

Ryan Parker
Professor Businge
CS 472
26 January 2024

Executive Summary

Repository: <https://github.com/munch2024/munch>

Forked Repository: <https://github.com/rparker2003/cs472project-fork>

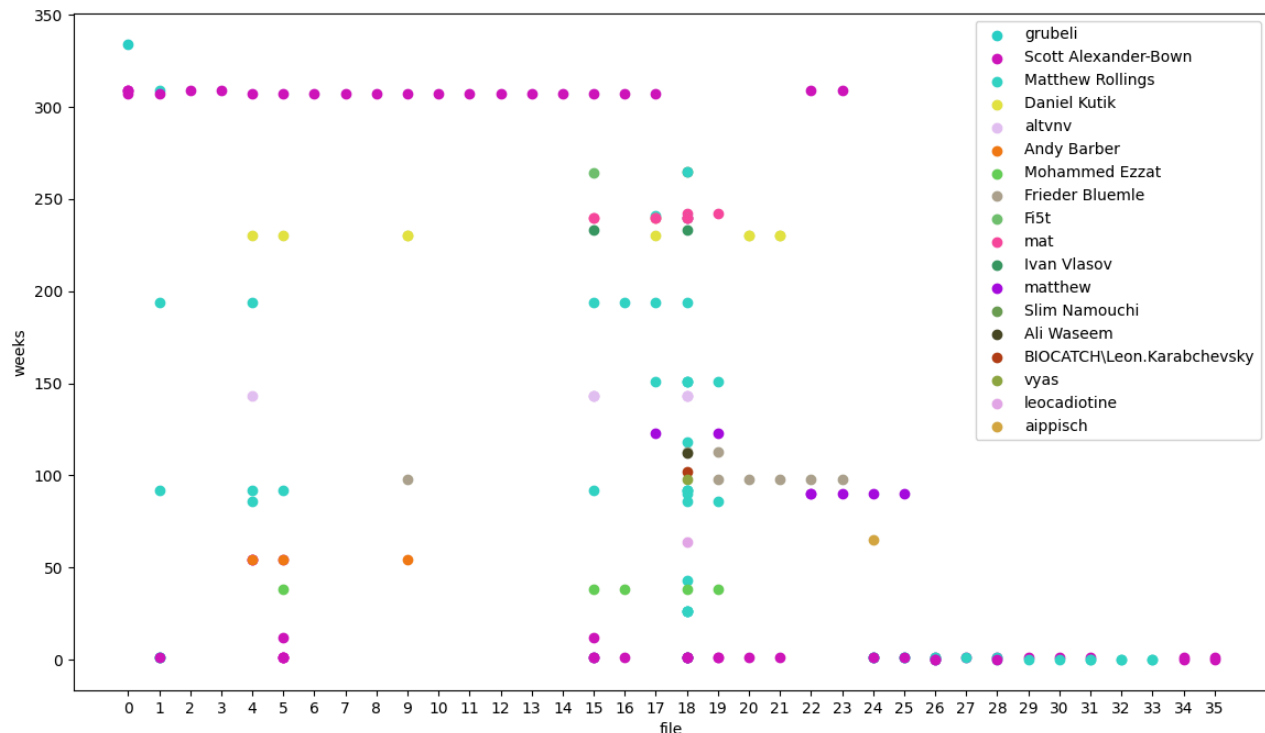


Figure 1. Source File Commit History for GitHub Repository scottyab/rootbeer

In the interest of determining which engineers are contributing the most to the company project, I have composed 3 scripts to gather the contribution metrics of a given GitHub repository.

- **Ryan_Parker_CollectFiles.py** - Gathers the source files from the repository.
- **Ryan_Parker_authorsFileTouches.py** - Gathers the authors and dates of the source files that were touched.
- **Ryan_Parker_scatterplot.py** - Outputs the data of touches to a scatter plot for a visual representation of authors and their total contributions over the timeline.

The scatter plot in Figure 1 easily shows you the distribution of work over time, but it doesn't show you the total amount of work done. Thus, I have created another script that takes the data from the scripts above and outputs a bar graph to show the total number of contributions per author.

- **Ryan_Parker_barplot.py** - Outputs the data of contributions to a bar plot for a visual representation of total contributions per author.

