The battle of data science languages: Are R users moving to Python or vice versa?

An analysis using GitHub activity data

Julian Cabezas
Master of Data Science
University of Adelaide
Adelaide, Australia
a1785086@student.adelaide.edu.au

ABSTRACT

Python and R are two of the most popular languages used for data science. While R's popularity has slightly declined in the last years, Python is becoming one of the most popular languages overall. This study seeks to determine if the R users are moving to use Python more frequently or vice versa. An analysis of longitudinal GitHub data using a ratio index between the number of push actions in R or Python for each user was developed. Monthly data from 2015-2019 was analysed using linear regression to determine how many users were migrating from one language to the other. The results indicate that in general long-term users tend to stay in either Python or R exclusively, but a small fraction of R users, about 11% of exclusive R users at the beginning of the time series. are migrating to Python, while on the other side the migration from Python to R is relatively small (less than 1%). The conclusion of this study was that only a small fraction of the Python rise in popularity can be explained by the migration from R users to this language.

KEYWORDS

R, Python, GitHub, Data Science

1 Introduction

The field of data science has been in a steady growth in the last decade, presenting an increasing demand of developers and tools for data analysis, that are being implemented in several languages. R and Python are some of the most used languages in data science and machine learning [11], both being open source and interpreted languages, with dynamic typing and counting with a huge variety of data science and big data related tools. While R is frequently used in the academia and has a focus on statistical analysis, Python is used as a general-purpose language, that has matured to count with data analysis libraries like *numpy* and *pandas*, being widely used for machine and deep learning. The features and easiness to use of both languages are usually compared in the literature on the subject [9], as both languages are used with very similar objectives.

Several methods are being proposed to measure programming language usage, being the TIOBE index one of the most cited. In this index, the R language reached the top ten of programming languages in January 2018 (position #8), presenting a consecutive decline in popularity and ending the position #10 of the ranking in May 2020. On the other hand, Python has experienced a steady growth in popularity from 2017 onwards, reaching position #3 in May 2020 [18].

To study developer usage of a programming language, there are various options available, being the GitHub archive database one of the most useful databases for this purpose, as it contains continuous data of all variety of user events and activities related with the Git version control system from 2012 to the present [5], mainly containing open-source projects, as it only shows the public repositories. The database is accessible both via an API or via the Google BigQuery platform.

The GitHub database has been used by several researchers to study the dynamics of the user contribution to a project [10], the characteristics and language of the code in a particular field of study (e.g. Bioinformatics [12]) and even the effect of prior social links in the onboarding of developers in open source projects [3]. The transitions of users from one languages to another, using the GitHub database as panel or longitudinal data (data on the same individuals taken on different moments in time [4]), has not been yet studied, although some have used this database to study the behaviour of users in time [14]

This study aims to describe the behaviour of long-time and constant R and Python users and how they use one or both of the abovementioned languages, looking to find if movements or migrations occur between these both languages. For this purpose, the GitHub archive of data will be used.

2 Motivation

The results of this research can have a broad spectrum of applications. On one side, enterprises could use these results to make decision investments on any of these two languages (or both). In the same line, companies involved in the development of these languages (e.g. Microsoft, Google, RStudio) could put more resources to develop integration methods between these two languages if their customers are turning into mixed R and Python users.

On the side of students and beginner programmers, this research can be useful to choose a programming language to learn, as it can determine the extent in which one language's popularity could be affecting the user base of the other, and potentially the help a programmer can find in the language's community

3. Background

R and Python are very popular languages in the data science field, and together with some commercial alternatives such as SAS, comprise the toolbox of a modern data scientist [1] or machine learning engineer. While they share many characteristics, as being open source, their origin is dissimilar and in general they fulfill different purposes

Python, on one side, was created by Guido van Rossum and originally released in 1991. It is commonly used for web and desktop applications, as well as for numeric and scientific computation. It is commonly praised for being an friendly and easy to learn language, because of that, universities and high schools are usually using it to teach programming lessons [7]. During the course of the current century the use of Python has increased due to its use in the field of data science and machine learning, due to the appearance of several useful libraries, such as *numpy* and *scipy* for scientific computation, *pandas* for data handling and *scikit-learn* for the development of machine learning models, together with the availably of several Python APIs for big data (PySpark) and deep learning (TensorFlow)

R, on the other side, was initially created by Robert Gentleman and Ross Ihaka at the University of Auckland, in New Zealand. This language was inspired by the S language. R is considered a statistical analysis and graphics environment, as well as a programming language. The base language is complemented by the contribution of more than 5000 packages, that are mainly created by the academic community, focusing mainly in statistics [16]

One characteristic that both languages share is being open source. Open source, according to the open source definition, means that the code follows the principles of free redistribution, an openly available source code and must allow derived work (modification and improvements in the code, among other requirements [17].

