D3.js

Piyush Shinde[1,*]

[1] School of Informatics and Computing, Bloomington, IN 47408, U.S.A.
[*] Corresponding authors: piyushsshinde1992@gmail.com

Paper-2, March 27, 2017

Data Driven Documents (D3) is a open source JavaScript library used to create dynamic, interactive visualizations on a webpage. D3 uses HTML, SVG and CSS to create visualizations with a data-driven approach to Document Object Model (DOM) manipulation, enabling users to utilize the full capabilities of modern browsers and the freedom to design the right visual interface for their data [1]. This paper provides a brief introduction to D3 and its various features. It also discusses D3's use cases, advantages and limitations.

Keywords: Document Object Model, Data Visualization, Chart

https://github.com/cloudmesh/sp17-i524/blob/master/paper2/S17-IR-2035/report.pdf

1. INTRODUCTION

Several graphical forms can be used to envision large quantitative data sets such as graphs, charts, maps and diagrams. These data sets are increasing exponentially since the advent of the World Wide Web, making the task of efficient data visualization more challenging. As a result, web browsers are considered as ideal platforms for visualizing large data sets.

Designers often employ multiple tools simultaneously for building visualisations like HTML for page content, CSS for aesthetics, JavaScript for interaction, SVG for vector graphics [2]. One of the most important advantages is the uninterrupted cooperation between most of the technologies using the web as a platform, which is enabled by a document object model (DOM). The DOM reveals the ordered structure of a web page, aiding it's manipulations.

Data Driven Documents (D3) is an open source JavaScript library, which helps in efficient manipulation of the document object models based on data. It was released in 2011 by Heer, Ogievetsky and Bostock, then part of Computer Science Department of Stanford University, Stanford, CA 94305. D3 provides a toolkit for visualizing data using web standards such as HTML, SVG, and CSS.[1].

D3 resembles to document transformers such as jQuery that simplify the act of document transformation in web browsers. D3 uses the DOM's standard SVG syntax, that shows similarities with the graphical abstractions used in graphics libraries such as Processing and Raphaël. D3 is a generalization of Protovis, and through helper modules more complex visualizations can be achieved efficiently.

D3's main features include selections, transitions and update, enter and exit functions. We will glance through these features in the next section.

2. FEATURES

D3's basic operand is the selection. Operators act on selections, modifying content. Data joins bind input data to elements, producing enter and exit sub-selections for the creation and destruction of elements with respect to data. Animated transitions interpolate attributes and styles smoothly over time [2].

2.1. Selections

Selections allow the user to select and manipulate HTML elements in a very simple way. D3 adopts the W3C Selectors API that contain predicates to select elements by tag, class, unique identifier, attribute, containment, or adjacency [3]. Unique selection methods like union and intersection can be used on these predicates. Multiple operations can be performed after selecting an element by chaining the operations.

D3 provides select and selectAll methods for single and multiple element selections. The former selects only the first element that matches the predicates, while the latter selects all matching elements in document traversal order [2].

2.2. Data

Styles, attributes, and other properties are represented as functions of data in D3. They are simple, but powerful. D3 provides many built-in reusable functions and function factories, such as graphical primitives for area, line, and pie charts.

The data operator binds input data to selected nodes. Data is specified as an array of values such as numbers, strings or objects, and each value is passed as the first argument, along with the numeric index to selection functions. By default, data is joined to elements by index, the first element in the data array is passed to the first node in the selection, the second element to the second node, and so on. Once the data has been bound to the document, you can omit the data operator, D3 retrieves the previously-bound data. This allows you to recompute properties without rebinding [3].

2.3. Enter, Update and Exit

D3's Enter and Exit selections, aid users to create new nodes for incoming data and remove outgoing nodes that are no longer required.

When data is bound to a selection, each element in the data array is paired with the corresponding node in the selection. In case of fewer nodes as compared to elements of the data array, the extra data elements form the enter selection, that can be instantiated by appending to the enter selection.

