

# Visualization of Real Time Data Driven Systems using D3 Visualization Technique

Eesha Karn

Department of Information Technology  
Poornima Institute of Engineering and Technology  
Jaipur, Rajasthan, India

**Abstract**— Integrating multiple information and communication technologies in a smart city requires numerous web applications as well as mobile-based applications which result in analysis terabytes or even zettabytes of data. Therefore handling such a huge amount of data in a very pristine and efficient manner is the need of the hour. So, It is a big challenge for us to visualize such a huge amount of data. There are various such types of Charting Libraries. One of them is D3.D3 stands for Data Driven Documents and follows the principle of DOM (Document Object Model).D3 helps designers in binding arbitrary data with the document elements and thus elements are modified and generated.[8] This increases the expressiveness and hence makes analysis easier.

**Keywords**—pristine, mobile-based, visualize, Charting Libraries, D3, DOM.

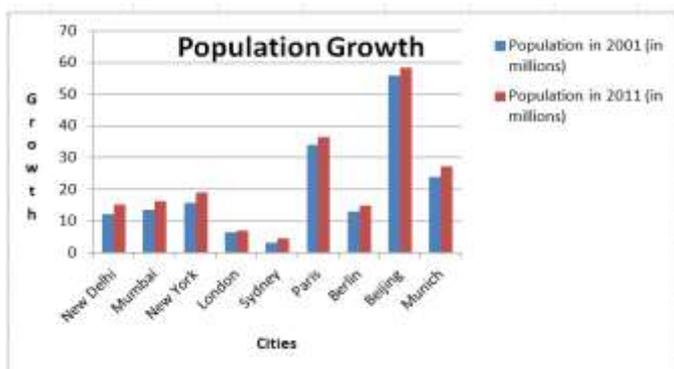
## I. INTRODUCTION

A smart city deals with the integration of all the technical solutions that helps in the efficient advancement of education system, medical system, transportation system, schools, libraries, law enforcement system and other community services. The goal of building a smart city is to improve the quality of life by reducing cost and resource consumption, increasing quality of services, enhancing functionalities and enhancing the interaction of the common people with the government.[3] For this the smart cities are going to use Information and Communication Technologies (ICT). Therefore, for the efficient development and in order to achieve the above goals, we need various algorithms and visualization techniques so that analysis of data can be made easier. Now-a-days hybrid applications require a terabytes of data to be visualized and analysed.[7] Visualizations can be done with the help of various Charting Libraries and Analysis of data is possible through Data Mining techniques. The Data Visualization is a three steps process- First is Zooming or defining the data, second is filtration of data and finally detailed analysis of data on demand is presented. In all these step, first of all users need to get an overview of the data that is with them, then interesting patterns need to be extracted from that data. These interesting patterns can be either one or more than one. For analysis, users need to drill-down and access the detailed data. On the other hand, for Visualization users often apply multiple visualization tools simultaneously.[2] This is easily possible in the web applications. For this various technologies such as HTML,

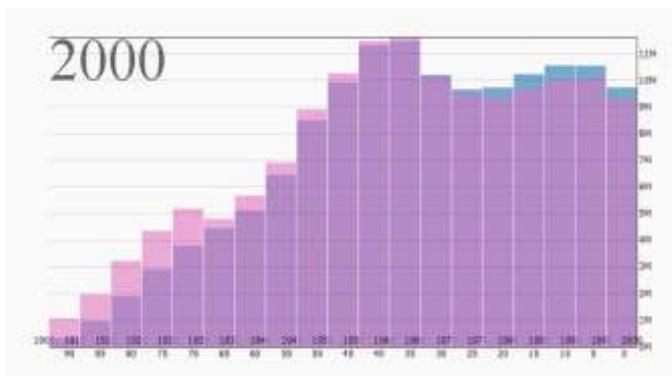
CSS, Javascript and SVG are used.HTML is used for page content, CSS is used for designing an user interface, Javascript is used for interaction between the document elements and SVG (Scalable Vector Graphics) is used for creating various graphics for different graphs and so on. The success of these technologies on web platform enables interpretation of DOM(Document Object Model). DOM defines hierarchical structure of a webpage including form, div, table and each and every element. Apart from various programming interfaces, modern browsers include various graphical tools which display elements trees and debug interactive scripts. But this interoperability lost with various visualization toolkits due to encapsulation of DOM with more specialized forms. Rather than visualizing only basic elements, such tools are empowered with more custom graphs and hence increasing efficiency and functionalities of data elements by reducing the cost.[6]

