Financial Econometrics Presentation

Analyzing Stocks

Exploring the world of investments and
financial decision making

6448108029 Puwit Sila
6548059029 Nithiwat Dararaj
6701059329 Trinita Sitorus

TEAM SPADES

Contents

- Overview of Stocks We Use
- MA
- RSI
- Bollinger Bands
- Cointegration
- Mean Reversion

Overview of Stocks Used

NVDA

AAPL

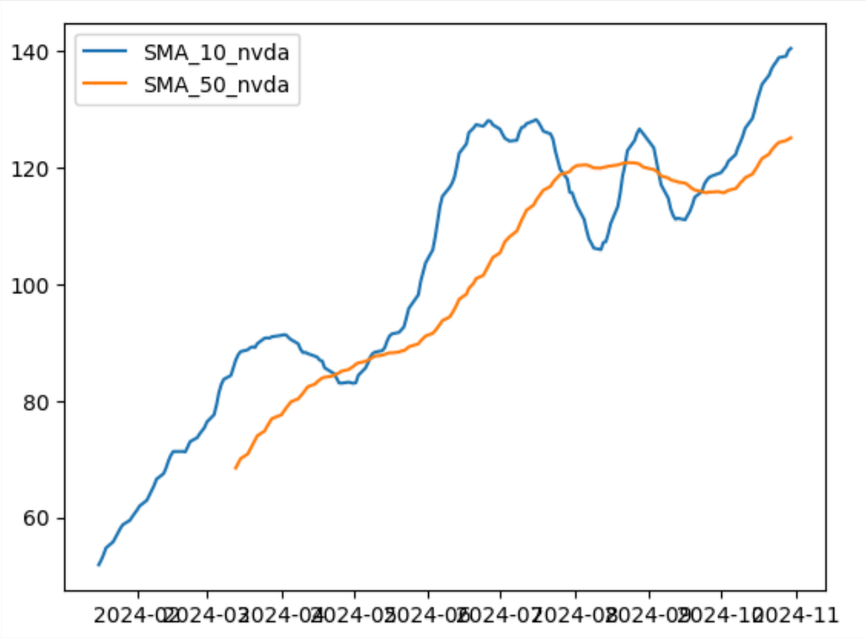
MA STRATEGY

NVDA

EXAMPLE CODE

```
nvda['sma_50_nvda']=nvda['Adj Close'].rolling(window=50).mean()
nvda['sma_10_nvda']=nvda['Adj Close'].rolling(window=10).mean()
import matplotlib.pyplot as plt
plt.plot(nvda['sma_10_nvda'], label='SMA_10_nvda')
plt.plot(nvda['sma_50_nvda'], label='SMA_50_nvda')
plt.legend()
plt.show()
```

