

Project focusing on Wall Street Data via VBA Scripting

Data Bootcamp – Homework-Excel with VBA Macros

kimberly GORDON

2018

Contents

Summary

* + - Figure 1 – 2016 Data
    - Figure 2 – 2015 Data
    - Figure 3 – 2014 Data
    - Figure 4 – Project VBA code
    - Figure 5 – VBA to loop sheets in workbook

Summary

Following, are the exercises that focused on key elements that can be used in Excel VBA scripting to analyze and present data to users. The complexity of the graphs is based on expert use of excel and clean presentation styles. The functionalities that were used included:

* Looping of worksheets together where all macros can be executed using F5
* Multiple If/Then Loops
* Formatting of data using Rounding and symbol inserts, such as the percentage sign
* Cell highlighting magnifying positive and negative impacts in the market
* Grouping of information showing yearly changes per stock type

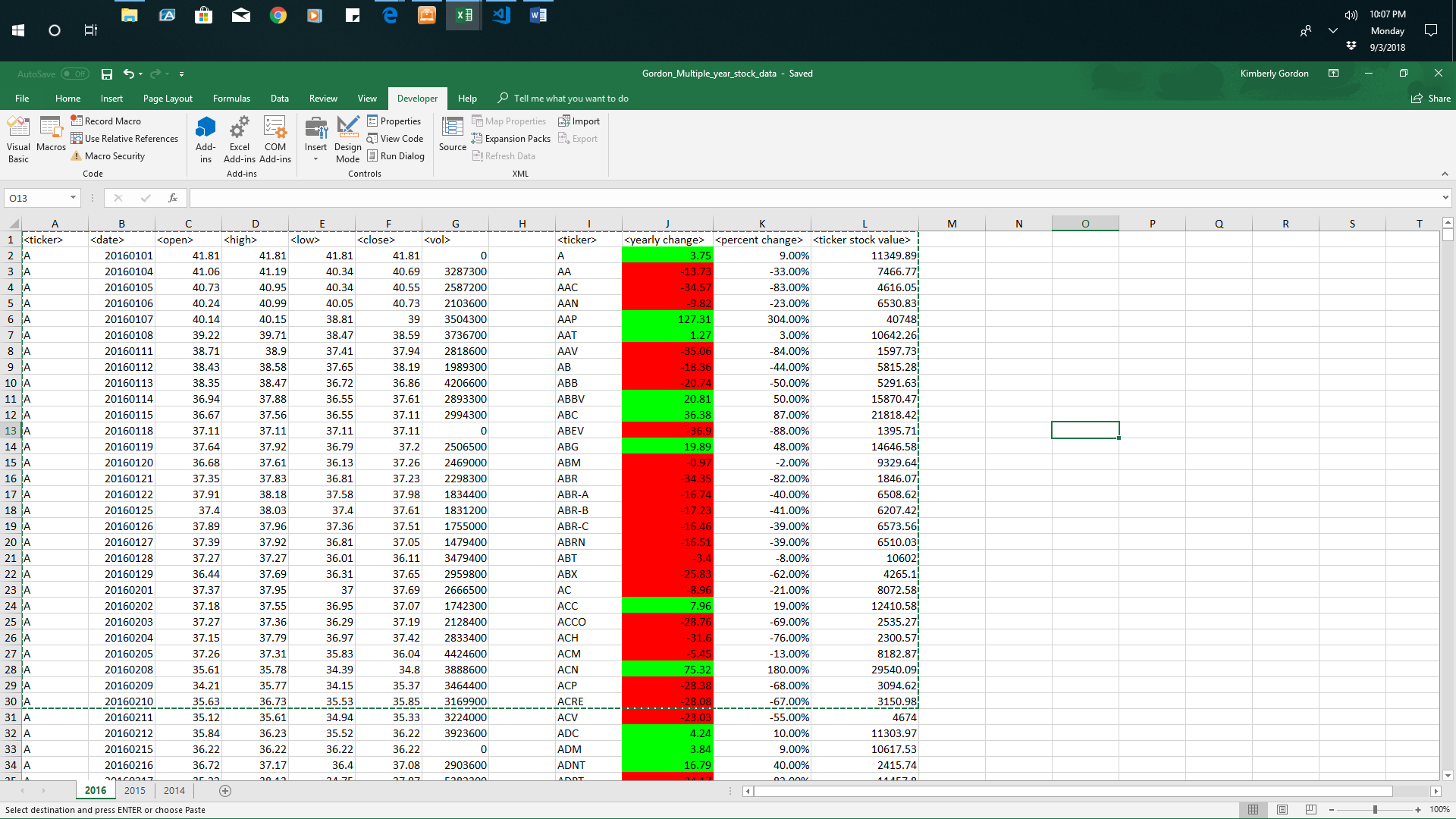


Figure 1- Exercise 1 – Snippet of 2016 data

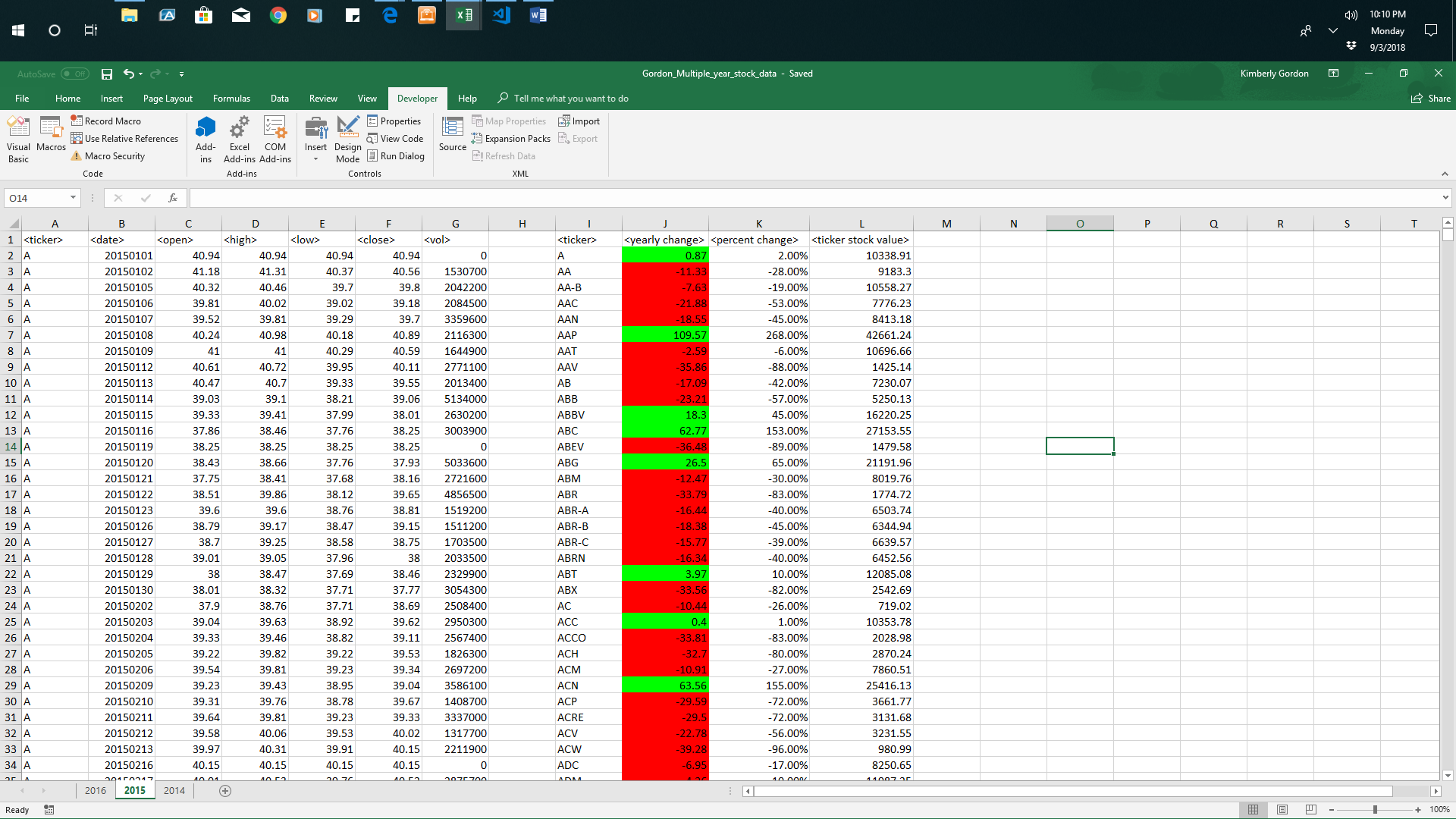


Figure 2- Exercise 2 – Snippet of 2015 data