The development of open source software is a collaborative effort, that many times needs a decentralized version control system to work, being Git one of the most popular. In this kind of system, every user writes code to his own local machine, where that can perform changes to the project, called commits, that represent a delta between the parent version of the project and the local version [10], allowing every collaborator to track all the changes made to the project in time.

Most of the work done in Git requires a platform in which the developers can get together and collaborate. One of the most popular platforms is GitHub. This platform allows the developers to collaborate in particular projects, that are grouped in public or private "repositories". GitHub is currently owned by Microsoft and is used by many big companies such as Facebook [8], while at the same time several developers host personal projects in this repositories.

Along the evolution of a project in GitHub, the developer has to perform several actions to manage the different versions of the code through Git, improving and making the code evolve into different versions, as well as fixing bugs. Each of these actions is recorded in GitHub, and make part of the history of the repository. Along the many actions that are used in this version control system, one of the most common ones is the execution of the 'push" action, when the developer pushes the commits made in their local repository to the remote GitHub repository, updating the code. All of the events and actions performed by GitHub users on public repositories are stored in the GitHub archive database, available to the public via BigQuery, a cloud data warehouse system managed by Google.

4. Research Method

4.1 Research Questions

This research aims to investigate the following research questions: RQ1) Are programmers using R and Python exclusively or a mixture of both? RQ2) Are the R users moving to Python or vice versa? RQ3) How many programmers are following these trends and how?

The research question proposed in this study is mainly descriptive (characterise the activities of users and their migration between these two languages) and also aims to find causality (can the decline of R usage be explained by the usage of Python?). To answer these questions, a positivist approach is proposed, seeking to understand certain phenomena mainly using quantitative data. Thus, this approach is used to determine the effect of time in the preference of users, breaking this complex problem into simple and easy to measure units (behaviour of single users in GitHub). Moreover, this study seeks to generate knowledge by using statistical methods to verify a theory (R users migrating to Python).

4.2 Data Collection

The GitHub user was the unit of analysis of this study, but the unit of observation will be the activity data of the user (pull requests, watch action and issues). Our population under study are the open source R and Python developers, but we used use convenience sampling to get a sample from the easily available data in the GitHub platform, which is not used by all open-source developers but still contains a great fraction of them, and is possible one of the best sources of information for software engineering research [5].

The research questions were answered using the datasets contained in the "GitHub archive", that gathers the activity data of public repositories in the GitHub platform from 2012 to the present. This data was collected using the Google Big Query platform, using the

monthly datasets from 2015 to 2019 (both years included), covering a total of 60 months.

The data that was collected in this case was the "push action", that can reflect how the user interacts with the repository on a day to day basis, reflecting the usage of a language. Thus, to obtain the database to be analysed, for each of the monthly GitHub databases, the records of data that contained the "PushEvent" label were counted for each user and repository combination, recording the number of push actions for all repositories and all users.

Given the fact that the push actions in the GitHub archive do not contain the language of the repository, the "pull request" actions were also collected to obtain the main language of the repository, as recorded by GitHub, meaning that the repositories that were analysed are the ones that register at least one pull request from 2015 to 2019. This data was included into the abovementioned push database to obtain the language of the repository and thus, the number of push events of each user for each programming language.

The initial push database had a total of 135,075,877 records, comprising 1,063,854,119 push actions. On the other hand. 10.052.304 distinct repositories were identified using the push request analysis, were 746,586 (7.81%) correspond to Python repositories and 38,727 (0.39%) to R repositories. Once filtering the total dataset of push actions to find the ones performed in either R or Python, the remaining corresponds to 2,911,144 records of data, comprising 41,442,838 push actions, from which 2,086,283 (5.03%) were performed in R repositories and 39,356,555 (94.97%) in Python repositories. The raw database contains a total of 633,677 distinct R or Python users.

