F21DV: Lab 1

# Introduction

This report will serve as an accompaniment to html files containing the JavaScript code to solve lab 1 of the F21DV course. It will go through all 16 parts as described in the pdf overview and highlight key aspects of the code which led to the completion of each of the 32 exercises and mention other possible stylistic choices when appropriate. This report assumes some prior knowledge of JavaScript and will focus on describing the use of the d3.js v7 library.

# Part 1

A GitHub repository was created and showed to be helpful with code management. With the help of the desktop app for GitHub, I managed to smoothly operate on both my IDE of choice (WebStorm) and see the progress on my GitHub page. The repository is the following but may be set to private depending on when this link is clicked : <https://github.com/YoussefBonnaire/DataVisualisation>.

# Part 2

The ground up explanation of the d3.js package we are using with JavaScript is the Document Object Model (DOM) selection. Using the key terms “d3.select(“x”)” and “d3.selectAll(“x”) we are able to select the first, or all objects of type x and give them different styles and attributes. The following code shows how to change the first paragraph with to the different styles described:

<p>First paragraph</p>  
<p>Second paragraph</p>  
<script>  
 // change paragraph to use several styles  
 d3.select("p")  
 .style("color", "red") // Changes colour to red  
 .style('font-style', 'italic') // Changes font style to italic  
 .style('font-family', 'sans-serif') // Changes font to sans-serif  
 .style('font-size', '16pt') // Increases text size to 16pt  
 .style('font-weight', 'bolder') // Makes the text bold  
 .style('text-shadow', '2px 2px blue') // adds a blue 3d aspect with a 2px by 2px shadow  
 .style('word-spacing', '20px') // increases the spaces between words to 20px  
 .style('text-transform','capitalize'); // Capitalises the first letter of each word  
</script>

The objects can also be given different class attributes which can be used to select solely on this class using d3.select(“.MyClass”). Here is an example of how to add a class to a div object with the number 3 written inside and then set the colour to purple.

d3.select('body').append('div').text(3).attr('class', 'under-five')

d3.selectAll('.under-five').style('color', 'purple')

It should be explicitly mentioned that we are working with the version 7 of the d3.js package. This version has some useful simplifications compared to older ones such as chaining which allows the following 4 lines of code to be simplified as the 5th one.

var bodyElement = d3.select("body");  
var div = bodyElement.append("div");  
div.text("Hello World!");  
div.style('color', 'green');  
// Making text green in one line of code  
d3.select("body").append("div").text("Hello World!").style('color', 'green');

This allows for cleaner more legible code.

# Part 3

Data in the form can be bound to a DOM object. What this means is you can iteratively use each object in the data source and perform style and attribute changes which will appear in the dom. This is done by using .data(“MyData”) after selecting a DOM. If you have 3 div objects in your body already and your data is composed of 3 datapoints, when you selectAll the div objects, the datapoints will be bound to each div and can be populated as in the following code:

<div></div>  
<div></div>  
<div></div>  
<script>  
 let data = [{name: 'test', val: 1, color: 'red'},  
 {name: 'other', val: 2, color: 'green'},  
 {name: 'b', val: 3, color: 'blue'}];  
 let paragraph = d3.select("body")  
 .selectAll("div")  
 .data(data)  
 .text(function (d, i) {  
 return 'Color: ' + d.color; // Return value is used to set the 'text'  
 }).style('color', d => d.color); // Set text to said color  
</script>

The .style, .attr, and .text attributes are very strong as they can be populated using functions as is shown above. **Note**, the d => d.color is equivalent to function(d){ return d.color}.

# Part 4

In the case where the div elements do not exists and you decide to selectAll(“div”), you may still bind your data to this selection, the important difference then is that you then add the .enter() function which checks for corresponding div elements. If any