LAB EXERCISE 3

Learning Outcomes:

1. To create box plot chart with SVG and D3.js
2. To create a scatterplot with tooltip

Lesson 1: Creating Box Plot

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

Step 1: Load d3.js in <head> section of HTML

<head>

```
<!-- Load d3.js -->
<script src="https://d3js.org/d3.v4.js"></script>
```

</head>

Step 2: Create a div to put the graph in <body> section of HTML

<body>

```
<div id="my_dataviz"></div>
```

…

Start writing the scripts in <body> section of HTML. There are few code fragments within the <script> compound.

1. Dimension and margin of the graph
2. SVG
3. Parsing the data
4. Adding X axis and Y Axis
5. Adding bars

Step 3: Set the dimensions and margins of the graph

```
<script>

const margin = {top: 10, right: 30, bottom: 30, left: 60},
    width = 460 - margin.left - margin.right,
    height = 400 - margin.top - margin.bottom;…

</script>
```

Step 3: Drawing graph with SVG

Append the SVG object to the body of the page. Add the following code to the script. The coordinate is based on the margin set in Step 3.

```
const svg = d3.select("#my_dataviz")
    .append("svg")
    .attr("width", width + margin.left + margin.right)
    .attr("height", height + margin.top + margin.bottom)
    .append("g")
    .attr("transform", `translate(${margin.left},${margin.top})`);
"translate(" + margin.left + "," + margin.top + ")");
```

Padding can be applied with a **<g>** element translated by the desired value. The SVG <g> element is used to group SVG shapes together. Once grouped you can transform the whole group of shapes as if it was a single shape.

For more details about <g> attribute → https://jenkov.com/tutorials/svg/g-element.html

Step 4: Parse the data from github and compute summary statistics for each species

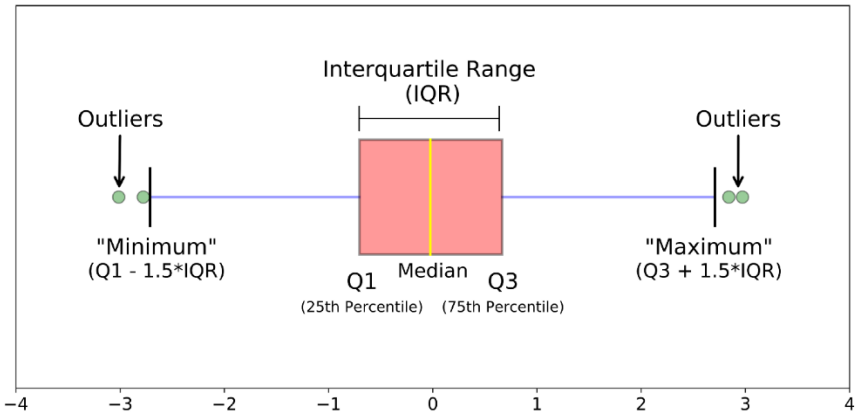
```
d3.csv("https://raw.githubusercontent.com/holtzy/D3-graph-
gallery/master/DATA/iris.csv", function(data) {


// Compute quartiles, median, inter quantile range min and max --> these info
are then used to draw the box.

  var sumstat = d3.nest() // nest function allows to group the calculation per
level of a factor
    .key(function(d) { return d.Species;})
    .rollup(function(d) {

      q1 = d3.quantile(d.map(function(g) { return
g.Sepal_Length;}).sort(d3.ascending),.25)
      median = d3.quantile(d.map(function(g) { return
g.Sepal_Length;}).sort(d3.ascending),.5)
      q3 = d3.quantile(d.map(function(g) { return
g.Sepal_Length;}).sort(d3.ascending),.75)

      interQuantileRange = q3 - q1
      min = q1 - 1.5 * interQuantileRange
      max = q3 + 1.5 * interQuantileRange
      return({q1: q1, median: median, q3: q3, interQuantileRange:
interQuantileRange, min: min, max: max})
    })
    .entries(data)
```



A boxplot is a standardized way of displaying the distribution of data based on a five number summary ("minimum", first quartile (Q1), median, third quartile (Q3), and "maximum"). It can tell you about your outliers and what their values are. It can also tell you if your data is symmetrical, how tightly your data is grouped, and if and how your data is skewed. Read More.

