# STOCK\_ANALYSIS

## OVERVIEW

Steve a recent graduate in finance has his first clients interested in investing into green energy stocks. So he has decided to take a deep look into a few green energy stocks data in order to find the stocks that would be profitable.

### Purpose

In this analysis we will be creating an excel file, use code to find information needed and finally conduct an analysis on a few number of stocks.

## RESULTS

### Stock performance 2017

The analysis found that in the year 2017 a crushing majority of the stocks that were analysed made a good return on investment with profit ranging from 1 to 3 digits and only one stock plunged in a one digit negative

Below is a screenshot of the 2017 results

![VBA\_Challenge\_2017](https://github.com/AminataMiller/stock-analysis/blob/main/Resources/VBA\_Challenge\_2017.png)

Graphical user interface, application, table, Excel

Description automatically generated

### Stock performance 2018

The analysis also found that in the year 2018 all but two stocks fell below the negative return on investment down to -62 % and only two stocks remained profitable up to 84% return on investment

Below is the illustration

![VBA\_Challenge\_2018](https://github.com/AminataMiller/stock-analysis/blob/main/Resources/VBA\_Challenge\_2018.png)

Graphical user interface, application, table, Excel

Description automatically generated

### Comparition of stock performance

Compared to 2017, the year 2018 was far less profitable for the vast majority of the stocks that we analysed with only two stocks remaining above 80% of return on investment.

It is also worth noting that those two (ENPH and RUN) had a significant change in either direction with the former going from 129.5% of profit in 2017 down to 81.9% in 2018 and the latter going from 5.5% in 2017 up to 84% in 2018.

Using the following code:

tickerVolumes(tickerIndex) = tickerVolumes(tickerIndex) + Cells(i, 8).Value

And also:

For i = 0 To 11

Cells(4 + i, 1).Value = tickers(i)

Cells(4 + i, 2).Value = tickerVolumes(i)

Cells(4 + i, 3).Value = tickerEndingPrices(i) / tickerStartingPrices(i) - 1

We were able to find the total volume as well as the return for each ticker.

Here is a screenshot of both years performance

![VBA\_Challenge\_2017](https://github.com/AminataMiller/stock-analysis/blob/main/Resources/VBA\_Challenge\_2017.png)

![VBA\_Challenge\_2018](https://github.com/AminataMiller/stock-analysis/blob/main/Resources/VBA\_Challenge\_2018.png)

### Comparition of execution time

When running, the original script seems to take more time to execute compared to the refactored script.

![2017\_stock\_performance](https://github.com/AminataMiller/stock-analysis/blob/main/Resources/2017\_stock\_performance.png)

A picture containing table

Description automatically generated

![VBA\_Challenge\_2017](<https://github.com/AminataMiller/stock-analysis/blob/main/Resources/VBA_Challenge_2017.png>)

Graphical user interface, application, table, Excel

Description automatically generated

![2018\_stock\_performance](<https://github.com/AminataMiller/stock-analysis/blob/main/Resources/2018_stock_performance.png>)

Table

Description automatically generated with medium confidence

![VBA\_Challenge\_2018](https://github.com/AminataMiller/stock-analysis/blob/main/Resources/VBA\_Challenge\_2018.png)

Graphical user interface, application, table, Excel

Description automatically generated

## SUMMARY

1. The advantages of refactoring code are:

- It makes the code run faster

- It makes the code shorter and easier to understand with fewer steps

The main disadvantage is it will take you time to rethink how to refactor

2. The advantages of refactoring our original VBA:

- The refactored subroutine took few different ones from the original script and made them into one

- That made the code run faster

The disadvantages we encoutered:

- Too long of a subroutine creating potential identation issues

- Risks of making mistakes or getting confused within certain lines of the script