Automated Trading Strategy Final Report

The design and implementation of
Short-term Timing Strategy and Long-term Momentum Strategy

COMP396 Team7

Tianyi Ye

Weiyi Liu

Yiwang Tian

Zhiyu Chen

Zixuan Zhu

University of Liverpool

Contents

Section 1 - Final Choice of Submitted Strategy

A combined strategy will be introduced in this report that incorporates multiple technical analysis approaches, with the primary framework incorporating the Short-term Timing Strategy and the Long-term Momentum Strategy. In this section, we will go through what our strategy accomplishes specifically, how they complement each other synergistically, and how to construct a solid and efficient trading plan.

Section 1.1 About the Strategy

- **Long-term Momentum Strategy**

The general idea of this strategy is trend-following over a long period. The main factors are the *Simple Moving Average* and the *Moving Average Convergence/Divergence (MACD)*. Both of these factors need to be satisfied at the same time when determining whether to ensure the accuracy of the strategy entering or exiting a trending market. However, it is significant to introduce a stop loss factor for limit order trading – a "Stochastic Oscillator Indicator".

1. **Simple Moving Average (SMA) Indicator**

   In time series data, a Simple Moving Average (SMA) is a common average of the past *n* data points. There are no weighting factors applied to each of the data points since each point is proportionally weighted (Hansun, 2013). Although SMA is clear to help identification, the intersection of two SMAs whose lookbacks are short-term and long term respectively will be more convincing.

   As a result, traders can get more objective and accurate signals reflecting the market forces. In R, the implementation of this indicator makes direct use of the “sma()” function in the TTR package.

2. **Moving Average Convergence/Divergence (MACD) Indicator**

   The MACD indicator utilizes the same notion of averaging as the SMA, but it calculates the mean using exponential decreasing weighting (Eric, Andjelic & Redzepagic, 2009). Given the lagging nature of the SMA indicator, the addition of the MACD indicator mitigates the lagging issues with the SMA and allows for the removal of misleading signals issued by the SMA wherever feasible (Mateia, 2013).

   Therefore, the MACD can be used to detect medium- and long-term trends more accurately. In R, the MACD function in TTR is used directly.

3. **Stochastic Oscillator Indicator (Calculating KDJ Indicator)**

The Stochastic Oscillator measures where the close is in relation to the recent trading range, which is optimized for risk management to realize stop loss (Lane, no date). This incorporates the highest and lowest prices for a particular period, as well as the closing price of the previous calculation period, and then mainly computes the K (also known as the fast line, written as %K). Moreover, in R, this indicator is realized by *stoch()* in the TTR package.

Since this strategy is set up as a long-term trend-following strategy, it will not work best in oscillating markets. The reasons for choosing this strategy and how it is combined are described in section 2.1.

- **Short-term Timing Strategy**

The Short-term Timing Strategy is the combination of Resistance Support and Relative Strength (RSRS) Strategy and Multi-factor Strategy. RSRS is the main strategy, while Alpha-006 and Alpha-018 are selected from multiple factors as a supplement to add positions where particular condition meets. By combining the RSRS and Alpha-006 indicators, the strategy benefits from the complementary information provided by these two factors. The RSRS indicator focuses on price momentum, while the Alpha-006 indicator supplements the dimension of volume analysis. This combination allows the strategy to capture different aspects of market behavior and potentially identify more robust trading opportunities.

The RSRS strategy is a trend-following strategy based on the Resistance Support Relative Strength (RSRS) indicator. The RSRS indicator focuses on the relative strength between daily high and low prices, reflecting the traders' expected judgment of the top and bottom of the current market state.

In the test set, RSRS has shown good returns but very large retracements. To control retracement and risk management, we introduced the Average True Range (ATR) indicator and limit order to limit trading losses.

1. **RSRS Indicator**

The RSRS indicator is designed to help us identify the lowest or highest point within a given period. Unlike the traditional fixed values of resistance and support levels (traders believe that the price will either decline from this position or rebound), we explore their predictive value for future market movements from a variable perspective.

We believe that the daily highest and lowest prices can act as resistance and support levels, respectively, and can quickly reflect the market's attitude towards these levels. Using the relative change in position, such as the value of *delta(high)/delta(low)* and the standardized beta value obtained from a fitting equation, the RSRS indicator represents relative strength. When the RSRS value is high, it indicates that the market has a greater difference of opinion regarding the support level than the resistance level, and the market tends to fall. Conversely, when the RSRS value is low, it suggests that the market has a higher level of acceptance of the support level than the resistance level, and the market is more likely to rise.

As SMA, MACD, and other indicators used to determine long-term trends still exhibit noticeable lag, the RSRS indicator is more sensitive to changes in market conditions and can serve as a leading indicator to quickly predict market outcomes. Therefore, in combination, we select the RSRS strategy as a short-term momentum strategy to capture entry and exit conditions that long-term momentum strategies may have missed, providing a more comprehensive analysis.

Another part of the short-term timing strategy is the multi-factor strategy, which is relatively simple and intuitive. It is calculated by using different stock data, such as opening price and trading volume. We have chosen two factors that meet our requirements: Alpha-006 and 018.

2. Alpha-006 Indicator

The Alpha-006 Indicator is presented as the following formula:

$$\alpha 006 = -1 * Correlation(close, volume, n)$$

The correlation coefficient of two random variables *x,y* in the past n days, whose value range is [-1,1]. For this factor, it can be interpreted as the daily opening price of the stock and the number of correlation coefficients in the last n days. Where n is an uncertain value. An optimal observation period will be found later based on the results traversed out.

First, the correlation between the opening price and the volume for the past n trading days is calculated. Where, before n days (except for the first day), the correlation refers to the correlation of all days before that day.

The lower the correlation, the more the opening price and volume tend to diverge. The final result is then added with a negative sign, which means that a lower factor value results in a lower correlation between the stock's opening price and volume.

We observed a phenomenon in securities trading when the price of a security has a new peak, the volume does not increase but starts to decrease. This implies that the price of the security is not proportional to the volume. In other words, the price of the stock has fallen to its lowest point and is about to end its downtrend and start an uptrend (Garfinkel & Sokobin, 2006). This is a buy signal and the investor may consider going for a certain number of shares. Conversely, the stock's volume continues to fall as the stock price continues to stage higher. This means that the stock price reaches its peak and the uptrend is about to end, which is a sell signal. When the above two phenomena are not obvious, it means that the stock has not exhibited a definitive buy or sell signal. As a result, investors may choose to refrain from trading the stock until a noticeable change occurs.

As can be seen from the above description, we need to set some values to measure the magnitude of its correlation. We decided to set a threshold value.

(1) When the value of the indicator for the day is greater than the threshold, it means that the stock price has fallen to its lowest point and is about to end the downtrend soon. At this point, we should long.

(2) When the value of the indicator for the day is less than the threshold, it means that the stock price has peaked and is about to end the uptrend. At this point, we should short.

(3) When the factor value of the day is exactly equal to the threshold, there is no clear signal to buy or sell. At this point we should not trade, so set the position to 0. In addition, the threshold is not a random value and needs to be traversed to find the optimal threshold to use.

3. Alpha-018 Indicator

The Alpha-018 Indicator is presented as the following formula:

$$\alpha 018 = -1 * rank\big(sd(abs(close - open), n = 5) + (close - open) + correlation(close, open, n = 10)\big)$$

In addition,

$$rank018 = \frac{\alpha 018}{total\ volume}$$

$rank(x)$ calculates the ranking of the stock. It gets the ranking of input value x. If the input value is null, the input value will not be included in the ranking.

$abs(x)$ calculates the absolute value. If the input value x is the difference between the closing price and the opening price of a stock, *abs(x)* will be the amplitude of the opening price and the closing price.

$sd(x, d)$ calculates the standard deviation of the past d days' x value. If x represents the daily price amplitude of the stock, *sd(x,d)* will be the *d* days' of the standard deviation of the price amplitude.

$correlation(x, y, d)$ calculates the correlation coefficient of the time series *x* and *y* in the past *d* days. For example, we assume that *x* and *y* correspond to the closing price series of two stocks. If the last 10 days' closing prices of these two stocks go up or down simultaneously, they will correspond to the correlation coefficient correlation *(x, y, 10) = 1*.

The Alpha-018 formula comprises three variables: the standard deviation of the absolute value of the intra-day spread (the difference between the closing and opening prices) over the past five days, the intra-day spread, and the correlation of the intra-day spread over the past 10 days. These variables are summed together. The intra-day spread plays a dominant role, and when the spread becomes excessively large, the corresponding value of the Alpha-0018 factor decreases (due to the negative sign multiplied in front).

The ranking calculated by function *rank()* is divided by the total number of stocks to obtain the corresponding ranking. Stocks with low correlation between closing and opening prices, small intra-day spreads, and small price changes tend to have higher rankings, and vice versa.

The Alpha-018 ranking logic is longing the stocks with low correlations between closing and opening prices, small intra-day spreads, and small price changes. We divide the ranking into three classes:

(1) If the stock ranks in the top 20 percent, it will imply that the stock has a low correlation. We choose to go long with a certain number of stocks.

(2) Stocks ranked between the top 20 and 50 percent are also suitable for long positions. However, it is slightly worse than the stocks ranked in the top 20. We chose to go long half of their positions.

(3) If the stock ranks after 50 percent, it will be considered to be underperforming. It implies that the stock has a high correlation. However, since the RSRS strategy decided to go long, we chose to go long a quarter of the initial volume of stocks.

```
if(rank018 <= 0.2){
  a006Pos[params$series[i]] <- round(1+alpha006) * posnorm[i]*multiplier_a
}
else if(rank018 <= 0.5 && rank018 > 0.2 ){
  a006Pos[params$series[i]] <- round(1+alpha006) * posnorm[i]*(multiplier_a/2)
}
else{
  a006Pos[params$series[i]] <- round(1+alpha006) * posnorm[i] *(multiplier_a/4)
}
```

Figure 1 - The Logic of Alpha-006 and Alpha-018 Indicators in Short-term Timing Strategy

Section 1.2 Consolidation Logic for Different Parts of the Strategy

The parallelism of the Long-term Momentum Strategy and the Short-term Timing Strategy is the key component of the strategy. For each series, the two techniques operate in parallel without interfering with each other. The basic logic of the Long-term Momentum Strategy is derived from the SMA, MACD and stochastic oscillator in Section 1.1, while the Short-term Timing Strategy incorporates the RSRS strategy and the multi-factor strategy.

```
#logic for rsrs & alpha006
  #Trade after lookback period
  if(store$iter > params$rsrs_lookback_m + params$rsrs_lookback){...}

#logic for dmaMACD
  #Trade after lookback Period
  if (store$iter > params$dmalookbacks$long && store$iter > params$macdlookback){...}
```

Figure 2 – The Parallelism of Long-term Momentum Strategy and Short-term Timing Strategy

Long-term Momentum Strategy
The long-term strategy will be implemented when the primary requirements are satisfied for a period of time longer than the lookback of the long-term SMA and the lookback of the MACD established in the parameters.

Short-term Timing Strategy
The short-term strategy will be executed when *store$iter* is greater than the observed days. Alpha-006 and RSRS are used in combination to determine the trading strategy. RSRS is used as the primary indicator to identify the investors' expectations toward the price trend, while Alpha-006 is used as a secondary indicator to confirm or contradict the signal given by RSRS and adjust the position size accordingly.

