Module 2 Challenge: VBA of Wall Street

Deliverable Part 2: Written Analysis of the Results

University of Toronto

Data Analytics Bootcamp
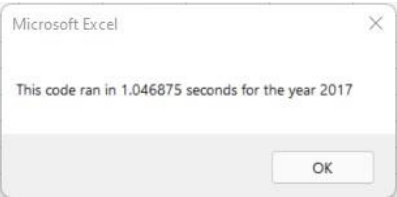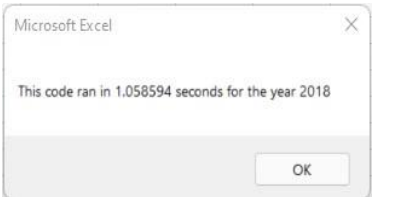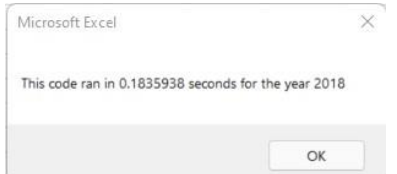

Submitted by: Anureet Kaur Virdi

March 20th, 2022

<u>Overview of the Project</u>

This project consisted of refactoring a Microsoft Excel VBA code to collect information and trends about stocks from green energy companies in the years 2017 and 2018. The data from this analysis would be used to determine if the companies are worth investing in by Steve's parents. This process was originally completed, this attempt was to increase the efficiency of the original code. This report will help gain a more valuable understanding of the stocks of green energy companies. This analysis will help Steve decide and gain knowledge from the raw data to plan future investments.

<u>Results</u>

An instruction code script was very useful in understanding and working on this project. The project included finding a Total Daily Volume for all the companies, finding DQ's Yearly Return, and comparing the data for the two given years. Refactoring the code helped by making the process faster and getting all the calculations and the formatting of the data results in one code. The following are some screenshots documenting the resulting analysis.

| | **2017** | **2018** |
|---|---|---|
| **Original Code** | Microsoft Excel ✕ <br><br> This code ran in 1.046875 seconds for the year 2017 <br><br> OK | Microsoft Excel ✕ <br><br> This code ran in 1.058594 seconds for the year 2018 <br><br> OK |
| **Refactored Code** | Microsoft Excel ✕ <br><br> This code ran in 0.1875 seconds for the year 2017 <br><br> OK | Microsoft Excel ✕ <br><br> This code ran in 0.1835938 seconds for the year 2018 <br><br> OK |

As evident from the run-time screenshots, the refactored code is efficient and orderly as compared to the original code written during the module practice. Similarly,

the output in the data tables was also organized and formatted correctly when prompted from the refactored code.

## Summary

Advantages and Disadvantages of the Refactoring Code:

The refactored code's both input and output were clean and organized. Its' advantages included debugging, faster run-time, and design and software improvement.

It is beneficial for new users of the data too because it is easier to read and concise. But some of its disadvantages are that refactoring is not always available, because there can be large applications, or having a small sample data that can risk the analysis and findings.

Refactoring Stock Analysis:

In this case, refactoring was very useful because it made the process faster and more convenient. The original analysis took one second to run, whereas the refactored code took one-fifth of the time. Also, the original code did not have the formatting included in it, so whenever the output was generated, it did not color-coded the results, which made it hard to look at the overall results.