Analysing the first ten contributions of developers to Open-Source Software

Bharath Sathuri*
cs20m011@iittp.ac.in
Indian Institute of Technology Tirupati
Tirupati, Andhra Pradesh, India

ABSTRACT

This paper talks about the study of developers contributing to open-source projects/software repositories of an organization on Github where we study the first ten contributions of the developers to these software projects. We have scrapped the data from Github repositories and made a final dataset from which we get the insights to answer the research questions.

CCS CONCEPTS

• **Software and its engineering** → **Open Source Software**.

KEYWORDS

Open-Source, First-Contributions

ACM Reference Format:
Bharath Sathuri. 2021. Analysing the first ten contributions of developers to Open-Source Software. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The importance of Open Source Software has increased a lot in the last decade. Also, the contributions to these open-source software have increased on par with the number of projects. The Developer's contributions to these open source projects are varied across all the projects including all types across languages. This developers list includes all people with all levels of expertise ranging from freshers to experts. This study is focused on the first ten contributions of developers to these open source projects. How their contributions varied from others and how they have improved over these first ten contributions. We also study the languages used in these open-source projects and their percentage. For this study, we are selecting one of the most popular open-source organizations: Mozilla. It has around 2200 repositories over various languages. For this organization, we are studying the first ten contributions of developers to the repositories in this organization.

*Have doen the project and written the report.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 DESCRIPTION

All the phases of this study have been described in detail.

2.1 Problem

The problem here is to analyze the first contributions of developers to open source projects or software of various repositories belonging to an organization on Github. They are spread over various technologies, frameworks, languages, and developers from all over the world with all levels of experience.

2.2 How is it solved in the existing literature? What is empirical research on that?

In the existing one(the base paper which I have selected for this study), was done on the first contributions of developers to open source projects(software). In that, they concluded that most of the first contributions were of trivial type(non-coding changes or coding changes which involving 2-3 lines of change or typos, etc). So they were unable to have a proper conclusion on the first contributions of open source developers. So I wanted to extend this study by choosing the first 10 contributions of developers instead of one where we get a better chance to analyze their contribution and their improvement.

2.3 Dataset and its selection

Data to be scraped:

Data of first 10 contributions of developers who are actively contributing to open-source projects. Data Source: Github API[4] Methods used for Data Scraping: Using the Github API and python requests module[9] to extract data from Github. Steps to be followed(with reference to as mentioned in the base paper[11]):

1. Data Scraping:

1.1. select the organization/group of the organization(to filter repositories)

1.2. select top 1000 repositories

1.3. select top 1000 contributors from each above-mentioned repositories and select unique contributors from the combined list

1.4. Check for their pull requests to the above repositories and also check if the above repositories have this commit. If yes then add this repo to the open-source contributions list.

1.5. Sort them according to the timeline and select the first 10 repositories and their commits, pull requests for each above contributor.

1.6. This data becomes the first 10 open source contributions of the above users. (At-least to the above organizations)(not sure!?)

2. Formatting the data and making a proper dataset.

2.4 Research Questions

1. How developers are contributing to Github during their initial intro to the new repository and how they have improved over 10 open source projects contributions?

This research question is mainly focused on how the developers contributions have increased from first contribution to next ten contributions. how their contributions ranged across various repositories, langauges etc.

2. How do the same developers have their first contributions in five repositories?

This is focused on how a repository having five contributions from five developers have ranged in various aspects.

2.5 Methods or Tools used

All the steps in detail will be described here.

The first task is to select an organization of which we wanted to scrap the data from Github repositories involving all the details related to repositories, developers, contributions, commits pulls, issues, and other descriptions.

For this purpose, I have selected the Mozilla Organization[1] to scrap its repositories(It has more than 2200 repositories) data. First, in this, we should get the list of repositories each one with its count of contributions it has received from all the developers.

Now, with this list, we should get all the details of these repositories provided by Github API[4] in the form of the JSON format. We use the request package in python to send a request using Github API with a user authentication token, and the JSON[5] package to hold the JSON data and pandas package to convert JSON into CSV[3].

Then with the list, we get all the other links to extract the data from. They are contributors list, commits, pulls, issues, repo description, and language, and all other details related to stats of commits, message, author, etc.

As discussed above in the data selection, we are getting the data of pull requests of he all the developers to the above repositories, with the details of pull requests id's, URL,

And all the commits data to the above repositories, which includes commit id, commit message, commit author, sha author, stats of total changes, total additions, total deletions.

All these are made into CSV files for each repository. Into folders of pulls and commits. A program will combine all these files into a single file called a commits dataset and pulls dataset.

