# Green Stock Analysis with VBA

## Overview of Project

In this project we will analyze the performance of various stocks over different years. We are working on improving the performance of the code so we can run more complex analysis on larger data sets.

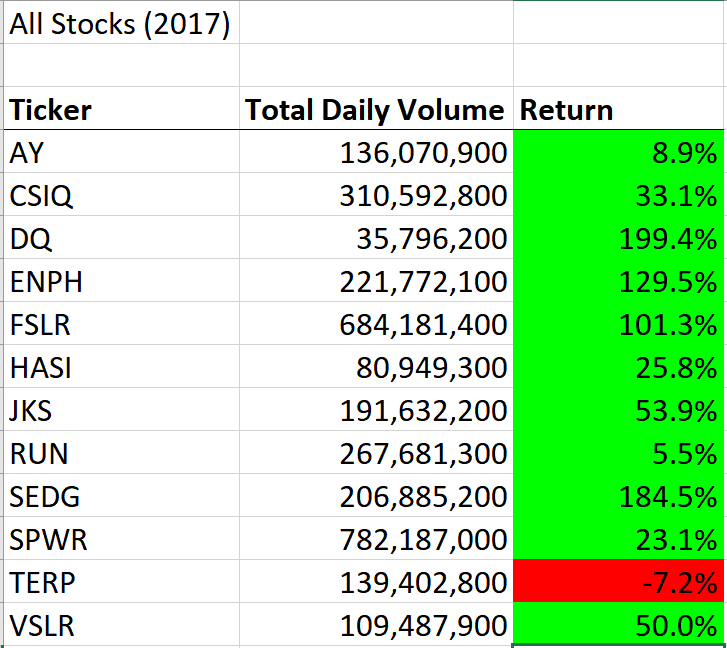
### Purpose

The purpose of the project is to help Steve analyze the performance of various green energy stocks so he can assist his parents with their investment plan. We measured the total daily trade volume and the yearly return for various stocks for the years 2017 and 2018. This helped Steve make investment recommendations. However, Steve wants to expand his analysis so we will work on improving our code so we can analyze a larger amount of companies.

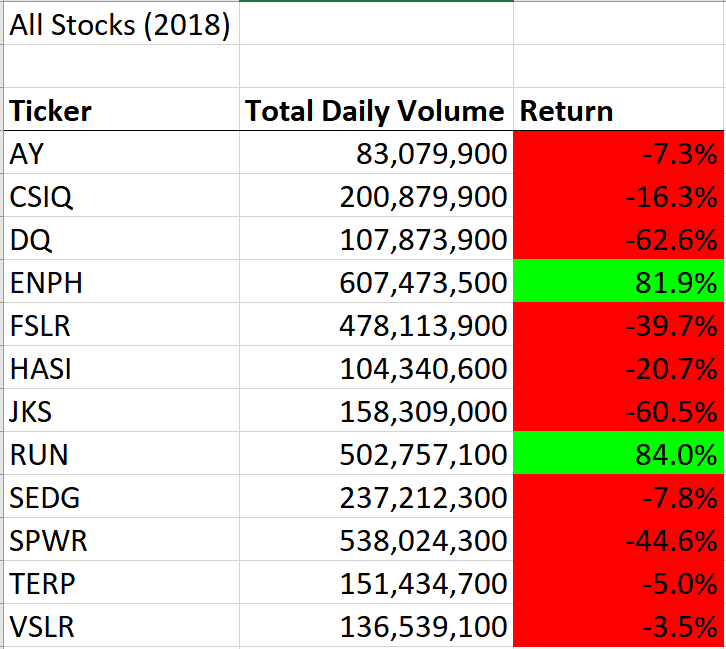
## Analysis and Challenges

### Analysis Outcomes

First, we will run our analysis for the year 2017. Because we formatted our output to highlight positive returns green and negative returns red, we notice immediately that this was a very good year for the stocks we are analyzing. All the stocks except TERP had positive returns this year; with multiple having gains over 100%. The average gain for all stocks was 67.3% and trading volume is consistently high. Looking at only 2017 this looks like a particularly good set of stocks for investing in, but we will continue our analysis by looking at 2018.



