On the Relationship Between the Developer's Perceptible Ethnicity and the Evaluation of Contributions in GitHub

by

Reza Nadri

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2020

Authors Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public

Statement of Contribution

This project is joint work with a postdoctoral fellow, Dr. Gema Rodriguez Perez. The algorithms and implementations of this study was done by the author, Reza Nadri. Dr. Rodriguez Perez helped with brainstorming sessions and developing ideas and suggestions. She also helped with reviewing and making suggestions for the writing of the thesis. I really enjoyed working with Gema and I appreciate all her help.

Abstract

Context: Open Source Software (OSS) projects are typically the result of collective efforts performed by developers with different backgrounds. Although the quality of developers' contributions should be the only factor influencing the evaluation of the contributions to OSS projects, recent studies have shown that diversity issues affect the acceptance or rejection of their contributions. **Objective:** This thesis assists this emerging state-of-the-art body on diversity research with the first empirical study that analyzes how perceptible ethnicity relates to the evaluation outcome of the contributions in GitHub. **Methodology:** We performed a large-scale quantitative analysis of the relationship between developers' perceptible ethnicity and the evaluation of their contributions. We extracted the perceptible ethnicity of developers from their names in GitHub using the tool, Name-Prism, and applied regression modeling of pull request data from GHTorrent and GitHub. **Results:** We observe that (1) among the developers whose perceptible ethnicity was captured by the tool, only 16.56% of contributors were perceptible as Non-White; (2) contributions from developers perceived as White have the highest acceptance rate; (3) being perceptible as White have a positive, and being perceptible as Asian, Pacific Islander, Hispanic, or Black might have a negative influence on the evaluation of the contributions. **Conclusion:** While we did not observe any conscious bias against any group, our initial analysis leads us to believe that there may exist an unconscious bias against developers with ethnicity perceptible as Non-White. Thus, our findings reinforce the need for further studies on ethnic diversity in software engineering to foster a healthier OSS community.

Acknowledgements

I would like to thank my supervisor Professor Nagappan for their consistent support and guidance during the running of this project. Furthermore, I would like to acknowledge Professor Godfrey for helping me and providing feedback on my work.

Many special thanks to Professor Shane McIntosh and Professor Rafael Oliviera for reviewing my thesis and their helpful suggestions.

I would like to express my gratitude and appreciation for all of the researchers colleagues in software analytics group at the University of Waterloo, especially Dr. Gema Rodriguez-Perez. I would not be able to finish this project if it wasn't for her constant help and support.

Finally, I would like to thank my family for supporting me during the compilation of this dissertation.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In any line of work, diversity regarding ethnicity, gender, age, or personality traits is beneficial beyond ethical reasons [1, 2, 3]. Particularly, in Software Engineering *(SE)*, diversity helps to address a problem from different perspectives, designs more robust software products, and seems to create more efficient teamwork [4]. Indeed, diversity has been recognized as a high value team property [5, 6, 7] and many companies have increased their efforts to create a more diverse team.[1][2][3]

More than ever, diversity can be seen in online collaborative coding platforms such as GitHub, because it attracts developers from all around the world to contribute to Open Source Software *(OSS)* development. Usually, the collaborative development cycle in GitHub starts with a developer submitting a contribution, e.g., a pull request. Then, a project member evaluates the developer's pull request to accept or reject the contribution. Notice that, through this thesis, we will refer to a developer submitting a pull request as the *Submitter* and a project member evaluating the pull request as the *Integrator*. It is believed that the integrator evaluates the pull request based on the quality or factors related to the quality of the source code being contributed [8].

However, the quality of the pull requests is not the only factor influencing their evaluation process in GitHub. Recent studies have demonstrated that specific diversity issues related to social [9], personality [10], gender [11, 5], and geographical [12] factors also affect the acceptance or rejection of pull requests. For example, Vasilescu *et al.* [5] found that pull requests from female developers have a lower chance of acceptance. Terrell *et al.* [11]

[1] https://diversity.google/
[2] https://www.microsoft.com/en-us/diversity
[3] https://diversity.fb.com/read-report/

found that among those who have identifiable gender outside the project, females have lower acceptance. Rastogi *et al.* [12] identified statistically significant differences in the pull request acceptance rate between developers from different countries. Recently, Iyer *et al.* showed that developers' personality traits also affect the pull request acceptance [10].

The influence of non-technical factors in the pull request evaluation process might contradict the apparent openness to OSS developers from around the world in GitHub. Furthermore, social psychology theories agree that individuals treat better and prefer working with others similar to them [13, 14]. Thus, it is reasonable to study whether there are unexamined non-technical factors in the emerging body of knowledge that may influence the pull request evaluation. Many different non technical factors have been studied [9, 5, 11, 12, 10], but the developers' ethnicity or race has not been examined. Therefore, in this paper we solve that gap studying the ethnicity as a non-technical factor.

Vasilescu et al.'s GitHub Survey [15] highlighted that some developers are aware of the ethnicity of their team members; and that 30% of GitHub developers have felt sometimes negative experiences due to diversity in terms of national origin, language, and ideology. These findings motivate the study of perceptible ethnicity as a non-technical factor in the pull request evaluation because it is important to understand whether GitHub developers experience the conflicts that have existed between ethnic groups through the history and the racial prejudice that stills remains in the world. Therefore, our paper extracts developers' perceptible ethnicity from their names in GitHub and studies whether a developer's perceptible ethnicity is correlated to the evaluation of pull requests in OSS development. Understanding if ethnicity based bias exists in the pull request evaluation process will be the first step in helping OSS developers take necessary steps to foster a healthy community.

We formed the following research questions for our study:

1. **RQ1: How many developers are there in each perceptible ethnicity?**
2. **RQ2: What is the distribution of the pull request acceptance rate among the perceptible ethnicities?**
3. **RQ3: To what extent does the developer's perceptible ethnicity affect the acceptance probability of a pull request?**

By answering these questions, we can help both contributors and team members and managers to incorporate more diversity which benefits both sides.

We analyzed more than four million pull requests from 46,191 projects and 493,170 developers in GitHub. We first identified developers' perceptible ethnicity based on their

names using the Name-Prism tool [16], which has an F1 score of 0.795. We then used GHTorrent alongside GitHub's developers API to extract pull requests and related features to link them with their respective developers. We used the regression techniques from past studies on the relationship between non-technical factors and pull request acceptance [5, 11, 12] to build regression models to assess the effect of developers' perceptible ethnicity on how likely it is for a pull request to be accepted (pull request acceptance probability).

We found that pull request submitter's perceptible ethnicity can influence pull request acceptance probability, and it can be used as a feature to predict whether a pull request gets accepted but with a small effect size comparing to other features. Our findings also reveal that developers who are perceptible as White have a higher acceptance rate (the number of accepted pull requests over the number of submitted pull requests).

The primary contributions of our thesis include:

- We empirically observed the relationship between developers' perceptible ethnicity and the evaluation of their pull requests in the OSS community in a collaborative platform such as GitHub.
- We demonstrate that there is a difference in the acceptance rate among different perceptible ethnic groups.
- We show that the pull request's submitter perceptible ethnicity can affect its acceptance probability, controlling for other variables.

The rest of the thesis is organized as follows. Chapter 2 discusses the background and related work. Chapter 3 presents our case study design, including the data collection, our dependent variable, and various independent variables. Chapter 4 presents the findings of our study. Chapter 5 discusses the results. Chapter 6 highlights the threats to validity, and Chapter 7 concludes the thesis and discusses future work.

Chapter 2

Related Work

2.1 Theoretical background

Bias is defined as a strong feeling of inclination or prejudice for something, someone, or a group in a way that is usually considered to be unfair [17]. Similarly, *ethnicity bias* is someone's bias based on another person's ethnicity. There are primarily two types of bias: conscious and unconscious. While conscious bias is a preconceived and unreasonable inclination, trend, feeling, or opinion, unconscious bias is a social stereotype formed outside people's conscious awareness.

This unconscious ethnicity bias can be seen in collaborative environments [18]. According to some social psychology theories, working in groups tends to trigger discriminatory behavior against individuals that are not a member of the group [13, 14]. For instance, *Similarity-Attraction theory (SA)* postulates that people prefer working with others similar to them [13] and *Social Identity and social Categorization theory (SIC)* suggest that people tend to categorize themselves into groups [14]. These theories suggest that members of one's group are treated better than outsiders. Furthermore, psychological research on dual-process theory claims that individuals use two different systems of thinking when making impressions and judgments [19]. One system is slower and more deliberate, while other is based on an individual's intuition or gut-feeling. This second gut-feeling system often becomes involved when there is enough available information about the target that activates an individual's stereotypical expectations.

Decades of social studies have demonstrated that ethnicity is an influencing factor in different social fields. In sports, experienced gymnastic judges ranked participants from

their nations higher than participants from other countries in international competitions during the 2013-2016 Olympic Cycle [20]. In academia, papers with authors from some regions receive fewer citations than papers from authors of other regions despite papers' quality [21]. In online platforms, if African-American people can easily be identified from their names when applying for a job, they need to send almost double the resumes, than people easily identified as White, to get one callback [22].

In online collaborative environments such as GitHub, developers might hold unconscious beliefs about various social groups that can be triggered by the perceived ethnicity derived from one's name. Such unconscious bias is far more prevalent than conscious prejudice and often incompatible with one's conscious values[19]. Therefore, we choose to study ethnicity bias in GitHub projects to analyze whether an integrator shows unconscious bias against the submitter's perceived ethnicity when evaluating the submitter's pull request. If such bias exists, it may influence the acceptance of software contributors.

