Interactive Data Visualisation:

GitHub stars

Introduction

In this essay, the data visualisation technique that I will be describing is GitHub stars, a metric that measures the quality and popularity of repositories on GitHub. I will first give an overview of this data visualisation technique as it is implemented on GitHub. I will then analyse the appropriateness of this technique by selecting three different repositories and comparing the conclusions that can be drawn from the number of stars attributed to each repository. I will investigate whether the stars feature allows users to make a verifiable assessment of the overall quality of the repository, or whether they are in fact much less meaningful and more similar to a simple bookmarking method. Finally, I will also outline a number of suggested improvements for the GitHub stars feature.

Description of visualisation technique

When a user logs in to GitHub, they can search for repositories such as React, Vue, or Angular. When the user clicks on a particular repository, various information is displayed, including the code files, a list of tracked issues, and pull requests, among other data. Usage statistics for the repository are also displayed in the top right-hand corner (see Figure 1 below).

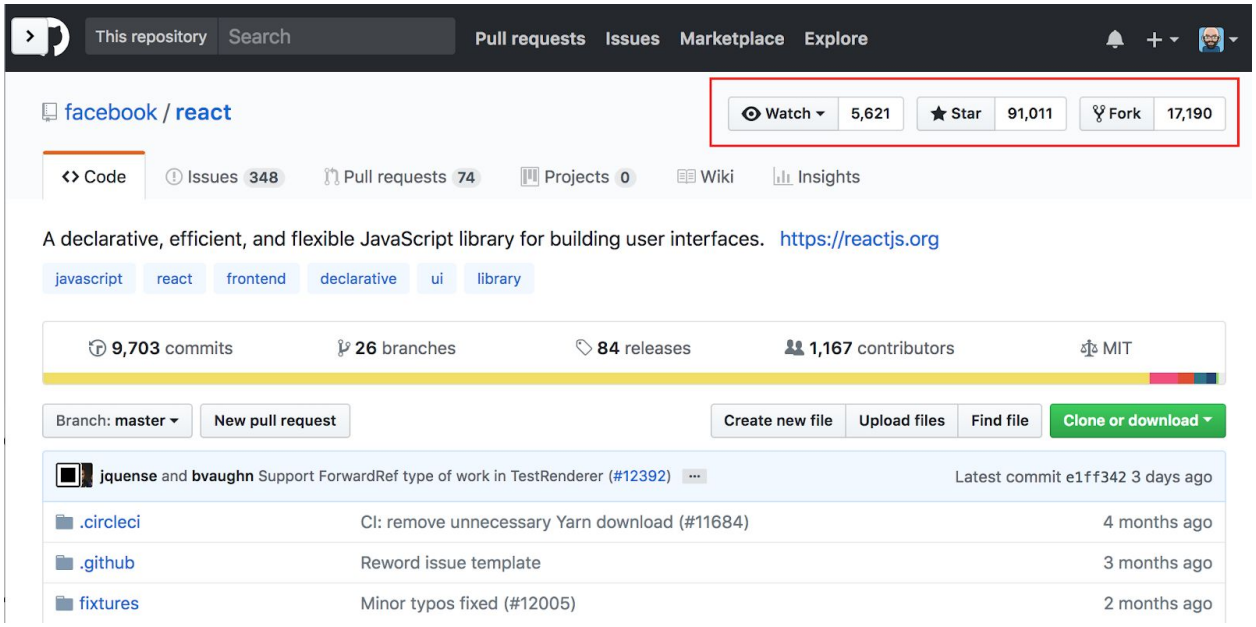

Figure 1. Usage data for the React repository on GitHub.

These usage statistics include the number of users “watching” the repository (i.e. they will get a notification if an update is made), the number of users who have “starred” the repository, and the number of users who have “forked” the repository (i.e. they have copied the repository in order to make and further develop their own version). In Figure 1 above, we can see that the React repository on GitHub had been starred 91,011 times as at the time of writing, which would seem to indicate that it is both of high quality and popular among users.

According to the GitHub help article on the stars feature [1], the purpose of stars is as follows:

- Stars make a repository easier to find again. When a user stars a repository, they can then go to a list of their starred repositories in their profile for easy access. (See figures 2 and 3 below).