Then we would have another program that runs by taking these two files and checks for whether the ids, repository name, and user name of developers and make the files with all required columns for the first ten contributions dataset.

Interactive plots are made with plot bokeh package[2]

The final dataset has 3400 contributions.

2.6 Results

All the results of this study will be explained in detail including plots and tables and related graphs.

Among all the repositories in various languages, the most popular language with max number of contributions is Javascript with a percentage of 33.8% and the second most popular language is Python with a percentage of 27.13 % and the least popular languages among the total number of contributions is ApacheConf, Swift, Lua, Brightscript0.029%

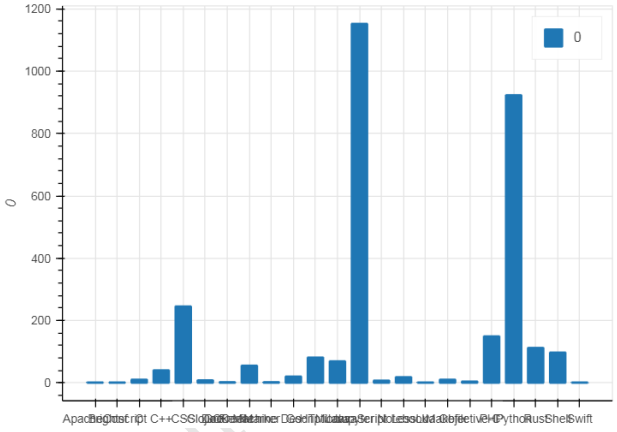

Figure 1: Bar Chart on Languages

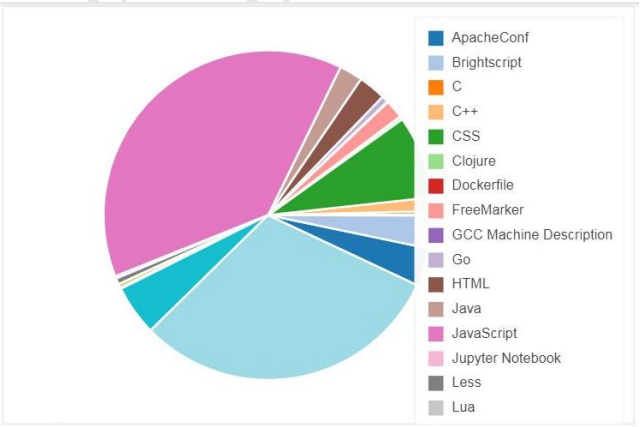

Figure 2: Pie Chart on Languages

3 RESEARCH METHODS

3.1 Part One

1. How developers are contributing to Github during their initial intro to the new repository and how they have improved over 10 open source projects contributions?

Here, to answer this research question, We should follow the steps mentioned in the Dataset section to get the final dataset. From that final contributions dataset which we get after checking for commits and pulls and when a pull request is committed to the repository, then it is considered as a contribution to the repository by the developer. From this, we need to get subset data of the first 10 contributions of each developer among all these repositories of the Mozilla organization. This dataset is first sorted according to users, then for the date of commit, and then concerning repositories. This way, the whole dataset is sorted which is ready to use to answer this question.

