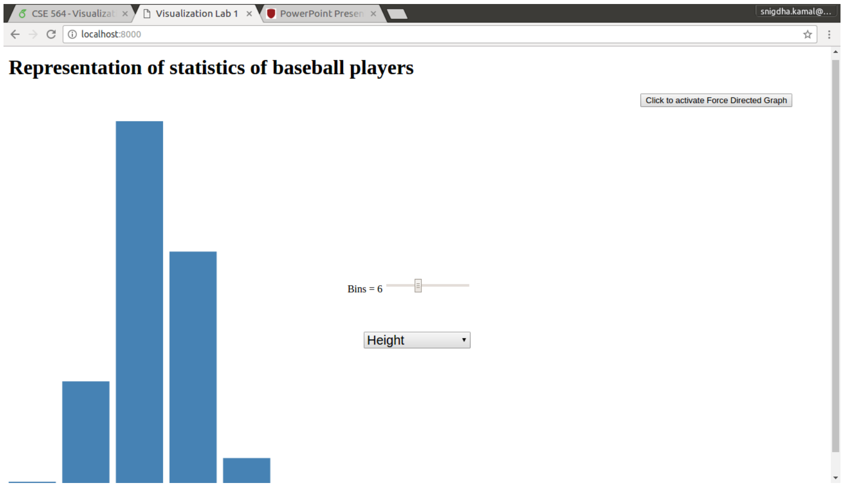
CSE 564 - Visualization: Lab Report 1

February 18, 2017

1 Introduction

In this assignment, I have plotted the statistical information of baseball players (height, weight, batting average and number of home runs) by binning them into discrete bins and incorporating interactive elements. The data for this lab is present in baseball.csv file in the submission folder and has over 1158 data points. The graph also transforms to pie chart and force-directed graph on-click. Figure 1 shows the front page of the project.



Representation of statistics of baseball players

Bins = 6

Height

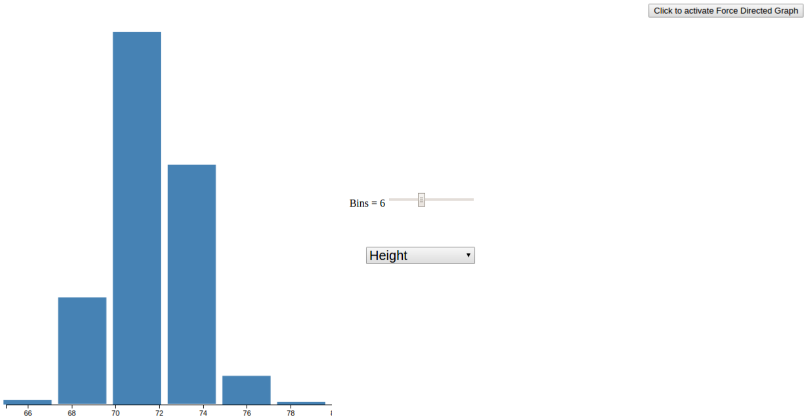
Figure 1: The user interface

1.1 Question 1

A variable(height) was picked and initially divided into 6 bins as shown in Figure 2

1.2 Question 2

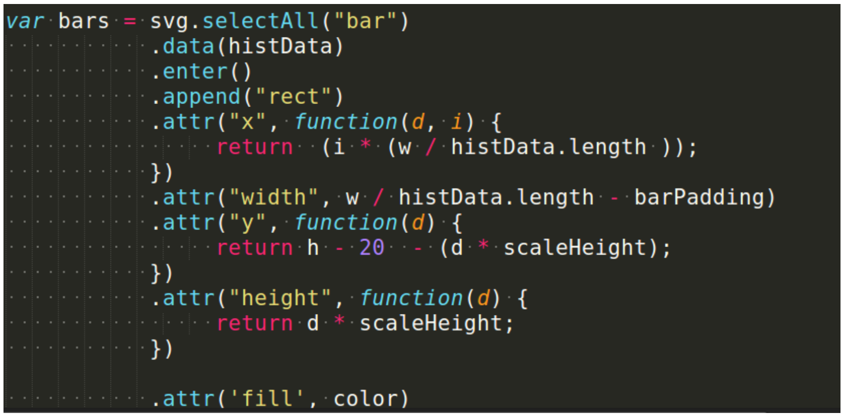
A bar chart was created using the chosen variable as shown in Figure 3