If RSRS indicates a negative signal toward the price trend ($rsrs\_z < 0.7$), the strategy will go short. On the other hand, if RSRS indicates a positive signal towards the price trend (*rsrs_z > 0.7*), the strategy will go long. In the long position scenario, if the Alpha-006 indicator confirms the positive price trend, the strategy will add more positions with different weights based on the ranking of Alpha-018.

```
#Investors have negative attitude towards price trend, go short
if (rsrs_z < 0.7){...}
#Investors have positive attitude towards price trend, go long
else if (rsrs_z > 0.7){

  rsrsPos[params$series[i]] <- round(rsrs_n * (posnorm[i]))* multiplier_r

  #rsrs limit orders stop loss
  if(as.numeric(op) <= as.numeric(mean_p) - as.numeric(stop)){...}
#If alpha006 further indicates that price trend is positive, go long further
  if (alpha006*100 > thr006){

    #If alpha018 ranking indicates that there is a top 20% ranking on positive price trend, add 100% more position
    if(rank018 <= 0.2){...}
#If alpha018 ranking indicates that there is a 20%-50% ranking on positive price trend, add 50% more position
    else if(rank018 <= 0.5 && rank018 > 0.2 ){...}
#If alpha018 ranking indicates that there is a under 50% ranking on positive price trend, add 25% more position
  else{...}
  }
}
#Investors' attitude towards price trend is unknown
else{...}
```

Figure 3 – Basic Logic of Short-term Timing Strategy

Overall, RSRS is used to capture the overall trend as quickly as possible, while Alpha-006 is used as a confirming indicator to identify the strength of the trend and adjust the position size accordingly. This approach aims to reduce risk and increase the potential for profit by combining the signals from multiple indicators.

Position Consolidation

The execution requirements of market orders vary for each of the three sub-strategies. RSRS strategy and multi-factor strategy need to clear the daily total position, while the Long-term Momentum Strategy is required to stack daily positions. Therefore, we need to set multiple position pools for each sub-strategy. However, there are certain stacking logic flaws that are not addressed effectively in the part 2 dataset, which will be evaluated in depth and applied for optimization in section 3.5.

```
#Initialize market orders and limit orders
marketOrders <- allzero

limitOrders1=allzero;
limitPrices1=allzero;
limitOrders2=allzero;
limitPrices2=allzero;


#Initialize dmaMACD strategy's market position
dmaPos <- allzero

#Initial position for rsrs & alpha006
rsrsPos <- -currentPos
a006Pos <- -currentPos
```

Figure 4 – Initialization of Positions

Each of the three independent sub-strategies is allocated its own pool of trading market orders, ensuring that their trades are not influenced by each other when the strategies enter, exit or stop loss. As a consequence, the combined strategy's total market order positions are overlaid on the three strategies' sub-positions. In the framework of this project, the position of each strategy is initialized in the *initStore()* and updated in *updateDmaPos()*, *updateRsrsPos()* and *updateAlpha006Pos()* respectively.

```
#Update positions from the above strategies
store <- updateDmaPos(store, dmaPos)
store <- updateRsrsPos(store, rsrsPos)
store <- updateAlpha006Pos(store, a006Pos)

#Sum to get the final market order positions
marketOrders <- marketOrders + a006Pos + dmaPos + rsrsPos

#dmaPos records the position of dmaMACD strategy
updateDmaPos <- function(store, dmaPos) {
  store$dmaPos <- dmaPos+store$dmaPos
  return(store)
}
#rsrsPos records the position of rsrs strategy
updateRsrsPos <- function(store, rsrsPos) {
  store$rsrsPos <- rsrsPos
  return(store)
}
#a006Pos records the pos of Alpha006 strategy
updateAlpha006Pos <- function(store, a006Pos) {
  store$a006Pos <- a006Pos
  return(store)
}

#Initialize store
initStore <- function(newRowList,series) {
  return(list(iter=0,cl=initClStore(newRowList,series),
              vol=initVolStore(newRowList,series),
              high=initHigh(newRowList,series),
              low=initLow(newRowList,series),
              ope=initOpeStore(newRowList,series),
              dmaPos=rep(0,ncol= length(params$series)),
              rsrsPos=rep(0,ncol= length(params$series)),
              a006Pos=rep(0,ncol= length(params$series))
             ))
}
```

Figure 5 – Storage and Update of Sub-positions

Section 1.3 Optimization and Robustness Testing

- Parameter Optimization

For the Long-term Momentum Strategy, the parameter optimization of individual strategies is based on cross-validation of in-sample and out-of-sample data (Berrar, 2019). Since the strategy focuses on long-term trends, the long-term lookback is chosen within 80-100 days because of trendy smoothness. The short-term lookback timings are chosen as 1/4 of the long-term timings owing to trendy volatility (Pavlov & Hurn, 2012).

The observed date determines the data time period on which the calculation of the

Alpha-006 value is based. In other words, it determines how many days we will observe the stock transaction volume and price in the last few days to judge the deviation of volume and price.

```
#For the days that before observe days
if(store$iter<=params$obday){
VOLUMELIST <- VOLUME[0:store$iter]
CLOSELIST <- CLOSE[0:store$iter]
}

#After the specified days, store the most recent n days' data
#n is the observed day and is passed in through parameter "obday"
else if(store$iter>params$obday){
VOLUMELIST <- VOLUME[as.numeric(store$iter-params$obday):store$iter]
CLOSELIST <- CLOSE[as.numeric(store$iter-params$obday):store$iter]
}
```

Figure 6 - Part of the Codes that Includes Parameter "obday"

The threshold determines where our strategy further goes long when the Alpha-006 value rises, under the circumstance that the RSRS strategy signals that investors have a positive attitude toward the price trend.

```
#If alpha006 further indicates that price trend is positive, go long further
if (alpha006*100 > thr006){
#If alpha018 ranking indicates that there is a top 20% ranking on positive price trend, add 100% more position
  if(rank018 <= 0.2){
  a006Pos[params$series[i]] <- round(1+alpha006) * posnorm[i]*multiplier_a
}
......
}
```

Figure 7 - Part of the Codes that Includes Parameter "thr006"

Both parameters are called from the *params* vector under *example_strategies.R*, and we need to find a way to search for the optimal parameter value. Our first thought is to add a loop inside our strategy to simulate the final P/D ratio under each parameter value to find the optimal. However, we came across a "predict the future" paradox: the whole 1000-day data is needed to simulate inside the strategy itself, while our strategy loop always runs within the 1000-day data. That circumstance is interpreted in coding as the *getOrder()* function and *backtest()* function call each other inside and cause infinite recursion. To avoid this mistake, We ran loops to test parameter values in *main.R* and record the profit result using in-sample data. This part is not shown in the submitted code as it changes the main function and the entire execution framework.

```r
#Initialize the threshold
thr006Vector = 60:70
obdayVector = 25:35

x <- c()
y <- c()
z <- c()

#Loop through the threshold
for (i in thr006Vector){
for (j in obdayVector){
thr006 <- i
obday <- j
results <- backtest(dataList,getOrders,params,sMult)
pfolioPnL <- plotResults(dataList,results,plotType='ggplot2')

x <- append(x,thr006)
y <- append(y,obday)
z <- append(z,pfolioPnL$fitAgg)
print(paste('threshold', i, 'obday', j, 'value', pfolioPnL$fitAgg))
}
}

#Store the threshold and its performance into a dataframe
#Output the data
results <- backtest(dataList,getOrders,params,sMult)
plotResults(dataList,results)
pfolioPnL <- plotResults(dataList,results,plotType='ggplot2')
cat("Profit:", results$aggProfit, '\n')

Mydata <- data.frame(x,y,z)
write.csv(Mydata, file = "Mydata.csv", row.names = FALSE)
Parameter Optimization in Main.R
```

Figure 8 – Parameter Optimization

The above codes loop the value of two parameters from one to one hundred and create a csv file in the default dictionary, recording the overall P/D ratio and corresponding parameter value combination. What we should do is just look up the csv file and seek the parameter combination with the highest P/D ratio. In-sample testing data is used in the parameter optimization part. We found the optimal threshold as 65 and the optimal observe day as 30. These two parameters value not only achieved the greatest benefit in the in-sample test, but also performed well in the out-sample test.

It is worth noting that in the above code, the transmission of parameters between R files has changed, so it cannot be directly added to *main.R* to run successfully. Parameter processing in the strategy file and *example_strategies.R* should also be modified correspondingly. It is also worth noting that we divided the work and tested different parameter value ranges in parallel. We gradually reduced the parameter range by observing the distribution of the results. The parameter testing range *thr006 60:70* and *obday 25:35* is the idea range we have found through previous works. If we directly set the parameter range as both 1 to 100, it shall test ten thousand combinations and take approximately two years until we get the result.

- **In-sample and Out-sample Tests**

We separated part one and part two data into three parts. The whole part 1 data (1100 days) was set as an initial test before acquiring the second part data. After obtaining the second part data, we further combined the first 400 days' data of the second part with the first part data as the in-sample testing set. The remaining 700 days' data of the second part is divided as the out-sample testing set.

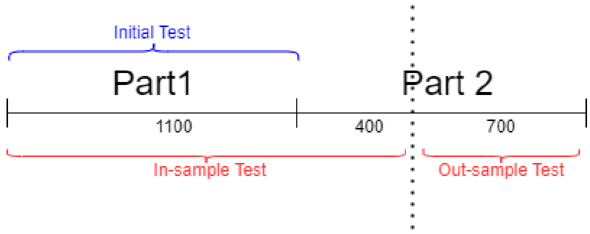

Figure 9 - Data Separation

We used the initial test to set the appropriate range of parameters and tested the approximate performance of the strategy. If there was a large area of loss or liquidation at this stage, we would change the parameters on a large scale, or even adjust the underlying logic of the strategy. In this way, we can prevent poor parameters and inappropriate strategy logic from entering the in-sample test with a large amount of data to save debugging time. We continued to fine-tune the parameters according to the results of the in-sample test and selected the best results for the out-sample test. If the out-sample test continued to perform well, we considered the strategy generally qualified for submission.

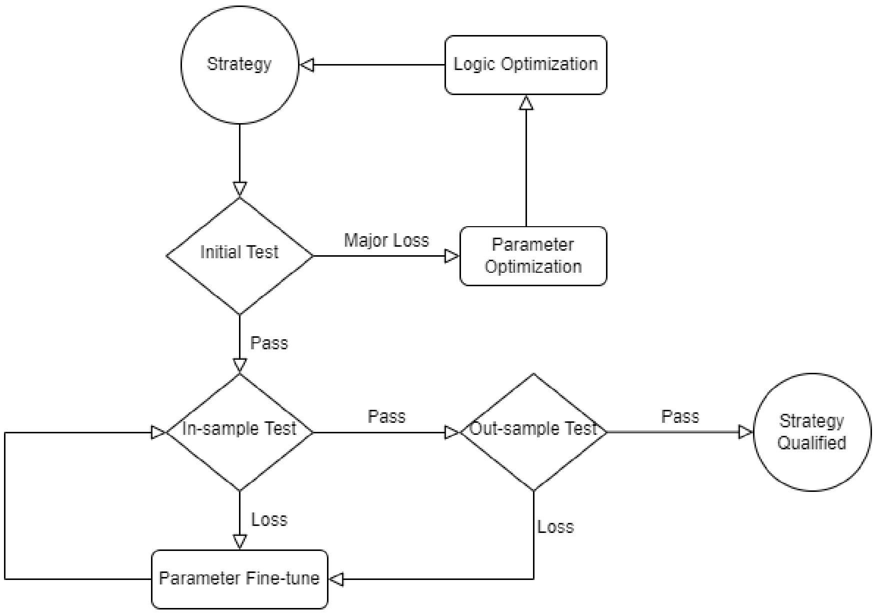

Figure 10 – Testing Framework

We considered that such a three-step testing strategy can help us prevent over-fitting, and reduce the time for optimizing logic and finding the best parameters.

- **Robustness test**

The first step in this technique for the robustness test is to extend the test data. We utilized financial data from the *quantmod* tool in R in addition to the three sets of data provided by the teacher for distinct periods. To broaden the test data, both in-sample and out-sample procedures are utilized. Except for the ten-series data, this method allowed us to determine whether there were any major issues with this merging procedure. This reference could help us better define the strategy’s characteristics to add value to the overall strategy, as well as consider whether the strategy's position management and capital allocation are reasonable. It allowed us to make appropriate adjustments to handle more unknown situations.

