Justin Byun
Professor Daniel Turek
Statistics 319
5 April 2022

Pairs Trading Project Report

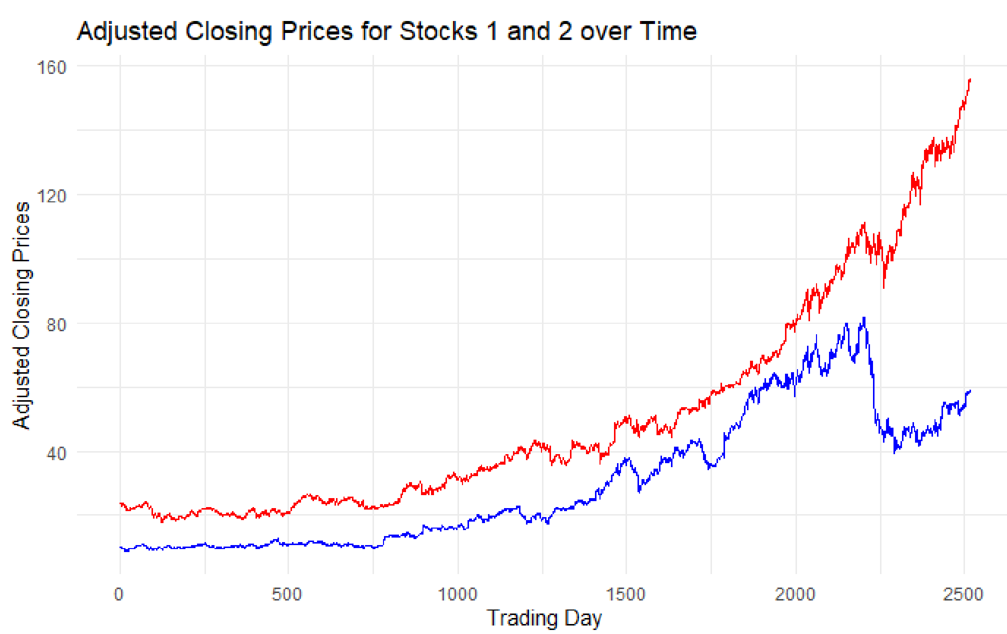
Introduction:

For this project, we will conduct an analysis of a stock trading strategy known as "pairs trading." The analysis will be based mainly on simulations that use vector autoregressive (VAR) models, which are often used to demonstrate temporal dependence.

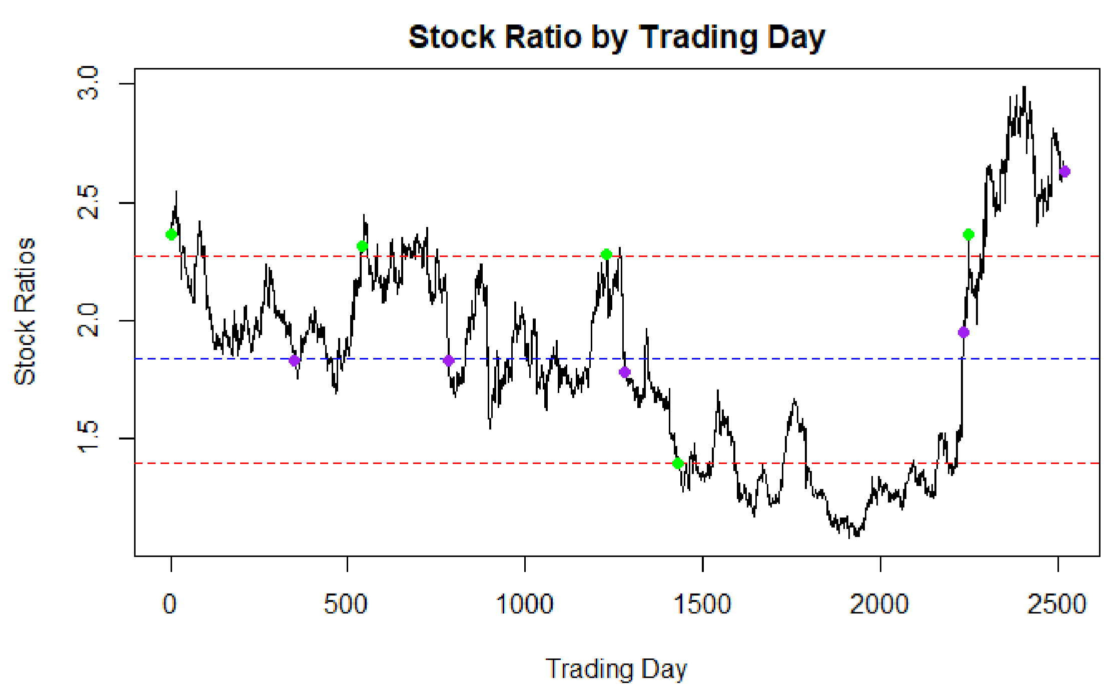
The stocks we will use in this analysis come from an R package called BatchGetSymbols, which provides the user with a function that automatically downloads historical stock prices, also known as BatchGetSymbols().