Bins = 6

Height

Figure 2: Variable height binned into 6 bins



```
var bars = svg.selectAll("bar")
           .data(histData)
           .enter()
           .append("rect")
           .attr("x", function(d, i) {
                return  (i * (w / histData.length ));
           })
           .attr("width", w / histData.length - barPadding)
           .attr("y", function(d) {
                return h - 20  - (d * scaleHeight);
           })
           .attr("height", function(d) {
                return d * scaleHeight;
           })

           .attr('fill', color)
```

Figure 3: Source code for Bar chart element

1.3 Question 3

Users can choose a new variable according to a drop-down menu and the bar-chart changes accordingly to reflect the new values as shown in Figure 4

1.4 Question 4

The value of the bar gets displayed on mouse-over. I have used d3 tooltip library for this purpose shown in Figure 5. Figure 6 demonstrates how to initialize the tip variable and pass it the data it needs. This variable is then called by the SVG later.

1.5 Question 5

The bar becomes highlighted by becoming wider and higher on mouse-over. It also changes colour as an added visual aid shown in Figure 7. Figure 8 demonstrates the code written to make the bar higher and wider to focus on while also changing its colour.

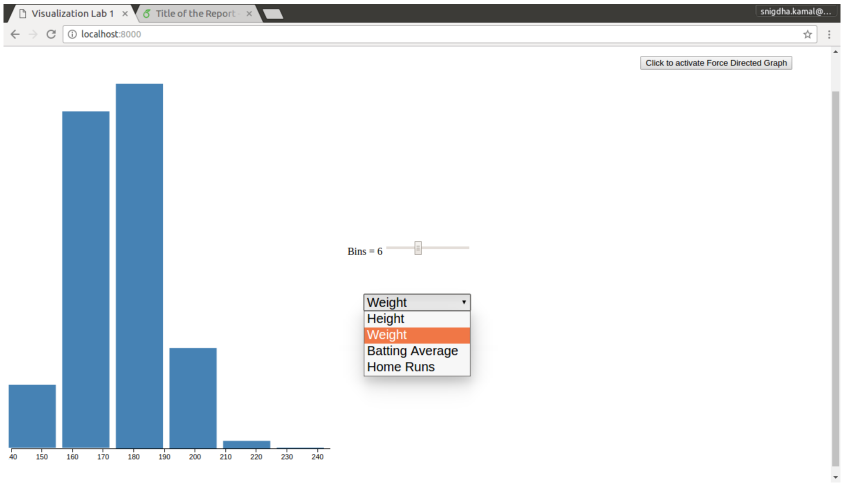


Figure 4: Graph changes with drop down menu

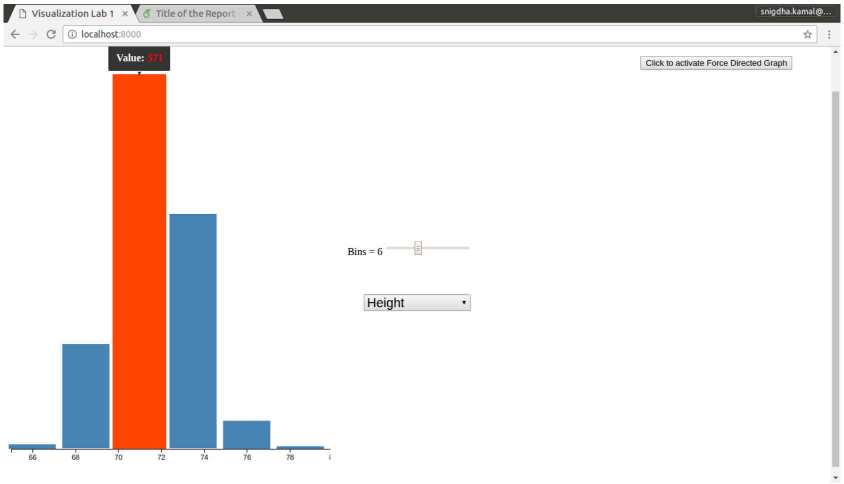


Figure 5: Value gets displayed on top of the bar

1.6 Question 6

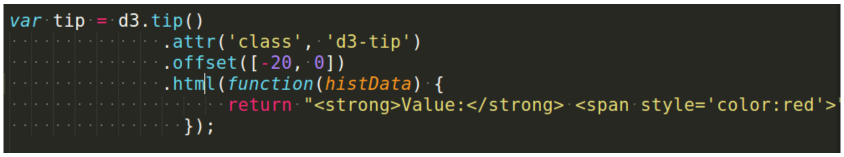
The bar chart gets transformed into a pie chart on mouse-click. It transforms back into a bar graph when clicked again shown in 9