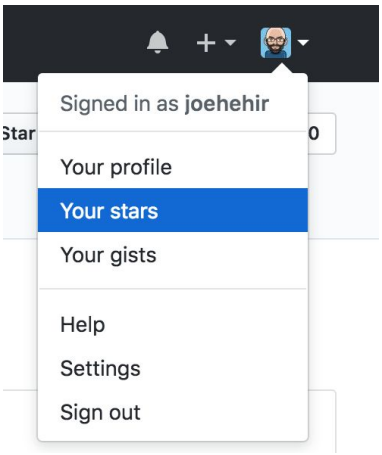

Figure 2. Accessing a user's starred projects on GitHub.

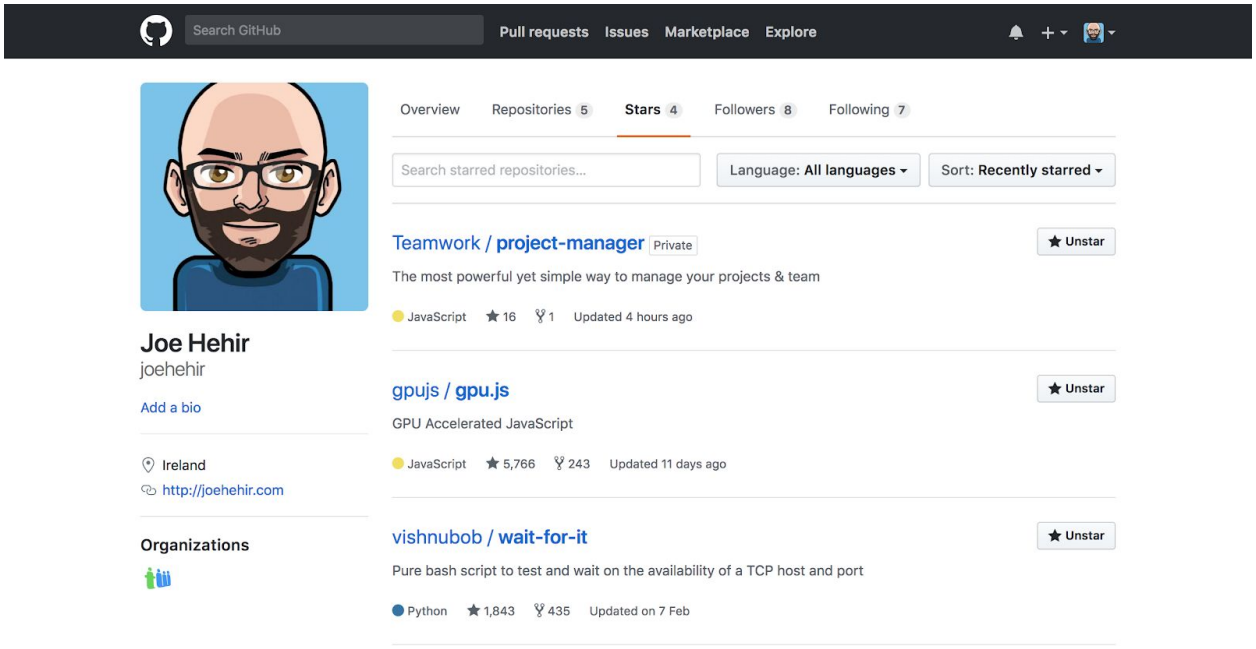

Figure 3. List of starred projects in a user's profile on GitHub.

- Starring repositories also allows users to discover similar projects on GitHub. When a user stars a repository, GitHub will recommend similar projects in the Discover repositories view of the user's news feed (see figure 4 below).

