Code reviews in open source projects: how do gender biases affect participation and outcomes?

Sayma Sultana¹ · Asif Kamal Turzo¹ · Amiangshu Bosu¹

Accepted: 17 March 2023 / Published online: 5 June 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

Context Contemporary software development organizations lack diversity, and the ratios of women in Free and Open-Source Software (FOSS) communities are even lower than the industry average. Although the results of recent studies hint at the existence of biases against women, it is unclear to what extent such biases influence the outcomes of various software development tasks.

Objective This study conceptually replicates two recent studies by Terrell et al. and Bosu and Sultana that investigated gender biases in FOSS communities. We aim to identify whether the outcomes of or participation in code reviews (or pull requests) are influenced by the gender of a developer. In particular, we focus on two outcome aspects (i.e., code acceptance, and review interval) and one participation aspect (i.e., code review participation) of code review processes.

Method With this goal, this study includes a total of 1010 FOSS projects. Ten out of those projects use Gerrit-based code reviews. The remaining 1000 are randomly selected from the GHTorrent dataset based on a stratified sampling of projects fitting certain criteria. We divided GitHub projects into four groups based on the number of distinct contributors. We developed six regression models for each of the 14 datasets (i.e., 10 Gerrit based and 4 GitHub-based) to identify if code acceptance, review intervals, and code review participation differ based on the gender and gender-neutral profile of a developer.

Results Our results find significant gender biases during code acceptance among 13 out of the 14 datasets, with seven favoring men and the remaining six favoring women. We found significant differences between men and women in terms of code review intervals, with women encountering longer delays than men in three cases and the opposite in seven.

Communicated by: Christoph Treude, Maria Teresa Baldassarre

This article belongs to the Topical Collection: Special Issue on Registered Reports.
Our results indicate reviewer selection as one of the most gender-biased aspects, with 12 out of 14 datasets exhibiting bias. A total of 11 out of the 14 cases show women having significantly lower code review participation than their men colleagues. Since most of the review assignments are based on invitations, this result suggests possible affinity biases among the developers. We also noticed a significantly higher likelihood of women using gender-neutral profiles. Supporting Terrell et al.’s claim, women with gender-neutral profiles had higher odds of code acceptance than men among three Gerrit-based projects. However, contradicting their results, we found significantly lower odds of code acceptance for women with gender-neutral profiles across all four GitHub project groups.

Conclusions Though gender bias exists among many projects, the direction and amplitude of that bias vary based on project size, community, and culture. Similar bias mitigation strategies may not work across all communities, as the characteristics of biases and their underlying causes differ. As women are less likely to be invited for reviews, FOSS projects should take initiatives to ensure the equitable selection of women as reviewers.

Keywords Code review · Diversity and inclusion · Pull requests · Gender bias

1 Introduction

According to the US labor market census in 2019, women make up approximately 18.8% of technical roles in the software industry and 18% of computer science graduates (Bertagnoli 2021). However, the number of females contributing to Free and Open Source Software (FOSS) projects is even lower (i.e., less than 10%) (Terrell et al. 2017; Vasilescu et al. 2015); which indicates a significant gender imbalance within the FOSS communities. Due to the lack of gender diversity, many women hesitate to join FOSS projects, as one quotes, “I feel like it’s a circle I can’t get into. But mostly I fear the excessive spotlight of being the sole female programmer on a publicly available project. In light of how women are treated on the internet, this fear does not seem unreasonable” Beneschott (2014). The lack of diversity among FOSS communities has been subjected to several prior studies (Ghosh 2005; Krieger et al. 2006; David and Shapiro 2008; Bosu and Sultana 2019). To increase diversity through recruitment of the underrepresented groups, FOSS communities such as Mozilla (Beard 2018), Debian (Ciceri 2021), and Fedora (Team 2019), as well as commercial software development organizations such as Google¹ and Deloitte (Bourke 2017), have adopted several initiatives.

However, these diversity initiatives may not achieve adequate results if a community does not promote ‘inclusion’ by providing equal access to opportunities and resources for people who might otherwise be excluded or marginalized (Built-in 2021). Vernā Myers, a leading diversity and inclusion strategist, says, “While diversity means being invited to the party, inclusion means to be asked for a dance” (Laura Sherbin 2017). Sense of inclusion drives people to be more productive and engaged with the workplace, as well as improves overall team performance and creativity (Laura Sherbin 2017; Built-in 2021). Organizations that practice inclusion generate up to 30% higher revenue per employee and greater profitability than their competitors (Bourke 2017).

Despite these known benefits, increasing Diversity Equity Inclusion (DEI) among FOSS communities is difficult due to various explicit (Foundation 2008) and implicit biases (Alba 2018) against minorities. For example, a 2017 study by Terrell et al. (2017) found that

¹ https://diversity.google/
women’s pull-requests are less likely to be accepted than those created by men if their gender is visibly identifiable. On the other hand, if a woman’s gender is not identifiable, her pull requests have higher odds of acceptance than those created by men. Another study by Bosu et al. (2019) investigated the state of DEI among ten popular FOSS projects. Their results hint at possible biases against woman developers in three out of the ten cases, as women had lower code acceptance rates as well as longer delays than men in getting feedback for submitted changes. Additionally, in one project, women received significantly fewer code review invitations than their men peers.

However, based on the results of these two studies (Bosu and Sultana 2019; Terrell et al. 2017), we cannot predict to what extent developers’ genders influence those outcomes, since several other confounding factors may have been influential as well. For example, the acceptance of a pull request may be influenced by its size, complexity, and whether it includes a new feature or a bug fix. Bosu et al.’s study (Bosu and Sultana 2019) did not account for any of these confounding factors. Although Terrell et al. (2017) conducted post hoc analysis to identify the influences of other factors such as experience, patch size, patch type, and status within the community, it is unclear how much variance in acceptance rates is influenced by the genders of the contributors. To answer these questions, this study aims to conduct a partial conceptual replication (Shull et al. 2008) of these two studies (i.e., Terrell et al. (2017), and Bosu et al. (2019)) by adopting measures to control for various confounding factors.

