DATA 23700 Winter 2025

Exercise 6: Making data interactive with D3

Due February 17, 2025

In this exercise, students will practice using [D3](#) to make a simple interactive data visualization. Specifically, you will use brushing to implement a cross-filtering design pattern similar to what you might see in many dashboards.

Students should be prepared to spend longer than they expect on this exercise. Learning to use D3 is notoriously difficult, especially if you have no experience with JavaScript. However, once you learn to use D3, you truly can create any interactive graphic that you can imagine!

Students should download the D3 demonstration from lecture notes to use as a starting point. Download the whole folder. You will modify the files we worked on in class to achieve the result described under "technical specification" below.

Students should submit their work by compressing the folder they are working in and submitting a .zip file to Gradescope. Submissions will only receive credit if the course staff can download and run your code on a localhost server with minimal troubleshooting, and if the interactive visualization is implemented to specification.

Technical specification

Students' task is to modify the example from our in class D3 demonstration to produce a scatterplot connected to two histograms using a cross-filtering interaction technique.

The *scatterplot* should show two quantitative variables from the provided `cars.csv` dataset. Students may choose any two variables as long as they are continuous rather than discrete.

The *two histograms* should show the distribution of each of the two variables shown in the scatterplot, respectively. In other words, these histgrams should show the marginal distribution along each axis of the histogram.

When the user *brushes to select values on the scatterplot*, the selected data points should be highlighted in both histograms. Background bars on the historgrams should show the un-filtered distributions. The resulting visualization will be a mock-dashboard, where multiple views show the selected data points using different encodings.

Students may choose the *layout* of these three visualizations. The easiest choice might be to create three separate SVG elements, one for each chart. However, a more effective layout would be to draw the histograms along the x- and y-axis of the scatterplot, such that the shared axes are not redrawn for multiple charts. This second solution would be more difficult to implement and would involve drawing all three visualizations in one composite chart, inside one SVG element.

There is *no written portion* to this exercise.

Hints

Students will need to consider the following:

1. How do you need to modify the provided scatterplot code in order to show different variables?

2. Do you need additional HTML elements or D3 selections to add charts to the display?

3. How will you bin the data from the scatterplot in order to plot it in the histograms?

4. How does using binned data in the histograms change the way you draw bars along the binned axis?

5. What's a good number of bins to show in each histogram? Experiment with this setting.

6. How does the brush end function need to change now that we have more than one chart responding to the selection?

7. Do you want to change the CSS in the style block of `index.html`, e.g., to use different colors or manipulate the layout of your charts?

When programming in D3, it's typical to Google search example code and modify it to meet your needs.

Please start early and come to office hours if you get stuck.