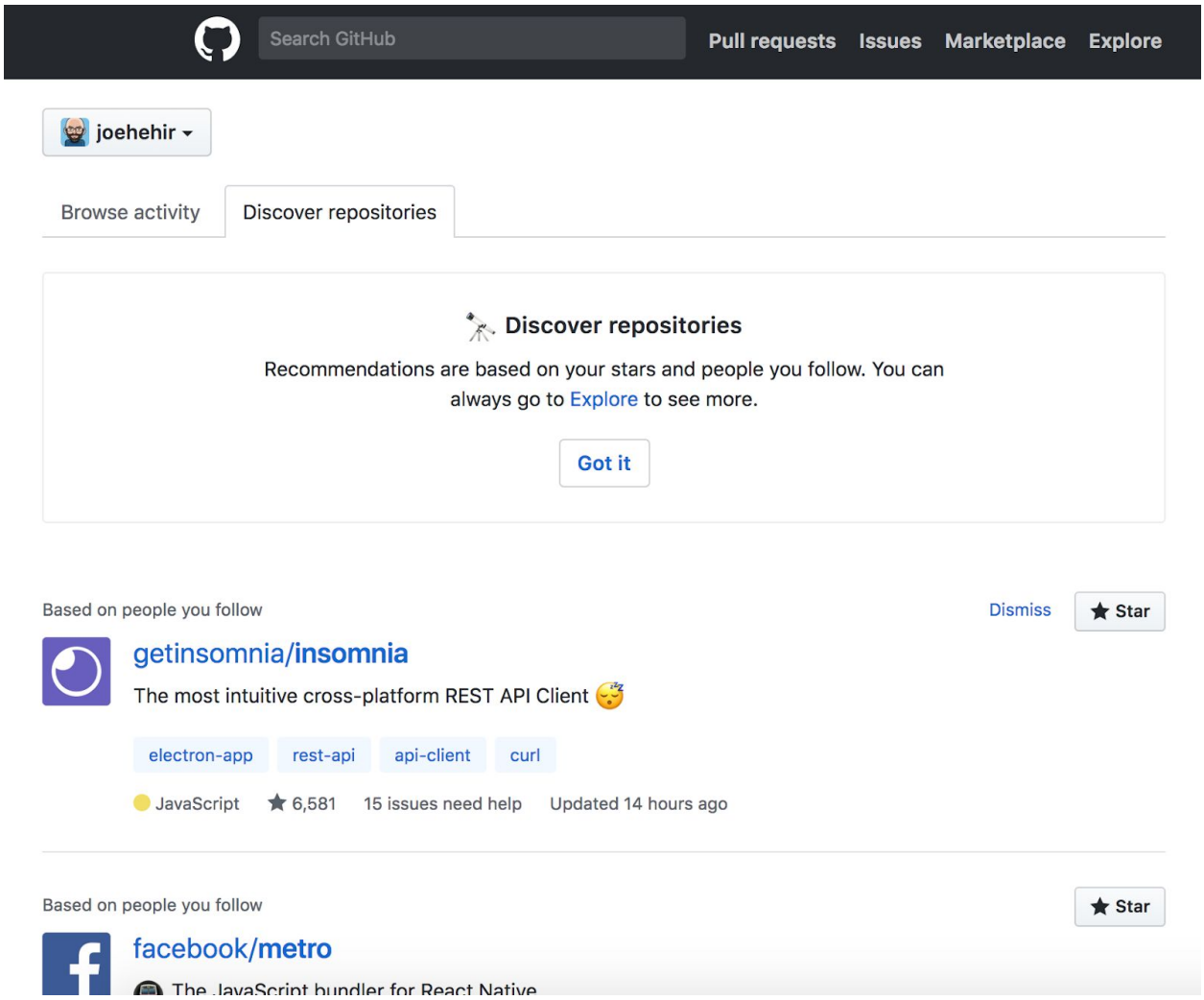

Figure 4. Repository recommendations based on previously starred repositories on GitHub.

- Finally, GitHub states that users can star a repository to “show appreciation to the repository maintainer for their work” [1], meaning that the user is essentially endorsing the repository as being a good resource. A repository’s number of stars is also used to determine its ranking on GitHub, and repositories can be sorted based on their star count.

From this description, we can see that GitHub stars are intended as a bookmarking and feedback tool combined, i.e. they make it easier for users to find starred repositories, and also enable users to endorse repositories that they find useful.

GitHub stars are often cited as a measure of the success of a repository. For example, at the 2018 Vue.js conference in Amsterdam, when creator Evan You presented the progress made by Vue.js during 2017, the first point he made was that Vue.js was the most starred project on GitHub that year, with over 40,000 stars [2].

Additionally, in an article on Medium, Diana Neculai, co-founder at Froala, states that GitHub stars are “a reliable insight that engender trust and influence people when they decide to use your product or not” [3].

Therefore, the number of stars a project or repository has on GitHub is seen as representative of its progress and value. In the next section, I will investigate whether this perceived value is a justifiable indicator of a resource’s value.