VISUALIZATION OF NVDA MA10/50

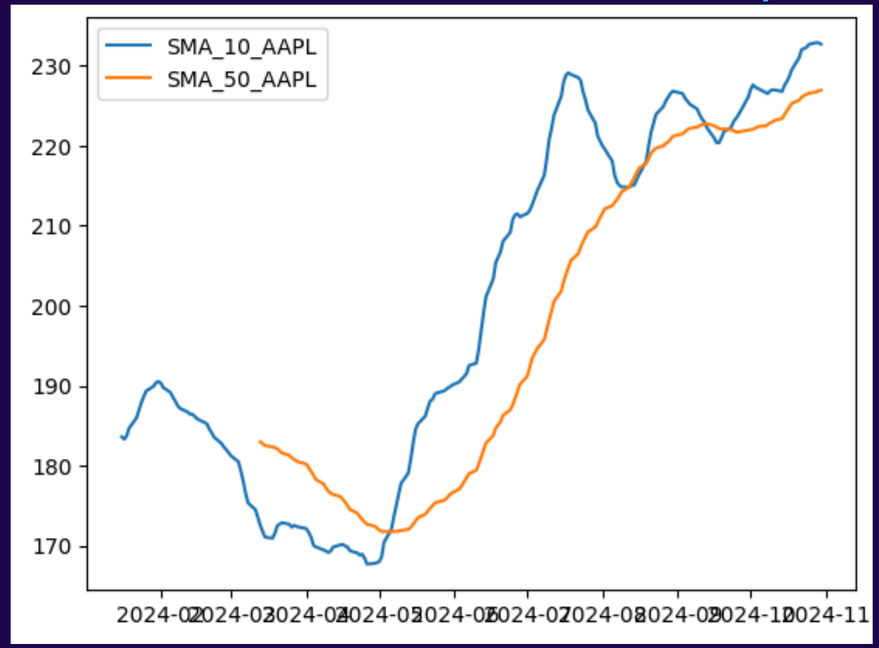


AAPL

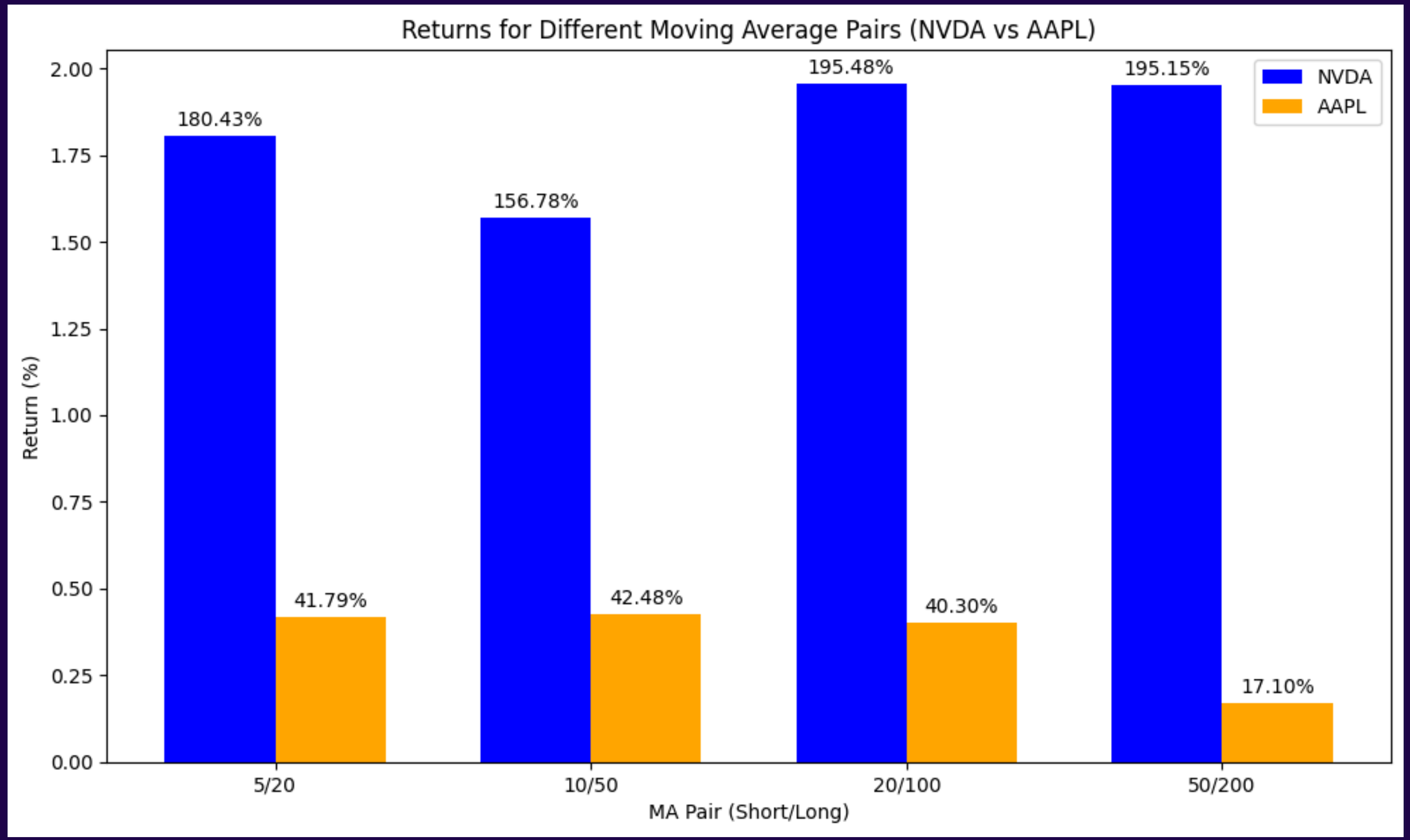
EXAMPLE CODE

```
aapl = yf.download('AAPL', start='2024-1-1', end='2024-10-31')
aapl['sma_50_AAPL']=aapl['Adj Close'].rolling(window=50).mean()
aapl['sma_10_AAPL']=aapl['Adj Close'].rolling(window=10).mean()
import matplotlib.pyplot as plt
plt.plot(aapl['sma_10_AAPL'], label='SMA_10_AAPL')
plt.plot(aapl['sma_50_AAPL'], label='SMA_50_AAPL')
plt.legend()
plt.show()
```

VISUALIZATION OF AAPL MA10/50



MA Optimazation

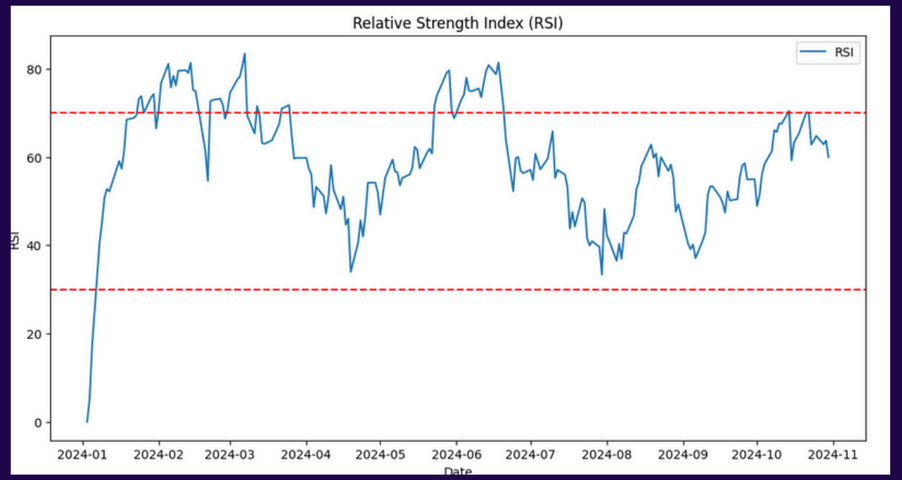


RSI STRATEGY

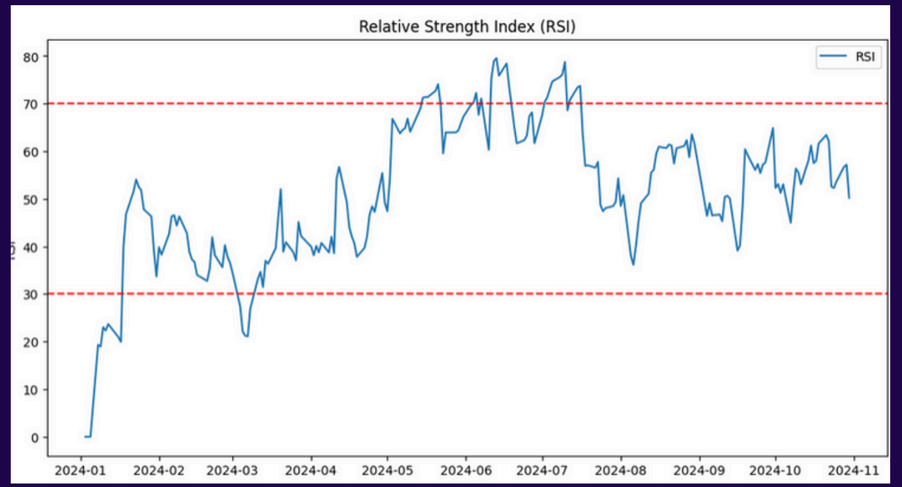
OUR CODE

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

def RSI(bbca, period=14):
    delta = bbca["Adj Close"].diff(1)
    delta = delta.dropna()
    up = delta.copy()
    down = delta.copy()
    up[up < 0] = 0
    down[down > 0] = 0

