Project

2024-05-05

### By: Sai Koushik Rao Dugyala

## Summary

This guide outlines the process of developing a sophisticated trading strategy for S&P 500 stocks, transitioning from backtesting to a production-level trading program through paper trading on Interactive Brokers. Utilizing a combination of insider market data, Bollinger Bands, MACD, RSI, and news-based trading signals, the strategy is designed to capitalize on market opportunities by identifying precise trading signals.

The development will start with a backtesting phase covering **January 2023 to April 2024**, using an initial investment capital of $100,000. This strategy supports both long and short positions, offering the flexibility to adapt to various market conditions.

Performance metrics such as total return, Sharpe ratio, maximum drawdown, and win-loss ratio will be employed to evaluate the effectiveness and risk-adjusted returns of the backtested strategy. Production trading program, implemented using IBrokers API, will manage operations including monitoring current portfolio values, updating open positions, and retrieving necessary market data. It will ensure precise trade executions, adjust trade sizes based on pre-tested criteria, and allow for manual user adjustments.

## Learning Outcomes:

**Part 1: Financial Market Data Analysis and Trading Strategy Development**

1. **Data Analysis and Manipulation**: Gain hands-on experience working with OHLC (Open, High, Low, Close) data, Insider data, and news data to identify trends and patterns in financial markets, and develop skills in data manipulation and analysis.
2. **Technical Analysis and Signal Generation**: Apply fundamental technical analysis concepts, including MACD (Moving Average Convergence Divergence), Bollinger Bands, RSI (Relative Strength Index) and News Analysis to generate buy and sell signals, make informed trading decisions, and develop a deeper understanding of market dynamics.
3. **Trading Strategy Development and Backtesting**: Understand the steps involved in building, testing, and optimizing a trading strategy, including defining initial capital, trading signals, position management, and risk management rules.
4. **Risk Management and Performance Evaluation**: Learn how to assess and manage risk through position sizing, stop-loss orders, and performance metrics such as total return, Sharpe ratio, and maximum drawdown.

**Part 2: Production-Level Trading Program Development**

1. **Live Trading and API Interaction**: Develop a production-level trading program that executes a backtested strategy developed in part 1 on live trading days using the Interactive Brokers API (IBrokers).
2. **Error Handling and Trade Management**: Design and implement error-checking mechanisms to ensure accurate trade execution and position management, and adjust trade sizes based on user input and strategy rules.
3. **Real-Time Portfolio Management**: Track and manage current portfolio positions and values in real-time, adapting to changing market conditions.
4. **Live Market Adaptation and Issue Reporting**: Adapt backtested strategies to function effectively in live market conditions, identify and report on any issues encountered, and optimize the production program for improved performance.

## Steps:

This tutorial breaks down the process of creating, backtesting and implementing a production version of trading strategy into 13 clear steps, making it easier to follow and implement.

**Step 1: Trading Strategy Definition** - Outline the objectives, buy/sell signals, and trade initiation and termination rules.

**Step 2: Set Parameters** - Specify goals, risk levels, maximum value per trade, and maximum daily trades.

**Step 3: Generate Indicators** - Use data sources like OHLC, NASDAQ insider data, and NEWS API to calculate: - Insiders weighted average transaction, RSI (Relative Strength Index), MACD (Moving Average Convergence Divergence), Bollinger Bands, Sentiment of News (positive or negative talk about the stock).

### 1. RSI (Relative Strength Index)

**Definition:** The Relative Strength Index (RSI) is a momentum oscillator that measures the speed and change of price movements. RSI oscillates between zero and 100 and is typically used to identify overbought or oversold conditions in the trading of assets.

**Formula:**

Where:

* **Average Gain** is calculated by summing up all the gains within the last n periods, divided by n.
* **Average Loss** is calculated by summing up all the losses within the last n periods, divided by n.
* We chose n to 14 periods, which is a standard setting.

### 2. MACD (Moving Average Convergence Divergence)

**Definition:** MACD is a trend-following momentum indicator that shows the relationship between two moving averages of a stock’s price. It is used to identify changes in strength, direction, momentum, and duration of a trend in the stock’s price.

**Formula:**

**MACD Line:**

**Signal Line:**

**Histogram:**

Where **EMA\_12** is the 12-period exponential moving average of the price. **EMA\_26** is the 26-period exponential moving average of the price. The Signal Line is the 9-period EMA of the MACD Line, and the Histogram represents the difference between the MACD Line and its Signal Line.

### 3. Bollinger Bands

**Definition:** Bollinger Bands are a type of price envelope developed by John Bollinger. They consist of a middle band being an N-period simple moving average (SMA) along with an upper and lower band that are calculated based on the standard deviation of the price.

**Formula:** -

**Middle Band:**

**Upper Band:**

**Lower Band:**

Where **SMA\_20** is the 20-period simple moving average. **Standard Deviation\_20** is the standard deviation of the closing prices over the past 20 periods. is the number of standard deviations to offset the upper and lower bands from the middle band, typically set to 2.

**Step 4: Generate Signals** - Analyze historical trends to identify optimal trading times based on the indicators from Step 3.

**Step 5: Apply Strategy Rules** - Translate theoretical strategy into actionable trading steps, detailing entry and exit strategies.

**Step 6: Manage the Portfolio** - Monitor and manage the portfolio diligently to ensure alignment with the strategic plan.

**Step 7: Review Performance** - Regularly evaluate trading outcomes against the strategic goals.

**Step 8: Set Up Interactive Brokers** - Configure a paper trading account with Interactive Brokers for simulation.

**Step 9: Obtain Data and Positions** - Retrieve current portfolio values and open positions from Interactive Brokers.

**Step 10: Generate Indicators, Signals and Execute Trades** - Apply the same indicator and signal generation techniques used in backtesting to execute trades. Implement trades based on the strategy with strict adherence to the planned size and risk parameters.

**Step 11: User Review and Adjustment** - Allow users to review and adjust trade sizes based on current market conditions and strategy performance.

**Step 12: Place Orders** - Execute trades, ensuring accurate trade execution and consistent error checking.

**Step 13: Demonstration and Reporting** - Showcase the functionality with a one-day trading demonstration including screen captures from Interactive Brokers and R.

## 1. Trading Strategy Definition

### Overview

This trading strategy uses a combination of technical indicators and market data to decide when to buy (enter long) or sell (enter short) a stock. It also uses sentiment from recent news articles and insider trading data to fine-tune these decisions.

### What We Need ?

1. **Market Data:** Daily stock prices including open, high, low, and close prices (OHLC).
2. **Insider Trading Data:** Information on insider transactions.
3. **News Sentiment:** Sentiment analysis results from recent news articles about the stock.
4. **Technical Indicators:** Bollinger Bands, MACD (Moving Average Convergence Divergence), and RSI (Relative Strength Index).

