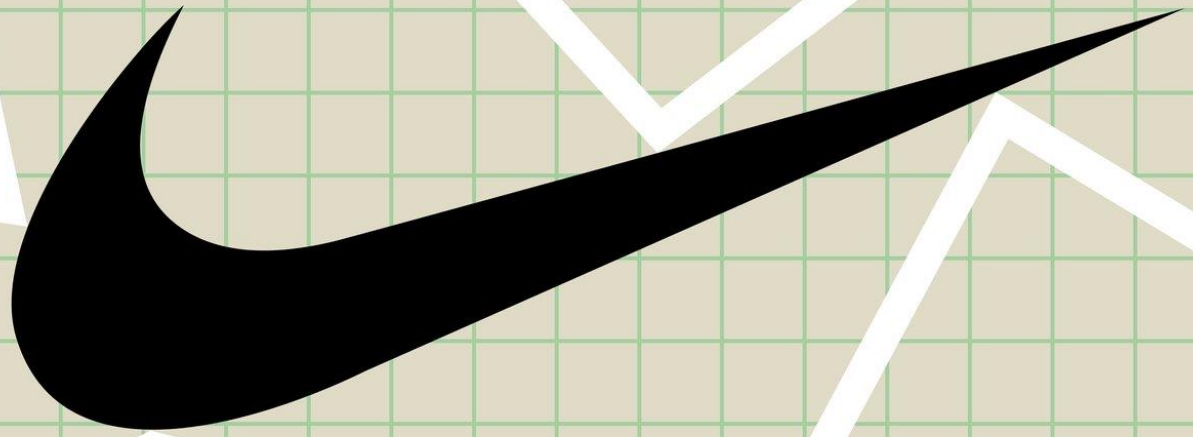
Nike Stock Investment

MGS617 Final Project



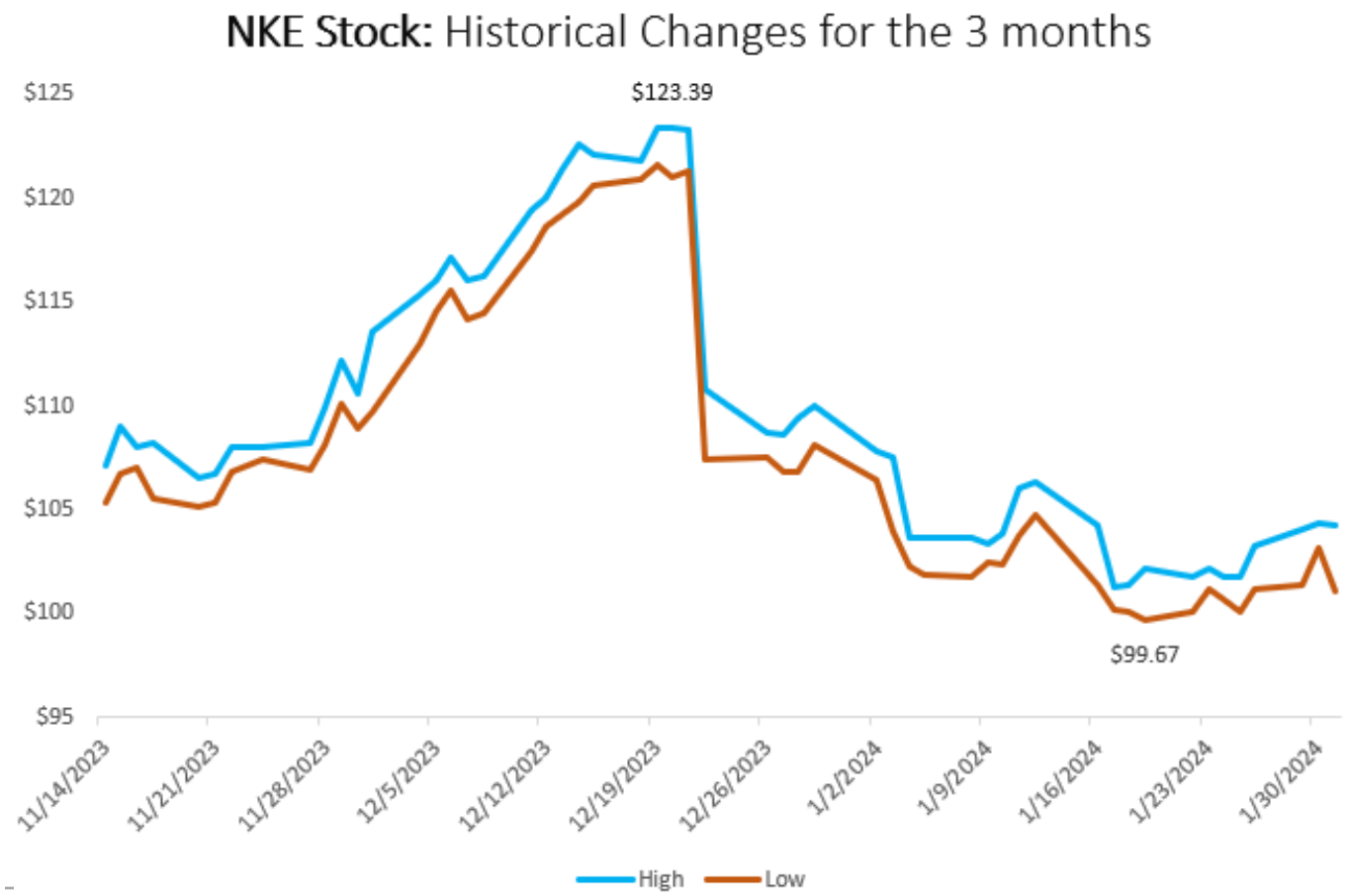
Group 2: Aylin Umarova, Dylan Rendina, Madison Whyte Ainsley, Fatima Khan

