Reported by Shiyuan Liang

Research About Equity Market Trends

based on AAPL, MSFT, GOOGL etc.

What I have done:

By analyzing the historical data of 10 stocks from 2019 to 2024, I calculated the long and short term moving averages, RSI, MACD and other key indicators, and analyzed the buying and selling points of different stocks. At the same time, I used statistical methods to obtain the market volatility of different stocks, the correlation coefficients between different stocks, and the correlation between each stock and external economic factors.Finally, I designed a dashboard to facilitate more convenient data visualization in the future.

Analysis steps

Through the basic steps of data analysis and quantitative trading, I divided my analysis process into the following five steps:

01. Data Acquisition

02. Data Cleaning and Preparation

03. Data Analysis

04. Data Visualization

05. Interpretation of Results

01. Data Acquisition

Yahoo Finance

Selected 10 stocks:
'AAPL', 'MSFT', 'GOOGL', 'AMZN', 'TSLA', 'JNJ', 'XOM', 'JPM', 'PG', 'NVDA'
from **2019-01-01** to **2024-01-01**
These stocks from different sectors to provide a market overview

The database: **Yahoo Finance**

Library in Python: **yfinance**

There are also some other libraries I will use, such as **pandas, matplotlib.pyplot, numpy, dash, seaborn**

```python
import yfinance as yf
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import dash
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input, Output
import pandas_datareader.data as web
from datetime import datetime
import seaborn as sns

# Predefined list of stocks
stocks = ['AAPL', 'MSFT', 'GOOGL', 'AMZN', 'TSLA', 'JNJ', 'XOM', 'JPM', 'PG',

# Define the time period
start_date = '2019-01-01'
end_date = '2024-01-01'

# Fetch historical stock prices and trading volumes
data={}

for stock in stocks:
  ticker = yf.Ticker(stock)
  hist = ticker.history(start=start_date, end=end_date)
  data[stock] = hist

# Combine all data into a single DataFrame with multi-level columns
combined_data = pd.concat(data, axis=1)
combined_data.index = pd.to_datetime(combined_data.index).date
```

02. Data Cleaning and Preparation

Solve NaN and Normolization

In this step, I clean and normalize the data.

First, check the missing data and use the forward filling method to replace the missing data.

After that, I normalized the data using statistical knowledge, (X-mean(X)) / std(X), to ensure the consistency of the data.

This is python code:

```
# Check missing values
combined_data.isnull().sum()

# Fill missing values
fillna_data = combined_data.fillna(method='ffill')

# Normalize data
normalized_data = (combined_data - combined_data.mean()) / combined_data.std()
normalized_data.head()
```

	AAPL							MSFT			...
	Open	High	Low	Close	Volume	Dividends	Stock Splits	Open	High	Low	...
2019-01-02	-1.800759	-1.793965	-1.792739	-1.787236	0.885147	-0.126609	-0.028194	-1.866970	-1.855295	-1.858962	...
2019-01-03	-1.856695	-1.860675	-1.856012	-1.867856	5.011481	-0.126609	-0.028194	-1.859860	-1.875290	-1.881669	...
2019-01-04	-1.853875	-1.846297	-1.846700	-1.836746	2.524915	-0.126609	-0.028194	-1.864772	-1.845555	-1.859093	...
2019-01-07	-1.832495	-1.844874	-1.835835	-1.838437	2.233774	-0.126609	-0.028194	-1.839952	-1.835814	-1.832340	...
2019-01-08	-1.828086	-1.829683	-1.822280	-1.823984	1.188172	-0.126609	-0.028194	-1.821854	-1.826842	-1.822814	...

This is part of the data after dealing with missing values and normalization

03. Data Analysis

Get MA, RSI and MACD

I calculate three important indicators:
MA:
I took a window period of **50 days** and **200 days** to calculate the **short-term and long-term** moving averages respectively, which are used to identify **trend**.

Std and Var:
I first calculated the daily **logarithmic return** of each stock using the closing price, and then calculated the std and var of each stock.

Then I used **20 days** as the window period and calculated the std and var of each stock in each window period.

This method can not only compare the volatility of different stocks as a whole, but also study the volatility of each stock.

RSI and MACD:
When calculating RSI, a **14-day** window is used, and the RSI of each day is calculated on a rolling basis.

When calculating MACD, **12 days** is used as the short-term window period, **26 days** is used as the long-term window period, and **9 days** is used as the signal window period.

Python Code For These Indicators

The code for MA

```
# Calculate MA
for stock in stocks:
  normalized_data[(f'{stock}','MA50')] = normalized_data[(f'{stock}','Close')].rolling(window=50).mean()
  normalized_data[(f'{stock}','MA200')] = normalized_data[(f'{stock}','Close')].rolling(window=200).mean()
```

The code for std and var

```
# Calculate standard deviation and variance
Close_data = normalized_data.xs('Close', level=1, axis=1)
Returns = np.log(Close_data/Close_data.shift(1))
std_close = Close_data.std()
var_close = Close_data.var()
std_return = Returns.std()
var_return = Returns.var()

# Calculate 20 days rolling std, var
std_20 = Close_data.rolling(window=20).std()
var_20 = Close_data.rolling(window=20).var()
```

Python Code For These Indicators

The code for RSI

```
#Implement technical indicators
def RSI(price, time_period=14):
  delta = price.diff()
  gain = delta.where(delta > 0, 0)
  loss = -delta.where(delta < 0, 0)
  # calculate average
  avg_gain = gain.rolling(window=time_period).mean()
  avg_loss = loss.rolling(window=time_period).mean()
  #calculate rs, rsi
  rs = avg_gain / avg_loss
  rsi = 100 - (100 / (1 + rs))
  return rsi
```

The code for MACD

```
def MACD(price, short_period=12, long_period=26, signal_period=9):
  # Get short and long term
  short_term = price.ewm(span=short_period, adjust=False).mean()
  long_term = price.ewm(span=long_period, adjust=False).mean()
  # Get MACD line
  macd_line = short_term - long_term
  # Get signal line
  signal_line = macd_line.ewm(span=signal_period, adjust=False).mean()
  # Get histogram
  histogrm = macd_line - signal_line
  return macd_line, signal_line, histogrm
```