Updating nodes are the default selection—the result of the data operator [3]. In case of skipped enter and exit selections, D3 automatically selects only the elements for which corresponding data exists. Properties that are constant for the life of the element are set once on enter, while dynamic properties are recomputed per update [2]. The initial selection can be divided into three parts: the updating nodes to modify, the entering nodes to add, and the exiting nodes to remove. Handling these three cases separately, precisely define the operations that run on each node, thereby optimizing performance and offering greater control over transitions.

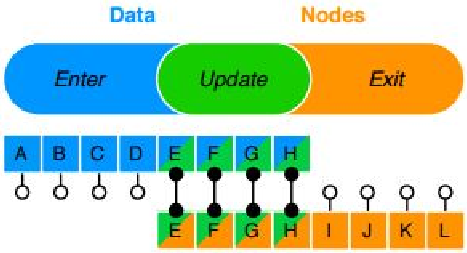

Fig. 1. New data (blue) joining old nodes (orange) results in three sub-selections: enter, update and exit [3].

2.4. Transitions

D3's focus on transformation extends naturally to animated transitions. Transitions gradually interpolate styles and attributes over time. D3's interpolators support both primitives, such as numbers and numbers embedded within strings (font sizes, path data, etc.), and compound values. D3's interpolator registry can even be registered to support complex data structures.

D3 reduces overhead by adjusting only the attributes that change and allows greater graphical complexity at high frame rates. D3 allows sequencing of complex transitions via events. D3 does not replace the browser's toolbox, but exposes it in a way that is easier to use [3].

3. USE CASES

The examples tab of the official D3 website displays more than 400 examples of data visualizations, built using D3 [4]. Examples include visualizations for newspapers, games, libraries and tools suggesting D3's reliability and ability to transform even the most complex data into clear visualizations. These visualizations run interactively inside web browsers and are available for anyone who wants to visualize data. Many of them are accompanied by a full documentation of the steps to manipulate documents based on data. The most commonly used types of data visualization are histogram, area chart, line chart, multi series line chart, bar chart, scatter-plot, pie chart, graphs, trees and maps [4]. D3 also shows visualizations updating in real time.

Here are three real world examples of simplified data visualizations using D3.

3.1. Wimbledon 2013 Data Visualisations

This example displays a series of ten different data visualizations of the Wimbledon tennis tournament created by Peter Cook [5]. The original data was collected from British tennis data website [6].

Amongst the 10 visualizations, 3 of them include a circular match tree, a bubble chart and a horizontal histogram. The circular match tree displays concentric circles labelled as the different rounds of the Wimbledon 2013. Each concentric circle displays the player names. The result was of each match in each round is displayed by hovering the mouse over the names. The winner's name is displayed in the center of the circle.

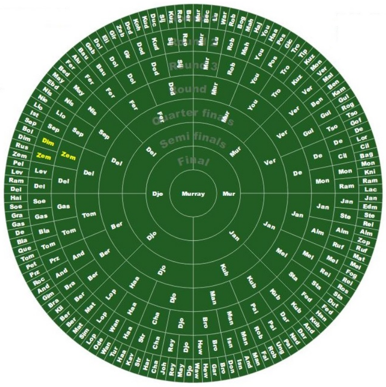

Fig. 2. Wimbledon 2013: Circular Match Tree [7].

The bubble chart displays pair of bubbles connected by an arrow. Each arrow indicates a match where the winner had a lower ATP ranking than the loser. The size of each bubble is proportional to the ATP points of the player. Hovering the mouse over a bubble displays both the players and their ATP points with the result of the match.

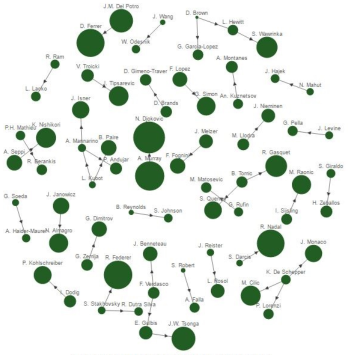

Fig. 3. Wimbledon 2013: Bubble Chart [8].

