**## Stock -Analysis**

**Purpose:**

The purpose of this project was to refactor the Microsoft VBA code to collect various stocks from Year 2017 and 2018. This way Steve /Investor can analyze the two year of stocks and make better discussion where it is worth of invest in the certain stocks. VBA code is use to collect those stocks information in order gather data more efficiently without going excel process.

**The Data:**

The data is presented that include 12 different stocks for Year 2017 and 2018. The charts includes the following: Stock Ticker, Data , Opening price, High price, Low price and Close price, Adj, Close price and total Volume daily basis for entire year. The goal is to retrieve the data from excel sheet and create two table for Year 2017 and 2018 with Total Daily Volume and Return

**Results**

**Analysis**

Before refactoring the code, It is good way to present the code by coping from the VBA script to this document that was needed to create the input box, chart headers, ticker array, and to activate the appropriate worksheet. The steps were then listed out in order to set the structure for the refactoring. Below is the instruction and 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.

For i = 0 To 11

tickerVolumes(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 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 the next row's ticker doesn't match, increase the tickerIndex.

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

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

'3d Increase the tickerIndex.

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

Sheets("AllStocksAnalysis").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

**Summary**

**Pros and Cons of Refactoring Code**

Refactoring helps make the code cleaner and more organized way to read and understand. There are some advantages of a cleaner code include design and software improvement, debugging, and faster programming. Other users may get benefit from it when they view our projects because it becomes easier to read, as it is more concise and straightforward. There are some disadvantages of code refactor which may range from having applications that are too large to not having the proper test cases for the existing codes, which may ultimately pose some risk if we try to refactor our code.

**The Advantages of Refactoring Stock Analysis**

The biggest benefit that occurred as a result of the refactoring was an decrease in macro run time. The original analysis took approximately one second to run, whereas our new analysis only took about (approximately 0.15 seconds) to run. Attached below are the screenshots that indicate the run time for our new analysis.