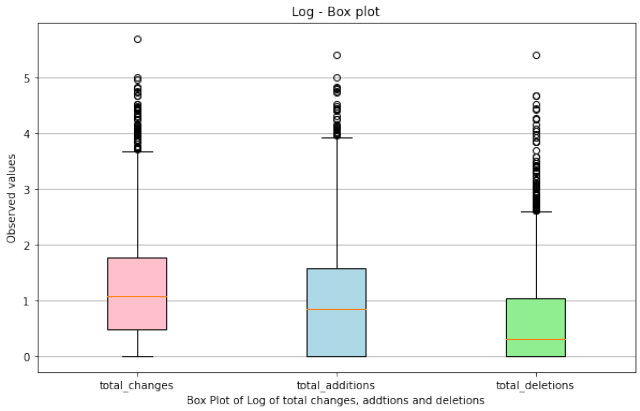

Figure 3: Box Plot on Log of total changes, additions and deletions

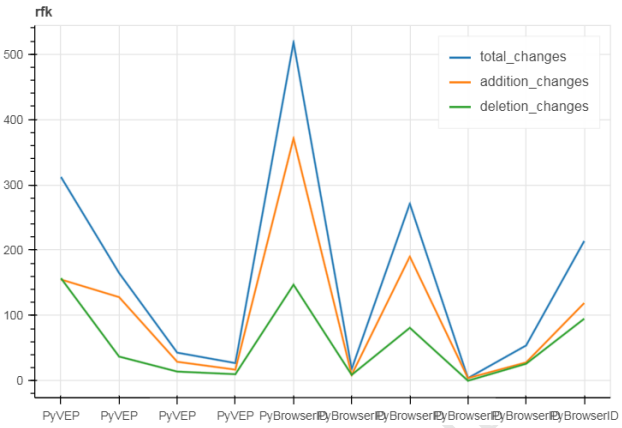

Figure 4: User Plot of all contributions

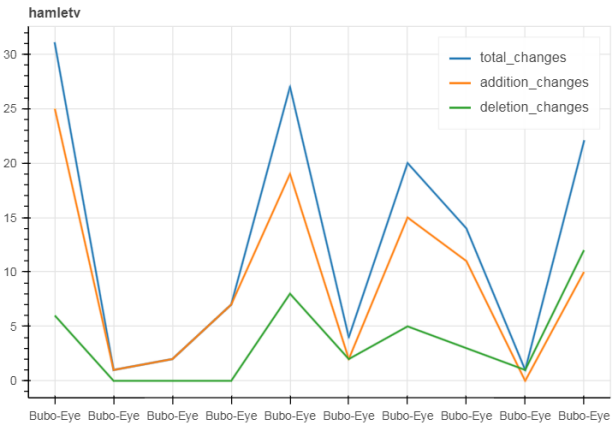

Figure 5: User Plot of all contributions

This dataset is imported into a pandas data frame and then it is grouped by the username which gives us a group for each user.

Table 1: Most used Languages

Language	No. of Contributions
Javascript	1152
Python	923
CSS	245
PHP	149
Rust	112
Shell	97
HTML	81
Java	69
FreeMarker	55
C++	40
Go	20
Less	18
C	10
Makefile	10
Clojure	8
Jupyter Notebook	7
Objective-C	4
GCC Machine Description	2
Dockerfile	2
Brightscript	1
Swift	1
Lua	1
ApacheConf	1

This group has all the columns from the dataset. Out of these, we would use the total changes count, total additions and total deletions in each contribution to answering this. These 3 columns from all contributions give us an overall understanding of user contributions. This data also has a column of repo language which gives the language that the user has contributed to.

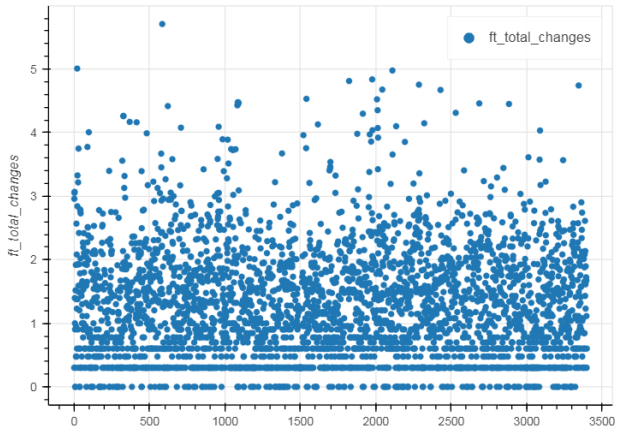

Figure 6: log of Total changes of First ten Contributions of All users

This plot shows that most of contributions are in range of 0.5 to 3 in above log of total changes plot.

3.2 Part Two

2. How do the same developers have their first contributions in five repositories?

In this, I have taken both the one repository with more than five users as contributors and each user contributed to more than 5 repositories. This gives the overall understanding of different users have contributed to the different repositories and different repositories have different contributions from different contributors. To answer this, on the final dataset, we need to group it by each repository for 1st one and by each user for 2nd one. On this, we need to check the uniqueness of each other attribute of username for 1st one and repositories for 2nd one and we should consider only that group to answer this question. From this grouped data, we need to get the columns related to each total changes, additions and deletions for each contribution in both cases.

