CSE 564 – Visualization Mini Project 1 – Report

Content

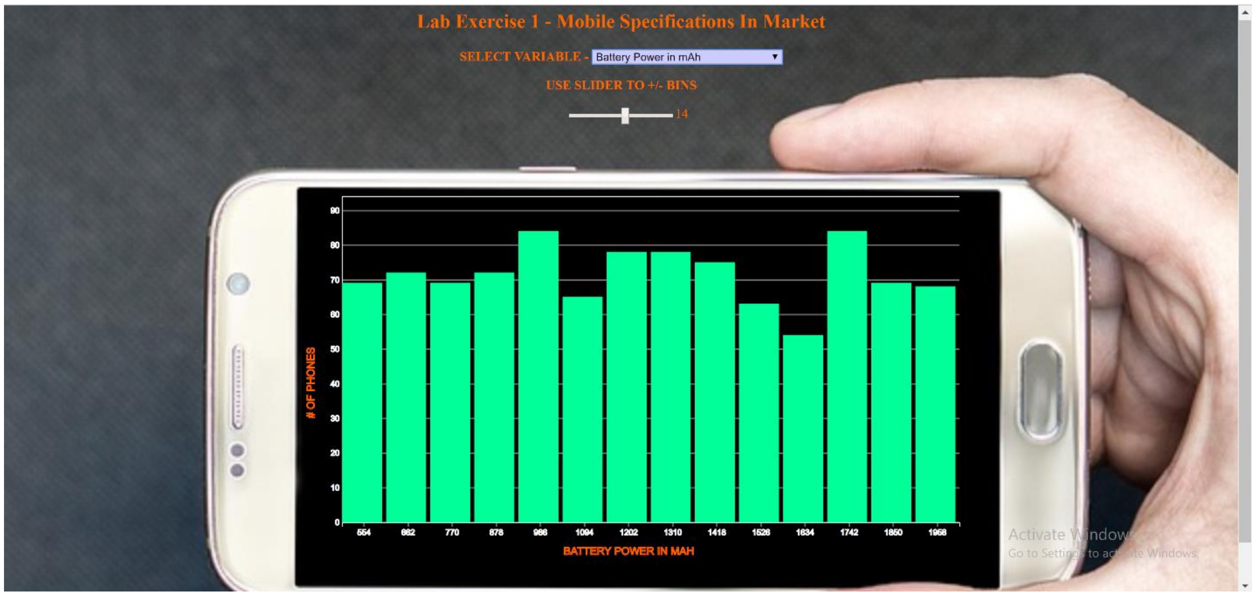
<u>Source CSV Data information</u> –

- The data set I have taken is from Kaggle and has 1000 data points and 21 dimensions. I have removed the (ID) Serial Number) column as it does not add value. So 1000 data points and 20 dimensions in total.
- The data set contains information about the various specification details like RAM, number of processor cores etc. of mobiles in the market.

