Pairs Trading Project

STAT 319: Statistical Computing

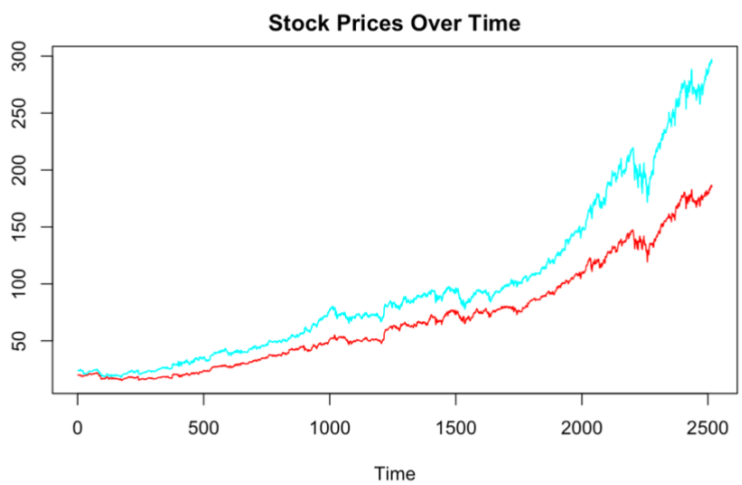
Professor Turek

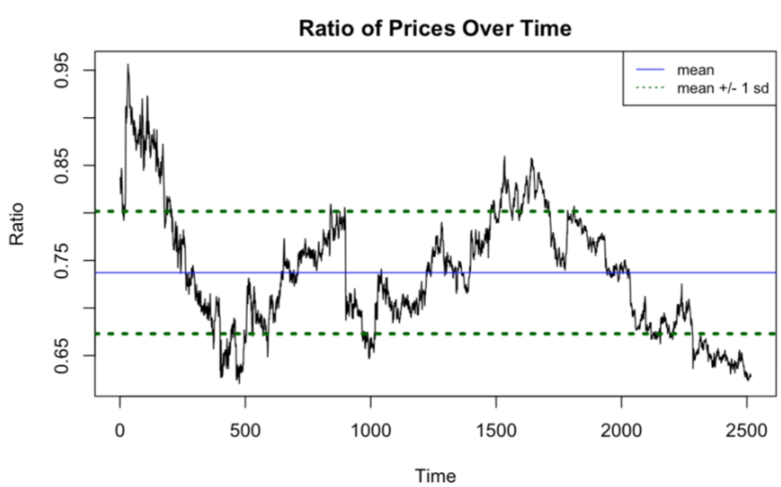
Isabel Arvelo

April 5, 2022

I. Deliverable 1

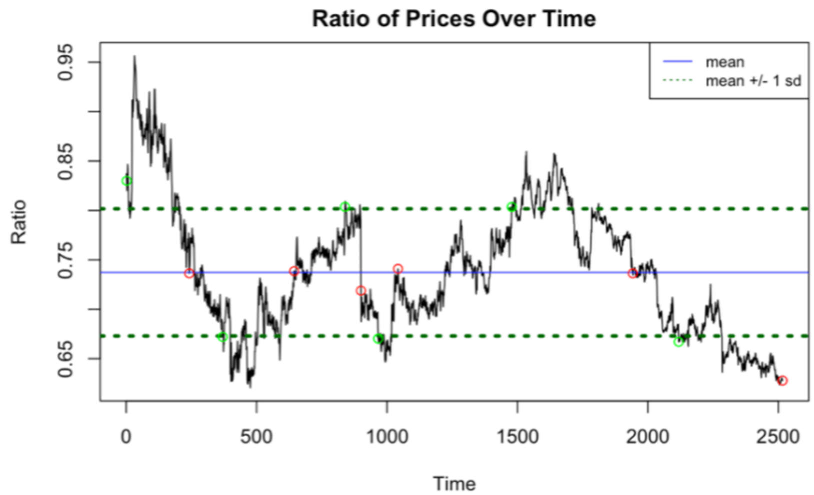
I downloaded data on the stock price of two publicly traded American multinational financial services corporations: Visa ("V") and Mastercard ("MA"). These stocks represent the financial sector of the market, made up of firms and institutions that provide financial services to commercial and retail customers. I looked at prices of both stocks between the years 2010 and 2020 and found that Mastercard has historically generally traded at higher prices, but the prices of both stocks consistently move together, with a correlation of 0.99.





II. Deliverable 2

Between the years of 2010 and 2020, using a pairs trading strategy with k = 1, we would have opened and closed six positions. The green points on the plot below indicate each day where we would open a position and the red points indicate the day that each position is closed.



III. Deliverable 3

For this scenario, we will assume all investments are always $1 and that there is constant fixed fee of 0.3% on all transactions. For Mastercard and Visa, we open the first set of positions on day 2, when the ratio of the stock prices is above 1 standard deviation from the mean ratio and close them on day 242 when the ratio dips back below its mean.

