VBA-challenge (WK2 Assignment): The VBA of Wall Street

Background

You are well on your way to becoming a programmer and Excel master! In this homework assignment you will use VBA scripting to analyze real stock market data. Depending on your comfort level with VBA, you may choose to challenge yourself with a few of the challenge tasks.

Approach

The VBA challenge was solved by using building blocks, i.e. several subroutines, from the class material.

Fortunately the data is sorted by Ticker so the first task was to identify where ticker symbol changes occur. This was achieved with following code comparing row i+1 with i:

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

If this condition is true then a series of updates to a summary table occur including the ticker symbol itself, a rolling sum of the volume (which updates each iteration of the loop) and a calculation of the price change which is the last close price minus the first open price. If the condition is not met the subroutine progress to the next ticker. The data loops through every row of data in that worksheet until the summary table is complete by:

For i = 2 To ws.Cells(Rows.Count, 1).End(xlUp).Row

The subroutine is adapted for use with any sheet of data by determining the last row of data with the expression Cells(Rows.Count, 1).End(xlUp).Row – this of course assumes the first row is 2.

Specific challenges encountered in the first phase included the correct dimensioning of variables which initially caused overflow errors (e.g. Volume).

The subroutine is further adapted to cycle through each sheet(of any Workbook)using the loop defined by:

For Each ws In Worksheets

This line loops through each sheet in the workbook and requires setting a variable

Set ws = ThisWorkbook.Worksheets(1)

…where Worksheets(1) is the first worksheet irrespective if its name (“A” or “2014” are irrelevant). All objects now need the suffix ws to indicate the current worksheet e.g. ws.range( or ws.cells(

Specific challenges encountered in the second phase included an divide by zero error as a result a ticker with a first open price value of zero (in fact there were all zeros in this data set). An if statement to check for a zero value for the first open price is used, and if TRUE, sets the PriceChangePercent variable to 0 without performing the calculation thus avoiding a fatal error.

The subroutine was then enhanced to return the stock with the "Greatest % increase", "Greatest % decrease" and "Greatest total volume". Six new variables which update themselves at each Ticker change. The update is triggered by an if statement comparing the current Ticker volume to largest volume of all previous Tickers.

If Volume > GreatestVolume Then

GreatestVolume = Volume

GreatestVolumeTicker = Ticker

Else: GreatestVolume = GreatestVolume

End If

Similar logic is used for GreatestPriceIncrease% and GreatestPriceDecrease% variables.

The subroutine should run on any worksheet irrespective of the number of sheets, number of tickers or number of sheets. If the data does not start at line 2 this would trigger an error but could be coded to find the first row of data if necessary.

I have submitted only one set of code – the final solution, but I developed it in the way described above – in 3 steps. The code runs in less than one minute on the larger, multi-year file.

John Russell

2nd April, 2021