# Overview of Project

## Purpose

### The purpose of this project was to use Microsoft Excel VBA code to restructure and collect stock information from the 2017 and 2018 stocks and to determine if they were worth investing in. We practiced this during this weeks module but the goal in the challenge was to increase the efficiency of our original code in a similar format to determine if our code successfully made the VBA code run faster. The stock data included two charts for the year 2017 and the year 2018 for 12 different stocks. They contained information on a ticker value, the date the stock was issued, the opening, closing and adjusted closing price, the highest and lowest price, and the volume of the stock. However for this project, the goal was to focus on the ticker, the total daily volume, and the return on each stock.

## Analysis

### Before refactoring the code, I copied the code from VBA\_Challenge.vbs, as instructed to create the input box, chart headers, ticker array and to activate the correct worksheet in the Excel. The order of steps were given to structure the refactoring, below is the instructions with the code as written in the file.

'1a) Create a ticker Index

tickerIndex = 0

'1b) Create three output arrays

Dim tickerVolumes(12) As Long

Dim tickerStartingPrices(12) As Single

Dim tickerEndingPrices(12) As Single

''2a) Create a for loop to initialize the tickerVolumes to zero.

' If the next row’s ticker doesn’t match, increase the tickerIndex.

For i = 0 To 11

tickerVolumes(i) = 0

tickerStartingPrices(i) = 0

tickerEndingPrices(i) = 0

Next i

''2b) Loop over all the rows in the spreadsheet.

For i = 2 To RowCount

'3a) Increase volume for current ticker

tickerVolumes(tickerIndex) = tickerVolumes(tickerIndex) + Cells(i, 8).Value

'3b) Check if the current row is the first row with the selected tickerIndex.

'If Then

If Cells(i, 1).Value = tickers(tickerIndex) And Cells(i - 1, 1).Value <> tickers(tickerIndex) Then

tickerStartingPrices(tickerIndex) = Cells(i, 6).Value

End If

'3c) check if the current row is the last row with the selected ticker

'If Then

If Cells(i, 1).Value = tickers(tickerIndex) And Cells(i + 1, 1).Value <> tickers(tickerIndex) Then

tickerEndingPrices(tickerIndex) = Cells(i, 6).Value

End If

'3d Increase the tickerIndex.

If Cells(i, 1).Value = tickers(tickerIndex) And Cells(i + 1, 1).Value <> tickers(tickerIndex) Then

tickerIndex = tickerIndex + 1

End If

Next i

'4) Loop through your arrays to output the Ticker, Total Daily Volume, and Return.

For i = 0 To 11

Worksheets("All Stocks Analysis").Activate

Cells(4 + i, 1).Value = tickers(i)

Cells(4 + i, 2).Value = tickerVolumes(i)

Cells(4 + i, 3).Value = tickerEndingPrices(i) / tickerStartingPrices(i) - 1

Next i

## Results

### Using images and examples of your code, compare the stock performance between 2017 and 2018, as well as the execution times of the original script and the refactored script.

The biggest advantage I noticed after refactoring was the macro run time decreased. On the original analysis, it took over one second to run on both the 2017 and 2018 analysis's and on the updated analysis, it took far less at approximately 0.16 of a second for both the 2017 and 2018 analysis's. Attached below are screenshots of the run time for the new analysis.

![alt text] (https://github.com/lyozamp/stock-analysis./blob/main/Resources/VBA\_Challenge\_2017.png)

![alt text] (https://github.com/lyozamp/stock-analysis./blob/main/Resources/VBA\_Challenge\_2018.png)

## Summary - Pros and Cons

### In summary, the disadvantages of refactoring code would be you could potentially introduce bugs that your test won't catch because anytime you make a change there is that potential and you need to realize the risk. Also, if the data is too large and there isn't proper test cases for the existing code. However, if you do refactor your code it will be organized and very thorough for others to read and it helps the program run faster. The biggest advantage of refactoring the stock analysis code was the decrease in macro run time. A disadvantage that could of occurred with the stock analysis code would be that we added a new bug during the refactoring of the code. Overall done correctly, refactoring makes code very easy to read, improve the software and faster to program.