Group Work Project - Econometrics - Group 6 - D

Shiqi Zhang
Mateusz Dubaniowski
Long Son Nguyen
Matthew Ang

Algorithmic Trading Strategy

Introduction

The aim of this report is to show how an algorithmic trading strategy could be designed, developed and implemented. To achieve this, we have employed pairs trading strategy and focused on two particular stocks traded on the New York Stock Exchange (NYSE). We have developed the strategy and algorithm. The results that our algorithm achieved have been presented and show that our algorithm presents good returns over the backtesting period, achieving annualized returns of around 5%. Furthermore, our report shows theoretical basis and a brief literature review on the topic specifying how our algorithm could be improved to achieve better results

The structure of this report is as follows: first, we outline the initial algorithm attempted, and the theoretical basis for pairs trading as described in literature; second, we present an improvement to the algorithm; and finally, we conduct a brief literature review of further methods to improve our described algorithm.

Algorithmic Trading

This section was contributed by Zhang Shiqi

Selected stocks: Coca-Cola and Pepsi
Strategy: pair trading

We chose Coca Cola (KO) and Pepsi (PEP) for our pair trading algorithm. As explained in the Module 7 notes, these two companies are highly correlated and they are both in the same consumer beverage industry with fairly similar product lines, geographical presence and consumer compositions. Since the notes mentioned correlation statistics until 2015, we decided to use data from 2015-01-01 till now to test our pair trading strategies in a more recent time frame.

Firstly, we pulled stock data from Yahoo Finance and kept the 'Adjusted' column only as it takes corporate events such as dividends and stock splits into considerations. To prevent any missing data points in the time series from distorting our analysis, we used back fill method to replace missing data with the next day's stock prices.

```r
library(quantmod)
library(tseries)
library(xts)
library(zoo)
library(PerformanceAnalytics)
library(knitr)
library(dplyr)

options(scipen = 999)

stock1 <- "KO"
stock2 <- "PEP"

##get data from Yahoo Finance for stock1
getSymbols(stock1, src = "yahoo")

##get data from Yahoo Finance for stock1
getSymbols(stock2, src = "yahoo")

stock1 <- KO
```

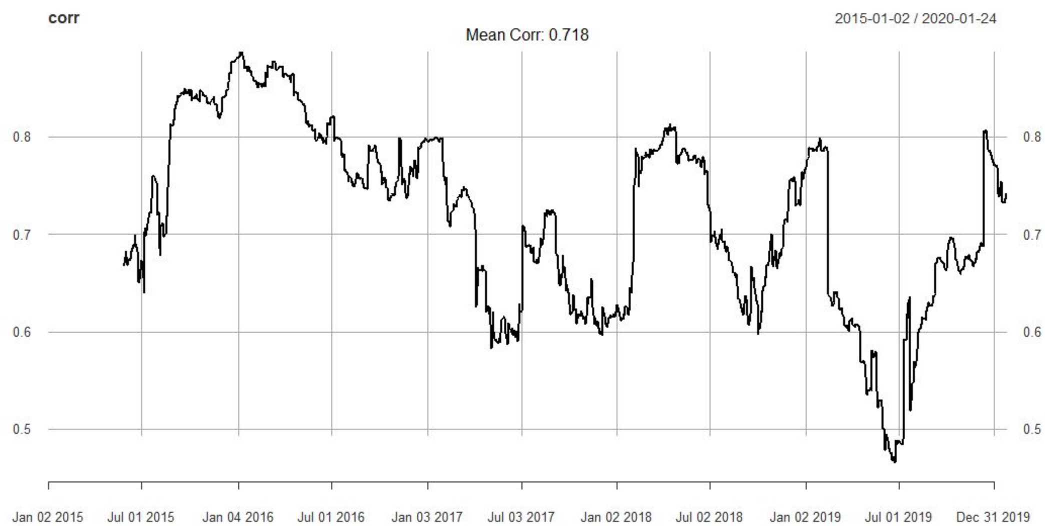
```
stock2 <- PEP

kable(head(stock1))
kable(head(stock2))

##keeping only Adjusted close data (which accounts for splits/dividends)

stock1 <- stock1[, grep("Adjusted", colnames(stock1))]
stock2 <- stock2[, grep("Adjusted", colnames(stock2))]

cut_off_date <- as.Date("2015-01-01")
stock1 <- stock1[index(stock1) >= cut_off_date]
stock2 <- stock2[index(stock2) >= cut_off_date]

stock1[is.na(stock1),]
stock2[is.na(stock2),]

##back filling missing data
stock1 <- na.locf(stock1)
stock2 <- na.locf(stock2)
```

Then we calculated the daily returns of KO and PEP and combined them into one data frame for easier calculations and back tests.