04. Data Visualization

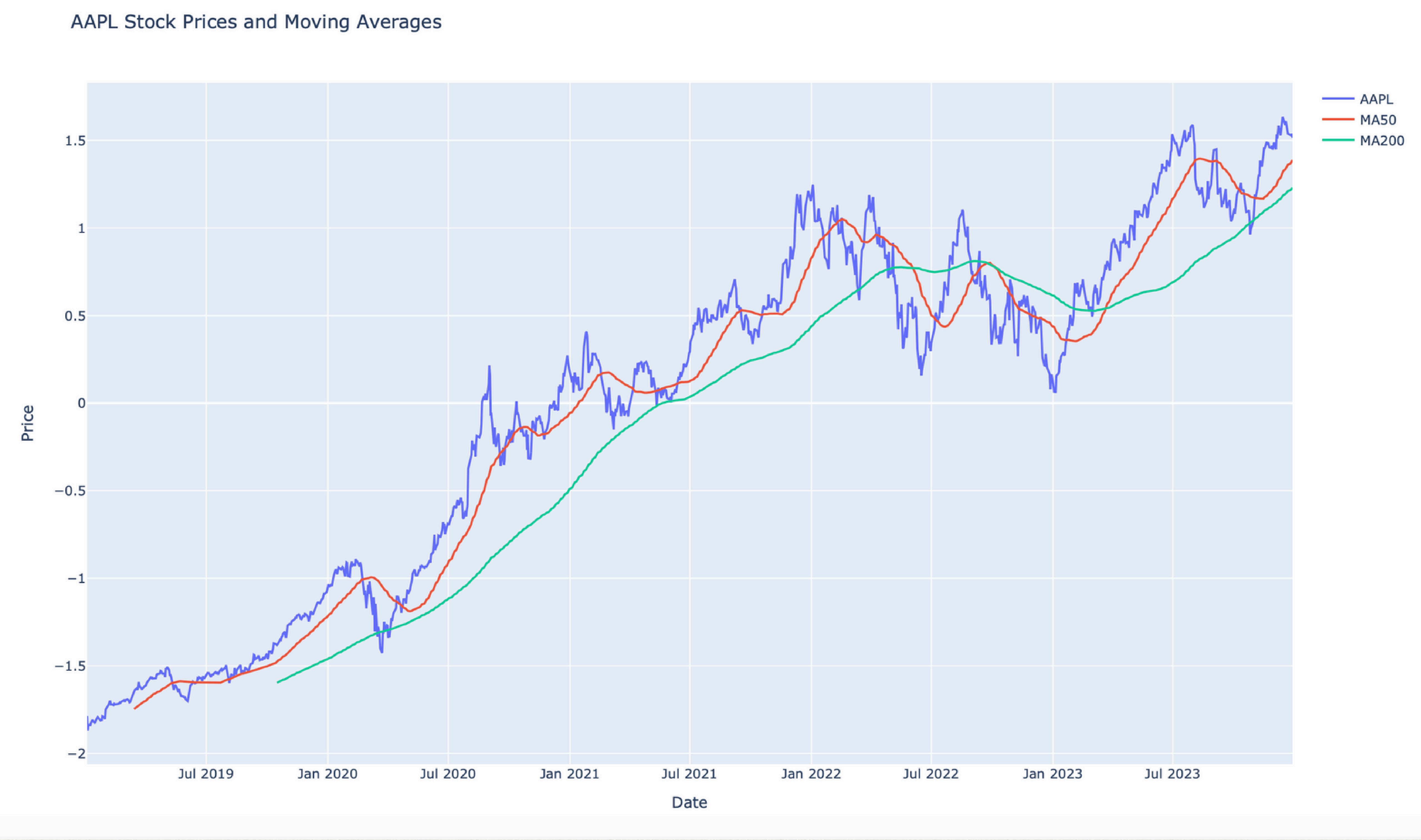
Visualization For Indicators

When visualizing data, I visualized the stock price, long-term and short-term moving averages, the number of stock transactions, and market volatility.

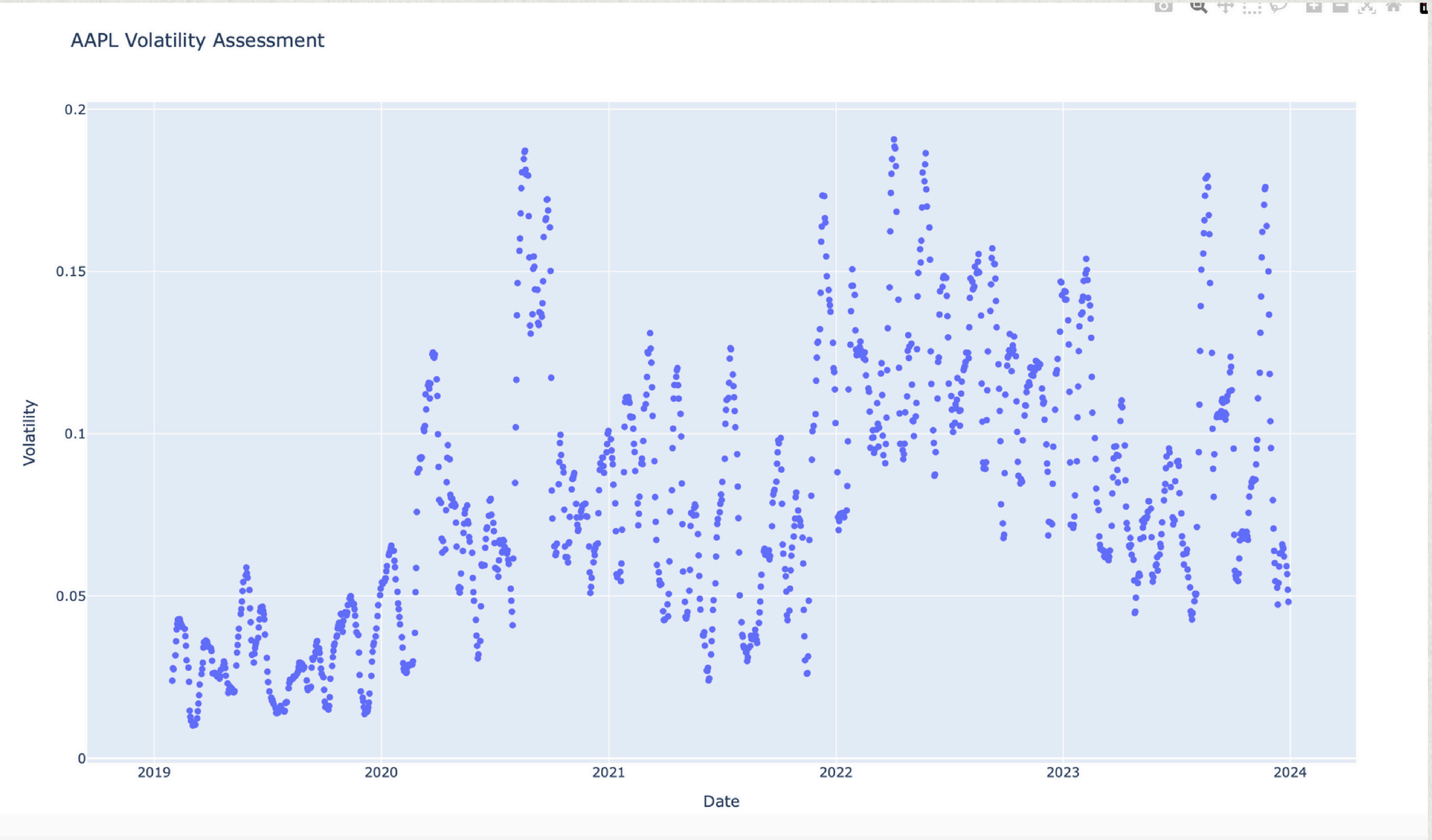
I also visualized its **RSI** and **MACD** to facilitate subsequent investment decisions.