    # Calculate the EWMA
    roll_up1 = up.ewm(com=period - 1, adjust=False).mean()
    roll_down1 = down.ewm(com=period - 1, adjust=False).mean().abs()

    # Calculate the RS
    RS = roll_up1 / roll_down1

    # Calculate the RSI
    RSI = 100.0 - (100.0 / (1.0 + RS))

    bbca['RSI'] = RSI
    return bbca

# Example usage:
# Assuming you have a DataFrame 'df' with 'Adj Close' column
bbca = RSI(bbca, period=14)

# Plot the RSI
plt.figure(figsize=(12, 6))
plt.plot(bbca.index, bbca['RSI'], label='RSI')
plt.axhline(30, color='r', linestyle='--')
plt.axhline(70, color='r', linestyle='--')
plt.xlabel('Date')
plt.ylabel('RSI')
plt.title('Relative Strength Index (RSI)')
plt.legend()
plt.show()
```

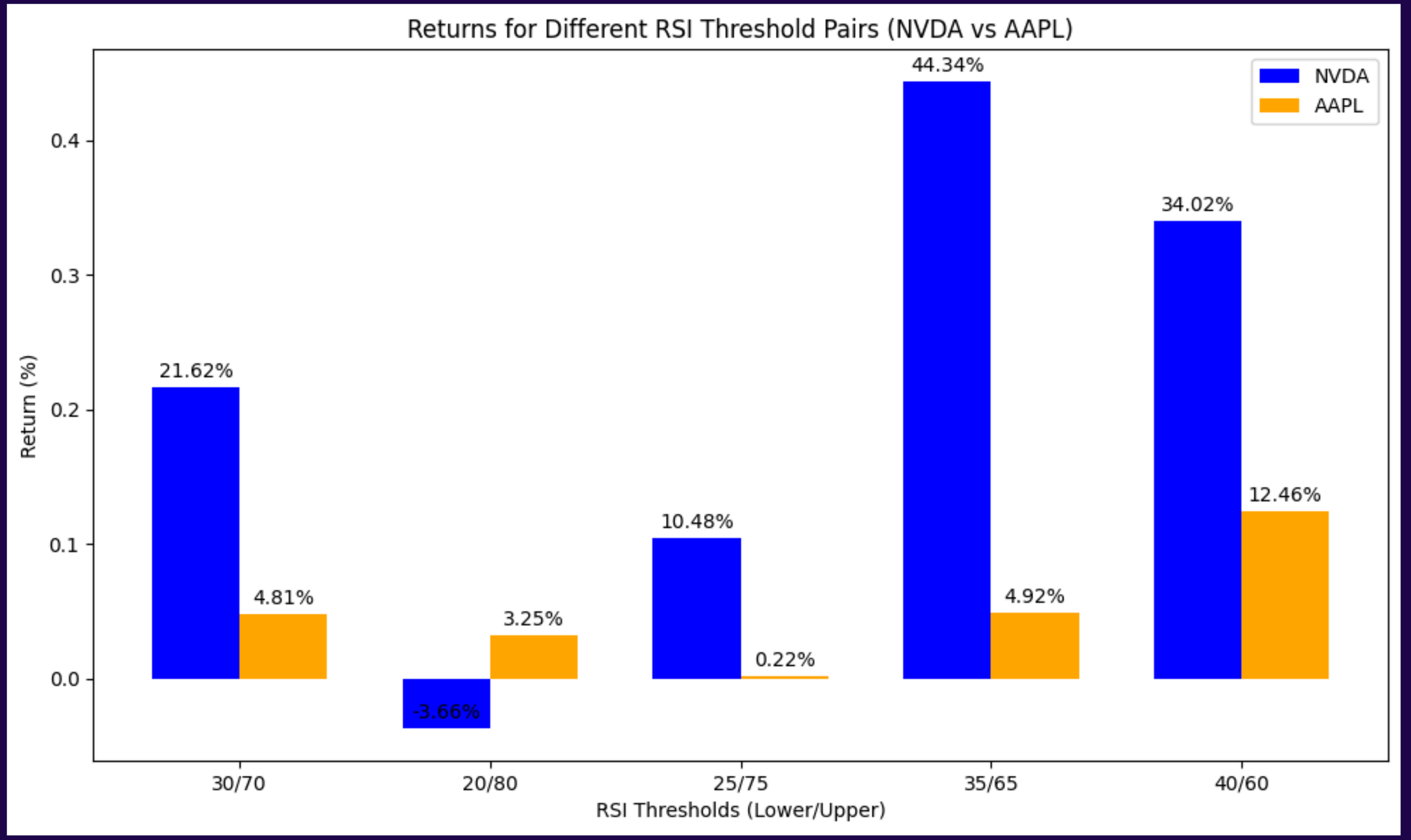
NVDA



AAPL



RSI Optimization

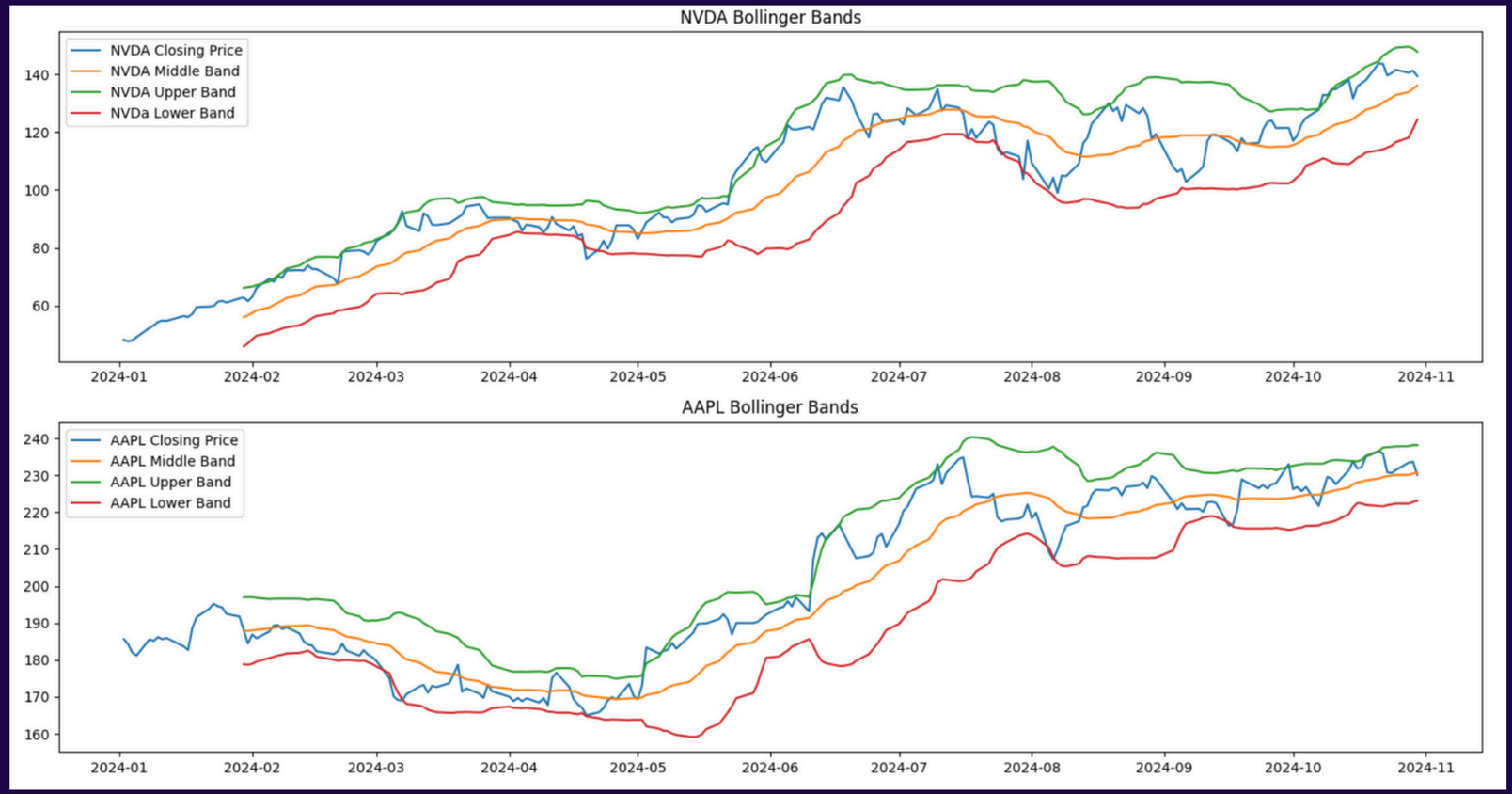


BOLLINGER BANDS STRATEGY

OUR CODE

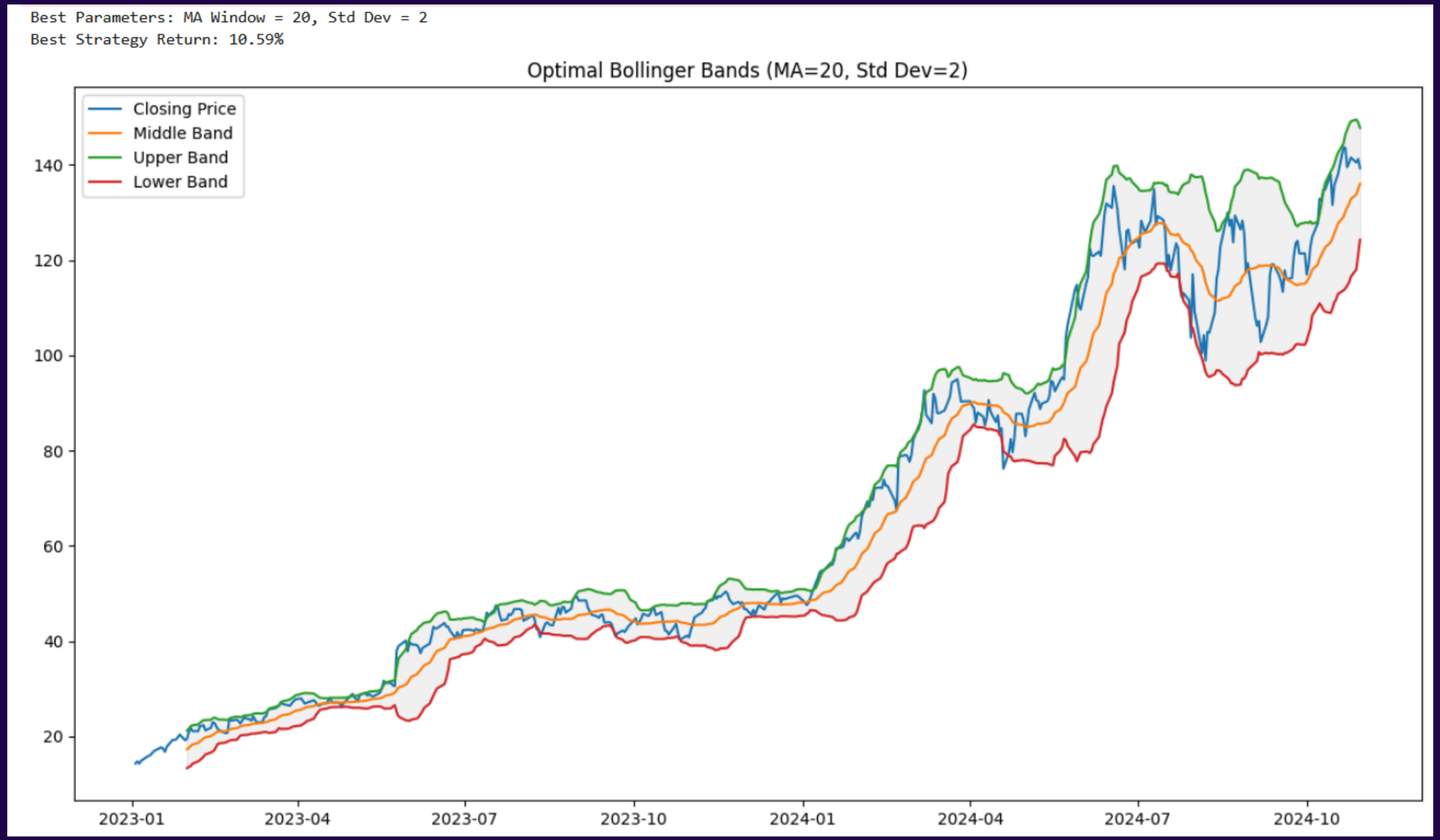
```python
import yfinance as yf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Fetch the data
ticker1 = "NVDA"
ticker2 = "AAPL"
start_date = "2024-01-01"
end_date = "2024-10-31"