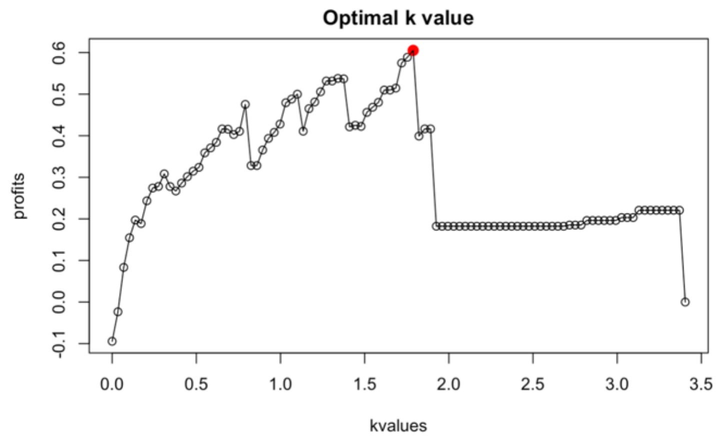
For the first positions, the price of Visa (stock1) is relatively high. Therefore, we'll sell (short) $1 of Visa stock, expecting the price to decrease, and buy $1 of Mastercard stock (stock2), expecting the price to increase. On day 2, Visa has a price of $19.98 and Mastercard has a price of $24.07. Dividing the $1 investment by each of these prices, allows us to find the shares sold of the relatively overpriced stock, Visa, and shares bought of the relatively cheap stock, Mastercard.

	Visa	*Mastercard*	*Fees*
Day 1	$1 ÷ $19.98 = 0.05 shares	$1 ÷ $24.07 = 0.042 shares	($1 + $1) * 0.03= $0.06
Day 242	0.05 * $15.51 = $0.776	0.042 ÷ $21.06= $0.875	($0.776+ $0.875) * 0.03 = $0.04953

We will sell (short) 0.05 shares of Visa stock and buy 0.042 shares of Mastercard stock. On day 242, one stock of Visa costs $15.51 and one stock of Mastercard costs $21.06. Therefore, we need to pay $0.776 to purchase Visa at the closing day price and can sell our shares of Mastercard for $0.875. Fees for this transaction amount to be $0.01. After opening/closing the first pair of positions, the net profit is 0.088% or 8.8 cents. The net profit from employing the trading strategy between 2010 and 2020 is 42.789%.

IV. Deliverable 4

The optimal value of k for pairs trading the stocks Visa and Mastercard between 2010 and 2020 is 1.78815, as indicated by the red point in the graph. The profit achieved by the optimal K (k = 1.78815) is 0.605%.

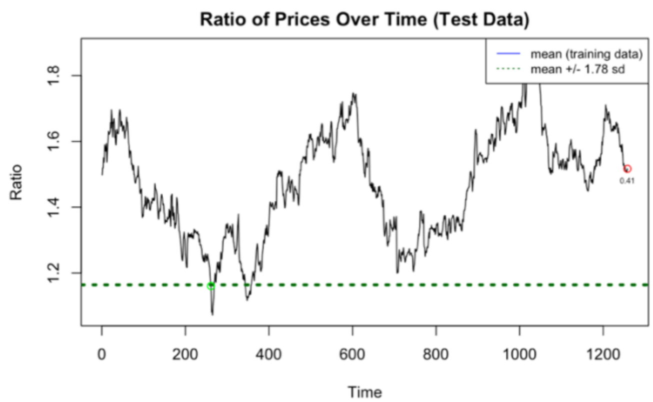


The chart below displays the price ratio, with opening/closing positions shown using this optimal value of k

V. Deliverable 5

For all the following pairs of stocks, we are training the algorithm data from the years 2010-2014 and then evaluating the strategy based on the 2015-2019 data. The stock prices for United Airlines and Delta have a high positive correlation of 0.96. Given that they are both major U.S. airlines, it makes sense that their stock prices move together.

Implementing the pairs trading strategy with these positively correlated stocks data results in a positive return of 0.4165913%.

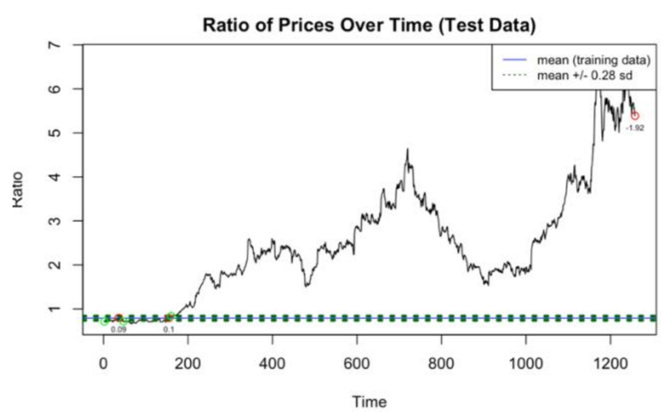


The stock prices for Apple and Barrick Gold Corporation have a negative correlation of -0.64. Apple tends to be correlated with the stock market, while the price of gold stocks tend to move inversely with the stock market so it makes sense that the stock prices of these two companies are negatively correlated.

Implementing the pairs trading strategy with these negatively correlated stocks, results in a negative return of -122.03%. This illustrates that employing this strategy on negatively correlated stocks can lead to major losses for investors.

The stock prices for Tyson and Macy's have a correlation close to 0 (0.0405). Tyson is a food company and Macy's is an American chain of department stores, so they represent two different sectors of the market that appear react differently to market conditions.

Implementing the pairs trading strategy with these uncorrelated stocks, results in a negative return of -171.77%. Out of the three pairs of stocks evaluated, investors would have incurred the biggest loss by employing the pairs trading strategy on these two uncorrelated stocks.