Here I take the two stocks '**AAPL**' as examples to show the visualization

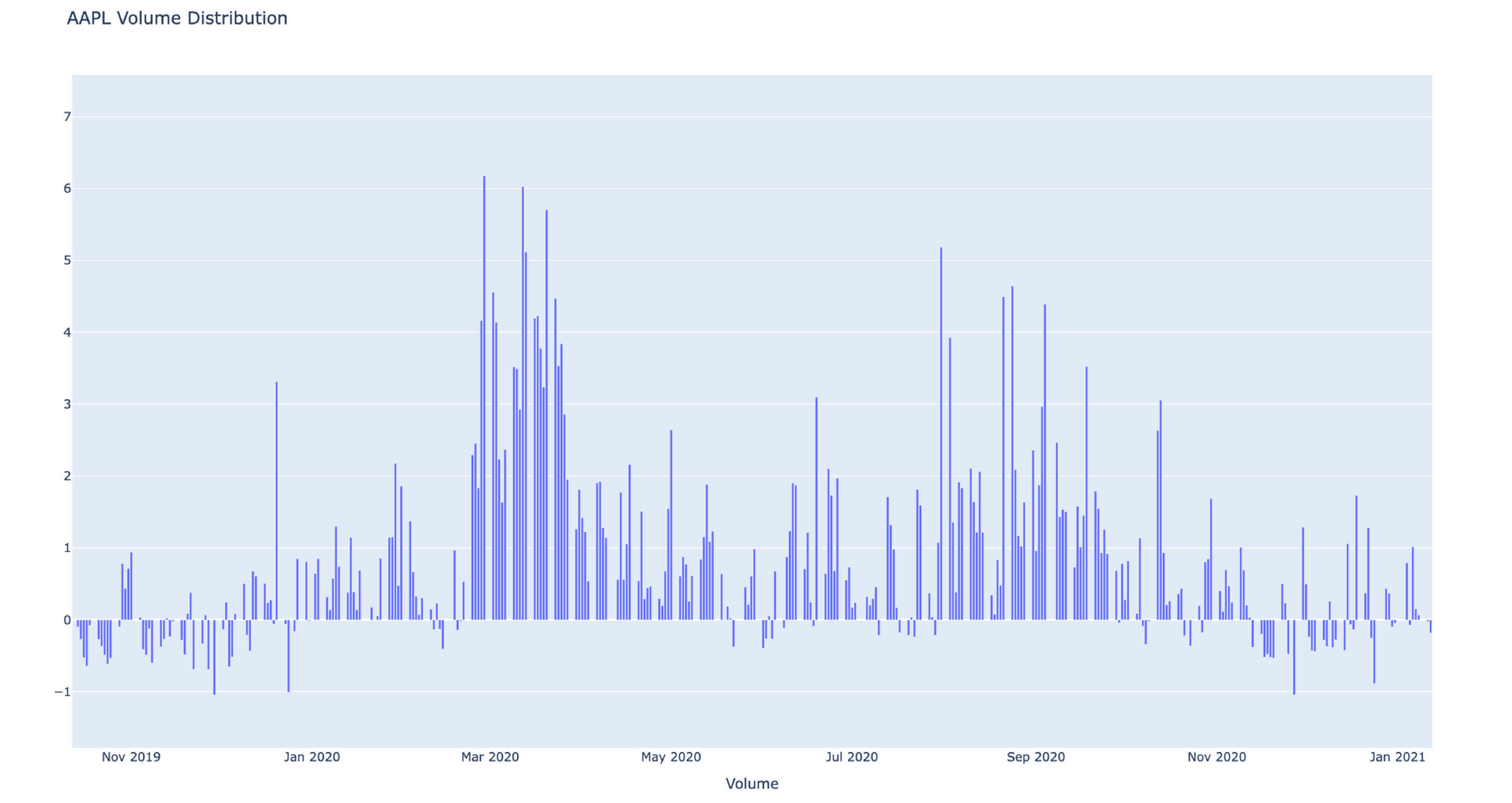
Visualization For Indicators - AAPL



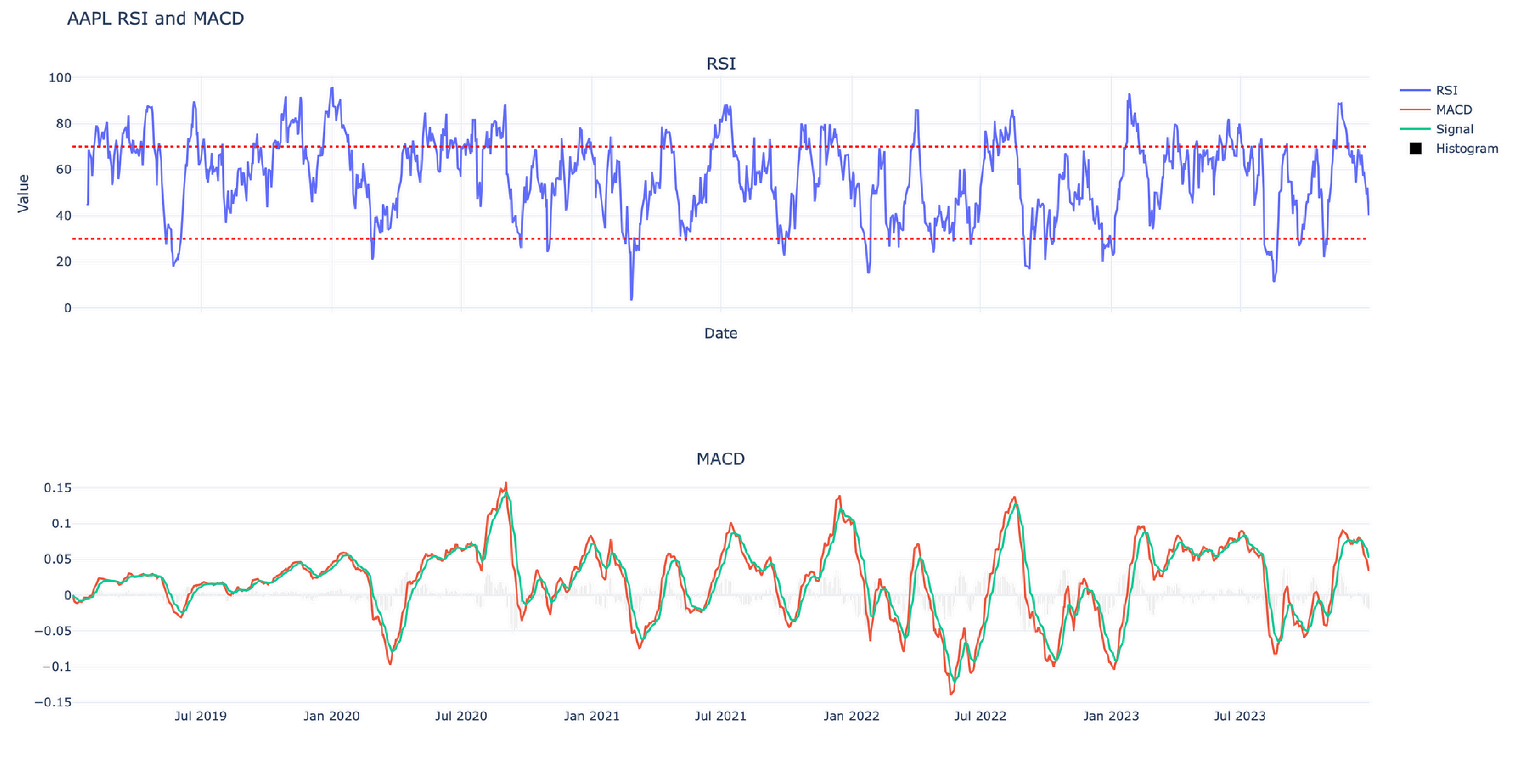
Visualization For Indicators - AAPL



Visualization For Indicators - AAPL



Visualization For Indicators - AAPL



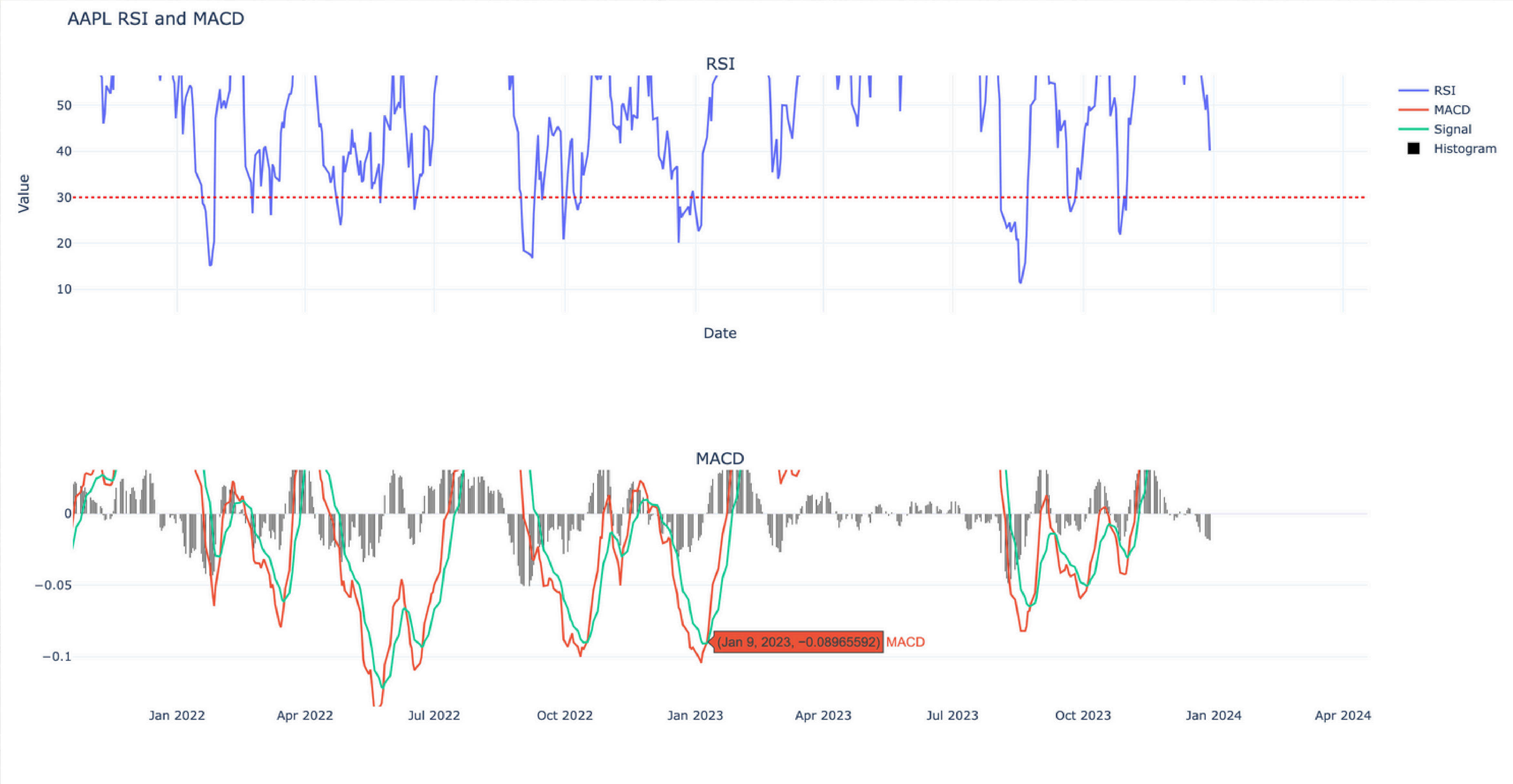
Visualization For Indicators - dashboard

In order to more conveniently obtain the data and images of 10 stocks at different times, I created a **dashboard**. Through the dashboard, you can select stocks and time at will to view their images and data.

05. Interpretation of Results

Interpretation of Results

First, let's analyze individual stocks: Take **AAPL** stock as an example:

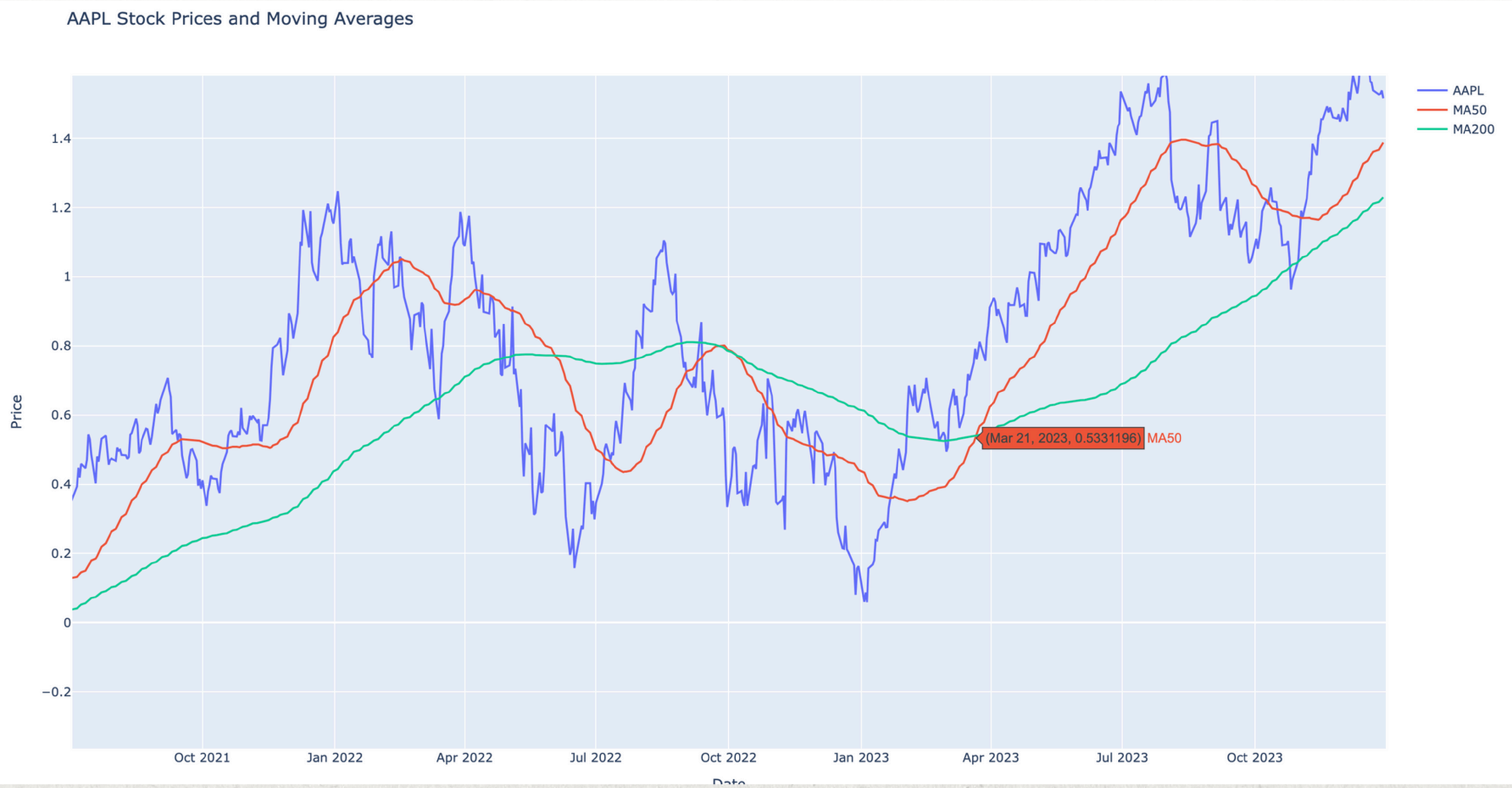


On Jan 9, 2023, the MACD line crossed the signal line upward, the RSI value was **less than 30**, and the MACD line was at the bottom of the market at this time. The **golden cross** was a very suitable **buy signal**.

Interpretation of Results

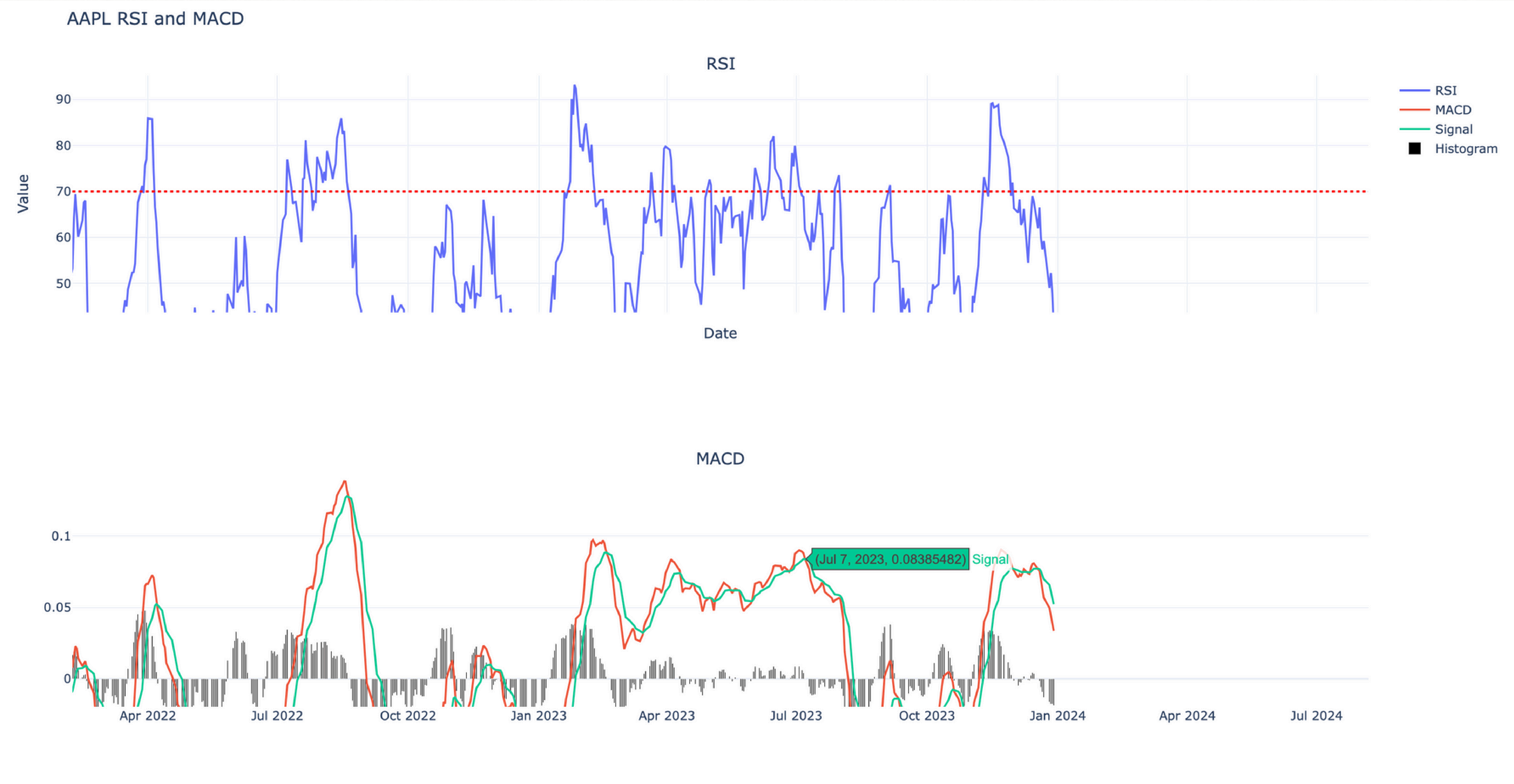
By February 17, 2023, MACD crosses the signal line downward, RSI is **above 70**, and the market is at a high level, forming a **death cross**, which can be used as a sell signal.

Interpretation of Results



On March 21, 2023, MA50 crossed MA200 upward, the market was at a low level, and a **buy signal** appeared.