In this replication, we aim to identify whether the outcomes of or participation in code reviews (or pull requests) are influenced by the gender of a developer. However, since the identification of non-binary genders is not feasible without input from the subjects, we focus primarily on men and women. With this goal, this study includes a total of 1010 FOSS projects. Ten of those projects are selected to replicate Bosu et al.’s study. The remaining 1000 are randomly selected from GHTorrent (Gousios 2013) dataset based on a stratified sampling of projects fitting certain criteria. We divided GitHub projects into four groups based on the number of distinct contributors. Similar to Bosu et al., we initially computed genders using the GenderComputer tool (Vasilescu et al. 2014), then followed up with manual validations using multiple sources, including avatars and social media profiles.

We select two outcome aspects and one participation aspect of code reviews: (i) acceptance rate—the ratio of submitted code reviews (or pull requests) that are accepted; (ii) review interval—the time from the beginning to the end of the review process, and (iii) code review participation—the number of code reviews within a period as a reviewer. We developed six regression models for each of the 14 datasets (i.e., 10 Gerrit based and 4 GitHub-based), where one of those three aspects is the dependent variable, and in addition to the genders of the contributors, various confounding factors that have been found to be influential in patch acceptance by prior studies are independent variables. We used multivariate regression modeling techniques since those can be used to estimate the relationships between a dependent variable and one or more independent variables (aka ‘predictors’) and are commonly used to eliminate the effects of confounding variables (Pourhoseingholi et al. 2012; Harrell 2015).

We proposed this study in a registered report titled “Are Code Review Processes Influenced by the Genders of the Participants?”. This registered report included our proposed study protocol and was published in ICSME 2021 (Sultana and Bosu 2021). According to the plan, we executed our study, where we considered thirteen characteristics of contributors and pull requests to identify the impact of gender in three aspects: pull request acceptance, code review interval, and code review participation. We have tried our best to execute the study exactly according to our proposed protocol. However, during the study execution, we decided to mine the selected projects again to include the most recent code reviews/pull requests. We also altered our protocol to determine whether a profile is gender neutral since,
during execution, we identified several shortcomings in our initial choice (i.e., Terrell et al.
’s approach). We have described those deviations in a disclaimer at the end of this paper.

In summary, the primary contributions of this study are:

- Empirical evidence regarding how gender influences code review processes.
- An investigation of how gender biases manifest across various project groups.
- Recommendations for projects to increase diversity and inclusion.
- An update of the status quo of gender diversity among FOSS projects.
- To promote replication, we have made our scripts, and dataset (without personally iden-
tifiable information) publicly available on Zenodo (Sultana et al. 2023).

The remainder of the paper is organized as follows. Section 2 briefly describes our replica-
tion protocol and the two studies being replicated. Section 3 presents formulated hypotheses
based on our research questions. Section 4 details our research methodology. Sections 6 and
7 present the results and discuss the implications, respectively. Section 8 describes prior
related works. Finally, Sections 9 and 10 discuss threats to validity and conclude the paper,
respectively.

2 Replication Protocol

Replications are crucial in the Software Engineering (SE) domain to build knowledge through
a family of experiments (Santos et al. 2019). According to Shull et al. (2008), SE replications
fall into two categories:

1. Exact replication: In an exact replication, study procedures are closely followed. Exact
replications can be further divided into two subcategories: (i) dependent and (ii) indepen-
dent. In a dependent replication, all the variables and conditions are kept as close to the
original studies as possible. However, some of the aspects of the original study can be
modified in an independent replication to fit a new context.

2. Conceptual replication: This category of replications uses different research methodolo-
gies to study the same set of questions.

Our replications for this study fall into the ‘partial conceptual replication’ category since:
(i) we plan to use a different type of statistical modeling technique; (ii) we will measure
and account for additional confounding variables not considered in the original studies; (iii)
we will use updated versions of the datasets to include recent code reviews to reflect the
contemporary state of the projects; and (iv) we are interested in partial replications to retest
only the results from the original studies hinting at explicit gender biases. The following
subsections briefly describe those two studies.

2.1 "Gender Differences and Biases During Pull Request Acceptance"
by Terrell et al. (2017)

To investigate gender biases among FOSS projects, Terrell et al. investigated pull request
(PR) acceptance on GitHub. The stated goals of their study were to what extent do gender
biases exist among contributors judging PRs for acceptance.

Study Design: Terrell et al. used the GHTorrent dataset (Gousios 2013) from June 7, 2010,
to April 1, 2015. To estimate the genders of the contributors, they first used the GenderCom-
puter tool (Vasilescu et al. 2014). They augmented this resolution with a Google+ profile
search based on a contributor’s GitHub-listed email address. To identify whether a profile is gender neutral or not, they downloaded GitHub profile pictures and used ImageMagick to identify ‘identicons’. They classified a GitHub account as gender neutral if: (i) the profile uses an identicon; (ii) the gender inference tool outputs ‘unknown’ for both the user login name and full name; and (iii) their mixed-culture panel judges could not resolve the gender. They considered each pull request as a data point, with men forming the ‘control group’ and women as the ‘treatment’. In addition to gender, they investigated the following confounding factors during posthoc analyses: i) number of lines added, ii) number of lines removed, iii) number of commits, iv) number of files changed, v) pull index (i.e., the author’s nth pull request), vi) number of references to issues, vii) licenses (whether a project is open-source or not), viii) creator type (i.e., insider or outsider), and ix) file extension.