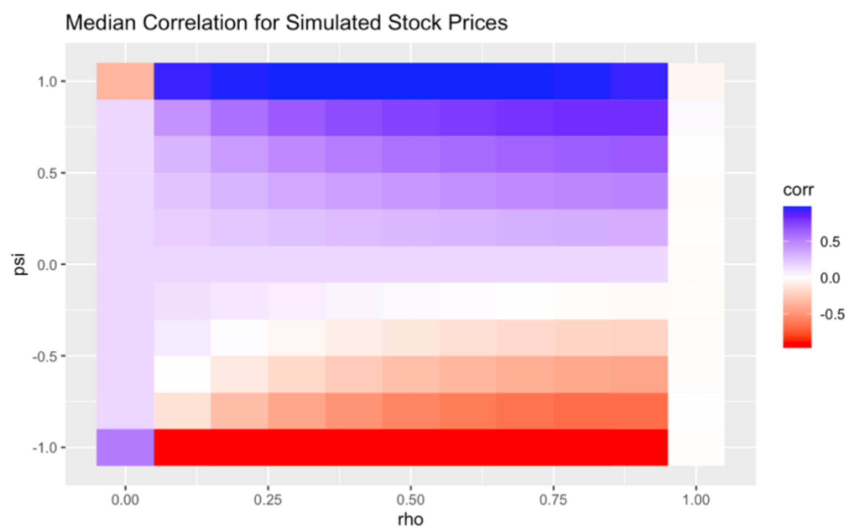
The price ratio in the test data deviates materially from the mean in the training data for all three pairs of stocks. Investors have the highest returns when stocks are positively correlated because they are to buy relatively cheap stocks and sell relatively expensive ones, with clear expectations of how the prices will change in the future. Uncorrelated stocks have the worst returns, which makes sense because there is no relationship between the stocks that can help us understand how their prices will change in relation to each other. Negatively correlated stocks also exhibit poor performance because the prices do not move in the same direction, so it is difficult to discern which stock in the pair is relatively cheap versus which one is relatively overpriced.

VI. Deliverable 6

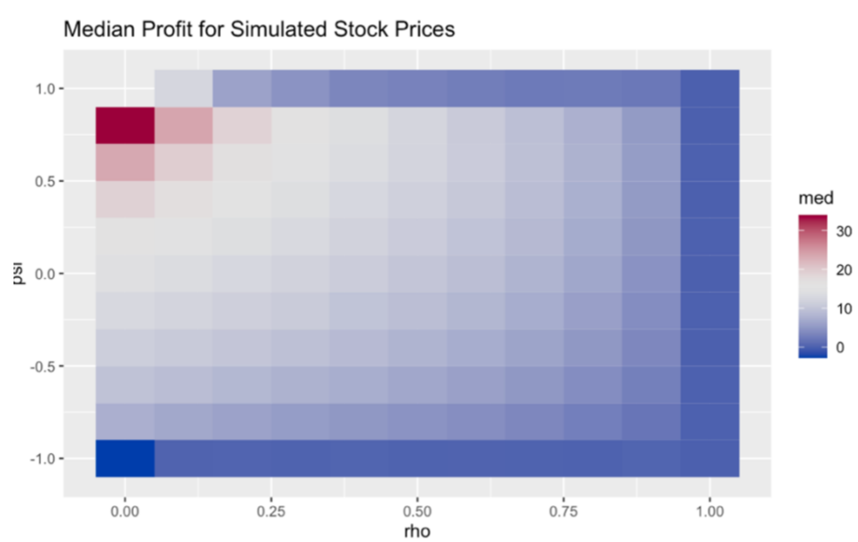
We simulated a pair of stocks with 1000 observations in each simulated stock price time series, no temporal trends with $\sigma 1=\sigma 2=1$ and $\rho=\psi=0.9$. The mean profit realized is about 455% with a standard error of 0.0382. We are 95% confident that the true mean profit realized is between 190.288% and 689.134%.

The mean correlation realized is about 0.887 with a standard error of 0.00133. We are 95% confident that the true mean correlation is between 0.7802 and 0.9495.

VII. Deliverable 7

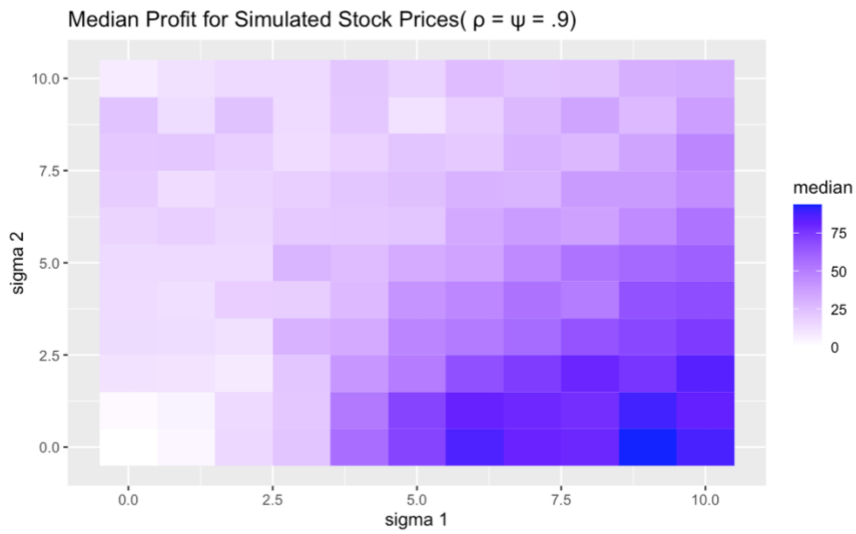


VIII. Deliverable 8



The median profit for the simulated stock pairs with rho = 0 and psi = 1 had a value of 279.3947. However, I decided to exclude the outlier value from the plot, because it made it difficult to discern the distribution of the rest of the median profits.