Step 5: Add X axis

```
var x = d3.scaleBand()
    .range([ 0, width ])
    .domain(["setosa", "versicolor", "virginica"])
    .paddingInner(1)
    .paddingOuter(.5)
  svg.append("g")
    .attr("transform", "translate(0," + height + ")")
    .call(d3.axisBottom(x))
```

Step 6: Add Y axis

```
var y = d3.scaleLinear()
    .domain([3,9])
    .range([height, 0])
  svg.append("g").call(d3.axisLeft(y))
```

Step 7: Add the main vertical line

```
svg
    .selectAll("vertLines")
    .data(sumstat)
    .enter()
    .append("line")
      .attr("x1", function(d){return(x(d.key))})
      .attr("x2", function(d){return(x(d.key))})
      .attr("y1", function(d){return(y(d.value.min))})
      .attr("y2", function(d){return(y(d.value.max))})
      .attr("stroke", "black")
      .style("width", 40)
```

Step 7: Add rectangle for the main box

```
// rectangle for the main box
  var boxWidth = 100
  svg
    .selectAll("boxes")
    .data(sumstat)
    .enter()
    .append("rect")
        .attr("x", function(d){return(x(d.key)-boxWidth/2)})
        .attr("y", function(d){return(y(d.value.q3))})
        .attr("height", function(d){return(y(d.value.q1)-y(d.value.q3))})
        .attr("width", boxWidth )
        .attr("stroke", "black")
        .style("fill", "#FFA500")
```

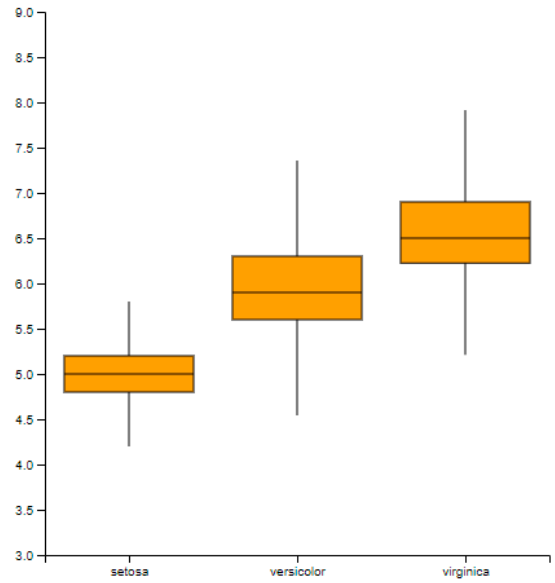
Step 7: Show the median

```
svg
    .selectAll("medianLines")
    .data(sumstat)
    .enter()
    .append("line")
      .attr("x1", function(d){return(x(d.key)-boxWidth/2) })
      .attr("x2", function(d){return(x(d.key)+boxWidth/2) })
      .attr("y1", function(d){return(y(d.value.median))})
      .attr("y2", function(d){return(y(d.value.median))})
      .attr("stroke", "black")
      .style("width", 80)
})
```

Step 8: Close the script

```
</script>
```

Sample Output



Lesson 2: Creating Scatterplot

Refer to Lesson 1 for Step 1 to step 3.

Step 4: Parse the data from internet

```
d3.csv("https://raw.githubusercontent.com/holtzy/data_to_viz/master/Example_da
taset/2_TwoNum.csv", function(data) {
```

Step 5: Add X axis

```
var x = d3.scaleLinear()
    .domain([0, 4000])
    .range([ 0, width ]);
  svg.append("g")
    .attr("transform", "translate(0," + height + ")")
    .call(d3.axisBottom(x));
```

Step 6: Add Y axis

```
// Add Y axis
  var y = d3.scaleLinear()
    .domain([0, 500000])
    .range([ height, 0]);
  svg.append("g")
    .call(d3.axisLeft(y));
```

Step 7: Add dots for data points

```
svg.append('g')
    .selectAll("dot")
    .data(data)
    .enter()
```