### Trading Rules

#### When to Buy a Stock (Enter Long)

* **Condition 1: Price and Momentum Analysis** - **Criteria:** The stock’s closing price must be below the lower Bollinger Band, and the MACD value must be above its signal line.
  + **Implication:** This setup suggests that the stock is potentially oversold but beginning to show upward momentum. The MACD being above its signal line indicates increasing bullish momentum.
* **Condition 2: Market Sentiment via RSI Adjusted for Insider Activity** - **Criteria:** The stock’s RSI is below a modified entry threshold, which is adjusted based on insider buying activity. If insiders are actively buying, the threshold is increased, making it more favorable to enter a long position.
  + **Implication:** Insider buying can be a strong bullish signal, indicating that those with intimate knowledge of the company expect the stock price to increase. Adjusting the RSI threshold in response to insider purchases allows for entry into a long position even if the RSI is somewhat higher than the typical oversold level (normally 30), reflecting increased confidence in potential price appreciation.
* **Condition 3: Confirmation through News Sentiment** - **Criteria:** After meeting either or both of the above conditions, check the sentiment of the most recent news regarding the stock.
  + **Action:** Only proceed with entering a long position if the news sentiment is positive. If the sentiment is negative or mixed, avoid entering the trade.
  + **Implication:** This step ensures that the decision to enter a long position is supported not only by technical indicators and insider behavior but also by the general market perception as reflected in media sentiment. Positive news can enhance investor confidence and lead to price increases, thereby aligning with the technical bullish signals for a more robust entry decision.

#### When to Sell a Stock (Exit Long)

When managing a long position, consider exiting the trade if any of the following conditions are met.

* **Condition 1: Price Reversion** - **Criteria:** The stock’s closing price rises above the lower Bollinger Band.
  + **Implication:** This indicates that the stock is moving out of the oversold territory, suggesting a normalization or decrease in the potential for further price drops.
* **Condition 2: Momentum Loss** - **Criteria:** The MACD falls below its signal line.
  + **Implication:** A crossover of the MACD line below the signal line is a common indicator of losing bullish momentum. This change suggests that the strength behind the stock’s price increase is waning, potentially leading to a stagnation or reversal of the price trend.
* **Condition 3: RSI Threshold** - **Criteria:** The RSI goes above a neutral value of 45.
  + **Implication:** If the RSI reaches neutral value, we wouldn’t want to take a risk and exit it.

#### When to Sell a Stock Short (Enter Short)

For entering a short position , the criteria is designed to be stringent to ensure high confidence before committing to the potentially riskier strategy of short selling. When considering entering a short position, the strategy demands that all the following conditions must be satisfied before proceeding.

* **Condition 1: Price Extremes** - **Criteria:** The stock’s closing price is above the upper Bollinger Band.
  + **Implication:** This condition suggests that the stock is trading at a higher price than usual relative to its recent volatility, indicating it may be overpriced.
* **Condition 2: Overbought Indicator** - **Criteria:** The RSI is higher than a modified exit threshold.
  + **Implication:** An RSI above a certain level (commonly set around 70, but is adjusted based on market conditions and insider actions) indicates that the stock may be overbought. An overbought stock is likely to experience a price correction, making it a good candidate for short selling.
* **Condition 3: Momentum Shift** - **Criteria:** The MACD is below its signal line.
  + **Implication:** The MACD being below the signal line can indicate that the bullish sentiment is waning and bearish sentiment is starting to dominate.
* **Final Confirmation: Negative Market Sentiment** - **Criteria:** Check the sentiment of the latest news regarding the stock.
  + **Action:** Proceed with entering a short position only if the news sentiment is negative. Negative sentiment can exacerbate the downward pressure on the stock, increasing the likelihood of a profitable short sale.
  + **Implication:** Since short selling involves borrowing shares to sell at current prices in hopes of buying them back cheaper as prices fall, confirming all conditions minimizes the risk of entering a short position during a false signal or temporary market fluctuation.

#### When to Cover a Short Position (Exit Short)

When managing a short position, consider exiting the trade if any of the following conditions are met.

* **Condition 1: Price Movement** - **Criteria:** The stock’s price falls below the upper Bollinger Band.
  + **Implication:** This movement suggests that the price is moving back within its normal range of volatility, indicating a potential easing of the previous overpriced condition.
* **Condition 2: Momentum Reversal** - **Criteria:** The MACD rises above its signal line.
  + **Implication:** When the MACD crosses above the signal line, it indicates a weakening of the downward momentum and a possible shift towards bullish momentum. This change can often precede an increase in the stock’s price, suggesting that continuing to hold a short position could become riskier.
* **Condition 3: Overbought/Oversold Indicator** - **Criteria:** The RSI drops below 45.
  + **Implication:** An RSI falling below 45, suggest neutral movements and to avoid any risk, we exit

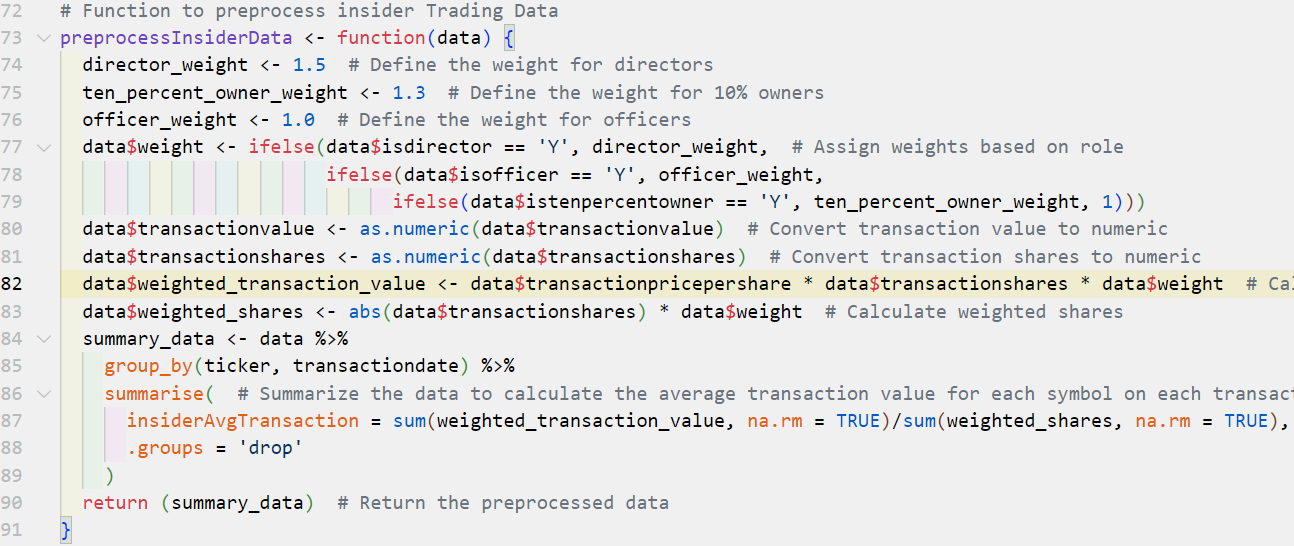
## 2. Set Parameters

We start by initializing key parameters for our trading strategy, setting the starting capital to $100,000, the maximum value per trade to $9,000, and the maximum number of long and short positions to 20 each. These parameters will guide the strategy’s risk management and position sizing decisions, helping to balance potential returns with risk exposure.