Key results: The key findings of their study are:

- **Acceptance rate:** Although they hypothesized that women have a lower acceptance rate than men overall, they found support for a contrasting hypothesis. Overall, women had 78.7% acceptance rate, while men had 74.6% ($\chi^2 = 1170$, $p < 0.001$).
- **Survivorship Bias:** They hypothesized that the acceptance rate starts low for women and that surviving women make the majority of pull requests. Their investigation of PRs over time (i.e., average acceptance on the first PR, second PR, and so on) found women having higher acceptance rates all along.
- **Focus on limited projects:** Their second hypothesis stated that women limit their contributions to fewer projects and maintain higher success rates. While women do tend to work on fewer projects, due to limited data points, their investigation was not conclusive enough to draw any clear conclusions regarding whether women working on a higher number of projects have higher or lower acceptance rates.
- **Pull requests for much-needed features or issues:** To evaluate if women’s PRs fulfill any immediate need, they examined the descriptions of PRs and the percentage of requests that refer to any issue. Results showed that women were less likely to submit PRs that satisfied immediate project needs.
- **Size of changes:** They investigated the number of women’s contributions using three metrics: lines of code changed, number of modified files, and number of commits in each pull request. They found out that on average, women make larger PRs than men.
- **Success in contributing program code:** Researchers considered Turing-complete programming languages as program code and showed that women possess a higher acceptance rate when submitting programming codes.
- **Appearance as woman:** To find out the effect of being revealed as a woman, the authors divided their analysis into two parts. For insiders, they found out that women possess almost the same acceptance rate whether their gender is identifiable or not. As outsiders, women with gender-identifiable profiles have 10.2% lower acceptance rates than women with gender-neutral profiles.

Artifacts: Among the 4.03 million GitHub users, who publicly listed their email addresses, their Google+-assisted linking strategy helped identify the genders of 1.4 million (35.3%) contributors. Since their gender-linking approach may raise privacy concerns, they did not publish their dataset publicly. However, their scripts are publicly available for replication.

---

2 5×5 pixel sprites that are generated using a hash of the user’s ID.
2.2 "Diversity and Inclusion in FOSS Projects" by Bosu and Sultana (2019)

Bosu et al. (2019) investigated the state of diversity and inclusion among ten FOSS projects based on the following four research questions:

– **RQ1**: What are the percentages of female contributors in popular FOSS projects?
– **RQ2**: What percentage of leadership roles are occupied by females among the FOSS projects?
– **RQ3**: Are female developers of FOSS projects significantly more/less productive than their male colleagues?
– **RQ4**: Is there any explicit bias for or against female developers in FOSS projects?

**Results**: Following are the results of their study that address the proposed research questions, respectively:

– **Gender diversity**: They found that in all of the projects, women comprise less than 10% of the non-casual developers.
– **Gender inclusion**: Among six out of the ten projects, the percentage of women as core developers is even lower than the percentage of women as non-casual developers. On average, 4.14% of the core developers were women, which also showed a poor representation of women in leadership roles.
– **Productivity comparison**: None of the projects showed any significant differences between men and women in terms of two productivity metrics: (i) code churn per month; (ii) code changes per month.
– **Explicit bias for/against women**: Their results suggest potential explicit biases against women in three of ten projects. In these projects, women had a lower code acceptance rate, had to wait significantly longer to get their initial feedback from the reviewers, and also had to wait a significantly longer time to get their complete review. On the contrary, in three other projects, the result shows biases favoring women. They also showed that men participate more in code reviews. As code review is done based on invitations, the authors hypothesized that women may be excluded to favor men.

**Artifacts**: For all ten projects, authors mined a total of 683,865 pull requests and identified 4,543 developers’ information in total. Among the developers, they were successful in inferring 4,496 (99%) developers’ gender. They also did not publish their data publicly due to privacy concerns. However, we do have access to their scripts and dataset due to an overlapping author.

### 3 Hypotheses

Our primary goal is to investigate whether the outcomes of or participation in code reviews are influenced by the gender of a developer. Although Terrell et al. (2017) found higher acceptance rates for women on GitHub, Bosu et al.’s (2019) results support that finding in only two out of the ten projects, while three projects in their study suggested the contrary. Therefore, our first research question is:

| RQ1 | Do the genders of the contributors influence acceptance of their code changes? |
While we acknowledge that there are several possible choices for the gender of a person, due to the lack of a reliable automated mechanism to identify genders that do not fall into binary choices (e.g., man, woman), prior Software Engineering (SE) studies (Vasilescu et al. 2015; Terrell et al. 2017; Bosu and Sultana 2019; Lin and Serebrenik 2016) confined comparisons between men and women while studying the impact of gender. Therefore, we plan to limit our investigations to identify the differences between men and women by formalizing this research question into the following two hypotheses:

**H1.10:** There are no significant differences in code acceptance rates between men and women.

**H1.1a:** There are significant differences in code acceptance rates between men and women.

Since Terrell et al. (2017) found that women have a significantly lower code acceptance rate than men when their gender is identifiable, we pose two additional hypotheses.

**H1.20:** There are no significant differences in code acceptance rates between men/women with and without gender identifiable profiles.

**H1.2a:** There are significant differences in code acceptance rates between men/women with and without gender-identifiable profiles.

‘Code review interval’ spans from the beginning to the end of the code review process (Rigby et al. 2008). Bosu et al. (2019) found that women developers had to wait three times longer than men in two out of the ten projects. However, two of their projects suggested the opposite. Our next research question aims to investigate those differences.

