LAB EXERCISE 6

Learning Outcomes:

To create linked views using d3.js, donut chart and bar chart.

Create new HTML file in VS Code. Type ! and press Tab key to get skeleton of HTML code.

Load d3.js libraries and CSS style in the <head> section of HTML

```
<!-- Load d3.js -->
<script src="https://d3js.org/d3.v2.js"></script>

<style type="text/css">

        #pieChart {
            position:absolute;
            top:10px;
            left:10px;
            width:400px;
            height: 400px;
        }

        #barChart {
        position:absolute;
        top:160px;
        left:410px;
        height: 250px;
        }

        .slice {
        font-size: 12pt;
        font-family: Verdana;
        fill: white;
        font-weight: bold;
                }

        .axis text {
            font-family: Verdana;
            font-size: 11px;
        }

        .title {
            font-family: Verdana;
            font-size: 20px;

        }

        .xAxis {
            font-family: verdana;
            font-size: 11px;
            fill: black;
        }

        .yAxis {
            font-family: verdana;
            font-size: 11px;
            fill: white;
        }
</style>
```

Add two div areas in HTML body

Link javascript file

```
<body>
    <div id="pieChart"></div>
    <div id="barChart"></div>
    <script type="text/javascript" src="dashboard.js"></script>
</body>
```

Save your HTML file

Lesson 1: Creating an interactive donut chart

Create a new javascript file named dashboard.js.

Step 1: Create a variable for text formatting

```
var formatAsPercentage = d3.format("%"),
              formatAsPercentage1Dec = d3.format(".1%"),
              formatAsInteger = d3.format(","),
              fsec = d3.time.format("%S s"),
              fmin = d3.time.format("%M m"),
              fhou = d3.time.format("%H h"),
              fwee = d3.time.format("%a"),
              fdat = d3.time.format("%d d"),
              fmon = d3.time.format("%b");
```

Step 2: Create a function to plot pie/donut chart.

There are few sections in this function including

a) **Dataset creation**
b) **Setting for the pie chart**
c) **SVG element**
d) **arc**
e) **arc animation**
f) **title**

```
function dsPieChart(){


…..


}

dsPieChart();
```

```
function dsPieChart(){

   var dataset = [
         {category: "Samad", measure: 0.30},
         {category: "Phang", measure: 0.25},
         {category: "Johan", measure: 0.15},
         {category: "Rita", measure: 0.05},
         {category: "Lenny", measure: 0.18},
         {category: "Pian", measure:0.04},
         {category: "Siti", measure: 0.03}
         ]
         ;

   var   width = 400,
            height = 400,
            outerRadius = Math.min(width, height) / 2,
            innerRadius = outerRadius * .999,
            // for animation
            innerRadiusFinal = outerRadius * .5,
            innerRadiusFinal3 = outerRadius* .45,
            color = d3.scale.category20()
            ;

   var vis = d3.select("#pieChart")
        .append("svg:svg")           //create the SVG element
        .data([dataset])             //associate data
        .attr("width", width)        //set the width and height
        .attr("height", height)
        .append("svg:g")             /make a group to the chart
        //move the center of the pie chart
        .attr("transform", "translate(" + outerRadius + "," +
outerRadius + ")");

   var arc = d3.svg.arc()        // create <path> elements
         .outerRadius(outerRadius).innerRadius(innerRadius);

   // for animation

   var arcFinal = d3.svg.arc()
               .innerRadius(innerRadiusFinal)
               .outerRadius(outerRadius);

   var arcFinal3 = d3.svg.arc()
               .innerRadius(innerRadiusFinal3)
               .outerRadius(outerRadius);

   var pie = d3.layout.pie()
        .value(function(d) { return d.measure; });

   var arcs = vis.selectAll("g.slice")
         .data(pie)             //associate the generated pie data
         .enter()               //create <g> elements
         .append("svg:g")       //create a group to hold each slice
         .attr("class", "slice")     //set style in the slices
         .on("mouseover", mouseover)
         .on("mouseout", mouseout)
         .on("click", up)
         ;
        arcs.append("svg:path")
               //set the color for each slice
               .attr("fill", function(d, i) { return color(i); } )
               .attr("d", arc)     // actual SVG path
               .append("svg:title") //mouseover title
               .text(function(d) { return d.data.category + ": " +
formatAsPercentage(d.data.measure); });

        d3.selectAll("g.slice").selectAll("path").transition()
                   .duration(750)
                   .delay(10)
                   .attr("d", arcFinal )
                   ;

     // Add a label to the larger arcs, translated to the arc centroid

     arcs.filter(function(d) { return d.endAngle - d.startAngle > .1; })
               .append("svg:text")
           .attr("dy", ".35em")
           .attr("text-anchor", "middle")
           .attr("transform", function(d) { return "translate(" +
arcFinal.centroid(d) + ")rotate(" + angle(d) + ")"; })
         //.text(function(d) { return formatAsPercentage(d.value); })
           .text(function(d) { return d.data.category; })
         ;

      // Computes the label angle of an arc, convert from rad to deg.

         function angle(d) {
             var a = (d.startAngle + d.endAngle) * 90 / Math.PI - 90;
             return a > 90 ? a - 180 : a;
         }


         // Pie chart title
         vis.append("svg:text")
         .attr("dy", ".35em")
           .attr("text-anchor", "middle")
           .text("Revenue 2022")
           .attr("class","title");



         function mouseover() {
           d3.select(this).select("path").transition()
               .duration(750)
               .attr("d", arcFinal3);
         }

         function mouseout() {
           d3.select(this).select("path").transition()
               .duration(750)
               .attr("d", arcFinal);
         }

   function up(d, i) {

   /* update bar chart when user selects piece of the pie chart */
                     updateBarChart(d.data.category, color(i));


   }
}

dsPieChart(); //execute
```

