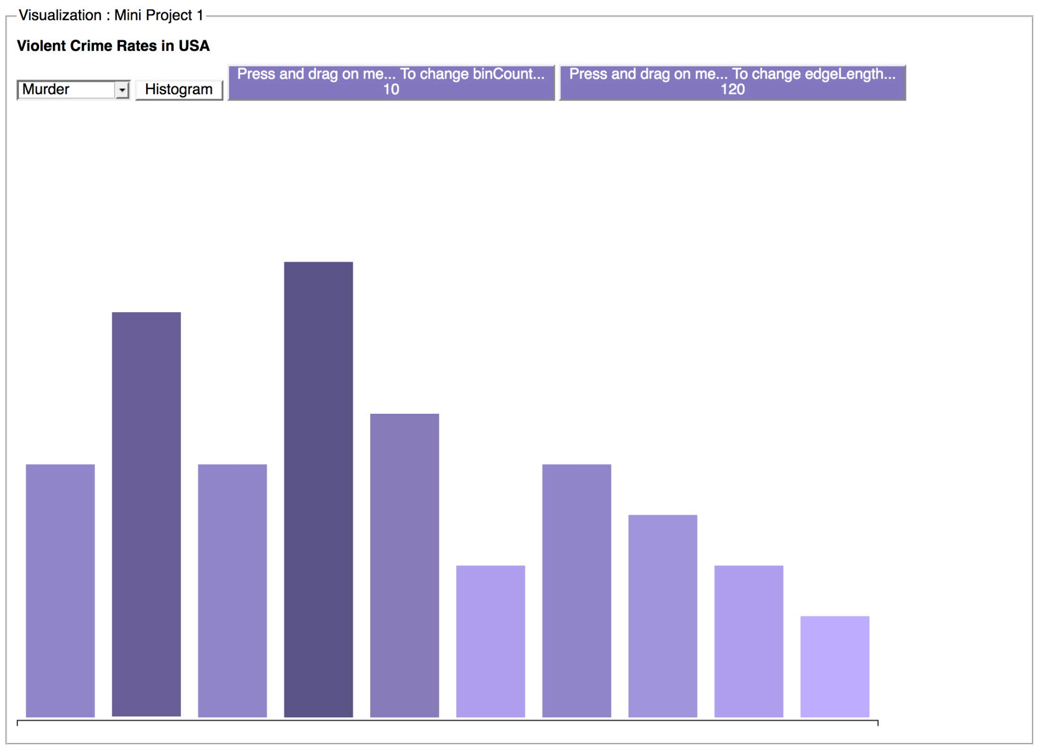
Prathak Rastogi
110951108
Visualization Mini Project -1

I have chosen dataset containing Violent Crime Rates by US State, where various variables are, Murder, Assault, Urban Pop and Rape.

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

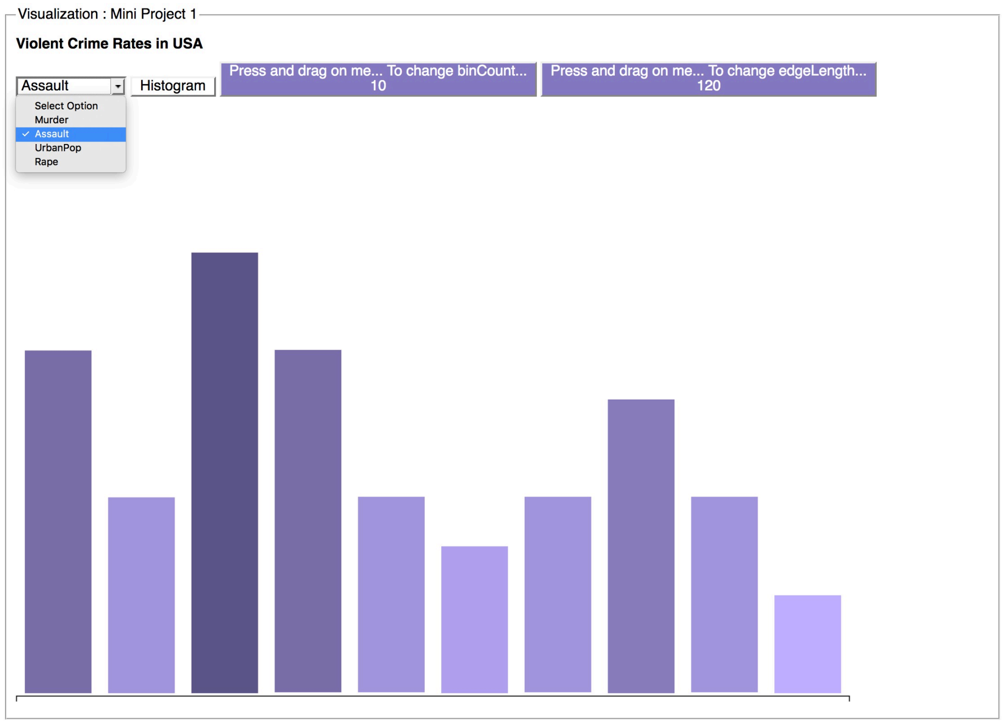
2. create a bar chart of the variable you picked in 1.