Previous tools can also be used by those developers not having deeper understanding of the standards and the toolkits. Internal structures are exposed only when the error arises. Thus visualization with a proper toolkit is quite important which requires in-depth knowledge of the toolkit to the average users. Thus, D3 came into existence. It helps in manipulating document based data using HTML, CSS and SVG. It uses powerful visualization components and a Data Driven approach to DOM manipulation which is compatible with any modern browser. Embedded with HTML WebPages, D3.js is a Javascript library that select element, create SVG and add transitions, dynamic effects and tooltips to them.[25] These objects can be styled with CSS. The data can be in the form of JSON, CSV, and Excel etc. Thus, the principle of D3.js is to select a given set of Document Object Model (DOM), style them with a CSS-Selector and finally applying operators to manipulate those documents.

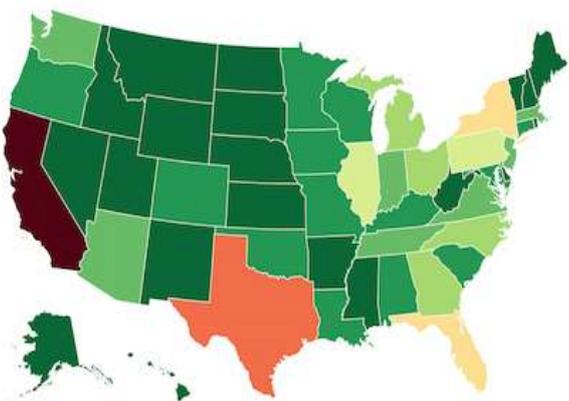
Example: Suppose we are having population data (in millions) for few places. When these data are shown in excel graph, it can be simply visualized as following:



On the other hand in D3, same data can be visualized in different formats. It can be shown as pyramid where we are having tooltip showing information at a particular point through hover effect as:



Also, we can visualize population data of different places on the map using D3 as:



## II. D3

D3 (Data Driven Documents) is a Javascript Library that uses HTML, CSS and SVG to manipulate document based data in a very efficient manner. Some analogues of D3 are JQuery, CSS and XSLT.[30] Thus, D3 core contribution is towards Visualization rather than a framework. As far as framework is

concerned, it provides declarative framework for mapping data to visual elements. D3 also helped in removing technical constraints, Complications in debugging and code abstraction problems. Rather D3 introduces features that may inform other visualization frameworks: Query-Driven Selection, Data Binding to Scenegraph elements, Document Transformation as an atomic operation and immediate property evaluation semantics.[27]

Some of the most effective graphs of D3 are listed below:-

A. Bullet Chart: Bullet charts are used to visualize data in a smaller space. It can be used to visualize data such as profit, revenues, performance etc.

B. Chord Diagrams: Chord Diagrams show relationships among group of entities. It demonstrates simple interactivity using mouseover filtering.

C. Bubble Charts: Bubble Charts encode data in the area of circles. They can pack hundreds of values into a small space.

D. Noncontiguous Cartograms: It is used for Geographic Visualization and is used by D3 geo module. It encodes values for geographic regions by scaling each region around its projected centroid.

E. Bezier Curves: It is a curve in which parameter animates from 0 to 1 and control points can be moved to affect the curve.