Using these returns, we plotted a graph of the rolling correlation between KO and PEP to visualize our pair trading strategy. In this case, we used 100 period look back window. From the graph below we noticed that the correlation always tends to revert back to a mean value, which is around 0.72 in our case. Hence, we are confident that a pair trading strategy would work for this pair.

```
##calculate daily returns
ret_stock1 <- Delt(stock1)
ret_stock2 <- Delt(stock2)

##combining two time series into one dataframe
data <- data.frame(matrix(NA, dim(ret_stock1)[1],2))
data[, 1] <- ret_stock1
data[, 2] <- ret_stock2
data <- xts(data, index(ret_stock1))
head(data)

##getting rolling correlation across a certain period
```

```
correlation <- function(x) {
  result <- cor(x[, 1], x[, 2])
  return(result)
}
corr <- rollapply(data, 100, correlation, by.column = FALSE)
plot(corr)
mtext(paste0('Mean Corr: ', mean(corr, na.rm=TRUE) %>%
format.default(digits=3)))
```



To devise the strategy, we need to know the hedging ratio so that we could maintain a 'dollar/market neutral' strategy.

For our trading signal generation, we used a 14-period look back window which is rather popular among industry practitioners and we chose a +/- 1 standard deviation level as our upper (shorting the spread) and lower (longing the spread) entry points. The two boundaries for entry signals are plotted below.

```
##calculate the hedge ratio
hedge_ratio <- stock1/stock2

##generate trading signals
n_period <- 14
roll_mean <- rollapply(hedge_ratio, n_period, mean)
roll_std <- rollapply(hedge_ratio, n_period, sd)

n <- 1
roll_ub <- roll_mean + roll_std*n
```

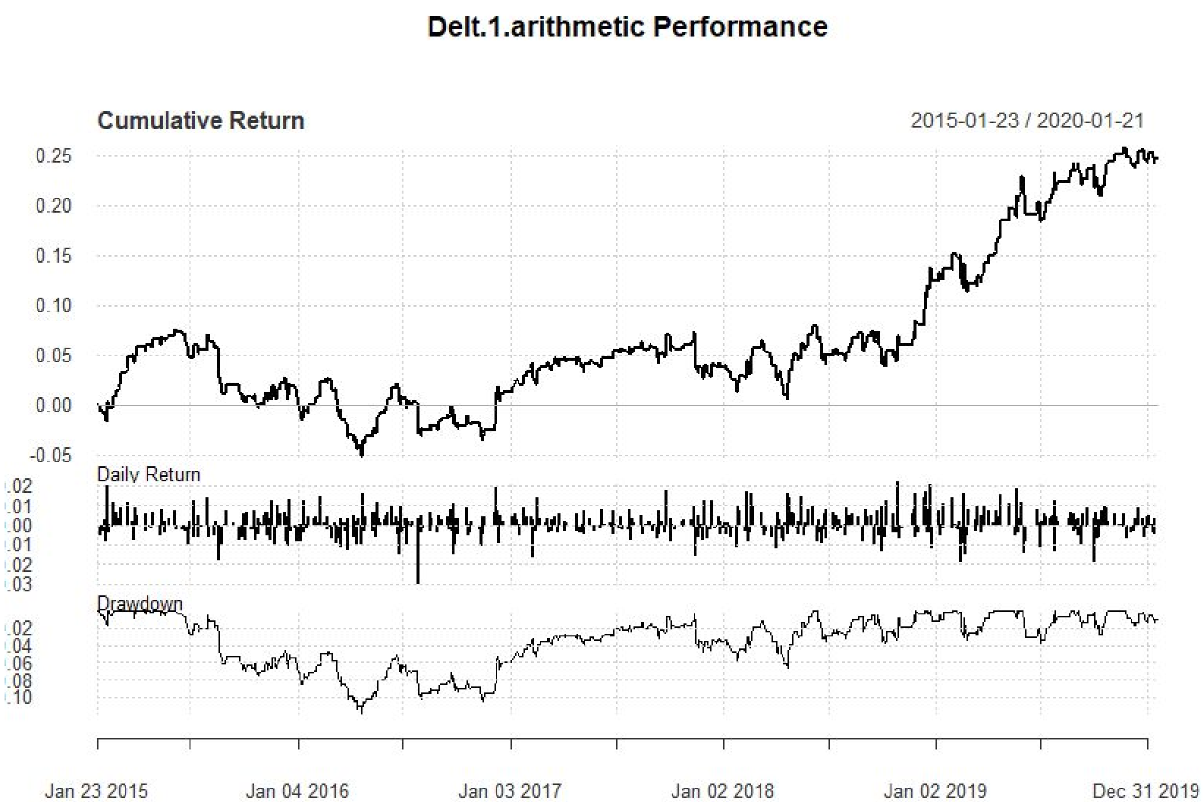
```
roll_lb <- roll_mean - roll_std*n

plot(cbind(hedge_ratio, roll_ub, roll_lb))

##define trading signals
signal <- NULL
signal <- ifelse(
  hedge_ratio > roll_ub, -1, ifelse(
    hedge_ratio < roll_lb, 1, 0
  )
)
```