Visualization : Mini Project 1
Violent Crime Rates in USA
Murder
Histogram
Press and drag on me... To change binCount... 10
Press and drag on me... To change edgeLength... 120

Above is the Histogram showing Murder data where binCount is 10.

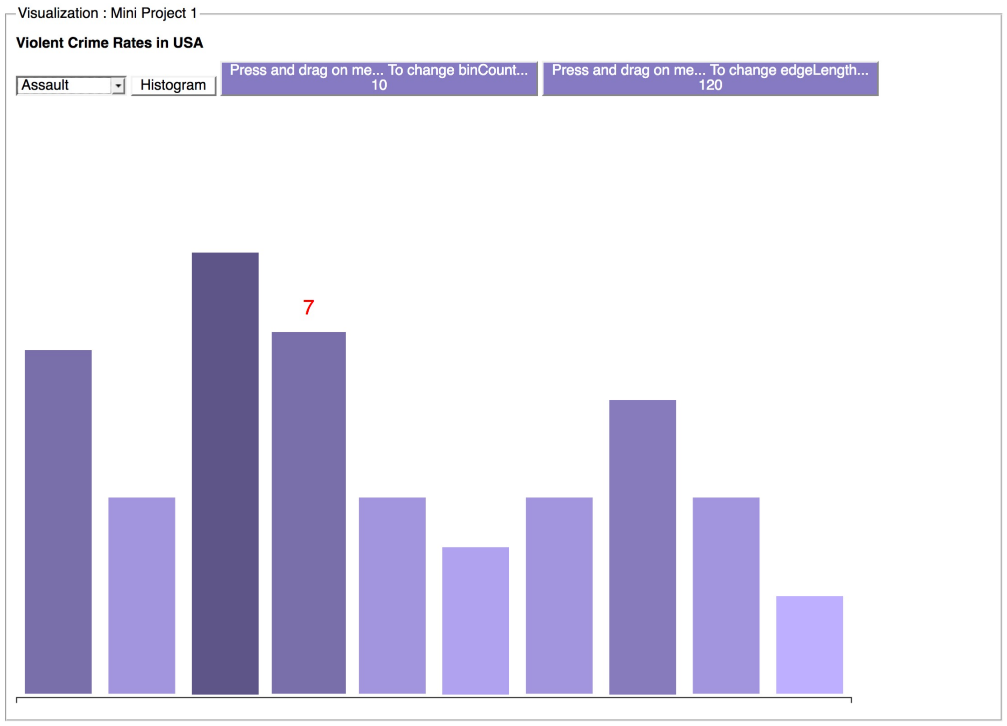
Edge Length is for Force Directed Graph.

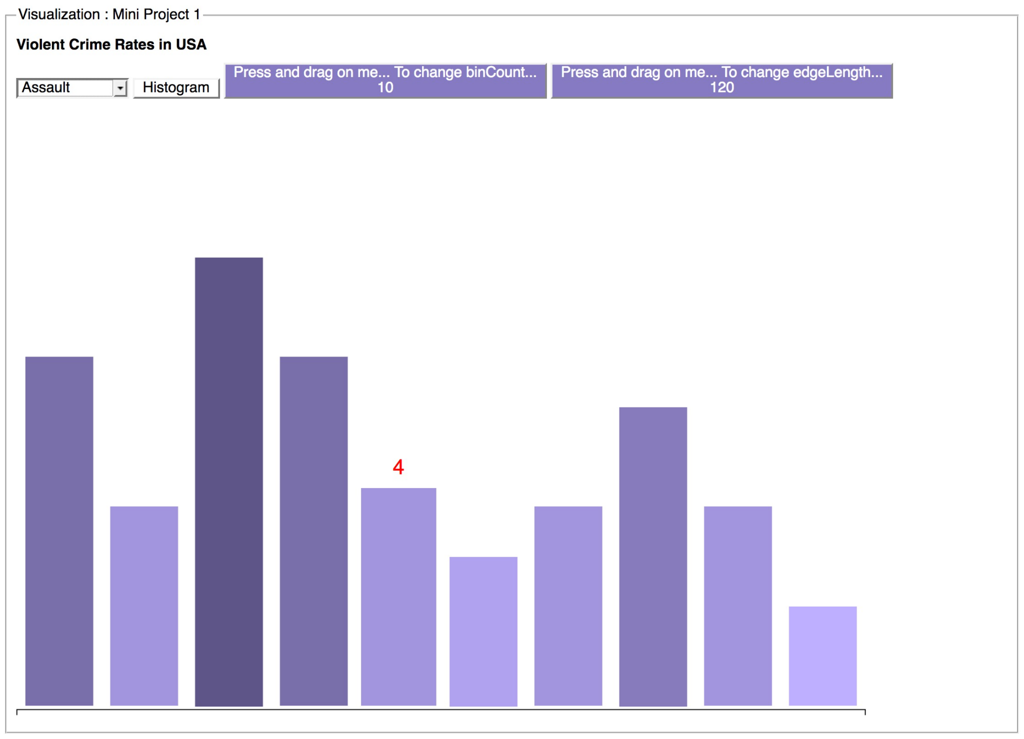
3. using a menu, allow users to select a new variable and update chart


Visualization : Mini Project 1
Violent Crime Rates in USA
Assault
Histogram
Press and drag on me... To change binCount... 10
Press and drag on me... To change edgeLength... 120
Select Option
Murder
✓ Assault
UrbanPop
Rape

User can select the variable from the drop-down menu as shown in the picture above.