2.2 Study of diversity and social factors in Software Engineering

To the extent of our knowledge, this is the first empirical study that identifies the diversity regarding perceptible ethnicity in OSS development and analyzes the relationship between developers' perceptible ethnicity and pull request evaluation.

However, recent studies have addressed other diversity issues in OSS contributions. Calefato *et al.* [23] study the personality of developers in large projects and categorize developers into three personality types. They find that personality traits are not changing over time, or with changing roles. Vasilescu *et al.* [5] identify the gender imbalance in *OSS*. They also found that gender diversity has a positive correlation with team productivity. In a similar work, Terrell *et al.* [11] found that for developers outside a project, men have a higher acceptance rate comparing to women. In another work, Califato *et al.* [7] study the gender of developers in *OSS*, and the effect of gender imbalance on community smells ("sub-optimal patterns across the organisational and social structure in a software development community that are precursors of such nasty socio-technical events").

In another body of work, researchers have tried to understand the pull request acceptance process and the factors influencing pull request evaluation. Tsay *et al.* [9] showed that project managers use not only technical factors but also social clues while evaluating pull requests. Prior interactions inside the project and "social distance" were important to the pull request acceptance process. Gousios *et al.* [24] studied the factors affecting

pull request acceptance on 1.9 million pull requests. The results reaffirmed the existence of non-technical factors involvement in the pull request evaluation process. Rastogi *et al.* [12] added geographical location to previous studies and studied 17 countries that have at least 1% of the total number of pull requests. They found that country of residence can influence pull request acceptance. They also found that when the submitter and merger are in the same country, the chance of pull requests getting accepted is higher. However, their study relies only on the country of the developers and does not address any form of ethnic diversity in the community. Furthermore, Iyer *et al.* [10] found that pull requests from developers who are more open and conscientious, but less extroverted, have a higher likelihood to be approved. They also found that developers who are more conscientious, extroverted, and neurotic have a higher likelihood of accepting a pull request.

Chapter 3

Methodology

To measure the effect of perceptible ethnicity on pull request acceptance, we mined data from projects in GitHub. We used GHTorrent [25] alongside GitHub's developers API to extract data. We collected data from projects, users, and pull requests. We then selected a subset of the dataset to continue our study. We finally used the collected data to build regression models for analysis. Fig 3.1 briefly depicts a summary of our methodology.

3.1 Project Selection

Although GitHub has 125, 486, 232 projects and more than 52 million pull requests[1], not all of the projects are interesting to study. To make sure that we excluded trivial projects (e.g. homework assignments) from our analysis, we only selected a subset of the projects. To obtain this subset, we used the published dataset from RepoReapers in 2017 [26]. RepoReapers uses score-based and random-forest classifiers (trained on two datasets, organization and utility dataset) to determine whether a project is non-trivial, which outperforms other approaches with high precision (82%) and high recall (86%). We chose the projects with more than ten stars, which at least three classifiers had classified them as non-trivial.

[1] According to GitHub's data publicly available on June 2019

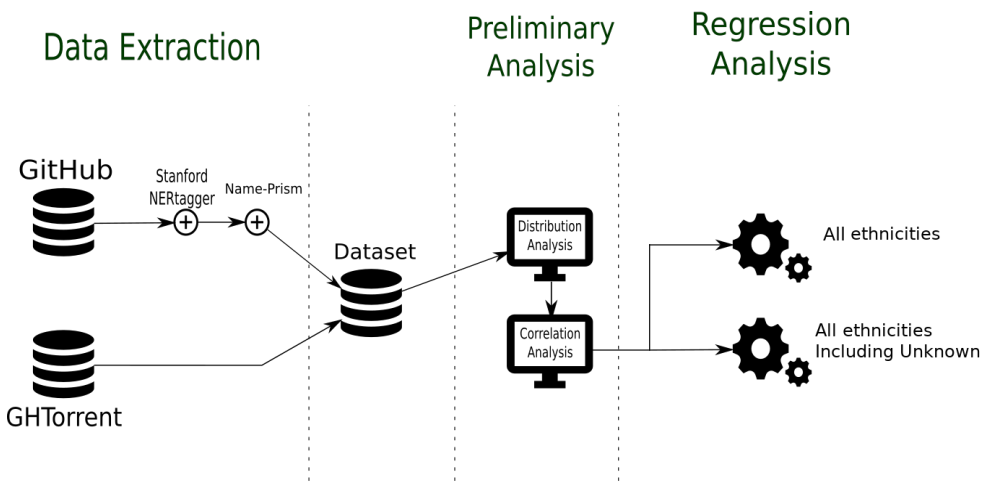


Figure 3.1: Brief Methodology and Data Collection Process

3.2 Pull Request Selection

We inferred a pull request acceptance using the pull request's status. The status is collected directly from GitHub API. A pull request's can be either open, merged, or not-merged (rejected). We used the merge time field to determine whether a pull request is merged (accepted). If the merge time is null and the pull request is closed, we considered it as not-merged. If the merge time is not null and the pull request is closed, we considered it as merged. Otherwise, when the pull request's status is not closed, we considered the pull request as open. We know for sure that if the pull request is merged, then the merge time would not be null. However, we cannot detect cherry-picked pull requests, where only parts of the pull request are merged. In that case, the pull request is labeled as not-merged, and the merge time would be null. In total we extracted $4,029,190$ pull requests.

Different developers can participate in a pull request. The developer submitting the pull request is the ***submitter***, the developer closing the pull request is the ***closer***, and the developer merging the pull request is the ***merger*** [12]. In this thesis, we use the term ***integrator*** to replace the merger (when the pull request is accepted) or the closer (when the pull request is rejected). Since the perceptible ethnicity does not have any effect on pull request acceptance when the submitter and the integrator are the same person, we removed the pull requests which have the same submitter and integrator. We also excluded open pull requests from our analysis because it may be accepted or rejected in the future. $1,521,599$ pull requests were removed by these filters. We labeled $2,039,601$ as merged, $467,990$ as not-merged.

Table 3.1: Number of projects, pull request, and developers identified in GitHub, after the first filtering (Section 3.1), and after the second filtering (Section 3.2).

Number of	GitHub	1st Filter	2nd Filter
Projects	125,486,232	46,191	37,762
Pull requests	52,018,443	4,029,190	2,507,591
Developers	32,411,734	493,170	365,607

Table 3.1 shows the number of projects, pull request, and developers in GitHub[2], after applying the projects' selection criteria and after applying the pull request's selection criteria.

3.3 Deriving ethnicity from name

We relied on the registered name of developers in GitHub to identify their perceptible ethnicity. In GitHub, the developer's name is an optional field. Therefore, developers can enter any valid characters as names. We started with 493,170 developers. To maximize the accuracy of our models, we identified developer's perceptible ethnicity using these tools:

1. **Stanford Named Entity Recognizer *(NER)***: First, we used the Stanford NER [27] to discover whether a set of characters includes names. In general, Stanford NER is a model that takes a set of names and labels, each of them as a class such as a person, organization, protein, etc. Stanford NER is a classifier based on linear-chain Conditional Random Field. There are multiple versions of Stanford NER for different classes and different languages. In this thesis, we used English Stanford NER with three classes. We used this tool to classify developers' names as either person, organization, or location. Our dataset only includes developers with at least one name labeled as a person. This step recognized 320,633 inputs as names.

2. **Name-Prism**: Second, we used Name-Prism [16] to infer the perceptible ethnicity of developers, using their names. Name-Prism introduces name-embedding and utilizes the concept of homophily to create a name-based perceptible nationality/ethnicity classification tool. Name-embedding, converts each name to a vector and tries to recognize contexts and similarity of names in the same context. The context in

[2]at June 2019

the case of name-embeddings is perceptible ethnicities/nationalities. Homophily is a term used in communication sciences, which alludes to the fact that people tend to communicate with similar people. Name-Prism uses this phenomenon in the context of instant messaging, i.e., people from an ethnic group tend to communicate with other people from the same ethnic group. By combining these two concepts and collecting 74M labeled names from 118 countries, Junting *et al.* created the most accurate classification tool to identify ethnicities with an F1 score of 0.795 [16]. The second best classifier *Ethnea* [28] has only F1 score of 0.580. Based on U.S. Census Bureau, Name-Prism uses six ethnic groups: White, Black, Hispanic, API (Asian, Pacific Islander), AIAN (American Indian and Alaska Native), and 2PRACE (Mixed Race) to build the classifier. It produces a confidence rate between 0 and 1 for each group. Name-Prism could identify ethnicity of all names but with different confidence levels, 282,312 with more than 0.8 confidence rate (Check chapter 4.1 for more details).

Furthermore, we manually evaluated the tool. We selected 25 random samples from each perceptible ethnicity except AIAN and 2PRACE (because we did not have enough samples for those groups) and manually identified their perceptible ethnicity using publicly available data (e.g., social media and search engines). Among the 100 developers, we could verify that 61 of them are certainly correct. The see whether a data is correct we used name, country, profile picture and any related data available on social medias, and the judgement was based on the author's own intuition. For 34 developers, we could not find additional information online, and therefore, we could not verify the result. However, our own perception of the ethnicity (based on names) and general search for people with similar names matched the results from the tool. For two developers identified as Black, we could not verify the correctness of the results. For just one sample, we could detect an obvious mistake where a White developer (based on the profile picture) was categorized as Black. Table 3.2 shows the result of the manual verification step.