Since we can only enter our trade on the next trading day after our signal has been generated, we gave our signal a one-day lag. The returns were calculated as below.

Lastly, we used Performance Summary package to analyze our trade returns and visualize them in the following graphs.

```
##we can only trade on the next trading day after a signal is generated
##hence we lag our trading signal by 1 day
signal <- lag(signal, 1)

##calculate returns generated
spread_return <- ret_stock1 - ret_stock2*hedge_ratio
trade_ratrun <- spread_return*signal
```

```
##analyze the return performances and print out key stats
charts.PerformanceSummary(trade_ratrun)

print(paste0("Cumulative Returns -- ", Return.cumulative(trade_ratrun)))

print(paste0("Annualized Returns -- ", Return.annualized(trade_ratrun)))

print(paste0("Maximum Drawdown -- ", maxDrawdown(trade_ratrun)))

print(paste0("Sharpe Ratio -- ", SharpeRatio(as.ts(trade_ratrun), Rf = 0, p
= 0.95, FUN = "StdDev")))
```




```
[1] "Cumulative Returns -- 0.246833991184567"

[1] "Annualized Returns -- 0.0452193923146706"

[1] "Maximum Drawdown -- 0.118856286999721"

[1] "Sharpe Ratio -- 0.0418060290327655"
```

To better understand pair trading and refine our approach, we researched on Ganapathy Vidyamurthy's book Pairs Trading: Quantitative Methods and Analysis other than our Module 7 notes and the materials provided in the notes (Michael Heydt's book, Mastering Pandas for Finance (2015)).

Further research papers and books on the topic

This section was contributed by Mateusz Iwo Dubaniowski.

Pole, Andrew. *Statistical arbitrage: algorithmic trading insights and techniques.* Vol. 411. John Wiley & Sons, 2011. - In this book various algorithmic approaches using statistical arbitrage are presented. This includes pair trading as a type of statistical arbitrage. It discusses various methods of searching for and identifying correlation of stocks. The book covers pair trading in large detail. However, provides also insights into other considerations to algorithmic arbitrage, such as modeling catastrophic events and considering relationships between stocks volatility.

Elliott, Robert J., John Van Der Hoek*, and William P. Malcolm. "Pairs trading." *Quantitative Finance* 5.3 (2005): 271-276. - In this paper Elliot et al. describe the concept and their example of pairs trading and what sort of approach they use to pair trading. The use Markov chain mean-reversion Gaussian model to describe the behavior of spread between two stocks.

Bolgün, Kaan Evren, Engin Kurun, and Serhat Güven. "Dynamic pairs trading strategy for the companies listed in the Istanbul stock exchange." *International Review of Applied Financial Issues and Economics* 2.1 (2010): 37. - In this paper the authors show how pairs trading was applied to stocks listed on the Istanbul Stock Exchange. They present approach that they used and show some empirical results regarding this.

