Due to the small-scale nature of this project, some of my usual conventions/methods were changed or not used. With larger scale projects, I do prefer to use any libraries and tools that simplify the production (ex. Sass, material ui, redux, etc.) However, for this small project I chose not to use those. Instead, I opted for a smaller build with minimal libraries to not clutter the project.

I do prefer to design my systems as modular as possible to make the code more maintainable. I try to avoid inline styling as much as possible, in favor of having a centralized location to do all of my styling. This is to allow making simple changes easier and faster.

For the components I generally allow for optional parameters or props to make things more customizable if it ever needs to be reused elsewhere. It did seem unnecessary for how simple this project was, but it is a general convention I follow.

For this project, I wasn’t sure if the data should be loaded directly from a URL, or from the file that was sent. As such, the only option I gave for data loading was a direct file upload from the hard disk. This data must follow the exact geojson format given be the file. I did try to allow for flexibility in the dataset though. So, given that the features have any number of “year” properties and a “total” property, the interface will still load in that information. As a result, the interface components aren’t specific to car accidents. This allows loading any research data as long as it follows the same format.

Given more time, I would like to allow for a URL upload as well. This would be a relatively quick addition as the d3 library has that functionality built in with the d3.csv and d3.json functions. All that would be needed is an extra Input component which takes in a URL.

I did not focus on the styling too much either, as my artistic sense isn’t the greatest.

The web page is somewhat responsive, it will not scale the components if the window is resized in real time. I felt, that adding listeners was unnecessary and does have a slight performance impact. This is because of how I utilized the fitSize function within d3. As a result, the UI component its self doesn’t respond to any css media queries, while the rest of the page does.

As for the rough estimate, I would say it took me around 5 hours spread in very short periods of development across several days.