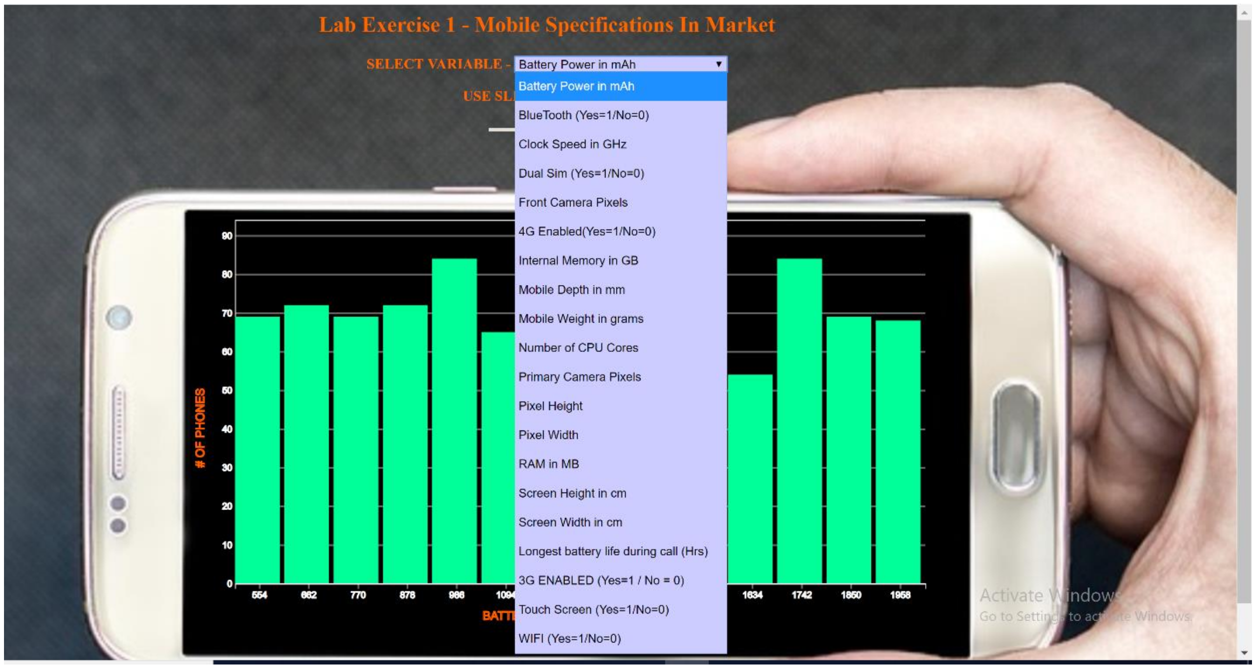
<u>Project capabilities</u> –

1. The webpage initially displays a title followed by a SELECT dropdown that populates values based on the attributes present in the data set. Bar chart updates based on value chosen.
2. A slider has been provided that will enable the user to increase or decrease the bins on slider movement. Default value of bins is set as 14 with minimum and maximum value being 2 and 24.
3. On page load, a bar chart initially loads with default select option "Battery power in mAh" with axes and axes labels.

Fig(1) and Fig(2) shows the above capabilities

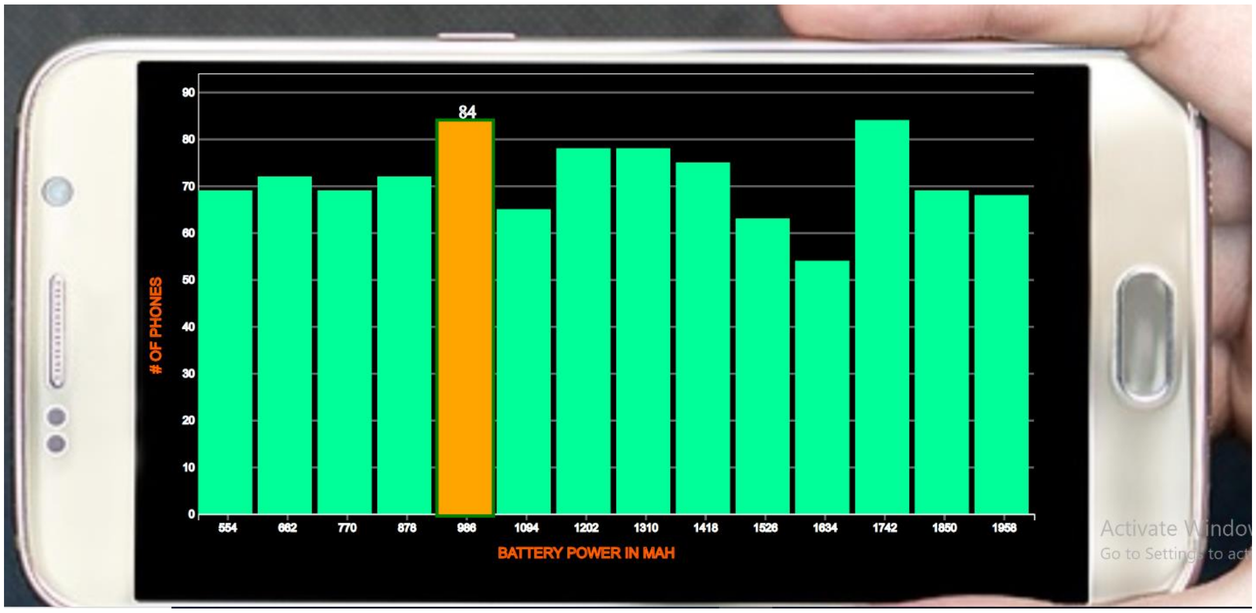


Fig(1)