Section 2. Justification of Submitted Strategy

The previous section presents an executive summary of the components of the overall consolidation strategy and its optimization. Based on the introduction in section 1, we will discuss in detail the reasons for selecting the strategies and combination in this section.

Section 2.1 Reasons for the Decision to Merge

- **Specific Reasons for Choosing Strategies**

1. The Long-term Momentum Strategy is able to grasp the overall market dynamics in general when there is a clear trend and the buying and selling effect is significant and more profitable.

2. The RSRS strategy serves as a flexible indicator of the short-term trend to capture more detailed profit space.

3. The Multi-factor Strategy is able to judge trades from a volume and price perspective.

Therefore, these three strategies represent *long-term stability*, *short-term flexibility* and *volume and price divergence*.

- **Reasons and Motivation for Merger**

  1. **Why combine RSRS and multi-factor strategy?**

  We named the combination of the RSRS strategy and the multi-factor strategy as the Short-term Timing Strategy. RSRS is used to identify market sentiment and investor attitudes toward price trends in the short term. Its essence is a flexible trend-following strategy. However, back-testing has shown that the timing signal of RSRS is less stable during oscillating markets. When experiencing an oscillating market outside the sample, the number of opening and closing trades of the RSRS signal will increase significantly. At this time, the stability of the signal will decrease, and the signal winning rate will also decrease significantly.

  When analyzing price patterns, trading volume is considered an important indicator for verification. Without the support of trading volume, price patterns are unreliable. In section 1.1, we introduced Alpha-006 as a volume-price combination factor that measures the degree of deviation or alignment of volume and price trends through their correlation. Essentially, it is a mean reversion strategy. Therefore, we attempt to combine these complementary concepts of RSRS and Alpha-006 to achieve a blunt effect on the indicator values in volatile markets, which can reduce the extent of strategy losses during oscillating periods. Empirical results of visualizing position and trading results demonstrate that the RSRS combined with Alpha-006 produces better results.

2. Why combine long-term momentum and short-term timing strategy in parallel?

Firstly, the Long-term Momentum Strategy is utilized to determine market trading directions. It can grasp the entry and exit under the basic trend in the long term. However, SMA, MACD and other averaging indicators still lag in the long-term situation. In addition, the simultaneous operation of the Short-term Timing Strategy allows for flexible timing of trades and timely trading. It prevents oversights caused by the lagging nature of the long-term strategy.

3. Why integrate SMA, MACD and stochastic oscillators in the Long-term Momentum Strategy?

It is apparent from the introduction of SMA and MACD indicators in section 1.1 that the combination of these two indicators may more properly estimate the timing of trades, particularly through MACD to offset the considerable lag of SMA. The primary logic for calculating entrance and departure circumstances is as follows:

- If the short-term SMA crosses the long-term SMA and the *macd* in the MACD crosses *signal*, it represents an uptrend and it is time to buy additional positions depending on the situation.

- If the short-term SMA crosses the long-term SMA and the *macd* also crosses *signal*, it represents a downtrend. It is time to short positions depending on the situation.

Secondly, as section 1.1 introduces the stochastic oscillator which calculates KDJ indicator, it is a stop-loss indicator that assists this strategy in oscillating markets. This indicator was chosen for the following reasons:

- With the position visualization from section 3.3, we found that its trading signals of market orders in the oscillating market have a dominant error from the expectation, and therefore there is a large risk of trade loss. Consequently, the purpose of using the fast-stochastic indicator (named *K*) is to quickly determine the position of the stock price in the recent market.

```
#Calculate KDJ indicator for stop loss
KDJ<- as.data.frame(last(stoch(store$cl[startIndexkdj:store$iter,i],nFastK = params$nFastK,
                                nFastD = params$nFastD,
                                nSlowD = params$nSlowD, bounded = TRUE,smooth = 1)))

#Calculate the assignment interval of the K indicator
Kline <- KDJ$fastK * 100
```

Figure 11 – Stochastic Oscillator Indicator

- The basic logic is that if the SMA and MACD indicators represent a buy signal for the market, investors long positions based on the signal's instructions. However, if the market soon goes on a downward trend, the price will fall through the short-term SMA. The fast-stochastic indicator will also appear to cross below 50. Therefore, it indicates that the trend may not continue. In this situation, it is better to stop loss.

- **Series Choice**

All strategies are chosen to run in each series to keep the principle of universality of this strategy. However, we wanted to filter the different characteristics of the series by code in *getOrders()* in order to apply them. Due to technical limitations, this idea could have helped the strategy to trade more precisely.

Section 2.2 Justification of Position Sizing Choice

- **Position standardization for overall strategy**

The knowledge from COMP396 lecture 4 (Fearnley, 2022) has been used for the standardization of positions in the overall strategy. The average value of each series close price is calculated in a for loop and the largest close price of all series is traversed. Since the data of the series vary from datasets, we choose to find the normalization parameter in the *getOrders()* equation to achieve generality.

```
#Position normalization
clall <- matrix(store$cl,ncol = length(params$series))
closeprice <- apply(clall,2, function(x) mean(x[x>0]))
largestclose <- max(clall)
posnorm <- round(largestclose/closeprice)
```

Figure 12 – Position Normalization

- **Capital Allocation**

The first thing we took into account for the allocation of funds is the characteristics of each strategy. We prioritized the Long-term Momentum Strategy since it demonstrated more consistent performance and better returns in the in-sample test. The technique of using the funds is also based on our understanding of COMP396 lecture 4 (Fearnley, 2022). We calculated the probable cost by standardizing the positions and dividing the supplied funds to produce the dynamic multiplier of the positions after funding allocation. Using this method, we could allow more money feasible to be spent.

```
#Position sizing and capital allocation
estCostToBuy <- sum(posnorm * closeprice)
targetspent_a <- 300000
targetspent_r <- 50000
multiplier_a <- targetspent_a/estCostToBuy
multiplier_r <- targetspent_r/estCostToBuy
```

Figure 13 – Capital Allocation for Short-term Timing Strategy

```
#Position Sizing & Capital Allocation
estCostToBuy <- sum(posnorm1 * closeprice)
targetspent <- 450000
multiplier_d <- targetspent / estCostToBuy
```

Figure 14 – Capital Allocation for Long-term Momentum Strategy

- **Position Personalized Adjustment**

We already have the position standardization and capital allocation described above. However, each strategy's characteristics are different. Therefore, we decided to perform dynamic management of positions based on strategy characteristics to ensure that longing and shorting trade sizes could better match the features of the strategy and maximize the benefits.

1. **Long-term momentum strategy**
   It sets a factor that indicates the transaction's intensity. The difference between the short-term and long-term SMAs must be determined in the present scenario. If the gap between the short-term SMA and the long-term SMA is higher, there will be a stronger uptrend (or downtrend) and more positions will be formed to long (or short) based on this component.

```
((abs(short_ma - long_ma)*multiplier_d)/long_ma)*(posnorm1[i])*100
```

Figure 15 – Position Management of Long-term Momentum Strategy

In addition, the Alpha-018 component mentioned in section 1.1 is used to improve the dynamic management of purchasing position size by ranking this factor (Syu, 2022). For example, if Alpha-018's ranking is less than or equal to 0.2, which means the rating of Alpha-018 shows that there is a top 20% ranking on a positive price trend, 100% of the buy position will be kept. If Alpha-018's ranking is between 20% and 50%, the position should be dropped at the appropriate period, and so on.

```r
#If alpha018 ranking indicates that there is a top 20% ranking on positive price trend, add 100% more position
if(rank018 <= 0.2){
  dmaPos[params$series[i]] <-  abs(short_ma - long_ma)*multiplier_d/long_ma*(posnorm1[i])*100
}

#If alpha018 ranking indicates that there is a 20%-50% ranking on positive price trend, add 50% more position
else if(rank018 <= 0.5 && rank018 > 0.2 ){
  dmaPos[params$series[i]] <-  (abs(short_ma - long_ma)/long_ma)*100*posnorm1[i]*(multiplier_d/2)
}

#If alpha018 ranking indicates that there is a under 50% ranking on positive price trend, add 25% more position
else{
  dmaPos[params$series[i]] <-  (abs(short_ma - long_ma)/long_ma)*100*posnorm1[i]*(multiplier_d/4)
}
```

Figure 16 – Position Management using Alpha-018

2. **Short-term Timing Strategy**

The RSRS strategy is based on standardization and capital allocation multiplied by *rsrs_n*, which reflects the slope of n days. The bigger the slope of n days, the more probable a trend will emerge, and the stronger the trend, the more investors are likely to recognize it.

```r
rsrsPos[params$series[i]] <- round(rsrs_n * (posnorm[i]))* multiplier_r
```

Figure 17 – Position Management for RSRS Strategy

For the multi-factor strategy, it is multiplied by the factor *(1+alpha006)*. Because Alpha-006 represents the degree of volume divergence, if the value is higher, then the higher the likelihood of volume divergence mean-reversion and therefore should be bought more, in line with the strategy logic.

```r
a006Pos[params$series[i]] <- round(1+alpha006) * posnorm[i]*multiplier_a
```

Figure 18 – Position Management for Multi-factor Strategy

Section 2.3 Other Key Elements

- **Store (Position Update)**

The position variable is updated by each strategy function (*updateDmaPos()*, *updateRsrsPos()*, and *updateAlpha006Pos()*) based on their own rules for entering and exiting positions. The *store* object holds the position for each strategy, allowing the algorithm to keep track of multiple positions at once.

The *updateDmaPos()*, *updateRsrsPos()*, and *updateAlpha006Pos()* functions are used to update the position of each strategy in the *store* object. The *dmaPos*, *rsrsPos*, and *a006Pos* variables within the *store* object hold the current position for the corresponding strategy. We can make sure that, for instance, if the stop loss of the momentum strategy is triggered, only the position of the momentum strategy will be cleared and the positions of the other two strategies will not be affected. Besides, the resulting positions can be used to calculate the profit and loss for the previous day (*pnl_yesterday*) and to set new positions based on the current day's indicator values.

```r
#Initialize store
initStore <- function(newRowList,series) {
  return(list(iter=0,cl=initClStore(newRowList,series),
              vol=initVolStore(newRowList,series),
              high=initHigh(newRowList,series),
              low=initLow(newRowList,series),
              ope=initOpeStore(newRowList,series),
              dmaPos=rep(0,ncol= length(params$series)),
              rsrsPos=rep(0,ncol= length(params$series)),
              a006Pos=rep(0,ncol= length(params$series))
             ))
}

#dmaPos records the position of dmaMACD strategy
updateDmaPos <- function(store, dmaPos) {
  store$dmaPos <- dmaPos
  return(store)
}
#rsrsPos records the position of rsrs strategy
updateRsrsPos <- function(store, rsrsPos) {
  store$rsrsPos <- rsrsPos
  return(store)
}
#a006Pos records the pos of Alpha006 strategy
updateAlpha006Pos <- function(store, a006Pos) {
  store$a006Pos <- a006Pos
  return(store)
}
```

Figure 19 – Store Management

- **Pnl_yesterday**

In this strategy, *pnl_yesterday* represents the profit or loss from the previous day's trading activities for a particular security.

The calculation of *pnl_yesterday* is according to *backtester.R* framework, which takes into account the position size, the difference between today's open price (*next_open*) and yesterday's close price (*cur_open*) and the slippage cost (*slippage*) incurred while trading the security.

```r
# run from day 2, where oldPos would always be 0, until penultimate day
slippage <- slip(prices$prevCl, prices$curOp, sMult)

# +/- (nextOp - curOp) * "held on cur" - slippage * "traded on cur"
pnl <- pnl + marketOrder * (prices$nextOp - prices$curOp) - abs(marketOrder) * slippage
```

Figure 20 – Pnl_yesterday Calculation

pnl_yesterday can be used to determine the current level of losses and adjust position sizes accordingly.