data1 = yf.download(ticker1, start=start_date, end=end_date)
data2 = yf.download(ticker2, start=start_date, end=end_date)

# Function to calculate Bollinger Bands
def bollinger_bands(df, window=20, std_dev=2):
    rolling_mean = df['Close'].rolling(window).mean()
    rolling_std = df['Close'].rolling(window).std()

    df['Bollinger_Mid'] = rolling_mean
    df['Bollinger_Upper'] = rolling_mean + (rolling_std * std_dev)
    df['Bollinger_Lower'] = rolling_mean - (rolling_std * std_dev)

    return df

# Calculate Bollinger Bands for BBRI and BBCA
data1 = bollinger_bands(data1)
data2 = bollinger_bands(data2)

# Visualize the Bollinger Bands
plt.figure(figsize=(15, 8))

# Plot NVDA
plt.subplot(2, 1, 1)
plt.plot(data1.index, data1['Close'], label='NVDA Closing Price')
plt.plot(data1.index, data1['Bollinger_Mid'], label='NVDA Middle Band')
plt.plot(data1.index, data1['Bollinger_Upper'], label='NVDA Upper Band')
plt.plot(data1.index, data1['Bollinger_Lower'], label='NVDa Lower Band')
plt.title('NVDA Bollinger Bands')
plt.legend()

# Plot AAPL
plt.subplot(2, 1, 2)
plt.plot(data2.index, data2['Close'], label='AAPL Closing Price')
plt.plot(data2.index, data2['Bollinger_Mid'], label='AAPL Middle Band')
plt.plot(data2.index, data2['Bollinger_Upper'], label='AAPL Upper Band')
plt.plot(data2.index, data2['Bollinger_Lower'], label='AAPL Lower Band')
plt.title('AAPL Bollinger Bands')
plt.legend()

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

OUR CHART



Bollinger Bands Return

Best Parameters: MA Window = 20, Std Dev = 2
Best Strategy Return: 10.59%



COINTEGRATION STRATEGY

OUR CODE

```
import yfinance as yf
import pandas as pd
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt
```

```
import yfinance as yf
bbri = yf.download('NVDA', start='2024-1-1', end='2024-10-31')
bbca = yf.download('AAPL', start='2024-1-1', end='2024-10-31')
```

```
[**********100%**********]  1 of 1 complete
[**********100%**********]  1 of 1 complete
```

```
# Check for cointegration
result = sm.tsa.stattools.coint(bbri['Adj Close'], bbca['Adj Close
print(result)
```

