Module 2 Challenge – All Stock Analysis VBA Analysis

Date: 10/02/2020

Student Name: Soona Cheon

1. Overview of the project:

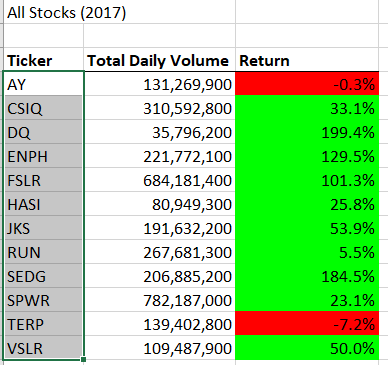
The purpose of the project is using the VBA coding to analyze the stock performance (=return) of 2017, 2018. The 2017 data consists 3004 stock transaction for 12 tickers, and the 2018 works, and 2018 Stock transaction consists 3013 records for the same 12 tickers for comparison.

In this report, I will analyze:

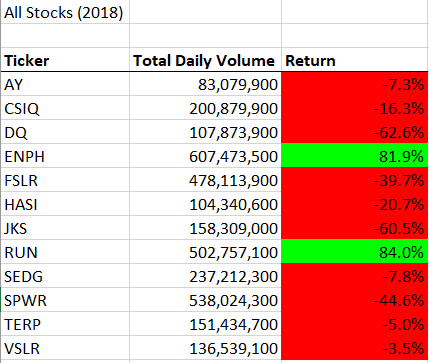
1. Data Analysis
   1. total volume of each ticket in 2017 and 2018
   2. return of each ticker analysis of 2017, and 2018
2. VBA Code Analysis
   1. Performance comparison of AllStockAnalysis Code vs. AllStocksAnalysisRefactored Code
3. Analysis and Challenges
4. The Benefit of VBA
   1. Compare to Excel Pivot Table and Chart, VBA code is more dynamic
   2. By using the VBA, the data analysis was much easier and fast.
5. Useful VBA Code
   1. It is useful to create the button, which can clear the worksheets and run the modules on the specific worksheet
   2. It is also useful to add the function to select “year” to analyze on the worksheets
   3. It is also very useful to add the timer function to analyze the VBA code performance.
6. Challenges
   1. There are many different coding styles in VBA, and the refactored VBA challenge showed that the wrong design causes the slowness of the code performance to process the same data.
7. Results
8. **Data Analysis Outcome**
   * There are 12 tickers:

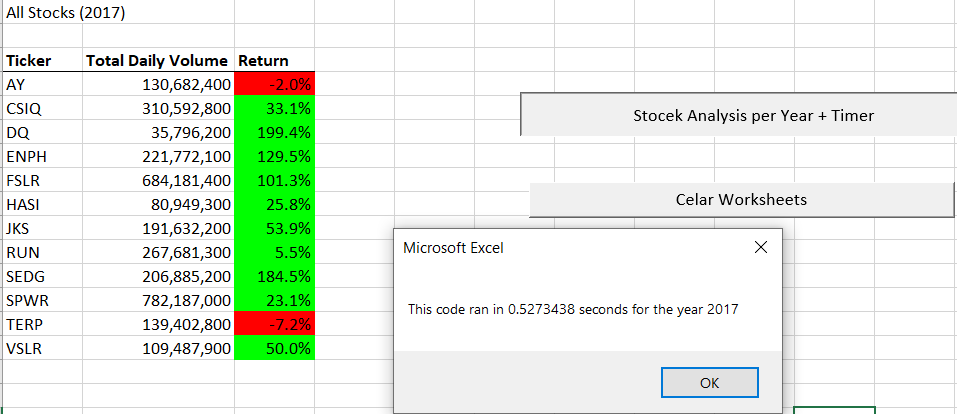
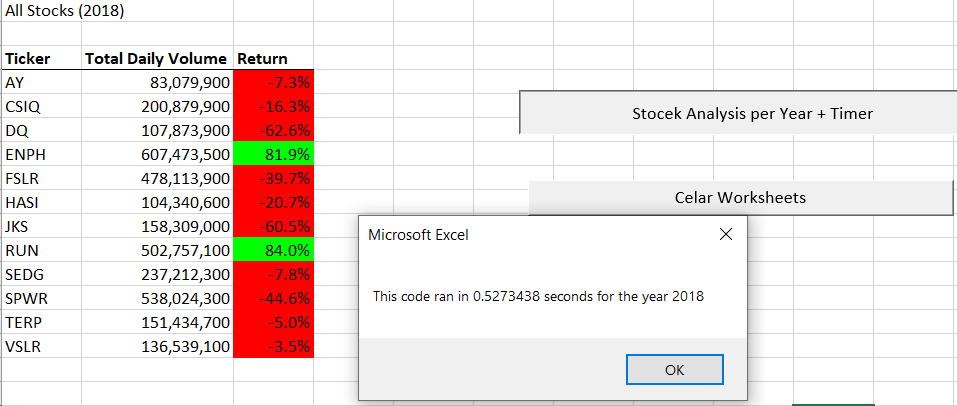
|  |
| --- |
| AY |
| CSIQ |
| DQ |
| ENPH |
| FSLR |
| HASI |
| JKS |
| RUN |
| SEDG |
| SPWR |
| TERP |
| VSLR |