F. Treemap: A treemap subdivides an area into rectangles. The area of any node in the tree corresponds to its value.

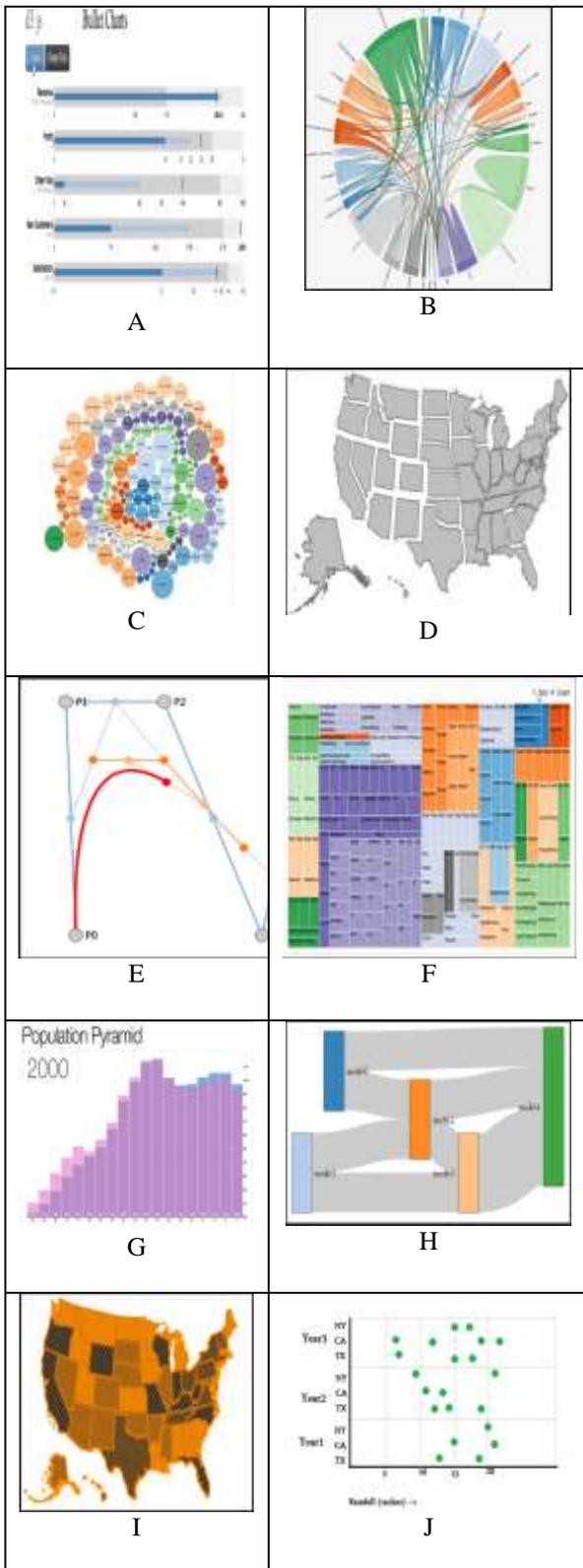
G. Population Pyramid: A population pyramid shows distribution of population in a specified region. It uses Arrow Key to show change in population and changing the displayed years.

H. Sankey Diagrams: Sankey Diagrams visualizes the magnitude of flow between the nodes in a network.

I. Zoomable Map: It visualizes geographic information about the whole world. It uses GeoJSON, TopoJSON, add Zoom functionality and tooltips for each country and this whole process is repeated with every demo.

J. Scatter Plot: A Scatter Plot is a Mathematical diagram that uses Cartesian Co-ordinates to display values for typically two variables in a given set of data. The number of variables can be increased if the points are colour-coded. The variables are displayed as a collection of points, each having value of one variable determining position on horizontal axis and the value of the other variable determining the position on the vertical axis.[28]

The diagrams of the above mentioned graphs are following:



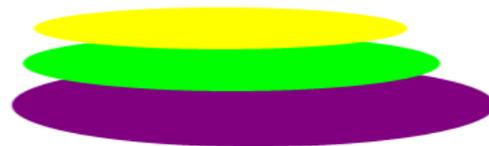
### III. SVG

Document Object Model (DOM) uses operators to manipulate document objects in a similar manner to JQuery. Other can be SVG. SVG stands for Scalable Vector Graphics. It is a vector image format for two-dimensional graphics that can be used for interactivity and animation. SVG was developed by World Wide Web Consortium (W3C) in 1999 and is open source. It is an XML text file and hence can be searched, scripted, indexed and compressed. Since SVG is written as a text file, therefore it can be created and edited using any text editor but this is generally done using Drawing Software. Research in this field has changed the behaviour of graphical representation of elements and new approaches to planning processes.[19] SVG is a web-technology based on native XML that is interpreted by the SVG Processor in the browser. In this case learning of new technologies are quite simplified because all the commands within XML elements and attributes are determined using a hidden micro code that is entered by a SVG Processor on the determined operating system. Thus SVG and XML are independent of hardware, software and the ways of transferring data. Some examples of SVG is given below:

A. In the first example, suppose we are willing to create a shape having three oval shapes with three different colours which are yellow, green and purple, and all of them are overlapped over one another sequentially. So, SVG for the same can be created using following:

```
<svg height="150" width="500">
<ellipse cx="240" cy="100" rx="220" ry="30" style="fill:purple" />
<ellipse cx="220" cy="70" rx="190" ry="20" style="fill:lime" />
<ellipse cx="210" cy="45" rx="170" ry="15" style="fill:yellow" /
</svg>
```

Output:



B. Suppose we want to draw a rectangle having circular corners and gray boundary and should be filled with pink colour. So, following is the SVG:

```
<svg width="400" height="180">
<rect x="50" y="20" rx="20" ry="20" width="150"
```

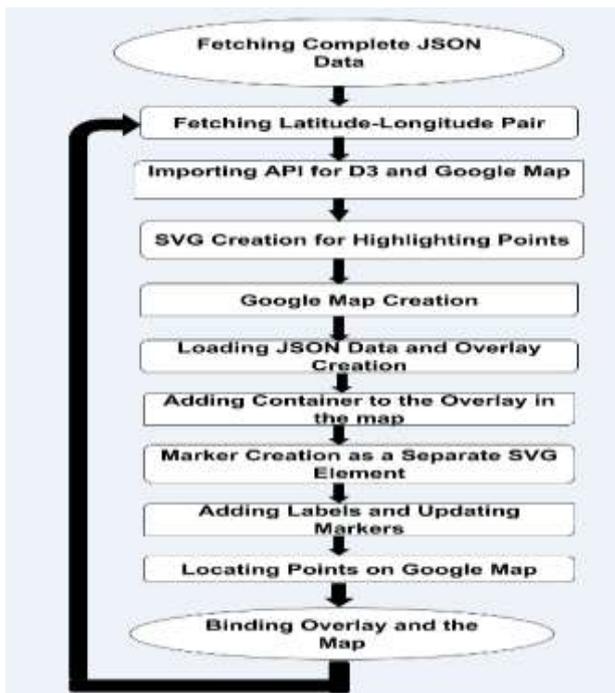
```
height="150"
style="fill:red;stroke:black;stroke-width:5;opacity:0.5">
</svg>
```

Output:



#### IV. INFORMATION VISUALIZATION SYSTEM

The goal of Information Visualization System is to visualize data on the Google Map using D3.js.[11]



Following are the steps to do the same:

**Step 1:** The dataset that we need to visualize is in JSON format named ‘test-graph.json’. It contains a number of latitude-longitude pairs and each pair is lined with a source and a target. The dataset is as following:

```
{ "directed": true, "graph": [], "nodes": [
  { "lat": 44.391643516091975, "lng": 23.159677682342053,
    "id": "1:a" },
  { "lat": 44.315988, "lng": 23.818359, "id": "a:a:" } ],
  "links": [ { "source": 1, "target": 25 },
  { "source": 1, "target": 26 } ],
```

```
"multigraph": false } etc.
```

**Step 2:** Now, we will import APIs for both Google Map and D3.js in our HTML code using JavaScript.

```
<script type="text/javascript" src =
"http://maps.google.com/maps/api/js?sensor=true">
</script>
<script type="text/javascript"
src="http://mbostock.github.com/d3/d3.js?1.29.1">
</script>
```