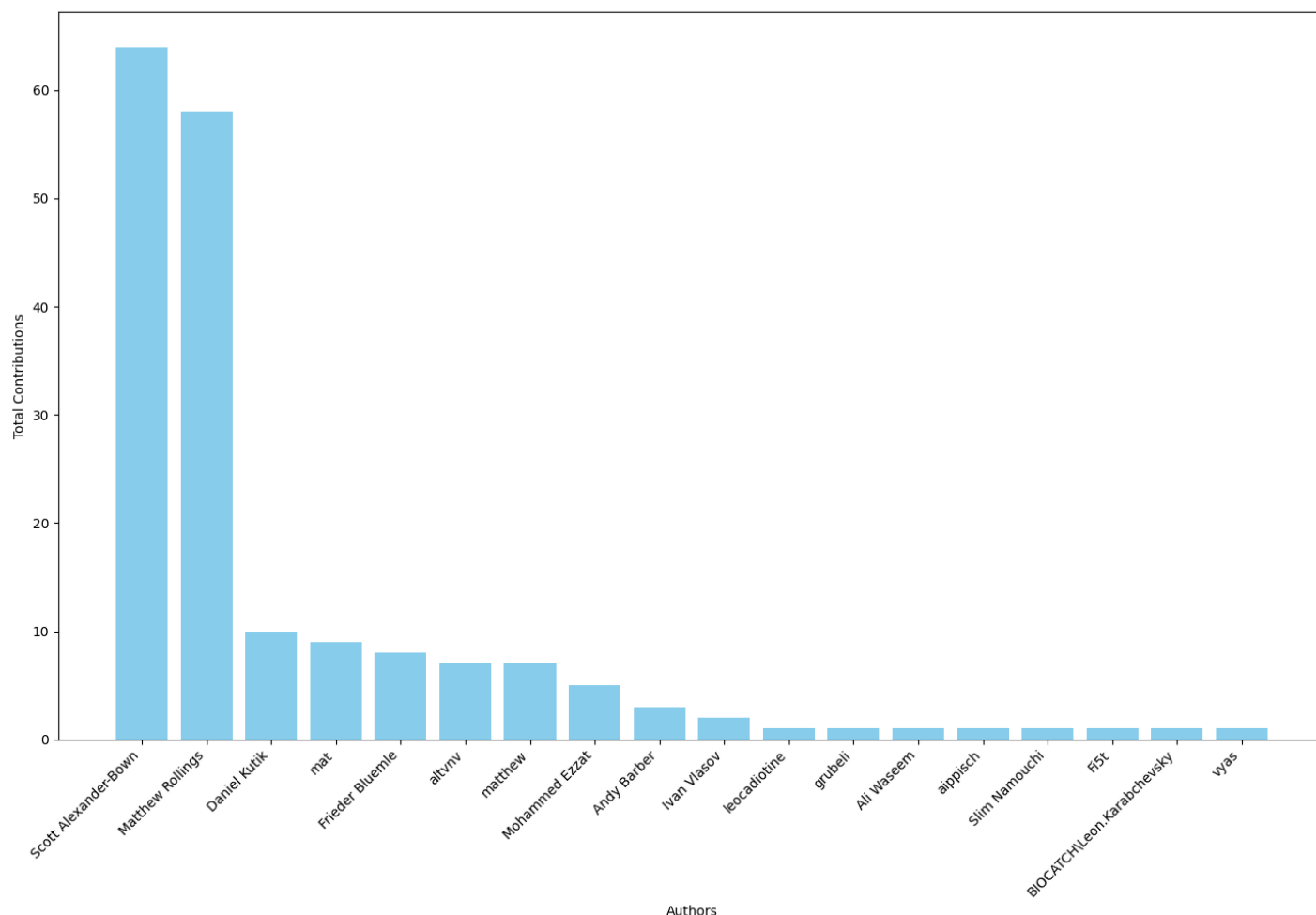


Figure 2. Author Total Contributions Graph for GitHub Repository scottyab/rootbeer

As you can see, Scott Alexander-Bown has the most contributions with Matthew Rollings right below him. These 2 authors have contributed most of the work to the project, and authors like vyas, Fi5t, and appisch have only 1 contribution to the entire project.

While creating these scripts and gathering data, I found many very useful git commands. My most used git command would have to be git status, as it shows the branch you're in, the files you have either staged or unstaged, and whether the working tree is up to date. Another essential git command is git commit. Without this command, you are not able to store any files or changes.