3.4 Feature Selection

To explain the effect of perceptible ethnicity on pull request acceptance, we first need to find out what features or characteristics affect pull request acceptance. Prior work has grouped these features into three categories: project characteristics, developer characteristics, and pull request characteristics [24]. This categorization is based on prior work in the areas of bug triaging, developer recommendation, pull request and patch acceptance studies. The

Table 3.2: Manual Verification of Name-Prism results

Perceptible Ethnicity	Verified Online	Verified Generally
White	19	6
Black	16	6
API	19	6
Hispanic	6	19
Total	61	36

features that we have extracted are chosen from three sources [24, 9, 12]. The project level features that need access to the source code (e.g. number of test cases, lines of codes) are not included in this study because of the large number of projects. We collected features from pull requests and their respective actors and projects. Table 3.3 shows our collected features vs. features introduced in similar studies. To obtain the features, we have two primary sources. The first source is GitHub's API. The second source is GHTorrent public dataset until June 2019. For the features that are common with Rastogi *et al.* (including country) [12], we replicate their methodology to collect the features. The collection of features from Table 3.3 play the role of independent variables in the regression model. In other words, the regression model measures the effect of each feature, holding all other features fixed [29].

Table 3.3: Independent Variables

Feature	Literature	Description
Project Characteristics		
Repository's popularity	[9, 24, 12]	This variable shows the popularity of the repository at the time of pull request's submission. Measured using the number of stars
Repository's team size	[9, 24, 12]	The number of users associated with the repository in any way. A proxy for measuring repository's size.
Repository's maturity	[9, 12]	This feature shows how long (in months) the repository has been existed before the pull request.
External Contribution	[24, 12]	What percentage of the contribution was made by users outside the repository's community.
Submitter Characteristics		
Submitter's role	[9, 24, 12]	This feature indicates whether the submitter is a main member of the repository. Extracted using GitHub API.
Submitter's popularity	[9, 24, 12]	This feature indicates the popularity of the submitter. Measured by the number of submitter's followers at the time of pull request's submission.
Submitter-Repository association	[9, 12]	This feature indicates that whether the submitter and the repository have prior association. Whether the submitter watched the repository before the submission of the pull request.
Submitter-Integrator association	[9, 12]	This feature indicates that whether the submitter and the integrator have prior association. Whether the submitter followed the integrator before the submission of the pull request.
Submitter's experience	[24, 12]	This feature indicates the experience of the submitter, measured using the number of pull requests made by the submitter on GitHub.
Submitter's past success	[24, 12]	Measured using the number of accepted pull requests divided by the total number of pull requests made by the submitter, on GitHub.
Submitter's tenure	[12]	This feature indicates how long the user has been registered on GitHub at the time of pull request's submission.
Submitter's country	[12]	Submitters country of residence, based on the user's profile.
Same country	[12]	This feature indicates whether the submitter and the integrator reside in the same place.
Pull Request Characteristics		
Pull Request's changed files	[9, 24, 12]	The number of files changed by the pull request. A proxy to measure pull request's size.
Pull Request's comments	[9, 24, 12]	The number of comments on the pull request, a proxy to measure the the importance of the pull request.
Intra Branch	[24, 12]	This feature shows whether the pull request was made intra branch.
Pull Request's number of commits		Number of commits made by the pull request. A proxy to measure the pull request's size.
Pull Request's changed lines		The number of lines changed by the pull request. A proxy to measure pull request's size.
Submitter-Repository Experience		The number of pull requests submitted by the same submitter in the same repository before this pull request. This variable captures the experience of the submitter in the project, gained through time.
Ethnicity		
Submitter's perceptible ethnicity		Submitter's perceptible ethnicity.
Same perceptible ethnicity		This feature indicates whether the integrator and the submiter have the same perceived ethnicity.

Chapter 4

Results

4.1 RQ1: How many developers are there in each perceptible ethnicity?

Motivation: There is little empirical evidence of perceptible ethnic diversity involvement in GitHub. Thus, knowledge of the sampling distribution can be very useful in making inferences about the perceptible ethnicity groups in the community. This knowledge, could reveal insights about their demographics and could quantitatively demonstrate the presence or lack of diversity in GitHub's community.

Approach: To infer developers' perceptible ethnicity, we first extracted developers' names from GitHub and then used Stanford NER [27] and Name-Prism [16] to obtain the confidence rate of the developers' perceptible ethnicity. Name-Prism classifies first names and surnames to six different ethnicities, e.g., AIAN, API, Black, Hispanic, White, and 2RACE, with a confidence rate between 0 and 1, as explained in Section 3.

We assigned a unique perceptible ethnicity to each actor whether the confidence rate obtained from Name-Prism was equal or higher than 0.8. We chose this high confidence threshold because if one person could infer ethnicity from a name, if that inference is of high confidence, then anyone could infer. Otherwise, there might be a confusion. Therefore, when Name-Prism could not predict a perceptible ethnicity with more than 0.8 of confidence, or when NER did not tag a developers name as a person name, we classified their perceptible ethnicity as "Unknown". Nonetheless, we wanted to err on the side of caution and hence only tagged people whose name was classified with high confidence. However, we did not remove the "Unknown" data points from our dataset because it is

Table 4.1: Ethnicities population description

perceptible Ethnicity	Population	Proportion (%)
Unknown	210,858	42.8
White	235,541	47.8
API	33,776	6.8
Hispanic	12,356	2.5
Black	638	0.1
AIAN	1	≈ 0
2RACE	0	0

interesting to see the difference between acceptance rate and distribution of developers with publicly perceptible ethnicity and other developers.

Finally, to gain more insights about the demographics of the developers, we extracted their geographical location following the approach proposed by Rastogi *et al.* [12]. We also used "country-NameManager" script provided by Vasilescu *et al.* [5].

Results: From the 493,170 developers in our dataset, we classified the perceptible ethnicity of 282,312 developers (57.2%). However, 210,858 (42.8%) developers were classified as "Unknown". Table 4.1 shows that 47.8% of developers were perceptible as White, 6.8% as API, 2.5% as Hispanic, 0.1% as Black, and ≈ 0 as AIAN. We did not identify any developer with 2RACE as perceptible ethnicity.

Among 493,170 developers, we identified the country of 245,881 (49.85%) of them. Figure 4.1 shows the top countries identified based on the number of developers. North American countries: US and Canada, count for 17.24% of developers. The top European countries: Germany, UK, France, and the Netherlands, count for 13.03% of developers. Russia, China, and India also appear in the top countries, with 5.18% of developers.

Furthermore, we looked at the top five countries for each ethnic group based on the number of developers as a sanity check for the perceptible ethnicity classification method. This process results in only fourteen distinct countries since three countries (US, UK, Canada) were among the top countries of different ethnicities. Each cell in Fig. 4.2 shows what proportion of the ethnicity resides in each country. We observe (1) because of the high number of developers in the US, it represents high percentage for each ethnicity, but this percentage is higher for perceptible White ethnicity. (2) other than US, API

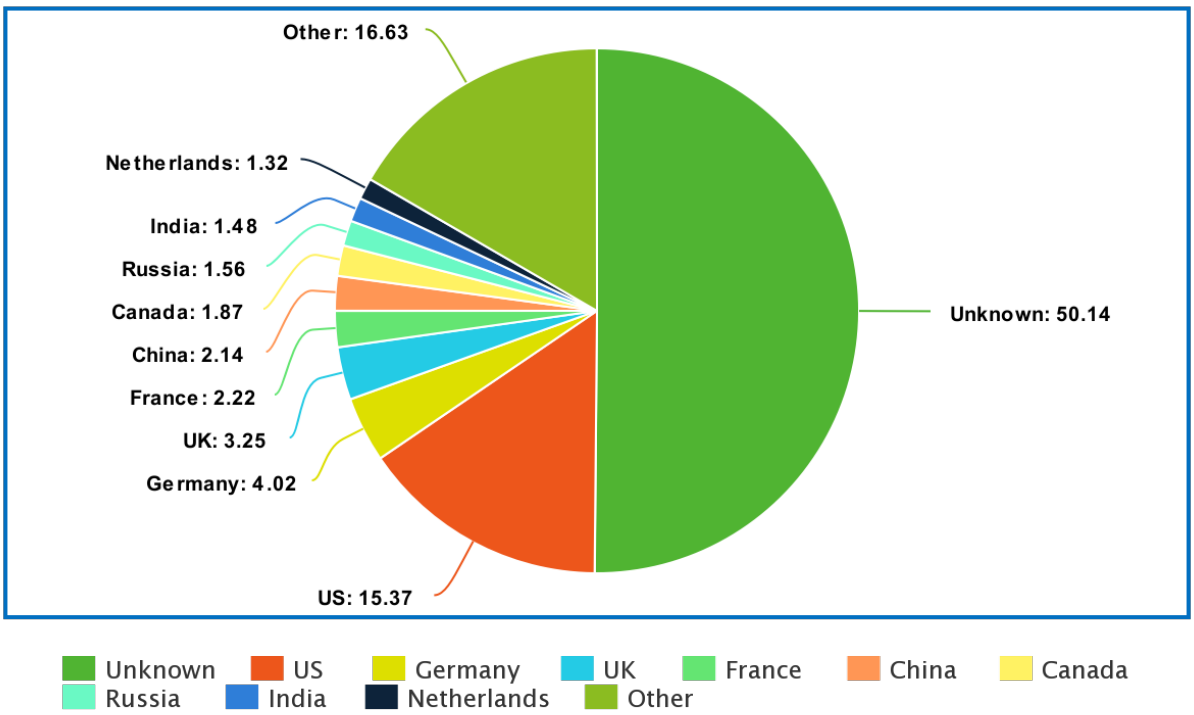


Figure 4.1: Top Countries according to the number of developers

developers mostly reside in Asian countries, (3) European and North American countries represent a higher proportion of White developers, (4) other than US, South American countries represent a higher proportion of Hispanic developers, and (5) other than US, African countries have a higher proportion of Black developers. These percentage numbers are normalised using RAS algorithm [30].

