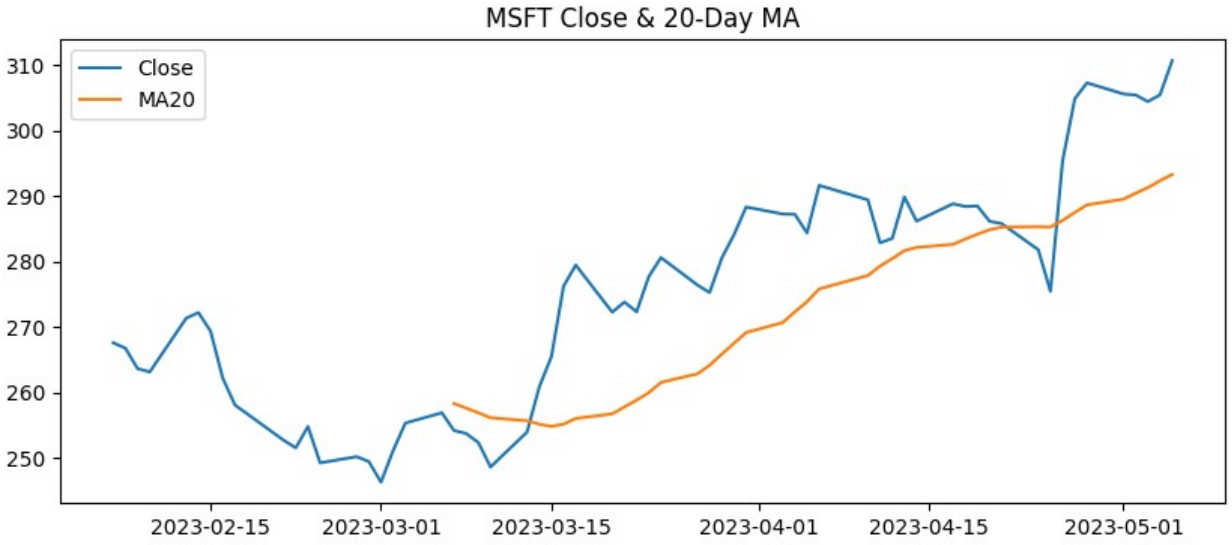
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

data = pd.read_csv('/content/drive/MyDrive/stocks.csv')
data['Date'] = pd.to_datetime(data['Date'])
data.sort_values(['Ticker','Date'], inplace=True)

tickers = data['Ticker'].unique()

for ticker in tickers:
    df = data[data['Ticker'] == ticker].set_index('Date')
    df['MA20'] = df['Close'].rolling(20).mean()
    df['Volatility'] = df['Close'].rolling(20).std()
    plt.figure(figsize=(10,4))
    plt.plot(df.index, df['Close'])
    plt.plot(df.index, df['MA20'])
    plt.title(f'{ticker} Close & 20-Day MA')