Step 3: Set Initial group value. By default, all sections of pie chart are selected.

```
var group = "All";
```

Step 4: Get data from CSV and Create Bar chart within d3.csv

```
d3.csv("revenueData.csv",function(revdata){

   function datasetChosen(group)  {
     var ds = [];
     for (x in revdata) {
         if(revdata[x].group==group){
             ds.push(revdata[x]);
         }
         }
       return ds;
    }


/*
        ############# BAR CHART ##################
        -------------------------------------------
        */

    //set margin to plot bar chart
    function dsBarChartBasics() {

        var margin = {top: 30, right: 5, bottom: 20, left: 50},
        width = 500 - margin.left - margin.right,
        height = 250 - margin.top - margin.bottom,
        colorBar = d3.scale.category20(),
        barPadding = 1
        ;

        return {
            margin : margin,
            width : width,
            height : height,
            colorBar : colorBar,
            barPadding : barPadding
        }
        ;
    }
```

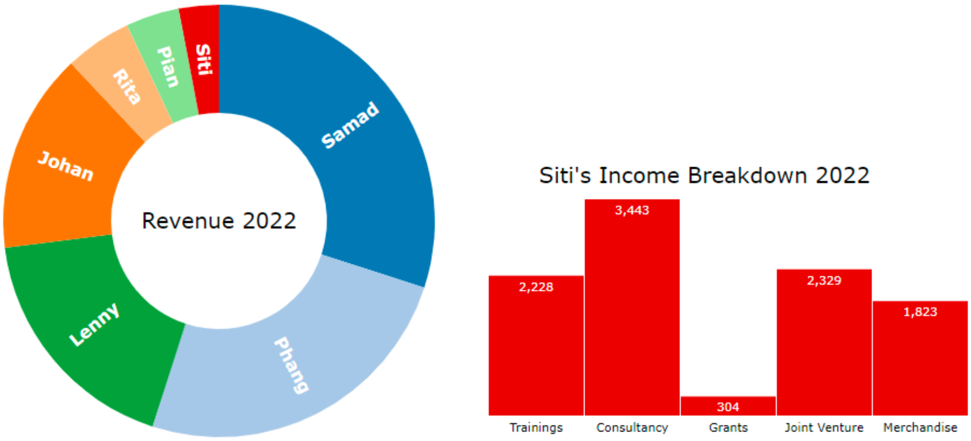
```
function dsBarChart() {

        var firstDatasetBarChart = datasetChosen(group);
        var basics = dsBarChartBasics();
        var margin = basics.margin,
            width = basics.width,
            height = basics.height,
            colorBar = basics.colorBar,
            barPadding = basics.barPadding
            ;

        //set x and y scale
        var xScale = d3.scale.linear()
                    .domain([0, firstDatasetBarChart.length])
                    .range([0, width]);
        var yScale = d3.scale.linear()
                    .domain([0, d3.max(firstDatasetBarChart, function(d) {
return d.income; })])
                    .range([height, 0]);


        //Create SVG element, select css style for bar chart

        var svg = d3.select("#barChart")
                    .append("svg")
                    .attr("width", width + margin.left + margin.right)
                    .attr("height", height + margin.top + margin.bottom)
                    .attr("id","barChartPlot")
                    ;

        var plot = svg.append("g")
                    .attr("transform", "translate(" + margin.left + "," +
margin.top + ")")
                    ;

                    plot.selectAll("rect")
                    .data(firstDatasetBarChart)
                    .enter()
                    .append("rect")
                    .attr("x", function(d, i) {
                        return xScale(i);
                    })
                    .attr("width", width / firstDatasetBarChart.length -
barPadding)
                    .attr("y", function(d) {
                        return yScale(d.income);
                    })
                    .attr("height", function(d) {
                        return height-yScale(d.income);
                    })
                    .attr("fill", "lightgrey")
                    ;

        // Add y labels to plot

        plot.selectAll("text")
            .data(firstDatasetBarChart)
            .enter()
            .append("text")
            .text(function(d) {
                    return formatAsInteger(d3.round(d.income));
            })
            .attr("text-anchor", "middle")
```

```
 // Set x position to the left edge of each bar plus half the bar width

                .attr("x", function(d, i) {
                        return (i * (width / firstDatasetBarChart.length)) +
((width / firstDatasetBarChart.length - barPadding) / 2);
                })
                .attr("y", function(d) {
                        return yScale(d.income) + 14;
                })
                .attr("class", "yAxis")
        ;

        // Add x labels to chart

        var xLabels = svg.append("g")
                .attr("transform", "translate(" + margin.left + "," +
(margin.top + height)  + ")")
        ;

        xLabels.selectAll("text.xAxis")
                .data(firstDatasetBarChart)
                .enter()
                .append("text")
                .text(function(d) { return d.category;})
                .attr("text-anchor", "middle")

    // Set x position to the left edge of each bar plus half the bar width