The histogram displays the list of the top 32 players of Wimbledon 2013. It displays the number of matches won, sets won, games won and ATP ranking of all the 32 players by clicked on the respective tabs [9].

3.2. Earthquakes in Chile since 1900

This example helps us visualize the most important seismic events in Chile since 1900 [10]. The earthquake data was retrieved from the ANSS Composite Catalog and joined with the Centennial Catalog from the USGS.

A seismic event is displayed as a circle with it's occurrence year. The radius and the color of the circles are a function of the earthquake magnitude.

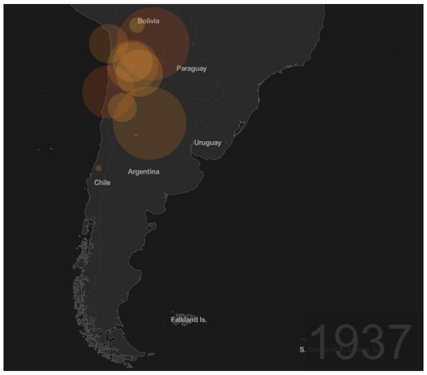

Fig. 4. Earthquakes in Chile since 1900: Map [10].

3.3. Visualizing the Racial Divide

This example displays a visualization that highlights the impact of the segregation that exists in many US cities today based on race using d3 and force-directed maps [11]. The cities include Milwaukee, Chicago, St. Louis, Syracuse, Dayton, Pittsburgh, Denver, Columbus, Kansas City, Oklahoma City, Wichita, Memphis, Baltimore and Charleston. Each city is made up of tracts from the 2010 Census.

The Census tracts are pushed away from neighboring tracts based on the change in proportion of white and black populations between each neighboring tract.

Tracts having similar racial mix as their neighbors form groups. Spaces occur where there is a significant change in the racial makeup between neighboring tracts. The space is proportional to the change in racial composition between neighbors.

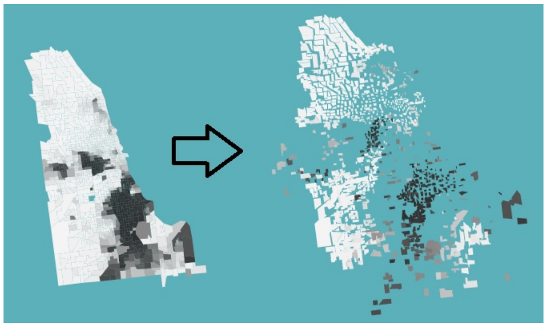

Fig. 5. Visualizing The Racial Divide: Chicago [12].

4. BENEFITS AND LIMITATIONS

D3 is an open source project, with it's source code readily available on Mike Bostock's github account [13]. It is free to use, which sets it apart from similar data visualization JavaScript libraries such as amCharts and FusionCharts, which need paid licenses. D3's development is supported by a big online community, where users can contribute to it's examples making it an exhaustive resource of various data visualizations.

D3's is also a drawing library unlike conventional data visualization tool-kits, that simply create data visualizations. D3 supports dynamic visualizations, as opposed to mere static visualizations.

D3 can be started using just one line of code, which is not the case with other data visualization tool-kits, often requiring a long installation procedure and periodic updating. D3's compatibility with web standards provide an added advantage that visualizations can be shared and viewed without the need for additional plug-ins. It is based on JavaScript which is compatible with browser's built-in debugger. This facilitates it's fast and easy debugging.

Amongst several advantages D3 has a few limitations. It's learning curve can be fairly steep. This is partly explained by the need for in depth knowledge of web standards, especially of SVG. Furthermore, the helper modules required for data transformations need studying too.

Unlike JavaScript libraries such as Google Charts, D3 does not offer in-built (standard) charts. This makes it less efficient as compared to other tool-kits with custom graphs.

5. USEFUL RESOURCES

The complete documentation of D3.js, including instructions to download and install it's latest release, examples, and several tutorials, in a systematic method are available on the main website of D3.js [3]. It is also a useful resource to D3's core concepts, techniques, blogs, books, courses, talks, videos and meetups [14].