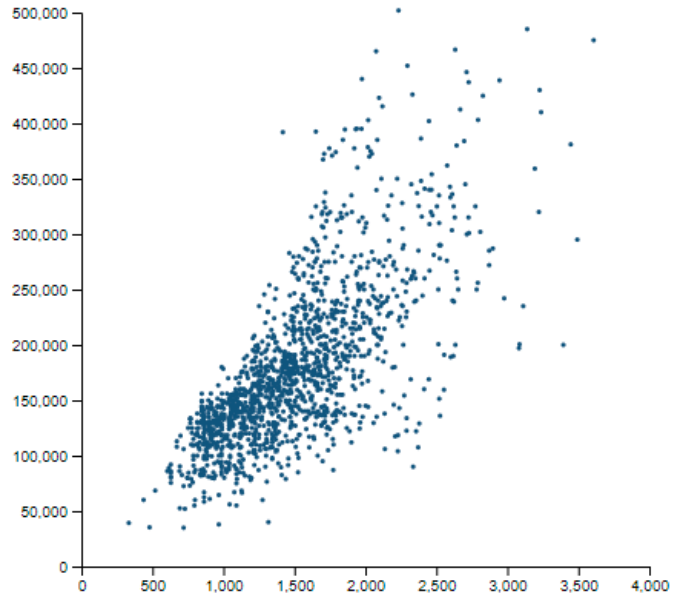
```
    .append("circle")
      .attr("cx", function (d) { return x(d.GrLivArea); } )
      .attr("cy", function (d) { return y(d.SalePrice); } )
      .attr("r", 1.5)
      .style("fill", "#2B547E")

})
```

Step 8: Close the script

```
</script>
```

Sample Output



Lesson 3: Creating Scatterplot with tooltip

Duplicate your HTML file that you have created in Lesson 2.

Step 1: Modify the margin

```
var margin = {top: 10, right: 30, bottom: 30, left: 60},
    width = 460 - margin.left - margin.right,
    height = 450 - margin.top - margin.bottom;
```

Step 2: Modify the X axis scale

```
  // Add X axis
  var x = d3.scaleLinear()
    .domain([0, 3000])
    .range([ 0, width ]);
  svg.append("g")
    .attr("transform", "translate(0," + height + ")")
    .call(d3.axisBottom(x))
```

Step 3: Modify the Y axis scale

```
var y = d3.scaleLinear()
    .domain([0, 400000])
    .range([ height, 0]);
  svg.append("g")
    .call(d3.axisLeft(y));
```

Step 4: Add a tooltip div and set the opacity to 0 so that it is not visible by default

```
var tooltip = d3.select("#my_dataviz")
    .append("div")
    .style("opacity", 0)
    .attr("class", "tooltip")
    .style("background-color", "white")
    .style("border", "solid")
    .style("border-width", "2px")
    .style("border-radius", "5px")
    .style("padding", "10px"

    .style("position", "absolute")
```

Step 5: Add a function that will change the opacity of the tooltip to 1 when the user hovers a point.

```
var mouseover = function(d) {
    tooltip
      .style("opacity", 1)
  }


var mousemove = function(d) {
    tooltip
      .html("The exact value of<br>the Ground Living area is: " + d.GrLivArea)
      .style("left", (d3.mouse(this)[0]+90) + "px")
      .style("top", (d3.mouse(this)[1]) + "px")
  }
```

***It is important to put the +90: otherwise, the tooltip is exactly where the point located.**

Step 6: Add a function that will change the opacity of the tooltip back to 0 when the user leaves a point.

```
var mouseleave = function(d) {
```

```
    tooltip
      .transition()
      .duration(200)
      .style("opacity", 0)
  }
```

Step 7: Modify the adding dots code

```
.selectAll("dot")
    .data(data.filter(function(d,i){return i<50})) // the .filter part is just
to keep a few dots on the chart, not all of them
    .enter()
    .append("circle")
      .attr("cx", function (d) { return x(d.GrLivArea); } )
      .attr("cy", function (d) { return y(d.SalePrice); } )
      .attr("r", 7) // radius size, could map to another data dimension
      .style("fill", "#69b3a2")
      .style("opacity", 0.3)
      .style("stroke", "white")
    .on("mouseover", mouseover )
    .on("mousemove", mousemove )
    .on("mouseleave", mouseleave )

})
```

Reference: https://d3-graph-gallery.com/

LAB TASK

Download customer segmentation dataset from > https://www.kaggle.com/code/heeraldedhia/kmeans-clustering-for-customer-data/data

a. Create TWO boxplots based on any two numeric attributes. Use *cluster* attribute as grouping that to be shown in X axes.
b. Select any TWO variables to create scatterplot with tooltips
c. Consider appropriate pre-attentive attributes such as label, color etc.
d. Write a summary of insights based on the graphs created in (a) and (b).

Submission item: HTML file.

Submission platform: MS Teams

Deadline: 8th October 2023