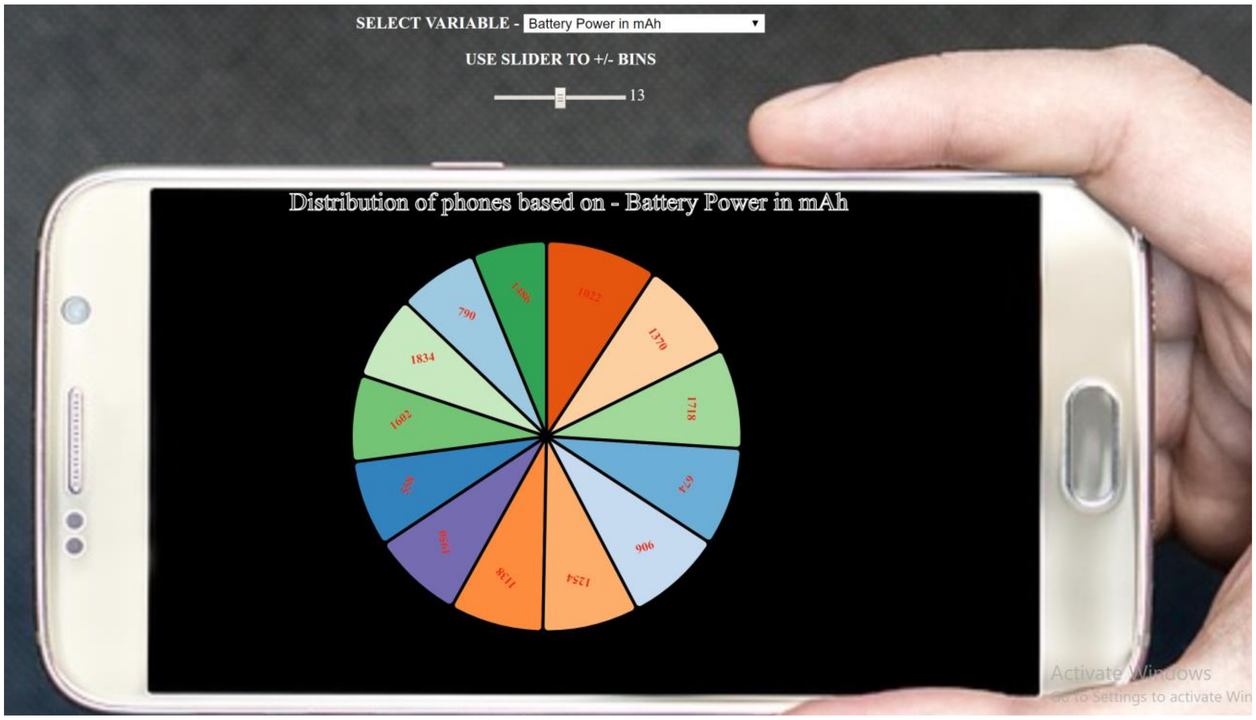


Fig(2)

4. On mouse hover of any bar, the width and height of the bar increases by a small offset and stroke along the border of the bar has been provided to create an effect of focus. The number of phones In that particular range(bin) is also populated on top of the bar.



5. On mouse out, the style settings are cleared, and the data is also removed, and the chart comes back to the normal state.
6. On click (anywhere on group element that holds bars) the bar chart disappears, and the corresponding pie-chart pops up with bin values displayed as labels for each pie.

