# 2017 – 2018 Stock analysis and comparison

### The purpose of this analysis is to outline stock performances from 2017 to 2018. The script used for this analysis creates a readable table that shows the 12 stock options and their overall returns. The goal of this analysis is to guide the reader towards an informed decision of which stocks have a greater chance of positive returns.

##Results

Between 2017 and 2018 there are a lot of major differences.

Table, Excel

Description automatically generated Table

Description automatically generated

First, 2017 shows a good year for profit. Almost all but one had positive returns. The highest being with DQ at almost 200%. And the lowest at -7.2%. If you invested in that year there is a good chance on making a profit. The thig with stocks however is that te market if very volitile. Depending on the country, politics, and sometimes enviromental factors like weather and natural disasters can play a major role in stocks. With this data set we don’t have a list of events that may have affected it but the asumption that we ca draw from the results is that it wasn’t good.

Looking at DQ for both you can see the dramatic drop in total daily volume from 2017-2018. This changes the returns from 199.4% down to -62.6%. Obviously for anyone that’s looking into that stock would think that in 2019 it’s not as great bet.

However if you look at 2018 as a whole the general story would read that that isn’t a great year for investing in stocks.

Refactoring the original script it did make an improvement excecution time wise but not by much.

Graphical user interface, text, application, website

Description automatically generated Graphical user interface, text, application

Description automatically generated

(Original Script) (Refactored Script)

Graphical user interface, text, application, email, website

Description automatically generated Graphical user interface, text, application, table

Description automatically generated

(Original Script) (Refactored Script)

For both sets the script ran at less than 1 second, but the refactored did technically run more quickly. With the data set that we were working with it was a good amount of code, but it is a great example of the phrase “If it broke, don’t fix it.”

##Advantages Vs. Disadvantages of refactoring code.

### When comparing the original script code with the new script it has a lot of the same functions. But even though a script runs fine it doesn’t mean that it couldn’t be improved.

##Advantages

###Directly looking at the original script vs the refactored it’s better to shed some of that code so that it runs a little more streamline. In the original script we use a nested for loop.

For i = 0 to 11

Ticker = tickers(i)

Totalvolume = 0

Worksheets(“2018”).activate

For j = 2 to rowcount

Vs. in the new script it’s stacked.

For i = 0 to 11

Tickervolumes(i) = 0

Next i

For i = 2 to rowcount

That way we are not adding additional sets that it loops through. It’s more direct. That way it saves on a bit of time as well. That’s why you can also tell that the new script it is less bulky. Even though they are looking for the same thing, the program is not reading through as name lines of code.

##Disatvantages

###Specifically in VBA if you want to fine turn a specific script you must be very careful when analyzing what can be changed and what can’t. Sometimes with making small changes in the beginning it can completely throw off the results because of a “/”. It’s a lot about playing around with the syntax and seeing how it responds. Because of that refactoring can take hours longer than when you wrote the original script. If you refactor just take into consideration how much time and effort you want to put into it. It’s a lot about trial and error.

##How do the pros and cons apply here?

### To me it’s straight forward that the new script, although did run faster and has less lines of code. Did take longer to format and debug to execute. With the only difference being less then 1 second on execution time. For this script I don’t think refactoring was really needed. For larger data sets however it might be necessary to cut down a script say from 30 seconds to 2. Ultimately it depends on the project in question.