ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 7, Issue 11 , November 2020

Exploring the subtle differences between JavaScript Data Visualization libraries

Varun Agarwal, Abhishek Chauhan

MTS I-Sys Engrg, Verizon Data Services India, Chennai, India
Analyst, Deloitte USI, Bangalore,India

ABSTRACT: As we prosper in the digital age, ignoring the importance of data in our lives is impossible. Data is not only critical to large multinational corporations like Facebook or Google but even for smaller organizations. A study suggests that 99.5% of the data collected by our systems go to waste because it is not appropriately processed [1]. Effective data visualization techniques provide clarification, which allows users to understand data passively. Various companies want their data visualization to be incorporated into their user interface to make the user experience flawless. JavaScript uses various open-source libraries, allowing customizations as per the user requirements. There exist many JavaScript libraries, which provide us with different graph types incorporated for data visualization in web applications. However, the front-end development team always has this conflict for which particular library to be used. This paper compares the five most popular JavaScript data visualization libraries, namely D3.js, Chart.js, Three.js, Highcharts, and Google Charts.

KEYWORDS: Front-end, JavaScript, data visualization, data analysis, graphs, and charts

I. INTRODUCTION

Data analysis has become an essential aspect of the IT industry in this growing era of technology. Representation of data in pictures, graphs, maps, or user interface is termed as data visualization. Data visualization is done mostly using various types of graphs, depending upon the size of the data. Data visualization helps in understanding the patterns, trends, or outliers in data. Many frameworks are available in JavaScript, which are used in the development of web applications. There are different data visualization libraries in JavaScript, which allows the viewers to contemplate the importance of the data. Many developers think of data visualization as intricate interactive graphics of extraordinary complexity. Creating compelling visualizations, however, does not require artistic skill or programming expertise. On the other hand, when you observe data visualization's primary purpose, it helps users understand data with utmost simplicity. [1]

Nowadays, we cannot imagine any front-end design without charts and graphs. Therefore, JavaScript data visualization tools are required now more than ever. Different teams have different data types, and all these rely on different charts and graphs to help them understand their data. There are many data visualization libraries available on the internet. These libraries have different supported chart types, data binding, interactivity, rendering technologies, or licenses [2] [3]. Depending upon the client's requirements and purpose, any library can render and visualize various charts.

II. DATA VISUALIZATION LIBRARIES

In this paper, we are considering D3.Js, Chart.Js, Three.Js, Highcharts, and Google Charts. Among these, to find the most popular JavaScript charting libraries used by the industry, we collected the number of stars and forks for each library from GitHub, the largest open source community, and weekly downloads via Node Package manager (npm).

A. D3.js

D3 stands for Data-Driven Documents, developed in 2011 as a successor for the Protovis framework. It is a JavaScript library that is primarily used for binding arbitrary data to a DOM (document object manipulation), and then data-driven transformation is applied to the document. It allows users to produce dynamic, interactive data visualizations in web browsers. Here, the user himself creates them dynamically with the help of definitions and their attributes. The D3.js library uses in-built functions to select elements, create SVG objects, customize them, or add dynamic animations, effects, or tooltips. [4]

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 7, Issue 11 , November 2020

B. Chart.js

Chart.js is an open-source JavaScript library created by Nick Downie in 2013. It is the second most popular data visualization library on GitHub compared based on the number of stars after D3.js. Data in Chart.js can be visualized in eight different chart types, namely bar, line, area, pie (doughnut), bubble, radar, polar, and scatter; each of them is customizable. It is a community maintained project, which accepts contributions from all developers. [5]

C. Three.js

Three.js is a 3-Dimensional JavaScript library first released in April 2010 by Ricardo Cabello to GitHub and, like most of the libraries, is available under the MIT license. A lightweight cross-browser library can create and display 3D computer charts on a webpage. It may be used along with the HTML5 canvas, SVG element, or WebGL. With Three.js, we can create cameras, objects, lights, materials, and more, using renderers like Canvas 2D, SVG, and CSS3D renderers. [6]

