



COS30045 Data Visualisation

Exercise 6.1 D3 Interactivity - mouse over effects

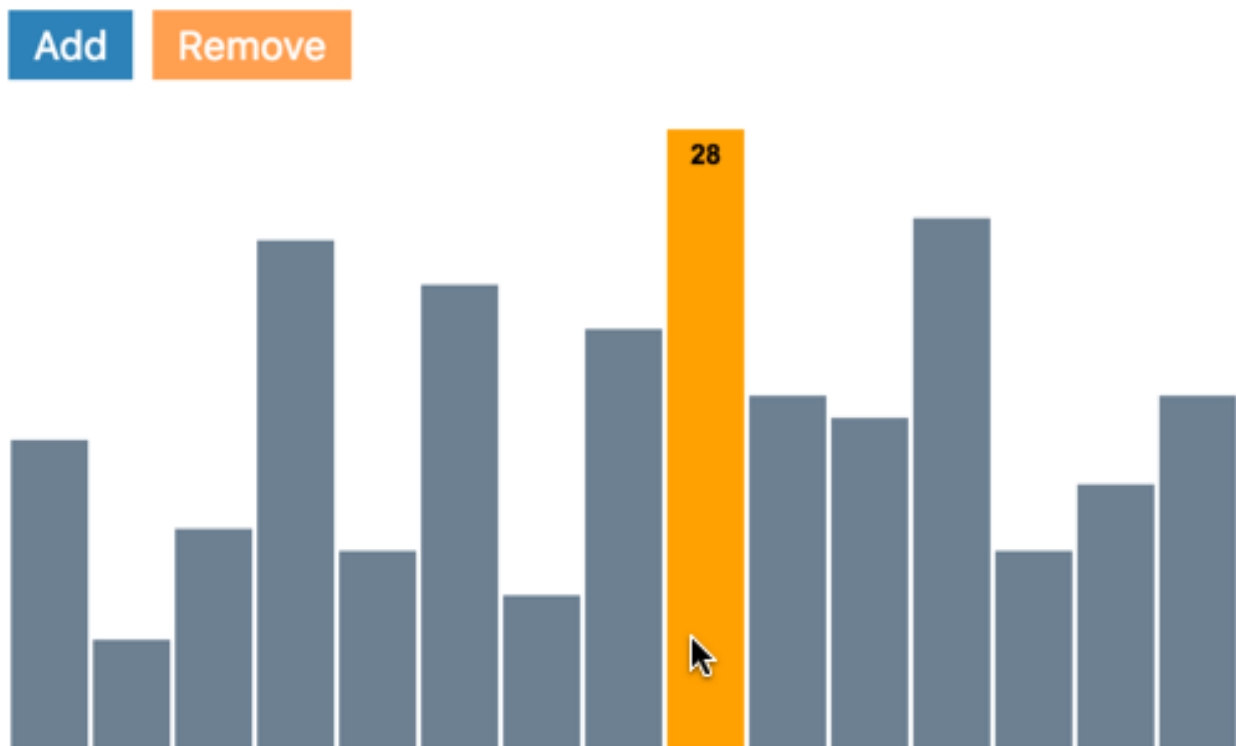
ILO	Create web-based interactive visualisations using real-world data sets.
Aim:	Demonstrate mouse over effects
Resources:	<i>Textbook:</i> Chapter 10 Interactivity - Murray (2017) Interactive Data Visualisation (2nd Ed) on ProQuest
Demonstration	If you are required to demonstrate this exercise we will be looking for: <ul style="list-style-type: none">- code that is appropriate for exercise, well formatted and commented- code that runs correctly and meets the requirements specified in this exercise- an explain programming features and concepts in the code- the ability to successfully edit code to change a specified feature of the program

Note: The functions handling scale have changed between D3 v3 and D3 v4. This is something to be aware of if you are doing your own research into this topic. Make sure you use Murray Ed 2. Code examples from Ed 1 will not work.

Overview

At the end of the Add and Remove Values exercise we had a bar chart that we could add and remove data from. In this exercise we will add a mouse over effect so that the colour of the bar changes and the data is displayed when the user mouses over it. The effects will disappear when the user moves their mouse away.

Bar Chart with Mouse Over



Requirements

- ☐ as per 5.3 plus following mouse over effects:
 - ☐ change in colour of bar
 - ☐ data point displayed as text at top of bar
- ☐ mouse over effect disappears when mouse leaves area of bar
- ☐ include transition effect on mouse overs
- ☐ mouse over effect works on new data added to chart

Step 1: Start with the code from Task 5.3

In the Add and Remove exercise we used D3 to bind an event listener for mouse click to a button. In this task we will use a similar procedure to generate a mouse over effect.

Step 2: Add mouse over effect

Add the `on()` to your main SVG to change the colour of the selected `rect`. Save and test. You find that when the mouse goes over a rectangle it turns orange. While this is nice, it might also be good for it go back to its original colour when the mouse has left the rectangle.

```
.attr("fill", "slategrey")
.on("mouseover", function() {
  d3.select(this)
    .attr("fill", "orange");
});
```

Add a mouse out effect to return the colour of the rectangle back to its original colour. Add a transition, such as a delay to make the mouse in/out smoother.

Step 3: Mouse effect on added bars

Using the add button to add some bars on your page. Does the mouse over effect work on the new bars? **Make it work.**

Step 4: Add a browser tool tip

Add a browser tool tip that displays the data value. As before, make sure that any new bars also display the tool tip.

```
.append("title")
.text(function(d) {
  return "This value is " +d;
});
```

Step 5: Add a SVG tool tip

We don't have much control over the tool tip generated by a browser. To get more control we can use SVG elements. Start by commenting out the code for the browser tool tip, and the highlight mouse overs.

As for the bar highlight mouse over effect, we will use `.on()` to trigger the effect. We will read the value of the bar and print the value at the top of the bar.

First identify the x and y positions of where you want the text to appear.

```
.on("mouseover", function(event, d) {

    var xPos = parseFloat(d3.select(this).attr("x"))
    var yPos = parseFloat(d3.select(this).attr("y"))
```

Then append the text into position.

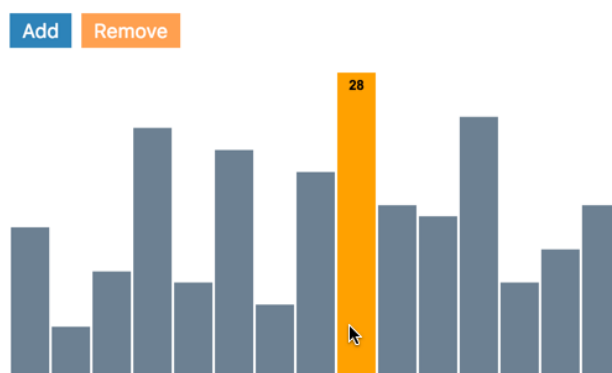
```
svg.append("text")
    .attr("id", "tooltip")
    .attr("x", xPos)
    .attr("y", yPos)
    .text(d);
```

Save and run your code. You will notice that the number is not well placed or well styled. Also the bar highlight mouse over is no longer evident.



Style your text and text position so that the tool tip appears similar to as follows:

Bar Chart with Mouse Over



You may need to do some trial and error to find the right placement!

To remove the tool tip when the user removes the mouse, use `remove()`.

```
.on("mouseout", function(d) {  
    d3.select("#tooltip").remove();  
});
```

Don't forget some code to **return the colour back to grey**.