First steps with Office Scripts: Create and format a column chart

Owen Price
flexyourdata.com, @flexyourdata

What are Office Scripts?

"Office Scripts in Excel let you automate your day-to-day tasks. Use the Action Recorder to turn manual steps into reusable scripts. Edit those scripts or create new ones with the Code Editor. Let others in the workbook run these scripts with a single button. Then, share them with coworkers so everyone can improve their workflow."

https://learn.microsoft.com/en-us/office/dev/scripts/overview/excel?view=office-scripts?wt.mc_id=MVP_310565

Before today, I'd written a total of about 3 lines of TypeScript.

I set out to spend 1-2 hours writing my first code in Office Scripts.

My goal was to write a script that would create a column chart with a reference line and change some of the formatting automatically.

Owen Price
flexyourdata.com, @flexyourdata

I started with some data imported from the Adventure Works DW 2019

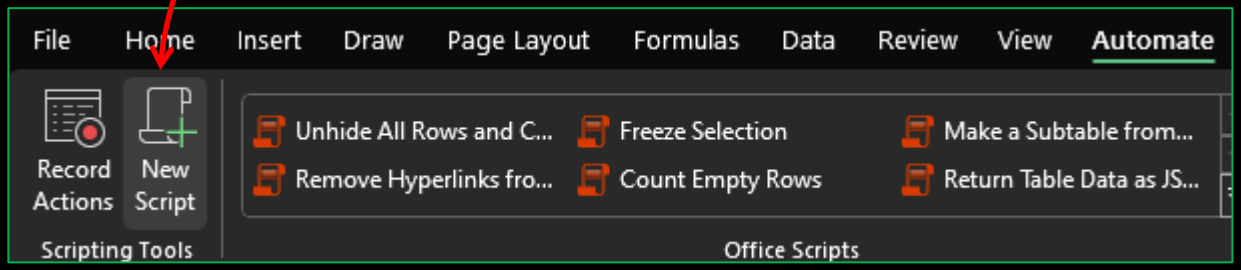
	A	B	C
1	Year	Month	SalesAmount
2	2010	12	43,421
3	2011	1	469,824
4	2011	2	466,335
5	2011	3	485,199
6	2011	4	502,074
7	2011	5	561,681
8	2011	6	737,840
9	2011	7	596,747
10	2011	8	614,558
11	2011	9	603,083
12	2011	10	708,208
13	2011	11	660,546
14	2011	12	669,432
15	2012	1	495,364
16	2012	2	506,994
17	2012	3	373,483
18	2012	4	400,336
19	2012	5	358,878
20	2012	6	555,160
21	2012	7	444,558
22	2012	8	523,917
23	2012	9	486,177
24	2012	10	535,159
25	2012	11	537,956
26	2012	12	624,502

The query shows sum of SalesAmount by Year and Month

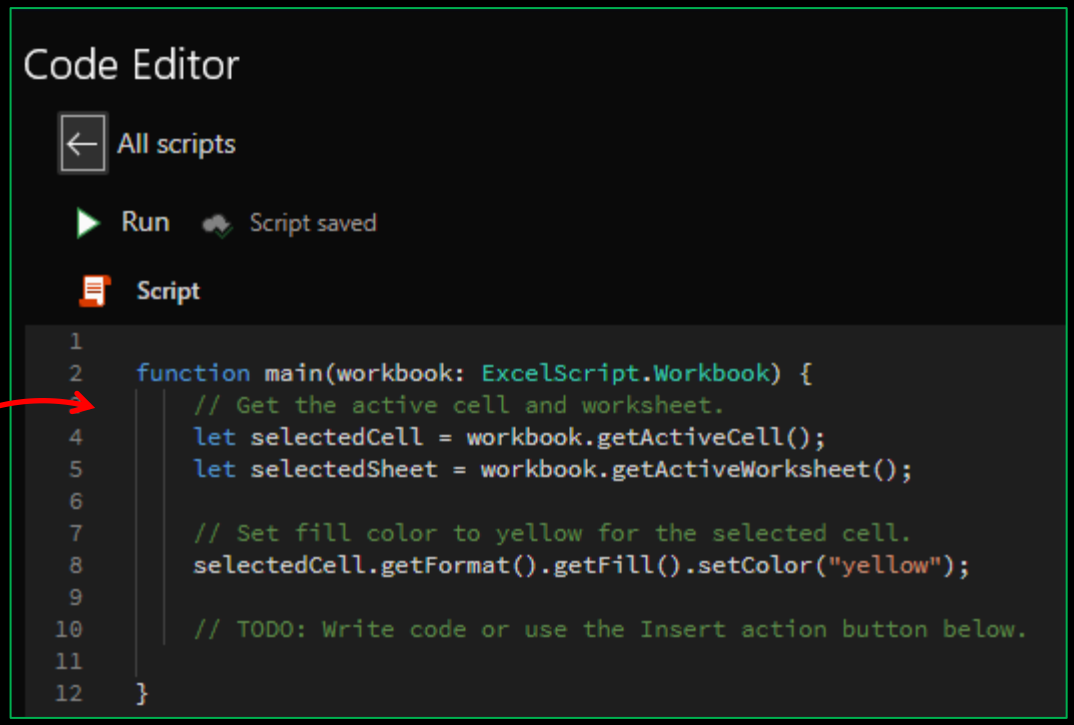
Owen Price
flexyourdata.com, @flexyourdata

