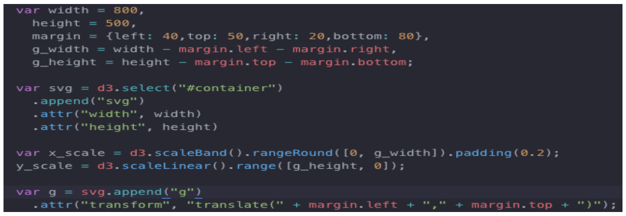
Assignment report

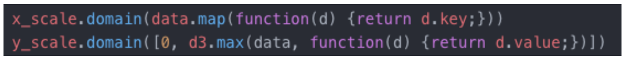
My program descripts the House price with different bedrooms in the 50 states of US. The CSV file is downloaded from Zillow, which is a housing company. The file includes 4542 rows and 30 columns of data points, from 2010 to 2017.

First of all, I made a svg to draw my chart, and scaled the range of the data. Other basic elements were also added into the canvas, such as the title of the chart. In the <head>part, some general styles were set for later element attributes usage.



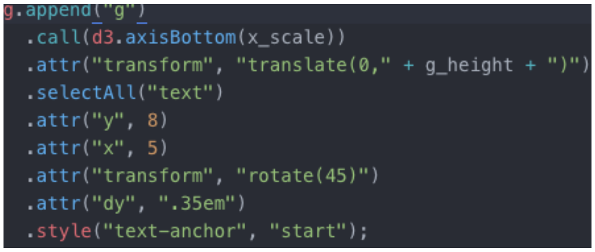
```
var width = 800,
  height = 500,
  margin = {left: 40,top: 50,right: 20,bottom: 80},
  g_width = width - margin.left - margin.right,
  g_height = height - margin.top - margin.bottom;

var svg = d3.select("#container")
  .append("svg")
  .attr("width", width)
  .attr("height", height)

var x_scale = d3.scaleBand().rangeRound([0, g_width]).padding(0.2);
y_scale = d3.scaleLinear().range([g_height, 0]);

var g = svg.append("g")
  .attr("transform", "translate(" + margin.left + "," + margin.top + ")");
```

In each mouse on-click event function latter on, the updated data were re-scaled.

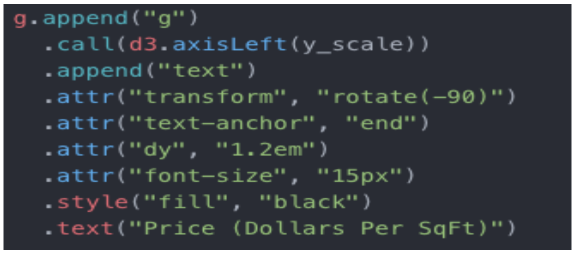


```
x_scale.domain(data.map(function(d) {return d.key;}))
y_scale.domain([0, d3.max(data, function(d) {return d.value;})])
```

The x and y axis was added by .append() and set by using the d3.asixBottom() and d3.axisLeft(). The y axis title and the x axis labels are implemented as followed.

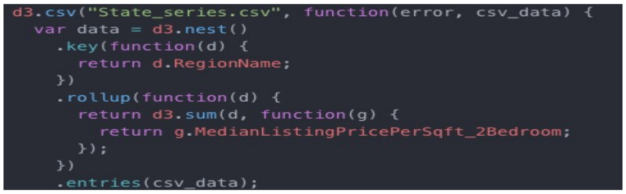


```
g.append("g")
  .call(d3.axisBottom(x_scale))
  .attr("transform", "translate(0," + g_height + ")")
  .selectAll("text")
  .attr("y", 8)
  .attr("x", 5)
  .attr("transform", "rotate(45)")
  .attr("dy", ".35em")
  .style("text-anchor", "start");
```



