INFO5100: Project 2 Report

Sichen Li (sl2947) Chen Li(cl953) Xiaoxi Zhang (xz577)

1. Data description

1.1 Dataset

Our mian dataset contains historical facts on GitHub since 2012. This was sourced from Google BigQuery https://cloud.google.com/bigquery/public-data/github. There are four datasets we used in the project: *gh-issue-event.jso*n, *gh-stat-event.json, lang_info.json and creation_year.json. Gh-issue-event.jso*n, *gh-stat-event.json* were using two major measurement matrices for the two major views we built – Github issues and stars by languages. The key variables were: *name(programming language), year, quarter, count(star/issue)*. We also gathered data for knowledge display purpose – *lang_info.jason*. It contains introduction and uses of each language. We hand created this dataset by gathering useful information that we think would be useful to display. Last but not least, also hand-pulled ourselves, *creation_year.json* is helpful to understand the ages of program languages.

To begin, we sketched our plan and the visualizations of correlations, growth and trends that we were hoping to present. We used this plan to manipulate data to show ranking instead of count/percentages. Otherwise it would be dominated by few languages in the dataset and the visualization would be hard to read(like one of the example we found online). We envisioned a compact visualization that could display trends of languages' popularity, but we knew we would need to be concise within languages so the whole product would not be muddied. In this sense, we cleaned the data by taking on the top 10 languages by count ranks. So users only need to focus top languages each quarter throughout the history.

1.2 Data Extraction

We use the nest and map function in d3 library to extract raw data into the specific format we want, so it will be easier for us to display the relationship between programming language ranking and year. In raw data, we have language's count and its corresponding year. Firstly, we use nest function in d3 to nest data by language's name, so in Json format, each key is the name of each programming language, and the value of the key is a array that represent the language's information for this language. Then, we use map function associating sort_function in d3 to sort the rank of language in each quarter for each year by the language's counts. However, because of the large amounts of the programming language name in our raw data and the aim of this project is that we only need to display the top 10 ranking of programming language, so we use slice function to get the first 30 ranking of entire programming languages, in this way,

we can also observe the tendency of the language's ranking although the ranking extend top 10.

2. Visualization description

2.1 Overall structure

There are three divisions in our visualization with two divisions that are interactive.

- Top division: introduction of the purpose of our visualization, and a brief guidance on how to interact with it;
- Middle division: MAIN interactive visualization for popularity of each programming languages where users can changed views based on interests;
- Bottom division: introduction of each language and examples of popular projects;

2.2 Top 10 rankings

The purpose of showing top 10 programmings ranking by time comes into 3 variables:

- Current ranking;
- Highest ranking per language had reached ever;
- Year they were created;

Current ranking: The original data is only organized by year and count. We first group the source data by year using **d3.nest().map(sort_function)** to get the rank information. Then we sorted the rank by year with increasing order by **data.sort(function(a,b) {return a.year - b.year}).**

Highest ranking: We represent highest ranking with the history maximum rank group by language name. Then get the highest rank with d3.nest().key(function(d) {return d.name;}).rollup(function(d) {return d3.min(d, function(g) return g.position)}. The thickness of lines indicate the highest ranking per language has ever reached.

Year of creation - Color scale: We use line color to represent the age of programming language. The dark color represents old language, and light color represents young language. The creation year data source comes from Wikipedia. The year gap in our data is about 40 years. Then we generated our color scale by 40 steps this website http://gka.github.io/palettes.

Interactivity:

1. Brush

To make the time choosing more flexible, we implemented a brush changing the focus of data. When user move the brush area, the focused area will redraw the data within the specific year. The API we used is d3.svg.brush(). You can find more information in https://github.com/d3/d3-3.x-api-reference/blob/master/SVG-Controls.md

2. Click

Two animation we implemented with mouse interaction. Mousemove will highlight the programming language user chose. To show specific information and popular github project related to that language, user need to click on one of lines in the focus area. We show small hand instead of cursor when hover on specific line of language, to guide to click for more information. The display is achieved by DOM element selector.