4.3 Data Analysis

The objective of this study is to investigate the mobility of existing Python or R users between these two languages, and not the data coming from new users, that can be influenced by particular decisions (e.g. Computer science lecturers or high school teachers deciding to teach Python [7]) or the recent rise in Python's popularity, along to filter for casual GitHub users or abandoned accounts [5]. For this purpose, we selected users with continuous activity in the study period (2015-2019). To perform this, the collected data was filtered to leave GitHub users with at least 30 months of data (half the months of the data set) either in R or Python repositories. The filtered data then consisted of a longitudinal or panel dataset (several observations of the same individual in time), suitable for transition analysis according to Croissant *et al.* [4].

With this filtered dataset, a ratio index that we called *R and Python Ratio Index* (RPRI) was calculated to address the proportion of actions performed with R in relation to the total amount of actions in R and Python repositories (Equation 1). This ratio index was calculated for each user and for each month. As this index is constructed to address for the total of R and Python push actions of the user, it presents values ranging from one to zero, being one if all the recorded actions are performed in R repositories, and zero if all the actions are performed in Python repositories.

$$RPRI = \frac{\#\ R\ pull\ actions}{\#\ Python\ pull\ actions + \#\ R\ pull\ \ actions} \quad (1)$$

Using this index, a classification of users was constructed based on their usage of Python or R in each year of the study, classifying them in five different categories according to their mean *RPUI* in the year (Table 1), these categories were constructed looking at the distribution of the data, that has a concentration of users as exclusive Python or R programmers (details in the Findings section), while the intermediate categories were constructed looking to find fixed length intervals of the RPRI index. With the classified user database, the number of users that fall in each category in each year was counted, determining the transitions between groups and answering RQ1 and RQ2.

Table 1: Classification of users according to their RPRI

R and Python Ratio Index (RPRI) Range	User classification
[1]	Exclusive R user
(1, 0.6666]	Mostly R user
(0.6666, 0.3333]	Mixed R and Python user
(0.3333, 0)	Mostly Python user
[0]	Exclusive Python User

To answer RQ3 the monthly data was used. The trend in the RPUI index was analysed for each user, using linear regression between the RPUI and the time from the beginning of the time series (expressed in months), in a technique often called "regression on time" [6], then, the significance of the slope was analysed using an ANOVA test to determine if the slope was significantly different to zero [15]. Thus, a significant negative trend was found if the user was moving to from R to Python, and a negative one if he/she was moving from Python to R. This method was chosen because of the simplicity of the interpretation of results, allowing us to analyse not only the significance, but the sign and magnitude of the slope, determining the rate in which the user might be changing from one language to the other.

Once the trend sign and the significance of the slope were tested for each user, the number and proportion of users with a significant positive and negative sign was be calculated and related (crossed) with the abovementioned classification of users, addressing the number of users and the rate in which the users are moving from one language to the other, answering RQ3.

5. Findings

The initial dataset of push actions in R and Python repositories had a total of 633,677 distinct GitHub users, but once the threshold of at least 30 months of activity between 2015-2019 was applied, the resulting database was restricted to 13,542 GitHub users (2.14%). The push activity of these filtered users is mainly concentrated into Python, and has a high dispersion, with a mean of 26.38 push actions per month, and a standard deviation of 148.72. The distribution of these values, filtering a few values over 500 for visualization purposes, is shown in Figure 1.

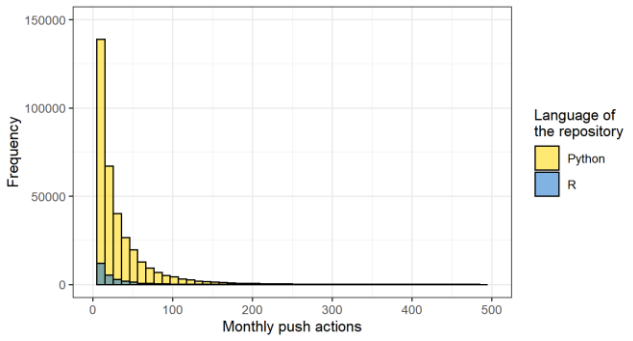

Figure 1. Distribution of the monthly push actions per user

RQ1: Are programmers using R and Python exclusively or a mixture of both?

The RQ1 was answered by the calculus of the R-Python Ratio Index (RPRI), that tells us that in the vast majority of the time the users are using either R or Python exclusively, being the months were the developers are using both languages very scarce. Figure 2 shows the distribution of monthly values in the dataset values in the data.