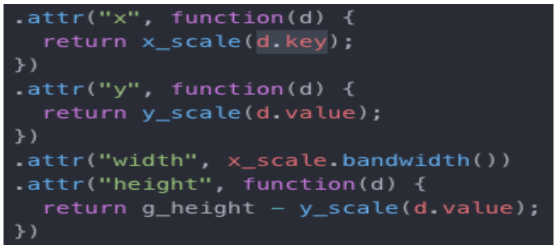
```
g.append("g")
  .call(d3.axisLeft(y_scale))
  .append("text")
  .attr("transform", "rotate(-90)")
  .attr("text-anchor", "end")
  .attr("dy", "1.2em")
  .attr("font-size", "15px")
  .style("fill", "black")
  .text("Price (Dollars Per SqFt)")
```

I used a nest function so as to regroup the data from different time period into different U.S. state regions, and then calculated the 2-bedroom, 3-bedrom and 4-bedroom average price of the whole 8 years (96 months).



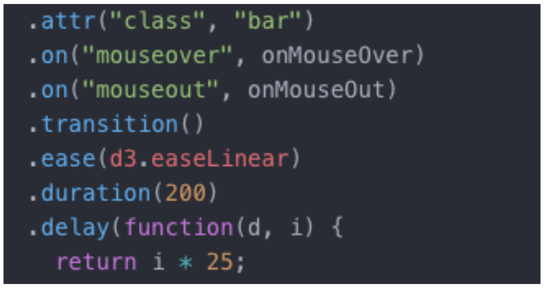
```
d3.csv("State_series.csv", function(error, csv_data) {
  var data = d3.nest()
    .key(function(d) {
      return d.RegionName;
    })
    .rollup(function(d) {
      return d3.sum(d, function(g) {
        return g.MedianListingPricePerSqft_2Bedroom;
      });
    })
    .entries(csv_data);
```

Then is the data load part, and in this part, bars were set by selecting all the bar class. The width was set by x_scale.bandwidth(d.key), and height was set by y_scale(d.value) according to the data pairs.



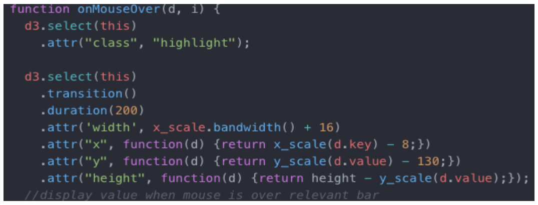
```
.attr("x", function(d) {
  return x_scale(d.key);
})
.attr("y", function(d) {
  return y_scale(d.value);
})
.attr("width", x_scale.bandwidth())
.attr("height", function(d) {
  return g_height - y_scale(d.value);
})
```

To implement the mouseover event, the .on("mouseover", onMouseOver) and .on("mouseout", onMouseOut) were added to the bars. The transition and duration time were also set on the bar class.

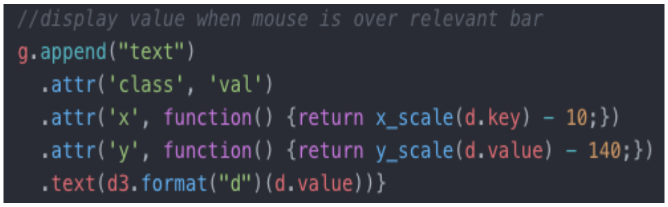


```
 .attr("class", "bar")
 .on("mouseover", onMouseOver)
 .on("mouseout", onMouseOut)
 .transition()
 .ease(d3.easeLinear)
 .duration(200)
 .delay(function(d, i) {
   return i * 25;
```

The wider and higher were set later on in the onMouseOver and onMouseout functions. The mouse-over value on-top display function was implemented by append text and adjust the relative position of the text.