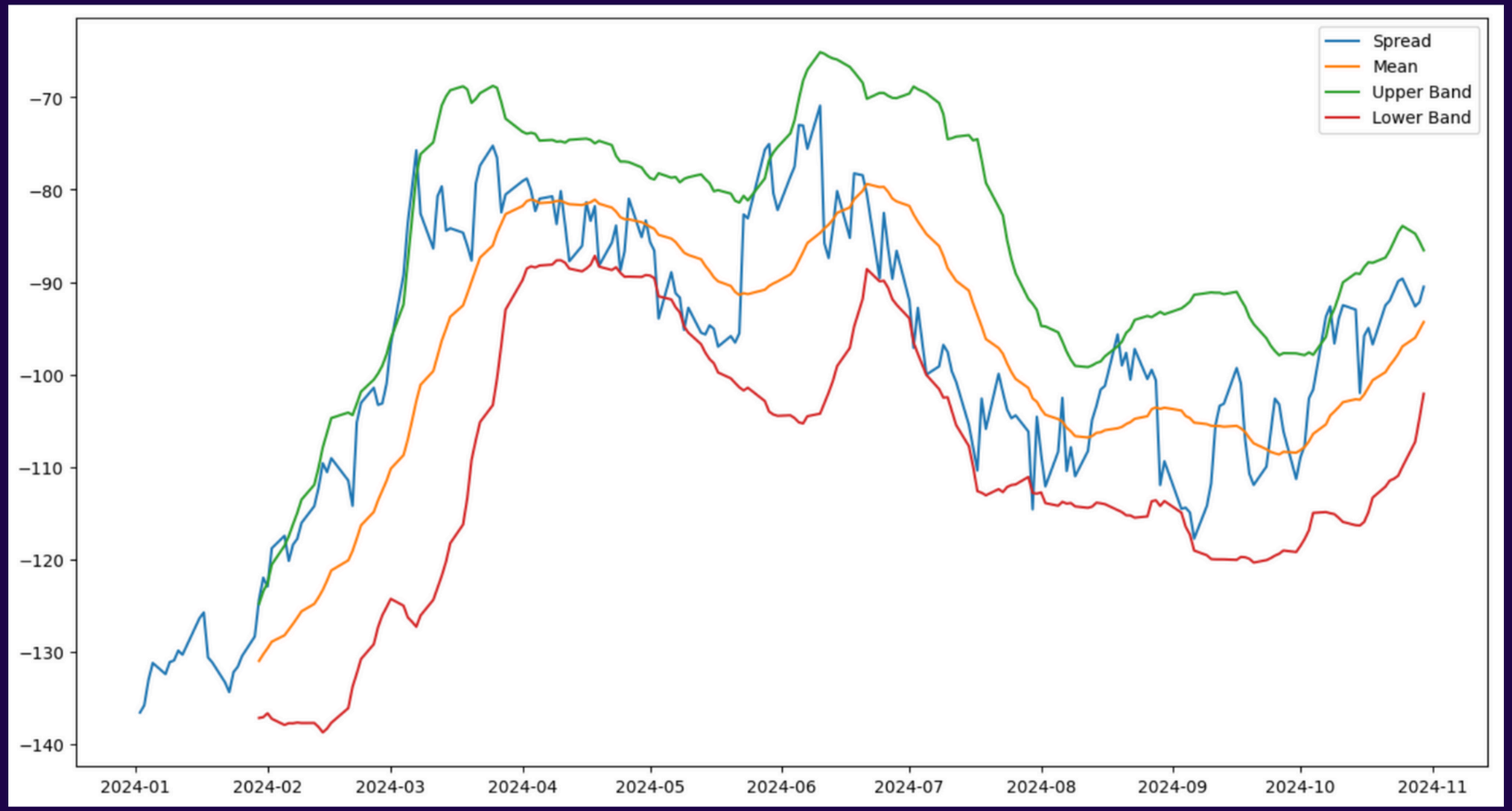
```
(np.float64(-2.350848491683655), np.float64(0.34847303289756365), a
```

```
# Calculate the spread
bbri['spread'] = bbri['Adj Close'] - bbca['Adj Close']
```

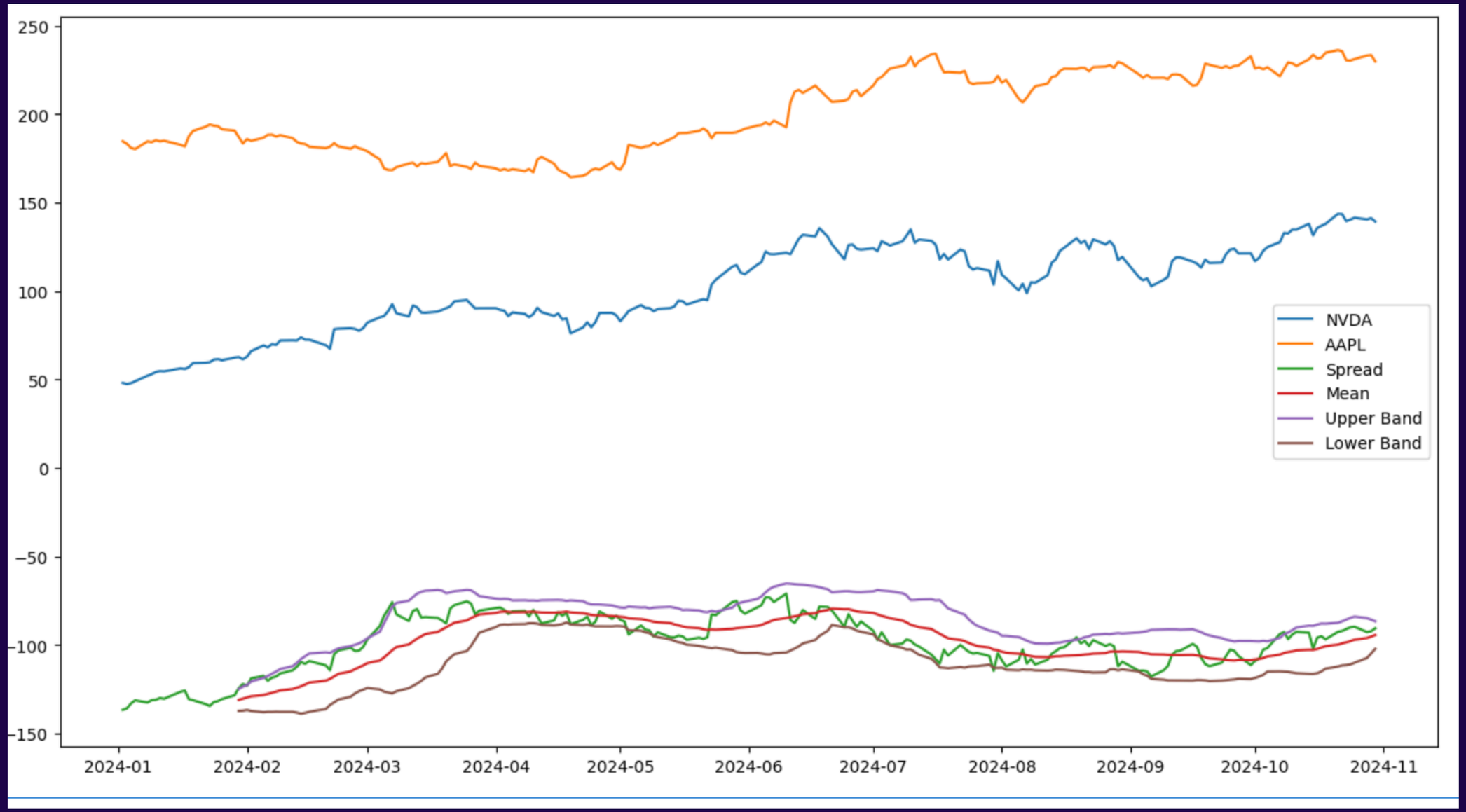
```
# Calculate Z-score
bbri['z_score'] = (bbri['spread'] - bbri['spread'].rolling(window=2

# Define trading signals
bbri['position'] = np.where(bbri['z_score'] > 2, -1, np.where(bbri[
```

```
plt.figure(figsize=(15, 8))
plt.plot(bbri.index, bbri['spread'], label='Spread')
plt.plot(bbri.index, bbri['spread'].rolling(window=20).mean(), labe
plt.plot(bbri.index, bbri['spread'].rolling(window=20).mean() + 2*b
plt.plot(bbri.index, bbri['spread'].rolling(window=20).mean() - 2*b
plt.legend()
plt.show()
```