## 3. Generate Indicators

Initially, Two sets of data are fetched from Quandl’s datasets for S&P 500 companies. The first set retrieves historical stock price data for the past two years, looping through each symbol in the S&P 500 index. The second set fetches insider trading data, looping through each symbol in the S&P 500 index. The data is filtered to include only relevant columns and rows with non-null values.

Now we preprocess insider data through *preprocessInsiderData* function, which cleverly weights insider trades based on their roles, acknowledging varying levels of influence and information access. Directors’ trades are highly indicative of future prospects, 10% Owners’ transactions signal long-term value, and Officers’ trades reflect short-term outlooks.

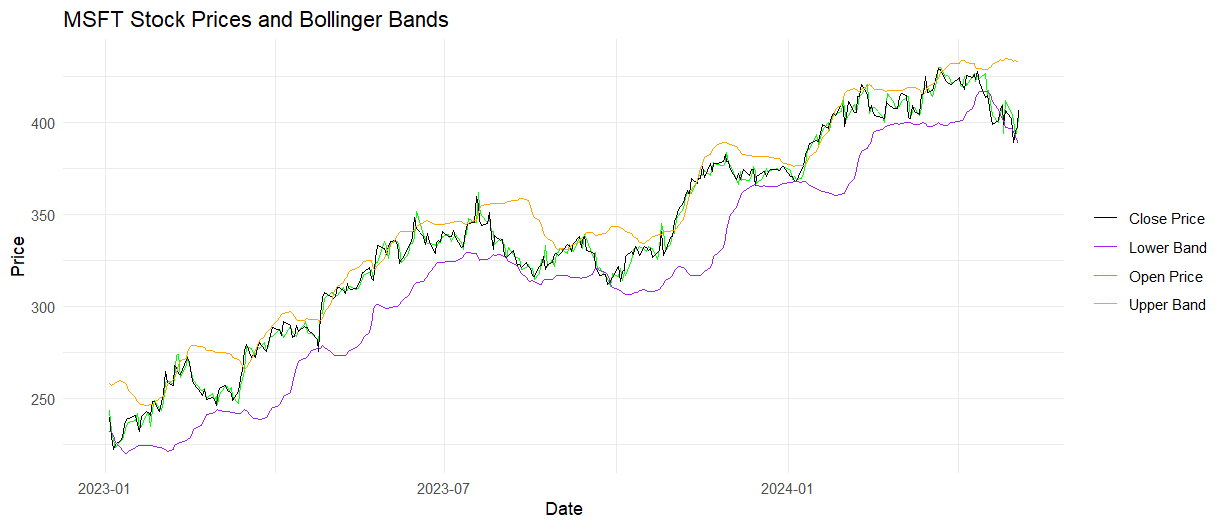


Directors are assigned a weight of 1.5, 10% owners a weight of 1.3, and officers a standard weight of 1.0 (lines 72-74). We assign the appropriate weights based on the insider role indicated in the data ((lines 77-79). Next, it calculates the weighted transaction values and shares by multiplying the respective quantities by the assigned weights (lines 83-84). It is then grouped by ticker and transactiondate (lines 85-90), and to compute the average weighted transaction value for each group.

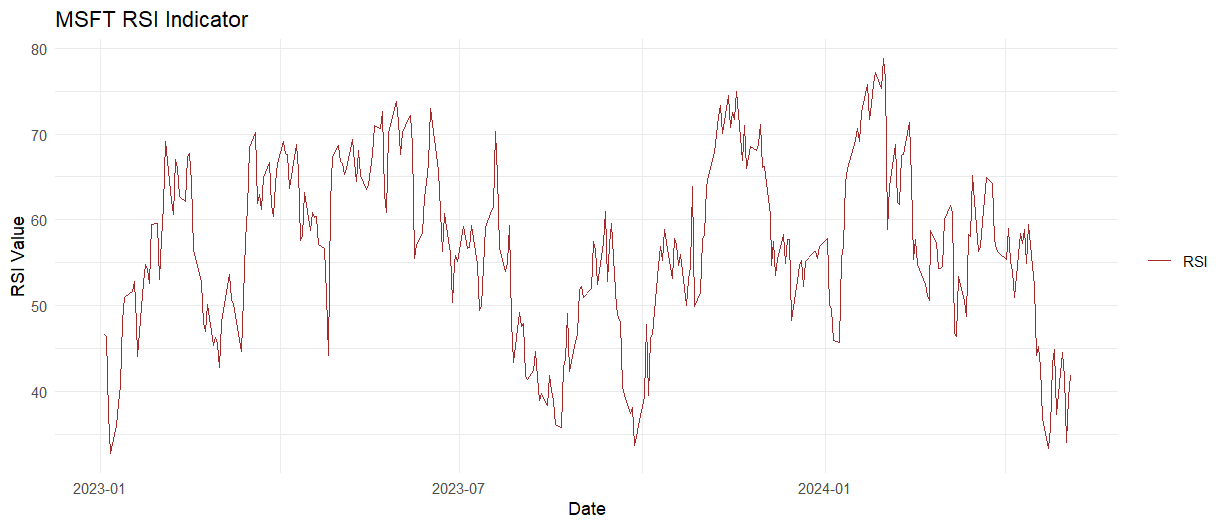
Finally, we generate technical indicators for a given stock symbol by preparing the data, calculating lagged values, and computing various technical indicators such as **RSI, MACD, and Bollinger Bands**.

Below are snapshots of our indicators for ‘MSFT’ stock.