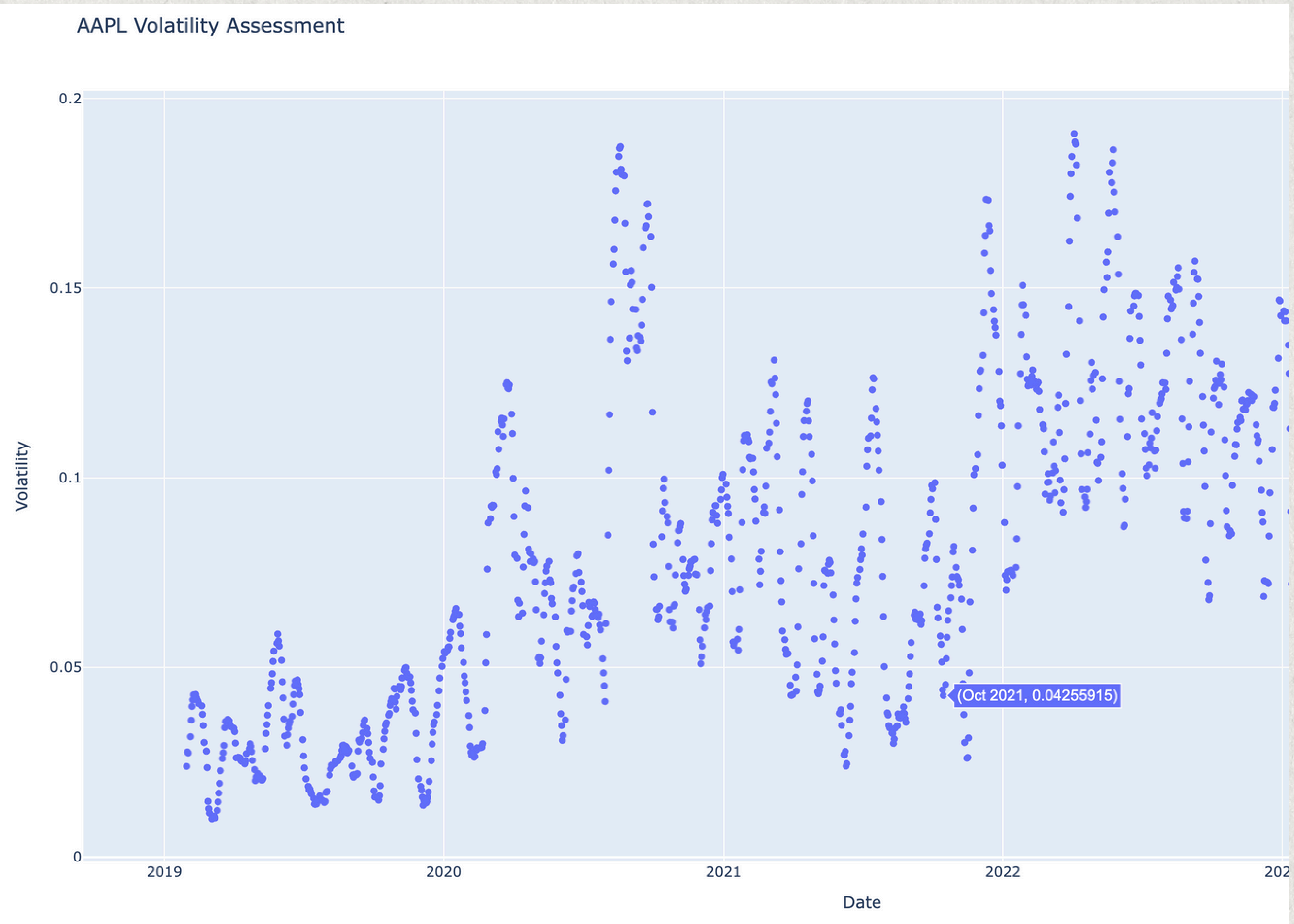
Interpretation of Results



On July 7, 2023, MACD crossed the signal line downward, and the RSI was about 70. The market was at a high level, forming a **sell signal.**

Interpretation of Results

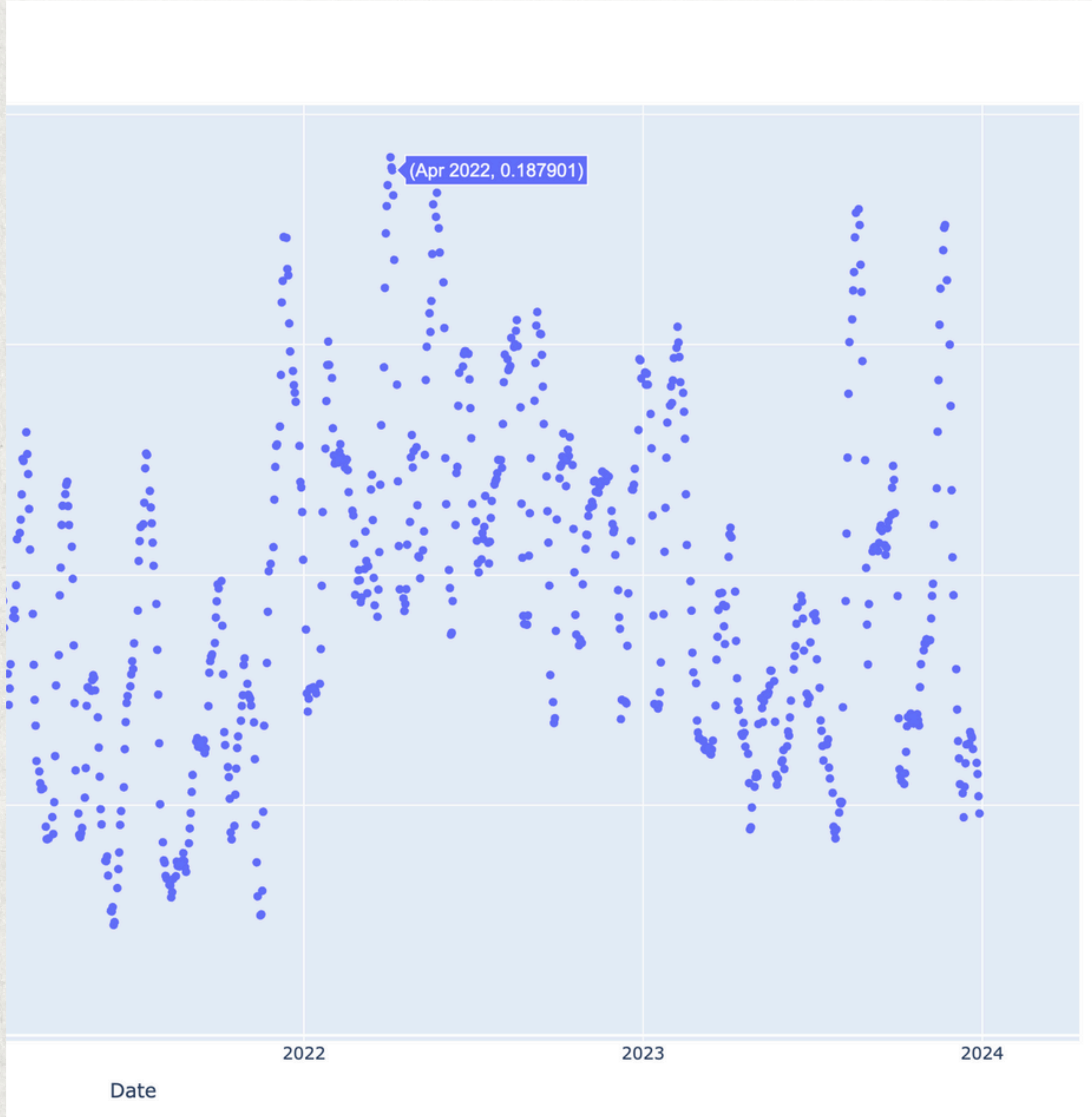
By performing the above two buy and sell operations, **short-term arbitrage** can be achieved. From a long-term analysis, we can analyze it from the following points:



In the third quarter of 2021, AAPL stock volatility was low and the market was in a stable period.