Do, Binh, and Robert Faff. "Does simple pairs trading still work?." *Financial Analysts Journal* 66.4 (2010): 83-95. - In this paper the authors consider the issues with pair trading and how pair trading might be still possible in a market that is becoming more and more perfect. They argue that pairs trading is especially beneficial during periods of turbulent and volatile market thus suggesting that the technique be employed especially during such times, while growth methods might be more beneficial otherwise.

Chan, Ernie. *Algorithmic trading: winning strategies and their rationale*. Vol. 625. John Wiley & Sons, 2013. - In this book pairs trading and its execution is described in detail. Particularly, they list many different techniques for modeling spread between stocks in a pairs trading approach. However, they also list closely the basic model and various considerations that need to be accounted for while performing pairs trading such as speed of transactions and choice of trading venues.

Improving the Strategy

Linear regression for an improved hedge ratio

This section was contributed by Long Son Nguyen

In this section we try to improve the algorithm above by defining a better value of hedge ratio. In the above algorithm we defined hedge ratio as adjusted value of KO / adjusted value of PEP.

A better hedge ratio can be obtained by regressing KO on PEP and defining the intercept, using historical data in 2014

```
## Get Adjusted returns (previously obtained via Yahoo Finance)
cut_off_date <- as.Date("2015-01-01")
start_date <- as.Date("2014-01-01")

stock1_in <- stock1[index(stock1) < cut_off_date & index(stock1) >=
start_date]
stock2_in <- stock2[index(stock2) < cut_off_date & index(stock1) >=
start_date]
```

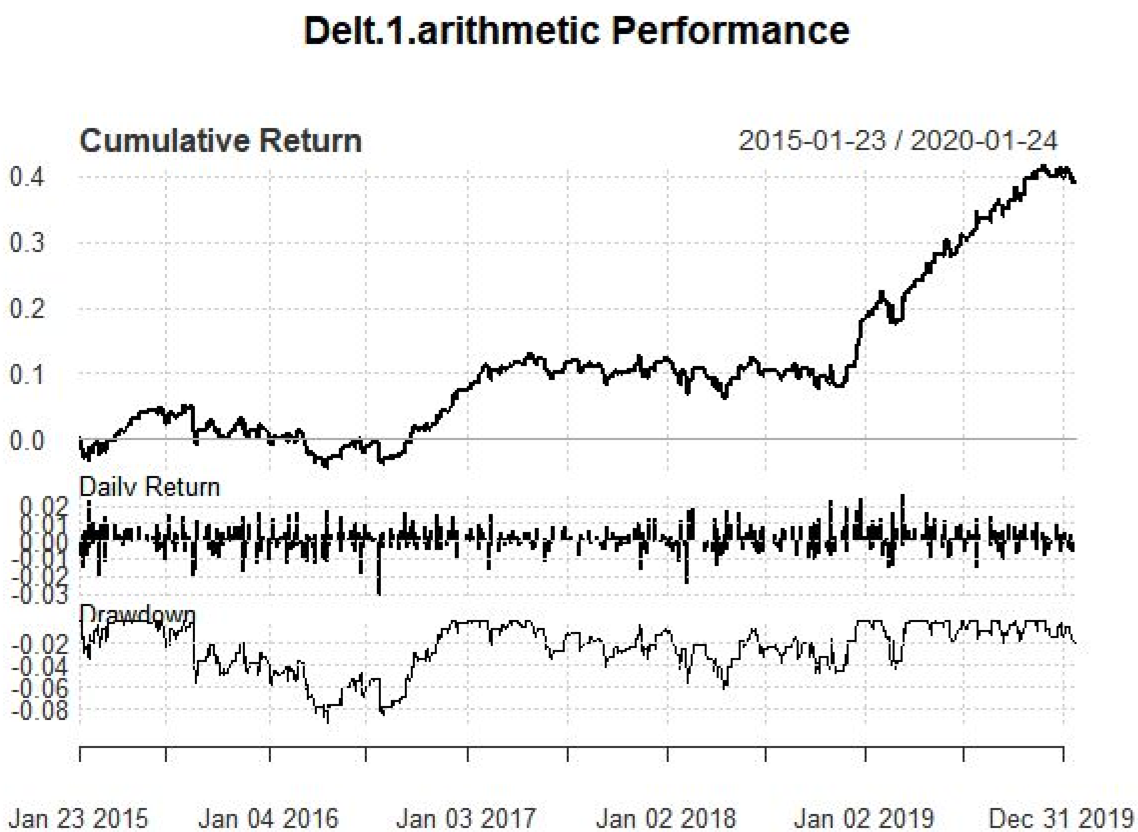
Regressing KO on PEP using historical data before 2015 and defining new hedge ratio

```
## Linear regression to calculate the intercept

lm <- lm(stock1_in ~ stock2_in)
intercept <- lm$coefficients[1]

##calculate the new hedge ratio
hedge_ratio <- (stock1-intercept)/stock2
```