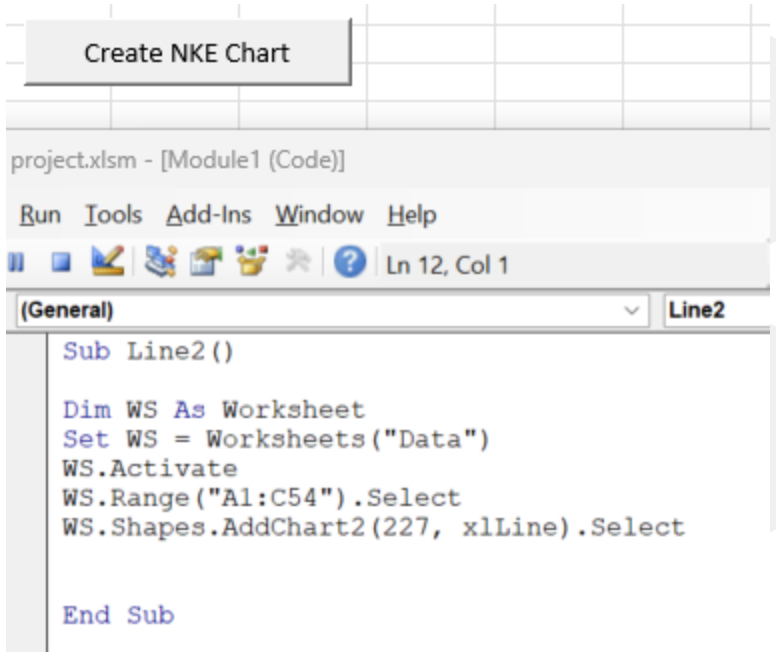
Summary/Concepts Covered

- Assess and analyze the performance of the NKE stock
- Utilize the data to capture changes in the stock based on microeconomics
- Utilize the data to graphically show the optimal time for purchasing/investing in Nike Stocks.
- Concepts used: VBA, Range of Excel Worksheets, Sub Procedures, Charts(Line Chart),
- Assess and analyze what date the stock price was the lowest using VBA code.

Key Outcome 1:

Capturing Historical Changes in NIKE Stock Price





Key Outcome 2:
Assessing ideal timing for purchasing stocks.

The key outcome was to "Utilize data visualization to analyze and assess the ideal timing for purchasing stocks."