To create a new script, use the Automate Tab

To create an empty script, hit the 'New Script' button



The Code Editor opens and is populated with some example code



Owen Price
flexyourdata.com, @flexyourdata

In TypeScript, we assign variables using *let*

Get the active sheet

```
function main(workbook: ExcelScript.Workbook) {

  let selectedSheet = workbook.getActiveWorksheet();

  /* Get the table,
  the rows containing data,
  the value column,
  and the category columns */
  let dataTable = selectedSheet.getTables()[0];
  let dataRange = dataTable.getRangeBetweenHeaderAndTotal()
  let valueRange = dataRange.getColumn(2)
  let categoryRange = dataRange.getResizedRange(0, -1)

  const referenceValue = 1500000
```

Get the first (and only) table from the sheet

Get all the data rows from the table

The values for the chart are in the 3rd column (index 2)

Create a constant to use for the reference line

We want the first two columns as category labels. getResizedRange(0, -1) removes zero rows and the right-most column

Owen Price
flexyourdata.com, @flexyourdata

...remove any previous charts, create the new chart

To access a range, we use the .getRange method

TypeScript provides a forEach method to apply a callback (lambda) function over a collection

```
14  const referenceValue = 1500000
15
16  // clear charts from sheet (while testing)
17  selectedSheet.getCharts().forEach(c => c.delete())
18  selectedSheet.getRange("E1:E39").clear()
19
20  /* Create a new clustered column chart using
21  the valueRange as the data */
22  let chart = selectedSheet.addChart(
23    ExcelScript.ChartType.columnClustered,
24    valueRange
25  )
```

The .addChart method expects two required arguments – the type of chart you want to create, and the sourceData for the chart's values

Owen Price
flexyourdata.com, @flexyourdata

...make some visual changes to the chart axes

Store the category axis in a variable

```
27  /* Assign the category data to the category axis,
28  Make the category axis *not* multi-level
29  Use center alignment and horizontal text for the axis labels */
30  let categoryAxis = chart.getAxes().getCategoryAxis()
31  categoryAxis.setCategoryNames(categoryRange)
32  categoryAxis.setMultiLevel(false)
33  categoryAxis.setAlignment(
34    ExcelScript.ChartTickLabelAlignment.center
35    )
36  categoryAxis.setTextOrientation(0)
37
38  // Remove the horizontal major gridlines
39  chart.getAxes()
40        .getValueAxis()
          .getMajorGridlines()
          .setVisible(false)
```

Chained methods can be wrapped onto separate lines

This code assigns the first two columns (categoryRange) as the category names of the axis, sets it to *not* a multi-level axis, sets the label alignment and text orientation.

Owen Price
flexyourdata.com, @flexyourdata

...make some visual changes to the chart axes

Collections can be indexed with square brackets

```
44  // Give the chart a sensible title
45  chart.getTitle().setText("Sales by Month against Target")
46
47  // Reduce the distance between the columns
48  let columnSeries = chart.getSeries()[0]
49  columnSeries.setGapWidth(12)
50
51  // Set all bars to silver, except those that are taller than
      the reference line
52  columnSeries.getFormat().getFill().setSolidColor("silver")
53  columnSeries.getPoints().forEach(
54    p => {if (p.getValue() >= referenceValue) {p.getFormat().
        getFill().setSolidColor("darkBlue")}}
    )
```

Set the column color for the series to “silver”

Iterate through the series points (the columns) and apply the function inside the forEach. If the point is >= the ref. value, make the column darkBlue

Owen Price
flexyourdata.com, @flexyourdata

...make some visual changes to the chart axes

Updating the worksheet is simple

```
57    // Create the data for the reference line
58    let referenceRange = selectedSheet.getRange("E2:E39")
59    referenceRange.setValue(referenceValue)
60
61    // Add the line as a new series
62    const referenceSeries = chart.addChartSeries("Target")
63    referenceSeries.setValues(referenceRange)
64    referenceSeries.setChartType(ExcelScript.ChartType.line)
65    referenceSeries.getFormat().getLine().setColor("darkGrey")
66
67    // Size and position the chart on the worksheet
68    chart.setHeight(500)
69    chart.setWidth(800)
70    chart.setPosition("E1")
71  }
72
```