IX. Deliverable 9



X. Deliverable 10

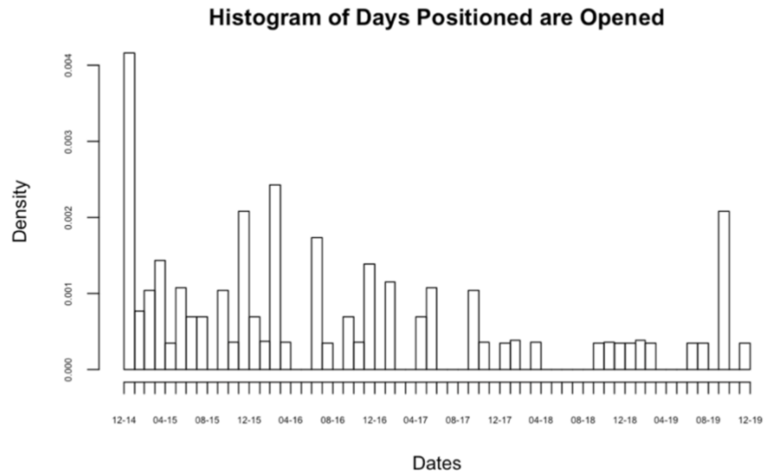
Rho, the (first order) within-stock correlation does not appear to materially affect the profits from pair trading once it takes on any value greater than 0. However, profits do tend to be slightly higher for pairs with rho values closer to 1. More importantly, as psi, the (first order) between-stock correlation, increases from -1 to 1, the profits from pairs trading distinctly tend to increase. The intercept for each stock price is trivial.

When considering the standard deviation of each stock, we are assuming psi and rho are constant. The highest profits occur when stock1 has a very high standard deviation and stock2 has a standard deviation close to 0. Holding sigma1 constant, profits tend to decrease as sigma2 increases. Holding sigma2 constant, the profits tend to increase as sigma1 increases.

VI. **Extensions**

For my first extension, I modified downloadStockPairDF() function to be robust against date ranges that don't exactly match. For example, between 2010 and 2020, the HCA Healthcare stock only has data between January 2011 and December 2019, but the Community Health Systems stock has data available from January 2010 to December 2019. Even though the stocks have different date ranges between 2010 and 2020, we are still able to download all the available data and evaluate a pairs trading strategy for the years that both stocks have data. When there is not data available for both stocks for the full range requested, we receive a message that the original date ranges are different. If one stock is missing data for those years, we receive a warning message that there is no available data for that stock in the requested date range.

For my second extension, I drew data from 30 pairs of positively correlated stocks across the 11 major sectors of the stock market: healthcare, materials, real estate, consumer staples, consumer discretionary, utilities, energy, industrials, communication services, financials, and information technology. I trained the data on the price ratio movements between 2010 and 2014 and then observed the pairs trading strategy on each pair of stocks from 2015 up to 2020 to see if there were any periods of time in which a lot of positions were opened and/or closed. I wanted to relate the work in the project to a real-world application and investigate what types of current events could have been impact the success or failure of a pairs trading strategy.



There are several periods worth investigating further between 2015 and 2020 where market conditions appear to be conducive to opening positions on a pairs trading strategy. Interestingly, the period observed was the first decade since 1850 that the U.S. did not experience a recession[1].

In the first month of 2015, a lot of positions are opened, which makes sense since positions are opened and closed based on the mean, standard deviation, and k value for 2010 through 2014 training data, but secular and/or cyclical trends could cause price ratios in the beginning of 2015 to be far from the training mean 5 years prior. Also, in the first week of 2015, the stock market swung up or down by at least 200 points in each of the first 7 days[2]. These major oscillations likely led many stock prices to dramatically increase or decrease, causing the price ratio of the test data to vary significantly from the mean of the training data.

It appears that positions were opened in every month of 2015, a year that was full of uncertainty for investors. There was major volatility driven by falling oil prices, China's economic slowdown and ceaseless speculation about when the Federal Reserve would raise

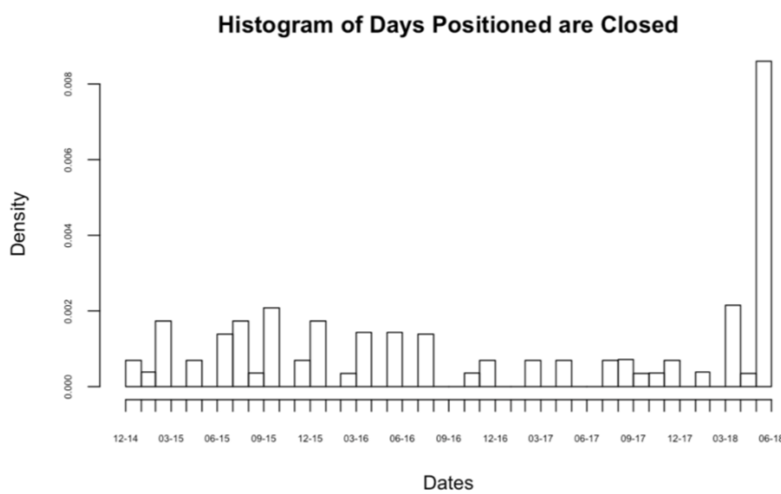
[1] Carlson, Ben. “For Investors, the Past Decade Was a Marvelous Run. but That Tells Only Half the Story.” *Fortune*, Fortune, 17 Dec. 2019, https://fortune.com/2019/12/17/investing-decade-in-review-s-p-500-tech-fed-interest-rates/.