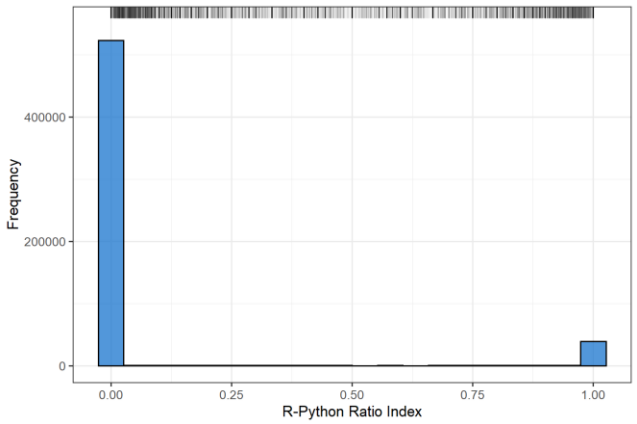

Figure 2. Distribution of the R-Python Ratio Index

This distribution can be complemented with the fact that during the complete time series, 12165 (89.83%) of the developers stayed as exclusive Python user, while 685 (5.05%) of the developers stayed as exclusive R users between 2015-2019, without producing one single push action in the other language. The fraction of the users that has used both languages at least once is just 5.1% (690 users). Thus, the answer to the RQ1 is that the vast majority of the developers stays in either R or Python, being a small fraction the ones that mix both at any moment of time.

RQ2: Are the R users moving to Python or vice versa?

The classification of users given their R-Python ratio index shows, as well as in the above-mentioned case, that the majority of users maintains their language over time. Figure 3 shows the transition between categories in time. For visualization purposes, the categories of Mixed users and mostly Python or R users were collapsed into a single Mixed user category

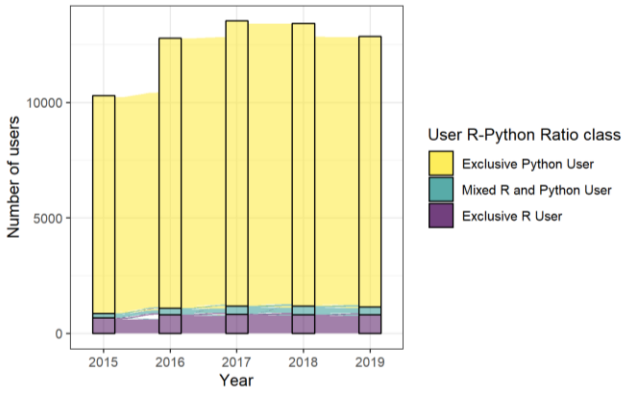

Figure 3. Transition of users between R-Python Ratio index categories

When removing the influence of the users that stayed as exclusive R and Python users through all the time series data, we remain with only 690 users. Analysing their behaviour, it is possible to see that the changes between categories are not abrupt, observing that in general these users tend to remain in one category from one year to another (Figure 4), only presenting subtle category changes.

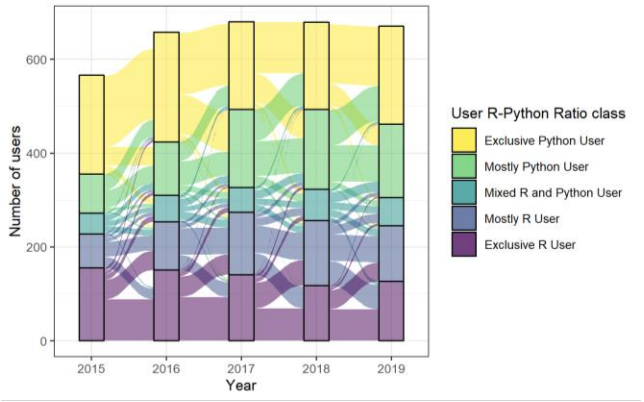

Figure 4. Transition of users between R-Python Ratio index categories, filtering exclusive R or Python users in the complete data series.

As the data does not show clear transitions between groups, the regression in time analysis in RQ3 will determine if the R users are indeed moving to Python.

RQ3) How many programmers are following these trends and how?

The trend analysis linear regression (Table 2) shows that, of the total users that experience language changes, (690) only 127 showed positive significant slope ($p < 0.05$) and 125 a negative slope in the linear regression ($p < 0.05$).