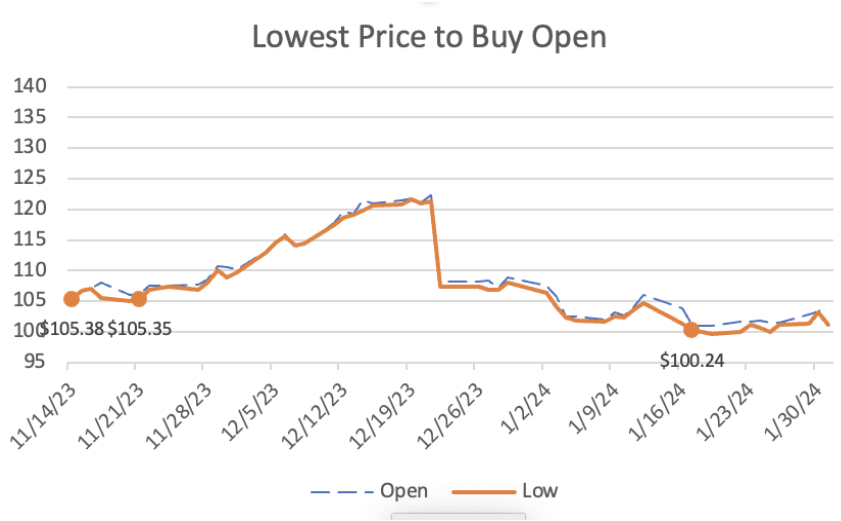
Through VBA, a graph was created to display the Nike Stock from dates 11/14/23 through 1/30/24. In this graph, the user will be able to distinguish what the stock looked like at open and at the lowest prices. Markers were used to signify the optimal times the user should have invested.

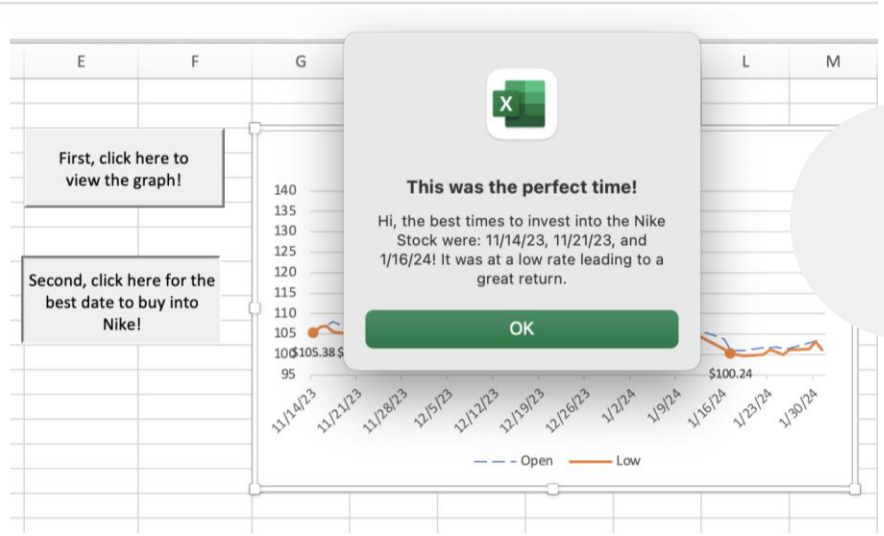
With using VBA, the user was also notified with a message box to explain the markers in the graph.

	Date	Open	Low
2	11/14/23	105.559998	105.379997
3	11/15/23	106.82	106.720001
4	11/16/23	107.099998	107
5	11/17/23	108.059998	105.519997
6	11/20/23	106	105.099998
7	11/21/23	106.169998	105.349998
8	11/22/23	107.5	106.82
9	11/24/23	107.5	107.400002
10	11/27/23	107.660004	106.900002
11	11/28/23	108.57	108.150002
12	11/29/23	110.800003	110.150002
13	11/30/23	110.529999	108.959999
14	12/1/23	110.330002	109.75

First, click here to view the graph!

Second, click here for the best date to buy into Nike!





Key Outcome 2 Cont:

Assessing ideal timing for purchasing stocks. (VBA Code)

```
(General)
Sub PurchaseStock()

Dim WS As Worksheet
Set WS = Worksheets("BuyStock")

WS.Activate
WS.Range("A1:C54").Select
WS.Shapes.AddChart2(227, xlLine).Select

ActiveChart.Axes(xlValue).Select
    ActiveChart.Axes(xlValue).MinimumScale = 95
    ActiveChart.Axes(xlValue).MaximumScale = 140
    ActiveChart.ChartTitle.Select
ActiveChart.ChartTitle.Text = "Lowest Price to Buy Open"

With Selection.Format.TextFrame2.TextRange.Characters(1, 6).Font
        .BaselineOffset = 0
        .Bold = msoFalse
        .NameComplexScript = "+mn-cs"
        .NameFarEast = "+mn-ea"
        .Fill.Visible = msoTrue
        .Fill.ForeColor.RGB = RGB(89, 89, 89)
        .Fill.Transparency = 0
        .Fill.Solid
        .Size = 14
        .Italic = msoFalse
        .Kerning = 12
        .Name = "+mn-lt"
        .UnderlineStyle = msoNoUnderline
        .Spacing = 0
        .Strike = msoNoStrike
    End With
    With Selection.Format.TextFrame2.TextRange.Characters(7, 18).Font
        .BaselineOffset = 0
        .Bold = msoFalse
        .NameComplexScript = "+mn-cs"
```