Deliverables:

#1. The two positively correlated stocks we will examine for this project are the Activision-Blizzard (ATVI) and Microsoft (MSFT) stocks. The MSFT stock represents the technology sector of the market, while the ATVI stock represents the communication services sector of the market. Below are two plots, one that demonstrates prices of both stocks over time, and another that demonstrates the ratio between the two stocks' prices over time. Both plots look at the years from 2010 to 2020. The correlation of the stock prices from 2010 to 2020 is 0.842.



#2. The opening/closing days for all positions, with k = 1, are (1, 348), (538, 781), (1228, 1278), (1426, 2231), and (2246, 2516). A plot of the ratios of stock prices with points indicating the opening and closing of all positions is below. Green points indicate the opening days, while purple points indicate the closing days.



#3. The exact calculations for the net profit after opening/closing the first pair position are below:

share1 = 1/23.85565 = 0.0419187907267

share2 = 1/10.077829 = 0.0992277205735

profit1 = 0.0419187907267*23.85565 - 0.0419187907267*19.69373 = 0.174462653501

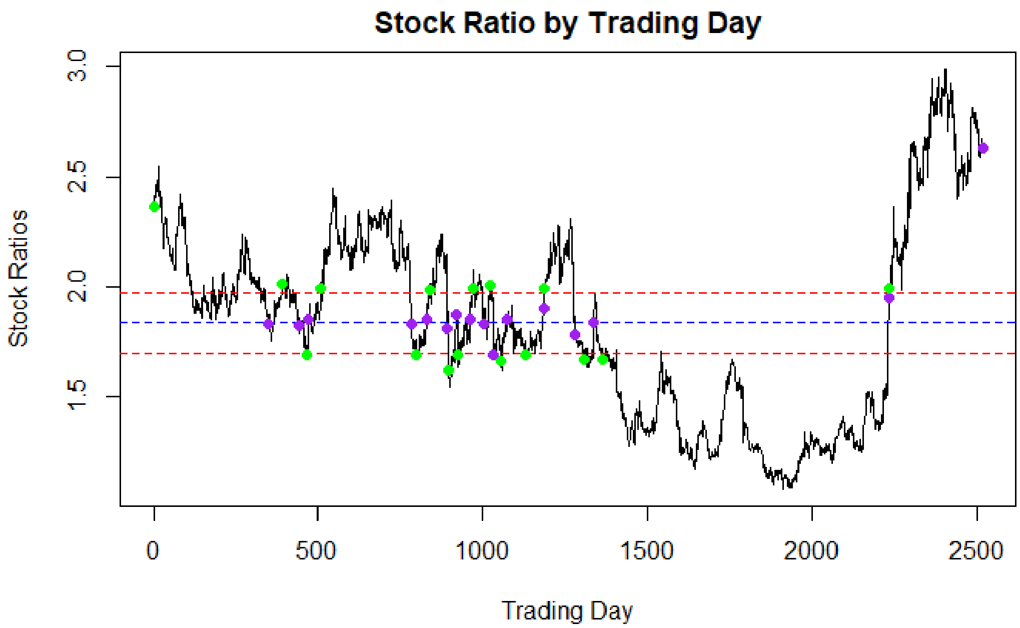
profit2 = -1*0.0992277205735*10.077829 + 0.0992277205735*10.737185 = 0.0654263929265