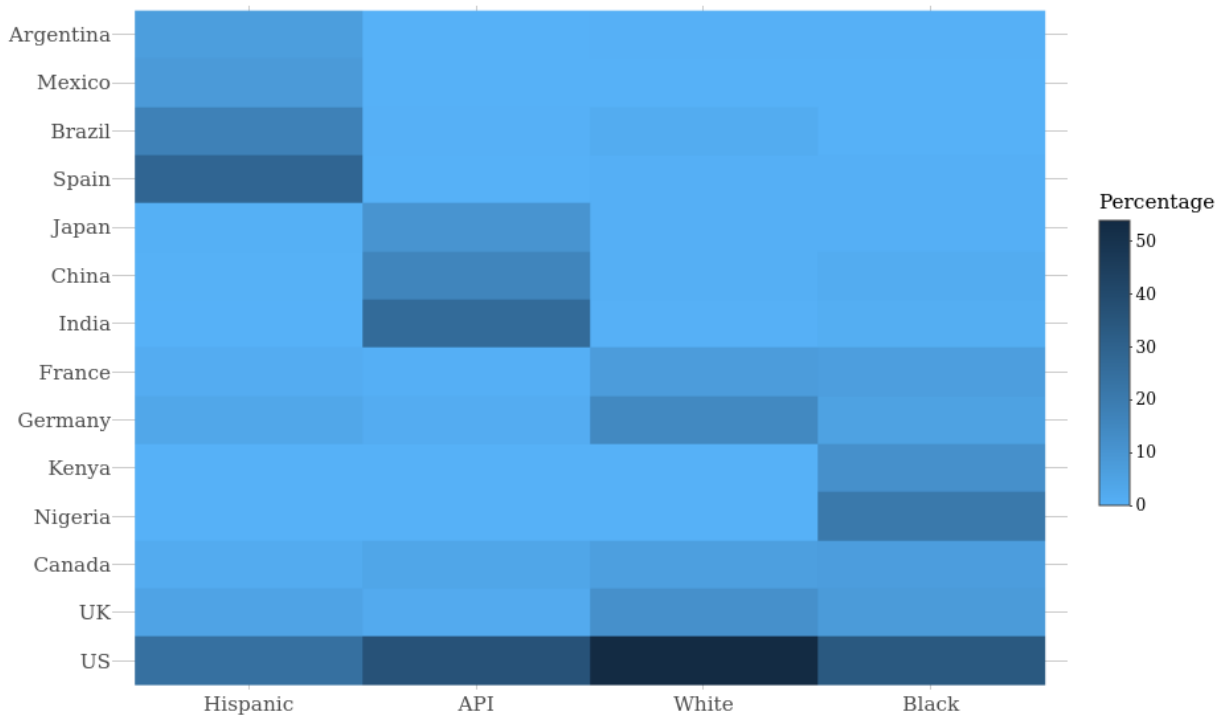


Figure 4.2: Perceptible Ethnicity population proportion in the top countries

4.2 RQ2: What is the distribution of the pull request acceptance rate among the perceptible ethnicities?

Motivation: We also need to understand the pull request acceptance rate of the different perceptible ethnic groups and their differences. The answer to this question is the first step to discover any potential bias toward different perceptible ethnic groups. Although, without more information, it is not possible to investigate the exact reasons behind the differences, looking at the performance of each group gives insight about the developers' perception of different groups (especially minorities) in the *OSS* community.

Approach: To calculate the acceptance rate for each perceptible ethnicity, we extracted the pull request acceptance status from the pull requests that survived the filtering explained in 3.2. Therefore, we identified the acceptance rate for each perceptible ethnicity

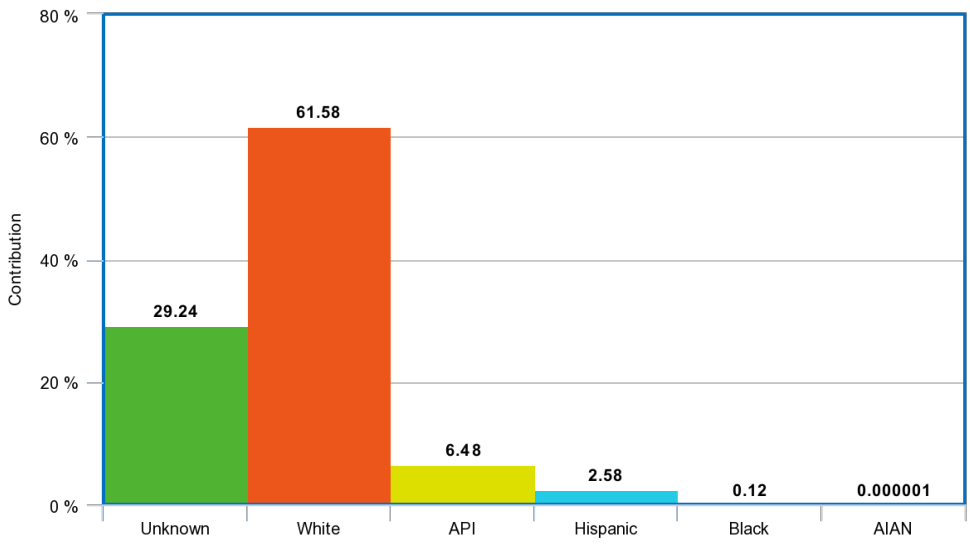


Figure 4.3: Percentage of pull request contributions per perceptible ethnic group

for the $2,507,591$ pull requests.

We used two distinct statistical tests to assess the results and compare different groups together. First, we applied the Kruskal-Wallis [31], which is a non-parametric distribution free test, to see if the differences between the averages are statistically meaningful. This test has been used in similar contexts [32]. Then we used the Dunn test [33] with Bonferroni correction [34] for pairwise comparison.

Results: Figure 4.3 shows the percentage of pull request contributions made by each group. While the vast majority of contributions (61.58%) were submitted by developers perceptible as White, developers perceptible as API, Hispanic, Black, and AIAN, in total, have submitted less than 10% of the contributions. 29.24% of contributions were submitted by developers with Unknown perceptible ethnicity.

Moreover, we analyzed the acceptance rate of each perceptible ethnicity to gain a deeper understanding of the difference between the perceptible ethnic groups. Figure 4.4 shows the general acceptance rate for each group. While the highest acceptance rate among all perceptible ethnicities is that of White with 82.6%, the successful acceptance rate of a developer with API, Hispanic, and Black perceptible ethnicity is 80.4%, 81.59%, and 81.34%, respectively. An interesting result shows that the lowest acceptance rate is that of Unknown perceptible ethnicity with 79.2%

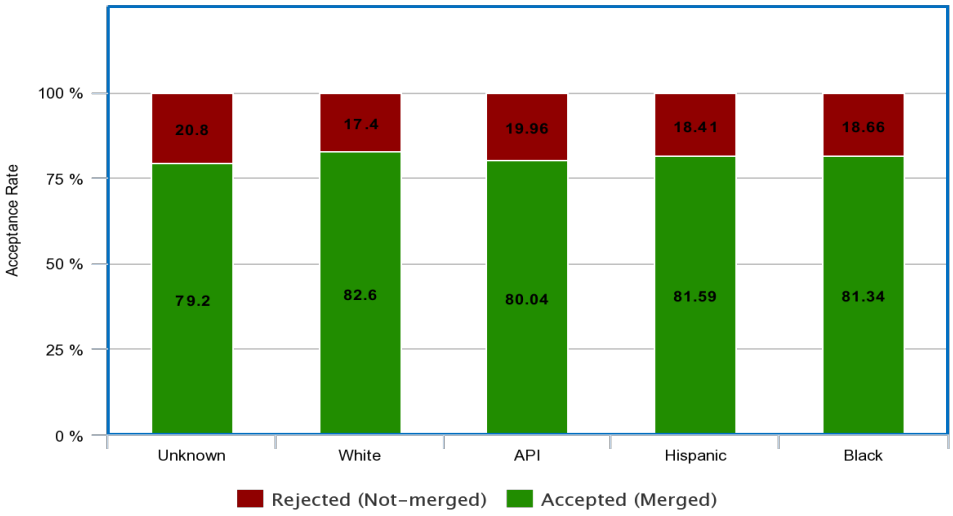
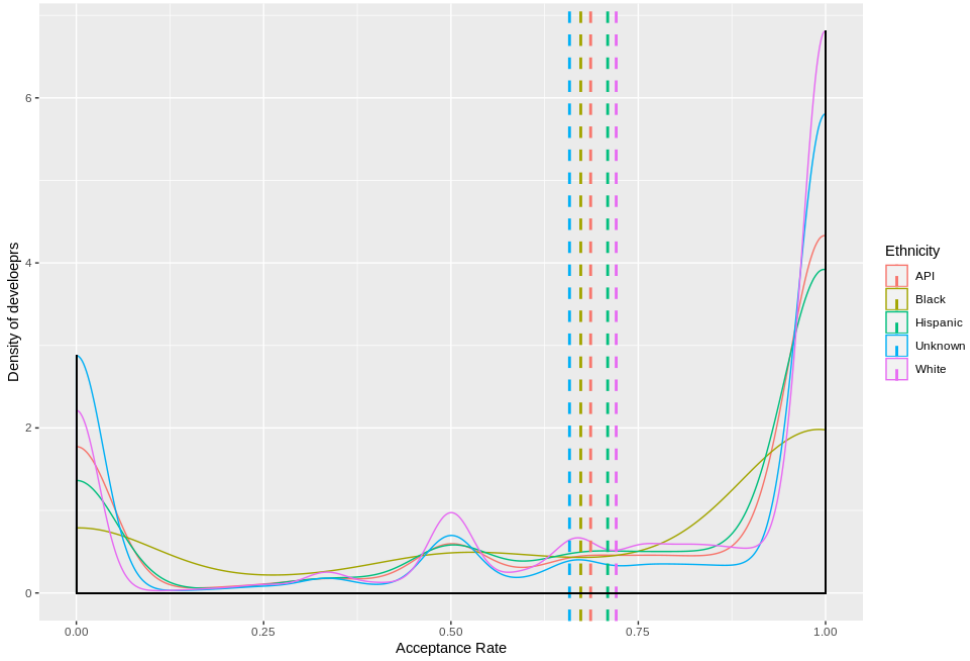