- **ATR and Limit Order**

ATR stands for Average True Range, which is a technical indicator used to measure market volatility (Yamanaka, 2015). It is often used to help traders identify the potential magnitude of price movements. Using ATR to set stop-loss levels allows traders to adjust the stop-loss level according to the current volatility of the market.

The ATR is calculated by taking the average of the True Range (TR) over a specified number of periods. The True Range is the greatest of the following:

- The difference between the current high and the previous close
- The difference between the current low and the previous close
- The difference between the current high and the current low

Then, we calculate the average of the True Ranges over a specified number of periods. Here, we define *Cal_ATR* to get the value of ATR and use the period of 18 days.

```
Cal_ATR <- function(HIGH,LOW,CLOSE,startIndex,lookback,iter){
nTR <- 0

for( i in startIndex+1:iter){
  #52~69
  TR <- max((HIGH[i]-LOW[i]), abs(HIGH[i]-CLOSE[i-1]), abs(CLOSE[i-1]-LOW[i]))
  nTR <- nTR + TR
}

ATR <- nTR/lookback
return(ATR)
}
```

Figure 21 – ATR Calculation

The RSRS strategy uses the ATR to determine the stop loss price for a trade. Once a trade is entered, a limit order is placed at the stop loss price determined by the ATR. This means that if the price reaches the stop loss level, the limit order will be triggered, and the position will be closed automatically.

Section 2.4 Alternatives and Comparison

- **Indicator Alternatives for Multi-factor Strategies**

When we designed our multi-factor strategy, we initially designed it with three factors and wanted to long or short when the conditions of all three factors met. We later found that such a design would decrease the strategy's active days. Also, we cannot simply combine them as there is some correlation between the three factors, which would affect each other's results. Therefore, we decided to use only Alpha-006 as a separate strategy.

Alpha-018 and Alpha-034 are the other two factors that rank stocks. We decided to use them as indicators for dynamic position management. However, as there is some functional overlap between the two factors. Besides, Alpha-018 covers a wider range of conditions than Alpha-034, we ultimately decided to use only Alpha-018 as a factor for dynamic position management.

```
alpha034 = rank((1 - rank((runSD(RETURN, n=2) / runSD(RETURN, n=5)))) + (1-rank(diff(CLOSE))))

alpha018 = rank(runSD(abs(CLOSE-OPEN), n=5) + amplitude + runCor(CLOSE, OPEN, n = 10))
```

Figure 22 – Code Comparison between Alpha-034 and Alpha-018

- **Combined Strategy Alternative Schemes**

After we had completed the design of the three strategies, we were ready to carry out a merger of these strategies.

We systematically studied the RSRS strategy, which we initially considered to be mean-reverting. Therefore, we merged the RSRS strategy and the long-term momentum strategy as complementary. However, based on data analysis of the stocks, we found that the RSRS strategy is actually biased toward the short-term momentum strategy. As the RSRS factors are more flexible, we discarded the initial complementary idea and continued to merge the RSRS strategy with the long-term momentum strategy. Then, we only started the Long-term Momentum Strategy when the judgment conditions of RSRS were met, ensuring a more reliable combination of the two indicators. However, as the RSRS and Long-term Momentum Strategy are both trend-following, the direct combination of short-term and long-term strategies may have trading conflicts, i.e., there may be a situation where the RSRS indicator is judged to long while the long-term indicator is judged to short. Therefore, we consider applying Alpha-006 to strengthen the constraints to filter out potential false signals from the RSRS.

Section 2.5 - Risk Management

- **General Risk Management**

Our train of thought for general risk management was to simulate and monitor the previous day's rate of return and implement position restrictions on transactions that may currently have a risk of loss (Duran & Bommarito, 2011). We stored each day’s positions and simulate the transaction return (noted as *pnl*) of the previous day based on the positions, using the formula below:

$$pnl_y = pos_y * (open - open_y) - abs(pos_y) * slippage$$

Where,

$$slippage = abs(close_{yy} - open_y)$$

In this formula, *abs* is the abbreviation of absolute function and *pos* is the representation of position. The *subscript y* represents the value that belongs to yesterday (one day before the day that the strategy is looped through). The *subscript yy* represents the value that belongs to the date before yesterday (two days before the day that the strategy is looped through). Value without *subscript y* means that the value refers to today (exactly the day that the strategy is looped through).

If pnl_y is negative, which implies that yesterday occurs a loss, we will halve the trading position of the day to trade conservatively. This general risk management method avoids large-scale losses that are very likely to result in bankruptcy.

It is worth noting that our return simulation and monitoring are not target for the entire large position, but for the split small positions respectively. Our total position for market orders is the sum of three small positions, which are called *rsrsPos*, *dmaPos*, and *a006Pos*. These three small positions represent the execution result of three small strategies that are merged or paralleled. In this risk management, three small strategies monitor their returns without interfering with each other. This makes our trading decisions more flexible and objective, without greatly limiting the total trading volume because of overall short-term losses.

```
#Get indicators
HIGH = store$high[,i]
LOW = store$low[,i]
OPEN <- store$ope[,i]
CLOSE <- store$cl[,i]
VOLUME <- store$vol[,i]

#Get profit & loss for yesterday
prev_close <- CLOSE[store$iter-2]
cur_open <- OPEN[store$iter-1]
next_open <- OPEN[store$iter]

slippage <- abs(prev_close-next_open)*0.2
#run from day 2, where oldPos would always be 0, until penultimate day

pnl_yesterday3 <- store$dmaPos[params$series[i]] * (next_open - cur_open) - abs(store$dmaPos[params$series[i]]) * slippage

#General stop loss for dmaMACD
#Cut the position to half when yesterdays' simulation on return is negative
if (pnl_yesterday3<0){
dmaPos <- dmaPos/2
}
```

Figure 23 – An example of general risk management for Long-term Momentum Strategy

- Risk Management for Short-term Timing Strategy

In the given trading strategy, the RSRS indicator is used to identify potential long or short signals. However, to manage risks and minimize losses, the strategy also utilizes limit orders based on ATR.

Since it is not possible to monitor real-time prices, we roughly used the average of the highest and lowest prices of the day (*mean_p*) to represent the current price. We used a smaller ATR multiple (0.8ATR), which captures a smaller trend (shorter trading time). By multiplying ATR with a factor of 0.8, the strategy sets a stop loss level that is 80% of the ATR. If short, then adding 0.8 ATR from the current price is our trailing stop loss. Conversely, if long, the trailing stop loss is 0.8 ATR minus the current price.

When the RSRS indicator suggests a buy or sell signal, the strategy calculates the current ATR value and sets a stop loss level based on it. If the current price falls below this stop loss level, a limit order is triggered to exit the position and minimize losses. For example, in the code snippet below, if the RSRS indicator suggests a long position and the current price falls below the stop loss level (*mean_p - stop*), a limit order will be placed to exit the long position:

```
#Investors have positive attitude towards price trend, go long
else if (rsrs_z > 0.7){
  rsrsPos[params$series[i]] <- round(rsrs_n * (posnorm[i]))* multiplier_r

#rsrs limit orders stop loss
  if(as.numeric(op) <= as.numeric(mean_p) - as.numeric(stop)){
    limitOrders2[i] <- -rsrsPos[params$series[i]]
    limitPrices2[i] <- mean_p - stop
  }
}
```

Figure 24 – RSRS Risk Management for Long

Similarly, if the RSRS indicator suggests a short position and the current price rises above the stop loss level (*mean_p + stop*), a limit order will be placed to exit the short position:

```
#Investors have negative attitude towards price trend, go short
if (rsrs_z < 0.7){
  rsrsPos[params$series[i]] <- -round(rsrs_n * (posnorm[i])) * multiplier_r

  #rsrs limit order stop loss
  if(as.numeric(op) >= as.numeric(mean_p) - as.numeric(stop)){
    limitOrders1[i] <- rsrsPos[params$series[i]]
    limitPrices1[i] <- mean_p + stop
  }
}
```

Figure 25 – RSRS Risk Management for Short

By using ATR to calculate the stop loss level and limit orders to exit the position, the strategy aims to manage risk and avoid significant loss in case the trade doesn't go as planned.

- **Risk Management for Long-term Momentum Strategy**

The specific stop-loss approach for this strategy is to prevent large losses due to rapid market shocks. As described in section 2.1 in relation to the stop-loss logic of the stochastic oscillator in the long-term momentum strategy, the stop-loss for this strategy is a limit order. This indicator realizes stop-loss through limit orders, initially taking into account that limit orders can limit the price of the stop loss, to avoid over-sensitive indicators and missing the time to earn money. However, this idea has been tested with many logical bugs and mistakes, resulting in poor back-testing results.

Section 3 - Evaluation and Analysis of Performance on Part 3

In this section, we will provide an in-depth analysis of the performance of our strategy in Part 3. We will present a critical assessment and analysis: We will summarize the reasons that led to the unsatisfactory results. In addition, we will evaluate the errors at the technical level and at the design stage. With this module, we have got a huge improvement, both in terms of technical aspects and in terms of group work.

3.1 Expected Result

The strategy generally performs well with the stock data we have. It achieves a profit of approximately $140 thousand with a 4.16 Profit-to-Drawdown ratio in part one data. It also achieves a profit of $80 thousand with a 1.2 Profit-to-Drawdown ratio in part two data. The profit curve is stable and positive without major retracements and fluctuations. The expected outcome is shown below.

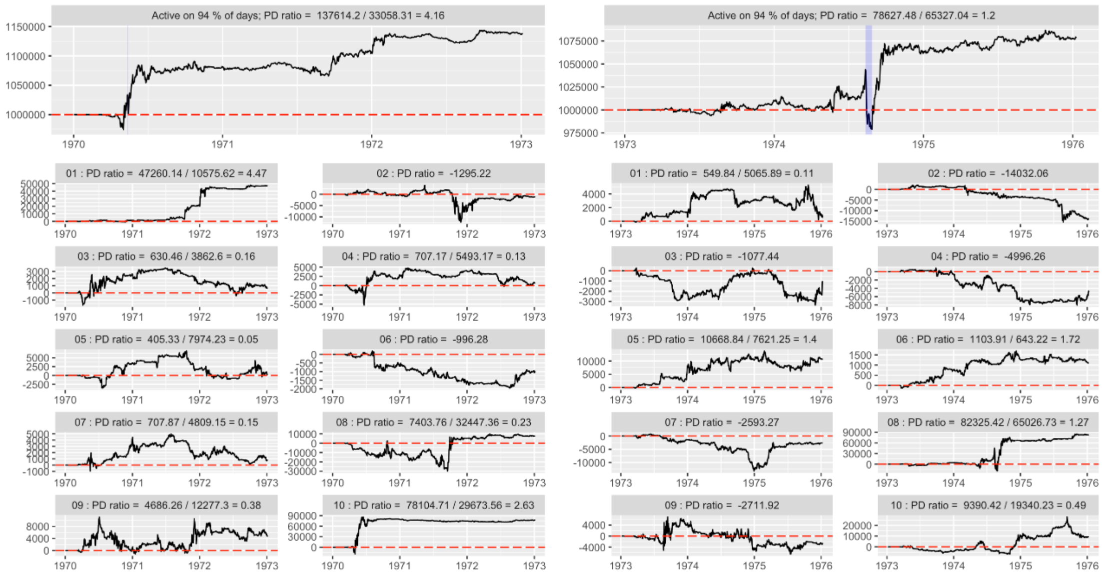

Figure 26 – Strategy Expected Performance on Part 1 (Left) and Part2 (Right) data

The strategy also passes our initial test and has a good performance during the out-sample test as well. The out-sample dataset is the split last 700 day-data in part two. Using the best-performing parameters in the in-sample test, the strategy achieves a profit of approximately 85 thousand with a 1.46 Profit-to-Drawdown ratio in the out-sample test, which is shown below.