[2] 3.7%, The Dow shed. “January Was a Terrible Month for Stocks.” *CNNMoney*, Cable News Network, 30 Jan. 2015, https://money.cnn.com/2015/01/30/investing/stocks-market-worst-january/index.html.

interest rates[3]. In a chaotic environment, with many stock prices declining, it is not surprising that the ratio of correlated stocks significantly increased or decreased - presenting the opportunity to open positions on pairs trading strategies. On December 16, 2016, the Fed made the first interest rate hike in almost a decade and we can see that there were a considerably high number of positions opened that month.

There also appears to be increased activity in the summer and early fall of 2016 coinciding with the stock market at all-time record highs in July of 2016, perhaps leading to some overvalued stocks that moved price ratios away from their previous mean[4], followed by a selloff in August. The last month in which many positions were opened was in October of 2019, a month that started with the Dow Jones Industrial plunging more than 800 points, after a report that September 2019 was worst month for U.S. manufacturing in a decade[5].

If we look at the distribution of opening positions, we can see that in 3 out of the 5 months of June’s observed, there are no positions opened. June is typically a quiet month for the stock market, on average the S&P 500 has historically ended the month with less than a tenth of a percent move[6]. There were no major stock market selloffs towards the end of the decade, so it makes sense that there weren't as many unexpected price fluctuations that investors could capitalize on with a pairs trading strategy.



---

[3] U.S. markets finished 2015 mostly in the red: The Dow was down 2.2%. The S&P 500 ended the year down 0.7%. It was the worst year for those two indexes since markets collapsed in 2008. “Dow Closes Worst Year since 2008.” *CNNMoney,* Cable News Network, 31 Dec. 2015, https://money.cnn.com/2015/12/31/investing/stocks-market-end-of-2015/.

[4] Rooney, Kate. “US Stocks Close at Record Highs, Dow Hits 7th Straight Day of Gains.” *CNBC*, CNBC, 18 July 2016, https://www.cnbc.com/2016/07/18/us-markets.html.

[5] Siegel, Rachel, and Thomas Heath. “Dow Plunges as Trump Tries to Pin 'Impeachment Nonsense' for Wall Street Rout.” *The Washington Post*, WP Company, 7 Oct. 2019, https://www.washingtonpost.com/business/2019/10/02/dow-plunges-more-than-points-wall-street-extends-october-losses/.

[6] Domm, Patti. “June Is Usually the Most Boring Month for the Stock Market, but It Likely Won't Be This Year.” *CNBC*, CNBC, 5 June 2018, https://www.cnbc.com/2018/06/04/june-is-usually-the-most-boring-month-for-the-stock-market.html.

There does not appear to be any clear patterns in the days positions are closed. Many positions end up being closed at the end of the observation period. This is likely due to fundamental changes in the stocks that make up each pair that are causing the price ratio between each stock pair to materially change between the decade long period. So, many positions are opened at the beginning of the period and not closed until the end because the relationship between the stocks has changed between the training and testing periods. In this case, the ratio in the testing data is materially different than in the training data so it never moves back towards the mean of the training data and the position is not closed until the observation periods ends.