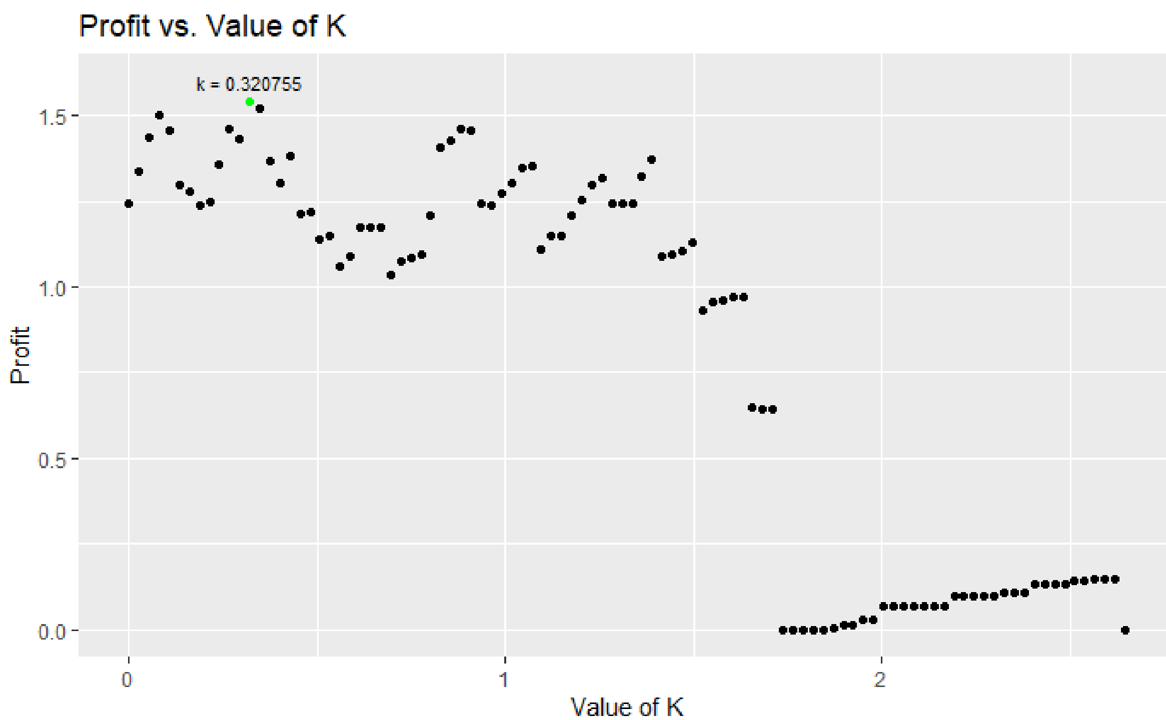
fees = 0.003 * (1+1+0.174462653501+0.0654263929265) = 0.00671966713929

positionProfit = 0.174462653501 + 0.0654263929265 - 0.00671966713929 = 0.233169379288

The net profit after opening/closing the first pair position, according to the positionProfit() function, is 0.2331694, which matches with the profit we found with our handmade calculations. The net profit calculated from the function for the pairs trading strategy in general is 1.313012, or 131.3012%.

#4. The optimal k value is 0.3207552, and the profit achieved with that k value is 1.542563184, or approximately 154.26%. The plot with the price ratios and the opening/closing positions using the optimal value of k is shown below, along with the plot of the profit versus the different values of k, from 0 to the maximum k value. The sequence from 0 to the maximum k value has been split into 100 different values of k. Furthermore, the point representing the profit corresponding to the optimal value of k has been colored green, with the value of the optimal k written above it.



