# Overview of Project

## Purpose

### This project aimed to refactor a Microsoft Excel VBA code to collect certain stock information for the years 2017 and 2018. The information was to determine whether the stocks are worth investing in or not.

The data that is presented in the excel workbook includes two charts with stock information on twelve different stocks. The 2017 and 2018 workbooks contain a ticker value, the date the stock was issued, the opening, closing, adjusted closing price, the highest and lowest price, and the volume of the stock. The aim is to recover the ticker, the total daily volume, and the return on each stock.

# Results

## Analysis

### For the stocks of 2017, we can see that DQ provided a 199.4% return and is the highest performing stock while TERP is an underperforming stock.

**Graphical user interface, application, table, Excel

Description automatically generated**

For 2018, we can see that RUN is a high-performing stock and ENPH is a good investment too. All other stocks performed poorly.

Graphical user interface, application, table, Excel

Description automatically generated

Sub AllStocksAnalysisRefactored()

Dim startTime As Single

Dim endTime As Single

yearValue = InputBox("What year would you like to run the analysis on?")

startTime = Timer

'Format the output sheet on All Stocks Analysis worksheet

Worksheets("All Stocks Analysis").Activate

Range("A1").Value = "All Stocks (" + yearValue + ")"

'Create a header row

Cells(3, 1).Value = "Ticker"

Cells(3, 2).Value = "Total Daily Volume"

Cells(3, 3).Value = "Return"

'Initialize array of all tickers

Dim tickers(12) As String

tickers(0) = "AY"

tickers(1) = "CSIQ"

tickers(2) = "DQ"

tickers(3) = "ENPH"

tickers(4) = "FSLR"

tickers(5) = "HASI"

tickers(6) = "JKS"

tickers(7) = "RUN"

tickers(8) = "SEDG"

tickers(9) = "SPWR"

tickers(10) = "TERP"

tickers(11) = "VSLR"

'Activate data worksheet

Worksheets(yearValue).Activate

'Get the number of rows to loop over

RowCount = Cells(Rows.Count, "A").End(xlUp).Row

'1a) Create a ticker Index

tickerIndex = 0

'1b) Create three output arrays

Dim tickerVolumes(12) As Long

Dim tickerStartingPrices(12) As Single

Dim tickerEndingPrices(12) As Single

''2a) Create a for loop to initialize the tickerVolumes to zero.

For i = 0 To 11

tickerVolumes(i) = 0

tickerStartingPrices(i) = 0

tickerEndingPrices(i) = 0

Next i

''2b) Loop over all the rows in the spreadsheet.

For i = 2 To RowCount

'3a) Increase volume for current ticker

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

'3b) Check if the current row is the first row with the selected tickerIndex.

'If Then

If Cells(i - 1, 1).Value <> tickers(tickerIndex) And Cells(i, 1) = tickers(tickerIndex) Then

tickerStartingPrices(tickerIndex) = Cells(i, 6).Value

End If

'End If

'3c) check if the current row is the last row with the selected ticker

'If the next row’s ticker doesn’t match, increase the tickerIndex.

'If Then

If Cells(i + 1, 1).Value <> tickers(tickerIndex) And Cells(i, 1) = tickers(tickerIndex) Then

tickerEndingPrices(tickerIndex) = Cells(i, 6).Value

End If

'3d Increase the tickerIndex.

If Cells(i, 1).Value = tickers(tickerIndex) And Cells(i + 1, 1).Value <> tickers(tickerIndex) Then

tickerIndex = tickerIndex + 1

End If

'End If

Next i

'4) Loop through your arrays to output the Ticker, Total Daily Volume, and Return.

Worksheets("All Stocks Analysis").Activate

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

Next i

'Formatting

Worksheets("All Stocks Analysis").Activate

Range("A3:C3").Font.FontStyle = "Bold"

Range("A3:C3").Borders(xlEdgeBottom).LineStyle = xlContinuous

Range("B4:B15").NumberFormat = "#,##0"

Range("C4:C15").NumberFormat = "0.0%"

Columns("B").AutoFit

dataRowStart = 4

dataRowEnd = 15

For i = dataRowStart To dataRowEnd

If Cells(i, 3) > 0 Then

Cells(i, 3).Interior.Color = vbGreen

Else

Cells(i, 3).Interior.Color = vbRed

End If

Next i

endTime = Timer

MsgBox "This code ran in " & (endTime - startTime) & " seconds for the year " & (yearValue)

End Sub

# Summary

## Advantages and Disadvantages of Refactoring Code

### To make our code more organized and easier to go through, refactoring is a good process. Some of the advantages include better debugging and faster programming. Also, it becomes much easier for the end-user to go through and make any edits or understand the code in a much concise and easier manner.

### Some disadvantage is that it can be difficult to work out a logic to refactor codes and multiple test cases or iterations may be required.

## Advantages and Disadvantages of the original and Refactored VBA script

Main advantage is that it decreases the macro run time and provides results much faster. The disadvantage is the complicated coding process which can be time consuming.