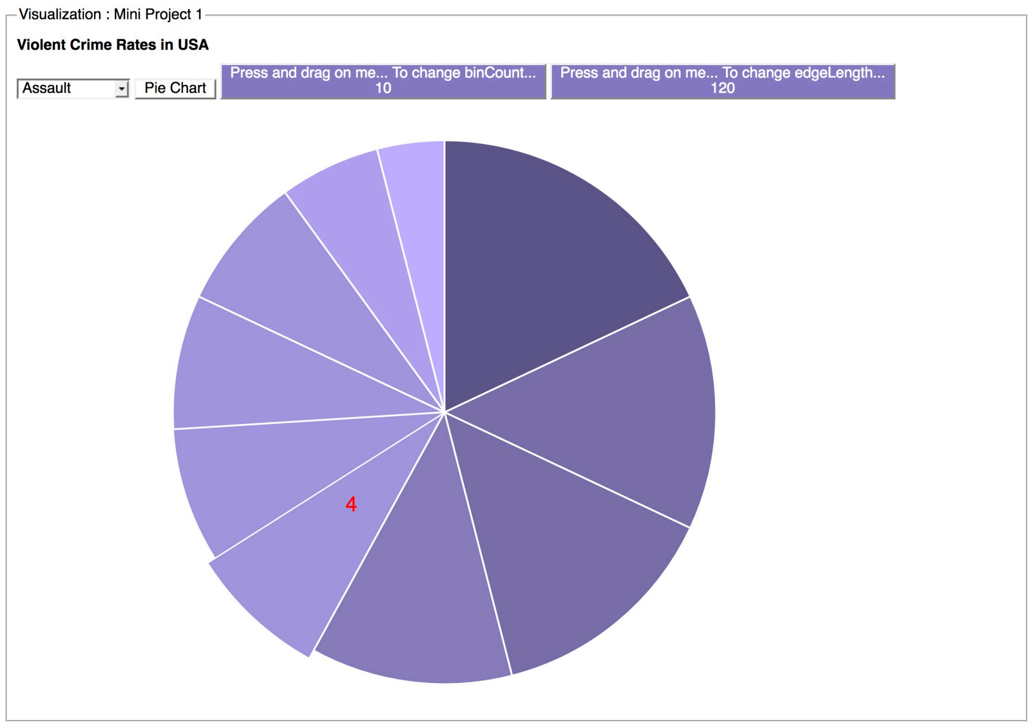
4. only on mouse-over display the value of the bar on top of the bar
5. on mouse-over make the bar wider and higher to focus on it


Visualization : Mini Project 1
Violent Crime Rates in USA
Assault
Histogram
Press and drag on me... To change binCount... 10
Press and drag on me... To change edgeLength... 120
7


Visualization : Mini Project 1
Violent Crime Rates in USA
Assault
Histogram
Press and drag on me... To change binCount... 10
Press and drag on me... To change edgeLength... 120
4