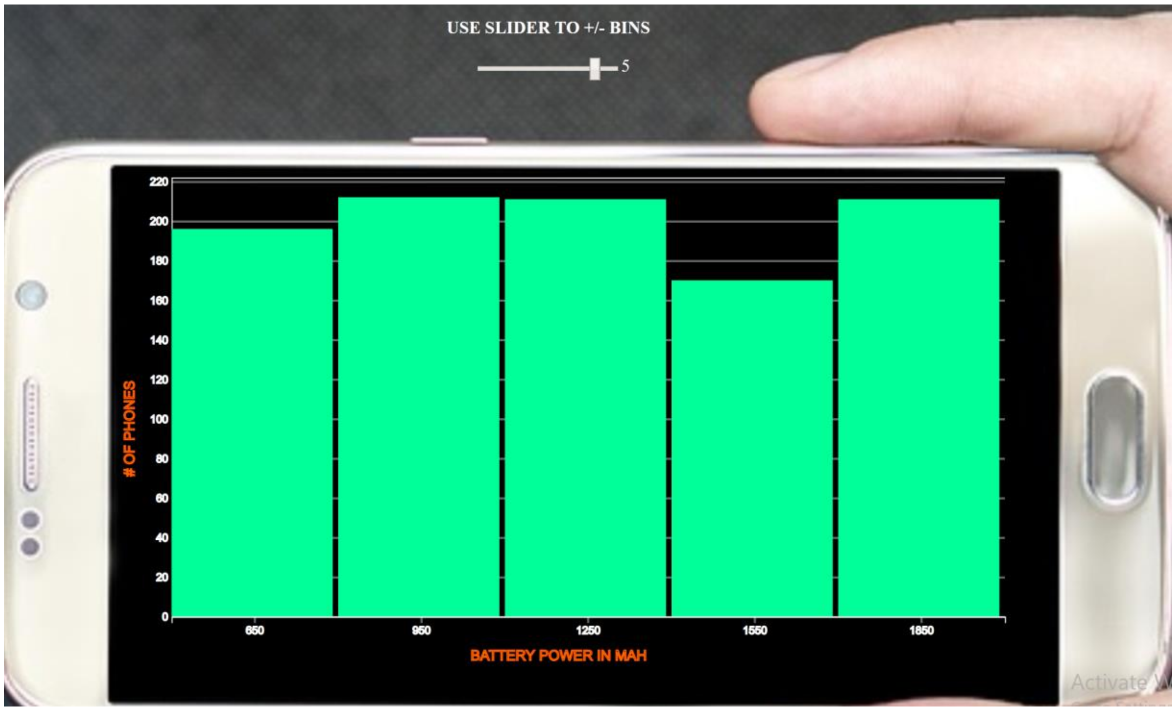

SELECT VARIABLE - Battery Power in mAh

USE SLIDER TO +/- BINS

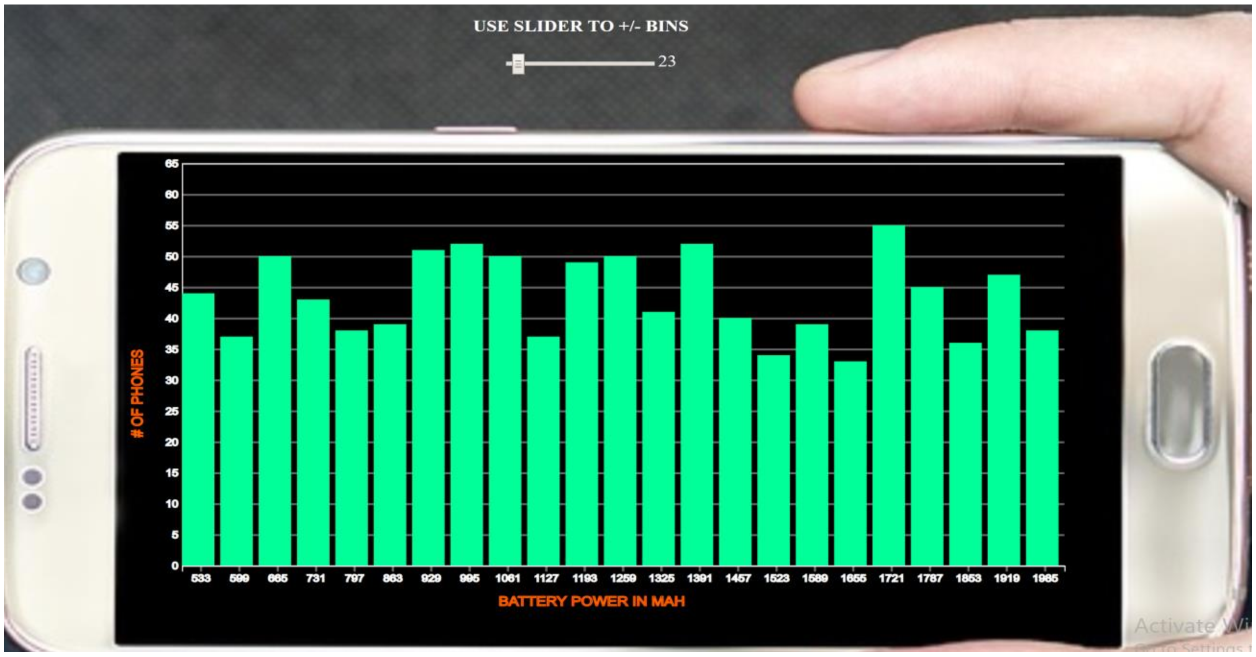
13

Distribution of phones based on - Battery Power in mAh

7. On click, the pie chart disappears, and the bar chart loads back.
8. On moving the mouse right on slider, the bin width/size increases leading to re-binning and recalculation of axes points.