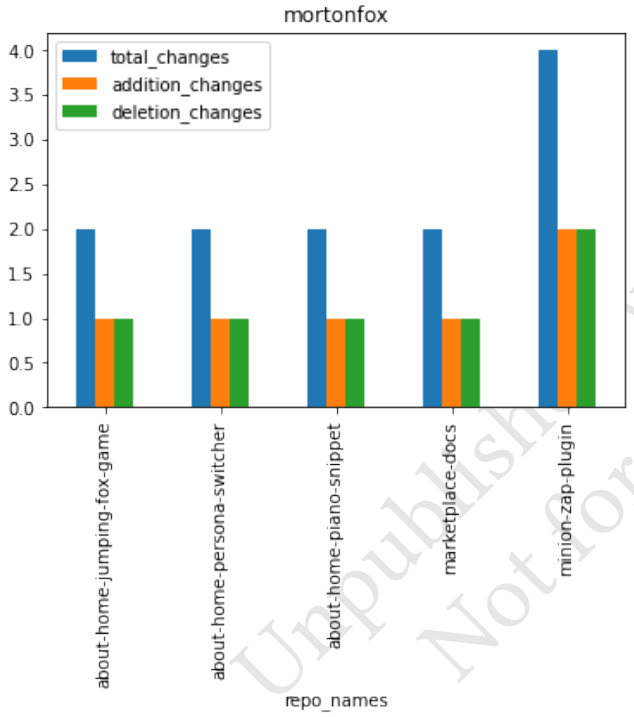

Figure 7: MortonFox Developer contributions

4 FUTURE DIRECTIONS

In this, I only included the total changes, additions and deletions but not the actual file details because of inconsistencies in the data and also in each contribution, some users have made changes in more than one file. So we can extend this study to study the actual files which users have changed. And also I took more than two months in extracting the data. It might be because of the modules that I have used(requests, JSON, pandas, Github API) which are slow and have a restriction on the number of extracts in given time duration. Another reason for this is inconsistency with the Github data which causes random errors and stops the program execution,

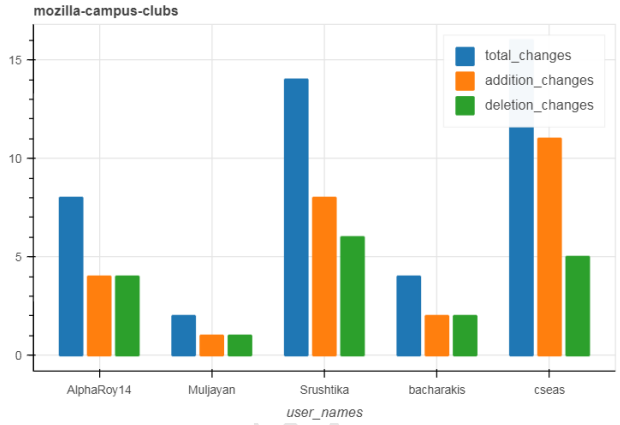

Figure 8: Mozilla Campus Clubs Repository Contributors

but when it resumed again from that file, it would run successfully but would stop at a later stage of 25 to 30 repositories after the current one. This caused a lot of trouble and I had to skip some of the contributions which are old due to the reason of inconsistent data from Github API.

Github also has a restriction of a maximum of 30 commits or pulls data on a page due to pagination issue and a maximum of 1000 repositories. This would also cause a lot of issue in getting the data. So in the Extension of this study, I can focus on better mechanisms to extract the data which speeds up the process but they also have some compatibility issues with other packages. This study could also be extended to study on general various topmost repositories from all over Github.

5 APPENDICES

ACKNOWLEDGMENTS

This study is an extension of *An empirical study of the first contributions of developers to open source projects on GitHub by Vikram N. Subramanian[11]*

REFERENCES

[1] [n.d.]. Mozilla Organization on Github. https://github.com/mozilla.
[2] 2021. Bokeh Interactive Plots. https://docs.bokeh.org/en/latest/index.html.
[3] 2021. CSV Python. https://docs.python.org/3/library/csv.html.
[4] 2021. Github API. https://docs.github.com/en.
[5] 2021. JSON Python. https://docs.python.org/3/library/json.html.
[6] 2021. Matplotlib Library. https://matplotlib.org/.
[7] 2021. Numpy. https://numpy.org/.
[8] 2021. Pandas. https://pandas.pydata.org/.
[9] 2021. Requests Package. https://pypi.org/project/requests/.
[10] Igor Steinmacher, Igor Scaliante Wiese, Tayana Conte, Marco Aurélio Gerosa, and David Redmiles. 2014. The Hard Life of Open Source Software Project Newcomers. In *Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering* (Hyderabad, India) *(CHASE 2014)*. Association for Computing Machinery, New York, NY, USA, 72–78. https://doi.org/10.1145/2593702.2593704
[11] Vikram N. Subramanian. 2020. An Empirical Study of the First Contributions of Developers to Open Source Projects on GitHub. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings* (Seoul, South Korea) *(ICSE '20)*. Association for Computing Machinery, New York, NY, USA, 116–118. https://doi.org/10.1145/3377812.3382165

,