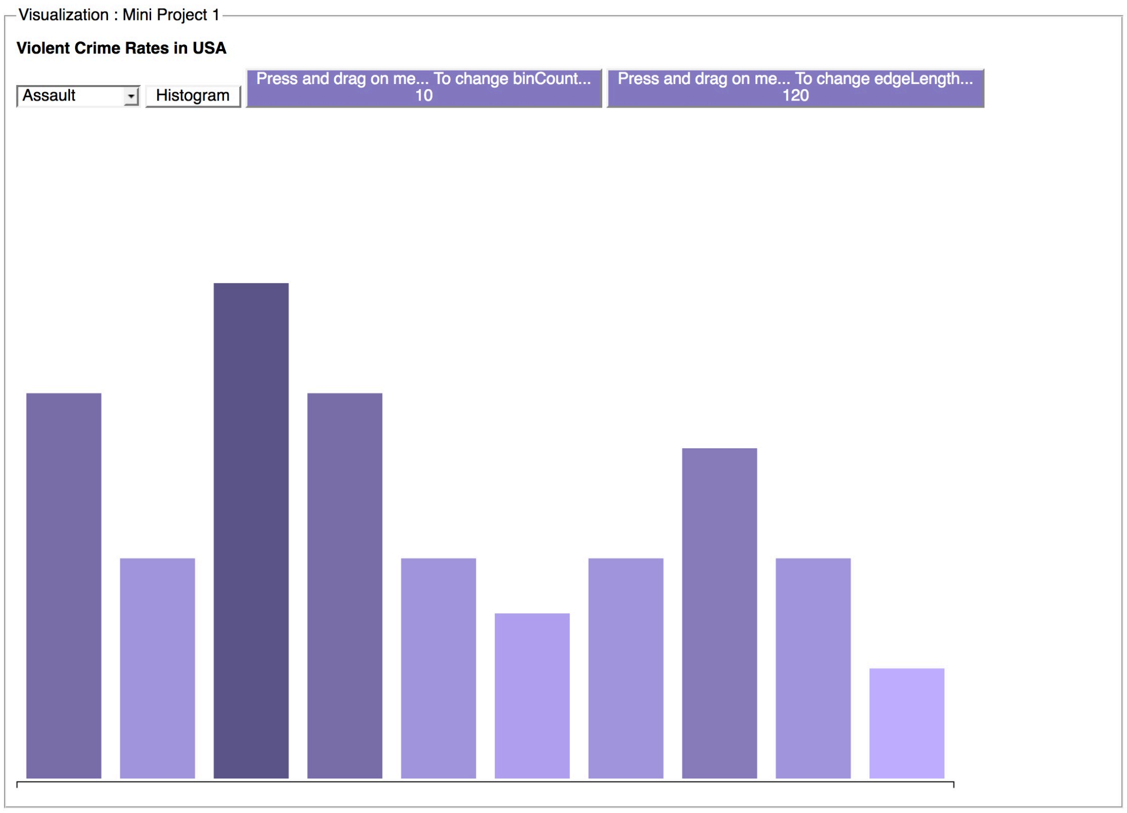
On mouse-over the height and the width of the bar changes and the value of that bar pops up. When mouse is removed the shape of the rectangle get reverted to its initial condition.

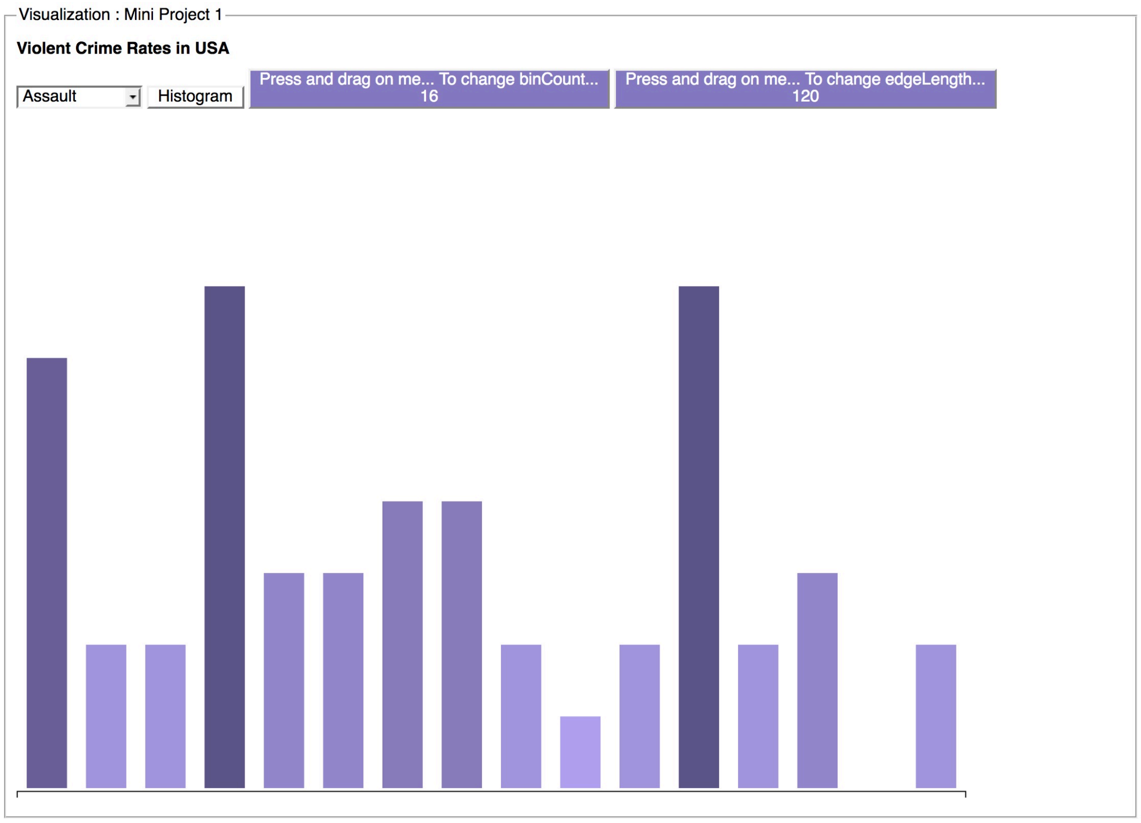
6. on mouse-click transform the bar chart into a pie chart (and back)


Visualization : Mini Project 1
Violent Crime Rates in USA
Assault
Pie Chart
Press and drag on me... To change binCount... 10
Press and drag on me... To change edgeLength... 120
4

On clicking the bar of the histogram, histogram gets converted to the pie chart, on clicking the pie chart it converted to the force directed and then clicking again on the node converts back to histogram.

