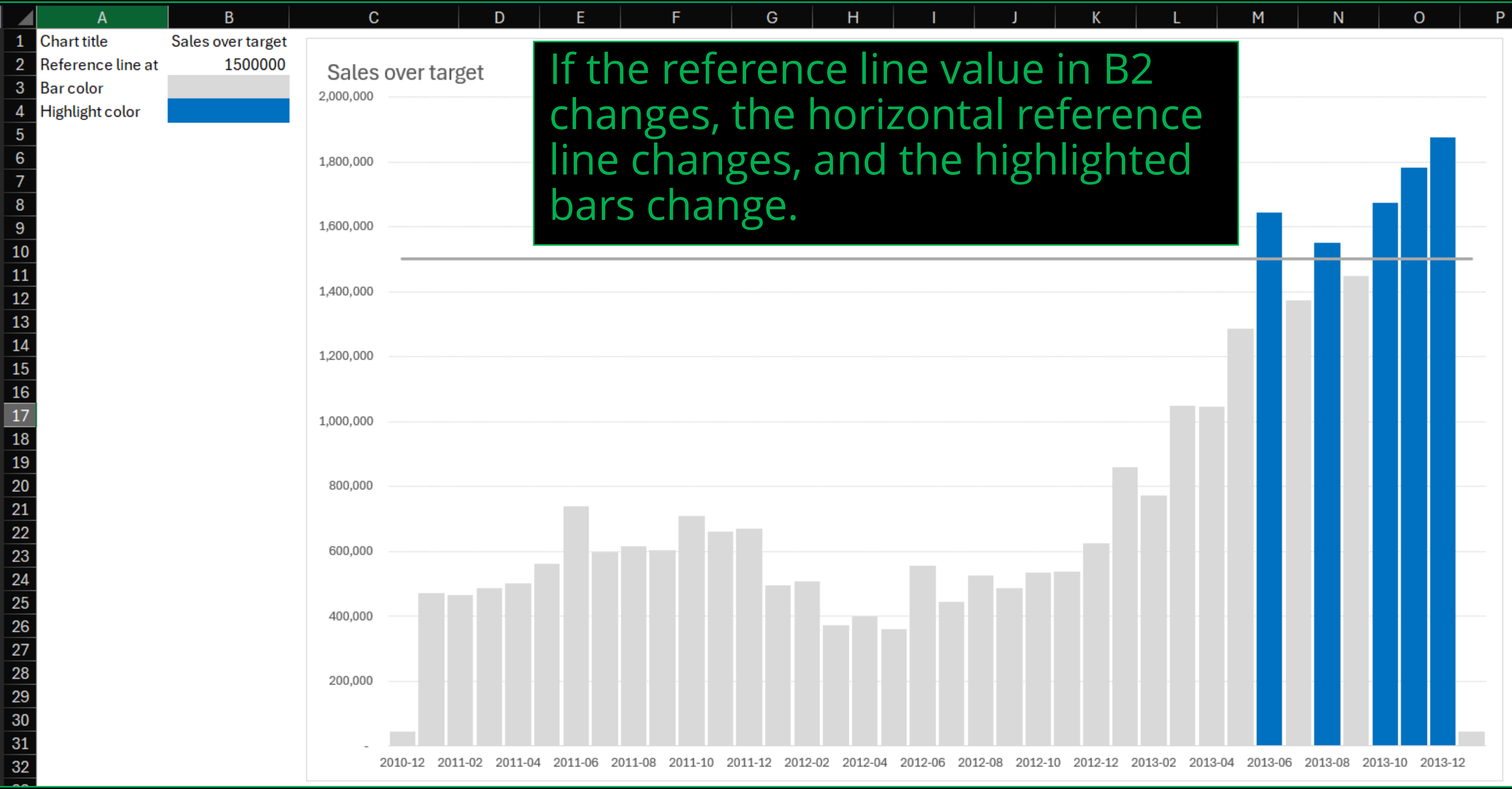
Creating charts with Office Scripts: Beware of context sync issues!

Owen Price
flexyourdata.com, @flexyourdata

This chart is created and configured by Office Scripts using the named ranges in cells B1:B4



A	B
Chart title	Sales over target
Reference line at	1500000
Bar color	⬜
Highlight color	🟦

If the reference line value in B2 changes, the horizontal reference line changes, and the highlighted bars change.

