# **Data Science and Data Visualization**

# **Lab 2 – Introduction to Javascript and D3.js**

Reference:

JSON: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON>

D3: <https://d3js.org/>

D3 tutorials: <https://github.com/d3/d3/wiki/Tutorials>

D3 load data with parser - <https://github.com/d3/d3-dsv#dsv_parse>

Instructions

* Use ONLY D3 library
* Read chapter 5 and 6 of the Interactive Data Visualization for the Web
* Submit your work (all html, js, css files) to the blackboard assignment for this lab

Details

1. **Info**

Create an HTML web page with the title “Lab 2”. It contains the following text:

* Your name
* Your student ID
* The course tile “Data science and data visualization”
* The lab title “Lab 2 – Introduction to Javascript and D3.js”
* The text “This is all my own work. I did not copy the code from any other source”

# Histogram (JavaScript + SVG)

In the same web page, create a JavaScript function that creates a histogram that shows the distribution of characters in a string. Use this function to draw a bar chart for the letters in your given name (without accents).

Your function should take a svg element and a string and update the svg element so that it displays a histogram of the string with six bins (A-D, E-H, I-L, M-P, Q-U, V-Z). The number of characters that are in those ranges will determine the height of the bars for each bin. Each bin should be 50 pixels wide and each letter occurrence should contribute 50 pixels of height to the bar. The bars should be filled with a blue color and the lines drawn in black.

<svg id=”histogram” width=”300” height=”400”>

<!—YOUR SVG HERE 🡪

</svg>

You should define a function createHistogram such that a call createHistogram(<svgElt>, "Tung") will create the histogram

A blue graph with black lines

Description automatically generated

Hints:

* You may use the following function to add the bar to the svg tag

function addEltToSVG(svg, name, attrs)  
{  
 var element = document.createElementNS("http://www.w3.org/2000/svg", name);  
 if (attrs === undefined) attrs = {};  
 for (var key in attrs) {  
 element.setAttributeNS(null, key, attrs[key]);  
 }  
 svg.appendChild(element);  
}

* Make sure your JavaScript code occurs after the svg elements are defined. HTML is processed sequentially. You may also use an onload function (<https://developer.mozilla.org/enUS/docs/Web/API/GlobalEventHandlers.onload>)
* Give the zero bin a height of 1 pixel so that it will show on the screen
* You can access a character of a string in Javascript via subscript
* You can compare characters with standard comparison operators (e.g. `<','>=') as in Java.
* JavaScript's String.toUpperCase (https://developer.mozilla.org/enUS/docs/Web/JavaScript/Reference/Global\_Objects/String/toUpperCase) or String.toLowerCase (<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/toLowerCase>) methods should make your code more robust.
* Use JavaScript's document.getElementById (https://developer.mozilla.org/enUS/docs/Web/API/document.getElementById) function to get a reference to the SVG element.

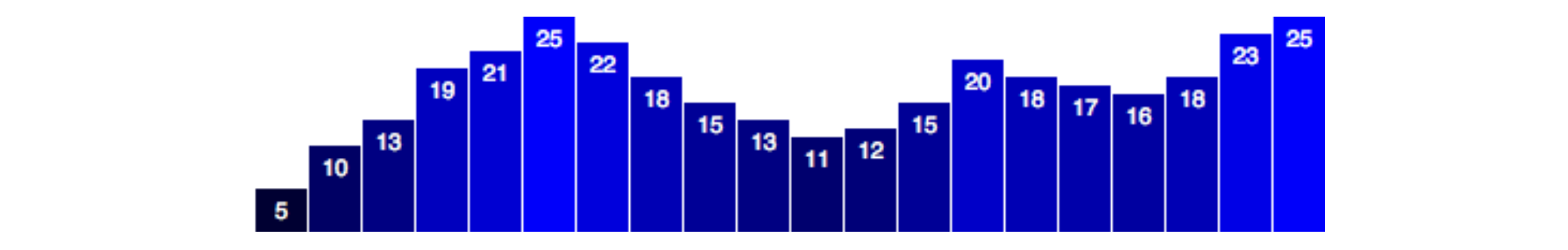
***(Not required) For a challenge, add text labels to the bins.***

# Bar chart with D3

* In the same page, write code to draw bar chart with D3.js.
* The bar chart should be dynamic. It should be able to draw any dataset
* Each column in the bar chart should
  + be colored based on the value that it represents
  + have label to show the value

Create an array of size 20 and initialized it with a random value, then draw the bar chart

The result should look like the following



Hints: Read about the following functions of d3

* selectAll
* data
* enter

# Histogram

* Use the same source file and the bar chart in part 1, draw an histogram for the data with 10 bins

**A computer code with text

Description automatically generated**