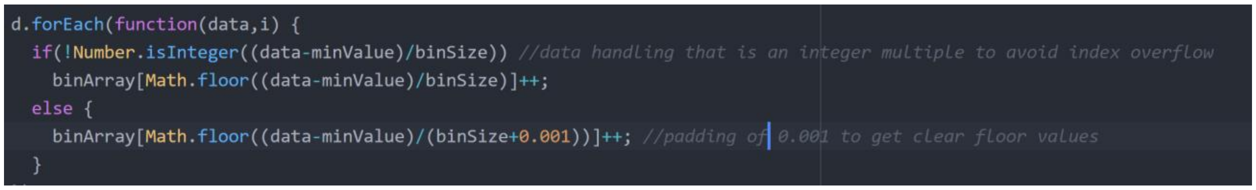


9. On moving the mouse left on slider, the bin width/size decreases leading to to re-binning and recalculation of axes points.



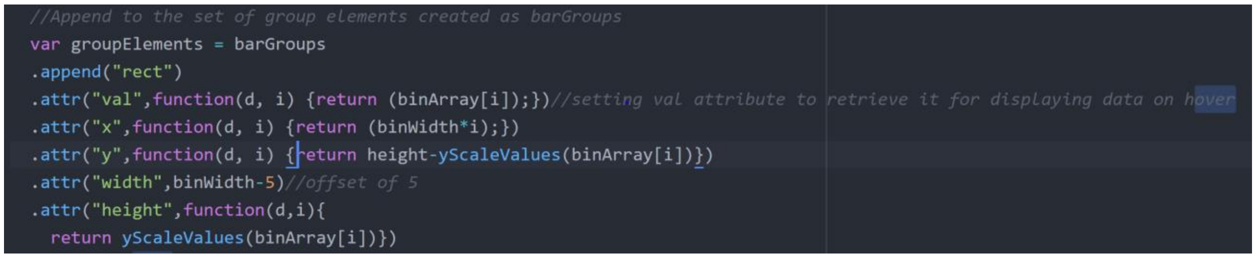
Code Snippets for each of the project capabilities

1) Pick a variable and bin it to fixed range (equi-width) of your choice.



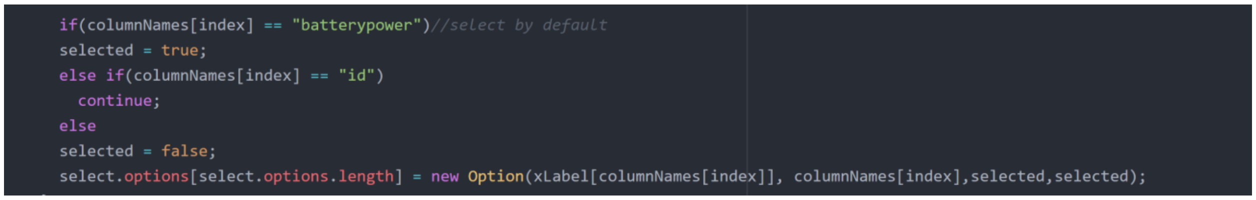
```
d.forEach(function(data,i) {
  if(!Number.isInteger((data-minValue)/binSize)) //data handling that is an integer multiple to avoid index overflow
    binArray[Math.floor((data-minValue)/binSize)]++;
  else {
    binArray[Math.floor((data-minValue)/(binSize+0.001))]++; //padding of 0.001 to get clear floor values
  }
```

2) Create a bar chart of the variable you picked in 1.



```
  //Append to the set of group elements created as barGroups
  var groupElements = barGroups
  .append("rect")
  .attr("val",function(d, i) {return (binArray[i]);})//setting val attribute to retrieve it for displaying data on hover
  .attr("x",function(d, i) {return (binWidth*i);})
  .attr("y",function(d, i) {return height-yScaleValues(binArray[i])})
  .attr("width",binWidth-5)//offset of 5
  .attr("height",function(d,i){
    return yScaleValues(binArray[i])})
```