Table 2. Results of linear regressions of the R-Python Ratio Index (RPUI) in time

Trend of the RPUI Index	Number of users	%
Exclusive R user (RPUI = 1 in the complete time series)	685	5.05%
Significant positive slope (moving from Python to R	127	0.94%
Changes in RPRI, but the trend is not different from zero	403	2.97%
Significant negative slope (moving from R to Python	152	1.12%
Exclusive Python user (RPUI = 0 in the complete time series)	12,175	89.83%

The analysis of the users that present significant slopes shows those who transition from R to Python (Negative slope values in the R-Python ratio index in time) show slope values of bigger magnitude, being the mean slope -0.0125 (meaning a rate of 1.25% of language conversion per month), and presenting a maximum of -0.0444. As we can see in Figure 5, it is possible to observe that some users quickly change from R to Python in the period comprised in this study (2015-2019)

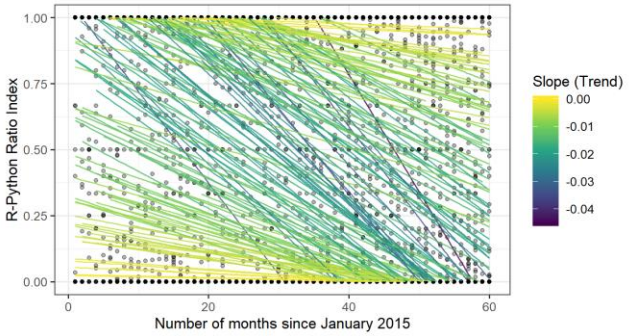

Figure 5. Trends of users moving from R to Python

On the other hand, the users that experience a significant positive slope (moving from Python to R), present smaller slope values (considering the absolute value), showing that the Python users might be trying the R language, but not changing entirely in most cases (Figure 6). In this case the maximum slope value was 0.0318, with a mean of 0.0108 (1.01% ratio of conversion per month)

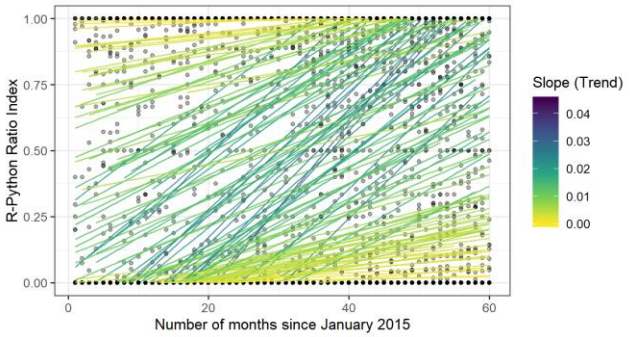

Figure 6. Trends of users moving from Python to R

When combining both results it is possible to appreciate, by looking at the initial year of the study (2015), that the number of users that use Python exclusively but are going towards R (67) is relatively similar to the opposite case (exclusive R users going towards Python, 74). Although these absolute numbers are similar, the percentage of exclusive R users in 2015 that present a trend towards Python and 11.08%, and the exclusive Python users showing a trend towards R is just 0.71% of the total Exclusive Python users. In the following year, the proportion of users follow the same trend (Table 3).

Table 3. User class and significant slope trends between 2015-2019

Year	User class	Significant slope from R to Python	Significant slope from Python to R	Total users
2015	Exclusive Py.	2 (0.02%)	67 (0.71%)	9437
	Mostly Py.	17 (20.48%)	13 (15.66%)	83
	Mixed	13 (29.55%)	15 (34.09%)	44
	Mostly R	22 (30.56%)	13 (18.06%)	72
	Exclusive R	74 (11.08%)	1 (0.15%)	668
2016	Exclusive Py.	14 (0.12%)	63 (0.54%)	11703
	Mostly Py.	25 (21.93%)	15 (13.16%)	114
	Mixed	23 (41.07%)	17 (30.36%)	56
	Mostly R	20 (19.42%)	21 (20.39%)	103
	Exclusive R	63 (7.79%)	8 (0.99%)	809
2017	Exclusive Py.	33 (0.27%)	41 (0.33%)	12341
	Mostly Py.	31 (18.67%)	25 (15.06%)	166
	Mixed	14 (26.42%)	10 (18.87%)	53
	Mostly R	27 (20.3%)	28 (21.05%)	133
	Exclusive R	47 (5.62%)	22 (2.63%)	836
2018	Exclusive Py.	49 (0.4%)	19 (0.16%)	12234
	Mostly Py.	37 (21.76%)	25 (14.71%)	170
	Mixed	18 (26.87%)	19 (28.36%)	67
	Mostly R	27 (19.57%)	33 (23.91%)	138
	Exclusive R	19 (2.34%)	31 (3.82%)	811
2019	Exclusive Py.	61 (0.52%)	0 (0%)	11717
	Mostly Py.	40 (25.48%)	32 (20.38%)	157
	Mixed	24 (40%)	23 (38.33%)	60
	Mostly R	24 (20.34%)	35 (29.66%)	118
	Exclusive R	1 (0.12%)	35 (4.32%)	810

