Histogram Visualization with D3.js

Exercise 4

- Download three files to your website:
  https://my.up.ist.psu.edu/xuz14/330/InClassExercises/Week3/data.tsv
  https://my.up.ist.psu.edu/xuz14/330/InClassExercises/Week3/histogram.html
  https://my.up.ist.psu.edu/xuz14/330/InClassExercises/Week3/histogram.js

Codes

- Separation of HTML and JS codes
- HTML
  - CSS: chart appearance, axes appearance
  - HTML: SVG (chart), JS file(s)
- JS codes
  - Major steps
    - Define the size and location of the chart
    - Define x and y axes
    - Read the file and draw the chart
      - Define the domains of x and y based on data input
      - Create x and y axes
      - Create bars

Bar

```
chart.selectAll(".bar")
  .data(data)
  .enter().append("rect")
  .attr("class", "bar")
  .attr("x", function(d) { return x(d.name); })
  .attr("y", function(d) { return y(d.value); })
  .attr("height", function(d) { return height -y(d.value); })
  .attr("width", x.rangeBand());
```

.attr("x", function(d) { return x(d.name); })

```
//scale of x axis
var x = d3.scale.ordinal()
    .rangeRoundBands([0, width], 0.1);

 x.domain(data.map(function(d) { return d.name; }));
```

.attr("y", function(d) { return y(d.value); })

```
//scale of y axis
var y = d3.scale.linear()
    .range([height, 0]);
```

```
y.domain([0, d3.max(data, function(d) { return d.value; })]);
```

.attr("height", function(d) { return height - y(d.value); })

Change the return value to a constant, say 20. (Line 58 in histogram.js)

See the result

Exercise 4

- Modify some parameters in the histogram.js and see what would happen (one by one)
  - 20 in Line 2 → 50
  - 960 in Line 3 → 500
  - .1 in Line 15 → .9
  - “height - y(d.value)” in Line 58 →20
- Change the column name in data.tsv
  - value → number
- How to fix the problem?

Mouse Events in JS and Array Sorting

User Interaction

- Events
  - Keyboard, mouse events
- What event happened and on what element?
- General approach
  - Specify which element can respond to an event
  - Define what event to respond
  - Define how to respond

Example

- http://www.w3schools.com/js/tryit.asp?filename=tryjs_events_srcelement

- https://jenkov.com/tutorials/svg/scripting.html

In This Assignment: Mouse Events

- Exercise 5:
  - Elements to respond event: all bars
  - What event to respond: mouseover and mosueout
    - To change color
  - Add these statements to the bar statement

```
 .on("mouseover", function () {
      console.log("Over at x: "+d3.mouse(this)[0]+" y: "+d3.mouse(this)[1]);})
.on("mouseout", function () {
      console.log("Out at x: "+d3.mouse(this)[0]+" y: "+d3.mouse(this)[1]);})
```

What to Look at …

- Check the outputs and figure out how to obtain the x and y coordinates of a cursor location
- Modify the JS codes so that bar color changes when the cursor is in and out.
  - Add the following statement to change the color of a bar inside each event function with an appropriate color name

```
d3.select(this).style("fill", colorname)
```

Exercise 6: Mouse Click Event

- The element to respond event: SVG
- What event to respond: mousedown
- What to do: sort the data and redraw the chart
- Add this statement after the bar statement

```
//Clicking on an axis to sort the data
  d3.select("svg")
    .on("mousedown", function() {
      console.log("mousedown");
      console.log("mousedown");
      var coords = d3.mouse(this);
      var xPos = coords[0];
      var yPos = coords[1];
      console.log ("Clicking location: x  "+ xPos+" y "+yPos);
    });
```

What to do next …

- Modify the JS codes so that only responding clicking the areas left to Y or below X axis.
    - Based on xPos or yPos value

Sorting Array Data in JavaScript

- See some examples
  - http://www.w3schools.com/jsref/jsref_sort.asp

Exercise 7: Sorting Array Data

- Pay attention to three statements in the JS codes:

```
var testArray = [6,1,8,5,2,7,4,3,9];
var bardata = [];

 bardata = data;
```

- Add a breakpoint on the statement of x.domain(…), and reload the page.
- When the browser pauses at the breakpoint, in the Console tool, try the following statements and compare the output of each statement with the original data arrays

```
testArray.sort();
tetsArray.reverse();
 bardata.sort();
 bardata.reverse();

 bardata.sort(function(a,b) {return d3.ascending(a.value, b.value)})
 bardata.sort(function(a,b) {return d3.descending(a.name, b.name)})
```

Exercise 8: Updating the graph with newly sorted data.

- General idea
  1. Sorting the data based on the clicking location
  2. Call a function to update the graph after each sorting operation
- What to do
  - Add statements in the mousedown function (Exercise 6) to sort data based on the clicking location.
    - if (xPos …) {}
    - if (yPos …) {}
  - Call a function called updateAxis() after sorting

Graph update function

- After the data is sorted, the graph needs to be updated.
- Add the function after the function type(d)

```
//function to update the graph
function updateAxis() {
    var x0 = x.domain(bardata.map(function (d) {return d.name;}));
    d3.selectAll(".bar")
      .attr("x", function(d) {return x0(d.name);});
   d3.select(".xaxis").call(xAxis);
}
```