Figure 4.4: Pull Request contribution per Ethnic group

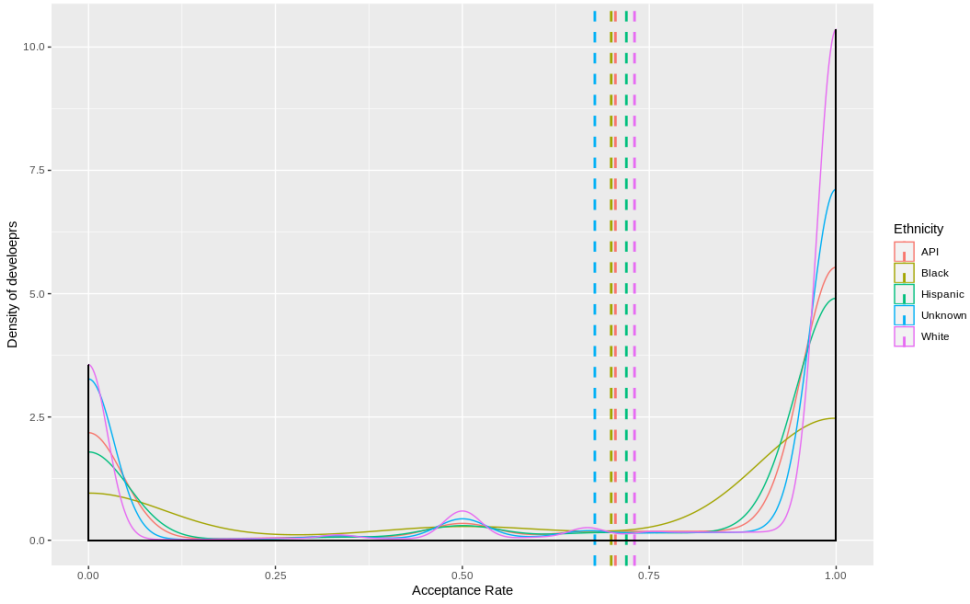
However, we cannot use the results from Figure 4.4 to make sound conclusions because there are developers that can make a lot of pull requests and can have high acceptance rates. Thus, we removed the bias that these developers are introducing by calculating the acceptance rate for each developer in each perceptible ethnicity, individually. Figure 4.5a shows the density of the number of developers in accordance with the acceptance rate for each perceptible ethnic group. The dashed vertical lines show the average acceptance rate. It is observable that developers with Unknown ethnicity have the least acceptance rate average (0.658). Developers perceptible as White have the highest acceptance rate average (0.720), followed by Hispanic (0.709), API (0.686) and Black (0.673).

We also removed any bias that might be introduced by developers in a project. For example, there might be cases of developers that are highly known and popular in a project, which might result in a high acceptance rate, but they are not successful in other projects. Therefore, we analyzed the acceptance rate of each developer in each project, developer-project pairs. Figure 4.5b shows that developers perceptible as White have the highest average (0.73), followed by Hispanic (0.719), API (0.704), and Black (0.699). Again, developers without a perceptible ethnicity have the lowest average. Although this result is consistent with the results shown in Figure 4.5a, it can be observed that the average for each ethnicity has slightly increased.

P-values added: Furthermore, we run Kruskal-Wallis and Dunn tests to analyze whether



(a) Acceptance rate for each user density plot

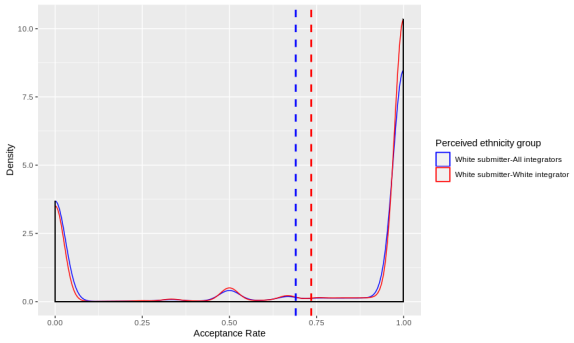


(b) Acceptance rate for each user-project pair density plot

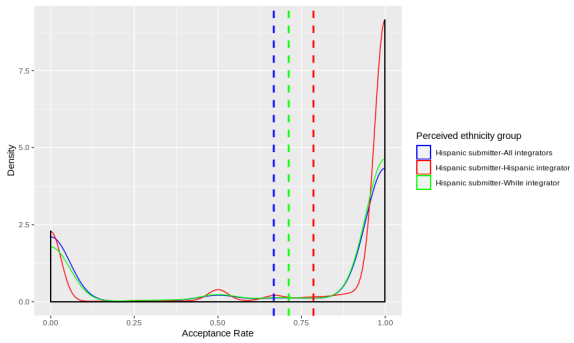
Figure 4.5: Acceptance density plot. Vertical dashed lines show the averages for each group

the differences shown in Figure 4.5 are statistically significant. Kruskal-Wallis test results for the first dataset (acceptance rate per developer) ($H = 451.14, df = 4, P < 2.2e-16$) show that the means of the perceptible ethnic groups are not equal. Pairwise comparisons using Dunn's test indicated that the mean for perceptible API developers is statistically different than perceptible White developers ($P < 0.05$), perceptible Hispanic developers ($P < 0.05$), and perceptible Black developers ($P < 0.05$). The results for the second dataset (acceptance rate per developer-project pairs) are similar. In this case, we found that the mean for perceptible Hispanic developers is statistically different than perceptible White developers ($P < 0.05$). Thus, our pairwise results indicate that all perceptible ethnic groups are statistically significant different, except the perceptible Black group. Based on the results, the difference between averages is not the outcome of chance and developers perceptible as White have the highest average followed by Hispanic and API. When comparing Black to other groups we cannot draw any conclusion because of the lack of data.

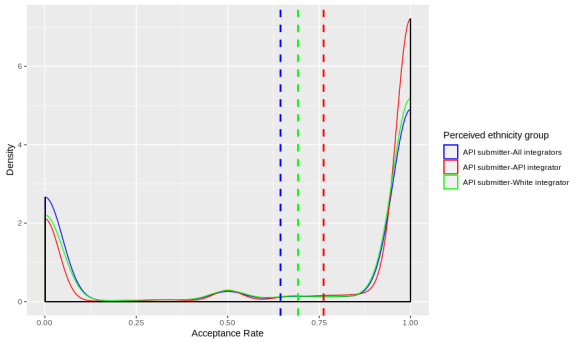
Finally, to further investigate the acceptance rate of different groups, we looked at submitter-integrator pairs. We wanted to find out what is the difference between distributions when taking the perceptible ethnicity of the integrator into account. We analyzed the acceptance rate of each developer against each integrator. Based on Figure 4.6a, submitters perceptible as White have an acceptance rate average of (0.734) when the integrator is also perceptible as White (but their acceptance rate average is (0.69) when measured against all integrators). The acceptance rate average of submitter-integrator pairs perceptible as API and Hispanic (when the submitter and integrator are perceptible to be in the same group) is 0.785 and 0.762, respectively. In addition, we found that this average is higher than the cases where the integrator is perceptible as White. This average is 0.692 for API-White pairs and 0.711 for Hispanic-White pairs. All the comparisons are statistically significant, according to Kruskal-Wallis and Dunn test. Note that we found no statistically significant difference when making the comparisons for submitters with perceptible ethnicity of Black because of the lack of enough data (See 4.1). Thus, we found that in all cases (except Black), acceptance rate average is higher when the submitter and the integrator are perceptible to be in the same group.



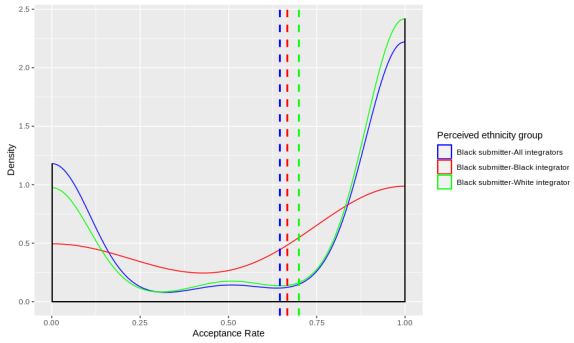
(a) Acceptance rate density plot for perceptible White developers



(b) Acceptance rate density plot for perceptible Hispanic developers



(c) Acceptance rate density plot for perceptible API developers



(d) Acceptance rate density plot for perceptible Black developers (No statistically significant difference was observed)

Figure 4.6: Acceptance density plot. Vertical dashed lines show the averages for each group

4.3 RQ3: To what extent does the developer's perceptible ethnicity affect the acceptance probability of a pull request?

Motivation: One of our goals is to determine how developer's perceptible ethnicity affects acceptance probability. When an integrator can identify the perceptible ethnicity of the submitter through their GitHub name, the integrator may make judgments based on intuitions or gut-feelings. Ethically, the OSS community, as a meritocracy, should avoid any conscious or unconscious bias against any perceptible ethnicity due to integrators' internal stereotypes. Answering this question is essential to find any empirical evidence that can help the OSS community to understand other non-technical factors that might influence the acceptance probability of a pull request.