**RQ2:** Do the genders of the contributors influence code review intervals for their code changes?

We formalize this research question into the following two hypotheses:

**H2.10:** There are no significant differences in review intervals between men and women.

**H2.1a:** There are significant differences in code review intervals between men and women.

Since identifiable gender may influence code review intervals, we pose two additional hypotheses.

**H2.20:** There are no significant differences in code review intervals between men/women with and without gender identifiable profiles.

**H2.2a:** There are significant differences in code review intervals between men/women with and without gender identifiable profiles.

Although women did not exhibit lower productivity than men, Bosu et al. (2019) found significantly lower participation of women in code reviews in one project. They hypothesize that, since participation in a code review is based on invitations from peers, a man may prefer other men over women contributors for reviewing code changes (Bosu and Sultana 2019). Hence, our next research question is:

**RQ3:** Do the genders of the contributors influence their participation in code reviews?

We formalize this research question into the following two hypotheses:

**H3.10:** There are no significant differences in code review participation between men and women.

**H3.1a:** There are significant differences in code review participation between men and women.

---

3 https://www.healthline.com/health/different-genders
However, as identifiable gender may influence code review invitations, we pose the following two hypotheses.

**H3.2**: There are no significant differences in code review participation between men/women with and without gender identifiable profiles.

**H3.2a**: There are significant differences in code review participation between men/women with and without gender identifiable profiles.

### 4 Dataset Preparation

We conducted our study on a total of 1010 projects, where 10 projects use Gerrit (https://www.gerritcodereview.com/) and 1,000 projects are mined from GitHub based on a stratified sampling strategy. While we acknowledge the subtle differences between Gerrit-based code reviews (Bosu and Carver 2014) and GitHub’s pull request-based ones (Gousios et al. 2014), the participation and outcome aspects addressed in this study are mostly similar. For the sake of brevity, we use the term ‘CR’ to refer to both code review and pull request hereinafter. The following subsections detail our approaches to preparing the datasets from Gerrit and GitHub-based projects.

#### 4.1 Gerrit Dataset Preparation

Due to an overlapping author, we had access to the dataset of 683,865 CRs curated in Bosu et al.’s (2019) study. Projects from this dataset were selected based on two criteria: 1) Projects are actively using Gerrit. 2) Contributors to that project reviewed at least 15,000 requests. Since OmapZoom, one of the projects from their selection, is no longer active, we replaced it with WikiMedia. WikiMedia also satisfies selection criteria laid out by Bosu et al. Using the same Gerrit-Miner tool used in their study, we updated Bosu et al.’s dataset to include all CRs completed through April 30, 2022. We identified the bot accounts for exclusion using a set of keywords (e.g., ‘bot’, ‘CI’, ‘Jenkins’, ‘build’, ‘auto’, ‘devop’, ‘hook’, ‘workflow’ and ‘travis’) followed by manual validations. Similar to Bosu et al., we used the Levenshtein distance between two names to identify similar names. We performed manual reviews of the associated accounts to identify and merge the multiple accounts belonging to the same person into one account. Similar to the original study (Bosu and Sultana 2019), we limited our analysis to non-casual contributors. We also wrote another script to download and store avatar images for all the users. Table 1 lists the demographics of the projects. This updated dataset includes more than 3.3 million CRs, which is approximately five times more than the original study. We include a total of 12,252 non-casual developers, which is also 2.66 times more than Bosu et al.’s.

#### 4.2 GitHub Dataset Preparation

We used the publicly available GHTorrent MySQL dump from March 2021. We imported this data dump onto a local MySQL instance. This dataset includes a total of 45.8 million users and 109.2 million pull requests. Since it is not feasible to include such a large number of PRs, we focused on selecting a sample of 1,000 projects. Following the recommendations of Kallamvakou et al. (2016), we defined the following selection criteria to include only software projects actively using the PR-based development model (Gousios et al. 2014).
Table 1 Demographics of Gerrit projects

| Project      | Sponsor           | Request mined | # of developers* |
|--------------|-------------------|---------------|------------------|
| Android      | Google            | 637,064       | 3,351            |
| Chromium OS  | Google            | 1,536,071     | 5,375            |
| Couchbase    | Couchbase Inc.    | 140,124       | 300              |
| Go           | Google            | 47,096        | 335              |
| LibreOffice  | Foundation        | 127,812       | 573              |
| oVirt        | Redhat Inc        | 113,465       | 292              |
| Qt           | Qt Company        | 184,290       | 815              |
| Typo3        | Foundation        | 65,399        | 412              |
| Whamcloud    | Intel             | 21,724        | 158              |
| Wikimedia     | Foundation        | 494,302       | 904              |
| Total        |                   | 3,367,347     | 12,252           |

* Number of developers, who have made at least five commits to this project

1. A software development project using one of the popular programming languages, such as C, C++, Java, JavaScript, Python, PHP, Ruby, C#, Scala, Lua, Go, Kotlin, Typescript, and Rust.
2. Source code is available under an open-source license.
3. At least 20 contributors.
4. Actively follows pull-based development with at least 20 PRs during the last three months in our dataset.

Project selection: To facilitate our selection, we wrote queries to compute the number of distinct contributors and the number of PRs for each repository. We hypothesized that the characteristics of gender bias may also depend on community size. To investigate this hypothesis, we divided the projects into four groups based on the number of contributors. To facilitate referencing these four groups, we named them according to community size, which is as follows:

1. Small [GitHub (S)]: Projects that have 20-100 contributors.
2. Medium [GitHub (M)]: Projects having 101-250 contributors.
3. Large [GitHub (L)]: Projects with 251-500 contributors.
4. Extra large [GitHub (XL)]: Projects with more than 500 contributors.