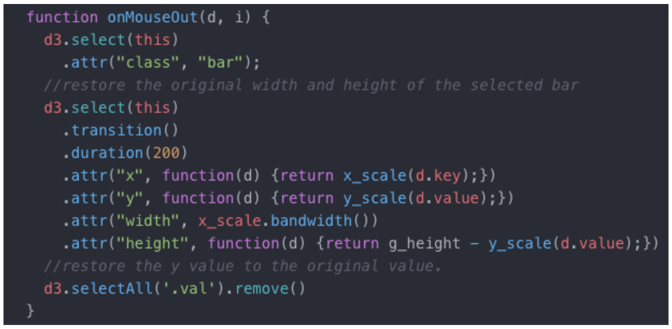


```
function onMouseOver(d, i) {
  d3.select(this)
    .attr("class", "highlight");

  d3.select(this)
    .transition()
    .duration(200)
    .attr('width', x_scale.bandwidth() + 16)
    .attr("x", function(d) {return x_scale(d.key) - 8;})
    .attr("y", function(d) {return y_scale(d.value) - 130;})
    .attr("height", function(d) {return height - y_scale(d.value);});
  //display value when mouse is over relevant bar
```



```
//display value when mouse is over relevant bar
g.append("text")
  .attr('class', 'val')
  .attr('x', function() {return x_scale(d.key) - 10;})
  .attr('y', function() {return y_scale(d.value) - 140;})
  .text(d3.format("d")(d.value))}
```

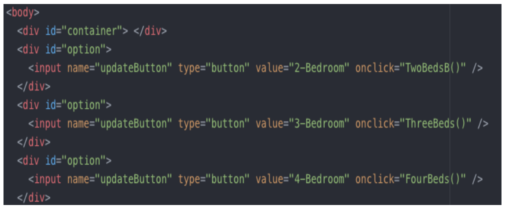
When move out of on bar, the bar width and height would be shrinking to the original size, all the val class will be remove after moving out. The duration time was also set in this function.



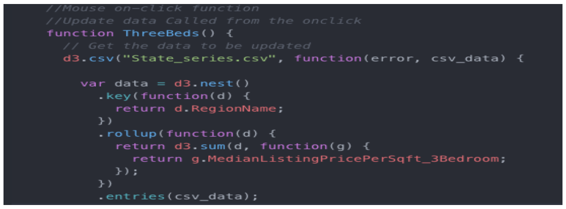
```
function onMouseOut(d, i) {
  d3.select(this)
    .attr("class", "bar");
  //restore the original width and height of the selected bar
  d3.select(this)
    .transition()
    .duration(200)
    .attr("x", function(d) {return x_scale(d.key);})
    .attr("y", function(d) {return y_scale(d.value);})
    .attr("width", x_scale.bandwidth())
    .attr("height", function(d) {return g_height - y_scale(d.value);})
  //restore the y value to the original value.
  d3.selectAll('.val').remove()
}
```

The menu were set by using option. There are three variables listed out on the svg as examples. When clicking on one of them, the chart will update to the relevant data key and values. And the mouseover function also works in each of them.
The mouseover function also works in each of them. The onclick functions made the different variables charts changeable to each other. In each onclick function, the main parts include

nesting and reading in the new data, seting the new x and y scales, updating the chart title, updating the x and y axises, and updating the widths and heights of the bars.



```html
<body>
  <div id="container"> </div>
  <div id="option">
    <input name="updateButton" type="button" value="2-Bedroom" onclick="TwoBedsB()" />
  </div>
  <div id="option">
    <input name="updateButton" type="button" value="3-Bedroom" onclick="ThreeBeds()" />
  </div>
  <div id="option">
    <input name="updateButton" type="button" value="4-Bedroom" onclick="FourBeds()" />
  </div>
```



```javascript
//Mouse on-click function
//Update data Called from the onclick
function ThreeBeds() {
  // Get the data to be updated
  d3.csv("State_series.csv", function(error, csv_data) {

    var data = d3.nest()
      .key(function(d) {
        return d.RegionName;
      })
      .rollup(function(d) {
        return d3.sum(d, function(g) {
          return g.MedianListingPricePerSqft_3Bedroom;
        });
      })
      .entries(csv_data);
```