7. mouse moves left (right) should decrease (increase) bin width/size

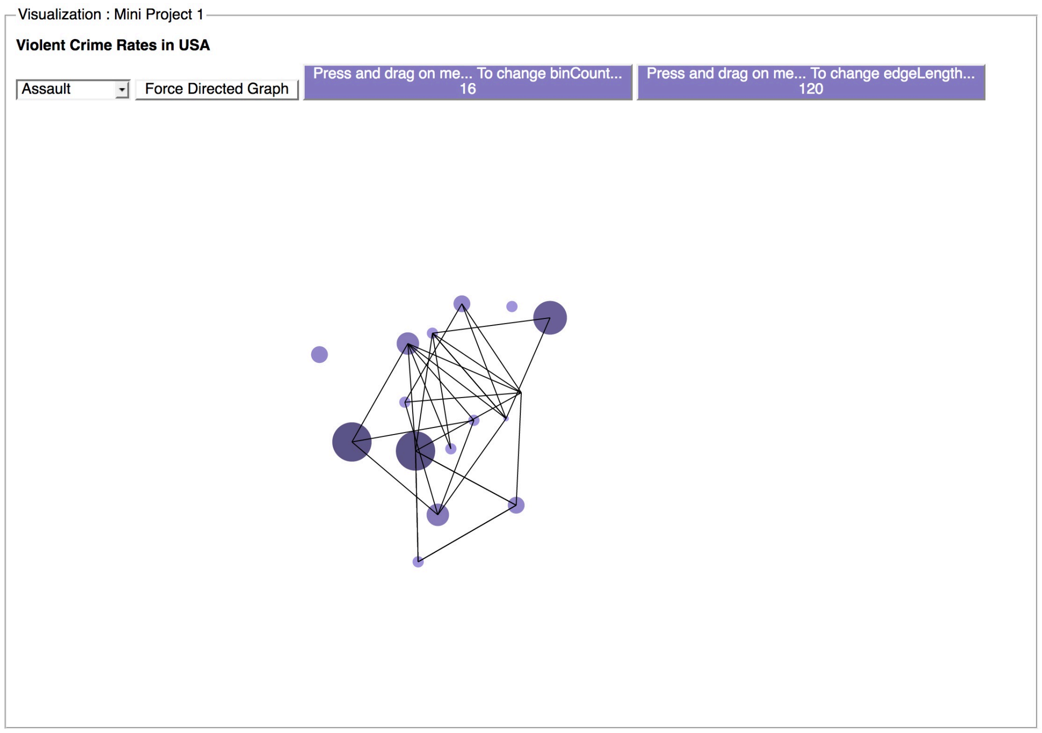

Visualization : Mini Project 1
Violent Crime Rates in USA
Assault
Histogram
Press and drag on me... To change binCount... 10
Press and drag on me... To change edgeLength... 120

