

The Power of D3.js – Data Visualization

S. Varun Prakash ¹, M. Mohamed Suhal ², M. Charan Mithesh ³, Mrs. R. K. Santhiya ⁴

¹B.Tech, Computer Science & Engineering, Manakula Vinayagar Institute of Technology, Puducherry, India.

²B.Tech, Computer Science & Engineering, Manakula Vinayagar Institute of Technology, Puducherry, India.

³Assistant Professor, Dept. of Computer Science & Engineering, Manakula Vinayagar Institute of Technology, Puducherry, India.

Abstract - This paper explores about building complex front-end systems for businesses or to represent a general cause. The data dashboard that is built using D3.js revolves much around the principles of design to effectively showcase the ideas - to explore, analyze and present data in the most meaningful way. Design decisions are precisely made to create exciting visualizations and to differentiate between good & bad designs.

1. INTRODUCTION

The Term “Data Visualization” is the representation of a situation, an object or a set of information as a chart or another image. Basically, abstracting the content in visual form. As a whole, visualization is used for 3 most common cases such as (i) to explore for finding the unknown, (ii) to analyze for testing a hypothesis and (iii) to present data for telling a story).

2. LITERATURE SURVEY

A. Explore

In 1887, Eadweard Muybridge, needed to find out if the legs of a galloping horse have all 4 of its foot in the air which is critically hard for the naked eye to figure it out. A series of experiments with photographs were used to find out the results, which seemed to prove the matter once and for all.

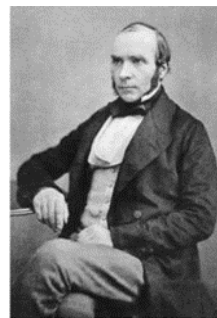


(Figure. 1)

In the image - Eadweard Muybridge (An English-American Photographer).

B. Analyze

Visualizations are mostly made to test the hypothesis that we have about the state of the world.



(Figure. 2)

In 1854, John Snow used a heat map of cholera cases in areas of London to prove his hypothesis that these deaths were spread by water pumps in the area rather than decaying organic matters as the world initially believed.

C. Present

The goal is to communicate some kind of story and hopefully inspire people to take some action. The New York Times often run interactive data visualizations on their website. These stories are accompanied by a block of text, and their related viewings are made to support it better.



(Figure. 3)

3. EXISTING SYSTEM

By the mid-1800s, the circumstances for the accelerated growth of visualization had been set — an "ideal storm" for data graphics. Official state analytical offices were installed throughout 13 Europe, in recognition of the increasing greatness of mathematical knowledge for social planning, industrialization, business, and shipping. Statistical theory, initiated by Gauss and Laplace, and lengthened to the social realm by Guerry and Quetelet, presented the means to make

sense of large bodies of data. What started as the Age of Enthusiasm for graphics completed with what can be called the Golden Age, with exceptional beauty and many discoveries in graphics and thematic cartography. So varied were these developments that it isn't very easy to be comprehensive, but a few themes stand out.

4. PROPOSED SYSTEM

A. Need for Visualization

- Visualization is suitable when there is a need to augment human capabilities rather than replacing people with computational decision-making systems, aka Machine Learning.”

- In other words, computer-based visualizations are systems that provide a visual representation of datasets designed to help people carry out tasks more efficiently.

- Data Visualization acts as an external representation to effectively replace cognition with perception.”

B. Power of D3.js

D3 (Data-Driven Documents) stands as the world's premier JavaScript library to visualize data using web standards. Ranks top among its number of alternative competitors such as FusionCharts, HighCharts, Chart.js, Google Developers Charts due to its flexibility, elegance & its support in the community.

C. Peculiarities of D3.js

- Can load our data from any source we like.
- D3 patches this data to shapes on the screen.
- Make completely new visualizations that we have never imagined before.
- D3 handles vast volumes of data efficiently.
- Can easily add smooth transitions for updating elements.
- Works with clean, well-designed code.
- Can find pre-built code to use in our projects.
- Everything is open-source, so we don't need to worry about licensing.
- Broad user base makes it easy to collaborate with others.