Summarizing the obtained information, the answer to RQ2 is that the great majority of long-term Python and R users are not moving to the other language. Although it is possible to observe that a non-despicable fraction of R users (around 11% in 2015) are moving towards a more constant use of Python, while in the opposite directions the absolute numbers are similar, but do not represent a great proportion of the Python users (less than 1%)

6. Discussion

These first results show that the vast majority of GitHub's R or Python developers do not have a constant activity in the platform (97.86% using the 30 months of activity filter) , as many of them register isolated push actions in time, that do not add up to the defined threshold, this is recognized as one of the biggest risks of GitHub data [5], showing the importance of a correct filtering of the data.

The results show that in general Python users do not tend to experiment with the R language. This can be due to the fact that Python is a general-purpose language, that can be used in web applications (using, for example, the Django framework), or desktop applications, so this kind of users would not be interested in the R language. On the other hand, given the origin and characteristics of the R language, it is possible to say that all R users work on data analytics, statistics, data science or a related field. Thus, it a very likely that a R user would be interested on the Python's libraries or the dedicated APIs for machine learning or deep learning (such as TensorFlow), being a possible reason why there is fraction of long-term R users that are started using Python in the last years.

Given the small numbers of the R community when compared with the Python users, and the even small number of R users transitioning to Python. It is possible to say that the rise in Python popularity cannot be entirely explained by the transition of R users to that languages, even though the phenomena exists in low magnitudes and can be a small fraction of the explanation for the rise in Python's popularity.

The results indicate that the rate in which the R users are moving to Python are bigger that to the opposite direction. This fact could be due to the interest of R users to try features like the largely popular TensorFlow library, that counts with a mature Python API, or in other cases, due to the fact that available APIs in R perform much slower than the ones provided for Python, one example of this are the Apache Spark APIs, where Python largely outperforms R [13].

These results indicate that companies involved in the developing of the language should invest time in integration tools for the subset of users that are interested in using both languages, such as the *reticulate*, *rPython*, and *PythonInR* package (for Python usage in R) or the *rpy2*, *PypeR* and *PyRserve* libraries (for calling R from Python). Although as the results of these study show a greater interest in R users to use Python than vice versa, the efforts could be focused on the tools that integrate Python inside R.

7. Threats to Validity

Even though several considerations were taken into account to produce this research paper, there are several issues that can arise from the methods and data employed, that can constitute threats to the validity of this research

Firstly, even though the target population are the R and Python users, the sample was obtained from GitHub only, and only using public repositories. This can be problematic because this approach could be neglecting all the work that is done in private, either because the users can not disclose their code, or because he/she

does not want to show work in a language the he/she does not dominate, not allowing us to detect possible changes in programming language. Along that possible sampling problem, many other Git or version control platforms that could contain R or Python code exists and could have been used (e.g. Bitbucket and Gitlab), increasing the size of the sample.

The GitHub data also has a problem regarding the main language in the repository, and can confound R or Python repositories for other languages. In this study, only the main repository language was used, possible occasioning a problem when the repository has both Python and R, especially if a user is transitioning between languages. Moreover, a confounding factor that was not taken into account is the existence of Jupyter Notebooks, an interactive tool that can contain either R or Python (although more frequently the latter). In this case, as the GitHub repository is usually labelled as "Jupyter Notebook", it was not considered in this analysis, but these repositories could contain rich data for this kind of study.