OUR CHART



COINTEGRATION STRATEGY cont.



Returns From Trading Strategies in 2 Years

Test	NVIDIA	Apple
MA (20,100)	195%	40%
RSI (40,60)	34%	12,4%
Bollinger (20,2)	10,59% (joint)	
Cointegration		

Mean Reversion Strategy

Mean Reversion Strategy

I. Prepare and cleanse data

II. Use mean-reverting tests to check data

→ Hurst / Adfuller / Variance Ratio

III. Calculate half-life of stocks

IV. Conduct a cointegration test

→ It is used to experiment what are the best duos of portfolio construction

Mean Reversion Strategy

V Combine two stocks in a portfolio

VI Conduct the following tests

VII Calculate the hedge ratio

VIII Visualize the results

Hurst / Adfuller / Variance Ratio

Mean Reversion Strategy

IX Incorporate Z-score index and stop-loss

X Visualize the results

XI Compare the performances

XII Optimize the results

Conditions:

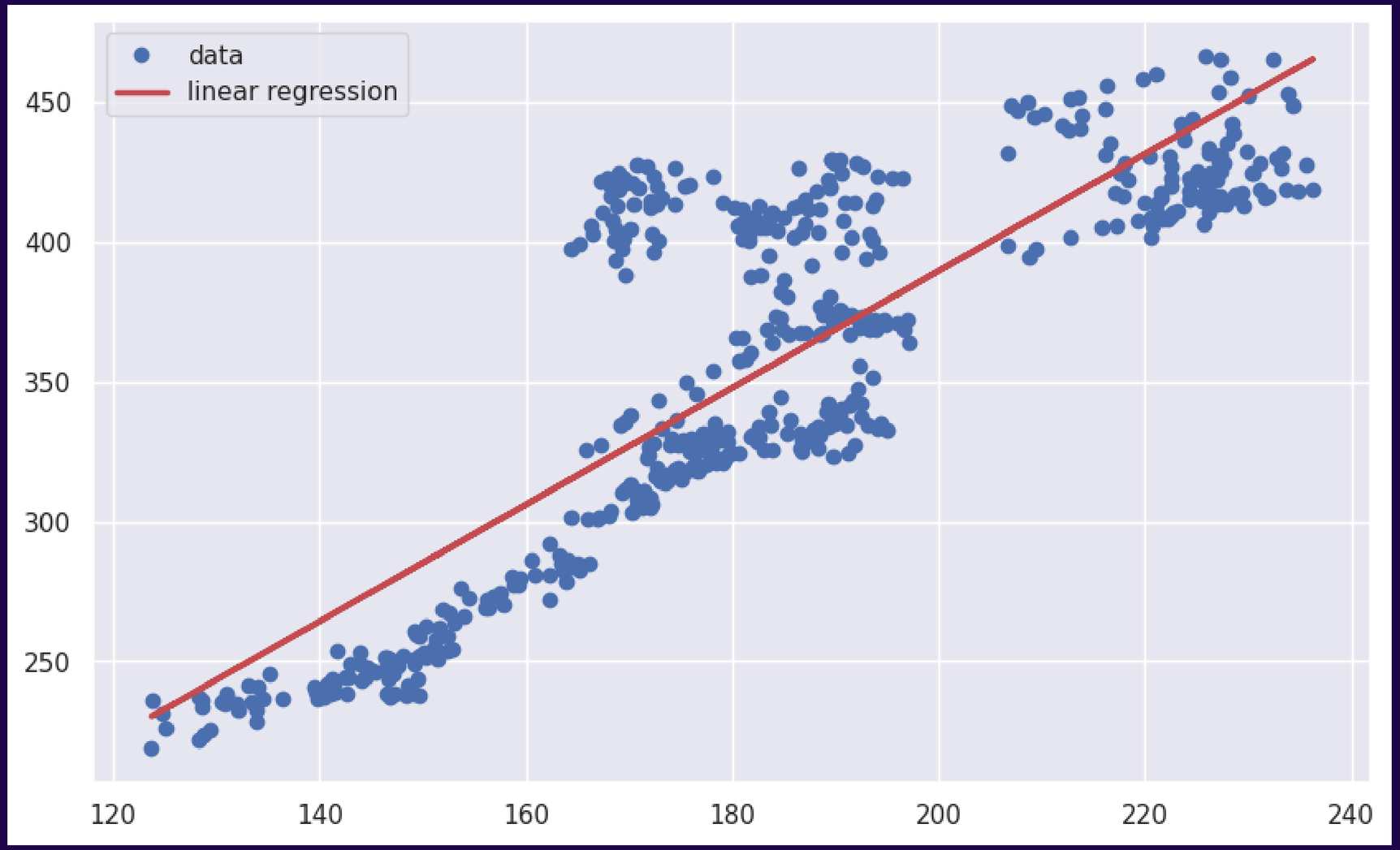
- If price-loss percentage < stop-loss: close the position if loss percentage > stop-loss threshold
- If Zscore < -threshold & no long-position => buy
- If Zscore > threshold & no short-position => sellll