5. SYSTEM DESIGN

“Design tips and technical guidance for better visualizations”

A. Scalable Vector Graphics

These are shapes drawn with the code. SVG includes rectangles, circles, ellipses, lines, texts, and paths. Since these are drawn using geometrical calculations, they do not lose the quality even if zoomed in or zoomed out. Replacing raster with vectors dramatically reduces file size without losing quality and increases the performance over the network.

With D3.js, SVG shapes can be added and attributes are set based on the needs of visualization.

B. Data that reflects

SVG shapes and colors can be derived based on the data we provide. It may be pre-defined in code or obtained through input events from 3 users or external data sources such as CSV TSV JSON GEOJSON / TOPOJSON.

C. Scales in D3

“Scales are used to map from an input domain to an output range.”

Scales differ based on the data,

- Linear Scales - For continuous data
- Logarithmic Scales - For data that shows exponential growth.
- Time Scale – For time-specific data
- Ordinal Scale – To assign color schemes for categorical data.
- Band Scales – To align the spacing between data along an axis.

D. Data & Its categorization

Data under visualization basically categorizes under three types,

- Quantitative Data.
- Ordinal Data.
- Categorical Data.

Brief Explanation about the 3 categories:

Quantitative Data: These datasets involve numerical data mostly such as population, life expectancy, count, rate, etc.

Ordinal Data: These are datasets that can be related to one another, such as XS, S, M, L, XL as one of the sizes indicates its relation & characteristics with the other size.

Categorical Data: These are datasets that cannot be categorized under anything rather than their own type, such as Asian countries, African countries, American countries, etc.

E. Visualization Supporters

D3 characteristics like Axes, labels, tooltips, and margins add detail & reduce the visual jargon to the best possible. D3 has min and max function to determine the least as well as max value present in the dataset. D3 library also has a range of color schemes to choose from and use in our visualizations rather than making us selecting the colors from scratch.

F. Design Basics & Design Clarity

Design factors that improves data visualization.

- Make the data easy to interpret.
- Build credible visualizations.
- Keep the focus on important factors.

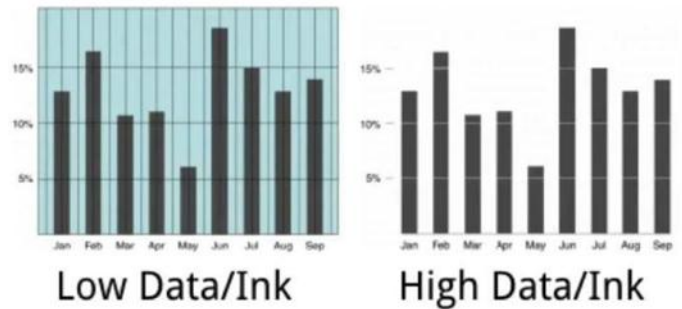
G. Edward Tufte's Design Principles.

- Graphical Integrity.
- The lie factor.
- Chart junk.
- Data to ink ratio.

Graphical Integrity:

The bar graph on the right, as well as left, conveys the same analytics, but the bar graph on the left exaggerates the difference between 35% and 39.6%, whereas the graph on the right doesn't give a shocking difference but conveys the data more truthfully.

Data-to-ink Ratio:



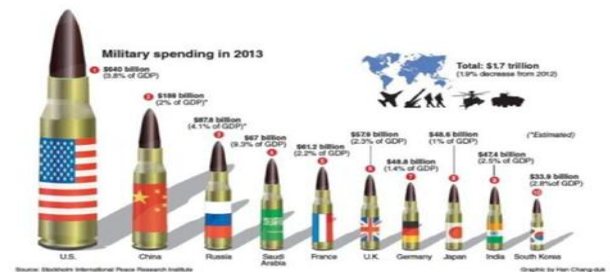
(Figure. 7)

The bar chart on the right is more precise than the bar chart on the left. Tufte's design principle utilizes data-to-ink ratio to measure the chart junk.