Add a new series as a line, referencing the data created in column E

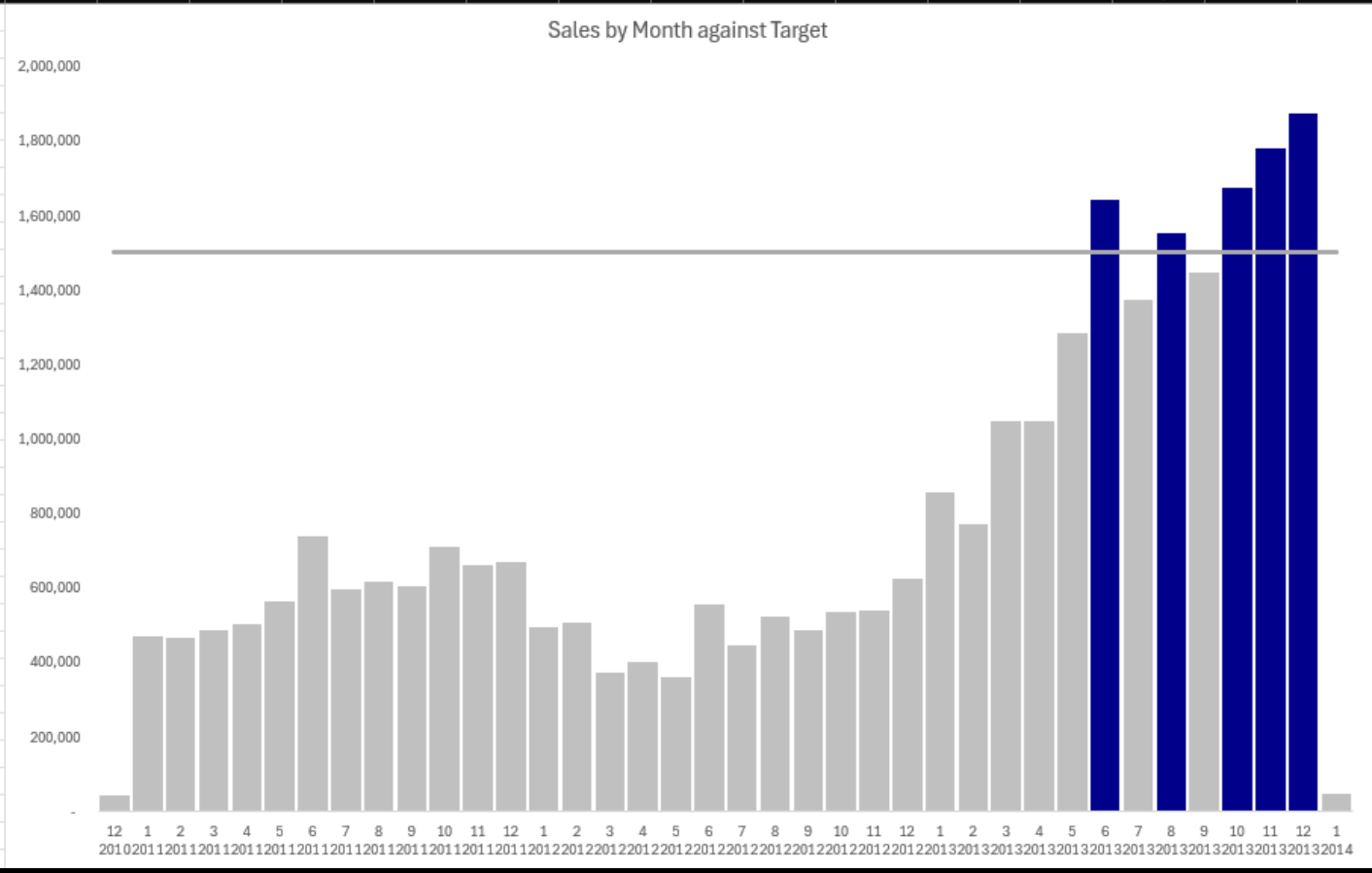
Finally, set the height, width and position of the chart

Owen Price
flexyourdata.com, @flexyourdata

And there it is!

Still some work to do on the category labels… but that's for next time!

Year	Month	SalesAmount
2010	12	43,421
2011	1	469,824
2011	2	466,335
2011	3	485,199
2011	4	502,074
2011	5	561,681
2011	6	737,840
2011	7	596,747
2011	8	614,558
2011	9	603,083
2011	10	708,208
2011	11	660,546
2011	12	669,432
2012	1	495,364
2012	2	506,994
2012	3	373,483
2012	4	400,336
2012	5	358,878
2012	6	555,160
2012	7	444,558
2012	8	523,917
2012	9	486,177
2012	10	535,159
2012	11	537,956
2012	12	624,502
2013	1	857,690
2013	2	771,349
2013	3	1,049,907
2013	4	1,046,023
2013	5	1,284,593
2013	6	1,643,178



Owen Price
flexyourdata.com, @flexyourdata

Takeaways:

1. Office Scripts uses TypeScript, a super-set of JavaScript
2. Variables are assigned with the `let` keyword
3. Access a range using `worksheet.getRange("A1")`
4. Most properties are managed with functions – `getX` and `setX`. This is in contrast to VBA's `property = value` syntax