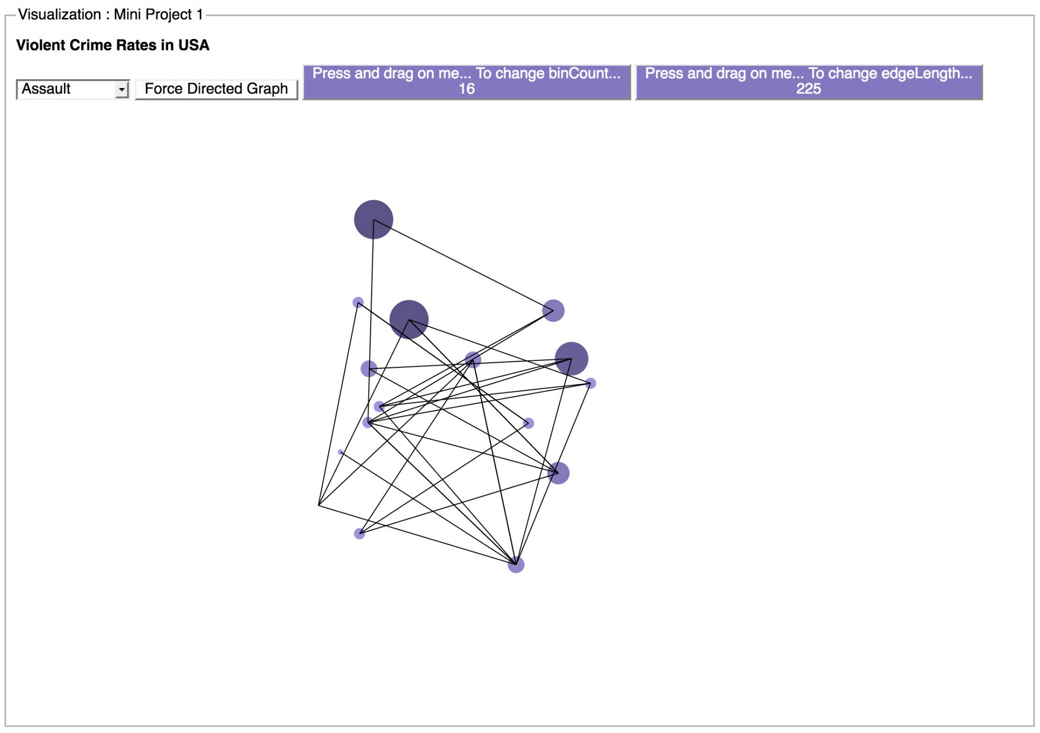

Visualization : Mini Project 1
Violent Crime Rates in USA
Assault
Histogram
Press and drag on me... To change binCount... 16
Press and drag on me... To change edgeLength... 120

Above we can see the change of the bin size according to the change in binCount by pressing the button and moving the move to right side, and decreasing the same by pressing the button and moving the mouse to left.

Extra Credit:

on mouse-click create a force-directed layout using a chosen distance


Visualization : Mini Project 1
Violent Crime Rates in USA
Assault
Force Directed Graph
Press and drag on me... To change binCount... 16
Press and drag on me... To change edgeLength... 120


Visualization : Mini Project 1
Violent Crime Rates in USA
Assault
Force Directed Graph
Press and drag on me... To change binCount... 16
Press and drag on me... To change edgeLength... 225

Above is the force directed graph generated by clicking the pie-chart. Edge length can be changed just by pressing the button for the same and moving the mouse left(decrease) and right(increase).

Code snippets are given below….

index.html

```
<!--
Visualization project 1..
Create histogram, piechart and force directed layout

Data : Violent Crime Rates by US State
-->

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Mini Project 1</title>

  <script type="text/javascript" src="lib/d3.v3.min.js"></script>

  <style type="text/css">
    legend,h1 {
      font-family: sans-serif;
      font-size: 16px;
    }
    .axis path, .axis line {
      fill: none;
      stroke: #000;
    }
    button {
      background: #857CC2;
      color: white;
      font-family: sans-serif;
      font-size: 16px;
    }
    #nameOfGraph,#propertydropdown {
      background: white;
      color: black;
      font-family: sans-serif;
      font-size: 16px;
    }
  </style>

  <script src="src/graphs.js"></script>
</head>

<body>
  <fieldset>
    <legend>Visualization : Mini Project 1</legend>

    <h1>Violent Crime Rates in USA</h1>

    <select onchange="chooseVariable(this.value,binCount)" id="propertydropdown" autocomplete="off">
      <option value="">Select Option</option>
      <option value="1">Murder</option>
      <option value="2">Assault</option>
      <option value="3">UrbanPop</option>
      <option value="4">Rape</option>
    </select>

    <button id="nameOfGraph"><div id="graphName">None</div></button>

    <button id = "binCountUpdate">Press and drag on me... To change binCount...<div
id="binsCount">10</div></button>
    <button id = "lengthUpdate">Press and drag on me... To change edgeLength...<div
id="edgeLength">120</div></button>
```

```
<div id = "bar" ></div>
<div id = "chart"></div>
<div id = "forceD"></div>

<script type="text/javascript">

  var yMax = 0;
  var yMin = 0;
  var color = "#857CC2";

  var binCount = 10;
  var edgeLength = 120;

  var w = window.innerWidth - 200;
  var h = window.innerHeight- 250;
  var barPadding = 0.02 * w;

  var currentFigure = 0;

  var array1 = [];
  var array2 = [];
  var array3 = [];
  var array4 = [];

  d3.csv("data/USArrests.csv", function(data) {
    data.forEach(function(d) {
      array1.push(Number(d.Murder));
      array2.push(Number(d.Assault));
      array3.push(Number(d.UrbanPop));
      array4.push(Number(d.Rape));
    })
  });

  var currentColumn = 0;

  function chooseVariable(val) {
    if(val=="1") {
      MurderHist();
    }
    else if(val=="2") {
      AssaultHist();
    }
    else if(val=="3") {
      UrbanPopHist();
    }
    else if(val=="4") {
      RapeHist();
    }
    else {
      d3.select('svg').selectAll('*').remove();

    }
  }

  function MurderHist() {
    currentColumn = 0;
    fillData(array1,binCount);
  }

  function AssaultHist() {
    currentColumn = 1;
    fillData(array2,binCount);
  }

  function UrbanPopHist() {
    currentColumn = 2;
    fillData(array3,binCount);
  }
```

```
      function RapeHist() {
        currentColumn = 3;
        fillData(array4,binCount);
      }

      function fillData(array, bin){
        d3.select('svg').selectAll('*').remove();

        var bins = d3.layout.histogram()
                     .bins(binCount)
                     (array);

        var widthBin = bins[0].dx;

        var dataArray = [];

        for(var i=0;i<bins.length;i++) {
          dataArray[i] = bins[i].length;
        }

        yMax = d3.max(bins, function(d){return d.length});
        yMin = d3.min(bins, function(d){return d.length});

        if(currentFigure == 0) {
          document.getElementById("bar").innerHTML = '';
          createHist(dataArray);
        }
        else if(currentFigure==1) {
          document.getElementById("chart").innerHTML = '';
          createPie(dataArray);
        }

        else if(currentFigure==2) {
          document.getElementById("forceD").innerHTML = '';
          createForceD(dataArray,edgeLength);
        }

      }

      activatebinCount();
      activateLengthChange();

    </script>
  </fieldset>
</body>
</html>
```