Approach: To analyze the influence of the developer's perceptible ethnicity on pull request acceptance decisions, we measured various features (see 3.4) that have been previously studied and identified as possibly influencing pull request acceptance [24, 12, 9]. To these features, we added the perceptible ethnicity of the submitter and whether it is the same perceptible ethnicity as the integrator.

To better understand the effect of each of these features (independent variables) on the pull request acceptance (dependent variable), we combined the data and built a mixed-effect regression model using lme4 library in R [35]. To build the mixed-effects model we used the generalized linear mixed-effects model [36] function (glmer) available in the R package lme [1]. Despite the similar previous works [12, 37], we selected mixed-effect models instead of logistic regression models because they can capture measurements from within the same group (i.e., within the same project) as a random effect [36]. We used the identification of submitters and projects as random effects. All other variables were modeled as fixed effects.

Although Rastogi *et al.* [12] found that the country of submitters also influences the pull request acceptance, we removed this feature from our models as we identified 197 different countries in our dataset, some of them with only one or two data points (making them "Rare Events"). This unbalanced and disperse feature might lead to an unstable regression model, insignificant coefficients, and skewed predicted probabilities [38]. We also removed AIAN perceptible ethnicity from dataset because there was only one data point.

As well as comparing different non-white ethnic groups with each other, it is interesting to compare submitters with imperceptible ethnicity (Unknown, based on their names on

[1] https://cran.r-project.org/web/packages/lme4/lme4.pdf

Table 4.2: Result of the Models. Signif. codes: 0: ***; 0.001: **; 0.05:*

-	Model coefficients	Model (including Unknowns) coefficients
Black	-1.815e-01*	-1.452e-01
API	-1.687e-01***	-1.372e-01***
Hispanic	-1.382e-01***	-1.266e-01***
Unknown	-	-2.062e-01***
Same Ethnicity	-1.171e-02	-5.223e-02***

GitHub and the tools we used) to submitters with perceptible non-white ethnicity. Therefore, we applied our model to a dataset in which we included the Unknown category as a value for perceptible ethnicity variable. The first dataset has $1,774,421$ pull requests whereas the second dataset has $2,507,591$ pull request.

Before building any of our models ($1 model \times 2$ *datasets*), we computed the correlation between independent variables and removed highly correlated variables from the models. For numerical variables, we used Spearman's correlation test with 0.7 as threshold [37]. For categorical and binary variables, we first applied the Chi-Square test to find the correlation significance, and then we applied Cramer's V test [39] to find the strength of association.

Even though we have applied methods to remove any possible correlation among the variables, we analyzed Variance Inflation Factors for one more level of confidence. *VIF* is a statistical measure to detect multicollinearity among independent variables in a regression model. *VIF* is calculated for each variable, and it is ranged from 1 upwards, lower value means lower multicollinearity. We found that all independent variables in all of our models have a value of less than two, which indicates that multicollinearity does not impact our models negatively [40].

Results: We found that in both datasets, the Repository's popularity was highly correlated with team size. Moreover, the number of changed files in a pull request was highly correlated with the number of changed lines. Therefore, we kept the Repository's popularity and the number of changed files in our study and removed the other two respective correlated variables. Either of these correlated features could have been removed, and there is no advantage of one over the other.

Table 4.2 presents a summary of the results. The results show that being perceptible as Black, API, or Hispanic negatively affects pull request acceptance compared to when the developer is perceptible as White. This disadvantage for these perceptible ethnicity groups,

Table 4.3: Analysis of Variance

	Model effect sizes				Model (including Unknowns) effect sizes			
Independent Variable	Df	Sum Sq	Mean Sq	F value	Df	Sum Sq	Mean Sq	F value
Repository's maturity	1	1574.2	1574.2	1574.1835	1	1461.1	1461.1	1461.120
Repository's popularity	1	228.5	228.5	228.5387	1	398.4	398.4	398.412
External Contribution	1	45.5	45.5	45.5235	1	80.3	80.3	80.320
Submitter's past success	1	8720.7	8720.7	8720.7119	1	16275.8	16275.8	16275.803
Pull Request's changed files	1	691.5	691.5	691.5052	1	1574.5	1574.5	1574.472
Submitter's role	1	451.3	451.3	451.2927	1	598.3	598.3	598.333
Submitter's popularity	1	812.9	812.9	812.8870	1	2575.7	2575.7	2575.726
Submitter-Repository association	1	727.6	727.6	727.6404	1	1033.7	1033.7	1033.665
Submitter-Integrator association	1	608.4	608.4	608.3520	1	828.6	828.6	828.617
Submitter's tenure	1	182.5	182.5	182.5341	1	912.1	912.1	912.082
Pull Request's comments	1	234.3	234.3	234.3173	1	375.6	375.6	375.619
Pull Request's number of commits	1	5367.4	5367.4	5367.4238	1	10014.8	10014.8	10014.796
Submitter's perceptible ethnicity	3	202.9	67.6	67.6210	4	604.0	151.0	151.005
Submitter's experience	1	1864.3	1864.3	1864.2993	1	2958.1	2958.1	2958.085
Submitter-Repository Experience	1	5261.4	5261.4	5261.4355	1	7258.0	7258.0	7258.001
Same perceptible ethnicity	1	0.1	0.1	0.0505	1	28.2	28.2	28.215
Intra Branch	1	348.8	348.8	348.8330	1	179.5	179.5	179.534

although not strong, is statistically significant. We also can observe that imperceptible ethnicity (Unknown) negatively affects the acceptance of pull requests, and this negative effect is higher than any other perceptible ethnicity group. In other words, having a perceptible ethnicity is better than an imperceptible ethnicity, when it comes to pull request acceptance probability, and this is true for all non-white perceptible ethnic groups.

Moreover, the results are consistent for both datasets, except for the perceptible Black group (when Unknowns are included, the result for the perceptible Black group is not significant), suggesting that including unknown developers does not affect the outcome. The lack of significant results for the perceptible Black group suggests that we need more data.

We also looked at the effect sizes of all the features in models using ANOVA statistical test [41]. Effect size is a quantitative measure of the magnitude of a phenomenon [42]. Looking at the results in table 4.3, we can see that both the same ethnicity and the perceptible ethnicity variables can explain some of the effects on pull request acceptance. However, the same ethnicity effect is relatively small in both datasets, compared to other variables, but the effect size of perceptible ethnicity is high. The low effect size for the same ethnicity variable explains the lack of significant results in table 4.2.

Chapter 5

Discussion

(RQ1) While the majority of GitHub developers are perceptible as White; other perceptible ethnicities might be underrepresented in GitHub. Among 493,170 developers, we identified that almost half (47.8%) were perceptible as White. This result is consistent with Rastogi *et al.*'s work [12]. Their findings indicate that among the top seventeen countries, twelve are in Europe or North America; therefore, developers might be dominantly White. This result is also in line with the 54% of White programmers working in the US who were reported in *The Bureau of Labor Statistics*.[1]

However, the number of developers perceptible as Non-White in our study is worrisome. 6.8% of the developers were perceptible as API, 2.5% of the developers were perceptible as Hispanic, and only 0.1% of the developers were perceptible as Black. These results are not consistent with the *The Bureau of Labor Statistics*, which shows that API, Hispanic, and Black programmers account for 37.7%, 5.1%, and 5.8% of the programmers in the US respectively.

In contrast to our small percentages for Non-Whites developers in GitHub, we found that a large proportion of the developers' ethnicity (42.8%) was perceptible as Unknown. This is an interesting result because it may indicate that GitHub developers can be worried about their perceptible identity or their ethnicity. We hypothesize that a high proportion of users in the Unknown category do not provide accurate and correct information because they prefer to save their privacy rather than being perceptible as a specific ethnicity.

(RQ2) While developers perceptible as White have a higher acceptance rate average, developers with Non-White perceptible ethnicity have lower acceptance rate averages.

[1] https://www.bls.gov/cps/cpsaat11.htm

Table 5.1: Reasons why a pull request is rejected

Reason	**Explanation**
Stale	The PR was closed because it did not have activity for a long time.
No comment (Or no reason provided)	The PR doesn't have any comment from the maintainers about why they closed the PR.
Chaotic PR	The PR was closed because it was chaotic because the requester was not familiarized with Github and she/he opened/closed several PRs.
Quality	The PR was closed because it did not meet the quality required.
Duplicate	The PR was closed because it was a duplicate.
No longer needed	The PR was closed because is not longer need.
Agreement	The PR was closed because the requester did not sign the Typesafe Contributors License Agreement.
Unnecessary	The PR is considered to be unnecessary for the maintainers.
Build failed/Integration Failed/Test failed	The PR has merge conflicts because the build was not passing.
Not PR	The PR is not describing any PR but a checklist or other issues.
Not fix the problem	The pull request did not fixed the problem described.
Irrelevant PR	The PR was closed because it was irrelevant for that branch. It should be moved to another branch.

We found a positive and negative influence on the pull request acceptance depending on developers' perceptible ethnicity. These results are according to previous studies, which found that non-technical factors such as gender [5], personality traits [10], or the number of repository's stars [9] also influence the pull request acceptance.

The influence of non-technical factors in the pull request acceptance may have unwanted consequences. Some underrepresented communities such as API, Hispanic, or Black might stop contributing OSS, and they could find difficulties in the high-tech job market. According to the Open Source Survey in 2017[2], half of the respondents stated that their OSS contributions were a crucial factor for launching their professional careers. Therefore, we should avoid any possible discrimination against developers' perceptible ethnicity.

Furthermore, the Open Source Survey in 2017 reported that around 50% of Github's respondents had witnessed bad behavior in Open Source. They found that about 11% of total respondents and 3% of experienced respondents have witnessed stereotyping as a negative behavior.