Running the algorithm again with data from 2015 and new hedge ratio

```
##generate trading signals
n_period <- 14
roll_mean <- rollapply(hedge_ratio, n_period, mean)
roll_std <- rollapply(hedge_ratio, n_period, sd)
```

```r
roll_ub <- roll_mean + roll_std
roll_lb <- roll_mean - roll_std

##define trading signals
signal <- NULL
signal <- ifelse(
  hedge_ratio > roll_ub, -1, ifelse(
    hedge_ratio < roll_lb, 1, 0
  )
)

##we can only trade on the next trading day after a signal is generated
##hence we lag our trading signal by 1 day
signal <- lag(signal, 1)

##calculate returns generated
spread_return <- ret_stock1 - ret_stock2*hedge_ratio
trade_ratrun <- spread_return*signal

##analyze the return performances and print out key stats
charts.PerformanceSummary(trade_ratrun)

print(paste0("Cumulative Returns -- ", Return.cumulative(trade_ratrun)))

print(paste0("Annualized Returns -- ", Return.annualized(trade_ratrun)))

print(paste0("Maximum Drawdown -- ", maxDrawdown(trade_ratrun)))

print(paste0("Sharpe Ratio -- ", SharpeRatio(as.ts(trade_ratrun), Rf = 0, p
= 0.95, FUN = "StdDev")))
```




```
[1] "Cumulative Returns -- 0.390852715087208"

[1] "Annualized Returns -- 0.0682089896937541"

[1] "Maximum Drawdown -- 0.091870944264238"

[1] "Sharpe Ratio -- 0.0566410008418975"
```

From the results above we can see improvements in annualized returns, cumulative returns as well as maximum drawdown.

Rolling regression for online spread evaluation

This section was contributed by Matthew Ang.

In this section, we attempt to improve the algorithm's performance by applying an online estimate of the hedge ratio.

To enhance the relevance of our trade signal, we applied a rolling linear regression with a 100-day lookback window. Through the daily regression, we obtained a more current hedge ratio, spread, and trading signal.

The steps of the algorithm are detailed below:

1. Define a lookback period (14 days was used, as before)
2. Every day, run linear regression over the lookback period
    - Obtain **hedge ratio** (regression slope) and residuals
3. Compute today's **residual *Z-score*** using model residuals. If *Z-score*...
    - exceeds a predefined threshold, we trade in that direction
    - is between 0.5 and the threshold, we hold our position
    - is less than 0.5, we divest any existing positions

```
cut_off_date <- as.Date("2014-12-11")

ret_stock1 <- KO [, grep("Adjusted", colnames(stock1))] %>% Delt()
ret_stock2 <- PEP[, grep("Adjusted", colnames(stock2))] %>% Delt()

ret_stock1 <- ret_stock1[index(ret_stock1) >= cut_off_date]
ret_stock2 <- ret_stock2[index(ret_stock2) >= cut_off_date]

window <- 14
sd_thresh <- 1

num_signals <- nrow(ret_stock1)-window
signals <- xts(matrix(rep(NA,num_signals*2), ncol=2),
               order.by=index(ret_stock1[(window+1):nrow(ret_stock1)]))
colnames(signals) <- c("hedge_ratio", "signal")
```