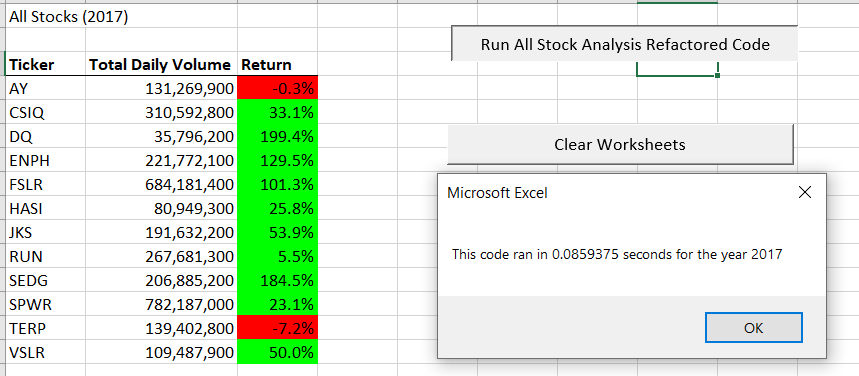
* + The performance of 2017 is:

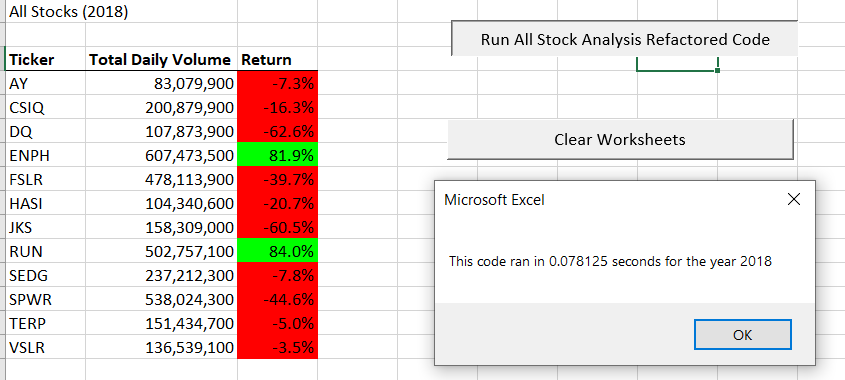


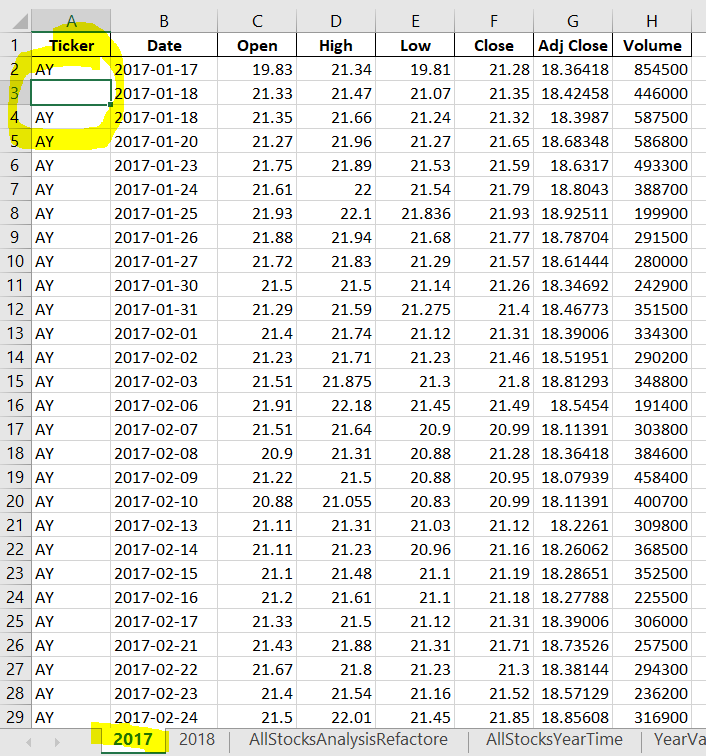
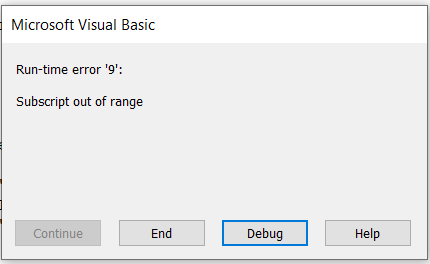
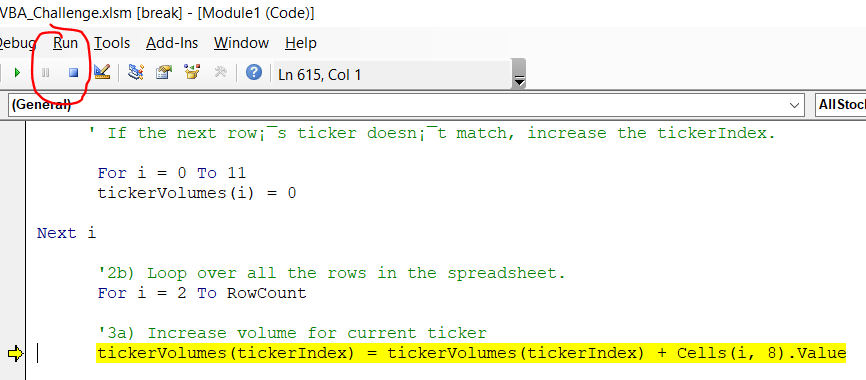
* + The performance of 2018 is:



1. VBA Code Analysis – before refactored
   * The performance of the AllStockAnalysis VBA code before refactored is as below, which was exactly same:
     + 2017 Data Processing Time: 0.5273438 sec
     + 2018 Data Processing Time: 0.5273438 sec
   * 
   * 
2. VBA refactored Code:
   * Changes: In the code, we added the tickerIndex variable, and 3 arrays (tickerVolumes, tickerStatingPrices, ticketEndingPrices), and rewrote the programming logic.
   * **Benefits**: Performance is much faster than before as below
     + 2017 Data Processing Time: 0.0859375 sec
     + 2018 Data Processing Time: 0.078125 sec





* Cons: we need to rewrite the code, by adding arrays and changing the structure of the code, and it is time consuming.
* Pros and Cons (important and interesting): when we have a missing ticker in the data sheet - I found this by accident, and I think it is very important in real world. **I had a missing data on the A3 column in the** 2017 data worksheet , which is missing the ticker name AY.
  + if I run the refactored code, **it will not work**, and **stop & throw an error** **message**. Because, we added tickerIndex variable, and it checks all the ticker, and when there is a missing ticker, the code stops to run. The code is pointing out the 3a), “tickerVolumes(tickerIndex) = tickerVolumes(tickerIndex) + Cells(i, 8).Value”. It means, since the ticker is missing, it does not perform the tickerVolumes array looping. How accurate is this! (all 3 screen shots are as below). Conveniently, it is pointing out the 2017 empty cell for missing AY, so it was easy to fix it.
  + Also, the refactored code cannot run the arrays due to the missing ticker, it **causes the system hang**, until I aborted the macro running and fix/clean the 2017 data. In real world, we deal with imperfect large data, it is good to know the code is pointing out the missing data, and stop to run, but it will be an huge issue if the system just hung until the programmer abort the run!
  + In contrast, the previous code (before refactored), runs **without hanging**, and conveniently **skip the data row**, if there is a data. For example, the AY ticker in A3 in the 2017 data worksheet is missing, it will calculate the AY sum **excluding** the AH (18,42458) vice versa. It is good not to hang the system, but it is not helping data scientists to do the data cleansing to improve accuracy.
  + 
  + 
* 

1. Conclusion:
   1. By refactoring the VBA code adding arrays, we can enhance the performance.

|  |  |  |
| --- | --- | --- |
| Data (Year) | Original Code Speed | Refactored Code Speed ☺ |
| 2017 | 0.5273438 sec | 0.0859375 sec |
| 2018 | 0.5273438 sec | 0.078125 sec |

* 1. Lessons learned: I compared the VBA code before and after refactored as below.

|  |  |  |
| --- | --- | --- |
|  | VBA Code (Before refactored) | VBA Code (after refactored by adding tickerIndex variable, and 3 arrays – tickerVolumes, tickerStartingPrices, tickerEndingPrices) |
| performance | Good | Much faster |
| Function | * unique ticker, * volume per ticker by year, * return calculation, * timer, * output formatting for easy understanding | * unique ticker, * volume per ticker by year, * return calculation, * timer, * output formatting for easy understanding |
| How to work | If there is missing data, it will skip the row and continue tot run | It is checking the tickerIndex in every row and Looping the arrays. If there is missing data, it stops to run the array |
| Pros |  | It is faster  It is pointing out the missing data to help programmers to fix (data cleansing)  It is |
| Cons | * No system hang is good, but does not indicate the missing data | * The code hang, until the it is stopped to execute an fix the data. (biggest disadvantage) |
| Consideration before adopting the VBA Code | * If it is OK, the imperfect data will not be included in the calculation??? | * if my data is clean * if it is OK, the system hung |
| In Real World | It is good for the organization has to deal with imperfect data in fast phase  Good fit for fast moving big data | It is good for the organization, who deals with clean data, willing to stop the code execution and fix the data issue.  Good fit for accurate data |
| For data scientists | It is not pointing out the missing data, so data scientists cannot improve the data inaccuracy | The code refuse to run when there is missing data, so data scientists MUST do the data cleansing prior to the code running, not to hang the system. |

Thank you!

Side Note:

As of October 2nd, the world 3rd largest stock market in Japan had the computer system glitch, and market trading was impaired for a whole day. I wonder if it was arrays related VBA coding? Ha Ha ☺