D. Highcharts

Highcharts is an SVG-based, multi-platform JavaScript charting library developed by Torstein Hønsi in 2009, sold in the market as a product. This JavaScript library provides interactive charts for mobile and web projects, is easy to use, and has a robust feature set and thorough documentation. The product comes with an inbuilt debugger that can swiftly complete the development because of constant checks on warnings and errors. The error logs provided are lucid on how to solve the error. [7]

E. Google Charts

The Google Company developed the Google Charts library. Its UI is highly interactive and makes it easier to render numerical data into a graphical visualization. These are then represented as various in-built charts, which can vary from a simple bar graph to a complex timeline chart. It does not require any separate plugins to render the graphical elements but uses the raw HTML5 or SVG techniques. These charts can be customized as per different business use cases. It can also include graphical parts in the vector representation directly in an HTML document autonomously using VML (Vector Markup Language) on the platform selected. [8]

III. FACTORS TO CONSIDER WHILE CHOOSING THE LIBRARY

A) Cross-browser compatibility

Cross-browser compatibility is a significant factor while selecting a charting library. Charting libraries that support most of the web browsers are always the right choice. You can look at the browser usage statistics (Fig 1) to decide which library to use that is, if we need a charting library that's compatible with all browsers or just a few, the most commonly used browsers depend on our target audience. If you are developing a web/mobile application for a government handled organization, then the chances are that they still use older versions of IE, but at the same time, you may not be able to use the latest features of the charting libraries, therefore in this case libraries that support only modern browsers might not be a good option.

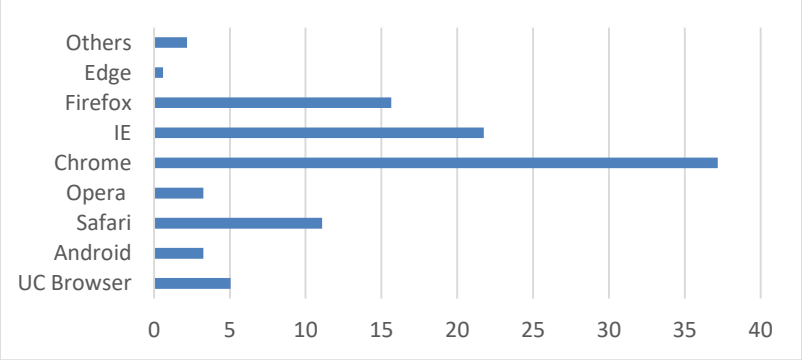

Fig 1: Usage stats of most popular browsers [9]

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 7, Issue 11 , November 2020

B) Cross-device compatibility

As per a research analysis, 56% of consumer traffic to websites in the US is mobiles. [10] Hence, it is crucial to consider whether your application is used mainly on desktop or other technological devices. If your application is being developed for a fixed resolution, most of the charting libraries will work for your application. However, if you are developing a responsive cross-device application (Fig 2 and Fig 3), you would like to prefer a responsive charting library in all the scenarios. Your application has to be developed to interact with a multitude of devices and screen sizes. To ensure a consistent experience, developers have to consider cross-device compatibility as a significant factor.

C) Variety of Charts

Charts are most commonly used for four types of analysis, displaying the relationship between data points, a comparison of data, a distribution of data, or a composition of data. You must select a library that satisfies all your data visualization needs but does not hamper your application's performance. You would not prefer to use multiple charting libraries in a single application as it might affect your style consistency for the user interface, so selecting a comprehensive and crisp charting library, which also has the charts you need, is a must.

