### Leslie Debassige UofT Bootcamp

# Module 2 VBA- Excel Stock Analysis Challenge

## Overview of Project

During the second and third week of the Trilogy UofT Data Analytics & Visualization Bootcamp our second challenge is to create a written analysis with two key deliverable as follows;

1) Refactor VBA code and measure performance, including an updated workbook and a folder with PNGs of the pop-ups with script run time.

2) A written analysis of your results (README.md)

The tools that will be used are Visual Basic (VBA) in Microsoft Excel for Windows. The data that we will be using is the GreenStocks6SEPT21 refactored to the VBA Challenge11SEPT21FINAL. to help Steve analysis entire dataset with the click of a macro button.

### Purpose

The purpose is to refactor the code to make it more efficient. But we had to start from the very beginning with learning how to set-up the developer tab editor. Then learn how to write syntax to create macros or subroutines, and debug syntax error.

## Analysis and Challenges

After receiving the introduction to VBA, and using the Excel Developer to analyze multiple stocks, worksheets that would be useful for Steve on Wallstreet. One realizes the power of VBA to write code that will automate these analyses for us. Using code to automate tasks decreases the chance of errors and reduces the time needed to run analyses, especially if they need to be done repeatedly.

In the challenge we completed several tasks such as the following:

* Calculate the total daily volume in 2018 using loops
* Calculate the Yearly return for 2018
* Create a new worksheet and subroutine
* Loop over all tickers
* Reuse code
* Format and conditionally format cells
* Make a run button
* Run the analysis for any year
* Lastly, Measure performance

In the Stock Analysis challenge code ran in 0.7890625 for the year 2017 and 0.8046875 seconds for year 2018. After refactoring, the code ran 0.9375 seconds for the year 2017 and 0.1171875 seconds for the year 2018. Shown Below.

2017:

Before:

After:

2018:

Before:

After:

## Results

The refactoring was successful with this data. The VBA macro triggering pop-ups message boxes and inputs, read and change, cell values and formatting cells. Replacing hard- coded values utilizing the concatenation “+” operator. The lines activated in the worksheet twice by the row count and then inside the for loop was improved to efficiently run the macro for both years to make sure it is working correctly and more efficiently. Debugging syntax errors and keeping track of where you are in a nested for loop is key and utilizing comments to reuse code, help you to map out a plan. The improvements were in some of the following areas:

* Activating the sheets versus the file created efficient in locating the worksheets all stocks analysis versus the worksheets by year value.
* The number of Rows counted by Rowcount by cells in a row heading location is faster than by Range with the concatenation joining the string of the column value as these lines activate the worksheet with the stock data: first to get the row count, and then inside the for loop to make sure we're in the right worksheet together concatenation the date in a Initializing the arrays.
* In this way, the need to define the columns can be replaced more efficiently with the Indexing the variables, as this will eliminate the need to define and reference the same variables every time it looks up the year and sheet.
* The ticker string referenced as an index versus a for loop thru a certain amount of cells, defined. The more efficient when you dim a ticker looped thru an array as a set variable utilizing the comparison operator in VBA.

### Challenges and Difficulties Encountered

The most difficult skills learned was syntax and writing readable code for the first time. Overall readability is key to code. Documenting code is done by adding comments. Formatting code involves the use of whitespace. In this case the new macro plan was as flows:

Our changeling was in moving from a new macro that should do the following:

* Format the output sheet on the "All Stocks Analysis" worksheet.
* Initialize an array of all tickers.
* Prepare for the analysis of tickers.
* Initialize variables for the starting price and ending price.
* Activate the data worksheet.
* Find the number of rows to loop over.
* Loop through the tickers.
* Loop through rows in the data.
* Find the total volume for the current ticker.
* Find the starting price for the current ticker.
* Find the ending price for the current ticker.
* Output the data for the current ticker.

Debugging syntax errors and keeping track of where you are in a nested for loop. However, there is a lot of research that can be found in official documentation, stack overflow, Quora or blog post, but for the course it was mostly having other teammates view your code and help debug that was the most beneficial.