Owen Price
flexyourdata.com, @flexyourdata

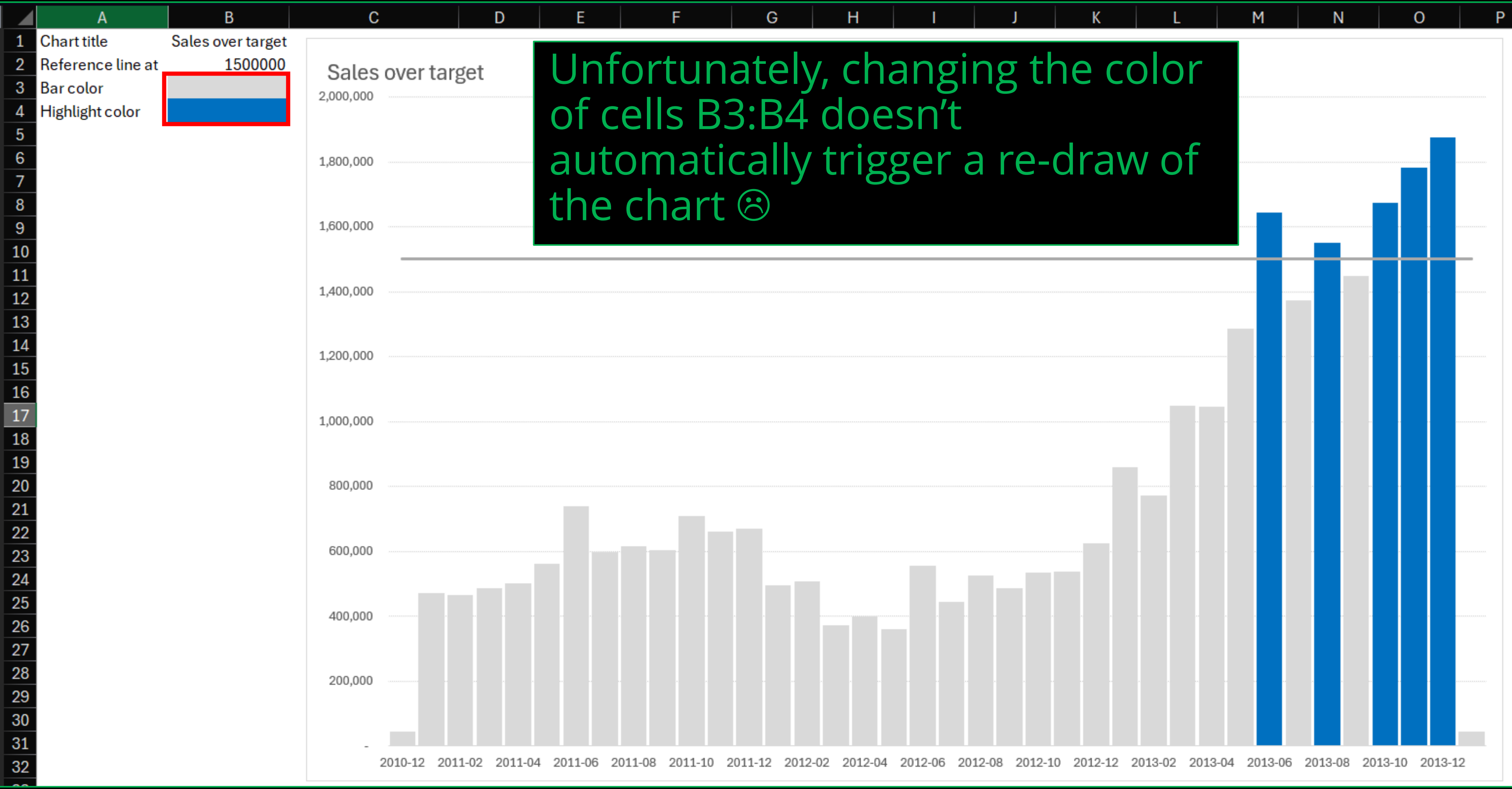
Originally, the code to create the series to highlight specific bars looked like this

```
// (code to create and setup other chart elements)

  // Add the line as a new series
  let s = chart.addChartSeries("Highlights")
  // Set the values to the range 3 cols to the right of the 'valueRange'
  s.setValues(valueRange.getOffsetRange(0, 3))
 // Set the series type to "Clustered Column"
  s.setChartType(ExcelScript.ChartType.columnClustered)
 // Set the new series to overlap the other series 100%
  s.setOverlap(100)
 // Set the fill color of the new series to dark blue
  s.getFormat().getFill().setSolidColor("darkBlue")

// (code to position chart on worksheet)
}
```

But I wanted to control the bar colors with the fill color of some named cells



	A	B
1	Chart title	Sales over target
2	Reference line at	1500000
3	Bar color	⬜
4	Highlight color	🟦

Unfortunately, changing the color of cells B3:B4 doesn't automatically trigger a re-draw of the chart ☹

Owen Price
flexyourdata.com, @flexyourdata

Since I wanted to control several series in this way, I created a helper function

```
function getFillOfNamedRange(
  name: string,
  workbook: ExcelScript.Workbook): string {

  // Return the fill color of the named range
  return workbook.getNamedItem(name)
              .getRange()
              .getFormat()
              .getFill()
              .getColor()
}
```

This function returns the HTML color code representing the background color of a named range.