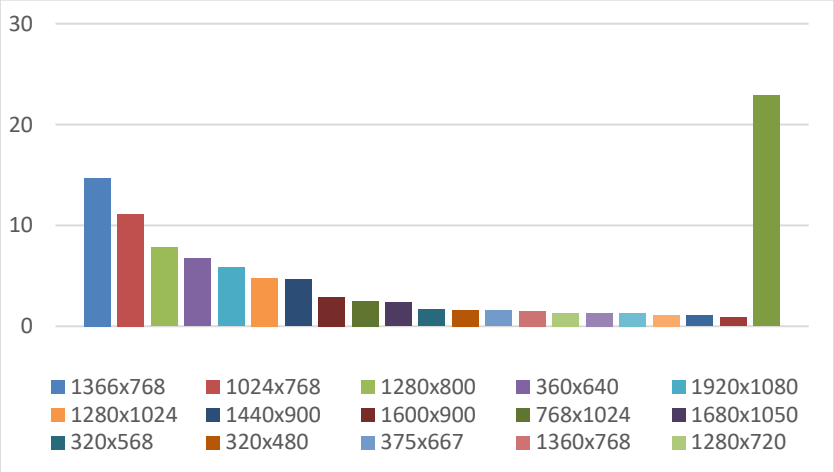

Fig 2: Usage statistics of different screen types from 2010-2020 [9]

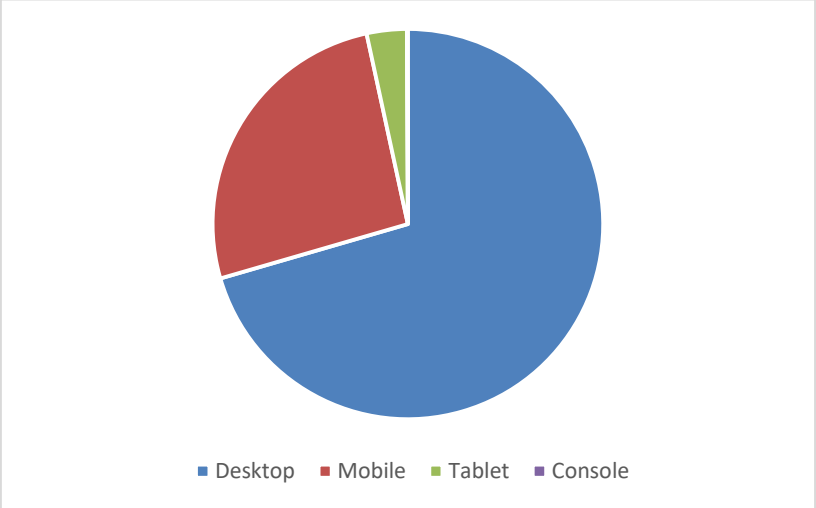

Fig 3: Usage statistics of different platforms from 2010-2020 [9]

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 7, Issue 11 , November 2020

D) Customization and performance

Customization is also an essential factor whenever there is a need to select a JavaScript library. It is always good to have libraries that allow maximum customization with minimal effort. Features such as displaying data or charts, legends, or design of the tooltips depend upon the charting library's customizable aspect. The library should support the user interface (themes, color scheme), but it should have a meaningful user experience (hover, click, or a keypress).

It is a user's general expectation that the application should be interactive and have a smooth performance. Hence, it is vital to use a library, which runs as efficiently as possible. The library's performance depends on various factors like the library's size, memory usage, garbage collection, and the number of browser repaint cycles [11].

E) Data Input and Output Forms

When it comes to passing data into these libraries, JSON (JavaScript Object Notation) is the one standard format compatible with almost all the libraries. However, some projects might require data to be passed as XML (Extensive Markup Language) files or in XML format. The library should support the most powerful, versatile data formats, such as combinations of JavaScript API/XML/JSON/CSV.

When the data size becomes enormous, the rendering takes quite a moment, and dynamic layouts become impossible. Javascript libraries should not limit the amount of data you can load, process, or the number of elements you can display. Even if you have displayed too much data in the browser, the application's load time should not be affected. Hence, supporting large data size is also considered to be a factor while choosing a charting library.

Different features can be found in the different libraries and offer a wide range of export options such as downloading and sharing over the web, size, format, etc. This point plays a significant role when making web apps, which are dashboards. The most common export formats are JPEG, PNG, PDF, GIF, and SVG.

F) Front end framework supported

There are over 20+ front-end frameworks/libraries available (Fig 4) in the market as of September 2020 [12]. Most of the charting libraries are framework specific. Most libraries have ready-to-use plugins for various frameworks included in a familiar development environment with a single code line. It also depends upon the framework whether the code is being executed on the client-side or the server-side. Therefore a developer needs to make sure that the library he opts for is compatible with the framework he uses.