Figure 10 shows the source-code for generating the piechart element

Figure 11 shows the code needed to transform the bar chart into a pie chart. The CreatePiechart function is called which defines the parameters of the pie-chart in place of the bar chart

1.7 Question 7

A slider has been implemented which when moved left decreases the number of bins and when moved right increases the number of bins in the graph shown in 12



```
var tip = d3.tip()
              .attr('class', 'd3-tip')
              .offset([-20, 0])
              .html(function(histData) {
                    return "<strong>Value:</strong> <span style='color:red'>'
                }); 
```

Figure 6: Declaring variable tip of tooltip library

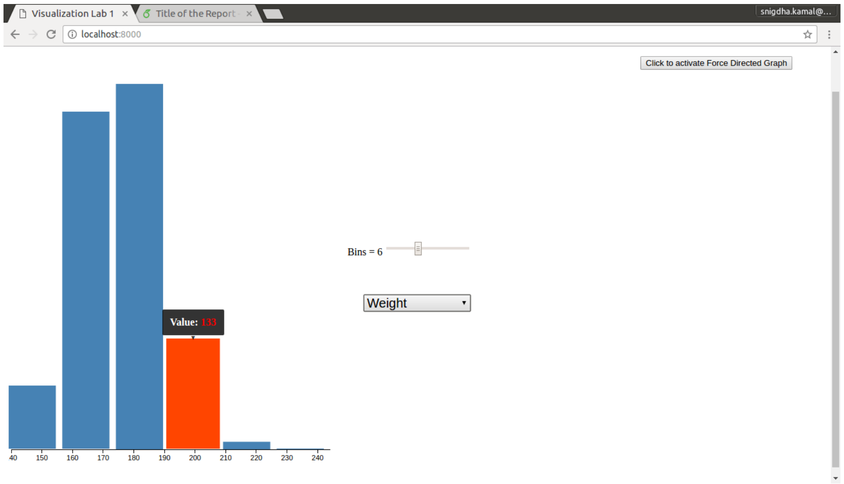
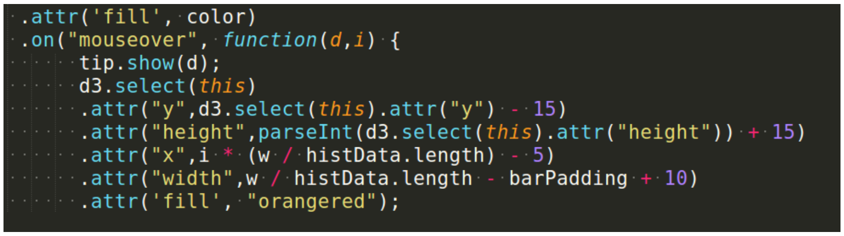


Figure 7: Bar becomes highlighted and changes colour

1.8 Question 8

The chart transforms into a force directed graph when the button is clicked shown in 13 Figure 14 shows the source-code for creating the force-directed element



```
 .attr('fill', color)
 .on("mouseover", function(d,i) {
      tip.show(d);
      d3.select(this)
      .attr("y",d3.select(this).attr("y") - 15)
      .attr("height",parseInt(d3.select(this).attr("height")) + 15)
      .attr("x",i * (w / histData.length) - 5)
      .attr("width",w / histData.length - barPadding + 10)
      .attr('fill', "orangered");
```

