Sub Stock\_Market\_Analysis():

' The below script will run an analysis on the generated stock market data and produce two summary tables with the findings.

' The variables referenced to multiple times in this script are declared below:

Dim ws As Worksheet

Dim Ticker\_Symbol As String

Dim Yearly\_Change As Double

Dim Opening\_Price As Double

Dim Closing\_Price As Double

Dim Percent\_Change As Double

Dim Total\_Stock\_Volume As Double

Dim i As Double

Dim j As Double

Dim Last\_Row As Double

Dim Summary\_Table\_Row As Integer

Dim Greatest\_Percentage\_Increase As Double

Dim Greatest\_Percentage\_Decrease As Double

Dim Greatest\_Total\_Volume As Double

' Begins loop which will run through each worksheet in the workbook.

For Each ws In ThisWorkbook.Sheets

' Populates the headings required for each worksheet.

ws.Range("I1").Value = "Ticker"

ws.Range("J1").Value = "Yearly Change ($)"

ws.Range("K1").Value = "Percent Change"

ws.Range("L1").Value = "Total Stock Volume"

ws.Range("P1").Value = "Ticker"

ws.Range("Q1").Value = "Value"

ws.Range("O2").Value = "Greatest % Increase"

ws.Range("O3").Value = "Greatest % Decrease"

ws.Range("O4").Value = "Greatest Total Volume"

' Formats the above heading's interior and font to be clearly distinguishable from the raw data.

ws.Range("I1:L1,O1:O4,P1:Q1").Font.Color = RGB(255,255,255)

ws.Range("I1:L1,O1:O4,P1:Q1").Interior.Color = RGB(0,0,0)

' Defining the variables for the currently observed worksheet:

Last\_Row = ws.Cells(Rows.Count, 1).End(xlUp).Row ' Defines the last populated row in column one of the current sheet.

Summary\_Table\_Row = 2 ' Defines the initial row that each unqiue ticker and its' yearly change, percent change, and total stock volume are inserted into.

Total\_Stock\_Volume = 0 ' Assigns the total stock volume variable to a initial value of 0 to the current sheet observed.

Opening\_Price = ws.Cells(2, 3).Value ' Defines the intial value of the opening price.

' Begins a loop to recover the values needed for our summary table.

For i = 2 To Last\_Row

Closing\_Price = ws.Cells(i, 6).Value ' Defines closing price as the current row's - column 6 value, has to be done within the loop as reliant on i's value.

If ws.Cells(i + 1, 1).Value <> ws.Cells(i, 1).Value Then ' Conditional: checks if the vertically neigbouring ticker symbols observed in column one differ within the set range.

Ticker\_Symbol = ws.Cells(i, 1).Value ' When condition is met, the uppermost obervation is stored within the Ticker\_Symbol variable.

Total\_Stock\_Volume = Total\_Stock\_Volume + ws.Cells(i, 7).Value ' When condition is met, the total volume in the current row's column is added onto the value currently stored in Total Volume (0).

Yearly\_Change = Closing\_Price - Opening\_Price ' When condition is met, the opening\_price is deducted from the closing price and stored within the Yearly\_Change variable.

' When condition is met, displays the value's stored in the Ticker\_Symbol, Yearly\_Change, and Total\_Stock\_Volume Variables into the appropritate location in the summary table.

ws.Range("I" & Summary\_Table\_Row).Value = Ticker\_Symbol

ws.Range("L" & Summary\_Table\_Row).Value = Total\_Stock\_Volume

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

'Nested loop helps apply conditional formatting to negative and positive yearly change values as they are recorded in the summary table.

If ws.Range("J" & Summary\_Table\_Row) > 0 Then ' Conditional: If the observed value in the summary table row/ yearly change column is positive...

ws.Range("J" & Summary\_Table\_Row).Interior.Color = RGB(0,255,0) ' Sets that yearly change cell's interior to the colour green. RGB used as a safeguard in case the colour palette indexes change in future.

ElseIf ws.Range("J" & Summary\_Table\_Row).Value < 0 Then ' Conditional: If the observed value in the summary table row/ yearly change column is negative...

ws.Range("J" & Summary\_Table\_Row).Interior.Color = RGB(255,0,0) ' Sets that yearly change cell's interior to the colour red.

End If

' Percent change variable is assigned to equal the (closing-opening)/ opening. Where (closing - opening) is the yearly change already computed.