Maximize Equity + Minimize Maximum Drawdown + Minimize Volatility

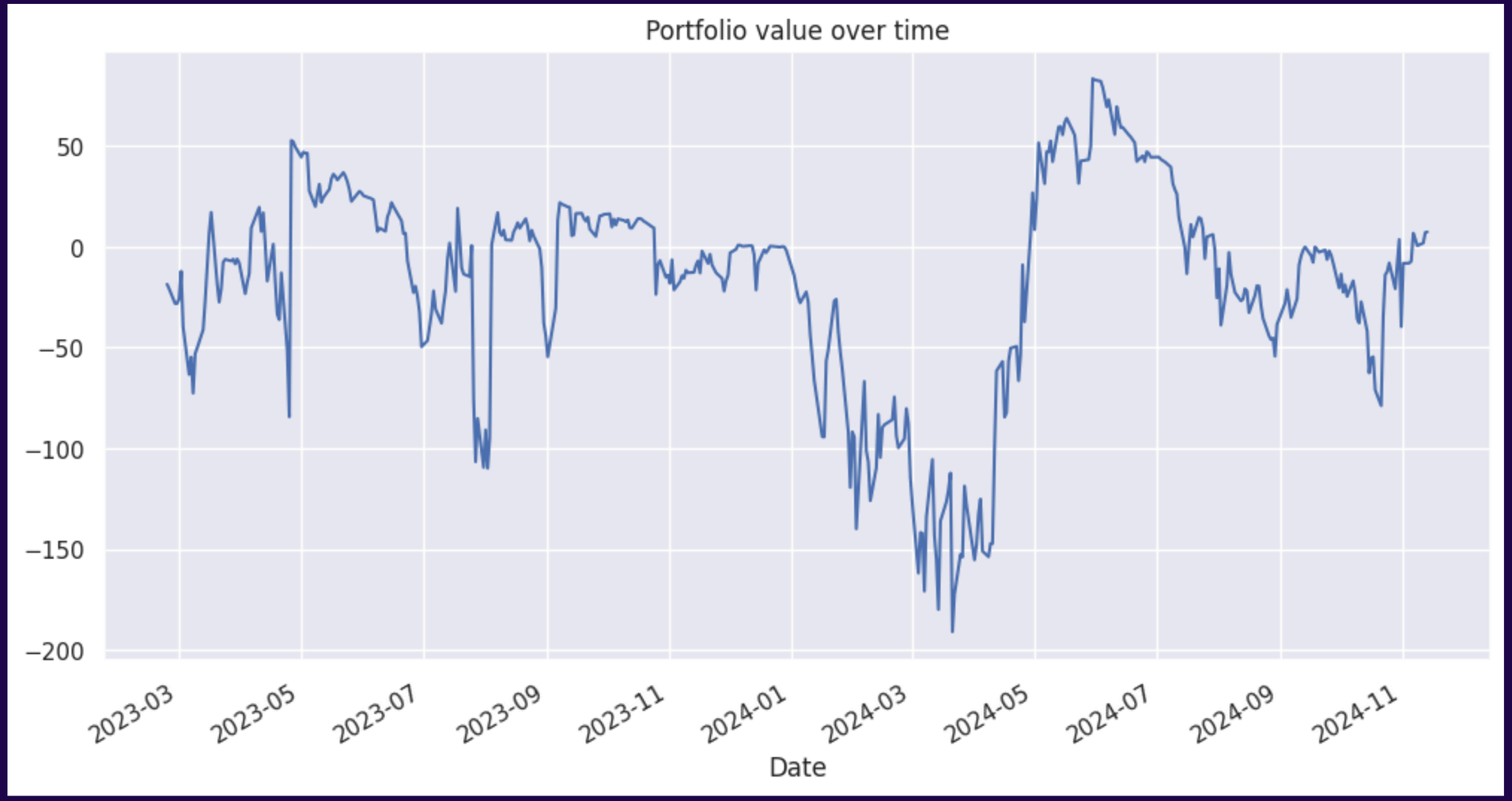
Experimentation of Mean-Reverting Indicators

Test	NVIDIA	Apple	Microsoft
Hurst Exponent	0.718513	0.929772	0.784275
Variance Ratio	0.184508	0.181548	0.678936
Half-life	-788	181	153

Cointegration of Two stocks & Portfolio Construction



Porfolio Value



Daily profit & loss

Backtesting

Given cash = 10,000

interval = 2 years

Start	2022-11-14 00:00:00+00:00
End	2024-11-13 00:00:00+00:00
Duration	730 days 00:00:00
Exposure Time [%]	4.572565
Equity Final [$]	80538.86331
Equity Peak [$]	83574.661864
Return [%]	705.388633
Buy & Hold Return [%]	-76.250713
Return (Ann.) [%]	184.382816
Volatility (Ann.) [%]	2904.461708

Apprendrix

Google Colab

google.com

arch

ARCH for Python

PyPI

Bokeh

Bokeh is a Python-based visualization library, capable of building plots from simple charts to interactive dashboards.

bokeh.org /

edgetrader/**mean-reversion-strategy**

Mean Reversion Trading Strategy

1 Contributor | 0 Issues | 26 Stars | 9 Forks

mean-reversion-strategy/notebook/mean-reversion-strategy.ipynb at...

Mean Reversion Trading Strategy. Contribute to edgetrader/mean-reversion-strategy development by creating an account on GitHub.

GitHub

Summary

Thank you