src/graphs.js

```
function activatebinCount() {
  d3.select("#binCountUpdate").on("mousedown", function() {

    var div = d3.select(this)
    .classed("active", true);

    var xPos = d3.mouse(div.node())[0]
    var w = d3.select(window)
    .on("mousemove", mousemove)
    .on("mouseup", function(){
      div.classed("active", false);
      w.on("mousemove", null).on("mouseup", null);
    });

    function mousemove() {
      if(d3.mouse(div.node())[0]+3 < xPos && binCount > 1){
        binCount -= 1;
        if(currentColumn==0) fillData(array1,binCount);
        if(currentColumn==1) fillData(array2,binCount);
        if(currentColumn==2) fillData(array3,binCount);
        if(currentColumn==3) fillData(array4,binCount);
        xPos = d3.mouse(div.node())[0];
      }
      else if(d3.mouse(div.node())[0]-3 > xPos && binCount < 40){
        binCount += 1;
        if(currentColumn==0) fillData(array1,binCount);
        if(currentColumn==1) fillData(array2,binCount);
        if(currentColumn==2) fillData(array3,binCount);
        if(currentColumn==3) fillData(array4,binCount);
        xPos = d3.mouse(div.node())[0];
      }
      document.getElementById("binsCount").innerHTML = binCount;
    }
  });
}

function activateLengthChange() {
  d3.select("#lengthUpdate").on("mousedown", function() {
    var div = d3.select(this)
    .classed("active", true);

    var xPos = d3.mouse(div.node())[0]
    var w = d3.select(window)
    .on("mousemove", mousemove2)
    .on("mouseup", function(){
      div.classed("active", false);
      w.on("mousemove", null).on("mouseup", null);
    });

    function mousemove2() {
      if(d3.mouse(div.node())[0]+10 < xPos && edgeLength > 59){
        edgeLength -= 15;
        if(currentColumn==0) fillData(array1,binCount);
        if(currentColumn==1) fillData(array2,binCount);
        if(currentColumn==2) fillData(array3,binCount);
        if(currentColumn==3) fillData(array4,binCount);
        xPos = d3.mouse(div.node())[0];
      }
      else if(d3.mouse(div.node())[0]-10 > xPos && edgeLength < 601){
        edgeLength += 15;
        if(currentColumn==0) fillData(array1,binCount);
        if(currentColumn==1) fillData(array2,binCount);
        if(currentColumn==2) fillData(array3,binCount);
        if(currentColumn==3) fillData(array4,binCount);
        xPos = d3.mouse(div.node())[0];
      }
      document.getElementById("edgeLength").innerHTML = edgeLength;
    }
  });
}

function createHist(histData){

  document.getElementById("graphName").innerHTML = "Histogram";

  var margin = {top: 10, right: 10, bottom: 10, left: 10};

  var x = d3.scale.ordinal()
  .rangeRoundBands([0, w]);

  var xAxis = d3.svg.axis()
  .scale(x)
  .orient("bottom")
  .ticks(10);
```

```
var svg = d3.select("#bar")
.append("svg")
.attr("width", w + margin.left + margin.right)
.attr("height", h + margin.top + margin.bottom)
.append("g")
.attr("transform", "translate(" + margin.left + "," + margin.top + ")");

var scaleHeight = (0.75*h/d3.max(histData));

var colorScale = d3.scale.linear()
.domain([yMin, yMax])
.range([d3.rgb(color).brighter(), d3.rgb(color).darker()]);

svg.append("g")
.attr("class", "x axis")
.attr("transform", "translate("+ (-0.50*barPadding) +"," + 1.005*h + ")")
.call(xAxis);

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 - (d * scaleHeight);
})
.attr("height", function(d) {
  return d * scaleHeight;
})
.attr('fill', function(d,i){
  return colorScale(d);
})
.on("mouseover", function(d,i) {
  d3.select(this)
  .attr("y",d3.select(this).attr("y") - 20)
  .attr("height",parseInt(d3.select(this).attr("height")) + 20)
  .attr("x",i * (w / histData.length) - 4)
  .attr("width",w / histData.length - barPadding + 8)

  d3.selectAll("text")
  .select(function(d, j) {
    if(j===i)
      return this;
    else
      return null;
  })
  .attr("fill","red")
  .attr("font-family","sans-serif")
  .style("opacity",100)

})
.on("mouseout", function(d,i) {
  d3.select(this)
  .attr("y",parseInt(d3.select(this).attr("y")) + 20)
  .attr("height",parseInt(d3.select(this).attr("height")) - 20)
  .attr("x",i * (w / histData.length))
  .attr("width", w / histData.length - barPadding);

