Name - Vasu Kalariya
Roll - PE29
Sub - BDA

Lab Assignment - 6

Aim: Big data analytics using big data visualization to D3.js for any application domain

Objective:
→ To explore visualization tool for big data
→ To learn D3.js for big data visualization

Problem statement: Big data analytics using big data visualization tool D3.js for application domain

Theory:

→ What is D3.js

D3.js is a javascript library for producing dynamic, interactive data visualizations in web browsers. It makes use of SVG (scalable vector Graphics), HTML5, CSS3. It is the successor of earlier framework

→ Features of D3.js
- Extremely flexible
- Easy to use & fast
- Supports large datasets
- Declarative Programming
- Code Reusability
- Wide variety of curve generating functions
- Associates data to an element or group of elements in html page

Vasu Kalariya (PE29)

D3.js selections: Ability to select move elements in webpage (modify/append/remove)
select(): select first matching element.
~~select()~~: ~~select~~
selectAll(): select all matching elements
eg:
Hello!

<script>
~~dv~~ alert (d3.selelted ("div").text());
</script>

Transitions: Changing from one state to another
eg. d3.select ("body").transition().style("background-color", "lightblue");

SVG Transformations:
i) Translate (tx, ty) : along resp axes
ii) Rotate (Cx, Cy) : Eg rotate(60)
(iii) scale (Sx, Sy) : Eg scale(10)
(iv) skew (Sx, Sy) : Eg skew(10)
Eg: SVG rectangle with translate
<body>
<svg width = "300" height = "300%.">
<rect x = "20%."
y = "20"
width = "60"
height = "60"
fill = "green"
transform = "translate (3030)>
</rect>
</svg>
</body>

Vasu Kalariya (PE29)

Input: Data can be loaded from various source like .jason, .csv files, RDBMS, NOSQL databases.

Output: Charts demonstrating all features mentioned above and all the charts mentioned in the presentation. Both Part I & Part II code & outputs

Platform: Ubantu / Windows.

Conclusion: Thus, learnt & implemented various visualizations with D3.js.

FAQs

1 What is D3.js stands for

→D3 stands for Data-Driven Document

2 How D3.js identify on which elements to operate?

D3.js uses following CSS-style selectors to identify elements on which to operate:
- Selection By Tag
- selection by Class Name
- selection by ID

3 Enlist Data-binding methods in D3.js

data(): Joins data to selected docs
enter(): create sel' w/ placeholder for missing elements
exit(): Removes nodes & add to exit selections
datum(): Injects data to selected element without completing a join