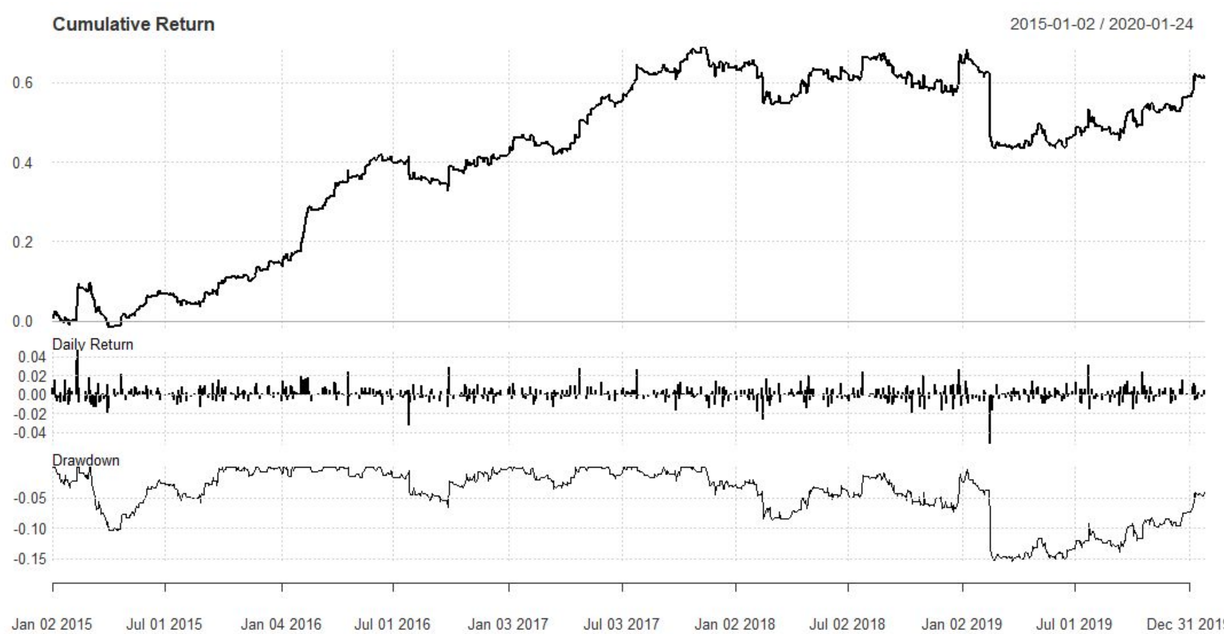
The code below runs the daily regression in a loop:

```
# use a rolling window to generate trading signal
for (i in 1:(nrow(ret_stock1)-window)) {
  y <- ret_stock1[i:(i+window-1)]
  x <- ret_stock2[i:(i+window-1)]

  model <- lm(y ~ x)
```

```
  errors <- resid(model)
  error_z <- (errors[[length(errors)]]-mean(errors))/sd(errors)

  # signal is defined as latest error deviating materially from the avg error
  hedge_ratio <- model$coefficients[[2]]
  if (abs(error_z) < .5) {
      signal <- 0
  } else if (error_z > sd_thresh) {
      signal <- -1
  } else if (error_z < sd_thresh) {
      signal <- 1
  }

  signals[i,] <- cbind(hedge_ratio, signal)
}
```

Like before, we apply the Performance Summary package to analyze our trade returns and visualize them in the following graphs:

```
trade_ratrun <- signals*(ret_stock1-ret_stock2)
charts.PerformanceSummary(trade_ratrun)

print(paste0("Cumulative Returns -- ", Return.cumulative(trade_ratrun)))
print(paste0("Annualized Returns -- ", Return.annualized(trade_ratrun)))
print(paste0("Maximum Drawdown -- ", maxDrawdown(trade_ratrun)))
print(paste0("Sharpe Ratio -- ", SharpeRatio(as.ts(trade_ratrun), Rf = 0, p =
0.95, FUN = "StdDev")))
```




```
[1] "Cumulative Returns -- 0.619040526894079"
```

```
[1] "Annualized Returns -- 0.0999973449373515"

[1] "Maximum Drawdown -- 0.152961635986656"

[1] "Sharpe Ratio -- 0.0649063380900476"
```

The trading algorithm, while not significantly different from prior implementations, provides online, updated values of the hedge ratio and spread deviations. Its Sharpe ratio has increased and aggregate returns has increased materially. However, the risk is higher too—it has the weakness of being more susceptible to short-term fluctuations—as seen from its increased max drawdown.