Analysis of appropriateness of visualisation technique

As discussed in the previous section, the stated purpose of the GitHub stars feature is to enable users to bookmark relevant repositories and provide positive feedback and appreciation to the repository maintainer.

As an objective user, I would expect the number of stars to reflect the following factors:

- The repository is used by a large number of users
- The repository is actively maintained and continually developed
- The repository has been tested extensively and is therefore a valuable resource

In this section, I will investigate whether the GitHub stars feature is representative of the three points listed above, in combination with the following quantifiable metrics available on GitHub:

- Commit frequency
- Issues open
- Time to issue close

In order to investigate the correlation between star count and overall value, I compared three different JavaScript virtual-dom implementation repositories. The three repositories selected were as follows:

1. React
2. Vue.js
3. Knockout.js

My findings were as follows (for a visualisation, see Figure 5 below):

1. React
    - Number of stars: 91,011
    - Commit frequency: 25.40/week
    - Number of issues open: 348
    - Time to issue close: 1 day, 48 minutes
    - Number of contributors: 1,167

2. Vue.js
    - Number of stars: 87,369
    - Commit frequency: 14.92/week
    - Number of issues open: 84
    - Average issue lifespan: 14 hours, 27 minutes
    - Number of contributors: 182

3. Knockout.js
    - Number of stars: 8,806
    - Commit frequency: 1.69/week
    - Number of issues open: 271
    - Time to issue close: 18 days, 4 hours, 30 minutes
    - Number of contributors: 74

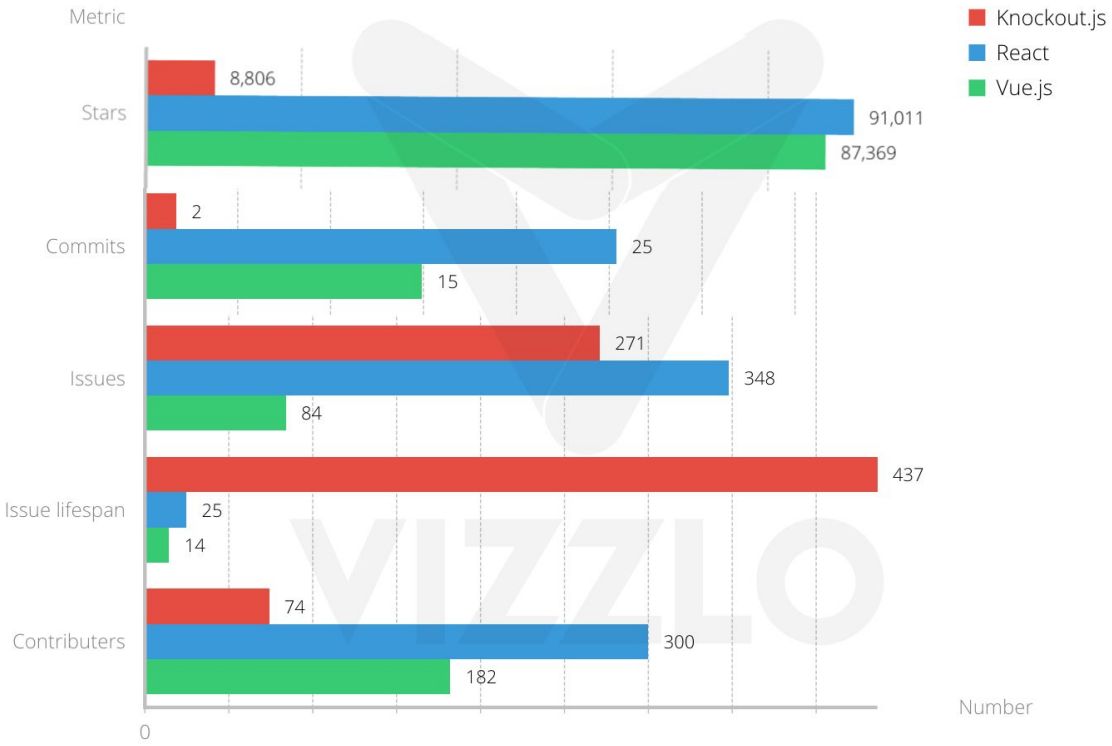

Figure 5. Visualisations of key usage data for the three GitHub repositories investigated.