H. Goals for Visualization

- Grab user's attention.
- Encourage further exploration.
- Build new and exciting types of visualization.

There are a few counter-examples of good visualization that don't conform to Tufte's principle to achieve the goals of visualization better. Military Spending across countries.



(Figure. 8)

The Lie Factor:

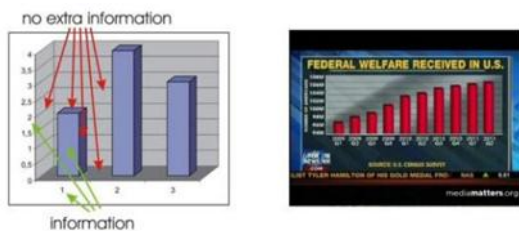
Both the Pie charts below convey the same statistics. Since the pie chart on the left is skewed, the difference between the green and blue portion of the chart doesn't matter much.



(Figure. 5 - The true analytics is conveyed on the right with no skewing in viewing perspective.)

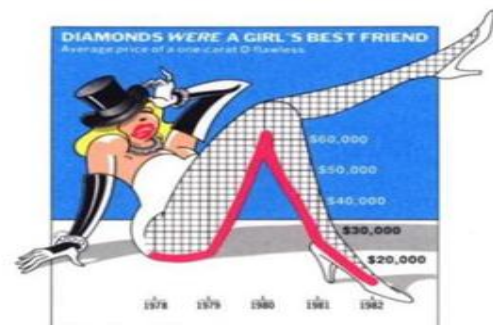
Chart Junk:

These are elements that distract the user from visualizing the data. 3D and background graphics are unnecessary framing elements and are considered as chart junk.



(Figure. 6)

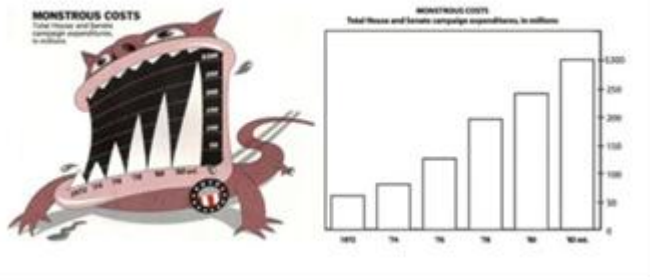
i. A Visualization that doesn't follow Tufte's Design principle but achieves the goal of visualization better.



(Figure. 9)

ii. Visualization of diamond's price over sometime using chart junk.

iii. Visualization of monstrous cost in campaign expenditure. Sometimes, conveying the data truthfully without chart junk may seem less impressive.



(Figure. 10)

iv. The choice of design principles with or without chart junk is based on the idea that we are trying to convey the users.



(Figure. 11)

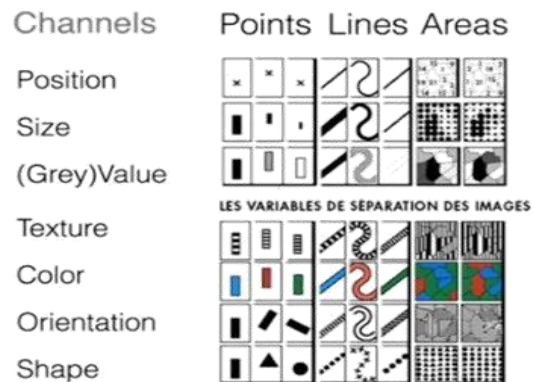
The node graph may seem more distracting, but that's what makes the visualization interactive and more enjoyable.

I. Design Guidelines

Jacques Bertin introduced a handful of design techniques to visualize the reference models better. His visual variables include,

- Position.
- Size.
- Shape.
- Value.
- Orientation.
- Texture.
- Color.

