

Web Development is Redundantly Complicated

Haoda Li

April 3, 2020

Summary

The experience with JavaScript, the primary programming language for front end development, is not pleasant. There are too many choices doing the same thing, while they have their own implementations that are not compatible with each other. Further more, the many frameworks vary so much in syntax, while similar in functionality. I've also seen similar voice on Quora, Reddit, and Medium.

This poster is targeted at the web developers. I want to find clues to show that the WebDev community is suffering from these unnecessary redundancy. Therefore, the developers can think about how to improve the community by building better standards and stop writing inefficient code.

1. A Glance of the Data

The data is found from Libraries.io, who tracks 37 package managers and 3 open source repository platforms. Near 5 million projects and repositories are collected, and their dependencies were tracked between 2015 and 2018.

This dataset [1](#) is quite sophisticated, it includes the primary language of the project, the number of dependent projects, as well as the keywords and description. These information is extremely useful for my purpose. As most web projects are developed in primarily JavaScript and its derivatives, I can filter out web projects by the language. Also, I can use the keywords and description to determine the general purpose of the project.

On the other side, not every project will grow to be a good project. Most of the projects are in low-quality and/or abandoned by the project owner. Libraries.io proposes one metric, the SourceRank, to measure the quality of the project. SourceRank is primarily determined by the number of dependent projects, the last update date, and the quality of the documentation. I dropped all projects that is lower or equal to 5 SourceRank score, which are mostly sketchy projects without any documentation and not updated for quite a long time.

2. Think About the Story

Looking at the huge data, I spent a long time determining what should be put on a poster. Considering the audience, who should be developers or the public without deep understanding with statistics. I decided to make the story visually appealing and simple in concepts.

2.1 WebDev is Complicated

The thought is very natural, to show the size of WebDev community and to show the complexity of working on these projects.

2.1.1 Web world is Huge

The first thing to show is that JavaScript dominates in number of projects. Although developers know that JavaScript has the largest community among all languages, most people may not understand how dominant

name	created	updated	description	keywords	sourcerank	dependent_projects_count	language
react	2015-01-29	2018-12-22	React is a JavaScript library for building use...	react	34	62071	JavaScript
react-dom	2015-01-29	2018-12-20	React package for working with the DOM.	react	33	48287	JavaScript
lodash	2015-01-28	2018-12-21	Lodash modular utilities.	modules, stdlib, util	33	78302	JavaScript
webpack	2015-01-29	2018-12-22	Packs CommonsJs/AMD modules for the browser. Al...		33	82495	JavaScript
typescript	2015-01-29	2018-12-22	TypeScript is a language for application scale...	TypeScript, Microsoft, compiler	33	66445	TypeScript
babel-core	2015-02-15	2018-12-19	Babel compiler core.	6to5, babel, classes	32	95760	JavaScript
eslint	2015-01-27	2018-12-19	An AST-based pattern checker for JavaScript.	ast, lint, javascript	32	124191	JavaScript
@types/node	2016-05-25	2018-12-19	TypeScript definitions for Node.js		31	39746	TypeScript
jest	2015-01-29	2018-12-19	Delightful JavaScript Testing.	ava, babel, coverage	31	47512	JavaScript
babel-loader	2015-02-15	2018-10-27	babel module loader for webpack	webpack, loader, babel	31	62716	JavaScript
chalk	2015-01-27	2018-12-08	Terminal string styling done right	color, colour, colors	31	43043	JavaScript
phpunit/phpunit	2015-01-21	2018-12-14	The PHP Unit Testing framework.	testing, phpunit, xunit	31	60607	PHP
activesupport	2015-01-21	2018-12-14	A toolkit of support libraries and Ruby core e...		31	11269	Ruby
mocha	2015-01-29	2018-12-21	simple, flexible, fun test framework	mocha, test, bdd	31	173275	JavaScript

project_name	dependency_name	dependency_project_id
react	envify	144990
react	promised-io	302230
react	fbjs	823625
react	tap	321029
react	eventemitter2	145666
react	tapr	321108
react	sprint	316921
react	scheduler	310860
react	Deferred	277390
react	loose-envify	986741
react	prop-types	649359
react	ensure-array	279520
react	create-react-class	2103382
react	amdefine	261799

Figure 1: A Glance of the datasets

it is. I want my audience to have a very direct view on the size.

2.1.2. Web world is Messy

I proposed several measures for “complexity” of a language. For example, I tried the depth of the dependency tree. However, I found most of the measurements are not easy to understand or it is not significant enough for the comparisons between JavaScript and other languages. I finally choose to use the average number of dependencies. This measurement is simple and JavaScript clearly beats others.

2.2 Redundancy

This is a very hard measurement. The initial idea is to build community graph from dependencies and then measure how similar their distribution over keywords are. However, I quickly faced many issue with this approach. First, the size of the dataset is still too large to construct a the dependency graph. Second, the concept is too complex to explain these in a poster.