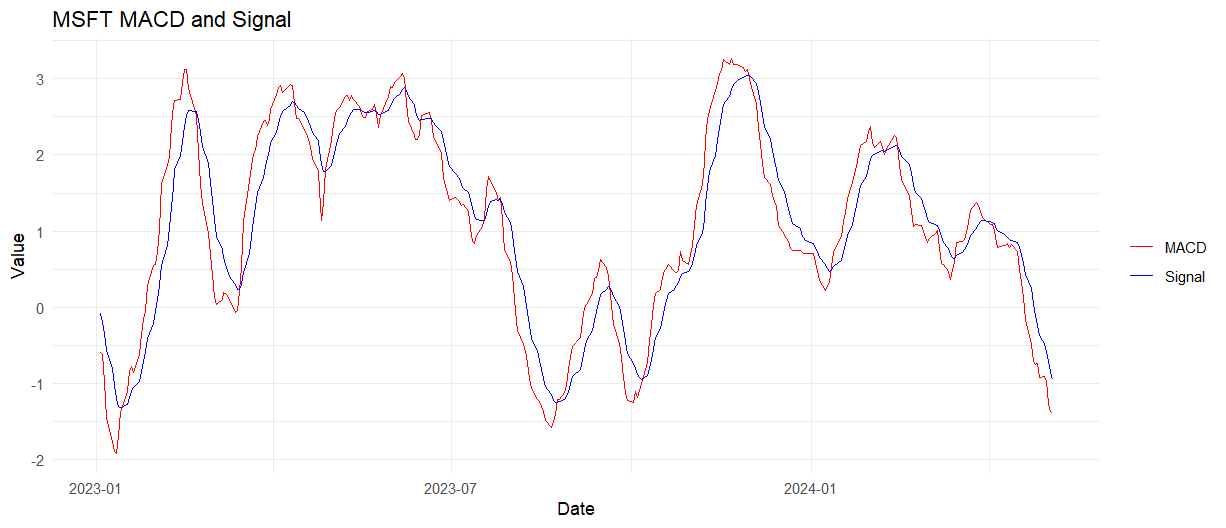
**Bollinger Bands:**



**RSI:**



**MACD:**

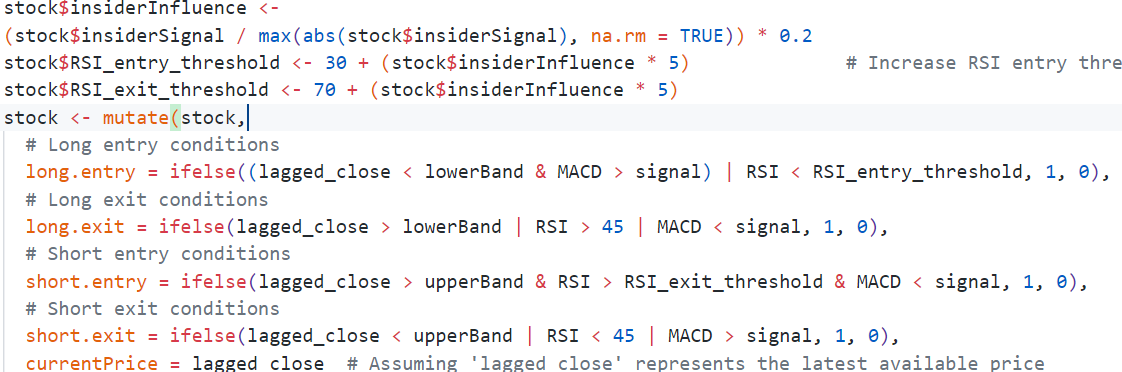


## 4. Generate Signals

We filter data for the current date and calculating trading signals based on RSI, MACD, and Bollinger Bands. We adjust RSI thresholds based on insider influence and generates long and short entry and exit signals. Additionally, we analyze news sentiment using the Bing sentiment lexicon and adjusts trading signals accordingly.

Insiders often trade based on their expectations for the company’s future, which might not yet be reflected in the public market data. Adjusting thresholds based on insider activities can provide early signals about potential bullish or bearish trends. Static trading thresholds can lead to missed opportunities or increased risk during abnormal market conditions. Dynamically adjusting these thresholds allows the strategy to be more flexible and reactive to new information, increasing the potential for higher returns or reduced losses.

By modifying entry and exit thresholds based on insider signals, traders can better manage the risk associated with their positions. For instance, raising the entry threshold when insiders are selling aggressively might prevent entering long positions just before a downturn.



The provided code dynamically adjusts the RSI thresholds based on insider trading signals, integrating this data into the trading strategy to make it more robust and informed:

* **Lines 185-187**: The code first computes the insiderInfluence, which quantifies the impact of insider trading signals relative to their maximum observed value, scaling it down by a factor of 0.2. This scaling is crucial as it tempers the influence of extreme insider activities, preventing overly aggressive adjustments that could skew the strategy.
* **Line 186**: The RSI\_entry\_threshold is adjusted upwards by up to 5% of the insider influence. This adjustment means if insiders are selling (indicating potential negative future prospects), the entry threshold is increased (making it harder to enter a long position). This prevents buying into a potentially falling price, enhancing the cautious approach of the strategy.
* **Line 187**: Conversely, the RSI\_exit\_threshold is adjusted downwards by the same factor. If insider activities suggest negative sentiment, lowering the exit threshold makes it easier to exit positions, thereby securing gains or minimizing losses before a potential decrease in stock value.

We then use sentiment derived from news articles to influence trading signals, aligning trades with the prevailing market sentiment about a stock. This method allows the strategy to react to fresh news, potentially capturing rapid market movements in response to news events before they are fully reflected in the stock price.



* **Lines 214 & 220-233**: The code starts by loading the Bing sentiment lexicon, essential for evaluating the tone of news articles. It sets a default sentiment as “neutral” and adjusts this based on the analysis of news content fetched from the News API.
* **Lines 221 & 223-232**: Constructs a URL to retrieve news articles for each stock symbol, fetches the news, and checks the success of this operation. If successful, the code performs sentiment analysis on the top 50 articles to determine if the overall sentiment is positive or negative.
* **Lines 232-233**: Based on the sentiment analysis, the trading signals (long.entry and short.entry) are adjusted: enabling a long entry if the sentiment is positive, and a short entry if it is negative.

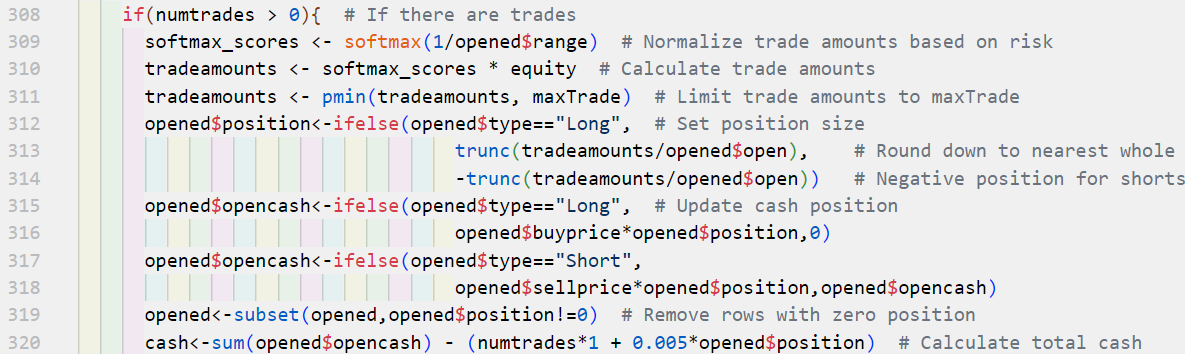
## 5. Apply Strategy Rules:

We then follow a logical sequence of closing existing positions, updating open positions and equity, identifying new positions to open.

The process of closing positions involves several key steps. First, existing long and short positions are identified as candidates for closure. Next, these positions are matched with corresponding exit signals, triggering the closure process. The closing cash value is then calculated, and the sell and buy prices are updated accordingly. The profit or loss for each closed position is determined, and finally, the closed positions and aggregate cash, including transaction costs, are noted down.

