

# 90529 Data Visualisation 2020/21

## Homework 2

*Due: Monday, November 23, 2020, 11:59 pm.*

In this assignment you will make the charts developed for Homework 1 interactive with Javascript and d3.js.

### Implementation

We have provided boilerplate code in `hw2.html`. You are encouraged to build on your own charts that you submitted for homework 1 (copy the corresponding parts of `hw1.html` into the appropriate SVG tags in `hw2.html`). You can also use our reference solution of Homework 1.

As in Homework 1, add your name, your e-mail address, and your uID to the HTML elements at the top. Also make sure your submission is a valid HTML5 file. Check that it is valid by uploading it to the W3C HTML Validator.

Your project structure should look like this:

```
hw2/  
  hw2.html  
  script.js  
  data/  
    anscombe_I.csv  
    anscombe_II.csv  
    anscombe_III.csv  
    anscombe_IV.csv
```

Remember, to be able to access the data files with javascript, you will need to be *serving* the `hw2` directory, not just opening the HTML file in a browser. If your development environment doesn't already launch a server for you, you can start one with one of these commands:

```
cd path/to/hw2  
# for python 2  
python -m SimpleHTTPServer 8080  
# for python 3  
python -m http.server 8080
```

You can view the page at `http://localhost:8080`

### Part I: Basic Events

First of all, fill in the header of the HTML page with your personal data. Next, paste your SVG code from HW1 into the appropriate places in the HTML file

to familiarize with the page structure and check it visualizes correctly. Then remove the SVG code again, as it shall be generated by the .js script.

We have **not** linked the HTML file with the new `script.js` file; your first task is to add this link, as well as a link to the proper version of the D3 library that you decide to use; the choice of version is up to you; in case you use a relative link for the library, the corresponding file shall be included in your project structure, too.

In the boilerplate HTML we provide two controls: a `select` menu for choosing a dataset and a “Random Subset” `checkbox`. Make the `select` menu call the `changeData()` function on a change, and make the “Random Subset” `checkbox` call `randomSubset()` when the box is checked or unchecked, however, we won’t be implementing these functions immediately.

## Part II: Reshaping Things with D3

Now we’re going to start using data to reshape things. We’ve given you the code that loads and parses the CSV data depending on which option the user selects (the data that you used in Homework 1 was Anscombe’s Quartet I. D3 reshapes the CSV file into an array of Javascript objects, each with an `a` and `b` field.

The bulk of this assignment is to adjust the existing SVG elements based on the data. We have provided the scale functions that you will need, but it would still be a good idea to look closely at the syntax that creates each scale function. These mappings from data space to screen space may be slightly different than the scales you implemented manually in Homework 1; feel free to adjust them to your taste.

We have also provided examples of D3’s line and area generator functions; when called with a list of Javascript objects with an `a` parameter, they each produce a `path` string.

*Hint: while not strictly necessary, adding `ids` to each chart will make them much easier to select and attach `.data()`!*

When you are finished implementing this section of the code, changing the dataset `select` menu should reshape all of the plots to fit whichever dataset was picked.

## Part III: More Events

We focus on interactivity in this class; you will usually implement interactivity with Javascript.

First, make any bar in either bar chart change color when the mouse hovers over it using JavaScript. You will notice that the template already includes a hover effect using CSS, however, this can’t be used to, e.g., trigger synchronized highlighting in other views.

Your second interactive component will be to use D3 to listen for clicks on each scatterplot point - when clicked, the browser should log the x and y coordinates of that point to the console.

Note that, because the events are attached in the `update()` function, clicking points won't work unless you first switch datasets! You can fix this by calling `changeData()` when the document loads.

**Extra credit 1:** instead of just dumping the coordinates to the console on clicks, display the coordinates in a tooltip when the mouse hovers over the points. You will need to create additional DOM elements to do this (there are lots of ways to implement a tooltip - how you do it is up to you).

## Part IV: Creating New / Removing Old Elements

Obviously, you don't want to go through hard-coding every element like you did in Homework 1. Your next task is to remove and add elements as needed. You will only need to add to the code that reshapes the bar charts and the scatterplot; the line and area charts should already handle the different data sizes.

The first check of the "Random Subset" checkbox will (probably) send a smaller dataset to the `update` function; the first thing you should do is remove elements that are no longer needed (D3 refers to this as the `exit()` subselection).

Once that is working, you should add new elements in case "Random Subset" is unchecked again and needs to add back bars or points that you previously removed. D3 refers to the set of elements that do not yet exist as the `enter()` subselection. Remember that, as these elements are new, you will need to use D3 to set the non-data attributes as well (such as `circle` radius).

Usually, when you create a visualization with D3, you will start with this `enter()` subselection of non-existent elements - this can be particularly confusing when you're starting out with D3. Always remember to pay attention to whether you are adjusting **all** the nodes in a selection, or just the new `enter()` ones.

## Grading

Your score on this assignment will be as follows:

- 5%: The buttons / select menu trigger the appropriate functions.
- 35%: All charts change in response to changing the dataset menu.
- 5%: All bar chart bars temporarily change color when hovered.
- 5%: X and Y coordinates are logged to the console when clicked in the scatterplot.
- 25%: Elements are removed when "Random Subset" loads a smaller dataset.
- 25%: Elements are created and styled appropriately when "Random Subset" loads a larger dataset.
- 10%: Extra Credit 1: Hovering over a point in the scatterplot shows a tooltip with the x and y data values

10% Extra credit 2: Hovering over a bar in the barchart temporarily highlights the corresponding points in the other charts (Tip: change color to the corresponding circle in the scatterplot; add a similar circle to the corresponding point in the line and area charts; remove them on mouseout)

Remark: Homeworks contribute to the final mark for 20%. The relative contribution of each specific homework to that fraction will be as follows: Homework0: 5%; Homework1 15%; Homework2 (this one) 30%; Homework3 50%.