The specific kind of GitHub action that was analysed (pull actions) can also be questioned if we consider that behind each push request many commits can be found, and these commits can be contributed by more than one user. In this case, the approach to take the user that finally pushed the commit to the repository was taken to link their GitHub account with the rest of his activity, but several user activities could have been missed by taking this approach.

Also, it is possible to notice that the GitHub Archive hosts data for more than 30 distinct actions and events, that could have been taken to this analysis. Some examples of actions that could have been considered are i) Pull requests (action when a developer requests a branch to be merged with the master branch) that according to Kalliamvakou *et al* [5], can be one of the most valuable sources of data of GitHub, reflecting the activity of experienced users. ii) Issue reports and comments of users in the repositories, that could reflect the activity of potential users of the repository. iii) Watch activity, as according to Sheoran *et al*. [14], "passive" users that watch repositories can eventually provide feedback, and a subset of the watchers of a project become active developers in time, and accounting for a great portion of the contribution in GitHub projects.

The statistical method that were employed in this descriptive analysis were chosen because of their simplicity, but it is important to note that not all the assumptions were tested. In the case of the linear regression method, for it to be valid, the assumptions of normality of residuals, constant variance and linearity of the relationship have to be tested, but were not tested in this study, fact that could treat the validity of the presented statistical results, a non-parametric methods should be considered [6]

One conceptual problem with the approach of this study is the fact that we are comparing users of a general-purpose programming language, as Python, with a language focused on statistics. This means that many of the Python users that were analysed might not ever heard of R, as it is out of the scope of their field, while the contrary cannot be said about R users knowing about Python.

Finally, even though a visual inspection of the data was performed, one of the aspects that was not considered in this study is the presence of bots or corporate accounts in the data, that according to Kalliamvakou *et al* [5] can produce noise in the data, possible threatening the validity of this study

8. Related Work

Although several analyses and surveys compare the R and Python figures, claiming sometimes that the rise of Python is at the expense of R (e.g. KD Nuggets [11] or State of the developer nation [2]). These studies use cross-sectional data, and no one has analysed the behaviour of users along time using longitudinal data, reaching conclusions that can be influenced by the appearance of new Python users

Focusing on the data science usage, the website KDnuggets made pools in 2016 and 2017 in which they ask for the most used data science/machine learning tools. Concluding that R users are in decline (42% to 36%) while Python users are increasing (from 34% to 41%) with the users that combine both languages representing 12% in 2017. It is worth mentioning that this survey is self-applied, so there is a high possibility of biases. Even considering this limitation, when comparing these figures with the current research, we can see that the amount of user that combine both languages is, in terms of order of magnitude. similar to the ones found here (5.1%), while in this case we found much more users in the Python language, it can be due to the fact that this study used the totality of the Python users, and not the ones focused in data science and machine learning, as the KDNuggets pool did.

In the scientific field, we can mention that this study reinforces the capability of the GitHub archive data to provide multitemporal longitudinal data, that was previously leveraged by Sheoran *et al*. [14], that investigated the behaviour of watchers in GitHub projects over time, that later became contributors of the project. Furthermore, this study reinforces the Kalliamvakou *et al* [5], that stated that a great part of the GitHub data is composed by abandoned projects and non-constant users.

9. Conclusions

This research has showed, in the first instance, that GitHub can be a very rich source of information, especially considering the long time series data it contains (2012 to the present). Moreover, this study shows that it is possible to perform longitudinal analysis with this kind of data (when considering appropriate data filtering technique), characteristic that could be harnessed in the future

The analysis of GitHub longitudinal data showed than the behaviour of Python and R users is rather stable, with the

developers generally staying in in the language they started with (considering 2015 onwards). In the case of the programmers going from one language to the other, the trend showed that the migration from R to Python has a bigger relative magnitude and speed. In conclusion, it can be said that a small fraction of the recent rise in Python popularity can be explained by the migration of users from R to this language, but probably the most important factors in this phenomena are others such as its use in educational, the rise in the use of machine and deep learning, web applications or others.

10. Future Work

As mentioned in the treats to validity section, there are many aspects that could be included in future works in this topic. In the first instance, a revision of the repository language labelled by GitHub should be considered, to include repositories where Python or R are being used but that are not detected by GitHub, with a special emphasis in the Jupiter Notebook repositories, that can include either of these languages. Also, an effort to include projects and users in other platforms such as GitLab or Bitbucket should be considered in order to get a wider sample of users. The data from these sources should also be carefully examined to detects bots or corporate accounts, that could cause noise in the analysis

One of the aspects that could be improved for future research is an analysis of the specific scope of the Python R repositories. For example, by analysing if a Python user includes or imports common data science focused packages, such as *numpy*, *pandas*, *scikit-learn* and *scipy*, a classification of users according to their work area could be developed. Moreover, a future research could identify the Python users that are focused in data science projects, as this kind of Python users could be also interested in R for its niche statistical methods.