A more nuanced treatment of the pairs trading model and further potential enhancements can be found in Kim & Kim (2019).

Papers and books on improving the model

This section was contributed by Mateusz Iwo Dubaniowski.

One extension is to employ a hybrid model that alongside our approach would use Elliot et al.'s (2005) Gaussian Markov chain model for mean-reversion. Such an approach would help to more closely model the behavior of spread between the stocks. Thus, it would likely help in generating better trading signals. Similarly, Chan (2013), in his book, describes various additional indicators and statistical methods for additional signals that can support trading entry and exit decisions. Krauss (2017) similarly presents several additional improvements and statistical techniques that can help in generating trading signals.These are evaluated and also shown to be able to work in hybrid models.

On the other hand, there exist approaches that aid in finding very quickly short-term correlations between stocks that would enable pair trading at a much greater scale and with much better performance (Wang et al., 2009). Similarly, there approaches that utilize stochastic control theory to generate signals, these can be used to improve our model (Mudchanatongsuk et al., 2008).

Another approach that could improve the model involves using machine learning techniques such as support vector machines to support the generation of entry and exit signals for the trading model (Madhavaram, 2013) (Wu, 2015) (Huang et al., 2019). Signals generated using this method can be combined with our model to create a hybrid model. Furthermore, a genetic algorithm for signal generation can be employed to improve our model (Huang et al., 2015).

Conclusions

In this report, we have shown the development and implementation of an algorithmic trading algorithm focused around pairs trading of Coca Cola and Pepsi companies. This model has allowed us to obtain annualized returns of around 5%. Furthermore, we outlined additional approaches described in the literature of the topic that can further improve our signals to obtain even better results from our algorithmic trading strategy.

References

Bolgün, Kaan Evren, Engin Kurun, and Serhat Güven. "Dynamic pairs trading strategy for the companies listed in the Istanbul stock exchange." *International Review of Applied Financial Issues and Economics* 2.1 (2010): 37.

Chan, Ernie. *Algorithmic trading: winning strategies and their rationale*. Vol. 625. John Wiley & Sons, 2013.

Do, Binh, and Robert Faff. "Does simple pairs trading still work?." *Financial Analysts Journal* 66.4 (2010): 83-95.

Elliott, Robert J., John Van Der Hoek*, and William P. Malcolm. "Pairs trading." *Quantitative Finance* 5.3 (2005): 271-276.

Huang, Boming, et al. "Automated trading systems statistical and machine learning methods and hardware implementation: a survey." *Enterprise Information Systems* 13.1 (2019): 132-144.

Huang, Chien-Feng, et al. "An intelligent model for pairs trading using genetic algorithms." *Computational Intelligence and Neuroscience* 2015 (2015).

Kim, Taewook & Kim, Ha (2019). "Optimizing the Pairs-Trading Strategy Using Deep Reinforcement Learning with Trading and Stop-Loss Boundaries." *Complexity* 3 (2019): 1-20.

Krauss, Christopher. "Statistical arbitrage pairs trading strategies: Review and outlook." *Journal of Economic Surveys* 31.2 (2017): 513-545.

Madhavaram, Gopal Rao. "Statistical arbitrage using pairs trading with support vector machine learning." (2013).

Mudchanatongsuk, Supakorn, James A. Primbs, and Wilfred Wong. "Optimal pairs trading: A stochastic control approach." *2008 American Control Conference*. IEEE, 2008.

Pole, Andrew. *Statistical arbitrage: algorithmic trading insights and techniques.* Vol. 411. John Wiley & Sons, 2011

Wang, Jieren, Camilo Rostoker, and Alan Wagner. "A high performance pair trading application." *2009 IEEE International Symposium on Parallel & Distributed Processing*. IEEE, 2009.

Wu, Jiayu. "A Pairs Trading Strategy for GOOG/GOOGL Using Machine Learning." (2015).