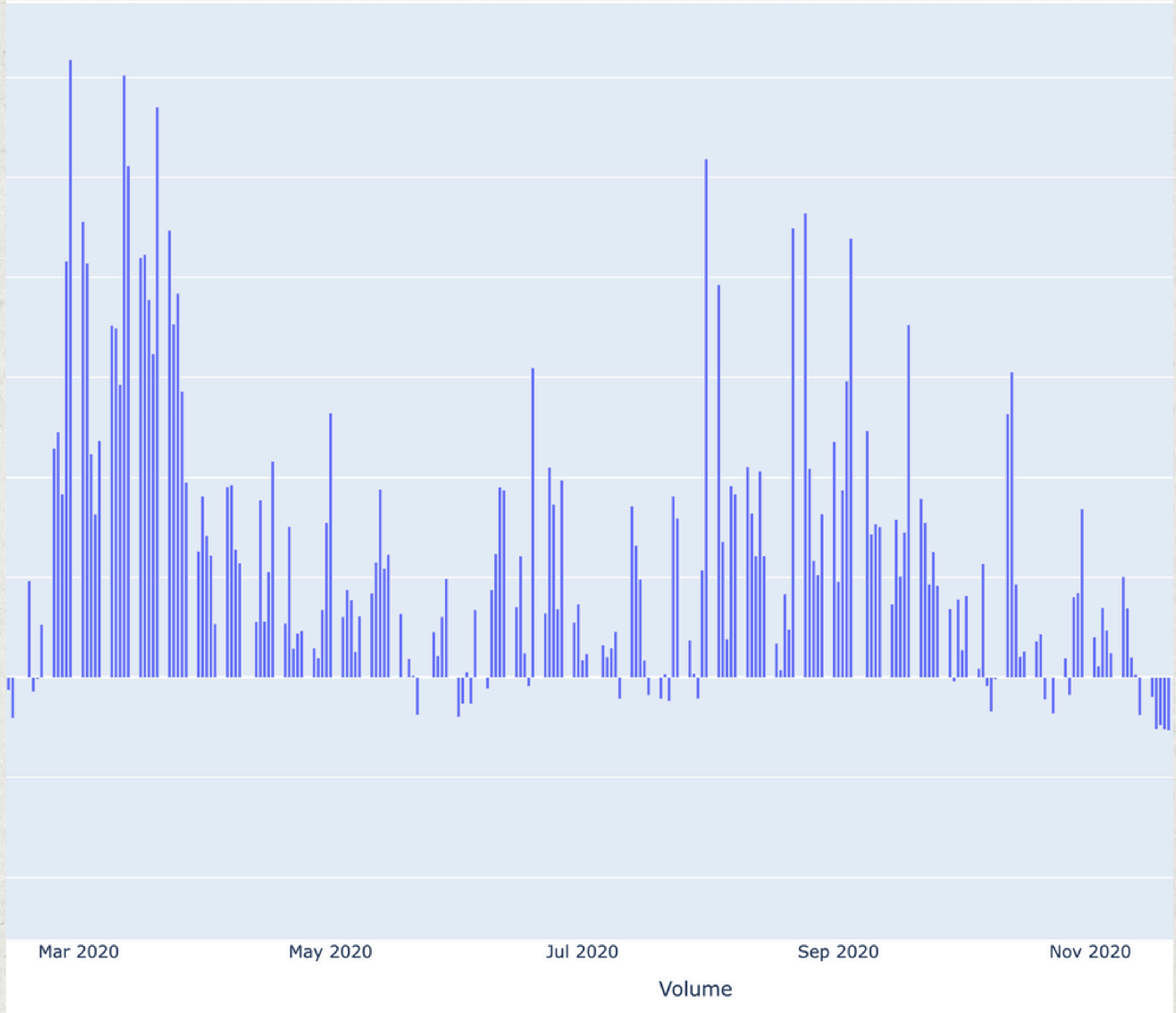
At this time, stockholders' trading sentiment is stable, **prices are rising steadily,** and the overall trend is upward.

Interpretation of Results



In April 2022, AAPL's stock price fluctuated greatly. At this time, stockholder sentiment fluctuated greatly, and the price was at a high level. There were multiple buy and sell signals. At this time, the risk of entering the market was relatively high.

Interpretation of Results



In terms of trading volume, the trading volume has been high since March 2020, and stock prices have continued to rise, and investors' confidence has greatly increased. If you enter the market early in this stage, you can get certain returns.

Interpretation of Results

Overall, although AAPL's stock price has occasionally fluctuated over the past five years, it has generally shown an upward trend and is a good investment target.

Interpretation of Results

At the same time, an excellent stock must be supported by excellent cash flow, excellent profits and an excellent company.

The stock prices of excellent companies also vary. For example, recently Mr. Buffett has been selling AAPL shares. The most likely reason is that the stock price is too high, so we also need to consider these factors when buying stocks.

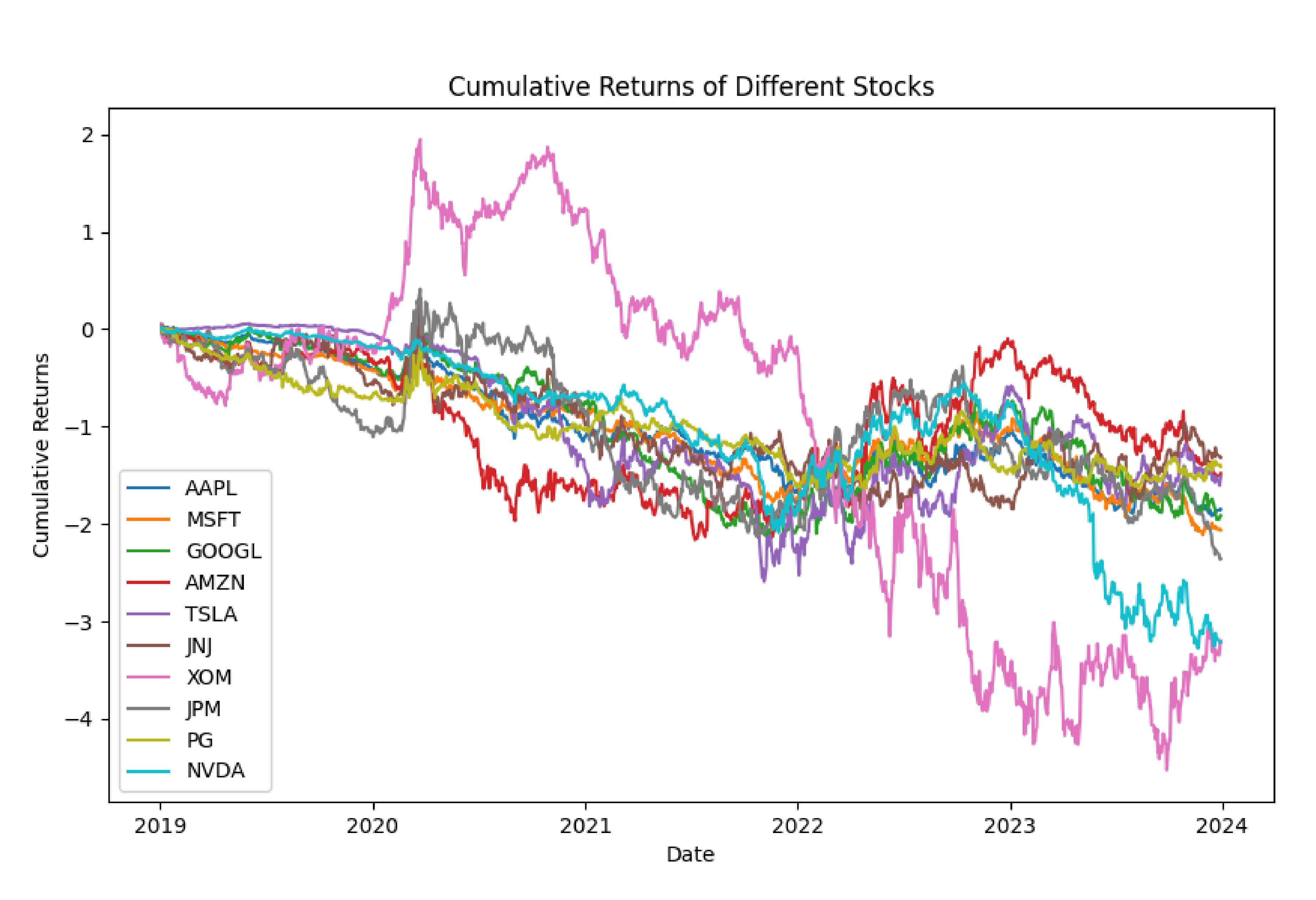
As for the remaining 9 stocks I picked, I will not go into detail because the analysis method is the same as AAPL. I will only compare the performance of each stock as a whole and explore the relationship between the stock market and external economic factors.

