Anthony McIntosh and Matt Sichlerman
Assignment 3 - Step 9

Overall, the code written in Part 3 is far more complicated than in Part 1. This is true for maintainability, extensibility, and implementation. For maintainability, the code in Part 3 is much more susceptible to bugs caused by changes in data or screen resolution. Part 3 relies heavily on absolute position coordinates to create the visual of the graph, where as Part 1 does not. This means that Part 1 will automatically adjust to changes better than the graph for Part 3. For extensibility and implementation, Part 1 allowed for much more flexibility in positioning and features in the graph. It was very easy to add axes to the graphs, as well as labels throughout. These features are simply added by appending additional “g” objects to the chart with the appropriate attributes. In Part 3 however, the attributes and elements available via DOM are less extensible, and require more looping and absolute positioning to effectively add features. This also makes the implementation of new features on to the Part 3 graph more difficult. Similarly, adding things such as axes, X scales, and Y scales is difficult using native javascript. There is no easy way to add all labels in native javascript without looping and hardcoding their position based on the position of other elements. However, in d3 it is easy to accomplish this by simply calling the appropriate method (d3.bottomAxis, d3. leftAxis).

There is overall less of a difference for the user between Part 3 and Part 1 then there is for the developer. Most of the differences between the two parts relate to implementation, maintainability, and extensibility. The main one of these that affects the user is extensibility. There are features and visuals that can be implemented using d3 that are simply not possible (or at least not feasible) using native javascript. For this reason, the user experience can be better when using d3 since they will experience more features. However, in our example, between Part 1 and Part 3 the features are the same and the only difference comes in implementation. For this reason, the user will not notice any major differences in usability at this level. Usability differences appear largely in complex visualizations using D3, rather than simple visualizations such as this.