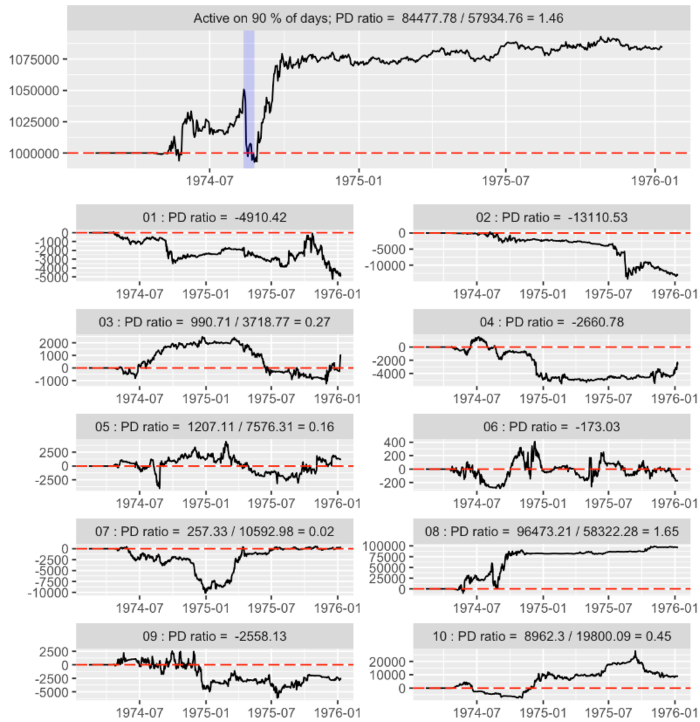

Figure 27 – Strategy Expected Performance on Out-sample Test

3.2 Results in Part 3.

However, our strategy does not perform well in part 3. It causes an overall loss of $75 thousand. This is not a satisfying result compared to what we expected in part 1 and part 2. We have analyzed the features of part 3 data and the strategy's trading position to find the reasons that caused this failure, which will be introduced in the next section.

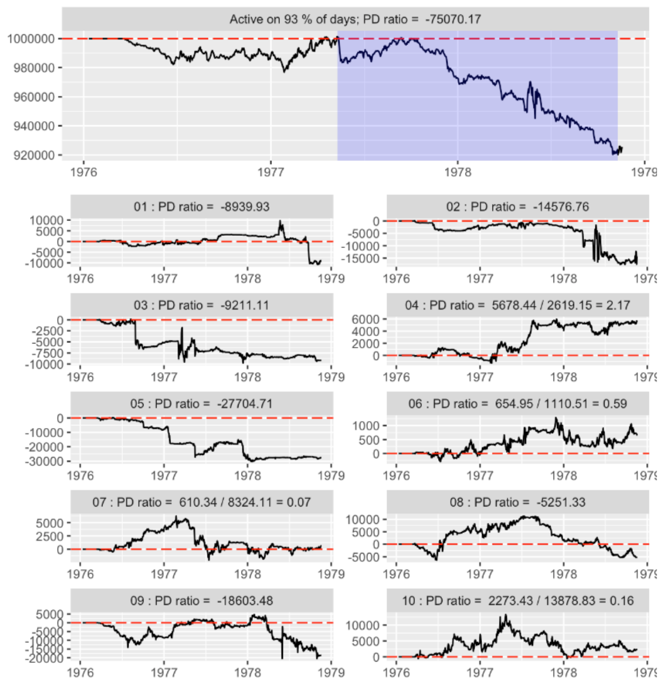

Figure 28 – Strategy Performance on Part 3 Data

3.3 Visualization and Reflections

We have conducted two kinds of visualization to explain the reason: stocks' trend visualization and position visualization.

- **Stock Visualization**

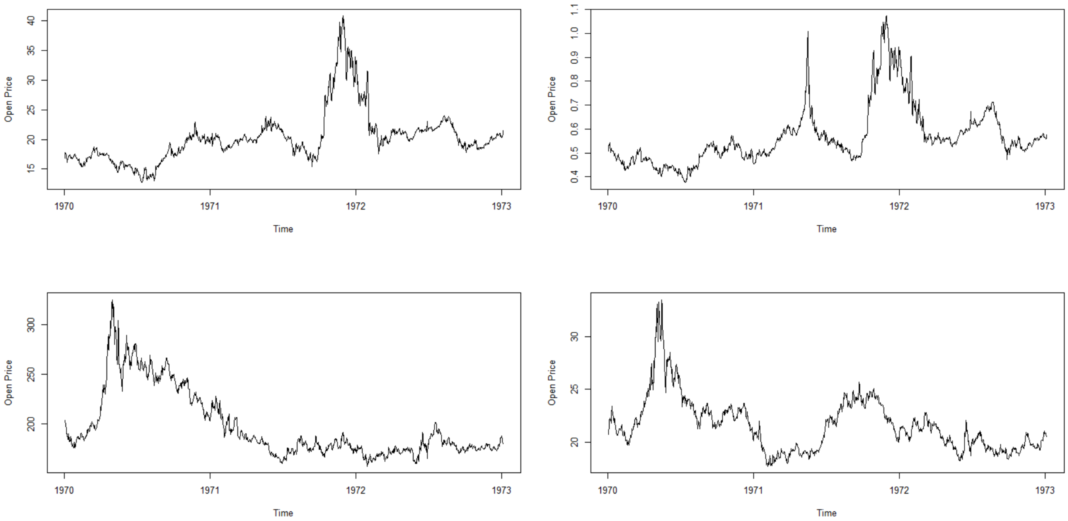

Figure 29 – Stock 1-4 Open Price Trend in Part 1 Data

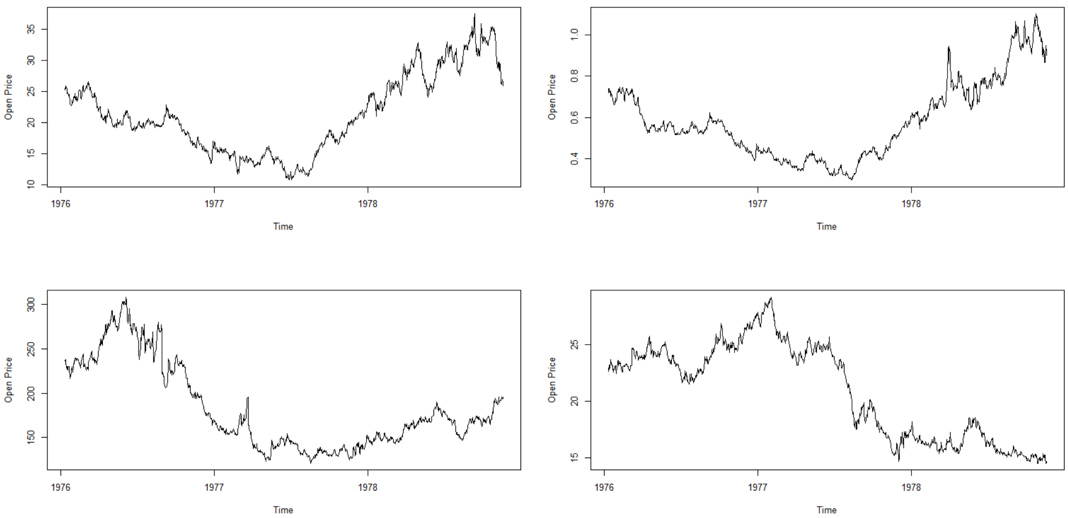

Figure 30 – Stock 1-4 Open Price Trend in Part 3 Data

It is clear that there exists at least one large price appreciation followed by an immediate large price depreciation in every stock of part 1. As both parallel component strategies, i.e., RSRS and long-term momentum strategy, are trend-following, this kind of large price crest allows both of them to make a large sum of money. This price characteristic is the major point that our strategy is greatly profitable with Part 1 and Part 2 data. It is also clear to find out that our strategy does not give good reflections to small price fluctuations. See the figure below.

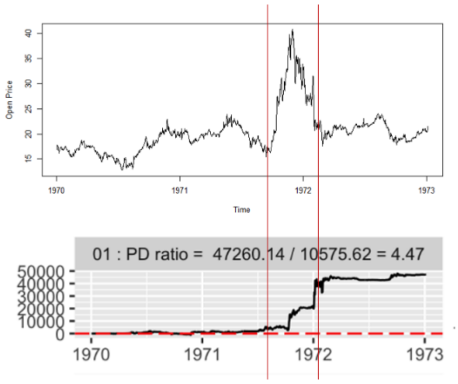

Figure 31 – Major Profit Section in Part 1 Stock 1

However, there does not exist a major price crest in part 3 stock data, and this feature does not allow our strategy to take the best advantage. On the contrary, part 3 data have very strong and continuous shocks. Our strategy will lose direction in such shocks, resulting in losses, as shown in the figure below.

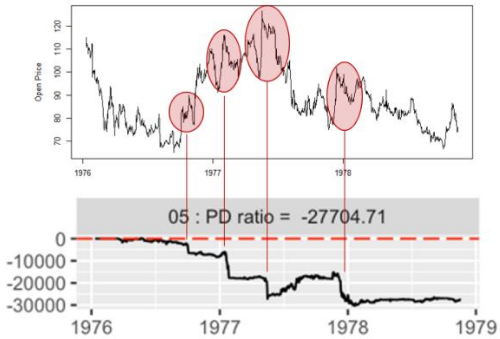

Figure 32 – Major Loss Section in Part 3 Stock 5

- **Position Visualization**

We have conducted position visualization to find if our strategy's performing direction fits the stock's trend.

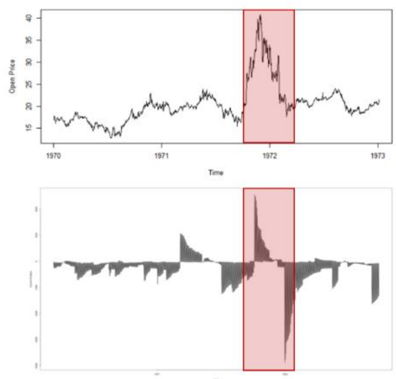

Figure 33 – Stock Trend (Up) and Position (Down) Comparison of Part 1 Stock 1

The position comparison chart shows that our strategy responds correctly to most part 1 and part 2 stock trends. The highlighted part of the figure shown above shows that our strategy responds quickly and strongly to stock price peaks with significant trends. During the crest period of part 1 stock 1, our strategy goes long a large sum of money when the price rise is obvious, and goes short a large sum of money when the price fell significantly. This behavior gains a large profit ($47k) for stock 1.

However, as mentioned in the previous stock visualization part, part 3 stock fluctuates continuously and does not have significant price peaks. The position chart reveals that our strategy is not flexible in changing positions with part 3 stock data. The red-highlighted parts demonstrate that our strategy can react correctly during large trends. However, the green-highlighted part in the position chart shows in mid and late-1978, our strategy reacts very slowly to the price fall. By the time our strategy decides to go a large sum of short position, the stock price is already going up, which resulted in a large loss. Moreover, position sizing is also a problem as our strategy should not react to hold such a large amount of short position based on such a small price fall, which is also shown in the green-highlighted section. Detailed reasons that cause the error of slow reaction and wrong position sizing will be explained in the mistake section.