From each group, we randomly selected 250 projects that satisfied our four selection criteria. Since GHTorrent dump does not include all the attributes necessary for this study, we mined those data directly from GitHub. To ensure that all the selected projects are still available, we wrote a Python script using the PyGithub library (Jacques https://pygithub.readthedocs.io/). From our initial selection, we found 27 unavailable projects. We randomly selected 27 available projects to replace those unavailable ones.

After finalizing the set of projects, we wrote queries to identify all the users who have opened at least one PR or made at least a commitment to one of the 1,000 selected projects. We found a total of 458,486 GitHub users meeting these criteria. After excluding organizations and deleted accounts based on the attributes listed in the Users table, we found a total of 310,983 users. GHTorrent also does not include personal information such as a full name, email address, bio, blog, location, avatar, or Twitter that may assist us in gender identification.
We wrote a Python script to mine that information and populate our ‘user_details table. We wrote another Python script to download the avatars of all users and store avatar images with the corresponding GitHub profile id. We used the same keyword-based filtering followed by the manual validation approach used on the Gerrit dataset, to identify and remove bot accounts from our dataset.

Due to the extremely large size of the GHTorrent dataset, queries take a significant amount of time. Therefore, we wrote a script to create new tables with a subset of rows that are related to our 1,000 selected projects from the full tables. We put those shortened tables in a new database called ‘ghtorrent_short’. Table 2 shows the demographics of the selected projects from our ghtorrent_short dataset. Our dataset includes a total of 2,839,192 PRs.

Initially, we selected 250 projects from each group (GitHub(S), GitHub(M), GitHub(L), GitHub(XL)) for data analysis. Unfortunately, while downloading pull request details for each of those projects, some of the projects were missing. So, we could not take exactly 250 projects from each group, instead, it is 250 ± 2. However, the total number of projects remains at 1000 for the GitHub dataset.

**Fetching pull request details:** Our study also requires several attributes of each PR that are unavailable in GHTorrent. We wrote a Python script using the PyGithub library (Jacques https://pygithub.readthedocs.io/) to mine the following attributes for each PR: the commit count, the number of changed files, the sum of lines added, the sum of lines deleted, the date created, the date pull merged, the date closed, the total number of comments, the current status, the merge SHA, and the raw URLs of the files. Our script also downloads all the listed files in a PR using the raw URLs and computes various attributes such as cyclomatic complexity, number of lines, number of lines that are comments, number of methods, lines added, lines deleted, and file status. We have added two new tables, pull_details and pull_files in our GHTorrent_short schema to store that information. Section 5.2 provides more details on how each of these attributes is computed.

## 5 Variable Selection & Computation

In the sections that follow, we explain how we chose confounding variables that were important to our research questions, how we calculated attributes, and how we analyzed the data.

### 5.1 Variable Selection

Table 3 lists the three dependent, two independent, and eleven confounding variables for our study. We have grouped the confounding variables into two categories, i.e., contributor characteristics and changeset characteristics. Both of our independent variables (i.e., gender

| Project  | # of contributors | PR mined   | Total users* |
|----------|-------------------|------------|--------------|
| Github(S)| 20-100            | 80,606     | 11,714       |
| Github(M)| 101-250           | 283,920    | 36,244       |
| Github(L)| 251-500           | 647,574    | 70,162       |
| Github(XL)| more than 500    | 1,827,092  | 192,863      |
| Total    |                   | 2,839,192  | 310,983      |

* Number of developers, who have opened at least one PR to these group
Table 3 Descriptions and rationale of the dependent and independent variables

| Name | Description | Rationale |
|------|-------------|-----------|
| **Independent variables** | | |
| **Variable group: Contributor characteristics** | | |
| Gender (Gender) | Whether a contributor is a man or woman | This study’s objective is to identify influences of gender. |
| Gender neutral profile (isGenderNeutral) | If a contributor’s profile is gender neutral | Terrell et al. found identifiable gender influencing acceptance rates (Terrell et al. 2017). |
| Number of total commits (totalCommit) | Number of code commits a developer has made to a project | Changes submitted by experienced developers who have higher number of commits can be subjected to less scrutiny and may include lower number of mistakes than inexperienced developers (Bosu et al. 2016). Experienced developers are also more likely to be invited for code reviews (Mirsaeedi and Rigby 2020). |
| Tenure (tenure) | Span of time developer is contributing to the project | Developers who are working on project for longer period time are more knowledgeable of the project’s design (Mirsaeedi and Rigby 2020), are able to identify potential reviewers quickly than newcomers (Bosu and Carver 2014), and are more likely to be invited as reviewers. |
| Review experience (revExp) | The number of pull requests that a developer has examined in the project as a reviewer | Experienced reviewers are more likely get invited for more reviews (Mirsaeedi and Rigby 2020) |
| **Variable group: Changeset characteristics** | | |
| Patch size (patchSize) | Number of lines modified/added/deleted in a code review request | Larger changes are more likely to be buggy, require longer time to review (Bosu et al. 2014) and less like to be accepted (Jiang et al. 2013). |
| Cyclomatic complexity (cyCmplx) | McCabe’s cyclomatic complexity | When a patch set is difficult to comprehend, it will need a longer review intervals (Bacchelli and Bird 2013) and is more likely to be rejected. |
| Number of patchset (numPatch) | Number of revisions executed in the submitted request | Higher number of patchsets would increase review intervals as well as probability of rejections (Bosu and Carver 2014). |
| Is bug fix (isBug-Fix) | Whether a change includes a new feature or bug fix | Bug fixes may be in a urgent need and get reviewed quickly (Paul et al. 2021). |
| Number of files (fileCount) | Number of files under review in review requests | Changes with higher number of files involved, are more likely to be defect-prone (Kononenko et al. 2015) and require longer review time. |
| Number of directories (dirCount) | Number of directories in a review request where files have been modified | Changes spread across a large number number of directories are more likely to be buggy, difficult to comprehend, and require longer review time (Barnett et al. 2015). |
| Comment volume (cmtVolume) | Ratio of add/modified lines that are comments | Well commented changes are easy to comprehend (Barnett et al. 2015) and require shorter review intervals. |
Table 3 continued

| Name                                      | Description                                                                 | Rationale                                                                 |
|-------------------------------------------|------------------------------------------------------------------------------|---------------------------------------------------------------------------|
| Ratio of new files (ratioNew)             | Ratio of newly created files to total number of files in a patch-set         | A new file may have more issues than a file which has been under review before (Xia et al. 2015). As a result, a new file can have longer review interval and less acceptance rate. |

Dependent Variables

- **Acceptance (isAccepted)**
  - Whether a code change was accepted or rejected
  - Indicates the outcome of a code review or pull request.