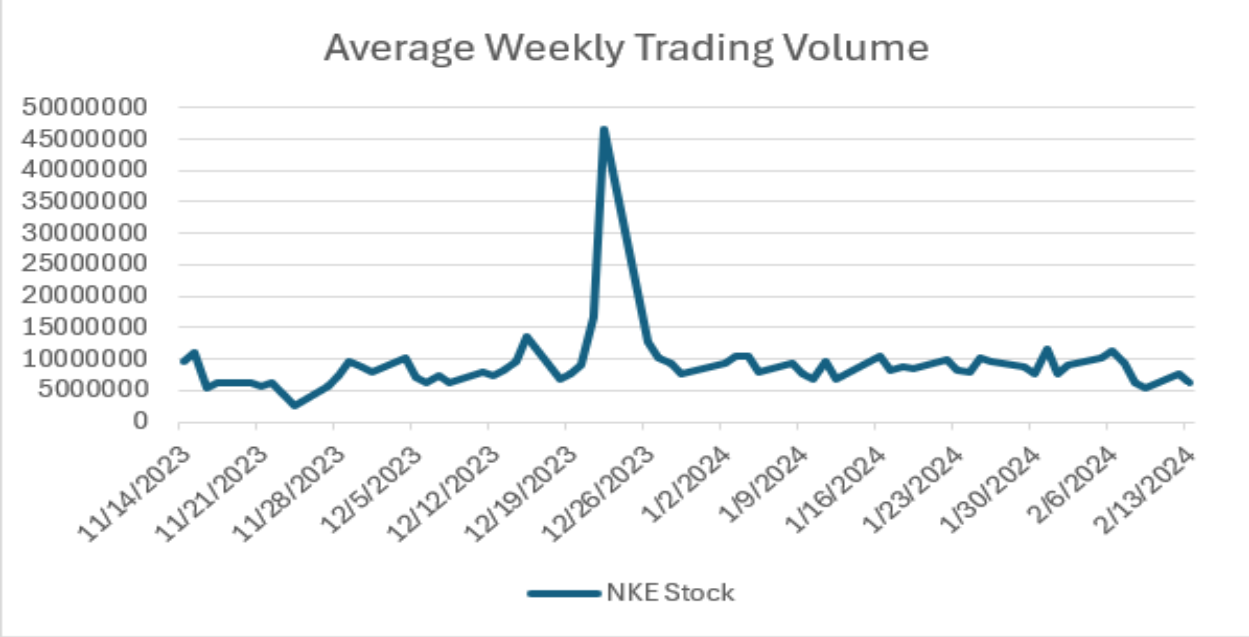
```
(General)
        .NameFarEast = "+mn-ea"
        .Fill.Visible = msoTrue
        .Fill.ForeColor.RGB = RGB(89, 89, 89)
        .Fill.Transparency = 0
        .Fill.Solid
        .Size = 14
        .Italic = msoFalse
        .Kerning = 12
        .Name = "+mn-lt"
        .UnderlineStyle = msoNoUnderline
        .Spacing = 0
        .Strike = msoNoStrike
    End With
    ActiveChart.FullSeriesCollection(1).Select
    With Selection.Format.Line
        .Visible = msoTrue
        .DashStyle = msoLineLongDash
    End With
    With Selection.Format.Line
        .Visible = msoTrue
        .Weight = 1
    End With
    ActiveChart.FullSeriesCollection(2).Select
    With Selection.Format.Line
        .Visible = msoTrue
        .Weight = 1.75
    End With
    ActiveChart.FullSeriesCollection(1).Select
    ActiveChart.FullSeriesCollection(2).Select
    ActiveChart.FullSeriesCollection(2).Points(3).Select
    ActiveChart.FullSeriesCollection(1).Select
    ActiveChart.FullSeriesCollection(2).Select
    ActiveChart.FullSeriesCollection(2).Points(2).Select
    ActiveChart.FullSeriesCollection(2).Points(1).Select
    With Selection
        .MarkerStyle = -4142
```

```
(General)
    ActiveChart.FullSeriesCollection(2).Points(2).Select
    ActiveChart.FullSeriesCollection(2).Points(1).Select
    With Selection
        .MarkerStyle = -4142
        .MarkerSize = 5
    End With
    Selection.MarkerStyle = 8
    ActiveChart.FullSeriesCollection(2).Points(6).Select
    With Selection
        .MarkerStyle = -4142
        .MarkerSize = 5
    End With
    Selection.MarkerStyle = 8
    ActiveChart.FullSeriesCollection(2).Points(43).Select
    With Selection
        .MarkerStyle = -4142
        .MarkerSize = 5
    End With
    Selection.MarkerStyle = 8
    ActiveChart.FullSeriesCollection(2).Points(43).ApplyDataLabels
    ActiveChart.FullSeriesCollection(2).DataLabels.Select
    Selection.Position = xlLabelPositionBelow
    Selection.Format.TextFrame2.TextRange.Font.Size = 8
    Selection.NumberFormat = "$#,##0.00"
    Selection.NumberFormat = "$#,##0.00;[Red]$#,##0.00"
    ActiveChart.FullSeriesCollection(2).Select
    ActiveChart.FullSeriesCollection(2).Points(6).Select
    ActiveChart.FullSeriesCollection(2).Points(6).ApplyDataLabels
    ActiveChart.FullSeriesCollection(2).Points(1).Select
    ActiveChart.FullSeriesCollection(2).Points(1).ApplyDataLabels
End Sub
Sub ShowMessageBox()
MsgBox "Hi, the best times to invest into the Nike Stock were: 11/14/23, 11/
End Sub
```