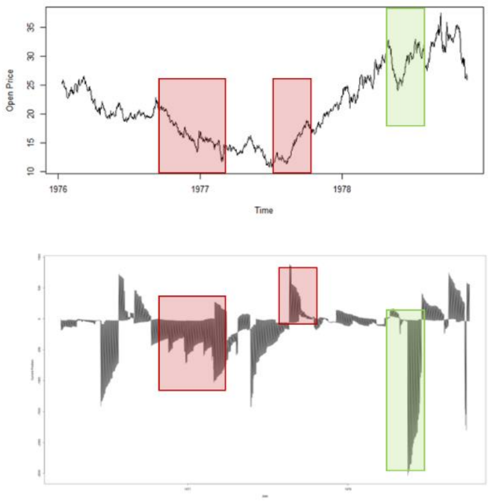

Figure 34 – Stock Trend (Up) and Position (Down) Comparison of Part 3 Stock 1

3.4 Strategy Division and Reflections

In this part, we divide our strategy into three small strategies. We test the three strategies separately and analyze the reasons why each strategy may lead to the poor result in part 3

- **Long-term Momentum Strategy**

For the long-term momentum strategy, part 3 performance is quite poor. This strategy was also one of the leading causes of the overall strategy performance loss. To begin with, because this is a momentum strategy, its performance is highly dependent on the presence of a trend in the stock. As section 3.1 analyses the overall strategy based on part 3 data, part 3 has a shorter and more volatile trend period. We have done a k-line chart visualization of part 3 and part 1 data in series1, which visually shows a more pronounced trend for series 1 data in part 1. Reasons for objective volatility were also a factor in the strategy's underperformance.

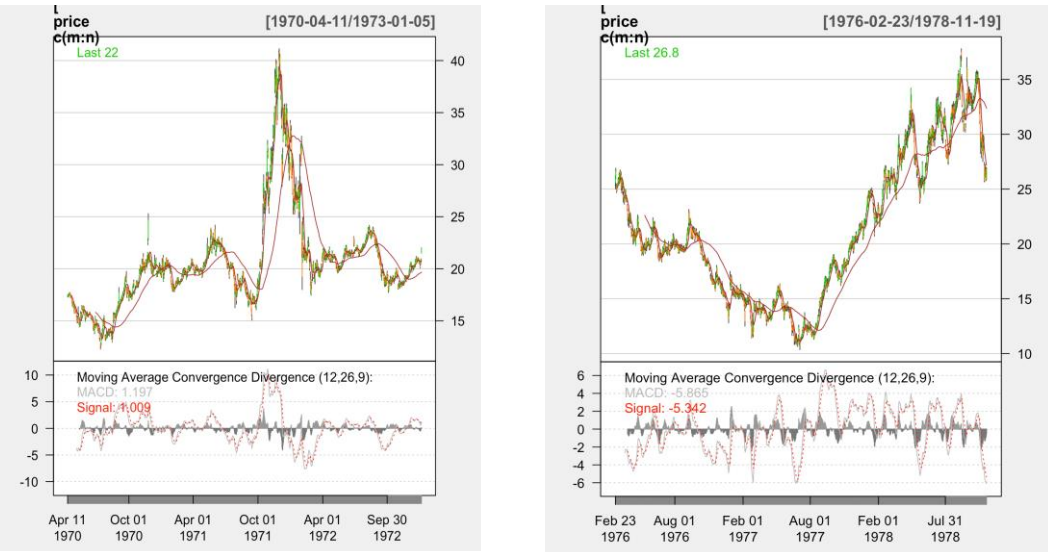

Figure 35 – Long-term Momentum Strategy K-line Chart on Part 1 Series 1 (left) and Part 3 Series 1 (right)

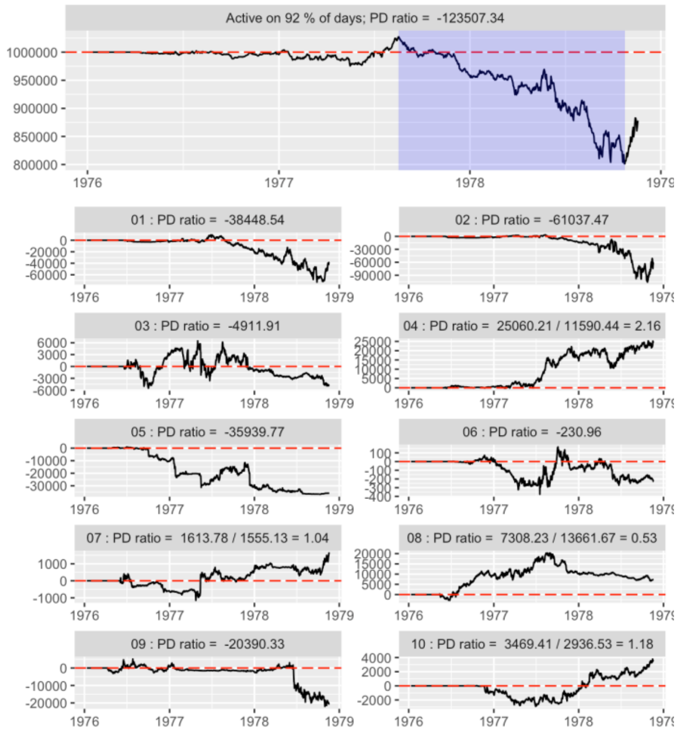

Figure 36 – Long-term Momentum Strategy Performance on Part 3

Secondly, there were subjective technical mistakes for market orders position combination related to long-term momentum strategy, and mistakes of choosing market orders and limit orders for stop-loss. Details will be described in section 3.5.2 and section 3.5.3, reflected upon, and addressed for optimization.

- **Multi-factor Strategy**

The Alpha-006 indicator has performed worse than we expected. First, the trading frequency in the third part of the data is 91%, which is slightly lower compared to the 93% in the first and second parts. But overall, it is a high trading frequency, proving that the strategy is indeed trading almost every day as we designed it. Secondly, the P/D ratio for the three parts' data (see figure below) shows that the strategy is able to control the losses and gains to a relatively balanced level even in different quotes, as we can confidently say that no overfitting occurs.

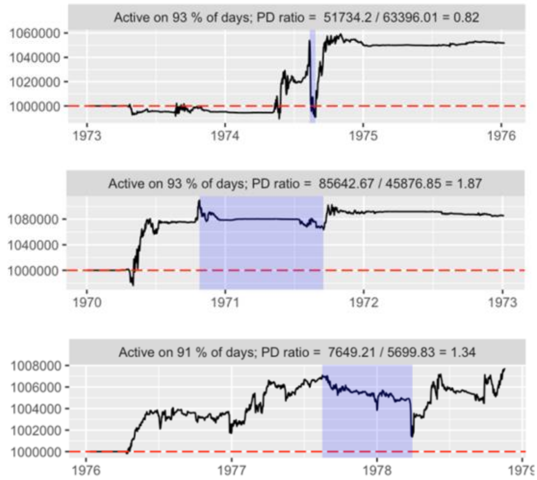

Figure 37 – Multi-factor Strategy Overall Performance on Part 1 (Up), Part 2 (Mid) and Part 3 (Bottom) Data

- **Resistance Support and Relative Strength Strategy**

The RSRS strategy performed well in part 1 and part 2, achieving a P/D ratio of 1.32 and 0.73. However, in the combined strategy, the performance of this strategy reversed dramatically and was one of the main reasons for the loss of the total strategy. The analysis revealed that as a trend-following strategy, the return chart was the exact opposite of the stock trend chart. In particular, stock 1 and 2 in part3 contribute the largest loss. From the comparison of the stock price trend chart and the return chart we can observe that the RSRS strategy was constantly selling when the price was rising sharply, thus losing significant amounts. However, for stocks that were moving down overall (e.g., stocks 4 and 6), RSRS made good gains due to constant selling.

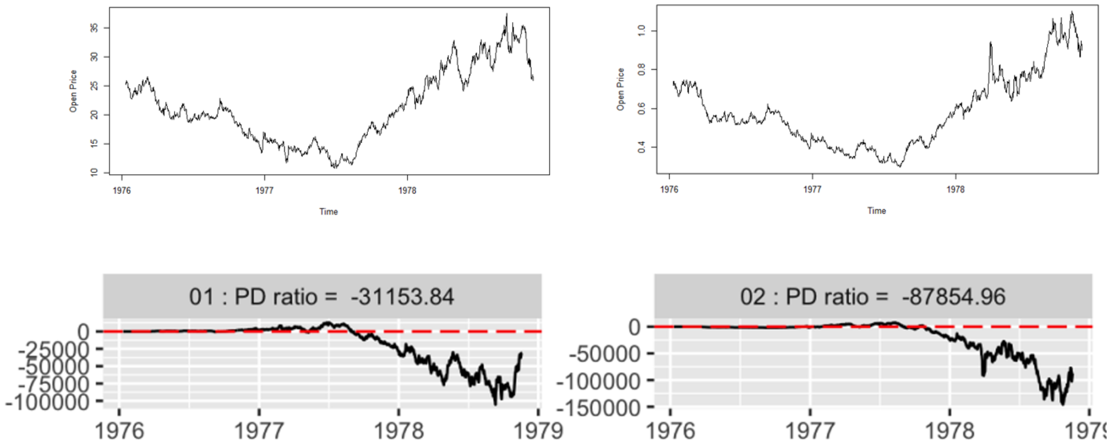

Figure 39 – Price Trend and Return Trend Comparison of Part 3 Series 1 (Left) and Series 2 (Right)

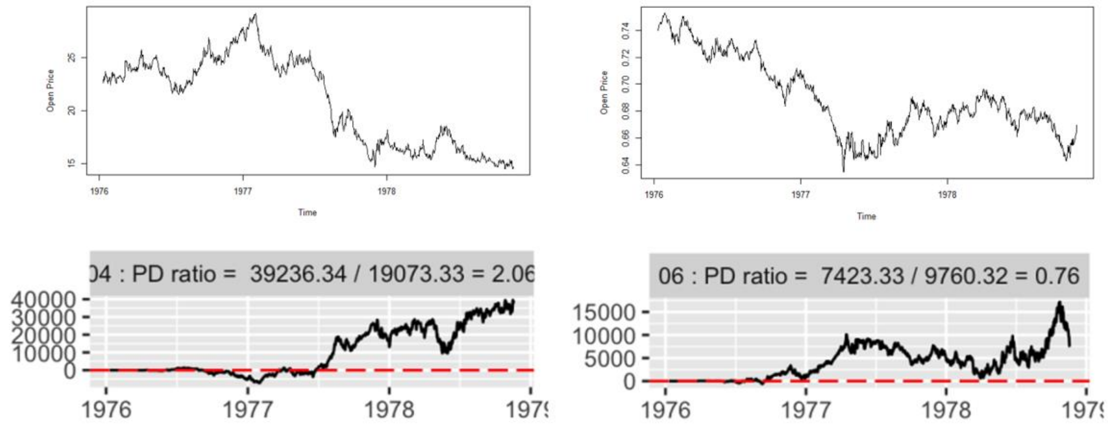

Figure 40 – Price Trend and Return Trend Comparison of Part 3 Series 4 (Left) and Series 6 (Right)

Ideally, a trend-following strategy should align with the direction of share price movements, rather than displaying an exact opposite pattern. This completely erroneous signal indicates a potential flaw in the design or execution of the combined strategy. Furthermore, there are specific issues with the RSRS indicator concerning the utilization of statistical optimization techniques, which will be extensively addressed in sections 3.5.5 and 3.5.6.

In addition to subjective factors, the performance of any trading strategy can be influenced by changes in market conditions. Several factors can potentially affect the performance of the RSRS strategy, including alterations in market volatility, shifts in the direction or strength of trends, and changes in the relationship between high and low stock prices.

For instance, in a highly volatile market, the RSRS strategy may generate a higher number of false signals due to rapid fluctuations in high and low prices. Conversely, in a low-volatility market, the strategy may generate fewer signals, resulting in lower returns. Likewise, if the direction or strength of trends changes, the RSRS strategy may produce unreliable signals, leading to losses. Additionally, modifications in the relationship between high and low prices of stock can impact the performance of the RSRS strategy, particularly if the correlation between these variables weakens or strengthens beyond the usual range.

3.5 Mistakes

3.5.1 Limited Performance of General Stop L

As mentioned in section 2.5, we simulate the previous day's rate of return and cut today's position to half if we get a negative return outcome. However, after analyzing the loss of part 3, we found that this stop-loss method can only reduce the loss to a relatively small extent, and cannot prevent the loss from continuing to occur. For example, if excessive positions and huge losses are simulated for several consecutive days, bankruptcy is likely to occur soon. Our general stop-loss method can detect this kind of crisis, but the measure of halving the position can only delay the occurrence of bankruptcy, but cannot prevent it from coming. This is the shortcoming of our general stop-loss method.

After reflection, we think liquidation might be a better option. In subsequent experiments, we gradually verified the reliability of the *pnl* parameter to simulate the profit and loss of the previous day, which further supported our decision to change the general stop loss to liquidation.

```
#Get indicators
HIGH = store$high[,i]
LOW = store$low[,i]
OPEN <- store$ope[,i]
CLOSE <- store$cl[,i]
VOLUME <- store$vol[,i]

#Get profit & loss for yesterday
prev_close <- CLOSE[store$iter-2]
cur_open <- OPEN[store$iter-1]
next_open <- OPEN[store$iter]

slippage <- abs(prev_close-next_open)*0.2
#run from day 2, where oldPos would always be 0, until penultimate day

pnl_yesterday3 <- store$dmaPos[params$series[i]] * (next_open - cur_open) - abs(store$dmaPos[params$series[i]]) * slippage

......

#General stop loss for dmaMACD
#Cut the position to half when yesterdays' simulation on return is negative
if (pnl_yesterday3<0){
dmaPos <- allzero
}
```

Figure 41 - An example of ***modified*** *general risk management (liquidation)*
For Long-term Momentum Strategy

In follow-up tests, we found that the effectiveness of such liquidation stop loss is very significant, and can be specifically displayed in the profit curve. Compared with the previous method of cutting positions in half and stopping losses, the liquidation method has been proven by us to be a better choice.

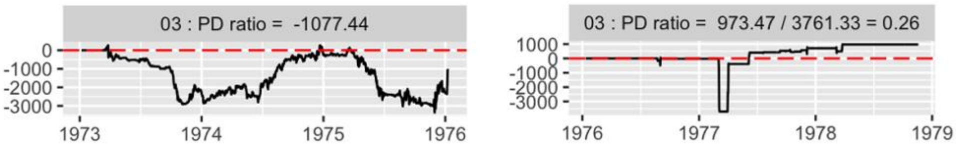

Figure 42 – Profit curve with position-halved general risk management (left)
Profit curve with liquidation general risk management (right)
On Part 3 Series 3

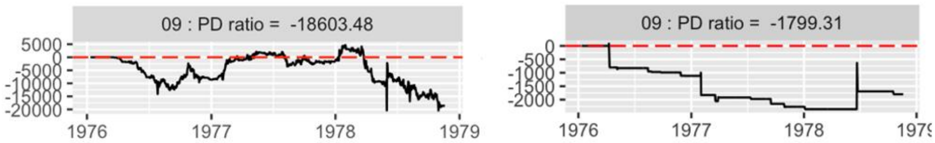

Figure 43 – Profit curve with position-halved general risk management (left)
Profit curve with liquidation general risk management (right)
On Part 3 Series 9

3.5.2 Unaccomplishment of Position Stacking

For Long-term Momentum Strategy, despite the objective reasons mentioned in section 3.1, the main causes of losses are the technical errors when we complete the combination.

The purpose of the initial *allzero* position setting for the Long-term Momentum Strategy is to allow for the stacking of positions when the strategy is executed. However, in the original code, we did not write the code for the "stacking" action, i.e., our code only determines whether to trade and stop loss for the day, but does not add up the positions that were traded to facilitate the overall stop loss liquidation.

```
#Initialize market orders and limit orders
marketOrders <- allzero

#dmaPos records the position of dmaMACD strategy
updateDmaPos <- function(store, dmaPos) {
  store$dmaPos <- dmaPos
  return(store)
  }
```

Figure 44 – Original Position Storage of Long-term Momentum Strategy

Therefore, we have now rewritten the code to add up the positions. We continue to initially set *dmaPos* to *allzero* and overlay the previous day's stored positions in the *updateDmaPos()*.

```r
#dmaPos records the position of dmaMACD strategy
updateDmaPos <- function(store, dmaPos) {
  store$dmaPos <- dmaPos+store$dmaPos
  return(store)
  }
```

Figure 45 – Modified Position Storage of Long-term Momentum Strategy

3.5.3 Limited Performance of Limit Orders Stop Loss

There are significant flaws in the specifics and rationale of the stop-loss approach. The strategy decides to execute trades that fulfill the stochastic oscillator indicator stop loss using limit orders in the code submitted by assessment 2. However, we found that the limit stop-loss order is ineffective when the price drops below the limit rapidly, and the loss continues to escalate (Chan & Ka, 2014). Therefore, if the stochastic oscillator indicator is used to determine the fluctuation of the stock as well as the high volatility, the effect of the limit stop loss order to assist stop losses will be minimal and will not accomplish the effect of stop losses.

```r
if(Kline<50 && closeP < short_ma){
  limitOrders1[i] <- -dmaPos[params$series[i]]
  limitPrices1[i] <- short_ma * (1 - 0.05)
}
```

Figure 46 – Original Limit Orders Stop Loss of Long-term Momentum Strategy

```r
if(Kline>50 && closeP > short_ma){
  dmaPos <- -store$dmaPos
}
```

Figure 47 – Modified Limit Orders Stop Loss of Long-term Momentum Strategy

Based on our reflection, we realized that if the market order stop-loss was exhausted, we would be able to shut as much as possible, regardless of whether it was sold at a cheap price or purchased at a high price. It is also worth mentioning that when the stop loss requirements are satisfied with a market order, it is much more critical to liquidate the cumulative position to avoid further losses. After improving this mistake, the performance of loss was increased to -9164 from -123507.34 in part 3.

3.5.4 Wrong Implementation of Position Clearing

Opposite to the long-term momentum strategy that needs position stacking, the short-term timing strategy (the combination of RSRS and Alpha-006) needs to clear its position every day. Our original implementation is to offset yesterday's added position by initializing the position with a negative *currentPos*.

```
#Initial position for rsrs & alpha006
rsrsPos <- -currentPos
a006Pos <- -currentPos
```

Figure 48 – Original Initiation of Short-term Timing Strategy

However, this is a wrong implementation as the *currentPos* represents yesterday's total position. The position of *RSRS* and *a006* are parts of yesterday's total position, see below the position merging codes:

```
#Sum to get the final market order positions
marketOrders <- marketOrders + a006Pos + dmaPos + rsrsPos
```

Figure 49 – Position Merging

The framework records the total position *marketOrders* and passes it to the next day as the *currentPos*. Since we want to initialize the small position instead of the total position, adding a negative total position is not effectively clearing the position.

Our modification is to add a function to store yesterday's each small position inside the variable *store* and update once the strategy ends each day. When initializing the small position the next day, we retrieve this data from the store and add a negative sign to offset it. It is worth noting that we cannot do this directly outside the loop, because the store has not been passed in at this time, which will cause an error that the variable cannot be found. Therefore, we temporarily set all small positions to 0 during initialization, and then assign them to yesterday's reverse positions in the loop.

```
#Initial position for rsrs & alpha006
rsrsPos <- allzero
a006Pos <- allzero

#Trade after lookback period
if(store$iter > params$rsrs_lookback_m + params$rsrs_lookback){

  #Iteerate through stocks
  for (i in 1:length(params$series)){

    #Clear yesterday's position
    rsrsPos <- -store$rsrsPos
    a006Pos <- -store$a006Pos
    ......
  }

  ......

}

......
```

Figure 50 – Modified Initiation of short-term timing strategy

```
#Update positions from the above strategies
store <- updateDmaPos(store, dmaPos)
store <- updateRsrsPos(store, rsrsPos)
store <- updateAlpha006Pos(store, a006Pos)

#Sum to get the final market order positions
marketOrders <- marketOrders + a006Pos + dmaPos + rsrsPos

#Initialize store
initStore <- function(newRowList,series) {
  return(list(iter=0,cl=initClStore(newRowList,series),
              vol=initVolStore(newRowList,series),
              high=initHigh(newRowList,series),
              low=initLow(newRowList,series),
              ope=initOpeStore(newRowList,series),
              dmaPos=rep(0,ncol= length(params$series)),
              rsrsPos=rep(0,ncol= length(params$series)),
              a006Pos=rep(0,ncol= length(params$series))
             ))
}
```

Figure 51 – Transfer of small positions

3.5.5 Poor Implementation of RSRS Normalization

RSRS strategy positioned as a short to medium-term strategy, therefore, we set the historical reference length of the calculation period M for the standardization parameter at 50. However, this presents potential limitations or trade-offs.

Shorter look-back periods may increase the sensitivity of RSRS values to short-term price fluctuations, which means more frequent and potentially noisy signals, resulting in increased trading activity and potentially higher transaction costs. For example, when different value of *m(30,50,55,60)* apply to part1 stock1, the return vary.

Therefore, comprehensive back-testing and sensitivity analysis is necessary to assess the impact of different back periods on strategy performance. However, due to project timeline constraints, this approach was not done on all datasets.

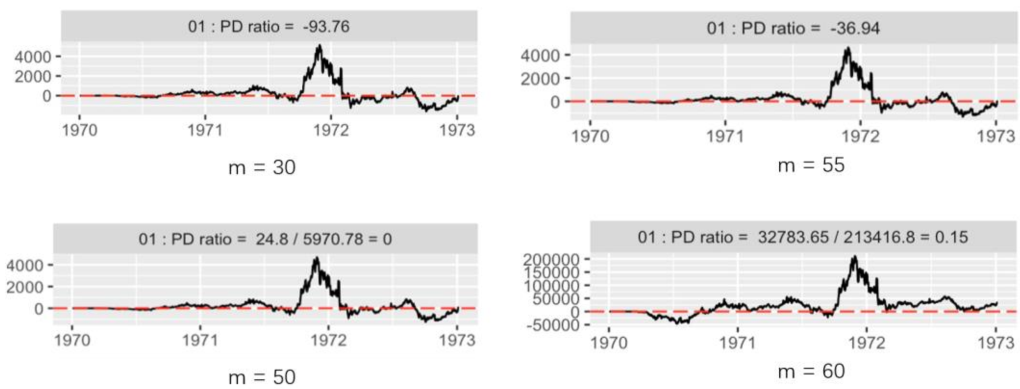

Figure 52 – Different lookback-m Values' Impact on Profit

While standardizing RSRS values using z-score calculations simplifies the comparison of relative strength among different stocks, it does not necessarily guarantee improved trading strategy performance. Firstly, standardization eliminates the original size and units of the data, potentially leading to a loss of information or distorted relative importance of different RSRS values (Pan, et al., 2016). Consequently, trading decisions based on standardized values may not capture the nuances or patterns present in the original RSRS values, potentially resulting in sub-optimal performance. Secondly, normalizing RSRS values using the z-score calculation may lead to worse performance in certain cases, as the assumption of normality may not always hold true. In the presence of a highly skewed distribution or outliers, extreme values may be classified as averages and vice versa, leading to incorrect trading decisions and poorer performance (e.g., part3 stock08) (Nayak, et al., 2014).