2.2.1. Redundancy in Simple Tasks

I worked around by “attacking” JavaScript’s own historical problem. Because JavaScript is not a well-designed language, it lacks a lot of standard libraries for simple tasks, hence the community build packages fill in. However, due to the lack of communication, many community packages work on the same problem and are incompatible with each other. Such problems have troubled me a lot, and I’ve seen many memes on the issues. Therefore, I used the same approach by showing how JavaScript have redundant packages on the simple tasks.

2.2.2. Redundancy in Frameworks

Finally, I looked at the redundancy over frameworks. There are so many question on the Internet about which framework should web developers choose, while they don’t show much difference over these packages. I decided to leave this topic as the final thoughts. These frameworks are leading ways for nowadays web development. However, do the webdev community need these many of frameworks on the same thing.

3. Designing Process

3.1 The Layout

The design is more for the flow of the story, and the actual content I want to fit. The first design (left 2) put more weights on the “redundancy” part, while I realized that the first part is too crowded and the two parts should be equally important. In addition, the four parts grid fits my story better.

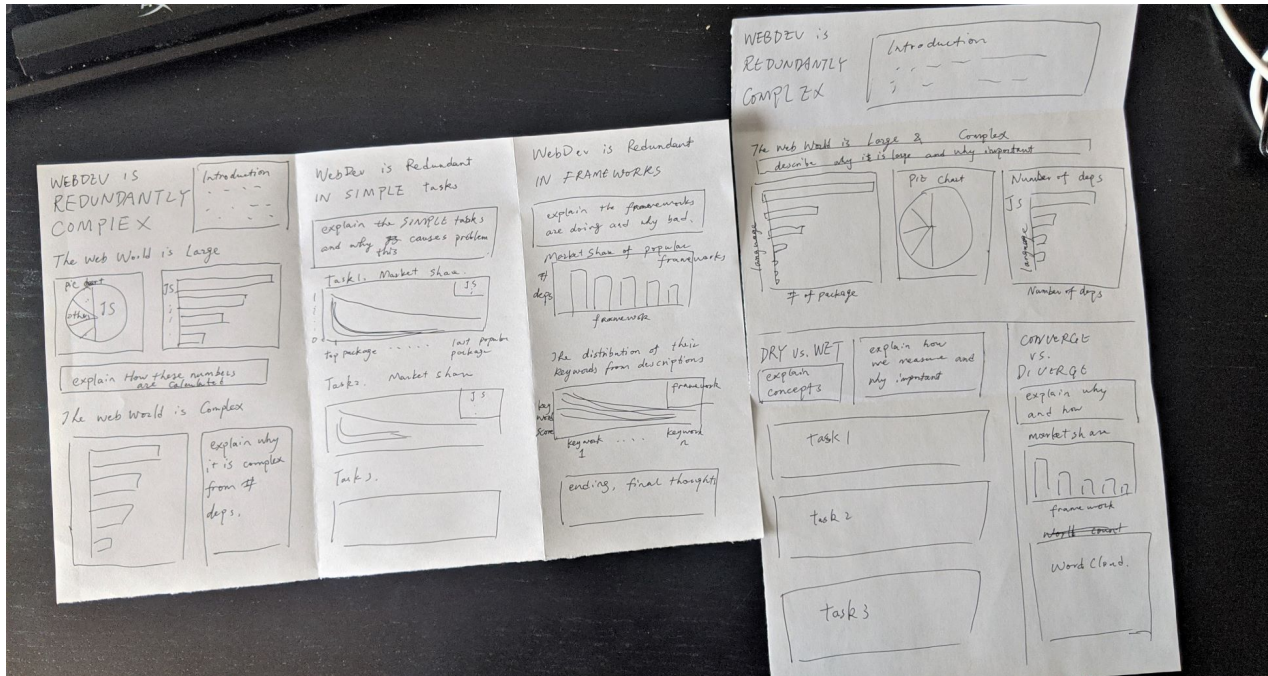


Figure 2: The layout

3.2 The Plots

Many decisions are made on the plots. For showing the WebDev world is huge. I tried to use a bar plot and a pie chart to show the number of projects in JavaScript and the market share it has. However, I find that the two plots have the same information. Therefore, I tried to plot the number of new projects in each language through time as a stacked bar plot, while it has too much information and does not convey the message well. Finally, I choose to have the line plot of the cumulative number of projects for each language, along with the market share as a bar.

For the redundancy part, I tried to balance between the amount of data manipulation and the intuition. I've thought about using the distribution of keywords. However, the information is not quite clear for the audience, and overall looks messy. Finally, I focused more on the intuition side. The plots I used are not “friendly” for statisticians and they don't have quite the exact information (circles, colors). However, they are more visually appealing for the public and allow the audience to consider more on the message.