Key Outcome 3 :

Conduct analysis to determine what the average weekly volume was throughout the period

The key outcome was to analyze the average trading volume of NKE stock during the approximately 3-month period. This was achieved via the creation of a line chart which enables the user to visually track how the average volume changed per week. Initially, we considered utilizing other methods to see the average daily volume; however, we thought it was more practical and useful to see a week-by-week snapshot.



Key Outcome 3 Cont. – Conduct analysis to determine what the average weekly volume was throughout the period (VBA Code)

- There are two sub-procedures being used here:
- 1.) The first sub-procedure ('AverageVolume' is straightforward – it takes all of the cells with numeric values in column G ('Volume') and calculates the average. It then displays (via the MsgBox function) a message to the user informing them what the average volume was.
- 2.) The second sub-procedure ('Chart') is what allows the user to generate a line chart and see how the volume changed throughout the period.

(General)

```
Sub AverageVolume()

    Dim Rng As Range
    Dim Avg As Double
    Set Rng = Range("G2:G63")

    Avg = WorksheetFunction.Average(Rng)

    MsgBox "The average daily volume was: " & Avg

End Sub
```

```
Sub Chart()
    Dim Chart1 As Chart
    Dim DataRange1 As Range
    Dim DataRange2 As Range
    Dim DataRange3 As Range

    Set DataRange1 = ActiveSheet.Range("G2:G63")
    Set DataRange2 = ActiveSheet.Range("A2:A63")
    Set DataRange3 = Union(DataRange1, DataRange2)
    Set Chart1 = ActiveSheet.Shapes.AddChart2.Chart

    Chart1.SetSourceData Source:=DataRange3
    Chart1.ChartType = xlLine
    Chart1.SetElement msoElementLegendBottom
    Chart1.ChartTitle.Text = "Average Weekly Trading Volume"

    Chart1.FullSeriesCollection(1).Name = "=""NKE Stock"""

End Sub
```

Key Outcome 4:

Conduct analysis to conclude what date the stock trading at its lowest based on the data

(General)

```
Sub FindLowestStockPrice()
    Dim ws As Worksheet
    Dim lastRow As Long
    Dim minPrice As Double
    Dim minDate As Date

    Set ws = ActiveSheet

    ' Find the last row with data in column A
    lastRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).Row

    ' Initialize minimum price with the first value in the "Low" column
    minPrice = ws.Cells(2, "B").Value '  "Low" is in column B
    minDate = ws.Cells(2, "A").Value ' Corresponding date

    ' Loop through the data to find the minimum price and corresponding date
    For i = 2 To lastRow
        If ws.Cells(i, "B").Value < minPrice Then
            minPrice = ws.Cells(i, "B").Value
            minDate = ws.Cells(i, "A").Value ' dates are in column A
        End If
    Next i

    ' Display the results
    MsgBox "The lowest stock price was $" & minPrice & " on " & minDate, vbInformation, "Lowest Stock Price"
End Sub
```