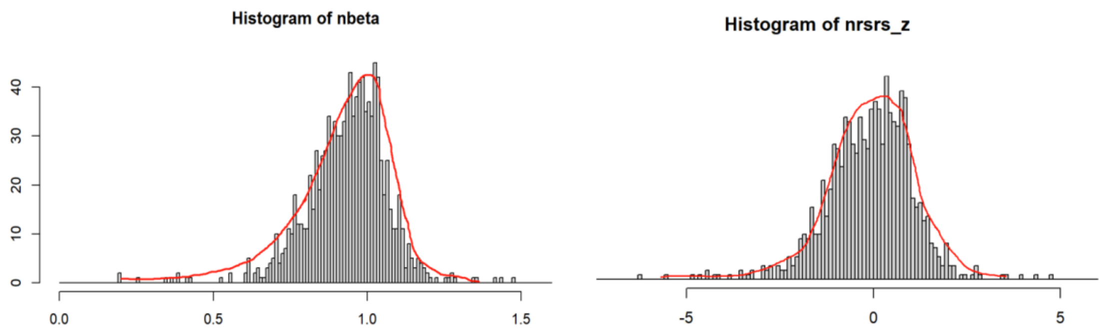

Figure 53 – Distribution of RSRS-n (left) and RSRS-z (right)

Therefore, normalization of RSRS did not result in better performance of the strategy and we decided to abandon this approach.

3.5.6 Poor Implementation of RSRS Threshold Selection

The use of a fixed threshold of 0.7 to determine trading decisions based on normalized RSRS values may not be optimal for a given data set. Dynamic threshold adjustment or using a different threshold value is likely to yield better results.

Additionally, normalizing RSRS values and applying a fixed threshold may lead to overfitting, as statistical calculations have only been performed for the Part 1 data, which may be outdated and not applicable to the current market conditions.

3.6 Modified Final Code

In summary, we have discovered the mistakes stated above, and have kept optimizing the code. We fixed the position initialization, allowing Long-term Momentum Strategy to correctly stack position, as well as allowing Short-term Timing Strategy to correctly clear position. We optimized the general risk management method, changing the implementation from position halving to liquidation. We changed the stop loss method of the Long-term Momentum Strategy from limit orders to market orders, and optimize it to position liquidation. We also repealed the use of the z-score of the Short-term Timing Strategy. After the above optimization, our strategy worked fairly better with part 3 data and made a profit of $13 thousand.

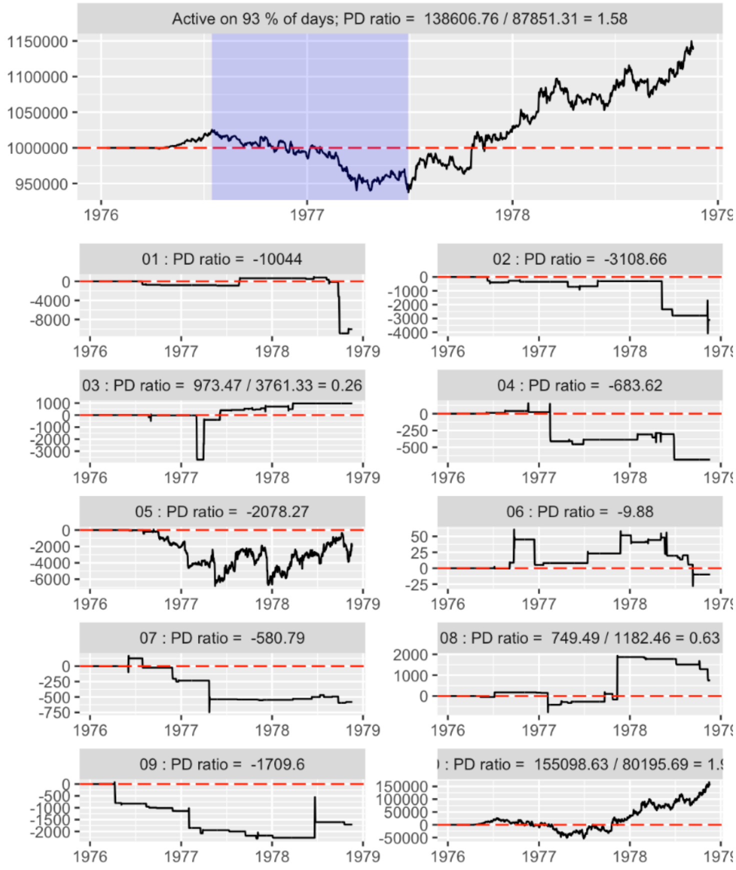

Figure 54 – Performance of Modified Code in Part 3 Data

3.7 Teamwork Reflection

Firstly, we encountered big problems when we started to integrate our strategy. That is because we didn't know anything about others' tasks, so we have to spend a lot of time understanding other people's parts.

Secondly, we are often result-oriented when we do tasks. For example, we gain an ideal result when we attempt to integrate our stock strategy. We solved this by going through and sorting out our design logic rather than relying entirely on the desired output results.

At last, what we all learnt from this project was not only the skill of collaboration but also the strengthening of our understanding of quantitative trading. We appreciate our supervisor Prof. Martin Gairing a lot for giving us so much good advice and timely response!

Section 4 - Break Down of Teamwork

Tianyi Ye

In the Automated Trading Project, I am mainly responsible for the multi-factor strategy. In Assessment 2, I wrote some codes, including The Establishment of Alpha-018 and Alpha-034 Strategy, Parameter Optimization and Position sizing management. For the optimization part, after discussing with the supervisor, I decided to write a for loop for overall optimization inside *main.R* and remove it after finding the best value. I also completed the position management part. We have three factors within our strategy, of which Alpha-018 and Alpha-034 are factors that rank stocks. After we looked up some information on the internet and then kept testing. We decided to use Aphal-018 to manage our positions in our strategy due to its better performance. In Assessment 3, I was responsible for How did this strategy compare with alternatives that you considered and What does the strategy do.

Weiyi Liu

In the Automated Trading Project, my role was the team secretary. In assessment 2, I kept track of the development of each overarching approach, identifying areas for improvement and challenges to address, to aid in further reflection. I recorded each group member's presentation and compiled recommendations to help them complete the strategy during the weekly group meetings. Simultaneously, I coordinated each meeting with the supervisor, taking notes on issues to be discussed and advice given by the supervisor, and reporting overall progress to the supervisor to help him understand it. In assessment 3, I gathered team members' thoughts and completed typesetting and editing. In addition, I completed the teamwork breakdown by myself.

Yiwang Tian

My primary responsibility in the Automated Trading Project is to implement short-term momentum and RSRS strategies. I completed part 1 of the RSRS strategy and advanced to part 2, where I optimized and combined it with the short-term momentum strategy. I was actively involved in all aspects of the merger's procedure, including designing a more logical approach and effectively managing risk. I calculated the approach's daily return and used it to develop a stop-loss component for the overall strategy. Additionally, I performed sensitivity tests and parameter optimization for the RSRS strategy to improve its performance. In part 3, I was accountable for introducing the RSRS strategy, developing and implementing the merger logic, comparing it with alternative merger strategies, and *evaluating and optimizing mistakes in our methods*.

Zhiyu Chen

In the Automated Trading Project, I was mainly responsible for the multi-factor strategy, as well as the strategy's integration and continuous optimization. I completed part 1 of the Alpha-006 strategy and advanced to part 2, where I optimized and combined it with the short-term momentum strategy. In the advanced coding part, I was the main coder for data analysis, stock trend analysis, parameter optimization, position visualization, risk management and code testing. For the final report, I was the author of *Data Separation*, *In-sample & Out-sample test*, *Risk Management*, *Parameter Optimization*, *Position Visualization* as well as *Section 3 - Evaluation and Analysis of Performance on part 3*. I was also the supervisor of coding quality and article structure, responsible for the integration, editing and typesetting of the final report.

Zixuan Zhu

In the Automated Trading Project, I was mainly responsible for the long-term momentum strategy. In assessment 1, I designed the double moving average and MACD strategy and helped the team to complete the design report. In assessment 2, I was involved in all aspects of the merger process, including completing the long-term momentum strategy and optimizing the stop loss (adding one more indicator which is called a stochastic oscillator). Besides, I also completed the position sizing part of all consolidation strategies, especially position management and risk management according to the characteristics of the strategy. In assessment 3, I did a lot of reflections on our strategy and analysis of my momentum strategy and completed most of the content of the final report. In summary, I completed a significant amount of code and report analysis throughout the completion of the project.

Reference

Berrar, D. (2019) “Cross-validation,” *Encyclopedia of Bioinformatics and Computational Biology*, pp. 542–545. Available at: https://doi.org/10.1016/b978-0-12-809633-8.20349-x (Accessed: 03 May 2023)

Chan, O. and Ka Chun Ma, A. (2014) “A framework for stop-loss analysis on trading strategies,” *The Journal of Trading*, 10(1), pp. 87–95. Available at: https://doi.org/10.3905/jot.2014.10.1.087 (Accessed: 03 May 2023)

Duran, A. and Bommarito, M.J. (2011) “A profitable trading and risk management strategy despite transaction costs,” *Quantitative Finance*, 11(6), pp. 829–848. Available at: https://doi.org/10.1080/14697680903449815 (Accessed: 03 May 2023)

Eric, D., Andjelic, G. and Redzepagic, S. (2009) “Application of MACD and RVI indicators as functions of investment strategy optimization on the financial market”, *Zb. rad. Ekon. fak. Rij*. 27(1), pp. 171-196. Available at: https://hrcak.srce.hr/file/60413 (Accessed: 03 May 2023)

Fearnley, J. (2022) “COMP 396: Honours Year - Automated Trading Project”, *Lecture Slides of the University of Liverpool*. Available at: https://student.csc.liv.ac.uk/internal/modules/comp396/lectures.html (Accessed: 03 May 2023)

Garfinkel, J. and Sokobin, J. (2006) 'Volume, opinion divergence, and returns: A study of post-earnings announcement drift', *Journal of Accounting Research*, 44(1), pp. 85–112. Available at: https://doi.org/10.1111/j.1475-679x.2006.00193.x. (Accessed: 03 May 2023)

Hansun, S. (2013) “A new approach of moving average method in time series analysis”, *2013 Conference on New Media Studies (CoNMedia)* [Preprint]. Available at: https://ieeexplore.ieee.org/document/6708545 (Accessed: 03 May 2023)

Lane, G. C. (no date) “Stochastic Oscillator”, *Technical indicator reference*. Available at: https://www.fmlabs.com/reference/default.htm?url=StochasticOscillator.htm (Accessed: 02 May 2023).

Mateia, N. A. (2013) “Simple Moving Average vs Linear Regression Forecast”, Annals – Economy Series, 16. Available at: http://www.universitateaeuropeanadragan.ro/images/imguploads/cercetare_ued/journal_annals_economy_series/journal_annals_no_16_2013/the_authors/nicolae_adrian_mateia.pdf (Accessed: 03 May 2023).

Nayak, S. C., Misra, B. B. & Behera, H.S. (2014) "Impact of Data Normalization on Stock Index Forecasting", *International Journal of Computer Information Systems and Industrial Management Applications*, 6, pp. 257-269. Available at: http://www.mirlabs.org/ijcisim/regular_papers_2014/IJCISIM_24.pdf (Accessed: 03 May 2023)

Pan, J., Zhuang, Y. and Fong, S. (2016) "The impact of data normalization on stock market prediction: Using SVM and technical indicators," *Communications in Computer and Information Science*, pp. 72–88. Available at: https://doi.org/10.1007/978-981-10-2777-2_7 (Accessed: 03 May 2023)

Pavlov, V. and Hurn, S. (2012) "Testing the profitability of moving-average rules as a portfolio selection strategy", *Pacific-Basin Finance Journal*, 20(5), pp. 825–842. Available at: https://doi.org/10.1016/j.pacfin.2012.04.003 (Accessed: 03 May 2023)

Syu, J. H. et al. (2022) "Fuzzy-based stock selection system through suitability index and position sizing," *2022 IEEE International Conference on Fuzzy Systems* (FUZZ-IEEE) [Preprint]. Available at: https://doi.org/10.1109/fuzz-ieee55066.2022.9882666 (Accessed: 03 May 2023)

Yamanaka, S. (2015) "Average true range (ATR)," *12 Simple Technical Indicators*, pp. 85–93. Available at: https://doi.org/10.1002/9781119204428.ch9 (Accessed: 03 May 2023)