Hence, we qualitatively analyzed whether there was any evidence of potential bias based on any perceptible ethnicity in the pull request acceptance process of our dataset. For that, we randomly selected 50 rejected pull requests from each of the submitter-integrator perceptible ethnicity pairs, e.g, all combinations between submitter perceptible as Black/Hispanic/API/White and integrator perceptible as Black/Hispanic/API/White. We removed the pull requests where the integrator and the submitter were the same person.

[2] https://opensourcesurvey.org/2017/

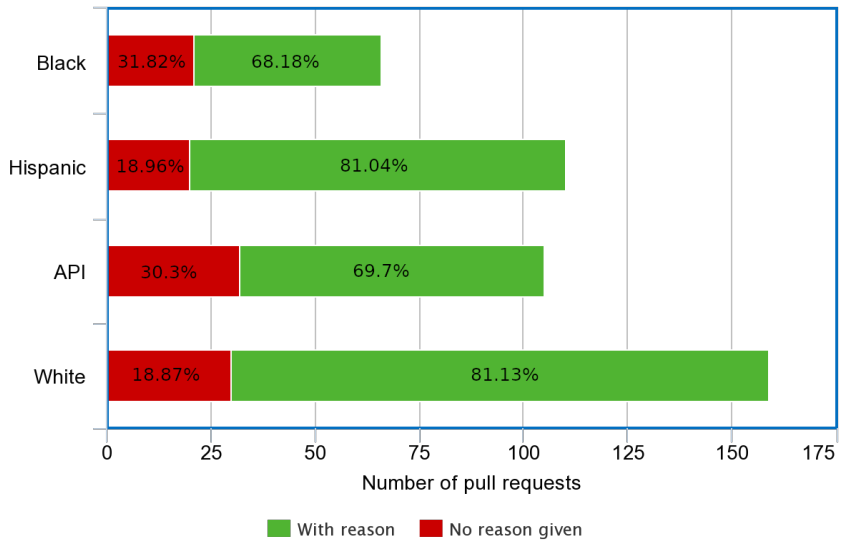


Figure 5.1: Number of rejected pull requests with a reason and without any reason for submitters perceptible as Black, Hispanic, API, and White.

Since some pairs have less than 50 pull requests, this process resulted in 463 pull requests in total. To identify any evidence of potential bias based on any perceptible ethnicity, we analyzed the comments made on these pull requests. Our results did not show any evidence of conscious bias in the comments; however, we found that 110 pull requests ($\approx$ 23.75%) did not have any comment by the integrators or any specific reason for being rejected. Figure 5.1 shows the number of pull requests with and without reason of rejection per each perceptible ethnicity group. An interesting result is that 31.82% of the pull requests submitted by developers perceptible as Black were rejected without providing any reason. This percentage is higher than that of developers perceptible as API (30.3%), Hispanic (18.96%), and White (%18.87). We believe that these results may show negative behavior, especially for projects with thousands of developers because contributors do not have any constructive feedback or a good reason of why their pull requests are being rejected.

(RQ3) Being perceptible as White has a positive influence on pull request acceptance probability, but being perceptible as API, Hispanic, or Black has a negative effect.

To further understand the effects of perceptible ethnicity in the pull request acceptance and explain what the effect sizes described in table 4.2 means, we carried out two more experiments: (1) we fed the models with manually generated test data points (using the

first dataset) to predict the pull request acceptance probability; and (2) we compared White and Non-White ethnic groups in terms of their expertise.

To predict the pull request acceptance probability, we used median and mode of variables in the accepted pull requests subset to generate the test data points. We predicted the pull request acceptance probability using the glmer functionality in R. Table 5.2 shows that when the test data point perceptible ethnicity is set to White, acceptance probability is always slightly higher comparing to other perceptible ethnicities.

To further understand the effects of perceived ethnicity in the pull request acceptance, we compared White and Non-White ethnic groups in terms of their expertise. We chose four features which are directly related to the developer's expertise: Submitter's experience, Submitter's past success, Submitter's popularity, and Submitter's tenure. We found that in all four cases, White developers have statistically significant higher or equal average/median than that of Non-White developers. These findings show that the difference between White and Non-White groups are not merely in the outcome (pull request acceptance) but also the variables leading to the outcome.

It is important to mention that these results may contradict the intention of OSS communities to behave as a meritocracy because integrators may consider developers' experience as an important factor to accept pull requests. We believe that this is a wrong behaviour that may promote a way to unconsciously bias towards white developers since White developers are more experienced, instead of choosing the quality of the contribution as the sole factor influencing acceptance, because they might have joined GitHub sooner than Non-White developers. The annual Octoverse report[3] states that just in 2019 the development of source code was more global as the number of non-white communities grow across Asia and Africa.

(RQ3) Same perceptible ethnicity of the submitter and the integrator has a negative effect on pull request acceptance.

[3] https://octoverse.github.com

Table 5.2: Acceptance Probability (%)

	Including Unknowns	Excluding Unknowns
White	78.69	80.56
Black	77.68	79.58
API	78.14	80.29
Hispanic	78.32	80.40
Unknown	77.40	-

Surprisingly, we found that the same ethnicity variable has a negative effect on pull request acceptance, which is not consistent with the results described in Fig. 4.6. However, we found that this result is not significant when unknowns are excluded and therefore not conclusive. We believe this negative effect is the result of the very high number of white developers. We found that among those pull requests which their submitter and integrator had the same perceptible ethnicity and were not merged, 98% had integrators with White perceptible ethnicity, this explains 43% of the total number of pull requests that are not merged.

(RQ3) Replicating previous studies and adding ethnicity features.

Previous studies [12, 24] suggest features that we could not extract as explained in chapter 3.4. Gousios et al. also suggest four heuristics to detect merged pull requests, an approach we could not utilize because of computational limitations (we have addressed this limitation in chapter 6). However, we tried to replicate previous studies, with their exact dataset, and add ethnicity related features.

After applying our suggested filters to dataset created by [24] and adding ethnicity features to the dataset, there were only 170,544 left. We applied the models suggest in 4 and found that because this dataset is too small, no significant result for perceptible Black and perceptible Hispanic is extractable (Table 5.3) However, for same ethnicity and perceptible API variables, the results were consistent but not as strong as table 4.2.

Table 5.3: Replication study results. Signif. codes: 0: ***; 0.001: **; 0.05:*

-	Excluding Unknowns	Including Unknowns
Black	-4.997e-01	-4.982e-01
API	-1.900e-01***	-1.913e-01***
Hispanic	4.682e-02	4.728e-02
Unknown	-	-1.707e-01***
Same	-1.302e-01***	-1.345e-01***

Chapter 6

Threats to Validity

We present our validity threats in terms of the four main threats in empirical software engineering research [43].

6.1 Construct Validity

Although some previous studies [24, 12] identified pull request status using a set of heuristics, we extracted this data using GitHub's API directly. In this approach, there is no difference between pull requests that are cherry-picked and the ones which are rejected. This criterion may introduce false positives in our dataset. However, we analyzed more than four million pull requests, and typically, the number of cherry-picked pull requests are few in numbers. We also replicated our study to previous ready made datsets and found that their dataset is too small to extract any significant result, however for those features that we could find significant results, there were completely consistent with our results.

6.2 Internal Validity

Identifying perceptible ethnicities using names is ongoing research, which can be further explored. Name-Prism [16] may identify misassigned ethnicities, but it has been evaluated in previous studies in [44], and it presents an F1 score of 0.795. Furthermore, Name-Prism [16] uses US-based ethnic categorization which may be a threat because predominant ethnicities may vary depending on country. However, this tool is trained on a 74M labeled

name set from 118 countries around the world, therefore, this categorization represents the biggest ethnicities in the world.

Providing a name is not mandatory on GitHub. Thus, users can fake their names, which may affect our results. However, our thesis studies whether a perceptible ethnicity affects (consciously or unconsciously) the evaluation process of the pull request. We only analyze what OSS developers perceive as the ethnicity of another developer, from their name, in the absence of any indicator on GitHub. This thesis does not analyze whether other's perception of one's ethnicity is more important than their actual ethnicity. Therefore, in our study it is not essential whether the developers are not using their real name. If they use a name associated with a different perceptible ethnicity other than their own, then any other developer would perceive the ethnicity derived from their chosen name, much like the tool we use. In addition, we used Stanford NERTagger to distinguish between human names and other groups of words.

Another internal threat to the validity is the lack of social factors (outside software engineering context) influencing the quality of contribution and expertise of developers, in our study. One of the main social factors influencing expertise is education. In an ideal situation, the education of each developer could be added as an independent variable.

6.3 External Validity

Even though our dataset is bigger than previous studies [9, 12, 24], it is not a representative of the whole community. Many GitHub users have unknown accounts, which makes it difficult to draw any conclusion. However, considering that other's perception of one's ethnicity is essential in our study, users with unknown accounts help us to investigate whether perceptible ethnicity from GitHub names affects acceptance probability.

6.4 Conclusion Validity

Although we captured most of the independent variables in the literature, there may be other variables that we have missed. We believe that research should actively look for more features affecting pull request acceptance. Our findings suggest that perceptible ethnicity can affect pull request acceptance probability. However, we cannot claim that this is due to the existence of any racial discrimination. Furthermore, our approach uses only five ethnic groups, which might not be a good representative of all ethnicities. Nonetheless, these groups are considered to include the majority of the population of the world.

Chapter 7

Conclusion and Future Work

Usually, OSS projects are the result of many collaborations from diverse developers with different backgrounds. The difference can either be in cultural background, ethnicity, age, and other social factors. Although the acceptance of such contributions should be based on the quality of the source code being contributed [8], recent studies have shown that diversity issues affect the acceptance or rejection of these contributions [9, 10, 5, 11, 12]. Therefore, this thesis assists with the first empirical study that analyzes how perceptible ethnicity relates to the evaluation process of the contributions in GitHub.