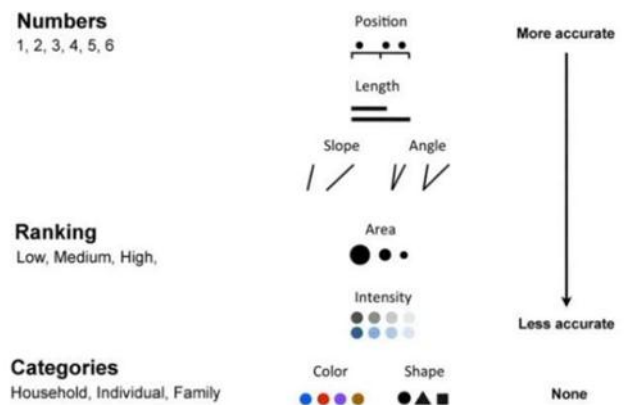
These design variables help to build the design architecture better & more precise based on the data.



J. Bertin, 'Semiology of Graphics' 1967

(Figure. 12)

Adapting to Design guidelines based on the type of data that we are trying to visualize.



(Figure. 13)

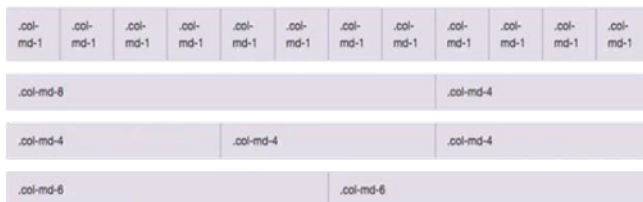
J. Bootstrap Grid System

As we need our visualizations to be responsive in all hand-held devices and desktop hardware, we adapt to a grid system to structure our pages. It is a CSS framework to build sites faster and easier to create responsive, mobile-first front end web applications.

	Extra small devices Phones (<768px)	Small devices Tablets (≥768px)	Medium devices Desktops (≥992px)	Large devices Desktops (≥1200px)
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints		
Container width	None (auto)	750px	970px	1170px
Class prefix	.col-xs-	.col-sm-	.col-md-	.col-lg-
# of columns	12			
Column width	Auto	~62px	~81px	~97px
Gutter width	30px (15px on each side of a column)			
Nestable	Yes			
Offsets	Yes			
Column ordering	Yes			

(Figure. 14)

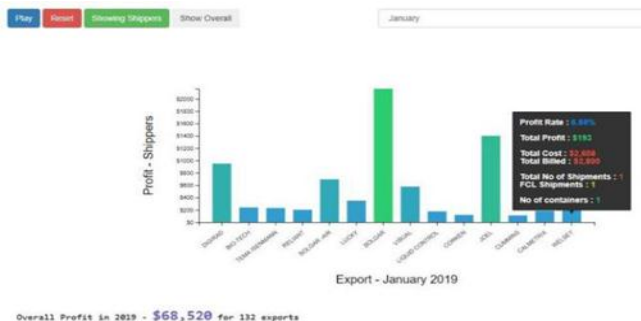
For different screen sizes, we use columns to layout the structure (default of 12 columns). Beyond 12 columns would result in wrapping thereby, staying within the viewports of the screen size.



(Figure. 15)

K. Data Interactivity

Factors that make the visualization more interactive & live for the users are transitions when updating the visualization, time intervals-based looping, dynamic updates based on user click events, adding a legend to better understand the visualization, tooltips for more info, jQuery & Bootstrap UI elements to help the end-user better specify his/her need, etc.



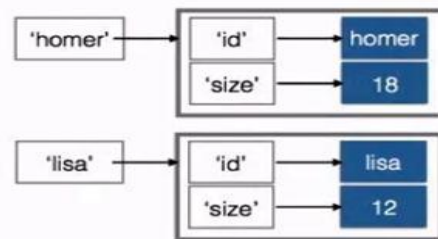
(Figure. 16)

L. Data Structure

As we have several forms of visualization, we need to consider the data structure for each layout. The 2 main data structures of JavaScript – Arrays and Objects. Storing data in array uses index to map the data whereas storing data in objects uses key value pair to map the data. Cases may occur as an array of objects or as an object of arrays.