                .attr("x", function(d, i) {
                                        return (i * (width /
firstDatasetBarChart.length)) + ((width / firstDatasetBarChart.length -
barPadding) / 2);
                                })
                .attr("y", 15)
                .attr("class", "xAxis")
                ;

        // Add Title of bar chart

        svg.append("text")
                .attr("x", (width + margin.left + margin.right)/2)
                .attr("y", 15)
                .attr("class","title")
                .attr("text-anchor", "middle")
                .text("Overall Income Breakdown 2022")
                ;

        }


//execute bar chart function
        dsBarChart();
```

```
/* ** UPDATE CHART ** */

        /* updates bar chart on request. Set the function as global */

        window.updateBarChart= function(group, colorChosen) {

                var currentDatasetBarChart = datasetChosen(group);

                var basics = dsBarChartBasics();

                var margin = basics.margin,
                    width = basics.width,
                    height = basics.height,
                    colorBar = basics.colorBar,
                    barPadding = basics.barPadding
                    ;

                var  xScale = d3.scale.linear()
                    .domain([0, currentDatasetBarChart.length])
                    .range([0, width])
                    ;


                var yScale = d3.scale.linear()
                .domain([0, d3.max(currentDatasetBarChart, function(d) {
return d.income; })])
                .range([height,0])
                ;

                var svg = d3.select("#barChart svg");

                var plot = d3.select("#barChartPlot")
                    .datum(currentDatasetBarChart)
                    ;


/* Note that here we only need to select the elements */

                plot.selectAll("rect")
                .data(currentDatasetBarChart)
                .transition()
                    .duration(750)
                    .attr("x", function(d, i) {
                        return xScale(i);
                    })
                    .attr("width", width / currentDatasetBarChart.length
- barPadding)
                    .attr("y", function(d) {
                        return yScale(d.income);
                    })
                    .attr("height", function(d) {
                        return height-yScale(d.income);
                    })
                    .attr("fill", colorChosen)
                    ;

// target the text element(s) which has a yAxis class defined

                plot.selectAll("text.yAxis")
                    .data(currentDatasetBarChart)
                    .transition()
                    .duration(750)
                    .attr("text-anchor", "middle")
                    .attr("x", function(d, i) {
                            return (i * (width /
currentDatasetBarChart.length)) + ((width /
currentDatasetBarChart.length - barPadding) / 2);
                    })
                    .attr("y", function(d) {
                            return yScale(d.income) + 14;
                    })
                    .text(function(d) {
                        return formatAsInteger(d3.round(d.income));
                    })
                    .attr("class", "yAxis")
                ;

// target the text element(s) which has a title class defined

                svg.selectAll("text.title")
                    .attr("x", (width + margin.left + margin.right)/2)
                    .attr("y", 15)
                    .attr("class","title")
                    .attr("text-anchor", "middle")
                    .text(group + "'s Income Breakdown 2022")
                ;
            }


});   //close d3.csv
```

Step 4: Test your script.

Sample Output



Reference: http://bl.ocks.org/diethardsteiner/3287802

LAB TASK

Modify the dashboard.js script.

a. Create Line chart and associate it to main pie/donut chart. Plot the line based on year and performance data in csv file.
b. Consider appropriate pre-attentive attributes such as main title, label, color, legend,etc.

Submission item: HTML file.

Submission platform: ULearn

Deadline: 30th October 2023