

VISUALIZATION ASSGINMENT 1

SUBMITTED BY
JAYAKRISHNA ANVESH PAIDY
(112675567)

DESCRIPTION:

The dataset used for the project is Hospital Inpatient Discharge record for the year 2017 in New York. The data consists of 31 attributes such as Hospital name, Hospital county, Total costs, CCS Diagnosis Description, Length of stay, APR DRG Code etc. All the attributes of type Integer and float are plotted using the histogram and the categorical attributes are plotted by bar chart graph. The range of values associated to each attribute are also diverse. For instance categorical attributes have more than 5 types of values.

CODE:

The SVG is defined as below. Using the width and height, the x and y-axis are defined where x-axis denote the attribute and the y-axis denote the number of items.

```
d3.select("svg").remove();
var svg = d3.select("#barchart")
    .append("svg")
    .attr("width", width+100)
    .attr("height", height+100)
    .style("background", "azure")
//scales
var xscale = d3.scaleBand()
    .domain(xam)
    .range([0,width]).padding(0.3);

var yscale = d3.scaleLinear()
    .domain([0, d3.max(data)])
    .range([height, 50]);

var x_axis=d3.axisBottom().scale(xscale);
var y_axis = d3.axisLeft()
    .scale(yscale);
```

For the Bar chart I have used scaleBand and for the Y-axis I have used linear scale. The domain of the x-axis scale is all the categorical attributes in the code and y-axis is the number of the items (0- max).

Binning:

For the histogram I have used Linear Scale for both X-axis and Y-axis. I have divided a range of the values into bins, and y-axis determine the number of values present in the bin. Sample code used for dividing the data into bins for histogram is:

```
var histogram = d3.histogram()
    .value(function(d) { return d })
    .domain(xscale.domain())
    .thresholds(xscale.ticks(bin));
```

```
// And apply this function to data to get the bins
var bins = histogram(data);
```

Plotting the values:

For plotting the bars (rectangle) I have used the following code snippet where in the position of X attribute and the Y attribute are computed dynamically after the axis transformation.

```
svg.selectAll("rect")
  .data(bins)
  .enter()
  .append("rect")
  .style('fill',"steelblue")
  .attr("x",function(d) {
    return xscale(d.x0)+52 })
  .attr("y",function(d) { return yscale(d.length); })
  .attr("width",function(d) { return xscale(d.x1) - xscale(d.x0)-1 ; })
  .attr("height",function(d) { return height - yscale(d.length); })
```

Dropdown menu to update the bar chart:

For Updating the chart based on the change in the dropdown value, I have used the event handler which checks the type of attribute directly based on the value present and plot the graph accordingly,

```
d3.select('select')
  .on('change', function() {
    myfunc();
  });
```

```
function myfunc(){
  d3.csv("hospital_impatient_1000.csv",function(error,data){
    if(error){
      alert("failed loading data");
    }
    var attributesList= Object.getOwnPropertyNames(data[0]);
    loadOptions(attributesList);
    var attr_index=d3.select('#d3-dropdown').property("value");
    if(attr_index==undefined)
      attr_index=4;
    if(isNaN(Number(data[0][attributesList[attr_index]]))){
      formatted_data=prepareData_barchart(data,attributesList[attr_index]);
      loadBarChart(formatted_data,attributesList[attr_index]);
    }
  })
}
```

```

        else{
            formatted_data=prepareData_histogram(data,attributesList[attr_index])
;
            hist_data=formatted_data
            hist_attr=attributesList[attr_index]
            bin=20
            loadhistogram(formatted_data,attributesList[attr_index],20);
        }
    })
}

```

Handling the Events:

For the (4-6) I have used the event handlers to dynamically update the chart based the user actions.

- a) **mouseover, mouseout events:** for the tasks 4,5 (on mouse-over display the value of the bar on top of the bar on mouse-over also make the bar wider and higher to focus on it) .On mouseover the bar I have change the color of the selected bar and made it wider and displayed its value at the top.The code is as follows:

```

        .on("mouseover", function(d,i){
            d3.select(this)
            .style('fill',"navy")
            .attr("width",(xscale.bandwidth()+5))
            .attr("height",function(d) { return height - yscale(d.value)+1;
        });

            svg.append("text")
                .attr("y", d3.select(this).attr("y")-10 )
                .attr("x", d3.select(this).attr("x") )
                .attr("text-anchor", "start")
                .attr("id","val")
                .text(d.value);

        })
        .on("mouseout", function(d,i){
            d3.select(this).style('fill',"steelblue")
            .style('fill',"steelblue")
            .attr("width",xscale.bandwidth())
            .attr("height",function(d) { return height - yscale(d.value); })
;
            svg.select("#val").remove();
        });
    });

```

- b) mousedown and mouseup event:** for the task 6, mouse (with left mouse button down) move left (right) should decrease (increase) bin width/size (for numerical variables only), I have used the following snippet of code. These event handlers captures the coordinates when the mouse left button in clicked and releases in order to compute the bin size for the histogram plot.

```
svg.on("mousedown", function() {  
    if(d3.event.button==0){  
        var mp=d3.mouse(this);  
        oldx=mp[0]  
    }  
})  
  
svg.on("mouseup",function() {  
    if(d3.event.button==0){  
        var mp=d3.mouse(this);  
        pageX=mp[0];  
        bin+=(oldx-pageX)/3;  
        loadhistogram(hist_data,hist_attr,bin);  
        oldx = pageX;  
        console.log(pageX)  
    }  
})
```

Youtube link : <https://www.youtube.com/watch?v=bNw0aYDUc5E>