3) Using a menu allow users to select a new variable and update chart

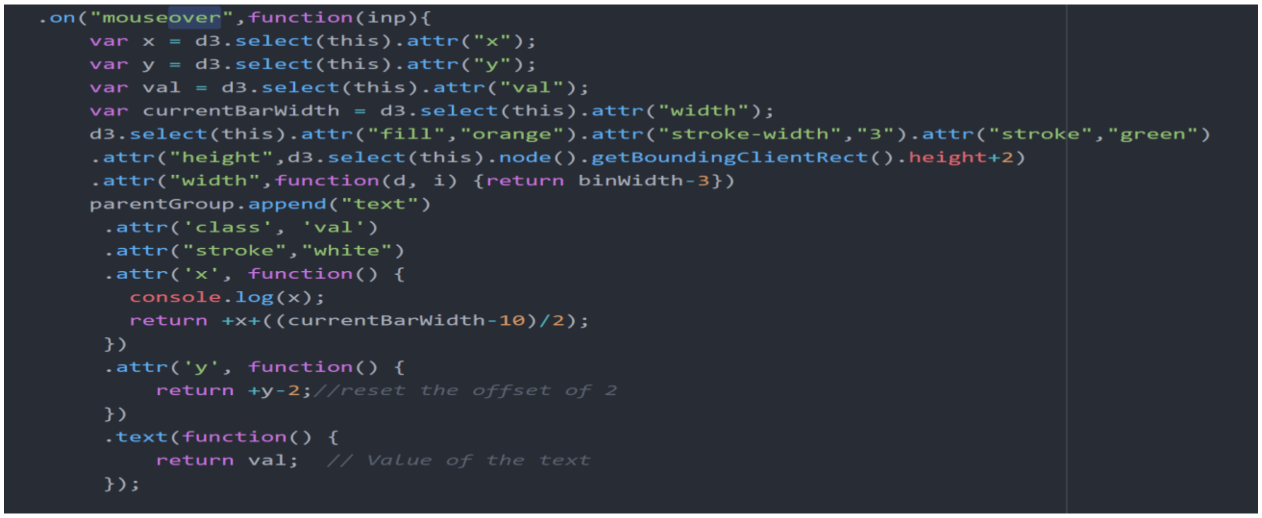


```
    if(columnNames[index] == "batterypower")//select by default
    selected = true;
    else if(columnNames[index] == "id")
      continue;
    else
    selected = false;
    select.options[select.options.length] = new Option(xLabel[columnNames[index]], columnNames[index],selected,selected);
```

