A screenshot of a spreadsheet

Description automatically generated

Code:

Sub AnalyzeStocks()

Dim ws As Worksheet

Dim lastRow As Long

Dim ticker As String

Dim dateValue As Date

Dim openingPrice As Double

Dim closingPrice As Double

Dim quarterlyChange As Double

Dim percentChange As Double

Dim volume As Double

Dim prevTicker As String

Dim outputRow As Long

Dim rng As Range

Dim condition As FormatCondition

' Open the workbook containing the data

Workbooks.Open "C:\Users\Classic\Downloads\\_Multiple\_year\_stock\_data\_.xlsx"

Set ws = ActiveWorkbook.ActiveSheet

' Find the last row with data

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

' Initialize output row

outputRow = 2

' Loop through each row of data

For i = 2 To lastRow

' Get data from current row

ticker = ws.Cells(i, 1).Value

dateValue = ws.Cells(i, 2).Value

openingPrice = ws.Cells(i, 3).Value

closingPrice = ws.Cells(i, 6).Value

volume = ws.Cells(i, 7).Value

' Check if the current ticker is different from the previous one

If ticker <> prevTicker Then

' Calculate quarterly change

quarterlyChange = closingPrice - openingPrice

' Calculate percentage change

If openingPrice <> 0 Then

percentChange = (closingPrice - openingPrice) / openingPrice \* 100

Else

percentChange = 0

End If

' Output ticker information

ws.Cells(outputRow, 9).Value = ticker

ws.Cells(outputRow, 10).Value = quarterlyChange

ws.Cells(outputRow, 11).Value = percentChange

ws.Cells(outputRow, 12).Value = volume

' Apply conditional formatting to quarterly change column (column J)

Set rng = ws.Cells(outputRow, 10)

With rng.FormatConditions.Add(Type:=xlCellValue, Operator:=xlGreater, Formula1:="0")

.Interior.Color = RGB(0, 255, 0) ' Green for positive change

End With

With rng.FormatConditions.Add(Type:=xlCellValue, Operator:=xlLess, Formula1:="0")

.Interior.Color = RGB(255, 0, 0) ' Red for negative change

End With

With rng.FormatConditions.Add(Type:=xlCellValue, Operator:=xlEqual, Formula1:="0")

.Interior.ColorIndex = xlColorIndexNone ' No fill for zero change

End With

outputRow = outputRow + 1

Else

' Output additional rows for the same ticker

ws.Cells(outputRow - 1, 12).Value = ws.Cells(outputRow - 1, 12).Value + volume

End If

' Update previous ticker

prevTicker = ticker

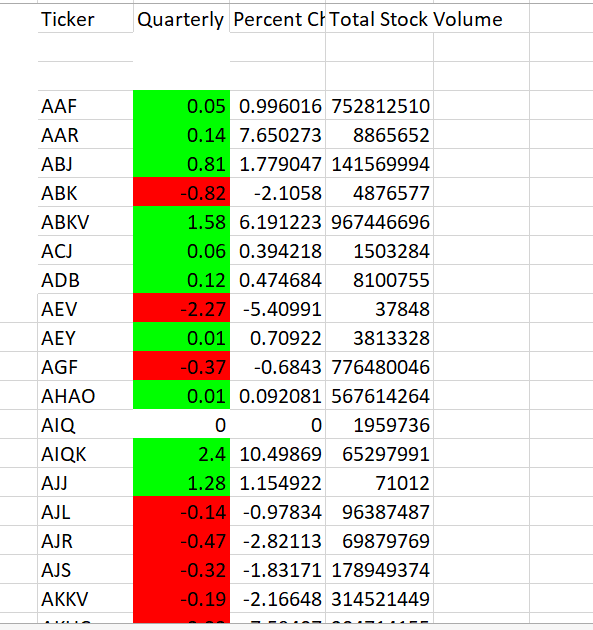
Next i

' Close the workbook

ActiveWorkbook.Close SaveChanges:=True

End Sub

output:



Code:

Sub FindGreatest()

Dim ws As Worksheet

Dim lastRow As Long

Dim maxIncrease As Double

Dim maxDecrease As Double

Dim maxIncreaseStock As String

Dim maxDecreaseStock As String

Dim outputRow As Long

' Initialize variables

maxIncrease = -999999999

maxDecrease = 999999999

outputRow = 1 ' Start output from row 1

' Loop through each worksheet in the workbook

For Each ws In ThisWorkbook.Worksheets

' Find the last row with data in column A

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

' Loop through each row and calculate greatest % increase and decrease for the current worksheet

For i = 2 To lastRow

' Calculate % increase

If ws.Cells(i, 5).Value <> 0 Then

increase = (ws.Cells(i, 5).Value - ws.Cells(i, 4).Value) / ws.Cells(i, 4).Value

Else

increase = 0

End If

' Check for greatest % increase

If increase > maxIncrease Then

maxIncrease = increase

maxIncreaseStock = ws.Cells(i, 1).Value

End If

' Check for greatest % decrease

If increase < maxDecrease Then

maxDecrease = increase

maxDecreaseStock = ws.Cells(i, 1).Value

End If

Next i

' Output the results for the current worksheet

With ThisWorkbook.Sheets(1) ' Output to the first sheet

' Output "Greatest % Increase"

.Cells(outputRow, 16).Value = "Greatest % Increase"

.Cells(outputRow + 1, 16).Value = maxIncreaseStock

.Cells(outputRow + 1, 17).Value = Format(maxIncrease \* 100, "0.00%")

' Output "Greatest % Decrease"

.Cells(outputRow + 3, 16).Value = "Greatest % Decrease"

.Cells(outputRow + 4, 16).Value = maxDecreaseStock

.Cells(outputRow + 4, 17).Value = Format(maxDecrease \* 100, "0.00%")

' Increment output row for the next worksheet

outputRow = outputRow + 6 ' Move to the next section

End With

' Reset variables for the next worksheet

maxIncrease = -999999999

maxDecrease = 999999999

Next ws

End Sub

Output:

A screenshot of a spreadsheet

Description automatically generated

A screenshot of a data

Description automatically generated

Sub ApplyConditionalFormatting()

Dim wb As Workbook

Dim ws As Worksheet

Dim rng As Range

Dim cell As Range

' Check if the workbook is already open

On Error Resume Next

Set wb = Workbooks("\_alphabetical\_testing\_.xlsx")

On Error GoTo 0

' If the workbook is not already open, open it

If wb Is Nothing Then

Set wb = Workbooks.Open("C:\Users\Classic\Downloads\\_alphabetical\_testing\_.xlsx")

End If

' Loop through each worksheet in the workbook

For Each ws In wb.Worksheets

' Define the range of data (adjust as needed)

Set rng = ws.UsedRange

' Loop through each cell in the range (excluding headers)

For Each cell In rng.Offset(1).Cells

' Check if the cell contains a numeric value

If IsNumeric(cell.Value) Then

' Check if the current cell is not in the first column

If cell.Column > 1 Then

' Check if the previous cell in the same row contains a numeric value

If IsNumeric(cell.Offset(0, -1).Value) Then

' Calculate the change

Dim change As Double

change = cell.Value - cell.Offset(0, -1).Value

' Apply conditional formatting based on the change

If change > 0 Then

cell.Interior.Color = RGB(0, 255, 0) ' Green for positive change

ElseIf change < 0 Then

cell.Interior.Color = RGB(255, 0, 0) ' Red for negative change

End If

End If

End If

End If

Next cell

Next ws

' Save and close the workbook

wb.Close SaveChanges:=True

End Sub  
OutPut:

