

D3– Part2

Dheeraj Chinni Ranga

Data Engineering

JSON Data

- Java Script Object Notation(JSON) light weighted interchange format.
- This is easy to both humans and machines to parse and generate outcome of it.
- This format is language independent and accessible to all the C-family languages and scripting languages.
- This format helps communication between to languages easy by defining Universal format.

JSON Data

- JSON is built on two structures:
 - A collection of Name/Value pairs, generally considered as object, record, dictionary according to the language.
 - An ordered list of values, this is considered as array vector, List.
 - An JSON object is an un ordered list which starts with a left brace '{ ' and ends with Right brace '}'.
 - Each Name and Value are represented in double quotations.
 - Each name is followed by colon : and each value is separated by comma ,

Example JSON

```
[{"label": "Jhon", "value": "481310.2195208175"},  
{"label": "ford", "value": "5675372.220709492"},  
{"label": "samantha", "value": "447772.17123047076"},  
{"label": "Lupan", "value": "4949497.18247217"},  
{"label": "jordan", "value": "505903.30021648406"},  
{"label": "elizabeth", "value": "418484.6013686644"},  
{"label": "siva", "value": "566813.3719400858"}];
```

D3 JSON

- D3 can show very dynamic behavior while showing the JSON file graphically.
- D3 can give options like adding, deleting the data from the graph.
- Lets build a Pie char with the JSON File show above.
- The JSON file has name of and the average time commuted by student going to school for 30 days.

D3 JSON

- Lets start adding the height , width, Radius and the color of the pie the Java script code looks like below.

```
var w = 400;  
var h = 400;  
var r = h/2;  
var color = d3.scale.category20c();
```

- D3.scale.category20c() will choose the color according to the scale of the specific category.

D3 JSON

- Add the data to a variable data which we will use in the whole d3.

```
var data = [{"label": "Jhon", "value": "481310.2195208175"},  
            {"label": "ford", "value": "5675372.220709492"},  
            {"label": "samantha", "value": "447772.17123047076"},  
            {"label": "Lupan", "value": "4949497.18247217"},  
            {"label": "jordan", "value": "505903.30021648406"},  
            {"label": "elizabeth", "value": "418484.6013686644"},  
            {"label": "siva", "value": "566813.3719400858"}];
```

D3 JSON

- Declare the scale and the chart type.

```
var vis = d3.select('#chart').append("svg:svg").data([data]).attr("width", w).attr("height", h).  
    append("svg:g").attr("transform", "translate(" + r + "," + r + ")");  
var pie = d3.layout.pie().value(function(d){return d.value;});
```

- #chart is used to access the D3 which is defined with the svg format adding data, width and height to it.
- Pie here we are declaring the pie chart based on the filed value.

D3 JSON

- Let us specify the arc radius with the following function.

```
// declare an arc generator function  
var arc = d3.svg.arc().outerRadius(r);
```

- This will give the outer radius of the arc.

D3 JSON

- Select paths using the path generators using the following code.

```
// select paths, use arc generator to draw
var arcs = vis.selectAll("g.slice").data(pie).enter().append("svg:g").attr("class", "slice");
arcs.append("svg:path")
    .attr("fill", function(d, i){
        return color(i);
    })
    .attr("d", function (d) {
        // log the result of the arc generator to show how cool it is :)
        console.log(arc(d));
        return arc(d);
    });
```

- The console.log will show how the arcs are generated can be seen using any of the Java script Debuggers.

D3 JSON

- Finally add the Label to the specific arc in the pie chart.

```
// add the text
arcs.append("svg:text").attr("transform", function(d){
    d.innerRadius = 0;
    d.outerRadius = r;
    return "translate(" + arc.centroid(d) + ")";}).attr("text-anchor", "middle").text( function(d, i
    ) {
    return data[i].label;}
);
```

- This will add the label to the middle of the arc that is available for that specific value in the JSON file.
- The value can also be show on the pie chart.

D3 JSON

- In the end just add the id in the div function where ever it is necessary.

```
<body>  
  <h1>Student commute time for School A</h1>  
  <div id="chart"></div>  
</body>
```

- Just add an id that is specified in the select all function.

Final Code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Student Statistics</title>
  <script type="text/javascript" src="http://d3js.org/d3.v2.js"></script>

  <style>

  </style>

</head>
<body>
  <h1>Student commute time for School A</h1>
  <div id="chart"></div>
</body>

  <script type="text/javascript">

    var w = 400;
    var h = 400;
    var r = h/2;
    var color = d3.scale.category20c();
```

Final Code

```
var data = [{"label": "Jhon", "value": "481310.2195208175"},  
            {"label": "ford", "value": "5675372.220709492"},  
            {"label": "samantha", "value": "447772.17123047076"},  
            {"label": "Lupan", "value": "4949497.18247217"},  
            {"label": "jordan", "value": "505903.30021648406"},  
            {"label": "elizabeth", "value": "418484.6013686644"},  
            {"label": "siva", "value": "566813.3719400858"}];  
  
var vis = d3.select('#chart').append("svg:svg").data([data]).attr("width", w).attr("height", h).  
            append("svg:g").attr("transform", "translate(" + r + "," + r + ")");  
var pie = d3.layout.pie().value(function(d){return d.value;});  
  
// declare an arc generator function  
var arc = d3.svg.arc().outerRadius(r);
```

Final Code

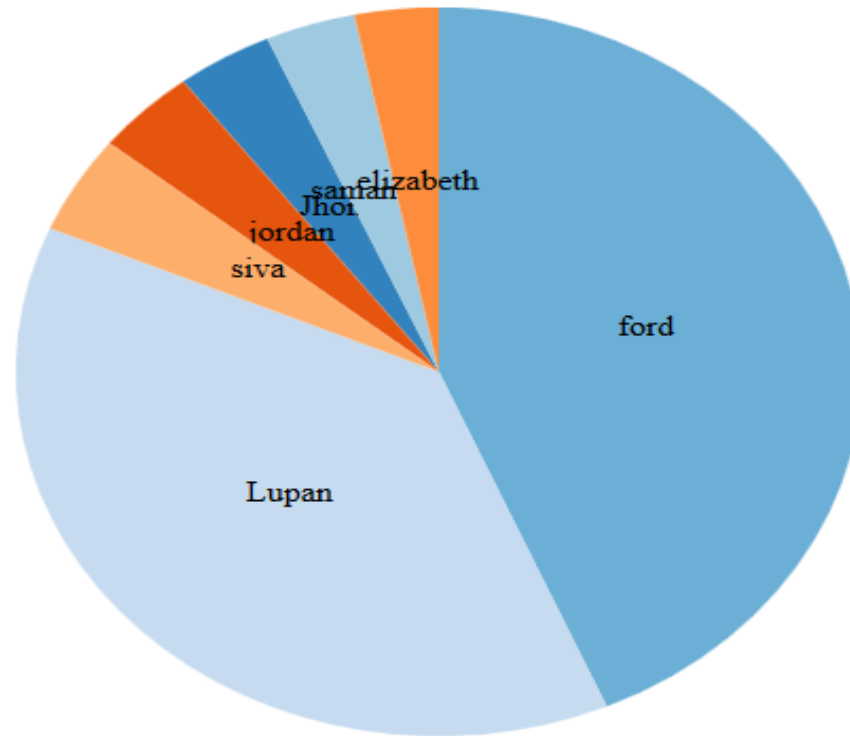
```
// select paths, use arc generator to draw
var arcs = vis.selectAll("g.slice").data(pie).enter().append("svg:g").attr("class", "slice");
arcs.append("svg:path")
  .attr("fill", function(d, i){
    return color(i);
  })
  .attr("d", function (d) {
    // log the result of the arc generator to show how cool it is :)
    console.log(arc(d));
    return arc(d);
  });

// add the text
arcs.append("svg:text").attr("transform", function(d){
  d.innerRadius = 0;
  d.outerRadius = r;
  return "translate(" + arc.centroid(d) + ")";}).attr("text-anchor", "middle").text( function(d, i)
  ) {
  return data[i].label;
});

</script>
</html>
```

D3 Chart

Student commute time for School A



Thank you 😊