Code Appendix

Isabel Arvelo

2022-04-04

```
downloadPriceDF <- function(stock, start = 2010, nyears = 10) {

  #String manipulation to find dates
  fd_string <- paste(as.character(start), "01", "01", sep = "-")
  ld_year <- start + nyears
  ld_string <- paste(as.character(ld_year), "01", "01", sep = "-")

  #Convert dates to date type to match it in dataset
  first_date <-as.Date(fd_string)
  last_date <- as.Date(ld_string)

  #download stock data for given ticker and date range
  stockData <- BatchGetSymbols(tickers = stock,
                            first.date = first_date,
                            last.date = last_date,
                            be.quiet = TRUE)

  #extract ticker data
  stockPrices <- stockData$df.tickers

  #cast dates as characters
  stockPrices$date <- as.character(stockPrices$ref.date)
  stockPrices$price <- stockPrices$price.adjusted
  stockDF <- stockPrices[  ,c("date", "price")]

  return(stockDF)
}
```

```
downloadStockPairDF <- function(stock1, stock2, start = 2010, nyears = 1) {

  stock1_df <- downloadPriceDF(stock1, start = start, nyears = nyears)
  stock2_df <-downloadPriceDF(stock2, start = start, nyears = nyears)

  not_aligned = 0
  n = length(stock1_df)

  if ( (length(stock1_df) > length(stock2_df)) | (length(stock1_df) < length(stock2_df)) |
       (nrow(stock1_df) != nrow(stock1_df)) ) {
    not_aligned = 1
  }

  if (nrow(stock1_df) == 0 ) {
```

```
    stop("No Data Available for Stock 1")
  }
  if (nrow(stock2_df) == 0 ) {
    stop("No Data Available for Stock 2")
  }

  for (i in 1:n) {
    if ( !identical(stock1_df$date[i], stock2_df$date[i]) ) {
      not_aligned = 1
    }
  }

  if (not_aligned) {
    print("Original date ranges are different. May be excluding available data")
    aligned_dates <- merge(stock1_df, stock2_df, by="date", all = T)
    stockPairDF <- aligned_dates[complete.cases(aligned_dates), ]
    colnames(stockPairDF) <- c("date", "stock1" , "stock2")
  } else {
    stockPairDF <- data.frame(stock1 = stock1_df$price, stock2 = stock2_df$price)
  }

  stockPairDF$ratio <- stockPairDF$stock1/stockPairDF$stock2

  return(stockPairDF)
}
```

```
plotStocks <- function(stocksDF) {
  #create time series
  stock1_ts <- ts(stocksDF$stock1, start = 1, end = nrow(stocksDF), frequency = 1)
  stock2_ts <- ts(stocksDF$stock2, start = 1, end = nrow(stocksDF), frequency = 1)
  ts.plot(stock1_ts, stock2_ts, main = "Stock Prices Over Time",gpars= list(col=rainbow(2)))
}
```

```
plotRatio <- function(stocksDF, k = 1) {
    ratio_ts <- ts(stocksDF$ratio, start = 1, end = nrow(stocksDF), frequency = 1)
    ts.plot(ratio_ts, main = "Ratio of Prices Over Time", ylab = "Ratio")

    sd = sd(stocksDF$ratio)
    mean = mean(stocksDF$ratio)

    upper_lim = mean + (k * sd)
    lower_lim = mean - (k * sd)

    abline(h = mean, col = "blue")
    abline(h = upper_lim, col="darkgreen", lwd=3, lty=3)
    abline(h = lower_lim, col="darkgreen", lwd=3, lty=3)

    legend("topright", legend=c("mean", paste("mean +/-", floor(k * 100) / 100, "sd", sep = " ")),
       col=c("blue", "darkgreen"), lty=c(1,3), cex=0.8)

}
```

```r
findPositions <- function(ratio, m, s, k = 1) {

    positions <- list()

    n <- length(ratio) #max time horizon
    uBound <-  m + (s*k)
    lBound <-  m - (s*k)

    currDay <- 1

    while ( currDay < n ) {

      possibleOpenDays <- ( (ratio > uBound) | (ratio < lBound) ) & (1:n > currDay)

      if (any(possibleOpenDays)) {
        openDay <- which(possibleOpenDays)[1]

      if (ratio[openDay] > uBound) {
        highLow <- 1
        currDay <- openDay + 1
        while (ratio[currDay] > m & currDay < n){
          currDay <- currDay +1
        }
          }
      if (ratio[openDay] < lBound) {
        highLow <- -1
        currDay <- openDay + 1
        while (ratio[currDay] < m & currDay < n) {
          currDay <- currDay + 1
        }
      }
      if (currDay > n ) {
        currDay <- n
      }
      closeDay <- currDay
      positions[[length(positions)+1]] <- c(openDay, closeDay, highLow)
      }
      else {break}
    }

    return(positions)

}
```

```r
addPositions <- function(ratio, positions, k) {

  for (i in 1:length(positions)) {
    x_start= as.integer(positions[[i]][[1]])
    x_end = as.integer(positions[[i]][[2]])
    points(x_start, ratio[x_start], col = "green", pch = 1)
    points(x_end, ratio[x_end], col = "red", pch = 1)
  }
```

```
}
```

```
positionProfit <- function(stocksDF, positions, net = TRUE) {
  if (length(positions) == 0) {
    profits <- 0
    return(profits)
  }