Objects of Arrays

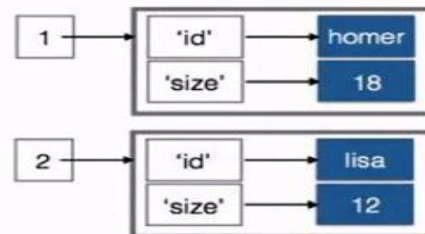
```
dataMap = {
  'homer': {
    'id': 'homer', 'size': 18},
  'lisa': {
    'id': 'lisa', 'size': 12}
}
```



(Figure. 17)

Arrays of Objects

```
data = [
  {'id': 'homer', 'size': 18},
  {'id': 'lisa', 'size': 12}
]
```

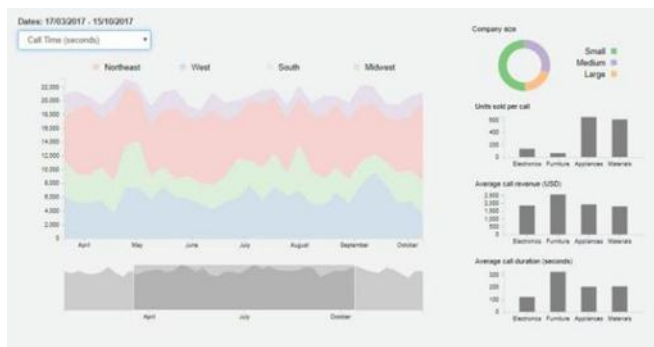


(Figure. 18)

Finding the size of the homer is more comfortable on the Object of Arrays (uses the key value to map the data instantly) rather than the Array of Objects (requires brute-forcing to loop through each index of array elements until finding the homer). D3 supports the nesting data function to do this conversion of data-structuring to quickly map the values that improve the overall efficiency of visualization.

J. Object Oriented Programming

We are making use of OOP concepts to render several visualizations as objects in one web application.



(Figure. 19)

OOP based Interactive D3 dashboard involving Area Chart, Donut Chart, Bar Graphs & Brushes

6. CONCLUSION

Massive data is being piled up every day at unbelievably rapid speed. Per statistics of 2015, 5.6 zettabytes of data production are estimated at an annual rate, which was three times the rate of growth in 2012. The data must be aggregated, classified, made interpretable, and then analyzed and acted on to provide any meaningful value & valuable insights. This is where data-visualization actions and allow the organization leads to access and evaluate data in real-time so that they can make highly informed conclusions quickly.

REFERENCES:

- [1] D. Kornack and P. Rakic, "Cell Proliferation without Neurogenesis in Adult Primate Neocortex," *Science*, vol. 294, Dec. 2001, pp. 2127-2130, doi:10.1126/science.1065467.
- [2] M. Young, *The Technical Writer's Handbook*. Mill Valley, CA: University Science, 1989.
- [3] R. Nicole, "Title of paper with only first word capitalized," *J. Name Stand. Abbrev.*, in press.
- [4] *Design Guidelines & Practices in D3* by Adam Jones, Data Visualization Engineer
- [5] FreeCodeCamp - A non-profit organization & community for web developers & learning web platform.
- [6] *Block Builder* by Ian Johnson for visualization references & practices (<https://blockbuilder.org/>)
- [7] *Blocks.org* by Mike Bostock, Data Visualization Specialist & one of the key founder of D3.js (<https://bl.ocks.org/>)
- [8] *Data Visualization* - Wikipedia (https://en.wikipedia.org/wiki/Data_visualization)
- [9] *Visual design rules & principles* by Joe Natoli - Design Veteran in UI + UX (<https://www.behance.net/joematoli>)
- [10] *Data Driven Documents (D3.js) - Official Documentation / GitHub Repository* (<https://github.com/d3/d3/wiki>)