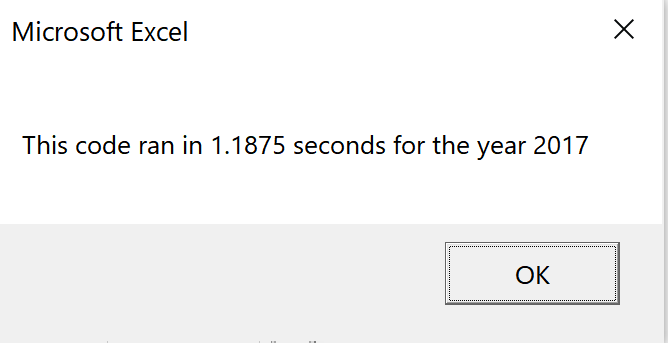
When looking at the analysis for 2018 we quickly notice the results are very different. In these results we see that the returns are nearly all negative, and over half of the negative stocks had double-digit losses. Only ENPH and RUN had positive returns in 2018. The trading volume did not change much from 2017 and is still consistently high for all stocks analyzed.

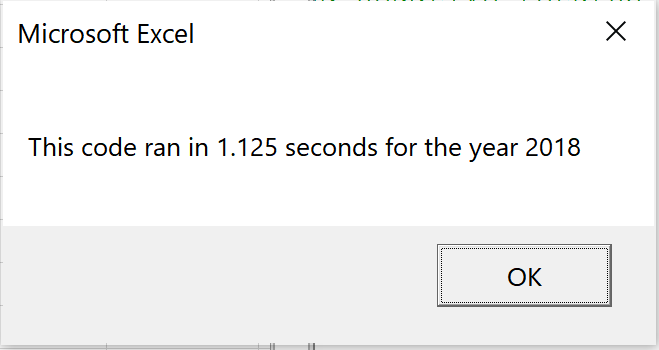


Our analysis shows that although the selected stocks looked promising in 2017, as we move onto 2018 we see these are actually not high performing companies. Steve should recommend to his parents not to invest in any of these green stocks with the exceptions of ENPH and RUN. ENPH had exceptional growth both years, with a yearly return of 129.5% in 2017 and 81.9% in 2018. I would recommend Steve do more due diligence before recommending his parents invest in ENPH, but it could be an extremely profitable investment if it continues to grow anywhere near its current rate. RUN had a positive return in 2017 but then increased that return greatly in 2018. It is another stock worth looking into investing in.

### Performance Comparison

Our original code provided the correct results we needed, but we wanted to make it more efficient so we could run it over a larger data set without processing or time issues. Running the code took over a second every time and was usually between 1.1 and 1.3 seconds.





After refactoring our code, it would almost always run in under one second. It would usually run between 0.75 and 1 second. Although the time difference is small, it is a significant improvement in the code. This slight time saving will compound when we run the code over larger and larger data sets improving our analysis on future data sets Steven may want to look at.

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

## Summary

- What are the advantages or disadvantages of refactoring code?

The advantage of refactoring code is that you can make your code more efficient. By rearranging the structure of your code you may be able to save time, memory space, and processing power. Refactoring code may allow you to work with larger data sets as well. Refactoring code does have disadvantages though. It can be time-consuming, so if there is not significant improvement in the code it may not have been worth it. Refactoring code also allows for an introduction of bugs that cause problems in your code that you were not having before.

- How do these pros and cons apply to refactoring the original VBA script?

The advantage of refactoring our code becomes apparent when we compare the time it takes to run each version. Our refactored code runs quicker than our original code. This will be beneficial when we try to analyze a larger set of tickers. The disadvantage of refactoring code became apparent to me towards the end of my work on the project. I had a bug in my refactored code and was getting only zeroes for my output. It took hours of investigating before finding the one line of code with a very simple mistake that was causing the problem. Refactoring the code caused issues with it and it took work getting the code running again, but in the end there was a significant improvement to the code.