In general, open positions function identifies and opens new positions based on entry signals. It first checks for existing positions and merges them with corresponding entry signals. If no positions exist, it considers all signals for the day. The function then sets the type of trade (Long or Short), sets buy and sell prices, and sorts the trades by range. It applies constraints based on maximum allowed Long and Short trades and calculates trade amounts based on risk. Finally, it updates the cash position and returns the opened positions and the total cash spent.

Instead of investing the same amount of money for each share, we can smartly adjusts the sizes of trading positions based on the associated risks and potential rewards of each trade. By using the softmax function, we can normalize trade sizes, giving preference to less volatile trades, thus aiming to maximize returns while minimizing risk exposure.



**Calculation of Trade Amounts ([Lines 311-312])**:

* Trade amounts are first calculated by applying a softmax function to normalize investment based on risk, and then multiplied by the total equity. These amounts are subsequently capped at a maximum allowable per trade (maxTrade), ensuring that investments are proportionally distributed according to risk and within set limits.

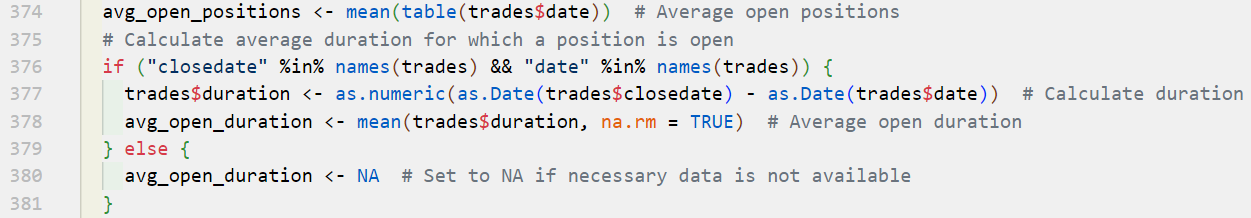
**Transaction Costs ([Line 320])**:

* After trading, the total cash is adjusted by deducting transaction costs, which include a fixed fee per trade plus a small percentage of the position size.

## 6. Managing the Portfolio

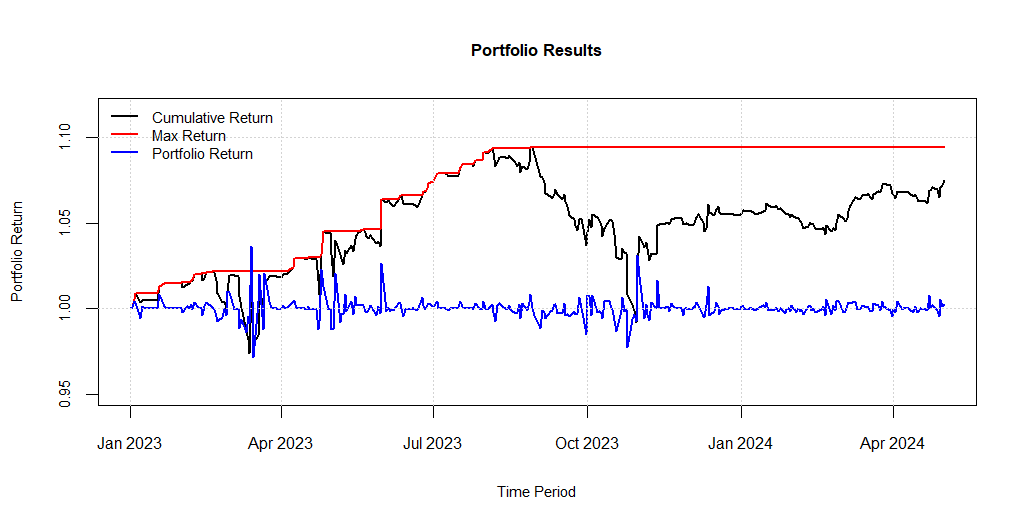
We evaluate the performance of our trading strategy by calculating various metrics. We start by counting the number of trading days, total days, and percentage of days with trades. Then, calculate the total number of trades, short and long trades, and returns for each trade and overall. Additionally, we determine the percentage of winning trades, average returns, and cumulative return. Also, lets calculate the maximum return, maximum drawdown, longest drawdown period, mean daily return, and Sharpe ratio.

We also calculated average duration for which positions are open provides insight into the trading strategy risk exposure. Shorter durations might indicate a high-frequency trading strategy or one that captures quick, short-term gains. Longer durations could imply a strategy that bets on long-term movements and may have different risk.



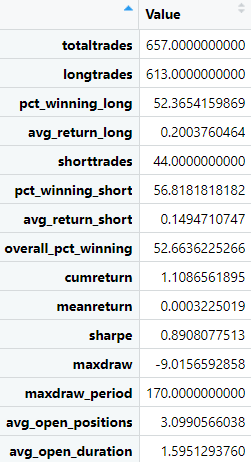
We calculate the duration of each trade by subtracting the opening date from the closing date, then computes the average of these durations to determine how long, on average, trades are kept open.

## 7. Review Performance



The above chart illustrates the performance of a portfolio using our strategy over time, depicted by three key metrics: Cumulative Return, Max Return, and Portfolio Return.

* **Cumulative Return (Black Line):** This line represents the total return of the portfolio accumulated over the given period from January 2023 to April 2024. The steady increase in this line suggests a generally positive performance across the period, with some fluctuations indicating varying performance at different times.
* **Max Return (Red Dotted Line):** This line indicates the maximum return achieved by the portfolio at any point in the given timeframe. It serves as a benchmark of the best performance the portfolio has reached, which appears to have remained constant after initially being achieved, suggesting that this peak was not surpassed in subsequent periods.
* **Portfolio Return (Blue Line):** Representing the actual return of the portfolio across different times, this line is notably more volatile. The fluctuations reflect the short-term ups and downs in the portfolio’s value, which are common in active trading scenarios.



Lets Analyze the performance metrics to get a detailed insight into the strengths and weaknesses of our trading strategy.