In order to gather this data, I used the following methods: I used the GitHub API to get the weekly commit count for each contributor, and divided that figure by the number of weeks since the first commit. The number of issues is provided on the main GitHub page for each repository. For the time to issue close, I again used the GitHub API to retrieve the date that each issue was created and closed on, and subtracted these figures to get the lifespan of each issue. I then added the total number of issues and calculated the average issue lifespan. The number of contributors and of stars is also provided on the main GitHub page.

As we can see from the figures above, both React and Vue.js have a very high star count. Both of these repositories also have a high commit frequency, at 25.40 and 14.92 per week respectively. This means that both of these repositories are actively being worked on and maintained on a regular basis. Both React and Vue.js also have a fast turnaround in terms of

resolving issues logged, which indicates that they are still in active development and therefore are likely to be more stable and can be used with confidence.

It is important to note that there is a significant difference between these two repositories in terms of the number of contributors: React has 1,167, while Vue.js has just 182. React is developed by Facebook, so the majority of contributors are likely to be paid employees of the company. On the other hand, Vue.js is a sponsored project that will have some sponsored contributors, but ultimately the majority of contributions will be voluntary. However, in spite of the difference in the number of contributors, the average time required to resolve issues is comparatively similar and very short for both repositories, indicating that they are both in active development. This finding correlates with the very high star counts for these repositories.

The third repository investigated above was Knockout.js, a project maintained on a voluntary basis. At 8,806, its star count is a fraction of React and Vue.js. Correspondingly, its commit frequency figure is also much lower, at 88.7% lower than Vue.js (1.69 commits per week) even though the number of contributors is quite high, at 40.7% of the contributor count of Vue.js. In addition, the time to resolve issues is much higher, at 18 days, 4 hours and 30 minutes. These figures indicate that Knockout.js is not being maintained on a sufficiently active basis to be used with full confidence in production. This finding correlates with its low star count.

My objective in conducting this investigation was to determine whether a link can be drawn between the number of stars for a repository and its overall quality as a resource. Based on the findings above, the repositories with the highest star counts seem to be in active development and are maintained on a regular basis. Issues logged by users are resolved in a short timeframe, which inspires confidence among users. On the other hand, the repository with the lowest star count is most likely not being actively developed, but merely maintained, and issues take a much longer time to be resolved. These findings support the assumption that a high star count indicates that a repository is a high-quality resource that can be used with confidence. As such, the GitHub stars feature is an appropriate measure of quality and popularity among users.

Suggested improvements for visualisation technique

While GitHub stars are a useful feature that are widely used by the GitHub community, I would propose one area that could be improved upon. If the number of stars is perceived to reflect the

quality and popularity of a repository, then stars should be monitored and refreshed in order to ensure they are still relevant. For example, if a user starred a repository five years ago, but now that repository has become obsolete and is no longer maintained or used, then the star should ideally be removed so that obsolete repositories should not maintain a high index position in the Discover repositories view on GitHub. This could take the form of GitHub asking users to review their starred repositories on a yearly basis, for example. This approach would ensure that stars remain relevant and up to date.

Conclusion

In this essay, I have presented the GitHub stars feature, which is officially intended to 1) allow users to find their "favourite" repositories easily, 2) receive recommendations for similar repositories, and 3) provide positive feedback to the repository maintainer. Based on my investigation of this feature, it fulfils all three intended functions effectively. However, having correlated the star count with other usage metrics for various repositories, I found that the star function inadvertently fulfils a fourth, "unofficial" function: it is reflective of the state of development and maintenance of the repository, and is therefore indicative of a repository's value to users. This supports Evan You's citing of Vue.js's star count as a measure of its success. This inadvertent function is also positive in terms of interface simplicity: rather than introducing a separate rating or review feature for repositories, the star feature incorporates this information in one neat bundle.

References

1. GitHub - "Exploring projects on GitHub / About stars". https://help.github.com/articles/about-stars/, accessed 16 March 2018.
2. Evan You - "State of VueJS 2018 | vuejs.amsterdam conference". https://www.youtube.com/watch?v=7YXt3Rmrib4, accessed 15 March 2018.
3. Diana Neculai - "How to get up to 3500+ GitHub stars in one week — and why that's important," Medium. https://medium.freecodecamp.org/how-to-get-up-to-3500-github-stars-in-one-week-339102b62a8f, accessed 16 March 2018.