Finally, regarding the statistical analysis that were performed, the use of linear regression could be questionable, so in future research the assumptions of this statistical method should be tested, and in case of these not being met, the use of non-parametric methods such as the Mann Kendal non-parametric test, as described in Kocsis *et al*. [6] should be used. On the other hand, in the case the time series shows a strong seasonal component or can be divided into different periods according to trend, other more complex methods based on ARIMA [15] can be tested.

ACKNOWLEDGMENTS

The author would like to thank Google and the data science company Kaggle, for providing a more than generous free quota for Google BigQuery analysis, that made this research possible.

REFERENCES

[1] Brittain, J. et al. 2018. Data Scientist ' s Analysis Toolbox : Comparison of Python , R , and SAS Performance. *SMU Data Science Review*. 1, 2 (2018), Article 7.

[2] Carraz, M. et al. 2019. *State of the Developer Nation 16th Edition*.

[3] Casalnuovo, C. et al. 2015. Developer On boarding in GitHub: The role of prior social links and language experience. *2015 10th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, ESEC/FSE 2015 - Proceedings* (Bargamo, Italy, 2015), 817–828.

[4] Croissant, Y. and Millo, G. 2019. *Panel Data Econometrics with R*. John Wiley & Sons Ltd.

[5] Kalliamvakou, E. et al. 2014. The Promises and Perils of Mining GitHub. *Proceedings of the 11th Working Conference on Mining Software Repositories* (Hyderabad, India, 2014), 92–101.

[6] Kocsis, T. et al. 2017. Comparison of parametric and non-parametric time-series analysis methods on a long-term meteorological data set. *Central European Geology*. 60, 3 (2017), 316–332. DOI:https://doi.org/10.1556/24.60.2017.011.

[7] Kui, X. et al. 2017. Research on the improvement of python language programming course teaching methods based on visualization. *ICCSE 2017 - 12th International Conference on Computer Science and Education* (2017), 639–644.

[8] Newton, O.B. et al. 2018. Developing Theory and Methods to Understand and Improve Collaboration in Open Source Software Development on GitHub. *Proceedings of the Human Factors and Ergonomics Society 2018 Annual Meeting* (Philadelphia, PA, 2018), 1118–1122.

[9] Ozgur, C. et al. 2017. MatLab vs . Python vs . R. *Journal of Data Science*. 5, (2017), 355–372.

[10] Padhye, R. et al. 2014. A study of external community contribution to Open-source projects on GitHub. *11th Working Conference on Mining Software Repositories, MSR 2014 - Proceedings* (Hyderabad, India, 2014), 332–335.

[11] Python overtakes R, becomes the leader in Data Science, Machine Learning platforms: 2017. *https://www.kdnuggets.com/2017/08/python-overtakes-r-leader-analytics-data-science.html*. Accessed: 2020-03-28.

[12] Russell, P.H. et al. 2018. A large-scale analysis of bioinformatics code on GitHub. *PLoS ONE*. 13, 10 (2018), 1–19. DOI:https://doi.org/10.1371/journal.pone.0205898.

[13] Salucci, L. et al. 2016. Lightweight Multi-language Bindings for Apache Spark. *Euro-Par 2016: 22nd International Conference on Parallel and Distributed Computing* (Grenoble, France, 2016).

[14] Sheoran, J. et al. 2014. Understanding " Watchers " on GitHub. *Proceedings of the 11th Working Conference on Mining Software Repositories* (Hyderabad, India, 2014), 336–339.

[15] Shumway, R.H. and Stoffer, D.S. 2011. *Time series analysis and its applications*. Springer.

[16] Stowell, S. 2014. *Using R for Statistics*. Apress.

[17] The Open Source Definition: 2017. *https://opensource.org/osd*. Accessed: 2020-05-26.

[18] Tiobe Index: 2020. *https://www.tiobe.com/tiobe-index/*. Accessed: 2020-03-26.