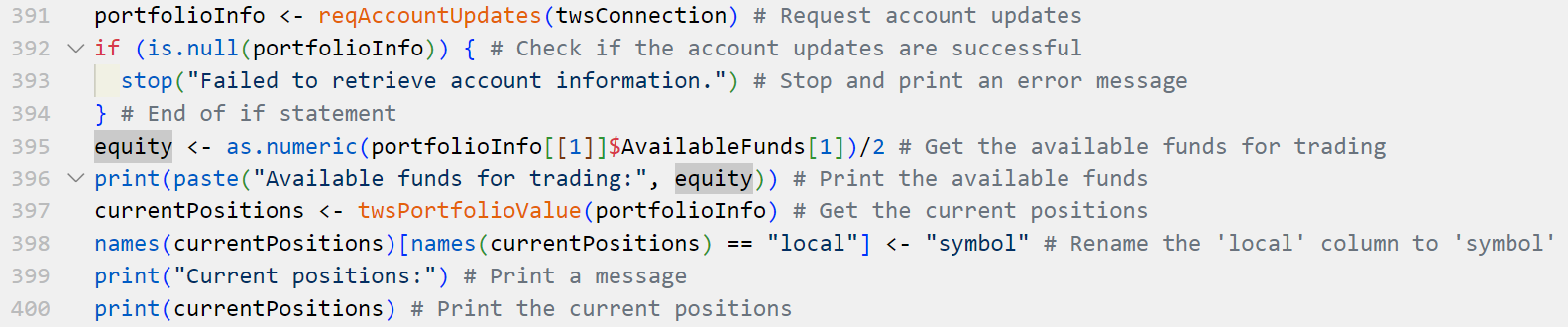
1. **Total Trades (657)**: A high number of trades indicates an active trading strategy. This could suggest a high-frequency trading approach or a strategy that capitalizes on short-term market fluctuations.
2. **Long Trades (613)**: The majority of our trades are long, which shows a bullish bias in our trading strategy. This can be advantageous in a rising market but may pose risks if the market turns bearish.
3. **Max Drawdown Period (170)**: This metric suggests that the longest period the portfolio was in a drawdown from the peak lasted about 170 days. This is quite significant and could indicate periods of high market volatility or some strategy weaknesses during certain market conditions.
4. **Short Trades (44)**: A smaller number of short trades compared to long trades might indicate less confidence or a less developed strategy for bearish market conditions. It’s also possible that market conditions favored long positions during the period studied.
5. **Pct Winning Long (52.36%)**: Being slightly above 50%, this win rate for long positions is modestly effective. It suggests that while the strategy is profitable, there might be room for improvement in trade selection or timing.
6. **Overall Pct Winning (52.66%)**: This overall win rate is just above the break-even point when not considering transaction costs. It’s important to evaluate the net profitability after accounting for all trading costs and fees.
7. **Pct Winning Short (56.81%)**: This win rate for short trades is below 50%, which could indicate that your strategy is less effective in bearish conditions or that there are fewer opportunities to profit from declines in the market.
8. **Avg Open Positions (3.09)**: This shows a moderate level of diversification. Managing the number of concurrent positions is crucial to balancing risk and ensuring that not too much capital is tied up in any single market exposure.
9. **Avg Open Duration (1.59 days)**: This short duration indicates a very active trading style, likely scalping or day trading. Such strategies require constant market monitoring and quick decision-making.
10. **Cumulative Return (1.10 times)**: Over the period analyzed, the portfolio returned a 10% increase over the initial investment, which is a positive outcome, especially considering the number of trades and the market environment.
11. **Sharpe Ratio (0.89)**: This Sharpe ratio is fairly moderate, suggesting that the risk-adjusted return of the portfolio is reasonable but not exceptional. Higher values would be more indicative of a superior risk-adjusted performance.
12. **Max Drawdown (-9.01%)**: This maximum drawdown is significant but not excessive for active trading strategies. It indicates the largest drop from peak to trough in portfolio value, which is crucial for understanding potential losses during adverse conditions.
13. **Mean Return (0.00032%)**: This very small mean return per trade highlights the challenges of maintaining profitability on a per-trade basis, especially in a high-frequency trading setup.

## 8. Set Up Interactive Brokers

It’s crucial to have the Interactive Brokers (IB) Gateway/Desktop client properly set up and configured. We need ensure a stable connection by closing open network connections and retrying to connect to Trader WorkStation (TWS).

## 9. Obtain Data and Positions :

We fetche historical data and insider data for the S&P 500 index from NASDAQ. We retrieve account information and current positions from a TWS (Trader Workstation) connection. It then retrieves current positions for further execution.



## 10. Generating Indicators, Signals and Execute Trades (tws):

The insider data is preprocessed by weighting trades based on insider roles, with directors, 10% owners, and officers assigned different weights. We then calculates weighted transaction values and shares, groups them by ticker and date, and compute average weighted transaction values. Finally, we generate technical indicators such as RSI, MACD, and Bollinger Bands for each stock symbol.

We generate trading signals based on RSI, MACD, and Bollinger Bands, and adjust RSI thresholds using insider trading data. Insider influence is quantified and used to adjust entry and exit thresholds, making the strategy more flexible and responsive to new information. Additionally, news sentiment analysis is used to adjust trading signals, aligning trades with prevailing market sentiment.

Below is a snap-shot of trading signals



Lets processes and executes trading signals, closing existing trades and opening new ones. We filter signals, enriches trade information, and executes trades while considering factors like equity, trade size, and maximum daily trades. The enrichTradeInfo function should prepares trade details, distinguishing between opening and closing trades, and determines trade quantities and actions like “SELL”, “BUY”, or “HOLD”.

## 11. User Review and Adjustment :

We presented a menu for users to review and modify trading decisions before execution. The logical flow begins with the user selecting an option: quitting without trading (“Q”), executing trades (“E”), modifying price (“M”), deleting a trade (“D”), or changing trade quantity (“C”). If the user chooses to execute trades, the function checks if the total trade value exceeds available equity, and if so, prompts the user to adjust trades. If the user chooses to delete a trade, they enter the trade’s row number, which is validated before removal. If the user chooses to change trade quantity, they enter the row number and new position size, which are validated to ensure a positive size and, for closing trades, a size not exceeding the existing position. After adjustments, the function calculates the final equity and returns the updated trades and equity. This process repeats until the user chooses to execute or quit.

## 12. Placing Order

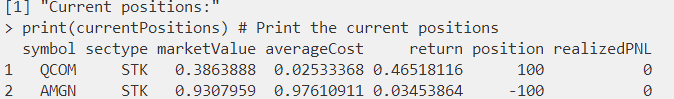
After user review, we initiate the trade placement process. It begins by extracting essential trade details such as symbol, action, quantity, and current price from the trades data frame, then determining the appropriate order type based on whether the trade is for opening or closing a position, selecting between Market on Close (MOC) or Market on Open (MOO)(although we utilized Market (MKT) orders as MOC is not supported by the API).

## 13. Demonstration and Reporting

Lets see how our script works through a demonstration via screen-shots. The trading system will determine trade sizes based on predefined criteria from the backtest. Once all orders are generated, the program will present a summary of the impending trades, offering options to modify, remove, execute, or cancel the orders before proceeding with the trading process.