Figure 8: Bar becomes prominent on mouse-over

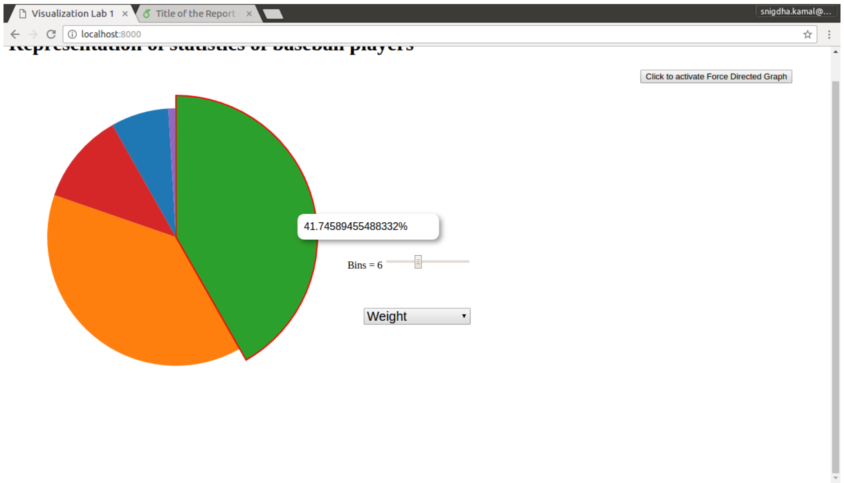
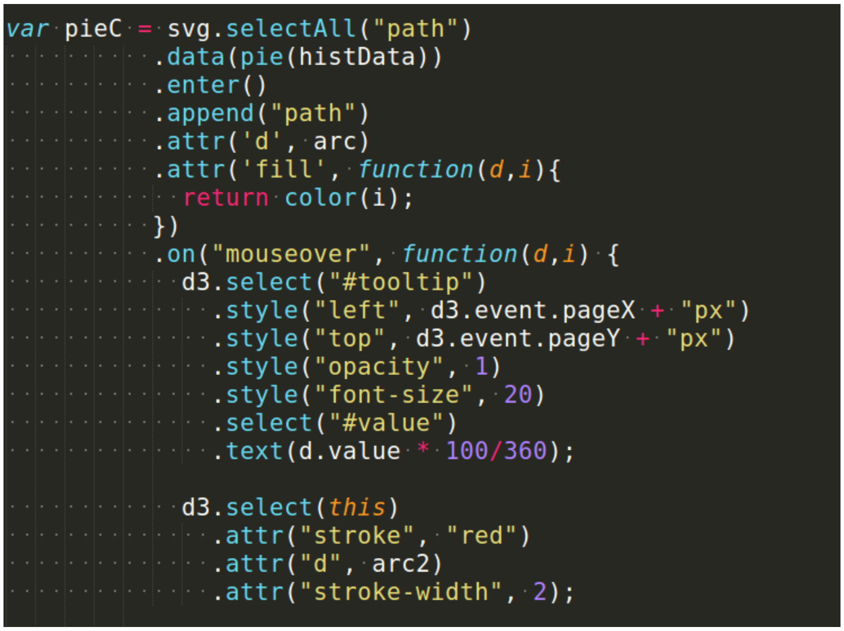


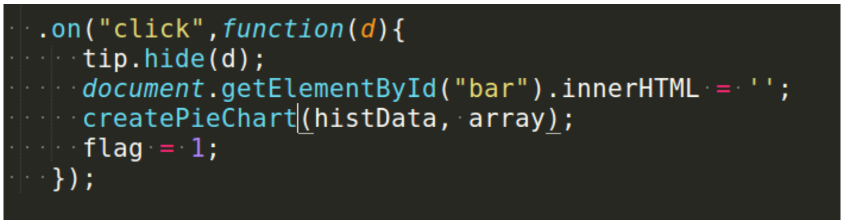
Figure 9: Bar chart transformed to Pie Chart



```
var pieC = svg.selectAll("path")
          .data(pie(histData))
          .enter()
          .append("path")
          .attr('d', arc)
          .attr('fill', function(d,i){
            return color(i);
          })
          .on("mouseover", function(d,i) {
            d3.select("#tooltip")
              .style("left", d3.event.pageX + "px")
              .style("top", d3.event.pageY + "px")
              .style("opacity", 1)
              .style("font-size", 20)
              .select("#value")
              .text(d.value * 100/360);

            d3.select(this)
              .attr("stroke", "red")
              .attr("d", arc2)
              .attr("stroke-width", 2);
```

Figure 10: Source code for pie chart



```
  .on("click",function(d){
     tip.hide(d);
     document.getElementById("bar").innerHTML = '';
     createPieChart(histData, array);
     flag = 1;
   });
```