- **Code review interval (reviewInterval)**
  - Time from beginning of the code review to the end
  - Indicates whether a contributor’s code gets preferential treatments in a FOSS community.

- **Code review participation (avgReviewPerMonth)**
  - Average number of code review where the developer participated in the project per month
  - Indicates whether a contributor is valued by his/her peers.

We group the independent variables into two categories: i) contributor characteristics and ii) changeset characteristics. For building the ‘CR participation model’, we will use only the independent variables listed in the ‘contributor characteristics’ group.

and gender neutral profile) also belong to the contributor characteristics group. Table 3 also includes a description and the rationale behind the selection of each of the variables. We selected a confounding variable for inclusion if either a prior study has found its influence on CR outcomes (Gousios et al. 2014; McIntosh et al. 2015; Lenarduzzi et al. 2021; Jeong et al. 2009; Tao et al. 2014; Fan et al. 2018; Barnett et al. 2015; Thongtanunam et al. 2015) or it was included in one of the two original studies (Terrell et al. 2017; Bosu and Sultana 2019). The name of the variable inside parentheses denotes its acronym, which we have used to refer to it hereinafter.

5.2 Variable Computation

The following subsections detail the computation procedure for the three dependent, two independent, and eleven confounding variables.

5.2.1 Gender (Gender)

Our four-step, semi-automated gender resolution steps are adopted from the methodologies used by Terrell et al. (2017) and Bosu et al. (2019). Similar to those studies, we used the GenderComputer tool (Vasilescu et al. 2014) to resolve developers’ genders from name and location as the first step. In the second step, we used face detection and gender recognition machine learning models to identify genders from profile avatars. In the third step, we manually verified genders using LinkedIn and social media searches if the first two steps failed or returned contradictory binary genders, e.g., name resolution suggests a man but avatar suggests a woman.

1. **Step 1: Gender resolution from name** To resolve the gender of the GitHub users, we started with using GenderComputer.4 This tool takes the first and last name of a person and their

---

4 [https://github.com/tue-mdse/genderComputer](https://github.com/tue-mdse/genderComputer)
location, if that is available. The GenderComputer tool classifies a person into one of the following four categories: 1) Male, 2) Female, 3) Unisex, and 4) Unknown. We executed this step for all the users from both the GHTorrent and Gerrit datasets.

2. **Step 2: Identification of gender from avatar** We used avatars downloaded from Gerrit and GitHub profiles to infer gender. If a user has not customized their avatar, an identicon, which is a 5×5 pixel sprite generated using the hash of the user’s ID, is used. We used empirically developed file size-based heuristics to identify icons. Since we noticed that all the identicons are less than 1.5 kilobytes, we moved all the avatars less than 1.5 kilobytes to our “identicon” directory. However, this size-based heuristic is not entirely accurate, as a very small number of custom avatars also fall into this size group and vice versa. To fix such misclassifications, we manually inspected all the identicons. To save inspection time, we inspected the files in the thumbnail view mode of the directory. However, since we have a large number of files, loading a directory with a large number of files in the thumbnail view is time/resource consuming. To counter this challenge, we executed a shell command to divide the “identicon” directory into multiple sub-directories, with each directory including 1,000 files. During our thumbnail view of each sub-directory, if we noticed any file that was misclassified, we used the “drag and drop” action to move the file to the correct directory.

After excluding the identicons, we used the Haar Cascade human face detection algorithm from the OpenCV library (Goyal et al. 2017) on the non-identicons. We wrote a Python script that places each non-identicon into one of the following three groups: 1) **Unknown**: images with no human faces; 2) **single**: images with only one human face; and 3) **multiple**: images with multiple human faces. After this script’s execution, we noticed misclassifications ranging from 10-15%. For correction, we used the same thumbnail-based viewing strategy used for grouping identicons /non-identicons to move misclassified images to the correct directories.

After manual corrections, we used another Python script to automatically infer gender from the “single” images. This script uses the OpenCV library to automatically identify rectangular sections containing faces. Next, it uses pre-trained models created by Eidinger et al. (2014) to classify each image as ‘man’ or ‘woman’. To correct misclassifications, we repeated our thumbnail view-based manual inspection strategy. During this manual inspection phase, we also inspected the misclassifications in a full-scale view, as some images could be only determined after careful inspections. Moreover, we also noticed many avatars are photos of children. During our manual inspections, we moved avatars with children seemingly less than 13 years old to the ‘Neutral’ directory based on the assumption that such photos are of someone other than the account holder. We chose 13 years as the threshold since GitHub policy requires a user to be at least 13 years old. During our manual inspections, we also noticed avatars using photos of famous singers, actors, cartoon characters, well-known internet memes, and athletes. We also categorized such avatars as “neutrals, as these avatars represent users’ interests rather than themselves. Finally, we manually inspected all the avatars with multiple human faces. If an avatar includes only one person over 13 years of age, we assign that avatar to that person’s gender. For example, such a photo could be of a father with his son or daughter, and this account belongs to the father. However, if multiple persons over 13 years of age were present, we classified avatars as “neutrals.” While our gender inferences for avatars were extremely time-consuming, we resorted to this process due to the unavailability of another reliable alternative. To ensure the reliability of our results, we wanted to ensure very high accuracy in this step. From GHTorrent, we identified a total of 10,081 women and 143,094 men, with the remaining having either identicons or neutral avatars. Our
replication package also includes examples from each category (Sultana et al. 2023). We used the same approach to infer genders from Gerrit avatars as well. However, among the ten Gerrit projects, four projects-LibreOffice, Qt, WhamCloud, and Wikimedia-do not use any profile avatar. Therefore, we skipped this step for those four projects.

