Assignment 1 - Part B

Submit Assignment

Due Monday by 11:59pm   **Points** 8   **Submitting** a file upload   **File Types** zip
Available Jan 28 at 12am - Feb 4 at 11:59pm 8 days

Assignment 1 - Part B

In Part B of Assignment 1, you will make the charts developed for Part A interactive with Javascript and d3.js.

Implementation

We have provided boilerplate code in a1b.html.

Download boilerplate

You are encouraged to build on your own charts that you submitted for part 1 (copy the corresponding parts of ``a1a.html`` into the appropriate SVG tags in ``a1b.html``).

As in Assignment 1A, 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** **(https://validator.w3.org/#validate_by_upload)** .

Your project structure should look like this:

a1b/

    a1b.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 a1b 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** **(http://localhost:8080)**

Part I: Basic Events

In the boilerplate HTML we provide three controls: a "Staircase" ``button``, a ``select`` menu for choosing a dataset, and a "Random Subset" ``checkbox``. Start by pasting your SVG code from Part A into the appropriate places in the HTML file.

We have **not** linked the HTML file with the new ``script.js`` file; your first task is to add this link, and then make the "Staircase" button call the ``staircase()`` function in ``script.js`` on click.

You should also take this opportunity to 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: Javascript

Now that ``staircase()`` is being called, let's make it do something. Fill out the stub so that the existing rectangles in the first bar chart look like this, **only using Javascript** (no D3 yet!):
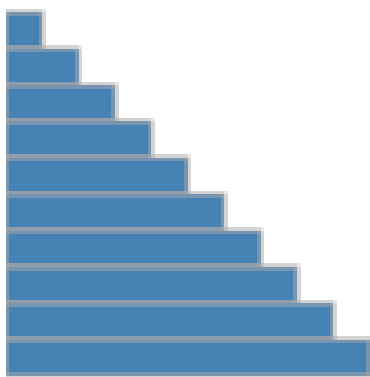
Hint: you will probably want to add an ``id`` to the first bar chart to make it easy to select and iterate through its children.

Part III: 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 (if you haven't noticed, the data that you used in Homework 1 was **Anscombe's Quartet** **(https://en.wikipedia.org/wiki/Anscombe%27s_quartet)** I. D3 reshapes the CSV file into an array of Javascript objects, each with `a` and `b` parameters.

Note that, like Part I, we have **not** linked to d3.js for you; make sure to add this to your HTML as well (you can use their direct link, or you can download the library and save it in your project directory - either is acceptable).

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 ``id``s 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 IV: More Events

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

1. 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.

2. 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 V: Creating New / Removing Old Elements

Obviously, you don't want to go through hard-coding every element as 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.

Extra credit 2: Transitions

D3 makes transitions very simple; for extra credit points, animate each D3 transition (gradually change sizes, positions, and shapes when switching datasets, and fade new and old items with opacity). As we will learn later in the course, animation is **very** attention-grabbing; make sure your animations are tasteful and subtle.

Collaboration.

As stated in the academic integrity policy, you are welcome to discuss the course's ideas, material, and homework with others to understand it better, but **the work you turn in must be your own**. You will be required to include one of the following statements in your README.txt:

- I did not use external resources.
- I used external resources. [followed by a citation list]

Submission.

Submit your work by Monday, February 4, 2019, 11:59 pm. Use the naming convention: "FirstnameLastname_a1b" (e.g., AlvittaOttley_a1b). Please rename the folder **before** compressing it.

Need Help?

If you have questions about this assignment:

1. First, check Piazza to see if others have had a similar problem.
2. If not, we encourage you to post a **public** question on Piazza. Feel free to do it anonymously if you are feeling particularly self-conscious.
3. If the question is sensitive, post a private question on Piazza.
4. Note that you are NOT allowed to post code or solutions on Piazza.
5. We hope to promote a welcoming and inclusive environment. We will monitor the forum and delete any inappropriate posts.

Grading

Your score on this assignment will be as follows:

5%: The buttons / select menu trigger the appropriate functions.
15%: The staircase button correctly reshapes the first bar chart, without using D3.
30%: 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.
20%: Elements are removed when "Random Subset" loads a smaller dataset
20%: Elements are created and styled appropriately when "Random Subset" loads a larger dataset
5%: Extra Credit 1: Hovering over a point in the scatterplot shows a tooltip with the x and y data values
5%: Extra Credit 2: All D3 transitions are animated tastefully