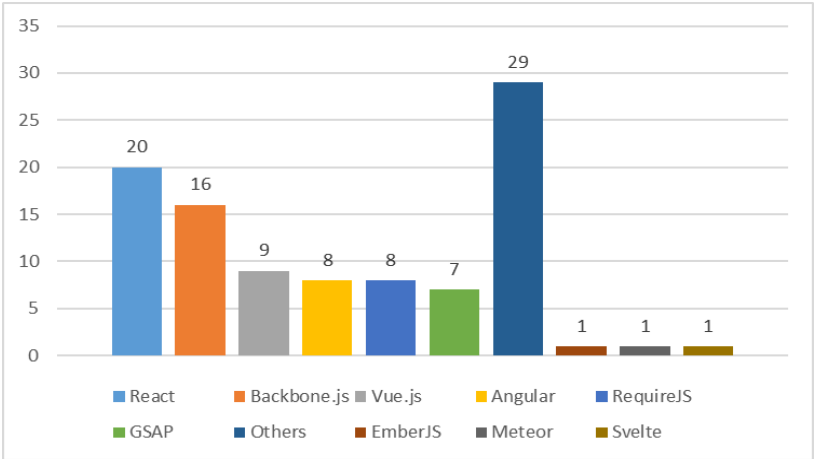

Fig 4: Usage statistics of Frontend Frameworks [12]

G) Learning Curve

Selecting a library that is easy to learn, and has a logic-based implementation should always be a priority. Some data visualization libraries have a steep and robust learning curve, while others are lucid enough. If you are running on a deadline or using a library for the first time, it is not recommended to go for a complex library. On the contrary, if your primary goal is to develop a beautiful UI with expressive content and have much time to explore, you can choose a complex chart as well.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 7, Issue 11 , November 2020

H) Documentation, Support, and Resources

Documentation can be referred to as any written material/illustration used to describe how the library operates or how it can be used [13]. It also instructs on the other aspects such as its components, installation, maintenance, and use. Documentation, along with demo/examples, can also help in the implementation of the library.

A paid library has its support available via the company's helpdesk or customer-service line and a community that has already subscribed to it. However, if a library is open-source or free to use, it is unnecessary that its creator would provide help; instead, it can be found on various discussion forums or the developer's community. A big community supports a library that has the edge over others as it is less likely to get abandoned in the future. It is crucial to consider the availability of resources and support when choosing a JS chart library.

I) Price and License

A few libraries provide the source code and the product when you purchase the licensed version, but the purchase does not guarantee you the freedom to use the library whatever way you want, i.e., the library can be used only within the terms of its purchase license. Different factors affect the license terms and pricing, such as the number of users, type of application, number of servers, chart types, features, customizability, API access, etc. [14]. The library should also have the licensing options that are required and, at the same time, should be within the project's price point. While comparing the prices, the total cost of ownership, including maintenance, support, and upgradation, should be considered and not just the purchases upfront cost.

IV. COMPARATIVE ANALYSIS

Table 1 : shows a comparison of various libraries while considering different factors, a few discussed above.

Table 1: Detailed analysis of Javascript Data Visualization libraries (values are as of October 2020)

Factors		D3.Js [4]	Chart.Js [5]	ThreeJs [6]	Highcharts [7]	Google Charts [8]
Cross-Browser Compatibility	Chrome	YES	YES	YES	YES	YES
	IE/Edge	YES	YES	YES	YES	YES
	Firefox	YES	YES	YES	YES	YES
	Safari	YES	YES	YES	YES	YES
Cross-Device Compatibility	Desktop	YES	YES	YES	YES	YES
	Mobile	YES	YES	YES	YES	NO
	Tablet	YES	YES	YES	YES	YES
Variety of Charts	Bar	YES	YES	NO	YES	YES
	Line	YES	YES	NO	YES	YES
	Pie	YES	YES	NO	YES	YES
	Mixed	YES	YES	NO	YES	YES
	Donut	YES	YES	NO	YES	YES
	Timeline	YES	NO	NO	YES	YES
	Scatter	YES	YES	NO	YES	YES
	Radar	YES	YES	NO	YES	YES
	Gantt	YES	NO	NO	YES	YES

