Data Analytics Challenge week 2

Overview of Project: The purpose of this analysis is to Refactor the VBA code for the stock analysis and measure the performance of the various stocks given the 2017 & 2018 returns. These returns may be positive or negative, large or small based on the total volume of the tickers, and the returns for the year.

Results: The way we performed this analysis is by refactoring the code that was given to us in the ReadMe file.

To start I Created a ticker index by declaring “tickerIndex” as a variable (integer) and setting that equal to 0

**Dim tickerIndex As Integer**

**tickerIndex = 0**

I then created the three output arrays

**Dim tickerVolumes(12) As Long**

**Dim tickerStartingPrices(12) As Single**

**Dim tickerEndingPrices(12) As Single**

Clear the tickerVolumes to 0 by creating a For loop

**For i = 0 To 11**

**tickerVolumes(i) = 0**

**tickerStartingPrices(i) = 0**

**tickerEndingPrices(i) = 0**

**Next i**

Next, I created a For loop to loop over the whole spreadsheet, sum up all the ticker volumes and differentiate the ticker index prices by ticker. I did this by adding up the ticker volumes for each ticker by using If Statements.

Last I spat out onto the spreadsheet the value of the tickers and ticker volumes, although I could not debug the spreadsheet to get the returns. The code for the refactored analysis was much simpler, thus the code ran much more quickly than that of the first analysis.

The stock performance for 2018 differs from the performance of 2017 by the fact that in 2018 most of the stocks in the list performed poorly from a returns standpoint, and in 2017 most of the stocks had positive returns.

Graphical user interface, application, table, Excel

Description automatically generated

Summary: The advantage of refactoring code is the fact that you may be able to run code that is more efficient and uses less memory if executed correctly. The disadvantage of refactoring code is that it takes a lot of effort and may not have a payoff where the code runs more efficiently than the first set of code.

The pro of refactoring this code is the fact that this code ran way more efficiently than the original code, while the con of refactoring this code was that there was more likelihood of the code producing an error, which in my case I was not able to print the returns properly in my refactored code.