3. Button

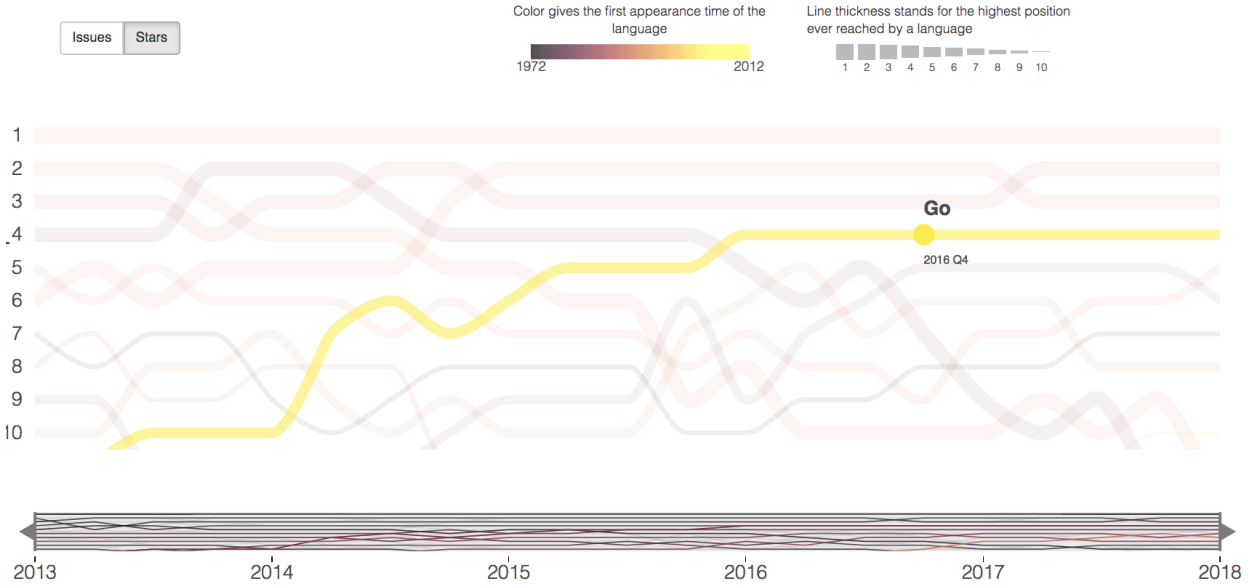
To refer language from both starts and issues, we need to display from two data source. The data change is achieved by button click. We use JQuery library to finish button selection function.