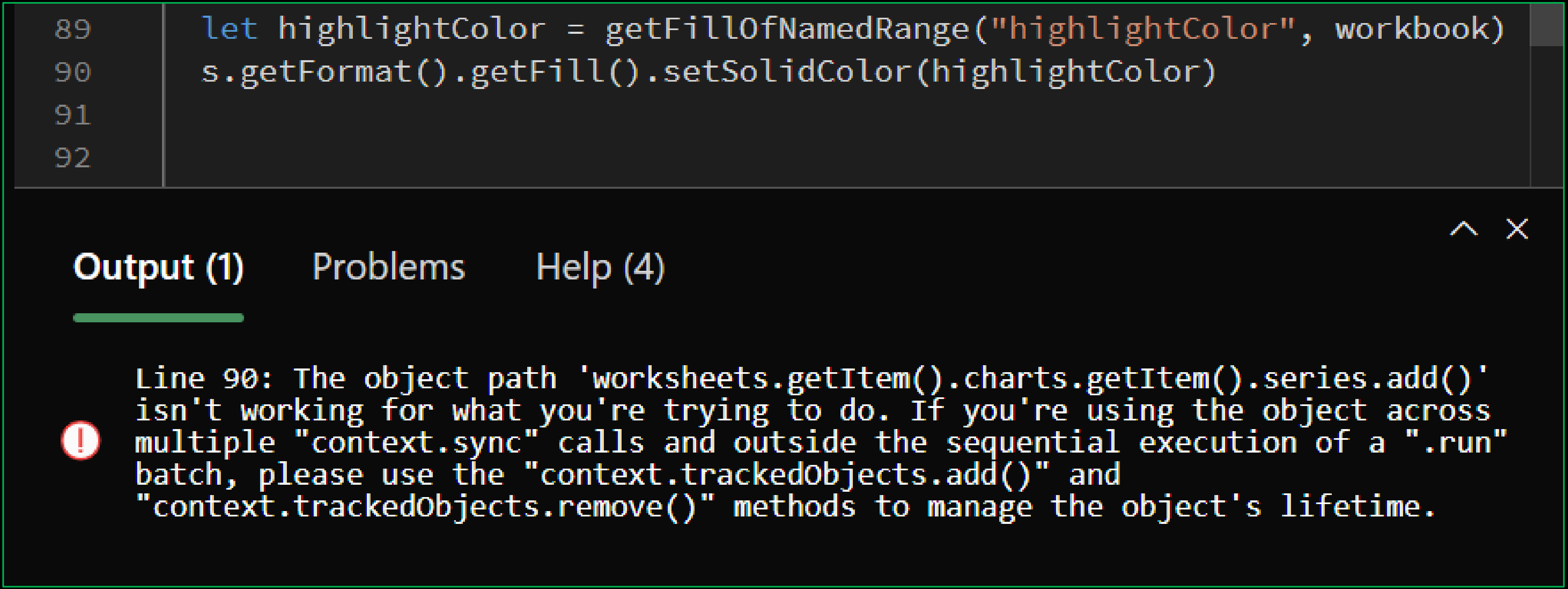
And changed the code to this

```
// (code to create and setup other chart elements)

 // Add the line as a new series
 let s = chart.addChartSeries("Highlights")
 // Set the values to the range 3 cols to the right of the ‘valueRange’
 s.setValues(valueRange.getOffsetRange(0, 3))
 // Set the series type to “Clustered Column”
 s.setChartType(ExcelScript.ChartType.columnClustered)
 // Set the new series to overlap the other series 100%
 s.setOverlap(100)
 // Get the background color from the named range
 let highlightColor = getFillOfNamedRange("highlightColor", workbook)
// Set the bar color of the new series
 s.getFormat().getFill().setSolidColor(highlightColor)

// (code to position chart on worksheet)
```

But this produces an error!



```
89    let highlightColor = getFillOfNamedRange("highlightColor", workbook)
90    s.getFormat().getFill().setSolidColor(highlightColor)
91
92
```

```
Line 90: The object path 'worksheets.getItem().charts.getItem().series.add()'
isn't working for what you're trying to do. If you're using the object across
multiple "context.sync" calls and outside the sequential execution of a ".run"
batch, please use the "context.trackedObjects.add()" and
"context.trackedObjects.remove()" methods to manage the object's lifetime.
```

This cryptic error message is referring to object lifecycle management in the Excel JavaScript API. Managing context in the way suggested isn't possible with Office Scripts.

To resolve this, we need to first create the series, then explicitly retrieve it before changing the formatting

```
// (code to create and setup other chart elements)

  // Create the series first
  let s = chart.addChartSeries("Highlights")
  // Then retrieve the series before changing any formatting
  s = chart.getSeries()[2]
  s.setValues(valueRange.getOffsetRange(0, 3))
  s.setChartType(ExcelScript.ChartType.columnClustered)
  s.setOverlap(100)
  let highlightColor = getFillOfNamedRange("highlightColor", workbook)
  s.getFormat().getFill().setSolidColor(highlightColor)

// (code to position chart on worksheet)
}
```

Takeaways:

1. Some error messages in Excel’s Office Scripts Script Editor may be confusing
2. We can easily format a column series using cell background colors

   ```
   range.getFormat().getFill().getColor()
   ```

3. When creating a new series, first add it to the chart, then retrieve it before making changes

   ```
   let s = chart.addChartSeries(“NewSeriesName")
   s = chart.getSeries()[seriesIndex]
   ```