Percent\_Change = Yearly\_Change / Opening\_Price

' When condition is met, the percent change is displayed in the summary table and formatted as a percentage number format.

ws.Range("K" & Summary\_Table\_Row).Value = Percent\_Change

ws.Range("K" & Summary\_Table\_Row).NumberFormat = "0.00%"

'Nested loop helps apply conditional formatting to negative and positive percentage change values as they are recorded in the summary table.

If ws.Range("K" & Summary\_Table\_Row) > 0 Then ' Conditional: If the observed value in the summary table row/ percentage change column is positive...

ws.Range("K" & Summary\_Table\_Row).Interior.Color = RGB(0,255,0) ' Sets that yearly change cell's interior to the colour green. RGB used as a safeguard in case the colour palette indexes change in future.

ElseIf ws.Range("K" & Summary\_Table\_Row).Value < 0 Then ' Conditional: If the observed value in the summary table row/ percentage change column is negative...

ws.Range("K" & Summary\_Table\_Row).Interior.Color = RGB(255,0,0) ' Sets that yearly change cell's interior to the colour red.

End If

' Adds one onto the summary table row so next time the condition is met in the loop, the values can be displayed in the correct position.

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

' When condition is met, after i = 2 is met in the loop, the opening price will always be in the row below the one being observed. The below helps define this.

Opening\_Price = ws.Cells(i + 1, 3).Value

' When condition is met, resets Total Stock Volume to zero as a new ticket symbol will be observed on the next loop.

Total\_Stock\_Volume = 0

' For any instances where the condition is not met, the tickers are of the same type and therefore, the row's volume will need to be added onto the total stored in Total\_Volume.

Else

Total\_Stock\_Volume = Total\_Stock\_Volume + ws.Cells(i, 7).Value

End If

Next i

' Adding the functionalitiy to this script to return the greatest percentage increase and total volume, as well as the greatest percentage decrease from our summary table.

Greatest\_Percentage\_Increase = WorksheetFunction.max(ws.range("K:K")) ' Extracts the highest value from column K and assigns it to the variable 'Greatest\_Percentage\_Increase"

ws.Range("Q2").Value = Greatest\_Percentage\_Increase ' Displays the greatest percentage increase in a new summary table.

ws.Range("Q2").NumberFormat = "0.00%" ' Converts the number format of the displayed greatest percentage increase into a percentage.

Greatest\_Percentage\_Decrease = WorksheetFunction.min(ws.range("K:K")) ' Extracts the lowest value from column K and assigns it to the variable 'Greatest\_Percentage\_Decrease"

ws.Range("Q3").Value = Greatest\_Percentage\_Decrease ' Displays the greatest percentage increase in the new summary table.

ws.Range("Q3").NumberFormat = "0.00%" ' Converts the number format of the displayed greatest percentage decrease into a percentage.

Greatest\_Total\_Volume = WorksheetFunction.max(ws.range("L:L")) ' Extracts the highest value from column L and assigns it to the variable 'Greatest\_Total\_Volume"

ws.Range("Q4").Value = Greatest\_Total\_Volume ' Displays the greatest total volume in the new summary table.

' Now the ticker symbols linked to the values in the new summary table need to be extracted.

' Redefines the last row to be the last populated row in column 11.

Last\_Row = ws.Cells(Rows.Count, 11).End(xlUp).Row

' Begins a loop to search through each of the percentage change and total stock volume columns in our summary table.

For i = 2 To Last\_Row

For j = 11 To 12

If ws.Cells(i, j).Value = Greatest\_Percentage\_Increase Then ' Conditional: If the observed cell matches with the greatest percentage increase...

ws.Range("P2").Value = ws.Cells(i, 9).Value ' When condition is met, the corresponding ticker symbol is stored in the new summary table.

ElseIf ws.Cells(i,j).Value = Greatest\_Percentage\_Decrease Then ' Conditional: If the observed cell matches with the greatest percentage decrease...

ws.Range("P3").Value = ws.Cells(i, 9).Value ' When condition is met, the corresponding ticker symbol is stored in the new summary table.

ElseIf ws.Cells(i, j).Value = Greatest\_Total\_Volume Then ' Conditional: If the observed cell matches with the greatest total volume...

ws.Range("P4").Value = ws.Cells(i, 9).Value ' When condition is met, the corresponding ticker symbol is stored in the new summary table.

End If

Next j

Next i

' Autofits the worksheet's columns to make all data visable and clear.

ws.Cells.EntireColumn.Autofit

Next ws

End Sub