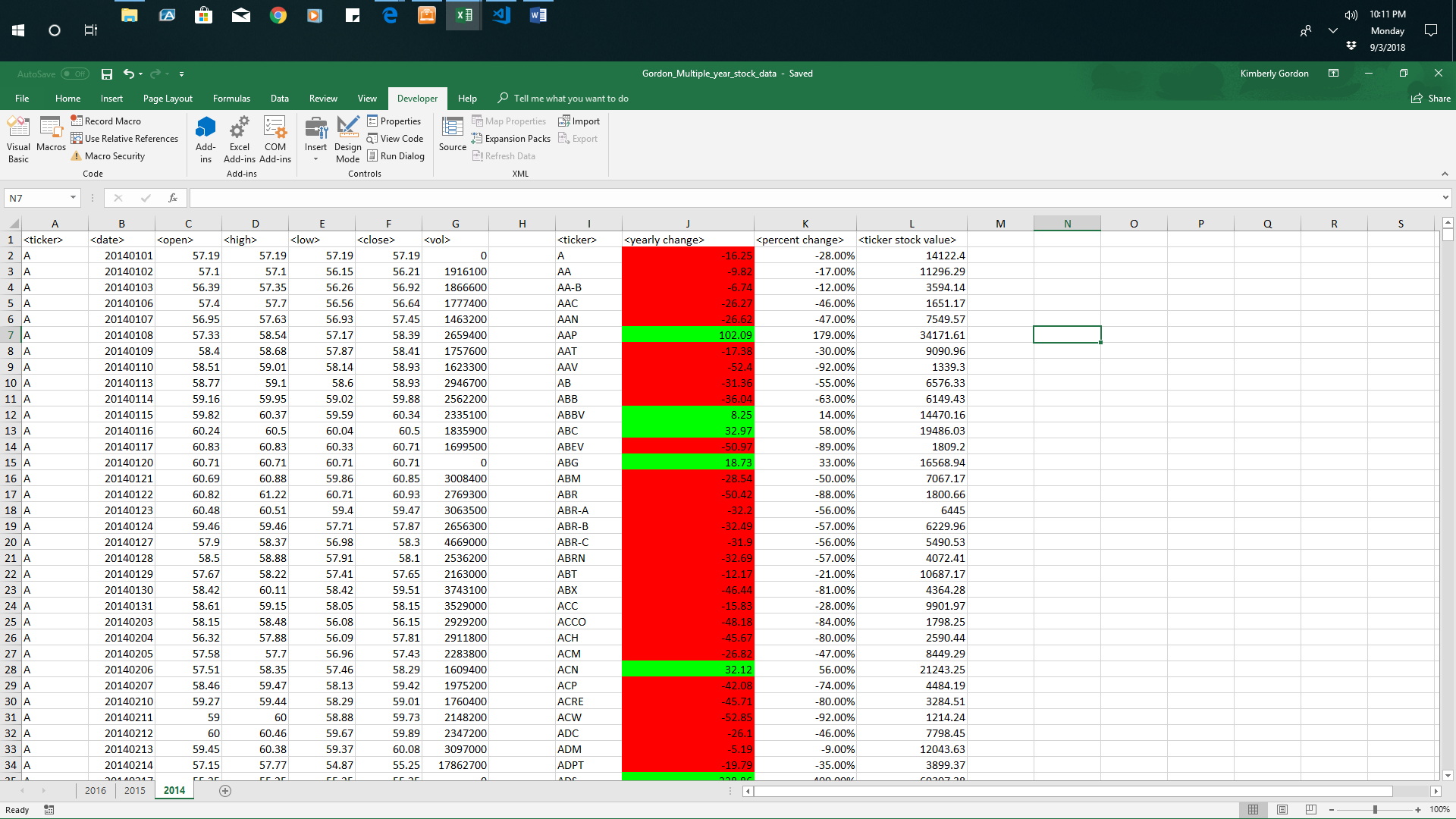


Figure 3- Exercise 3 – Snippet of 2014 data.

Figure 4- VBA Code for project

Sub wall\_street()

'Part2 - Create a script that will loop through all the stocks and take the following info.

'Yearly change from what the stock opened the year at to what the closing price was.

'The percent change from the what it opened the year at to what it closed.

'The total Volume of the stock

'Ticker Symbol

'You should also have conditional formatting that will highlight positive change in green and negative change in red.

' Set an initial variable for holding the type of stock (brand name)

Dim Stock\_Type As String

' Set an initial variable for holding the total per each stock (credit card brand)

Dim Stock\_Total As Double

Stock\_Total = 0

' Set initial variable for holding the total change per each stock

Dim Stock\_Change As Double

Stock\_Change = 0

' Keep track of the location for each stock in the summary table

Dim Summary\_Table\_Row As Integer

Summary\_Table\_Row = 2

' Loop through all stock listings