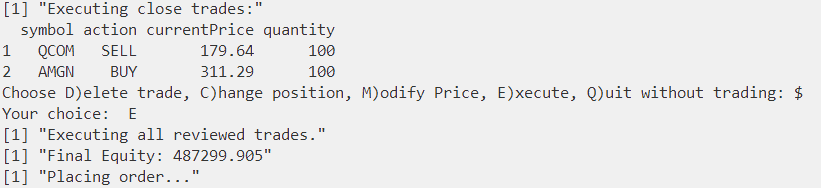
We begin with showing the amount we have and shares in our portfolio as shown below:



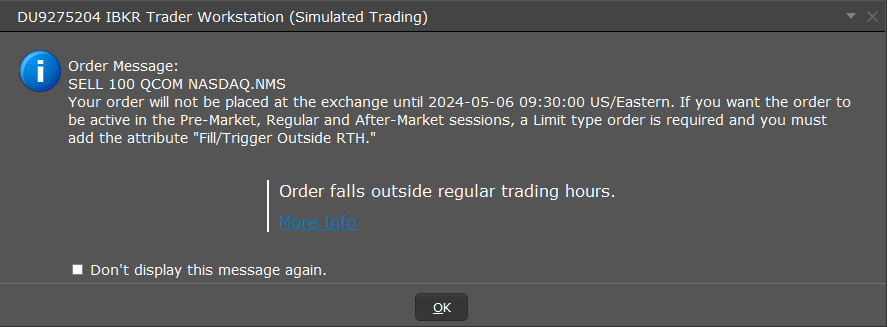


### 1. Placing Order in IBrokers

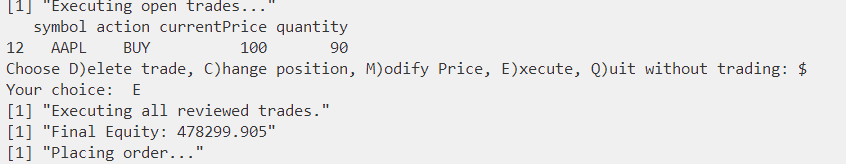
Lets close out existing positions:

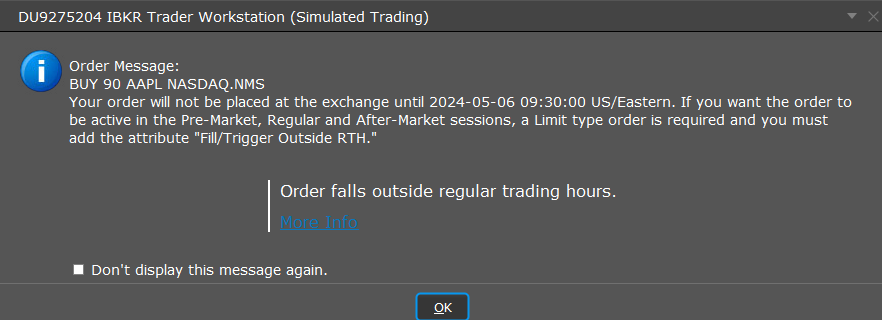


We get a confirmation message for order placed



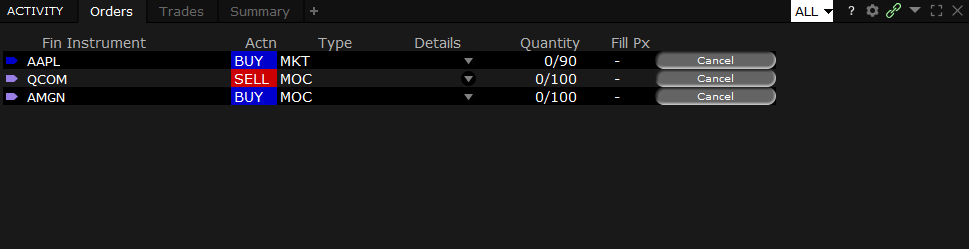
Lets open new positions:





### 2. Orders in IBrokers

After executing the trades, we can switch back to IBrokers and review the orders we’ve placed. The figure below illustrates close orders we have placed.

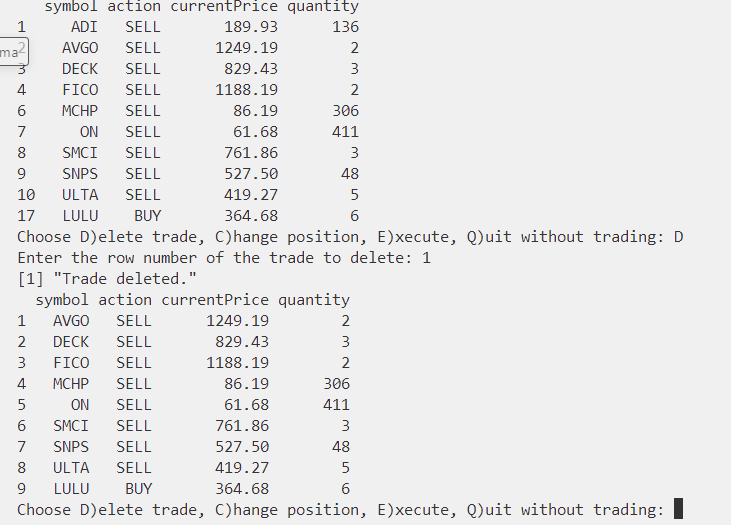


### 3. Final Checks and Execution

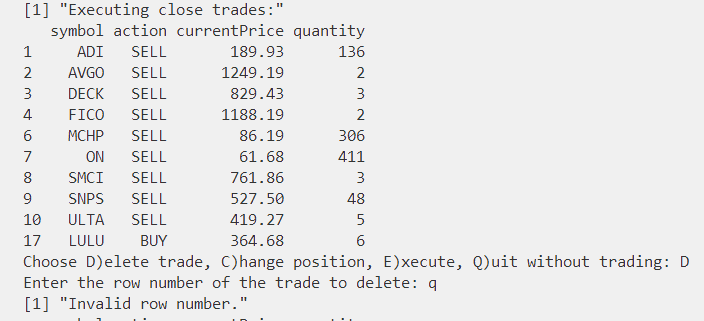
Once the trading strategy executes, it will first display the order to close (assuming we have open orders in our portfolio) allowing the user to either continue executing, modify the orders, or exit the process altogether. We have the options D,C,E,Q as shown in below image.

#### 1. **Delete Trade**

This feature enables us to remove any trades from our positions before proceeding with execution. The attached image below demonstrates a successful deletion of an order and the updated trades following the successful deletion.

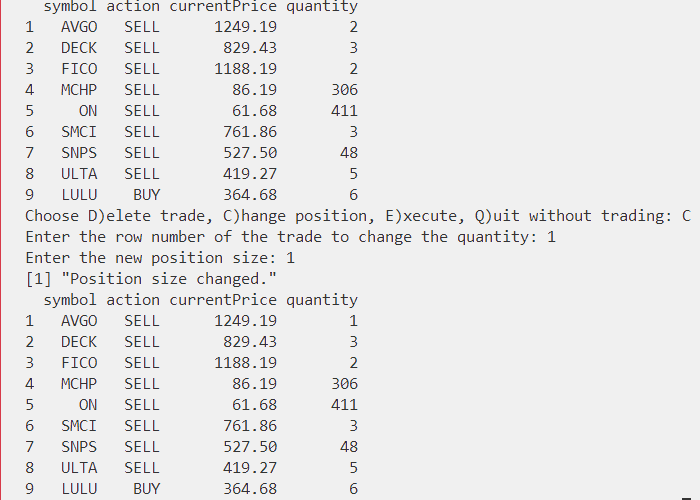


If an incorrect row number is provided as input, our program will detect it and display a message indicating invalid. The figure below illustrates this scenario.