  nPositions <- length(positions)
  profits <- numeric(nPositions)
  fixedFee <-0.003

  for (i in 1:nPositions) {
    openPosition <- positions[[i]][[1]]
    closePosition <- positions[[i]][[2]]

    # number of shares using prices when we open the positions
    shares1 <- 1 / stocksDF$stock1[openPosition]
    shares2 <- 1 / stocksDF$stock2[openPosition]

    highLow <- positions[[i]][[3]]

    profit1 <- -1 * highLow * shares1 * stocksDF$stock1[closePosition]
    profit2 <-  highLow * shares2 * stocksDF$stock2[closePosition]


    fees <- fixedFee * (1 + 1 + abs(profit1) + abs(profit2))

    netprofit <- profit1 + profit2 - fees

    profits[i] <- netprofit
     }

  if (!net) {return(profits) }

  else {return(sum(profits)) }
}
```

```
findOptimalK <- function(stocksDF, plot = FALSE) {
  ratio <- stocksDF$ratio
  m <- mean(ratio)
  s <- sd(ratio)

  kmax <- max(sapply(ratio-m, abs)) / s
  kvalues <- seq(0, kmax, length = 100)

  positionvecs <- sapply(kvalues, function(x) findPositions(ratio = ratio, m = m, s = s, k = x))
  profits <- sapply(positionvecs, function(x) positionProfit(stocksDF = stocksDF, positions = x, net = 

  optimalK <- kvalues[which.max(profits)]

  if (plot) {
    plot(profits~kvalues, col=ifelse(kvalues==optimalK, "red", "black"),
```

```
    pch=ifelse(kvalues==optimalK, 20, 1), cex=ifelse(kvalues==optimalK, 2, 1), main =
      "Optimal k value")
    lines(kvalues, profits)
  }

  return(optimalK)

}
```

```
evalPlot <- function(stocksDF, k = 1, sd = 1, mean = 1) {
    ratio_ts <- ts(stocksDF$ratio, start = 1, end = nrow(stocksDF), frequency = 1)
    ts.plot(ratio_ts, main = "Ratio of Prices Over Time (Test Data)", ylab = "Ratio")


    upper_lim = mean + (k * sd)
    lower_lim = mean - (k * sd)

    abline(h = mean, col = "blue")
    abline(h = upper_lim, col="darkgreen", lwd=3, lty=3)
    abline(h = lower_lim, col="darkgreen", lwd=3, lty=3)

    legend("topright", legend=c("mean (training data)", paste("mean +/-", floor(k * 100)
                                                            / 100, "sd", sep = " ")),
    col=c("blue", "darkgreen"), lty=c(1,3), cex=0.8)

}
```

```
evaluatePairsTrading <- function(stocksDF, trainingFrac = 0.5, plot = FALSE) {
    n= nrow(stocksDF)
    train_cutoff = floor(trainingFrac*n)

    training =  stocksDF[1:train_cutoff, ]
    test = stocksDF[(train_cutoff+1):n, ]

    train_ratio <- training$ratio
    train_m <- mean(train_ratio)
    train_s <- sd(train_ratio)

    train_k <- findOptimalK(training, plot = FALSE)
    test_positions <- findPositions(test$ratio, m = train_m, s = train_s, k = train_k)
    profit <- positionProfit(test, test_positions)

    if (plot) {
      if (length(test_positions) > 0 ) {
        profit_vec <-positionProfit(test, test_positions, net = FALSE)
        evalPlot(test, k = train_k, sd = train_s, mean = train_m)
        addPositions(test$ratio, test_positions)
        for (i in 1:length(test_positions)) {
          text(test_positions[[i]][[2]], test$ratio[test_positions[[i]][[2]]],
               as.character(floor(profit_vec[i] * 100) / 100), cex = .5 , pos = 1)
        }
      } else {
        evalPlot(test, k = train_k, sd = train_s, mean = train_m)
```

```
      }

    }

    return(profit)
}
```

```
simulateStockPair <- function (n=1000, sigma1=1, sigma2=1, rho=1, psi=0, b1=0, b2=0, plot=FALSE) {

  x1 = 10
  x2 = 20

  x_1i <- list(x1)
  x_2i <- list(x2)

  e_1 <- rnorm(n, 0, sigma1 ^ 2)
  e_2 <- rnorm(n, 0, sigma2 ^ 2)

  a1 <- 20
  a2 <- 30

  y_1i <- list()
  y_2i <- list()


  for (i in 2:n) {
    x_1i[i] <-
      rho * x_1i[[i - 1]] + (1 - rho) * psi * x_2i[[i - 1]] + e_1[[i]]
    x_2i[i] <-
      rho * x_2i[[i - 1]] + (1 - rho) * psi * x_1i[[i - 1]] + e_2[[i]]
  }

  x_1i <- unlist(x_1i)
  x_2i <- unlist(x_2i)

  y_1i <- a1 + b1 * 1:length(x_1i) + x_1i
  y_2i <- a2 + b2 * 1:length(x_2i) + x_2i

  stocksdf <- data.frame(stock1 = y_1i, stock2 = y_2i)

  stocksdf$ratio <- (stocksdf$stock1) / (stocksdf$stock2)

  if (plot) {
    plotStocks(stocksdf)
  }

  return(stocksdf)
  }
```

```
simulateDistribution <-function(nreps = 100, returnCorrelation = FALSE, ...) {
    trials <- list()
    cor <- list()
```

```
    if (returnCorrelation) {
      for (i in 1:nreps) {
        sim <- simulateStockPair(...)
        cor[[i]] <-cor(sim$stock1, sim$stock2)
      }
      return (unlist(cor))
    }

    else {
      for (i in 1:nreps) {
        trials[[i]] <- simulateStockPair(...)
      }
      dist <- lapply(trials, evaluatePairsTrading)
    }

    return(unlist(dist))
  }
```