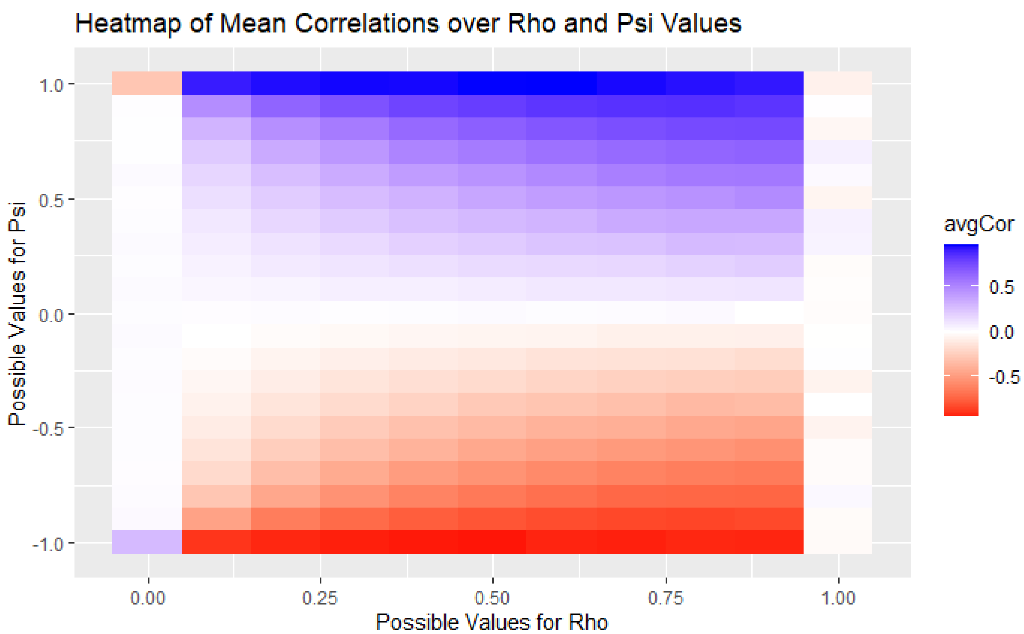


#5. The pair of stocks we'll be using that exhibits a positive correlation is Microsoft and Activision-Blizzard, with a correlation of 0.602. Hyundai and Alphabet will be our pair exhibiting a negative correlation (-0.446). Hyundai and Bitcoin USD will be the pair exhibiting the near-zero correlation (as close to zero as we could find), with a correlation of -0.248. Using the evaluatePairsTrading() function with the starting year set as 2015 and going over a period of 5 years, we have observed that the Microsoft and Activision-Blizzard pair of stocks grants us a profit of -23.456%. From the Hyundai and Alphabet pair, we observed a return of -71.194%, while the Hyundai and Bitcoin USD pair gave us a return of -179.447%. Based purely on the pairs of stocks chosen (as well as the number of years and the starting year chosen), it appears that the pairs trading strategy demonstrates the best performance on positively correlated stocks, and it performs worse on negatively correlated stocks. The strategy, however, performs the worst on stocks that have close-to-zero correlation between one another.