The paper titled "D^3: Data-Driven Documents" authored by the creator of D3, compares D3 to existing web-based methods for visualization. The paper demonstrates how D3 is at least twice as fast as Protovis, through performance benchmarks. It also describes D3's potential for dynamic visualization [2].

Another website provides a complete path to create interactive visualization using D3.js [15]. It provides few real world examples and steps to create basic pie charts, animated bar charts and map.

6. CONCLUSION

The increasing popularity of JavaScript, has caused a major shift in the direction of web development with reduced dependencies on plug-ins. Consequently, developers are trying rely on the web browsers alone to avoid dependence on external plug-ins. This reduces the possibility of bugs or incompatibility issues as well the need to update. As an added benefit, developers can work with the existing web standards, increasing D3's compatibility with technologies like HTML, CSS and JavaScript. This makes preparation of data visualizations easy and readily available to everyone. D3.js is an toolkit for efficiently creating complex visualizations based on large datasets. It can be used for solving real world problems by creating visualisations to better infer trends in complex large data sets.

7. ACKNOWLEDGMENTS

This project was a part of the Big Data and Software and Projects (INFO-I524) course. I would like to thank Professor Gregor von Laszewski and the associate instructors for their help and support during the course.

REFERENCES

[1] Mike Bostock, “Home,” Web Page, accessed: 2017-03-24. [Online]. Available: https://github.com/d3/d3/wiki

[2] M. Bostock, V. Ogievetsky, and J. Heer, “D3: Data-driven documents,” *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2011. [Online]. Available: http://vis.stanford.edu/papers/d3

[3] Mike Bostock, “D3 Data Driven Documents,” Web Page, accessed: 2017-03-24. [Online]. Available: https://d3js.org/

[4] ——, “Gallery,” Web Page, accessed: 2017-03-24. [Online]. Available: https://github.com/d3/d3/wiki/Gallery

[5] Peter Cook, “Wimbledon 2013 Data Visualisations,” Web Page, accessed: 2017-03-24. [Online]. Available: http://charts.animateddata.co.uk/tennis/index.html

[6] Joseph, “Tennis-Data.co.uk,” Web Page, accessed: 2017-03-24. [Online]. Available: http://www.tennis-data.co.uk/wimbledon.php

[7] Peter Cook, “Wimbledon 2013 Circular Match Tree,” Web Page, accessed: 2017-03-24. [Online]. Available: http://charts.animateddata.co.uk/tennis/matchTree.html

[8] ——, “Wimbledon 2013 David Goliath,” Web Page, accessed: 2017-03-24. [Online]. Available: http://charts.animateddata.co.uk/tennis/davidgoliath.html

[9] ——, “Wimbledon 2013 Top 32 Players,” Web Page, accessed: 2017-03-24. [Online]. Available: http://charts.animateddata.co.uk/tennis/top32.html

[10] Pablo Navarro, “Earthquakes in Chile since 1900,” Web Page, accessed: 2017-03-24. [Online]. Available: http://pnavarrc.github.io/earthquake/

[11] Jim Vallandingham, “Visualizing The Racial Divide,” Web Page, accessed: 2017-03-24. [Online]. Available: http://vallandingham.me/racial_divide/

[12] ——, “Visualizing The Racial Divide,” Web Page, accessed: 2017-03-24. [Online]. Available: http://vallandingham.me/racial_divide/#ch

[13] Mike Bostock, “Mike Bostock,” Web Page, accessed: 2017-03-24. [Online]. Available: https://github.com/mbostock

[14] ——, “Tutorials,” Web Page, accessed: 2017-03-25. [Online]. Available: https://github.com/d3/d3/wiki/Tutorials

[15] Analytics Vidhya, “Newbie to D3.js Expert: Complete path to create interactive visualization using D3.js,” Web Page, accessed: 2017-03-24. [Online]. Available: https://www.analyticsvidhya.com/learning-paths-data-science-business-analytics-business-intelligence-big-data/newbie-d3-js-expert-complete-path-create-interactive-visualization-d3-js/