**Description:**

For our project we decided to explore the data available on data.seattle.gov. We looked through many different data sets and while we found some that could be very interesting, many did not have the data in a format that was easily consumed (.pdf files) or were missing critical information.

The data provided by data.seattle.gov was also appealing because it provides an API through Socrata and thus it was relatively painless to pull in the required data.

The data set we chose to work with was the Seattle Police Department 911 Incident Response data set. The data is updated every four hours and thus provides an up-to-date look at crime, accidents, and other incidents around the city. We thought it would be interesting to visually display this on a map that would allow users to display the data based on date and time, type of activity and by district or neighborhood. Furthermore, we wanted to provide functionality to click on an individual data point to further display information about that particular incident. We thought about different ways we could display the data as well. We thought it would be great to have a heat map of the selected incident around the Seattle area. This would allow for a quick glance comparison if individual data points were not necessary.

Our final interactive visualization allows for all of the above with the exception of selecting based on a neighborhood and the heat map. We were able to find GeoJson data from Zillow that gave us the geometry for the neighborhoods around Seattle. However, the problem was that we decided to use Google Maps to display individual data points with D3. The svg elements are placed in an overlay on the map. This overlay is sandwiched in between other layers for the map. Thus that layer cannot receive mouse events. We overcame this with data points by putting “invisible” Google Maps markers on all the same locations as the svg elements. This makes it appear that the svg elements are clickable, when they actually are not. This would make tracking mouse coordinates to select a particular neighborhood very difficult.

The other option was to have a drop down that would allow the user to select a neighborhood. The problem with this is that the API only allows us to bring in data based on square coordinate blocks. The neighborhoods are obviously not square and thus quite a bit of additional logic would need to be added to filter anything outside of the boundaries. We decided to hold off on implementing the neighborhood selection to work on other aspects in case we ran out of time. We did run out of time and thus we were unable to implement it.

The heat map didn’t make it into the final visualization because we simply ran out of time. There was a considerable amount of logic to get this to work properly and we were too busy fixing bugs and polishing other aspects of the visualization.