3. **Step 3: Semi-automated resolution** For each user, we inferred their gender from two sources: their full name and location, and their profile avatar. If both of these results match, we label a person’s gender as that. If either of the results is ‘unknown’ and the other one suggests man or woman, we assign that gender to the user. If the results conflict, we check the profiles manually and conduct an additional investigation using social media (i.e., LinkedIn and Facebook) to decide. If both name and avatar-based resolutions fail for a person, we assign ‘Unknown’ to that user.

4. **Step 4: Manual search** We conducted manual investigations of the unresolved profiles (i.e., ‘Unknown’ at the end of Step 3) to determine genders. First, we exclude all the users that do not list at least one of the following: name, email address, blog, or Twitter. For the remaining users, we used an approach similar to Bosu et al. (2019). However, since Google+ is no longer available, we replaced it with the Google People API. Moreover, since the manual resolution is extremely time-consuming, we also excluded users, who had opened fewer than five CRs. In the first step, we searched for Google people profiles using email addresses to identify photos. If a person is not found or we cannot decide based on the associated photo, we search for that person on LinkedIn using their name and email domain. If a Twitter account or a personal website was listed on a person’s profile, we also inspected those.

Although our gender resolution steps are time-consuming, we were able to achieve a high-resolution rate. For GitHub, we were able to resolve 71.5% users. For the GitHub users who have opened at least 5 PRs and have emails listed on their profiles, we resolved 89.6%. For Gerrit-based projects, overall we achieved 98.8% success for non-casual developers. Since our study is limited to investigating gender bias in men and women, we removed user ids for whom we assigned “unknown” as gender. For data analysis, we coded ‘0’ for men and ‘1’ for women.

Table 4 shows the demographics of gender for the Gerrit projects and four groups of GitHub projects. The ratio of women varies between 4-19%, with the highest ratio seen in Wikimedia. Among the GitHub groups, the ratio of women increases as the project size grows. Overall, we found 7.4% women.

### 5.2.2 Gender Neutral Profile (isGenderNeutral)

Prior study shows that women often use gender-neutral profiles (GNP) to avoid unwanted attention. Terrell et al. also showed that women have higher acceptance rates when their gender is not identifiable. To test our formulated hypotheses: $H_{1.2}$, $H_{2.2}$, $H_{3.2}$, we need to identify if a profile is gender neutral. If users’ genders are identifiable from their avatars, we consider them to be gender identifiable profiles (GIP) and put down 0’s as the values for isGenderNeutral variable. Otherwise, we considered their profiles as gender-neutral profiles (GNP) and put 1’s for this variable.

We like to mention that our GNP/GIP resolution strategy is different from Terrell et al.’s. To identify if a profile uses identicon, Terrell et al. checked for image dimensions using ImageMagick. If a profile used an identicon and the gender inference tool classified its user

---

5 https://developers.google.com/people
### Table 4  Demographics of gender in Gerrit and GitHub datasets

| Project         | Men (with gender) | Women (with gender) | Men using GNPs | Women using GNPs |
|-----------------|-------------------|---------------------|----------------|------------------|
| Android         | 2896 (86.7%)      | 444 (13.3%)         | 1619 (55.9%)   | 306 (68.9%)      |
| Chromium OS     | 4,466 (83.3%)     | 897 (16.7%)         | 3,347 (74.9%)  | 728 (81.2%)      |
| Couchbase       | 265 (88.3%)       | 35 (11.7%)          | 160 (60.4%)    | 28 (80.0%)       |
| Go              | 313 (93.4%)       | 22 (5.6%)           | 188 (60.1%)    | 18 (81.8%)       |
| LibreOffice‡    | 518 (90.9%)       | 52 (9.1%)           | -              | -                |
| oVirt           | 258 (88.4%)       | 34 (11.6%)          | 208 (80.6%)    | 29 (85.3%)       |
| Qt‡             | 777 (95.3%)       | 38 (4.7%)           | -              | -                |
| Typo3           | 392 (95.2%)       | 20 (4.8%)           | 195 (49.7%)    | 13 (65.0%)       |
| Whamcloud‡      | 145 (91.8%)       | 13 (8.2%)           | -              | -                |
| Wikimedia‡       | 637 (81.3%)       | 147 (18.7%)         | -              | -                |
| Github(S)       | 9,064 (94.1%)     | 566 (5.9%)          | 2,929 (32.3%)  | 277 (48.9%)      |
| Github(M)       | 26,279 (93.3%)    | 1,886 (6.7%)        | 8,810 (33.5%)  | 934 (49.5%)      |
| Github(L)       | 49,561 (93.2%)    | 3,604 (6.8%)        | 17,002 (34.3%) | 1,853 (51.4%)    |
| Github(XL)      | 121,765 (92.7%)   | 9,639 (8.5%)        | 42,241 (34.7%) | 4,887 (50.7%)    |
| **Total:**      | **217,336 (92.6%)** | **17,397 (7.4%)**  | **76,699 (35.3%)** | **9,073 (52.2%)** |