  d3.selectAll("text")
  .select(function(d, j) {
    if(j===i)
      return this;
    else
      return null;
    })
    .style("opacity",0)})
  .on("click",function(){
    document.getElementById("bar").innerHTML = '';
    createPie(histData);
    currentFigure = 1;
  });

  var scaleHeight = (0.77*h/d3.max(histData));
  var text = svg.selectAll("text")
  .data(histData)
  .enter()
  .append("text")
  .text(function(d) {
    return d;
  })
  .attr("text-anchor", "middle")
  .attr("x", function(d, i) {
    return i * (w / histData.length) + (w / histData.length - barPadding) / 2;
  })
  .attr("y", function(d) {
    return h-(d * scaleHeight)-30;
  })
  .attr("font-family", "sans-serif")
  .attr("font-size", 23)
  .attr("fill", "red")
  .style("opacity",0);

}

function createPie(histData){

  document.getElementById("graphName").innerHTML = "Pie Chart";

  var colorMax = 255/d3.max(histData);

  var colorScale = d3.scale.linear()
  .domain([yMin, yMax])
  .range([d3.rgb(color).brighter(), d3.rgb(color).darker()]);


  var radius = h*0.45;

  var svg = d3.select("#chart")
  .append("svg")
  .attr("width", w)
  .attr("height", h)
  .append('g')
  .attr('transform', 'translate(' + w/2 + ',' + 1.15*radius + ')');

  var pie = d3.layout.pie()
  .value(function(d) {return d;});
  var arc = d3.svg.arc()
  .outerRadius(radius)
  .innerRadius(0);

  var arc2 = d3.svg.arc()
  .innerRadius(0)
  .outerRadius(radius + 10);

  svg.selectAll("path")
  .data(pie(histData))
  .enter()
  .append("path")
  .attr("stroke", "white")
  .attr('d', arc)
  .attr('fill', function(d,i){
    return colorScale(d.value);
  })
  .on("mouseover", function(d,i) {
    d3.select(this)
    .attr("stroke", "white")
    .attr("d", arc2)
    .attr("stroke-width", 2);

    svg.append("text")
    .attr("transform", function() {
      return "translate(" + arc.centroid(d) + ")";
    })
    .style("text-anchor", "middle")
    .attr("font-size", 23)
    .attr("class", "label")
    .attr("fill","red")
    .attr("font-family","sans-serif")
    .style("opacity",100)
    .text(parseInt(d.value));

  })
  .on("mouseout", function(d,i) {
    d3.select(this)
    .attr("d",arc)
    .attr("stroke", "white");

    svg.selectAll("text")
    .style("opacity",0);
  })
  .on("click",function(){
    document.getElementById("chart").innerHTML = '';
    createForceD(histData,edgeLength);
    currentFigure = 2 ;
  });
}

function createForceD(histData,dist) {

  document.getElementById("graphName").innerHTML = "Force Directed Graph";

  var nodes = [];

  for(var i=0;i<histData.length;i++) {
    var node = {};
    node.name = histData[i];
    nodes.push(node);
  }

  var links = [];
  for(var i=0;i<2*histData.length;i++) {
    var link = {};
    link.source = parseInt(Math.random()*histData.length);
    link.target = parseInt(Math.random()*histData.length);
    links.push(link);
  }


  var dataset = {};
  dataset.nodes = nodes;
  dataset.links = links;

  var colorScale = d3.scale.linear()
  .domain([yMin, yMax])
  .range([d3.rgb(color).brighter(), d3.rgb(color).darker()]);


  var svg = d3.select("#forceD")
  .append("svg")
  .attr("width", w)
  .attr("height", h);

  var force = d3.layout.force()
  .links(dataset.links)
  .nodes(dataset.nodes)
  .size([w, h])
  .linkDistance([dist])
  .charge([-200])
  .start();

  var nodes = svg.selectAll("circle")
  .data(dataset.nodes)
  .enter()
  .append("circle")
  .attr("r", function(d) {
    return Number(d.name)*3;
  })
  .style("fill", function(d, i) {
    return colorScale(Number(d.name));
  })
  .call(force.drag)
  .on("click",function(){
    document.getElementById("forceD").innerHTML = '';
    createHist(histData);
    currentFigure = 0;
  });

  var lines = svg.selectAll("line")
  .data(dataset.links)
  .enter()
  .append("line");

  force.on("tick", function() {
    lines.attr("x1", function(d) { return d.source.x; })
    .attr("x2", function(d) { return d.target.x; })
    .attr("y1", function(d) { return d.source.y; })
    .attr("y2", function(d) { return d.target.y; })
    .attr("stroke", "black");

    nodes.attr("cx", function(d) { return d.x; })
    .attr("cy", function(d) { return d.y; });

  });
}
```