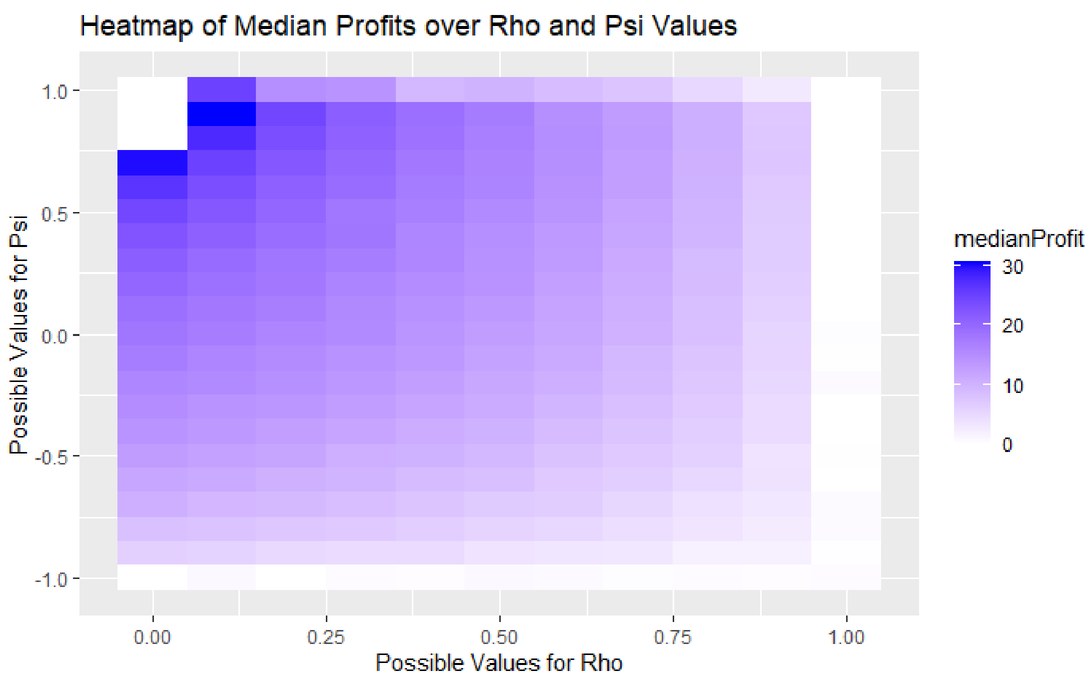
The plot above shows the ratio of the prices between the Hyundai and the Bitcoin USA stocks.

#6. According to the simulationDistribution() function, the mean profit realized between a pair of stocks with 1000 observations in each simulated stock price time series, no temporal trends, sigma 1 and sigma 2 equaling 0, and the within-stock and between-stock correlations equaling 0.9, is 5.057, or approximately 505.7%. The standard error for the mean profit is 0.388, or 38.8%. The confidence interval for the mean profit is approximately (428.7%, 582.7%). Between a pair of stocks with the same VAR process parameters, the mean correlation obtained from the function is approximately 0.849. The standard error for the mean correlation is 0.0064, and the confidence interval is approximately (0.836, 0.862).