    plt.legend(['Close','MA20'])
    plt.show()
    plt.figure(figsize=(10,4))
    plt.plot(df.index, df['Volatility'], color='red')
    plt.title(f'{ticker} 20-Day Volatility')
    plt.show()
```

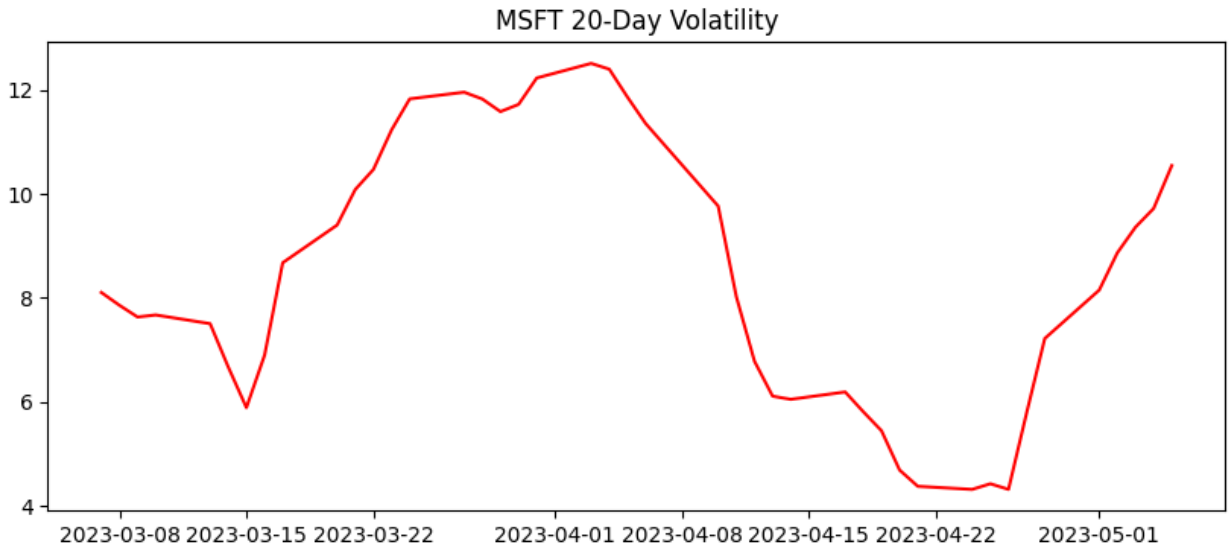


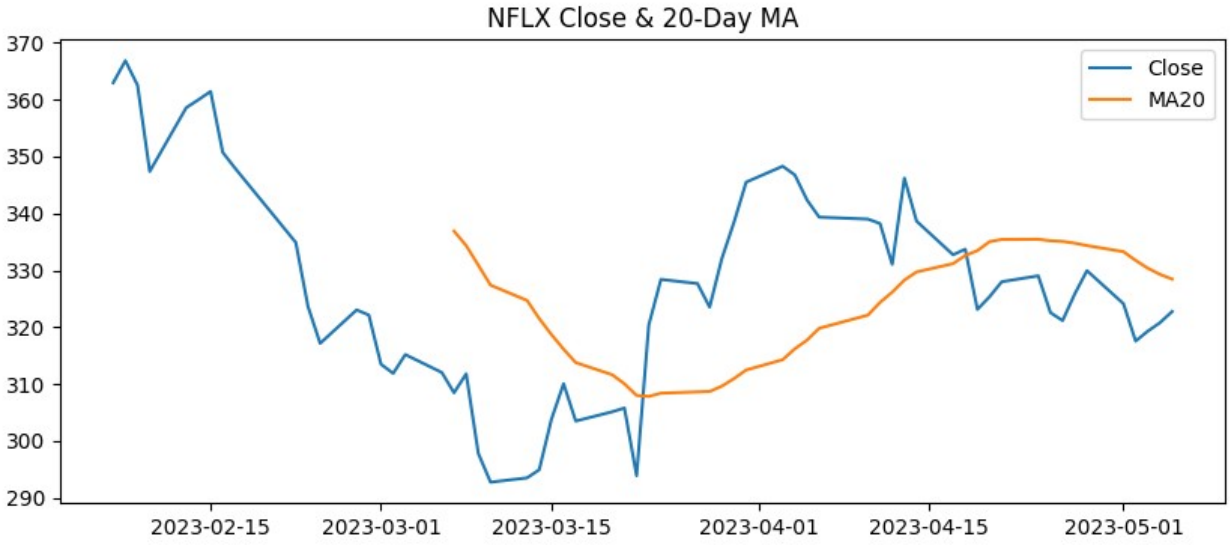

AAPL 20-Day Volatility
5.0
4.5
4.0
3.5
3.0
2.5
2.0
2023-03-08
2023-03-15
2023-03-22
2023-04-01
2023-04-08
2023-04-15
2023-04-22
2023-05-01

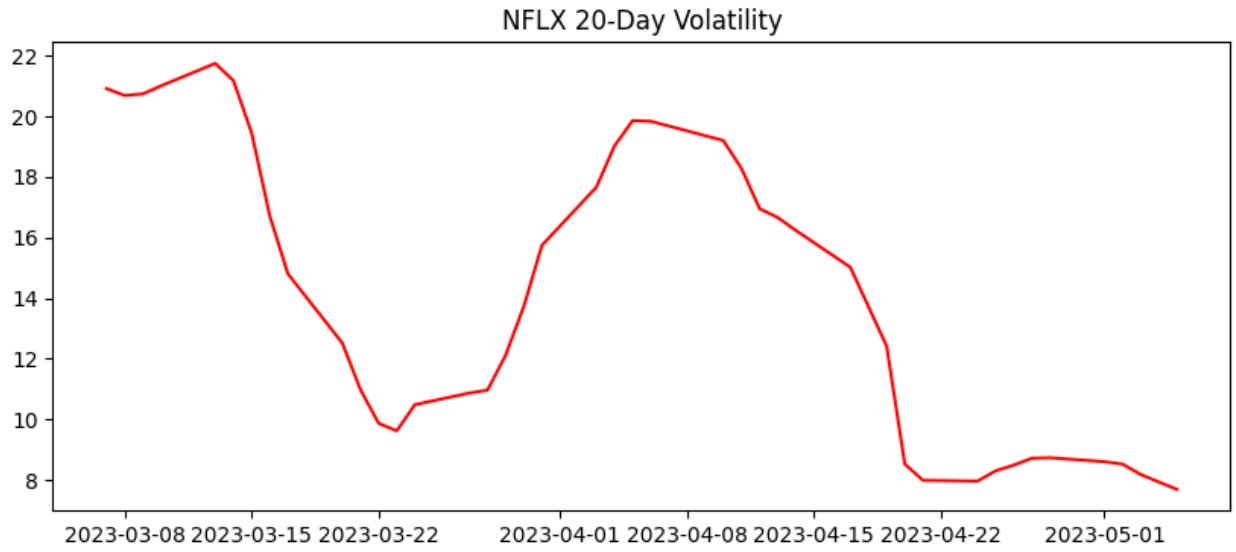

GOOG Close & 20-Day MA
110.0
107.5
105.0
102.5
100.0
97.5
95.0
92.5
90.0
Close
MA20
2023-02-15
2023-03-01
2023-03-15
2023-04-01
2023-04-15
2023-05-01


GOOG 20-Day Volatility