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 7, Issue 11 , November 2020

	Treemap	YES	NO	NO	YES	YES
	Heatmap	YES	NO	NO	YES	YES
	Bubble	YES	YES	NO	YES	YES
	3D Charts	YES	NO	YES	YES	NO
	Box & Whisker	NO	NO	NO	YES	YES (Images only)
	Histograms	NO	NO	NO	YES	YES
	Choropleth map	YES	NO	NO	YES	YES
	Gauges	NO	NO	NO	YES	YES
	Funnel	YES	NO	NO	YES	NO
Customization	Drilldown Charts	YES	NO	NO	YES	NO
	Region Selection	YES	NO	NO	YES	NO
	Panning	YES	YES	YES	YES	NO
	Zooming and Scrolling	YES	YES	YES	YES	NO
	Drag	YES	YES	NO	YES	NO
	Crosshair	YES	YES	NO	YES	YES
	Label Customization	YES	YES	YES	NO	NO
	Legend Customization	YES	YES	NO	YES	YES
	Tooltip Customization	YES	YES	YES	NO	YES
	Axis Customization	YES	YES	YES	YES	YES
	Animation	YES	YES	NO	YES	YES
	Inverse Axis	YES	NO	NO	YES	NO
	Annotations	YES	YES	NO	YES	YES
	Themes	NO	NO	NO	YES	NO
	Accessibility	YES	YES	NO	YES	NO
Performance		Slow and Heavy	Fast and Lightweight	Slow and Heavy	Fast and Lightweight	Fast and Lightweight
Input Data Format		CSV, XML, JSON	Javascript API	JSON, OBJ	CSV, XML, JSON	CSV, JSON
Exporting Options		PNG, JPG, SVG, PDF	NO	glTF (GL Transmission Format)	PNG, JPG, SVG, XLS, PDF	PNG
Front-end	React	YES	YES	YES	YES	YES

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 7, Issue 11 , November 2020

Frameworks Compatibility	Angular	YES	YES	YES	YES	YES
	Vue	YES	YES	YES	YES	YES
	Vanilla JS	YES	YES	YES	YES	YES
Learning Curve		HIGH	MEDIUM	HIGH	LOW	LOW
Documentation and Resources		Official Documentation, API references, and Sample Code	Getting started guide, API references, and Sample Code	Official Documentation and Sample Code	Getting started guide, API references, and simple examples.	Getting started guide, API references, and Sample Code
Support		Github Issues and community forums	Github Issues and community forums	Github Issues and community forums	Realtime chat, personalized tech support, Github issues, and community forums	Github Issues and community forums
Price		Free	Free	Free	Starting from $778	Free
License		BSD-3-Clause License	MIT License	MIT License	Proprietary	Apache 2.0
GitHub Data	Stars	93.8k	50.4k	63.6k	9.7k	1.9k
	Language	Javascript	Javascript	Javascript	TypeScript, JavaScript and HTML	Dart
	Contributors	137	336	1352	116	8
	Forks	22.3k	10.6k	25.0k	2.6k	462
Npm Downloads (weekly)		1296.6k	1154.4k	257.9k	497.8k	6.6k

V. CONCLUSION

Based on our observation from the resources available, we can conclusively say all the libraries are a perfect fit as per the client's requirement. It is up to these requirements that the client/developer can decide which charting library is to be used. The following are the use-case/requirement upon which a library should be decided.

If the requirement is to use charts without any complex interactions, you should opt for Google Charts as it offers a large number of charts among different open-source libraries. However, it does not come with a high level of customizations and interactivity. If your project is a product-based entity that needs to exist for a longer time, Google Charts is not recommended because Google's open-source project direction is changed without notice or backward compatibility.

HighCharts should be considered when the project has a large team and manage complex data visualization. As compared to other charting libraries, Highcharts is a paid product with a relatively right amount of integrations and support available over email and chatbots, making its integration into projects more comfortable and less time-consuming.

Chartsjs comes with a simple set of charts with minimal configurations. The smaller bundle size of this library is a plus point when it comes to lightweight projects. However, as your project requirements, team size, the complexity of the data viz, and various charts increase, it would be better to consider other libraries.