lastrow = Cells(Rows.Count, 1).End(xlUp).Row

For I = 2 To lastrow

' Check if we are still within the same stock type, if it is not...

If Cells(I + 1, 1).Value <> Cells(I, 1).Value Then

' Set the Brand name

Stock\_Type = Cells(I, 1).Value

' While in each row, loop through each stock change column

For j = 3 To 6

Stock\_Beg = Cells(2, 3).Value

Next j

' Set the Change initial Value

'Stock\_Beg = Cells(i, 3).Value

Stock\_End = Cells(I, 6).Value

Stock\_Change = Stock\_End - Stock\_Beg

Stock\_Percent = Format(Round(((Stock\_Change / Stock\_Beg)), 2), "Percent")

'Stock\_Percent = Format(Round(((Stock\_Change / Stock\_Beg) \* 100), 2), "Percent")

' Add to the Brand Total

Stock\_Total = Stock\_Total + Cells(I, 3).Value

' Print the Credit Card Brand in the Summary Table

Range("I" & Summary\_Table\_Row).Value = Stock\_Type

'Print the Stock\_Change in the Summary Table

Range("J" & Summary\_Table\_Row).Value = Stock\_Change

Set r1 = Range("J" & Summary\_Table\_Row)

If r1.Value >= 0 Then r1.Interior.Color = vbGreen

If r1.Value < 0 Then r1.Interior.Color = vbRed

'Print the Stock\_Change in the Summary Table

Range("K" & Summary\_Table\_Row).Value = Stock\_Percent

' Print the Brand Amount to the Summary Table

Range("L" & Summary\_Table\_Row).Value = Stock\_Total

' Add one to the summary table row

Summary\_Table\_Row = Summary\_Table\_Row + 1

' Reset the Brand Total

Stock\_Total = 0

' If the cell immediately following a row is the same brand...

Else

' Add to the Brand Total

Stock\_Total = Stock\_Total + Cells(I, 3).Value

'Add to the Stock\_Change (- Cells(i, 6).Value

' Stock\_Change = Stock\_Change

End If

Next I

End Sub

Figure 4- VBA Code for looping worksheets

Sub WorksheetLoop()

Dim WS\_Count As Integer

Dim I As Integer

' Set WS\_Count equal to the number of worksheets in the active

' workbook.

WS\_Count = ActiveWorkbook.Worksheets.Count

' Begin the loop.

For I = 1 To WS\_Count

' Insert your code here.

' The following line shows how to reference a sheet within

' the loop by displaying the worksheet name in a dialog box.

MsgBox ActiveWorkbook.Worksheets(I).Name

Next I

End Sub