3.3 Twinkling the Details

Finally, I can knit the figures into a story. For example, I slightly shift the position of the introduction and add a word cloud on the top right. The word cloud is a “glance” of the WebDev world and the developers can be attracted by it. Also, I unify the color and share the legend for some of the figures to make it look nice and precise. Finally, I chose green for the overall color scheme, green is the color of Github's contribution plot. The color hints the open source development community and their contributions to the projects.

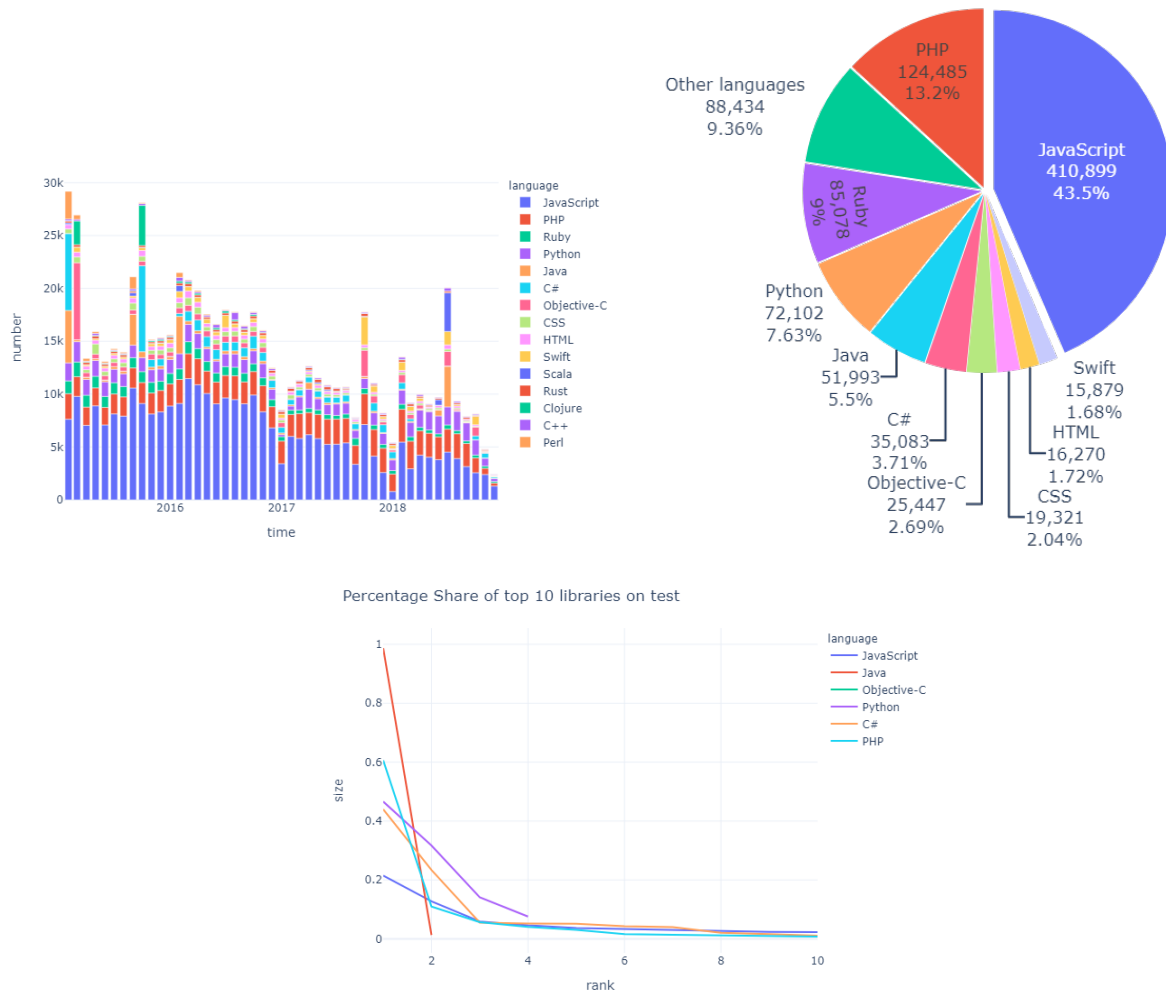


Figure 3: Design of the plots

4. Reflections and Conclusions

First, the most significant difference for a poster is the amount of explanations and information it can hold. Compare to a data report or a interactive dashboard, a poster contains significantly less information. Most of the information are carried through the figures and the lack of interactivity further limit the amount of information. Therefore, we have to focus on the key ideas and highlight it by size, color, or shape. Such task is quite challenging.

Also, the targeted audience matters. Because the difference in the audience and the level of background knowledge. A poster will focus less on the technical details and focus more on the appealing visuals. Sometimes, these visuals may not be very statistically useful, but attractive for the audience.

In addition, Compare to a data analysis or data explorations, the poster is more biased on nature. When I try to convey our audience with the story, I chose clues that are on my side. We choose the data that looks appealing and drop some data that are important but against our story. Although every data analysis will have some degree of bias, the poster is more impacted by these bias due to the targeted audience and limited information it contains.