2.2.4 Get DQ's Yearly Return for 2018

Steve wants to know how DQ performed in 2018. One way to measure this is to calculate the yearly return for DQ. The **yearly return** is the percentage increase or decrease in price from the beginning of the year to the end of the year. In other words, if you invested in DQ at the beginning of the year and never sold, the yearly return is how much your investment grew or shrunk by the end of the year.

Let's calculate the yearly return of DQ's stock. To do this calculation, we need DQ's first closing price and last closing price.

To find the first closing price of DQ's data, we'll need to do the following:

1. Loop through all the rows.
2. Check if the current row is the first row of DQ's data.
3. If so, set the starting price to the closing price in the current row.

We already have a loop going through all the rows, so we don't need to make another loop. This means Step 1 is done. But how do we do Step 2? We need to check for two conditions:

1. If if the ticker in the current row is DQ

2. If the ticker in the previous row is *not* DQ

We can check both conditions at once using **logical operators**.

IMPORTANT

> **Logical operators,** also called Boolean operators, link more than one condition together, which allows for more complicated conditional arguments. The logical operators in VBA are `And`, `Or`, and `Not`. That is:
>
> - condition1 `And` condition2 will only evaluate as true if **both** condition1 and condition2 are true.
> - condition1 `Or` condition2 will evaluate as true if **either** condition1 or condition2 are True.
> - `Not` condition will give the **opposite** value of whatever condition is.
>
> In addition, there is a "not equal to" **comparison operator** that checks whether two values are not equal to each other. In VBA, the "not equal to" operator is two angle brackets: `<>`

The condition to check if the current row's ticker is DQ is

```
Cells(i, 1).Value = "DQ"
```

And the condition to check if the previous row's ticker is not DQ is

```
Cells(i - 1, 1).Value <> "DQ"
```

Since we want both conditions to be true, we'll join them together with the `And` operator in our if statement. This new code will be inserted in the `for` loop directly after the `if` block that computes totalVolume.

```
If Cells(i, 1).Value = "DQ" And Cells(i - 1, 1).Value <> "DQ" Then

    'set starting price

End If
```

Before the `for` loop, create a variable for starting price. Since the prices have decimal values, we'll use the `Double` data type.

```
Dim startingPrice As Double
```

Price data is in the sixth column, so the value in `Cells(i, 6)` has the starting price.

```
startingPrice = Cells(i, 6).Value
```

Put together, the code should look like this:

```
If Cells(i, 1).Value = "DQ" And Cells(i - 1, 1).Value <> "DQ" Then

    startingPrice = Cells(i, 6).Value

End If
```

To find the ending price, follow the same code pattern.

REWIND

That's right: we're reusing a design pattern! We'll apply the pattern used to find the starting price to our process for finding the ending price by following these steps:

1. Initialize a variable to store ending price as a double data type.
2. Check if the current row is the last row of DQ's data:
    - Check that the current row's ticker is DQ.
    - Check that the next row's ticker is not DQ.
3. If so, set the ending price to the current row's closing price.

Give it a try. Your code should look like this:

```
    Dim startingPrice As Double
    Dim endingPrice As Double
```

```
If Cells(i, 1).Value = "DQ" And Cells(i + 1, 1).Value <> "DQ" Then

    endingPrice = Cells(i, 6).Value

End If
```

With the starting and ending prices stored, our `for` loop is finished. We can now add a line to our output to show the yearly return for DQ:

```
    Worksheets("DQ Analysis").Activate
    Cells(4, 1).Value = 2018
    Cells(4, 2).Value = totalVolume
    Cells(4, 3).Value = endingPrice / startingPrice - 1
```

The finished `DQAnalysis` macro should look like this:

```
Sub DQAnalysis()
    Worksheets("DQ Analysis").Activate

    Range("A1").Value = "DAQO (Ticker: DQ)"

    'Create a header row
    Cells(3, 1).Value = "Year"
    Cells(3, 2).Value = "Total Daily Volume"
    Cells(3, 3).Value = "Return"
    Worksheets("2018").Activate

    'set initial volume to zero
    totalVolume = 0

    Dim startingPrice As Double
    Dim endingPrice As Double

    'Establish the number of rows to loop over
    rowStart = 2
    rowEnd = Cells(Rows.Count, "A").End(xlUp).Row

    'loop over all the rows
    For i = rowStart To rowEnd

        If Cells(i, 1).Value = "DQ" Then

            'increase totalVolume by the value in the current row
            totalVolume = totalVolume + Cells(i, 8).Value

        End If

        If Cells(i - 1, 1).Value <> "DQ" And Cells(i, 1).Value = "DQ" Then

            startingPrice = Cells(i, 6).Value

        End If

        If Cells(i + 1, 1).Value <> "DQ" And Cells(i, 1).Value = "DQ" Then

            endingPrice = Cells(i, 6).Value

        End If
```

```
    Next i

    Worksheets("DQ Analysis").Activate
    Cells(4, 1).Value = 2018
    Cells(4, 2).Value = totalVolume
    Cells(4, 3).Value = (endingPrice / startingPrice) - 1


End Sub
```

Whew! That's a long piece of code. Let's walk through it again and remind ourselves what it's doing.

Save your code and run it to make sure everything works correctly.

FINDING

Daqo dropped over 63% in 2018—yikes! Steve will definitely want to offer some better stocks to his parents.

ADD/COMMIT/PUSH

Once everything is working correctly, save `green_stocks.xlsm` and push the changes to the "stocks-analysis" repository in GitHub.