**Step 3:** Again the Styling of SVG is to be done.

```
.stations, .stations svg { position: absolute; }
.stations line { position: absolute; stroke: black;
stroke-width: 2px; }
.stations svg { width: 60px; height: 20px;
padding-right: 100px; font: 10px sans-serif; }
.stations circle { fill: brown; stroke: black;
stroke-width: 1.5px; }
```

**Step 4:** After this we will create a Google Map by keeping a particular latitude-longitude pair in focus.

```
var map = new google.maps.Map( d3.select("#map") .node(),
{ zoom: 7, center: new google.maps. LatLng (44.331216,
23.927536), mapTypeId: google.maps.
MapTypeId.TERRAIN });
```

**Step 5:** Again we will add the container when the overlay is added to the map.

```
overlay.onAdd = function() {
var layer = d3.select(this.getPanels().overlayLayer)
.append("div") .attr("class", "stations");
```

**Step 6:** Now we will draw a marker as a separate SVG element. Here we are also specifying the size of each SVG element and also updating the existing markers.

```
overlay.draw = function() { var projection = this
.getProjection(), padding = 10; var node_coord = { };
var marker = layer.selectAll("svg").data(json.nodes)
.each(transform) .enter().append("svg:svg")
.each(transform).attr("class", "marker"); marker .
append("svg:circle").attr("r", 5) .attr("cx", padding)
.attr("cy", padding);
```

**Step 7:** After all these, a label is added and again existing marker gets updated.

```
marker.append("svg:text").attr("x", padding + 7).attr("y", padding).attr("dy", ".37em")
.text(function(d { return d.id; });
var markerLink = layer.selectAll("link")
.data(json.links).each(pathTransform)
.enter().append("svg:svg").each(pathTransform)
.attr("class", "line").append("svg:line")
.style("stroke-width", 5);
```

**Step 8:** Now we are trying to add a circle in order to locate points on the Google Map.

```
function pathTransform(d) { console .log (node_coord [d.source-1 + "," + 1]);dsrc = new google.maps.LatLng(node_coord[d.source-1 + "," + 1], node_coord[d.source-1 + "," + 0]);dtrg = new google.maps.LatLng(node_coord[d.target-1 + "," + 1], node_coord[d.target-1 + "," + 0]);console.log (dsrc);console.log(dtrg);console.log("source" + d.source);console.log("target" + d.target);dsrc = projection.fromLatLngToDivPixel(dsrg);dtrg = projection.fromLatLngToDivPixel(dtrg);console.log(dsrg); console.log(dtrg);return d3.select(this) .attr("x1", dsrg.x - padding).attr("y1", dsrg.y - padding).attr("x2", dtrg.x - padding).attr("y2", dtrg.y - padding);} function transform(d,i) {node_coord[i + "," + 0] = d.lng;node_coord[i + "," + 1] = d.lat;d = new google.maps.LatLng(d.lat, d.lng); d = projection.fromLatLngToDivPixel(d); return d3.select(this).style("left", (d.x - padding) + "px") .style("top", (d.y - padding) + "px");};};
```

**Step 9:** Finally we will bind our overlay to the map.

```
overlay.setMap(map);});
```

The Output of the above mentioned steps is the Superimposition of the data that we have with us over the Google Map.[18] It can be shown as:



## V. CONCLUSION

Hence Data Visualization is the scientific representation of interactive and sensory data. Thus, it helps executives and other knowledge workers to dramatically improve their ability to grasp information hiding in their data. It absorbs information in a new and more constructive way. It also visualizes the relationship and patterns between operational and business activities. Other functionalities include identifying and acting on emerging trends by manipulating and interacting directly with data. Therefore Data Visualization has a big role in today's world by fostering a new business Language and so plays an important role in building a smart city.