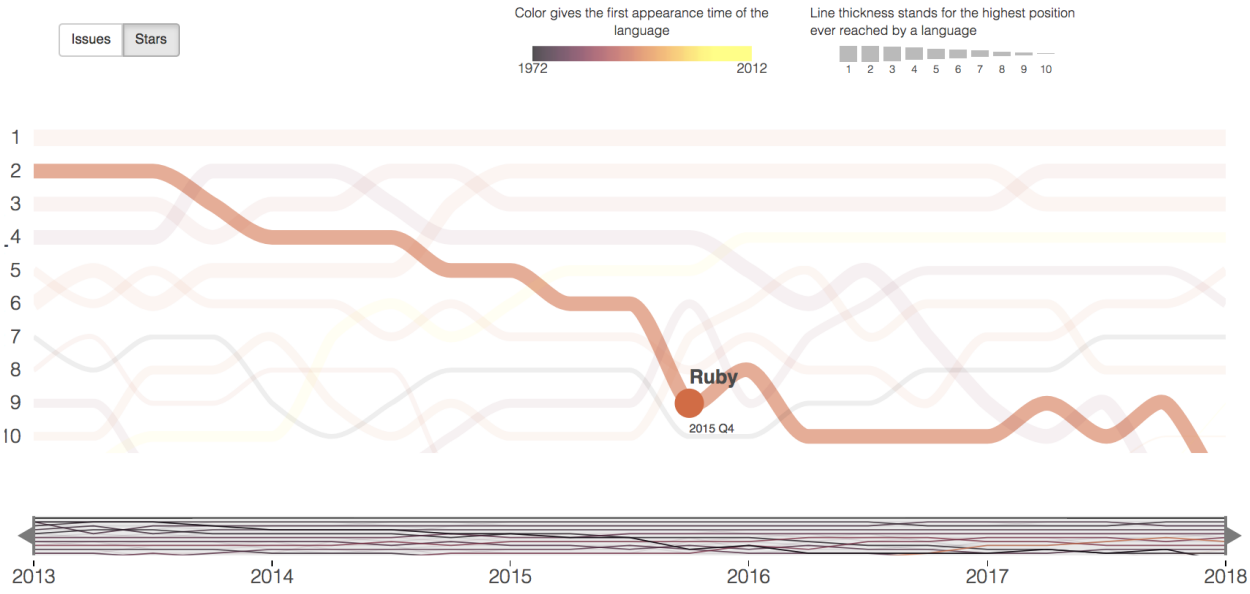
3. The story

Learning how to code has been “the trend” for younger generation during the past decade. Schools started to introduce coding technics earlier and earlier each year. What’s more, working professionals in varieties of industries had expressed interests and increased needs in learning basic coding logics as well. At the same time, Github is currently the largest online code repository. We believe by tracking “Issues” and “Stars” per language, we can not only get a good overview of how popular they were, but to understand each language’s characteristics.

1)It’s interesting to see few languages that were getting hotter, while some were fading in communities’ view. Since Google introduced Go, it was consistently growing each year and reached to a decent position for such young programming language While

Ruby were losing attraction to people year after year.

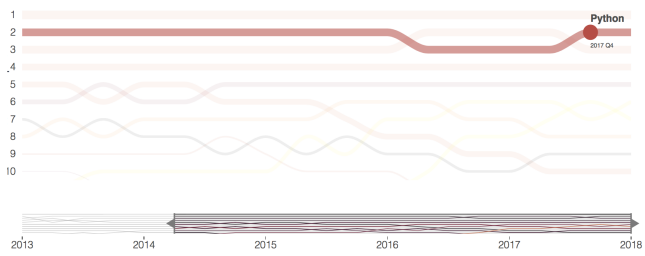


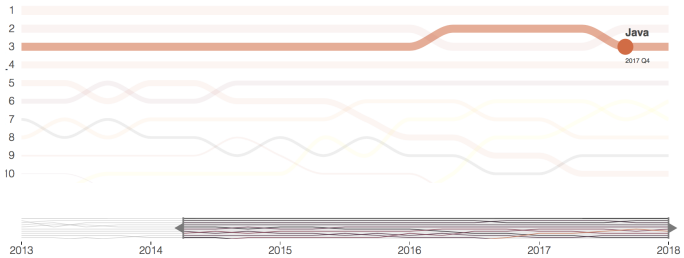


2)

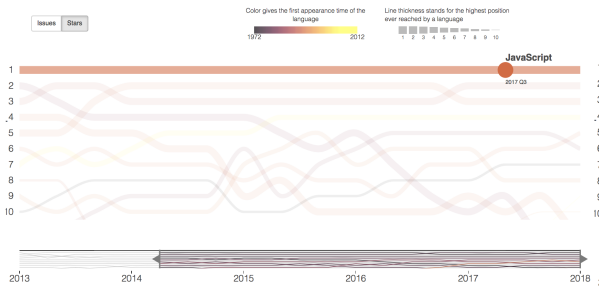
Issues should be relatively approximate stars, since the more popular a language is, the more issues people would raise.