D3js is a perfect option when the project is long-term and requires a variety of charts with customization. Since the

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 7, Issue 11 , November 2020

learning curve of D3js is very steep, and it supports the most variety of charts, it is a perfect fit for projects where the team is skilled and is working on a long term project.

Being a burdensome framework, ThreeJS takes a little time to load on the front end but is very good for 3D animations. Also, ThreeJS offers a limited variety of charts. Hence ThreeJS graphs are suitable for projects where the number of graphs is significantly less, applications are single page or project is of smaller scale, and User Interface is a priority.

REFERENCES

[1] S. A. Thomas, Data Visualization Using Javascript, No Starch Press, 2015.

[2] P. P. and S. K., "Chart visualization of large data amount," in Computer Science On-line Conference, Springer, 2017, pp. 460-468.

[3] V. a. S. C. A. Karavirta, "JSAV: the JavaScript algorithm visualization library," in Proceedings of the 18th ACM conference on Innovation and technology in computer science education, ACM, 2013, pp. 159-164.

[4] "Data Driven Documents," [Online]. Available: https://d3js.org/. [Accessed 21 September 2020].

[5] N. Downie, "Chart.js," 2013. [Online]. Available: https://www.chartjs.org/. [Accessed 21 September 2020].

[6] J. Dirksen, Learning Three.js: The JavaScript 3D Library for WebGL, UK: Packt Publishing, 2013.

[7] T. Honsi, "Highcharts," HighsoftAS , 2009. [Online]. Available: https://www.highcharts.com/blog/about/. [Accessed 24 September 2020].

[8] K. Król, "Data presentation on the map in Google Charts and jQuery JavaScript technologies," in Geomatics, Land management and Landscape, 2016.

[9] "Statcounter Global Stats," Creative Commons Attribution, [Online]. Available: https://gs.statcounter.com/. [Accessed 25 September 2020].

[10] G. Sterling, "Mobile Devices Now Driving 56 Percent Of Traffic To Top Sites — Report," Marketing Land, 23 February 2016. [Online]. Available: https://marketingland.com/mobile-top-sites-165725. [Accessed 25 September 2020].

[11] P. Lewis, "Rendering Performance," Google, 2 December 2019. [Online]. Available: https://developers.google.com/web/fundamentals/performance/rendering. [Accessed 25 September 2020].

[12] E. Alias, "Technology driven lead generation," Wappalyzer, 2020. [Online]. Available: https://www.wappalyzer.com/technologies/javascript-frameworks/. [Accessed 29 September 2020].

[13] M. Skusa, "FileStage," [Online]. Available: https://filestage.io/blog/project-documentation/. [Accessed 2020 October 18].

[14] "Open Source," [Online]. Available: https://opensource.org/licenses. [Accessed 18 October 2020].

[15] Y. a. H. J. a. L. Y. Xing, "Research and Analysis of the Front-end Frameworks and Libraries in E-Business Development," in Proceedings of the 2019 11th International Conference on Computer and Automation Engineering, 2019, pp. 68-72.

[16] M. a. S. A. E. a. M. S. M. a. A. C. M. Sadiku, "Data visualization," International Journal of Engineering Research And Advanced Technology (IJERAT), vol. 2, no. 12, pp. 11-16, 2016.

[17] J. a. C. V. a. W. L. Choy, "Data visualization techniques: from basics to Big Data with SAS visual analytics," SAS: White Paper, 2013.

[18] H. Da Rocha, Learn Chart. js: Create interactive visualizations for the web with chart. js 2, Packt Publishing Ltd, 2019.

[19] M. a. O. V. a. H. J. Bostock, "D3 data-driven documents," IEEE transactions on visualization and computer graphics, vol. 17, no. 12, pp. 2301-2309, 2011.

[20] O. a. J. D. a. P. P. a. S. S. ElTayeby, "Comparative case study between D3 \&Highcharts on Lustre metadata visualization," in 2013 IEEE Symposium on Large-Scale Data Analysis and Visualization (LDAV), IEEE, 2013, pp. 127-128.