6
5
4
3
2
2023-03-08
2023-03-15
2023-03-22
2023-04-01
2023-04-08
2023-04-15
2023-04-22
2023-05-01

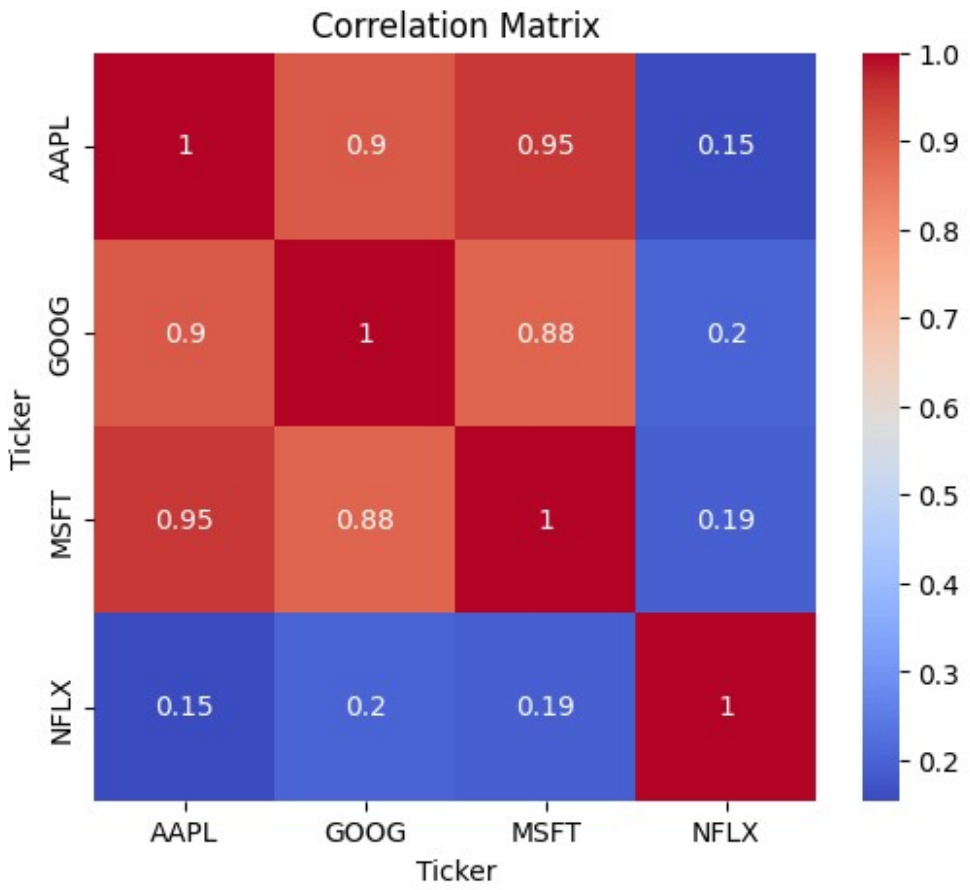

MSFT Close & 20-Day MA
Close
MA20
310
300
290
280
270
260
250
2023-02-15
2023-03-01
2023-03-15
2023-04-01
2023-04-15
2023-05-01


MSFT 20-Day Volatility
12
10
8
6
4
2023-03-08
2023-03-15
2023-03-22
2023-04-01
2023-04-08
2023-04-15
2023-04-22
2023-05-01


NFLX Close & 20-Day MA
Close
MA20
370
360
350
340
330
320
310
300
290
2023-02-15
2023-03-01
2023-03-15
2023-04-01
2023-04-15
2023-05-01

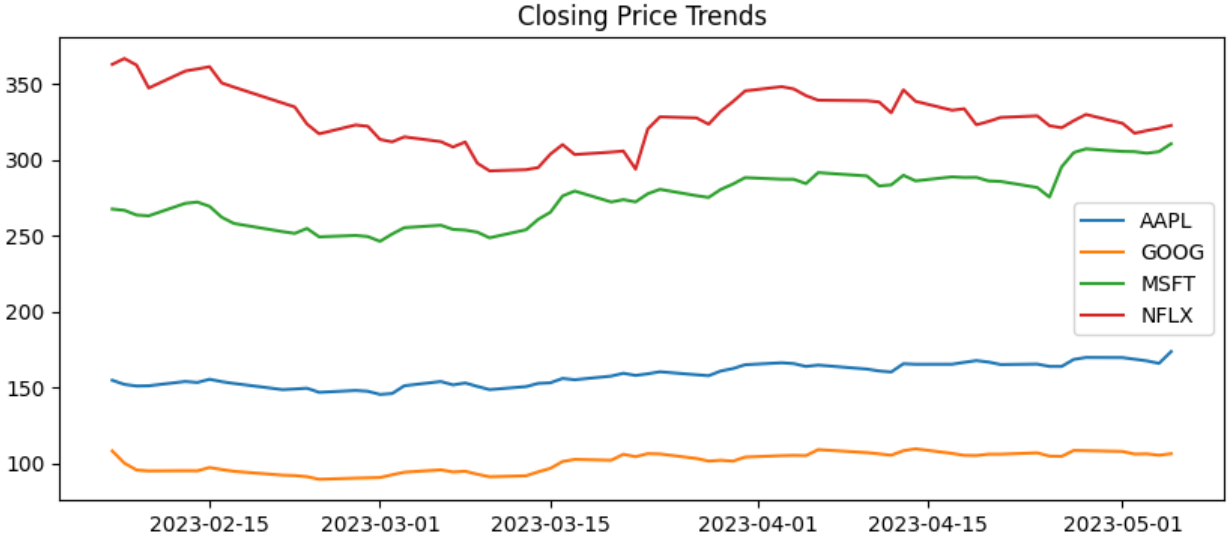



```
pivot = data.pivot(index='Date', columns='Ticker', values='Close')