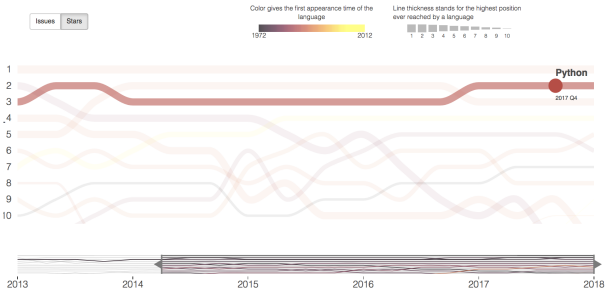
By looking at the top three languages across history, we noticed Javascript, Python and Java are the top three(see example screenshots below) vastly due to their scalability and Versatility.





This is consistent with “stars” visualization.

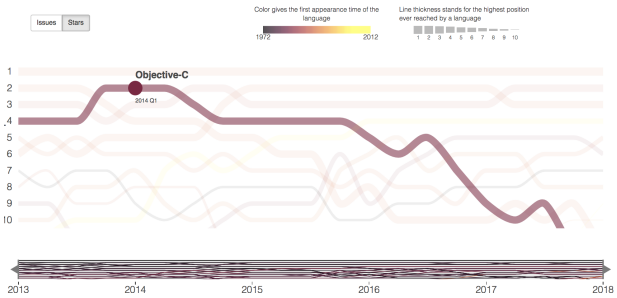


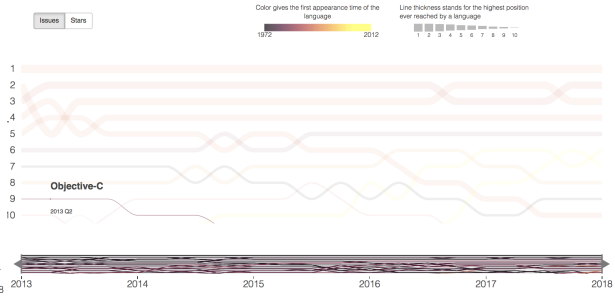


3)

Interestingly, some languages did not follow the expected behavior. Objective C and CSS for example, have way higher rankings for stars than issues. This indicates that they are relatively user-friendly languages. Also, please see a counter example of PHP, which stars ranked about 6th and 7th, but issues ranked about 4th.

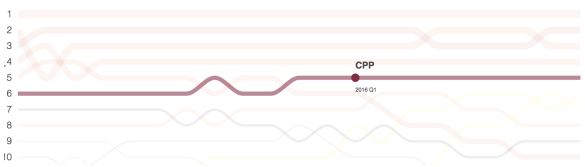
Stars vs Issue shown below for **Objective C**.

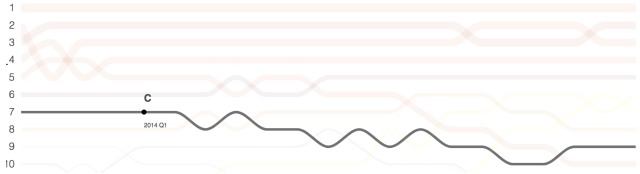




4)

We can also notice that the relatively “low-level” language(C/C++) were tend to be more stable, but not most popular. Possibility due to their deep learning curve, and require users to have solid programming background(Hence, they are not the best programming language to start with for newbies!)





Overall, with language information and popular projects down below, we hope this visualization to help people, especially who are new to programming, to have a quick overview of what should they learn first(or not learn).

Reference

Color scale http://gka.github.io/palettes.
Github ranking by time: https://madnight.github.io/githut/#/issues/2017/1
Baby name trending: https://www.visualcinnamon.com/portfolio/babynames
Introduction of programming languages: languages:http://www.businessinsider.com/the-9-most-popular-programming-languages-according-to-the-facebook-for-programmers-2017-10#15-objective-c-1
https://mikkegoes.com/14-programming-languages-explained/
More detailed github ranking by time: http://githut.info/
Github ranking: https://fontawesome.com/
Ranking insides: https://www.benfrederickson.com/ranking-programming-languages-by-github-users/