## VI. REFERENCES

- [1] R. A. Becker and W. S. Cleveland. Brushing scatter plots. *Technometrics*,29:127–142, May 1987.
- [2] M. Bostock and J. Heer. Protovis: A graphical toolkit for visualization. *IEEE Trans Vis and Comp Graphics*, 15(6):1121–1128, 2009.
- [3] <https://www.bartlett.ucl.ac.uk/casa/pdf/paper188>
- [4] L. Byron and M. Wattenberg. Stacked graphs – geometry & aesthetics. *IEEE Trans. Vis. and Comp. Graphics*, 14(6):1245–1252, 2008.
- [5] S. K. Card, J. D. Mackinlay, and B. Shneiderman, editors. *Readings in information visualization: using vision to think*. Morgan Kaufmann, San Francisco, CA, 1999.
- [6] d3.js. <http://mbostock.github.com/d3/>, Mar 2011.
- [7][https://www.ctg.albany.edu/publications/journals/hicss\\_2012\\_smartcities/hicss\\_2012\\_smartcities.pdf](https://www.ctg.albany.edu/publications/journals/hicss_2012_smartcities/hicss_2012_smartcities.pdf)

- [8] [http://unhabitat.org/wp-content/uploads/2015/04/Habitat-III-Issue-Paper-21\\_Smart-Cities-2.0.pdf](http://unhabitat.org/wp-content/uploads/2015/04/Habitat-III-Issue-Paper-21_Smart-Cities-2.0.pdf)
- [9] Flare. <http://flare.prefuse.org>, Mar 2011.
- [10] M. A. Harrower and C. A. Brewer. Colorbrewer.org: An online tool for selecting color schemes for maps. *The Cartographic Journal*, 40:27–37, 2003.
- [11] <https://developers.google.com/maps/tutorials/visualizing/earthquakes>.
- [12] J. Heer and M. Agrawala. Software design patterns for information visualization. *IEEE Trans Vis and Comp Graphics*, 12(5):853–860, 2006.
- [13] J. Heer and M. Bostock. Declarative language design for interactive visualization. *IEEE Trans Vis and Comp Graphics*, 16(6):1149–1156, 2010.
- [14] J. Heer, S. K. Card, and J. A. Landay. prefuse: a toolkit for interactive information visualization. In *Proc. ACM CHI*, pages 421–430, 2005.
- [15] J. Heer and G. G. Robertson. Animated transitions in statistical data graphics. *IEEE Trans Vis and Comp Graphics*, 13(6):1240–1247, 2007.
- [16] P. Hudak. Building domain-specific embedded languages. *ACM Computing Computing Surveys*, 28:196, December 1996.
- [17] S. Hudson and J. T. Stasko. Animation support in a user interface toolkit: Flexible, robust, and reusable abstractions. In *ACM UIST*, pages 57–67, 1993.
- [18] JavaScript InfoVis Toolkit. <http://thejit.org/>, Mar 2011.
- [19] jQuery. <http://jquery.com/>, Mar 2011.
- [20] M. Krzywinski, J. Schein, I. Birol, J. Connors, R. Gascoyne, D. Horsman, S. J. Jones, and M. A. Marra. Circos: An information aesthetic for comparative genomics. *Genome Research*, 19(9):1639–1645, Sep 2009.
- [21] B. Shneiderman, “The eye have it: A task by data type taxonomy for information visualizations,” in *Visual Languages*, 1996.
- [22] A guide to choosing the right chart type, Szoka, Kathryn, *Professional Communication*, IEEE Transactions on Volume: PC-25 , Issue: 2 , Publication Year: 1982 , Page(s): 98 - 101.
- [23] Daniel A. Keim, “Information Visualization and Visual Data Mining”, *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS*, VOL. 7, NO. 1, JANUARY-MARCH 2002.
- [24] “Use and Misuse of Graphical Representations” <http://www.montereyinstitute.org/courses/DevelopmentalMath/COU>
- [25] “Chart” <http://en.wikipedia.org/wiki/Chart>
- [26] Detecting and visualizing refactorings from software archives, Gorg, C. ; Weissgerber, P. *Program Comprehension*, 2005. *IWPC 2005. Proceedings. 13th International Workshop on Publication Year: 2005 , Page(s): 205 – 214.*
- [27] “FusionCharts by Features” <http://www.fusioncharts.com/explore/features/>
- [28] “Highcharts” [www.highcharts.com/](http://www.highcharts.com/)
- [29] <http://github.com/mstock/d3/wiki>.
- [30] “Data Driven Document” [d3js.org](http://d3js.org)