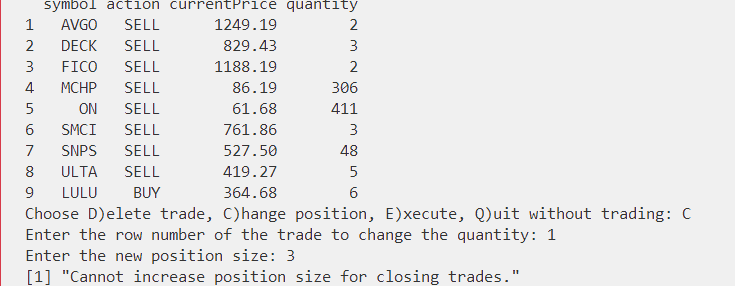


#### 2. **Change Position**

We also have the capability to adjust the size of orders within our positions. The figure below illustrates a successful change in position for our orders. Initially, the program prompts for the row number of the order we wish to modify, followed by the desired quantity to which we want to change it.

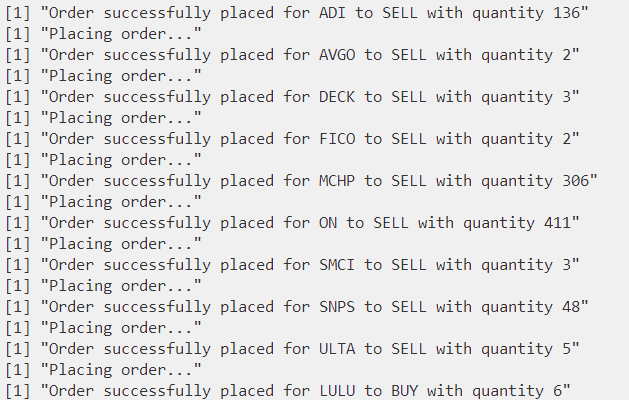


Our program includes error handling during position changes. For instance, if we attempt to increase the quantity for a close order beyond what we currently have, the program will detect this and prevent execution. The figure below illustrates this error scenario.



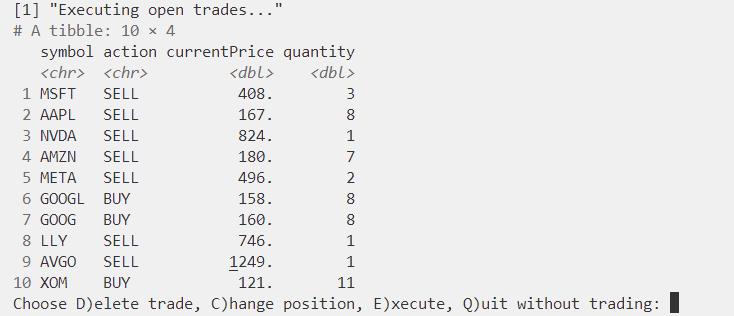
#### 3. **Execute**

This option enables the execution of all pending trades. The figure below depicts the screen after selecting this option.



Similar process is done for opening new trades

We begin with showing the possible to trades (limited to 10)



These can be modified the same way we did for closing positions

#### 4. **Quit without Trading (option Q)**

Selecting this option will halt the program from proceeding with trade execution, effectively canceling the process.

## Challenges and Issues

We faced several limitations and constraints that impacted our development process. Due to News API call limitations, we couldn't generate enough data for a year of back testing. Additionally, we didn't integrate order tracking with IBrokers due to limited documentation, resources, and time.

Several features were considered but not implemented due to time and resource constraints, including in-depth insider data analysis, robust statistical analysis, transformer/neural network integration for advanced sentiment analysis, efficient cache management, order execution tracking, and incorporation of real transaction costs in the production version.

## Appendix

### Integration of Sentiment Analysis in Trading Strategy

This appendix details the methodology and code used to incorporate sentiment analysis into a trading strategy, leveraging news data fetched from the News API and the Bing sentiment lexicon for sentiment scoring. The purpose of this integration is to enhance the trading signals with insights derived from recent news articles related to the stocks in question.

#### A. Overview of Components

1. **News API**: This service provides recent news articles which are queried for each stock symbol based on trading signals. The API returns the most popular articles within the last five days from the current trading date. [News API](https://newsapi.org/)
2. **Bing Sentiment Lexicon**: Part of the tidytext package in R, this lexicon categorizes words into positive or negative sentiments, which are used to evaluate the sentiment of news content.
3. **Sentiment Analysis Logic**: Combines the outputs of the News API and Bing lexicon to assign a sentiment score to each stock which influences trading decisions.

#### B. Detailed Code Description

1. **Loading Sentiment Lexicon**:

* bing\_lexicon <- get\_sentiments("bing") # Load Bing sentiment lexicon from tidytext package

1. **Iterating Over Stock Entries**:

* for (i in 1:nrow(stock)) {  
   symbol <- stock$symbol[i]  
   long.entry <- as.numeric(stock$long.entry[i])  
   short.entry <- as.numeric(stock$short.entry[i])  
   sentiment <- "neutral"

1. **Fetching News Articles**:

* if (long.entry == 1 | short.entry == 1) {  
   url <- paste0("https://newsapi.org/v2/everything?q=", symbol, "&from=",   
   as.Date(trade\_date - 5), "&to=", as.Date(trade\_date),   
   "&sortBy=popularity&apiKey=", API\_KEY)  
   news <- GET(url)

1. **Processing News Data**:

* if (status\_code(news) == 200) {  
   jsonData <- fromJSON(rawToChar(news$content))  
   titles <- jsonData$articles$description[1:20]

1. **Analyzing Sentiment**:

* sentiment\_scores <- tibble(title = titles) %>%  
   unnest\_tokens(word, title) %>%  
   inner\_join(bing\_lexicon, by = "word") %>%  
   summarise(sentiment = sum(case\_when(sentiment == "positive" ~ 1,   
   sentiment == "negative" ~ -1,   
   TRUE ~ 0)))  
  if (sentiment\_scores$sentiment > 0) { sentiment <- "positive"}   
  else if (sentiment\_scores$sentiment < 0) { sentiment <- "negative"}

1. **Applying Sentiment to Trading Decisions**: Explained how sentiment analysis results influence trading entries, enhancing the decision-making process with additional market sentiment data.

### Usage of softmax from LDATS package

The softmax function is a mathematical function to convert a vector of real values into a probability distribution. Each output of the softmax function represents the probability that the respective input belongs to a particular class. The sum of all the probabilities in the output vector is 1, making it a proper probability distribution.

### Mathematical Formula

Given a vector of real numbers for classes, the softmax function applied to at index is given by:

is the base of the natural logarithm. - is the exponential of the element of vector .

The denominator is the sum of the exponentials of all elements in , which acts as a normalization constant ensuring the output values are between 0 and 1 and sum up to 1.

|  |
| --- |
|  |