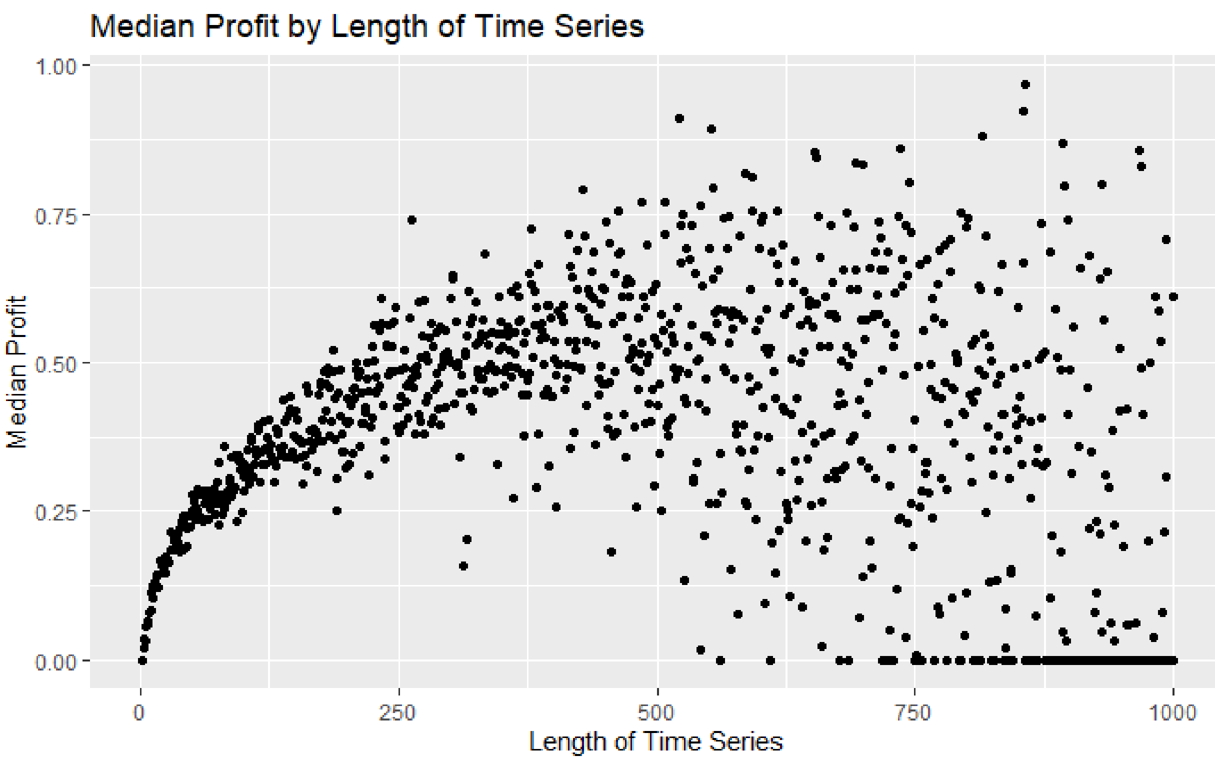
#7. Below is a heatmap representing how the mean correlation between a pair of simulated stock prices fluctuates over a range of values of rho and psi. It appears that as psi gets closer to the midway point of its reasonable range of values, specifically 0, the mean correlation will also start getting closer to 0, no matter what the value of rho is. Furthermore, if psi is near one of the extreme ends of its range (-1.0 or 1.0), then the absolute value of the mean correlation also becomes quite high, as long as the value of rho is not 0. As the value of rho increases from 0 to around 0.90, the absolute value of the mean correlation increases as well. This pattern holds true no matter the value of psi, as long as psi is not 0. When rho is either 0 or 1.00, the mean correlation becomes almost 0.



#8. Below is a heatmap representing how the median profits from a pair of simulated stock prices fluctuates over a range of rho and psi. The general trend that we observe is that as the value of rho increases and/or the value of psi decreases, the median profit decreases. When rho is 1.00 or psi is -1.0, the median profit drops to 0 or nearly 0.



#9. Another additional VAR process parameter we looked into was the length of the time series. For this investigation, the range of possible lengths we looked into was n = 2 to n = 1000, which is the default length that was set in the simulateStockPair() function. We start from n = 2, primarily because of how simulateStockPair() was implemented. A scatter plot demonstrating how the median profit varies with the length of the time series is shown below:



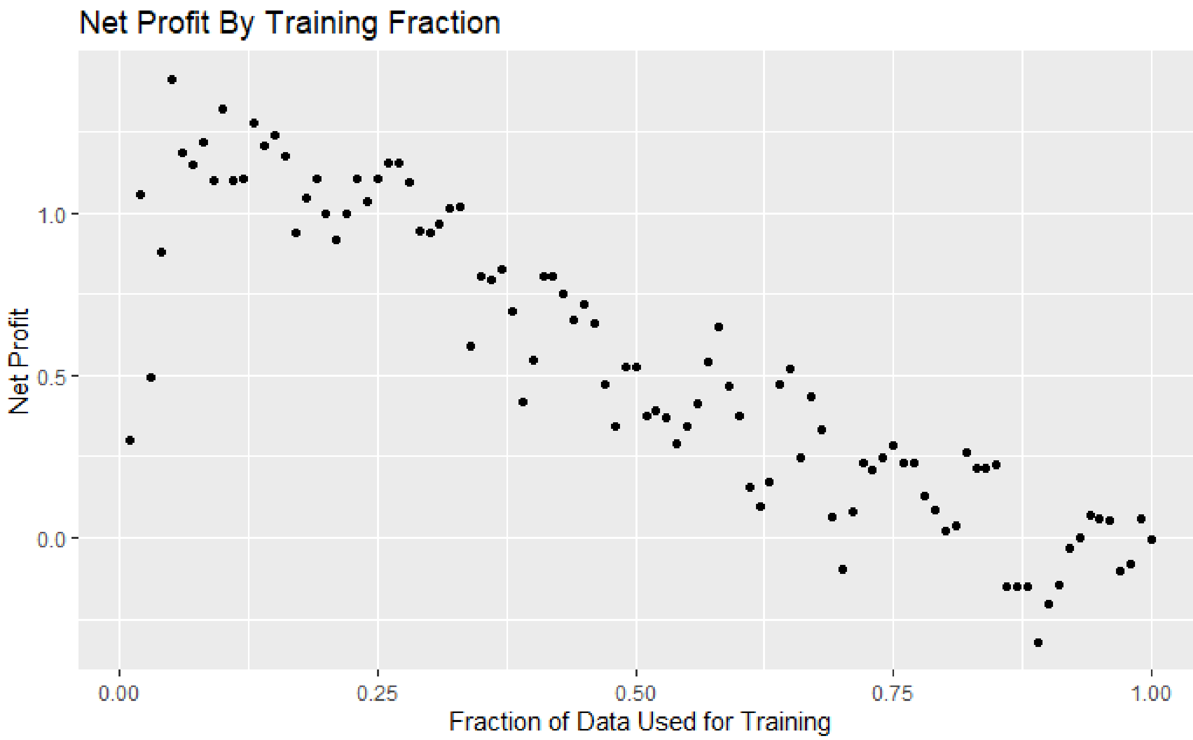
What we observe from the plot above is that there appears to be what resembles a logarithmic relationship between the median profit and the length of the time series. Generally, as the length of the time series increases, the median profit also increases. The increase in the median profit is especially sharp when the length is still low, but the increase itself levels off as the length gradually increases. The strength of the relationship does not seem very strong, since the variation in the median profit increases as the length of the time series also increases.

#10. Based on the investigations done for deliverables #7-9, it appears that rho, psi, and the length of the time series have a considerable effect on the profits made by the pairs trading strategy. For rho and psi, this means, in context, that the first order correlation between prices within the same stock (rho), as well as the first order correlation for prices between two different stocks, directly affects the profits observed from employing the pairs trading strategy.

<u>*Extensions*</u>

#1. The first extension we made to the pairs trading project is the modifications we made to the downloadStockPairDF() function, specifically to ensure that it can account for date ranges between the two selected stocks that don't exactly match. The modifications to the actual code itself are as follows: rather than simply using the stop() function and outputting a "The date ranges are not identical" message, we made use of the match() function on the stock prices data frames with one another to see which reference dates (denoted ref.date) do not have a match in the other dataframe. Since the match() function outputs NA if no matching object can be found, we used the is.na() command to find out which reference dates do not have a match (which would be outputted as TRUE). We then used the filter() function from the dplyr package in tandem with the match() function on both data frames to remove the observations containing reference dates with no match.

#2. The second extension to our pairs trading project is our investigation regarding how the fraction of data used for training our algorithm affects our net profits. Using the Microsoft (MSFT) and Activision-Blizzard (ATVI) stocks for our investigation, we decided to look into every possible fraction of data for training the algorithm, specifically in increments of 0.01. And then, with a for loop on every individual fraction of data specified above, we used the evaluatePairsTrading() function to calculate the net profits gained. Once we gathered all of the net profit values into a single vector, we put the sequence of possible fraction values and their respective net profits into a dataframe, which we then used to create the scatterplot below:



As we can see, starting from a relatively low proportion of data, the net profit begins to steadily decrease as the fraction of data used for training the algorithm increases. But since we used the MSFT and ATVI stocks for this extension, we cannot not assume this trend applies to other pairs of stocks. Until we investigate other pairs of stocks, we can only say that it is true between MSFT and ATVI stocks.