We analyzed more than four million pull requests from 493,170 developers in GitHub. We first identified developers' perceptible ethnicity based on their GitHub names using the Name-Prism tool [37]. We then linked the developers' perceptible ethnicity with their pull requests, and we finally built regression models to study the effect of developers' perceptible ethnicity on pull-request acceptance probability.

Our findings indicate an alarmingly small number of developers with perceptible ethnicity as Non-White, 6.8% of the developers were perceptible as API, 2.5% of the developers were perceptible as Hispanic, and 0.1% of the developers were perceptible as Black. Non-White perceptible ethnicities have a negative effect on pull request acceptance, but this effect is positive for developers perceptible as White. Furthermore, we found that a high proportion of pull requests are rejected without providing any specific reason (23.75%). In conclusion, we find that there may exist an unconscious bias against developers perceptible as Non-White. These results may indicate that Non-White developers need to be trained and included more in OSS communities.

Although our quantitative results are a first step to be aware of the perceptible ethnicity problem in OSS, further research should be done. For example, in our future work we hope

to complement this study with a thorough qualitative survey to support our quantitative results. In addition, another line of research is developing new tools that avoid possible bias against some perceptible ethnicities, and tools that allow developers speak out against wrong behavior when they see it. These tools can help OSS projects by fostering a healthier OSS community.

7.0.1 Replication package

Supplementary material associated with this article as well as the replication package can be found in https://bit.ly/2IFX2xv.

References

[1] Derek R Avery et al. "Is there method to the madness? Examining how racioethnic matching influences retail store productivity". In: *Personnel Psychology* 65.1 (2012), pp. 167–199.

[2] Adam D Galinsky et al. "Maximizing the gains and minimizing the pains of diversity: A policy perspective". In: *Perspectives on Psychological Science* 10.6 (2015), pp. 742–748.

[3] Alison Reynolds and David Lewis. "Teams solve problems faster when they're more cognitively diverse". In: *Harvard Business Review* 23 (2017), p. 2019.

[4] Christopher P Earley and Elaine Mosakowski. "Creating hybrid team cultures: An empirical test of transnational team functioning". In: *Academy of Management journal* 43.1 (2000), pp. 26–49.

[5] Bogdan Vasilescu et al. "Gender and tenure diversity in GitHub teams". In: *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. New York, NY: ACM, 2015, pp. 3789–3798.

[6] Vreda Pieterse, Derrick G Kourie, and Inge P Sonnekus. "Software engineering team diversity and performance". In: *Proceedings of the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*. Grahamstown, South Africa: South African Institute for Computer Scientists and Information Technologists, 2006, pp. 180–186.

[7] Gemma Catolino et al. "Gender diversity and women in software teams: How do they affect community smells?" In: *Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Society*. New York, NY: IEEE, 2019, pp. 11–20.

[8] Walt Scacchi. "Free/open source software development: Recent research results and methods". In: *Advances in Computers* 69 (2007), pp. 243–295.

[9] Jason Tsay, Laura Dabbish, and James Herbsleb. "Influence of social and technical factors for evaluating contribution in GitHub". In: *Proceedings of the 36th international conference on Software engineering*. New York, NY: ACM, 2014, pp. 356–366.

[10] Rahul N Iyer et al. "Effects of Personality Traits on Pull Request Acceptance". In: *IEEE Transactions on Software Engineering* (2019).

[11] Josh Terrell et al. "Gender differences and bias in open source: Pull request acceptance of women versus men". In: *PeerJ Computer Science* 3 (2017), e111.

[12] Ayushi Rastogi et al. "Relationship between geographical location and evaluation of developer contributions in github". In: *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. New York, NY: ACM, 2018, p. 22.

[13] Donn Erwin Byrne. *The attraction paradigm*. Vol. 11. Cambridge, MA: Academic Pr, 1971.

[14] Henri Tajfel. "Social psychology of intergroup relations". In: *Annual review of psychology* 33.1 (1982), pp. 1–39.

[15] Bogdan Vasilescu, Vladimir Filkov, and Alexander Serebrenik. "Perceptions of diversity on git hub: A user survey". In: *2015 IEEE/ACM 8th International Workshop on Cooperative and Human Aspects of Software Engineering*. IEEE. 2015, pp. 50–56.

[16] Junting Ye et al. "Nationality classification using name embeddings". In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. New York, NY: ACM, 2017, pp. 1897–1906.

[17] Oxford Dictionary. *bias*. https://www.oxfordlearnersdictionaries.com/definition/english/bias. Accessed: 2020-01-03. 2020.

[18] Megumi Hosoda, Lam T Nguyen, and Eugene F Stone-Romero. "The effect of Hispanic accents on employment decisions". In: *Journal of Managerial Psychology* 27.4 (2012), pp. 347–364.

[19] Jonathan St BT Evans. "In two minds: dual-process accounts of reasoning". In: *Trends in cognitive sciences* 7.10 (2003), pp. 454–459.

[20] Sandro Heiniger and Hugues Mercier. *National Bias of International Gymnastics Judges during the 2013-2016 Olympic Cycle*. 2018. arXiv: 1807.10033 [stat.AP].

[21] Gianmarco Paris et al. "Region-based citation bias in science". In: *Nature* 396.6708 (1998), p. 210.

[22] Marianne Bertrand and Sendhil Mullainathan. "Are Emily and Greg more employable than Lakisha and Jamal? A field experiment on labor market discrimination". In: *American economic review* 94.4 (2004), pp. 991–1013.

[23] Fabio Calefato, Filippo Lanubile, and Bogdan Vasilescu. "A large-scale, in-depth analysis of developers' personalities in the Apache ecosystem". In: *Information and Software Technology* 114 (2019), pp. 1–20.

[24] Georgios Gousios and Andy Zaidman. "A dataset for pull-based development research". In: *Proceedings of the 11th Working Conference on Mining Software Repositories*. New York, NY: ACM, 2014, pp. 368–371.

[25] Georgios Gousios. "The GHTorent dataset and tool suite". In: *2013 10th Working Conference on Mining Software Repositories (MSR)*. New York, NY: IEEE, 2013, pp. 233–236.

[26] Nuthan Munaiah et al. "Curating GitHub for engineered software projects". In: *Empirical Software Engineering* 22.6 (2017), pp. 3219–3253.

[27] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. "Incorporating non-local information into information extraction systems by gibbs sampling". In: *Proceedings of the 43rd annual meeting on association for computational linguistics*. Stroudsburg, PA: Association for Computational Linguistics, 2005, pp. 363–370.

[28] Vetle Ingvald Torvik and Sneha Agarwal. *Ethnea – an instance-based ethnicity classifier based on geo-coded author names in a large-scale bibliographic database*. Washington DC: ., 2016.

[29] Gareth James et al. In: *An introduction to statistical learning*. Vol. 112. Springer, 2013. Chap. 3.

[30] Michael Bacharach. "Estimating Nonnegative Matrices from Marginal Data". In: *International Economic Review* 6.3 (1965), pp. 294–310. ISSN: 00206598, 14682354. URL: http://www.jstor.org/stable/2525582.

[31] Yadolah Dodge. "Kruskal-Wallis Test". In: *The Concise Encyclopedia of Statistics*. New York, NY: Springer New York, 2008, pp. 288–290. ISBN: 978-0-387-32833-1. DOI: 10.1007/978-0-387-32833-1_216. URL: https://doi.org/10.1007/978-0-387-32833-1%7B%5C_%7D216.

[32] Xin Xia et al. "Personality and project success: Insights from a large-scale study with professionals". In: *2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. New York, NY: IEEE, 2017, pp. 318–328.

[33] Olive Jean Dunn. "Multiple comparisons using rank sums". In: *Technometrics* 6.3 (1964), pp. 241–252.

[34] Olive Jean Dunn. "Estimation of the medians for dependent variables". In: *The Annals of Mathematical Statistics* 30.1 (1959), pp. 192–197.

[35] Douglas Bates et al. "Fitting Linear Mixed-Effects Models Using lme4". In: *Journal of Statistical Software* 67.1 (2015), pp. 1–48. DOI: 10.18637/jss.v067.i01.

[36] Douglas Bates et al. *lme4: Linear mixed-effects models using Eigen and S4. R package version 1.1-7*. 2014.

[37] Patanamon Thongtanunam et al. "Review participation in modern code review". In: *Empirical Software Engineering* 22.2 (2017), pp. 768–817.

[38] Gary King and Langche Zeng. "Logistic Regression in Rare Events Data". In: *Political Analysis* 9 (Spring 2001), pp. 137–163.

[39] Michael Kearney. "Cramér's V". In: vol. 1. Thousand Oaks, CA: SAGE Publications, Dec. 2017, pp. 290–290. DOI: 10.4135/9781483381411.n107.

[40] Patricia Cohen, Stephen G West, and Leona S Aiken. *Applied multiple regression/correlation analysis for the behavioral sciences*. Psychology Press, 2014.

[41] Antonio Cuevas, Manuel Febrero, and Ricardo Fraiman. "An anova test for functional data". In: *Computational statistics & data analysis* 47.1 (2004), pp. 111–122.

[42] Ken Kelley and Kristopher J Preacher. "On effect size." In: *Psychological methods* 17 2 (2012), pp. 137–52.

[43] Claes Wohlin et al. *Experimentation in software engineering*. Berlin: Springer Science & Business Media, 2012.

[44] Bedoor K AlShebli, Talal Rahwan, and Wei Lee Woon. "The preeminence of ethnic diversity in scientific collaboration". In: *Nature communications* 9.1 (2018), pp. 1–10.