Introduction to Basic D3

What is D3

D3.js is a JavaScript library which provides powerful visualization components. To link to D3's v5 release, add following code to the html file.

```
<script src="https://d3js.org/d3.v5.min.js"></script>
```

A basic syntax is like following, D3's syntax is "chain syntax," which means the order matters.

```
d3.select(...)
  .[other operations]
```

Check D3's wiki for more detailed tutorials.

Append SVG

We may want to append a svg on the html body first, on which we can "draw" something.

```
var svg = d3.select("body").append("svg")
  .attr("width", width + margin.left + margin.right)
  .attr("height", height + margin.top + margin.bottom)
```

Load Data from CSV

It is very easy to load data in D3.

```
d3.csv("/path/data.csv")
  .then(function(data) {
  // operations with the data
  }
```

Visualization Components

D3 has a large gallery of visualization components, from which you can select one with your need of visualization. Like box plot, treemap, force-directed graph, etc. Check D3's wiki for the gallery.

Scale Function

We may want to re-scale points, so that the plot could "fit-in" the area of our canvas.

```
xScale = d3.scaleLinear()
 .range([0,width]) //rescale into the canvas' width
 .domain([min_x, max_x]) //domain of the values
```

Append Axis

```
svg.append("g") //append group
   .attr("class", "x axis")
   .attr("transform", "translate(0, height)") // axis' location
   .call(d3.axisBottom(xScale));
```

Enter the Data

Make a selection first, and each element in the data array is paired with the corresponding selected node.

```
svg.selectAll(".dot")
    .data(data_array)
    .enter()
    .append("g")
    .append("circle")
    .[...] //define attributes with elements in data_array
```

Update the Data

Once data are plotted, one can update the data with similar ways as entering the data.

```
d3.selectAll('...')
  .attr(...) // updates data
```

Tooltip

Tooltip is a useful tool to display additional information. For example, when mouse move to something, it will show additional information regarding that thing.

```
var tooltip = d3.select("body")
    .append("div")
    .attr("class", "tooltip");
```

After the tooltip is defined, use it when mouse hover to something.

```
svg.selectAll("...")
   ...
   .on("mouseover", function(d){
        //location of tooltip
        tooltip.style("left", ...)
               .style("right", ...)})
   .on("mouseout", function(d) {
        // operations of restoring to normal});
```

Transitions

If you are interested in how the plot changes when data change, an animated transitions could be very useful. The transition code is added in the code for updating the data.

```
d3.selectAll("...")
  .transition()
  .duration(100) //duration of the transition
  .delay(function(d, i) {return i * 10;}); //if want to follow the trace of each point, delay could be useful
  .attr(...) //other operations of the data
```