VBA Challenge

Cara Davidson

May 8, 2022

Overview of Project

This week I used VBA to help someone named Steve to look at a dataset to see how well a stock performed. He wanted to see a stock's performance to see whether he should suggest buying this stock to his parents or not. His parents wanted Steve's support on a stock investment. He wanted me to refactor this code so that all of the data can loop to gather information on various stocks throughout the stock market. After refactoring this data, I can see whether the time ran is efficient or not.

Results

I created three output arrays. These arrays were:

Dim tickerVolumes(12) As Long (this held volume)

Dim tickerStartingPrices(12) As Single (this held starting price)

Dim tickerEndingPrices(12) As Single (this held ending price)

These arrays are a great coding skill to use so that stock performance data can be stored and then for loop analysis was ran on each stock. After creating these arrays, I used nested for loops to loop through the data to be able to compare stocks.

Summary

When looking at the data results for the 12 stocks ran in 2017 and 2018, there is a huge data result difference. In 2017 there was 11 out of the 12 stocks with a positive increase in volume. In 2018 there was 2 out of 12 stocks with a positive increase in volume. Looking at these results, it may be a suggestion to Steve to look at another industry of stocks.

Looking at the time of execution for the data results, each data set improved by running in .18 or .17 seconds compared to the previous .94 or 1.06 seconds. That is a significant change in the time execution. This time execution will help when running this code for a higher number of stocks to compare.

As I was refactoring my previous code, I saw both disadvantages and advantages. A disadvantage is messing up my previous code. Sometimes when you are refactoring code and you mess up an important line, it could cause an issue to run your code. This is why looking back, and taking notes, is so vital to coding. When you write an organized code with notes, it helps you revisit the code to make changes. An advantage of refactoring code is improvement. As I was refactoring this code, I went from analyzing one stock to being able to analyze multiple stocks throughout multiple years of data points. Although this was a challenge for me, I took the time to thoroughly look at each line, explain each line to myself, and ensure that each line of code would work for this analysis.