4) Only on mouse-over display the value of the bar on top of the bar

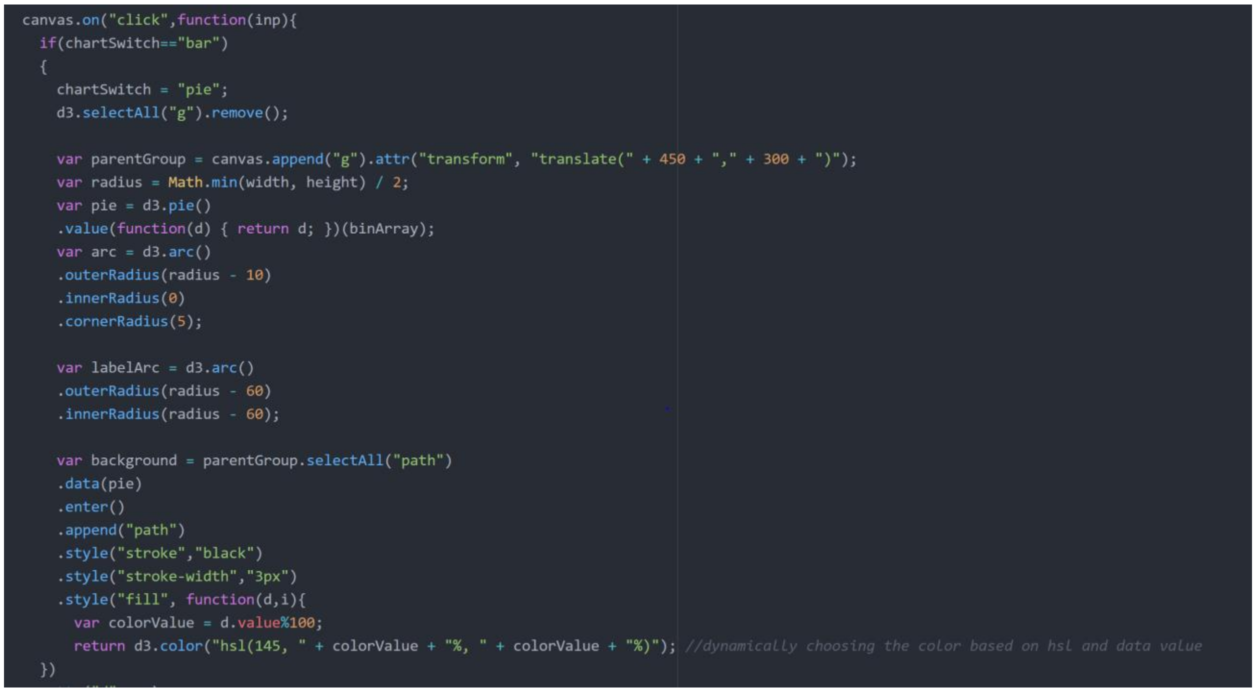
AND

5) On mouse-over make the bar wider and higher to focus on it



```
.on("mouseover",function(inp){
    var x = d3.select(this).attr("x");
    var y = d3.select(this).attr("y");
    var val = d3.select(this).attr("val");
    var currentBarWidth = d3.select(this).attr("width");
    d3.select(this).attr("fill","orange").attr("stroke-width","3").attr("stroke","green")
    .attr("height",d3.select(this).node().getBoundingClientRect().height+2)
    .attr("width",function(d, i) {return binWidth-3})
    parentGroup.append("text")
     .attr('class', 'val')
     .attr("stroke","white")
     .attr('x', function() {
       console.log(x);
       return +x+((currentBarWidth-10)/2);
     })
     .attr('y', function() {
         return +y-2;//reset the offset of 2
     })
     .text(function() {
         return val;  // Value of the text
     });
```

6) On mouse-click transform the bar to pie chart



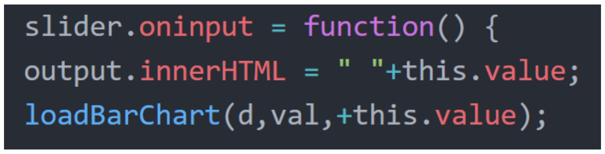
```
canvas.on("click",function(inp){
  if(chartSwitch=="bar")
  {
    chartSwitch = "pie";
    d3.selectAll("g").remove();

    var parentGroup = canvas.append("g").attr("transform", "translate(" + 450 + "," + 300 + ")");
    var radius = Math.min(width, height) / 2;
    var pie = d3.pie()
    .value(function(d) { return d; })(binArray);
    var arc = d3.arc()
    .outerRadius(radius - 10)
    .innerRadius(0)
    .cornerRadius(5);

    var labelArc = d3.arc()
    .outerRadius(radius - 60)
    .innerRadius(radius - 60);

    var background = parentGroup.selectAll("path")
    .data(pie)
    .enter()
    .append("path")
    .style("stroke","black")
    .style("stroke-width","3px")
    .style("fill", function(d,i){
      var colorValue = d.value%100;
      return d3.color("hsl(145, " + colorValue + "%, " + colorValue + "%)"); //dynamically choosing the color based on hsl and data value
  })
```

7) Mouse moves left/right should decrease/increase bin width/size



```
slider.oninput = function() {
output.innerHTML = " "+this.value;
loadBarChart(d,val,+this.value);
```

<u>Pseudocode</u> –

<u>ENTRY POINT</u>

```
INITIALIZE VARIABLES
CREATE place holder using the SVG element
LOAD DATA from CSV
    - CURATE data type from string to number
CALL CALLBACK routine with data just loaded
    - LOADCOLUMNS() in SELECT dropdown
    - LOADBARCHART(data,dropDownValue,sliderValue) loads bar
      chart for default value selected
    - ADD ON CHANGE event listener
        o LOAD BAR CHART on change of select value
```

<u>LOADBARCHART() Routine –</u>

```
SET numBins = slider value
SET binWidth = width/numBins
SET currentMode = barChart
RESET other variables
LOAD binArray that holds range of values
LOAD xRanges array that holds range of x axis values
DEFINE X,Y Axis
CREATE bars based on data
```

ADD MOUSE-OVER and MOUSE-OUT handlers

ADD MOUSE CLICK handlers to load PIE CHART when currentMode = barChart

ON CHANGE of slider input or ON CLICK of canvas when mode=PIECHART

- CALL LOADBARCHART(data,dropDownValue,sliderValue)

<u>LOADCOLUMNS() ROUTINE</u>

CREATE new OPTIONS for INPUT data