Interpretation of Results

When comparing the overall performance of 10 stocks, I used the cumulative return comparison method. I first calculated the daily return of each stock, and then calculated the cumulative return of the 10 stocks. The specific code is as follows:

```
# Compare the performance of different stocks

# Get returns of different stocks
returns = Close_data.pct_change()

# Cumulative returns
cumulative_returns = (1 + returns).cumprod() - 1

# Plot cumulative returns
cumulative_returns.plot(figsize=(10, 6)
                        , title='Cumulative Returns of Different Stocks')
plt.xlabel('Date')
plt.ylabel('Cumulative Returns')
plt.savefig('cumulative_returns.png')
plt.show()
```

Interpretation of Results



The figure shows the cumulative returns of 10 stocks, and we can see:

- XOM's cumulative returns has **changed the most** over the past five years, from the highest cumulative returns at the beginning to the worst cumulative yield in 2024.
- At the beginning of 2024, the cumulative returns of XOM and NVDA are close, while the cumulative returns of the remaining eight stocks are relatively close.

The cumulative returns of most stocks reached their peak at the beginning of 2023 and the end of 2022, and then began to decline, and the cumulative return of each stock fluctuated greatly.

Interpretation of Results

Next, I used the **"pandas_datareader"** library in python to extract the **10-Year Treasury Constant Maturity Rate** as an economic factor representing interest rates, and explored the correlation coefficient between the return of each stock and the 10-Year Treasury Constant Maturity Rate.

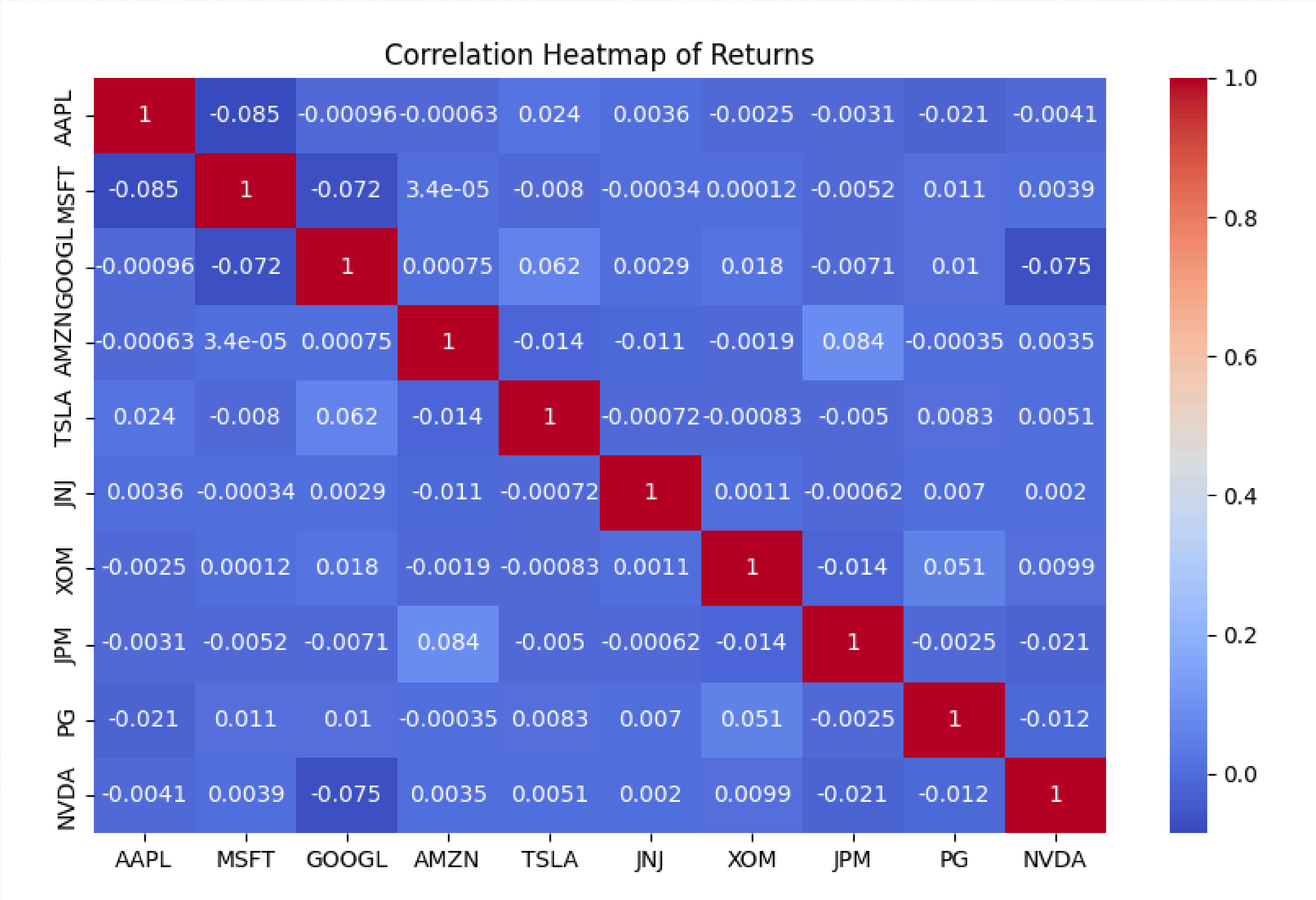
The code and result is as below:

```
# Get correlations between market movements and external economic indicators

# Get 10-Year Treasury Constant Maturity Rate

# Set time
start = datetime(2019, 1, 1)
end = datetime(2024, 1, 1)
t10yr = web.DataReader('DGS10', 'fred', start, end)

# Cal the corr between stocks and t10yr
correlation = returns.corrwith(t10yr['DGS10'])
print("Correlation with Interest Rate:")
print(correlation)
```

```
Correlation with Interest Rate:
AAPL    -0.022496
MSFT     0.028362
GOOGL    0.013865
AMZN     0.056015
TSLA    -0.047709
JNJ     -0.066603
XOM     -0.003563
JPM      0.014000
PG       0.039268
NVDA     0.012882
dtype: float64
```

By reviewing the result, we can simply see that "AAPL", "TSLA", "JNJ" and "XOM" are negatively correlated with this indicator, while the returns of other stocks are positively correlated with the stock interest rate, which can be used as an investment reference.

Interpretation of Results



In addition, I also drew a heatmap of the correlation coefficients between each stock to better understand the correlation between the returns of different stocks and make better investment decisions.

e.g.The correlation coefficient between the returns of "NVDA" and "GOOGL" is negative, so you can consider whether to buy another stock when the return of one stock is negative.

Reported by Shiyuan Liang

Thank you very much!