Figure 11: Code to transform bar chart into pie chart

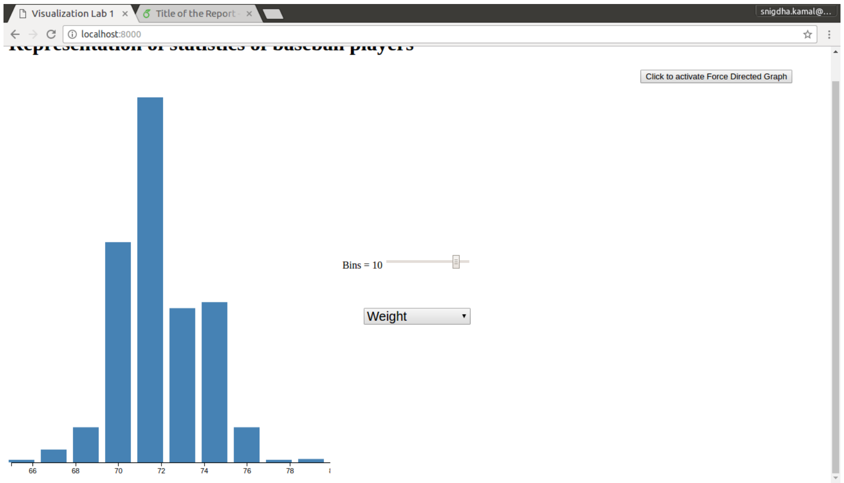


Figure 12: Slider used to change number of bins

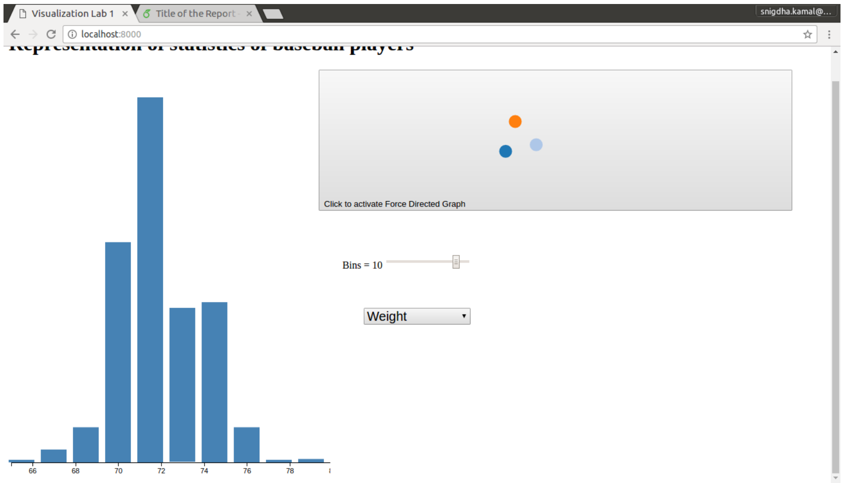
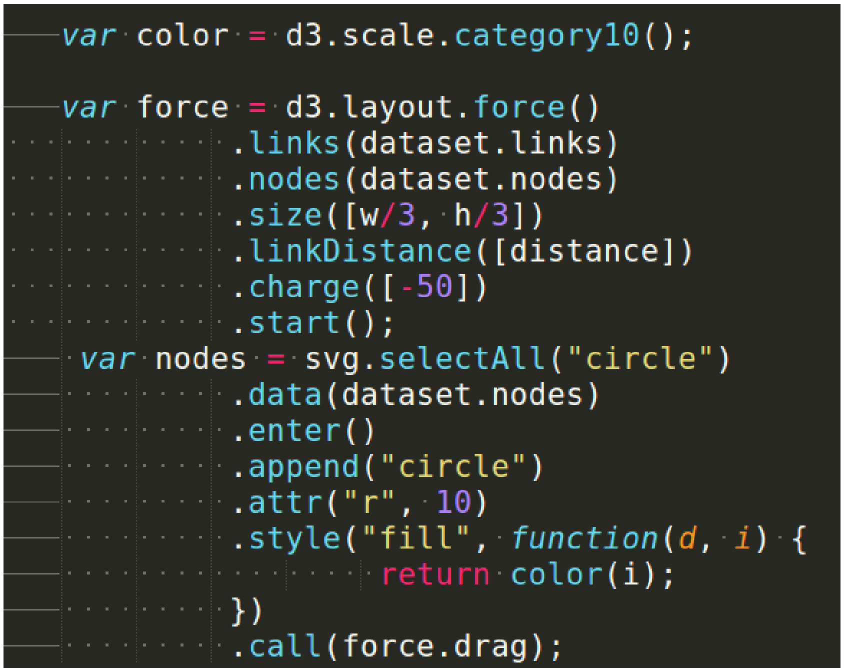


Figure 13: Force Directed Graph activated on button click



```
var color = d3.scale.category10();

var force = d3.layout.force()
            .links(dataset.links)
            .nodes(dataset.nodes)
            .size([w/3, h/3])
            .linkDistance([distance])
            .charge([-50])
            .start();
 var nodes = svg.selectAll("circle")
            .data(dataset.nodes)
            .enter()
            .append("circle")
            .attr("r", 10)
            .style("fill", function(d, i) {
                    return color(i);
            })
            .call(force.drag);
```

Figure 14: Source code for Force-directed element