Extension 2

```
positions <- function(stocksDF, trainingFrac = 0.5, plot = FALSE) {
    n= nrow(stocksDF)
    train_cutoff = floor(trainingFrac*n)

    training =  stocksDF[1:train_cutoff, ]
    test = stocksDF[(train_cutoff+1):n, ]

    train_ratio <- training$ratio
    train_m <- mean(train_ratio)
    train_s <- sd(train_ratio)

    train_k <- findOptimalK(training, plot = FALSE)
    test_positions <- findPositions(test$ratio, m = train_m, s = train_s, k = train_k)


    return(test_positions)
}

listToDf <- function(nestedlist) {
  if (length(nestedlist) >  0 ) {
      df= as.data.frame(t(as.data.frame(nestedlist)))
  rownames(df)<-NULL
  df %>%
  rename(
    Open = V1 ,
    Close = V2
    )
  }
}
```

```r
dfToDays <- function(df, openDays, closeDays) {
  df_pos <- positions(df)
  df_df <- listToDf(df_pos)
  return(df_df)
}
```

```r
openDays <- c()
closeDays <- c()

#all stocks downloaded have same date range - ensured by not receiving any of the warning messages impl

#healthcare
healthcare_1 <- downloadStockPairDF("PFE", "JNJ", start = 2010, nyears = 10)
healthcare_2 <- downloadStockPairDF("UNH", "HUM", start = 2010, nyears = 10)
healthcare_3 <- downloadStockPairDF("AFL", "CI", start = 2010, nyears = 10)

healthcare_1_df <- dfToDays(healthcare_1)
healthcare_2_df <- dfToDays(healthcare_2)
healthcare_3_df <- dfToDays(healthcare_3)

#materials
materials_1 <- downloadStockPairDF("ECL", "APD", start = 2010, nyears = 10)
materials_2 <- downloadStockPairDF("APD", "DD", start = 2010, nyears = 10)

materials_1_df <- dfToDays(materials_1)
materials_2_df <- dfToDays(materials_2)

#real estate
re_1 <- downloadStockPairDF("O", "ARE", start = 2010, nyears = 10)
re_2 <- downloadStockPairDF("SPG", "DRE", start = 2010, nyears = 10)
re_3 <- downloadStockPairDF("ARE", "SUI", start = 2010, nyears = 10)

re_1_df <- dfToDays(re_1)
re_2_df <- dfToDays(re_2)
re_3_df <- dfToDays(re_3)


#consumer staples
cs_1 <- downloadStockPairDF("PEP", "COKE", start = 2010, nyears = 10)
cs_2 <- downloadStockPairDF("PG", "UL", start = 2010, nyears = 10)
cs_3 <- downloadStockPairDF("COST", "WMT", start = 2010, nyears = 10)

cs_1_df <- dfToDays(cs_1)
cs_2_df <- dfToDays(cs_2)
cs_3_df <- dfToDays(cs_3)

#consumer discretionary
cd_1 <- downloadStockPairDF("MCD", "SBUX", start = 2010, nyears = 10)
cd_2 <- downloadStockPairDF("M", "JWN", start = 2010, nyears = 10)
cd_3 <- downloadStockPairDF("H", "MAR", start = 2010, nyears = 10)
cd_4 <- downloadStockPairDF("NKE", "ADDYY", start = 2010, nyears = 10)

cd_1_df <- dfToDays(cd_1)
```

```r
cd_2_df <- dfToDays(cd_2)
cd_3_df <- dfToDays(cd_3)
cd_4_df <- dfToDays(cd_4)


#utilities
util_1 <- downloadStockPairDF("ED", "DUK", start = 2010, nyears = 10)
util_2 <- downloadStockPairDF("NRG", "CNP", start = 2010, nyears = 10)

util_1_df <- dfToDays(util_1)
util_2_df <- dfToDays(util_2)

#energy
energy_1 <- downloadStockPairDF("RYDAF", "CVX", start = 2010, nyears = 10)
energy_2 <- downloadStockPairDF("PXD", "EOG", start = 2010, nyears = 10)
energy_3 <- downloadStockPairDF("XOM","CVX", start = 2010, nyears = 10)

energy_1_df <- dfToDays(energy_1)
energy_2_df <- dfToDays(energy_2)
energy_3_df <- dfToDays(energy_2)

#industrials
indus_1 <- downloadStockPairDF("RTX", "HON", start = 2010, nyears = 10)
indus_2 <- downloadStockPairDF("LMT", "GD", start = 2010, nyears = 10)

indus_1_df <- dfToDays(indus_1)
indus_2_df <- dfToDays(indus_2)

#communication services
com_serv_1 <- downloadStockPairDF("VZ","T", start = 2010, nyears = 10)
com_serv_2 <- downloadStockPairDF("NFLX","DIS", start = 2010, nyears = 10)
com_serv_3 <- downloadStockPairDF("NXST","TGNA", start = 2010, nyears = 10)

com_serv_1_df <- dfToDays(com_serv_1)
com_serv_2_df<- dfToDays(com_serv_2)
com_serv_3_df <- dfToDays(com_serv_3)


#financials
fin_1 <- downloadStockPairDF("MA", "V", start = 2010, nyears = 10)
fin_2 <- downloadStockPairDF("JPM", "MS", start = 2010, nyears = 10)
fin_3 <- downloadStockPairDF("BAC", "C", start = 2010, nyears = 10)

fin_1_df <- dfToDays(fin_1)
fin_2_df <- dfToDays(fin_2)
fin_3_df <- dfToDays(fin_3)


#information technology
tech_1 <- downloadStockPairDF("ACN", "ORCL", start = 2010, nyears = 10)
tech_2 <- downloadStockPairDF("AAPL", "MSFT", start = 2010, nyears = 10)

tech_1_df <- dfToDays(tech_1)
```

```r
tech_2_df <- dfToDays(tech_2)

openDays <- append(openDays, c(healthcare_1_df$Open, healthcare_2_df$Open,
                               healthcare_3_df$Open,
                               materials_1_df$Open, materials_2_df$Open,
                               re_1_df$Open, re_2_df$Open, re_3_df$Open,
                               indus_1_df$Open, indus_2_df$Open,
                               energy_1_df$Open, energy_2_df$Open, energy_3_df$Open,
                               cs_1_df$Open, cs_2_df$Open, cs_3_df$Open,
                               cd_1_df$Open, cd_2_df$Open,cd_3_df$Open, cd_4_df$Open,
                               util_1_df$Open,util_2_df$Open, com_serv_1_df$Open,
                               com_serv_2_df$Open, com_serv_3_df$Open, fin_1_df$Open,
                               fin_2_df$Open,  fin_3_df$Open, tech_1_df$Open,
                               tech_2_df$Open
                               ))

closeDays <- append(closeDays, c(healthcare_1_df$Close, healthcare_2_df$Close,
                                 healthcare_3_df$Close, materials_1_df$Close,
                                 materials_2_df$Close, re_1_df$Close, re_2_df$Close,
                                 re_3_df$Close, indus_1_df$Close, indus_2_df$Close,
                                 energy_1_df$Close,
                                 energy_2_df$Close, energy_3_df$Close, cs_1_df$Close,
                                 cs_2_df$Close,
                                 cs_3_df$Close,
                                 cd_1_df$Close, cd_2_df$Close, cd_3_df$Close,
                                 cd_4_df$Close, util_1_df$Close,
                                 util_2_df$Close, com_serv_1_df$Close,
                                 com_serv_2_df$Close, com_serv_3_df$Close,
                                 fin_1_df$Close,
                                 fin_2_df$Close,fin_3_df$Close, tech_1_df$Close,
                                 tech_2_df$Close ))


fulldates <- downloadPriceDF("AFL", 2010, 10)
decadedates <- fulldates $date

testLength = length(openDays)
OpenDates <- c()
for (i in 1:testLength) {
  OpenDates <- append(OpenDates, decadedates[openDays[i] + 1258])
}

CloseDates <- c()
for (i in 1:testLength) {
  CloseDates <- append(CloseDates, decadedates[closeDays[i] + 1258])
}
```