corr = pivot.corr()
plt.figure(figsize=(6,5))
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```



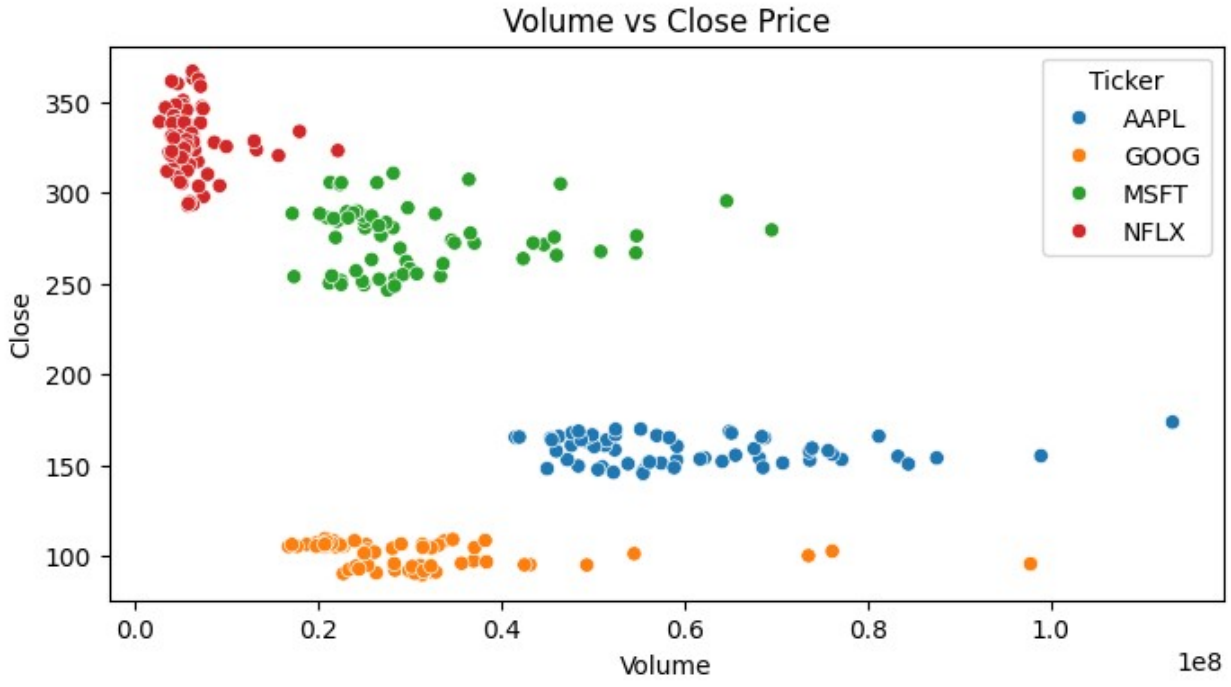

```
plt.figure(figsize=(8,4))
sns.histplot(data['Close'], kde=True)
plt.title('Closing Price Distribution')
plt.show()
```


```
plt.figure(figsize=(10,4))
for t in tickers:
    subset = data[data['Ticker'] == t]
    plt.plot(subset['Date'], subset['Close'], label=t)
plt.title('Closing Price Trends')
plt.legend()
plt.show()
```

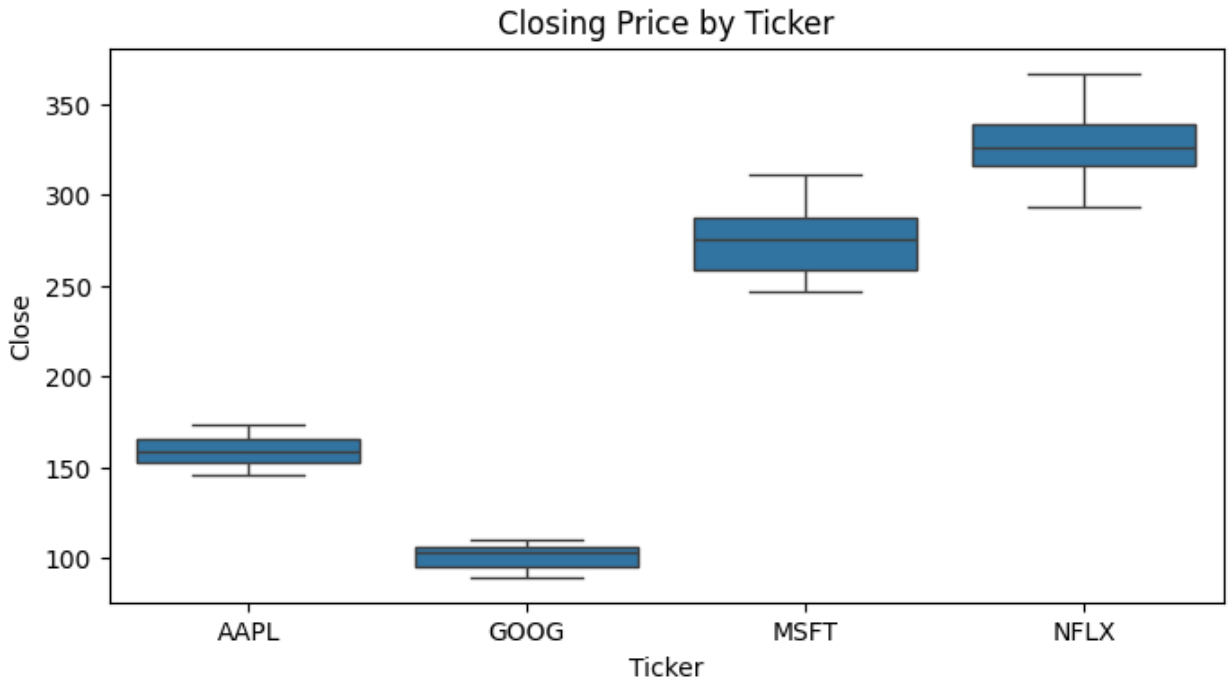



```
plt.figure(figsize=(8,4))
sns.scatterplot(data=data, x='Volume', y='Close', hue='Ticker')
```

```
plt.title('Volume vs Close Price')
plt.show()
```




```
plt.figure(figsize=(8,4))
sns.boxplot(data=data, x='Ticker', y='Close')
plt.title('Closing Price by Ticker')
plt.show()
```




```
df_aapl = data[data['Ticker'] == 'AAPL']
X = df_aapl[['Open','High','Low','Volume']]
y = df_aapl['Close']
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
model = LinearRegression().fit(X_train, y_train)
y_pred = model.predict(X_test)
print('AAPL MSE:', mean_squared_error(y_test, y_pred))
print('AAPL R²:', r2_score(y_test, y_pred))

AAPL MSE: 1.089260745266554
AAPL R²: 0.9854542039483511
```