‡ - since these projects do not provide an option to use avatar, we consider all profiles as gender neutrals

Bold one represents the sum of all the rows for that column

name as “unknown, they used a panel of three people to look into the login and user name for 10 seconds to determine whether they could reliably guess the gender. However, they added this manual step only for 3,000 randomly selected profiles. During our study, we noticed two shortcomings of this strategy. First, we noticed many custom avatars with the same dimensions as identicons. Therefore, a non-negligible number of identities identified based on this strategy are false positives. Second, custom avatars can also be gender-neutral (e.g., pets, celebrities, kids, inanimate objects). Third, inferring gender from a name also requires familiarity with the origin. For example, most people from the Indian subcontinent may easily infer the name ‘Sayma’ belongs to a woman. However, people of different origins may have difficulties. Moreover, even though a name-to-gender inference tool can reliably guess gender from a name, CR participants are less likely to use such tools during their collaborations. Therefore, to avoid such limitations, we considered only avatars as sources to identify gendered profiles, i.e., we marked a profile as GIP only if we could reliably identify its owner’s gender using the associated avatar. This step is similar for both Gerrit and GitHub datasets. Since four of the Gerrit projects (i.e., LibreOffice, Qt, Whamcloud, and WikiMedia) do not use profile avatars, values for this variable are 0 for all of the users from those projects. Table 4 also shows the distributions of developers from various projects using GNPs. We also noticed the majority of women (52.2%) prefer GNPs, while almost two-thirds of men prefer GIPs.

### 5.2.3 Commit Experience (totalCommit)

For each CR or PR, we count code commit experience as the number of prior code commits for the current project. We wrote a SQL script to compute this attribute.
5.2.4 Author Tenure Months (tenure)

We measured the tenure of a developer from the difference between the timestamp of the first submitted CR and the most recently submitted ones for a particular project. We first computed the differences in terms of the number of days. After dividing by 30, we take the ceiling values to convert those differences into the number of months.

5.2.5 Review Experience (revExp)

For each CR or PR, we measure the review experience of a participant by computing the number of distinct CRs that he/she has participated in, where he or she was not the owner (or creator of the PR on GitHub). We wrote a SQL script to compute this attribute.

5.2.6 Patch Size (patchSize)

We first mined code churn for individual files (i.e., file_code_churn under a CR). We computed the patch size using the following formula:

\[
\text{patchSize} = \sum_{i=1}^{n} \text{file\_code\_churn}_i,
\]

where a CR includes \( n \) files and \( \text{file\_code\_churn}_i \) denotes the code churn of \( i_th \) file.

5.2.7 Cyclomatic Complexity (cyCmplx)

We wrote a Python script that downloads all files included in a CR and uses the Lizard python library (Yin https://pypi.org/project/lizard/) to compute cyclomatic complexity for each file. For the regression model, we compute the average cyclomatic complexity of all files included in a CR.

5.2.8 Number of Patchset (numPatch)

For Gerrit, this variable denotes the number of review iterations made before a decision was made for a CR. For GitHub, this variable denotes the number of commits included in a PR.

5.2.9 Is Bug Fix (isBugFix)

We used a heuristic-based approach similar to the one proposed by Zafar et al. to identify whether a CR is for a bug fix (Zafar et al. 2019). To identify CRs with bug fixes, we checked the description of the CR for links to bug repositories. Every Gerrit-based project uses a different style to link bug reports. We empirically investigated each Gerrit project to create regular expressions to identify its bug links. For example, CRs associated with bug fixes for oVirt include ‘Bug-Url: https://bugzilla.redhat.com/([0-9]+)’ in descriptions. For GitHub-based projects, we first marked bug-associated labels using a regular expression search. Such labels typically include keywords such as ‘bug’ and ‘defect’. We checked if a PR is linked with an issue, and that issue is assigned one of the bug-related labels.

5.2.10 Number of Files (fileCount)

We wrote a Python script using the Pygithub library (Jacques https://pygithub.readthedocs.io/) to mine the number of files included in a PR. Our Gerrit miner also mines this information for all CRs.
5.2.11 Number of Directories (dirCount)

We wrote an SQL query to extract directory names from file paths. Another query was used to compute the number of distinct directories in a CR.

5.2.12 Comment Volume (cmtVolume)

We wrote a Python script that downloads exact versions of all the files included in a CR and computes the number of lines (nLOC) and the number of lines that are comments (nCOM) using the Lizard library (Yin https://pypi.org/project/lizard/). We compute the average comment volume based on the following formula.

\[
\text{cmtVolume} = \frac{1}{n} \times \sum_{i=1}^{n} \frac{nCOM_i}{nLOC_i}
\]

where a CR includes n files and \(nCOM_i\) and \(nLOC_i\) denote nCOM and nLOC of the \(i_{th}\) file respectively.

5.2.13 Ratio of New Files (ratioNew)

The Gerrit miner already mined whether a file was new or modified. Our Python script mined the same for all the PRs. We wrote a SQL script to compute the total number of files (i.e., numNewFile) that are new in a review. We compute this attribute as: \(\text{ratioNew} = \frac{\text{numNewFile}}{\text{fileCount}}\)

5.2.14 Acceptance (isAccepted)

The Gerrit miner downloads the status of CRs as one of the following: ‘New’, ‘Merged’, or ‘Abandoned’. We excluded all the CRs with the status ‘New’, since those are ongoing. For the reviews with status marked as ‘Merged’ and ‘Abandoned, we set \(\text{isAccepted} = 1\) and \(\text{isAccepted} = 0\) respectively.

For each PR, we mined timestamps for ‘merged_at’ and ‘closed_at’. We excluded open PRs (i.e. ‘closed_at’ = NULL). We noticed that if a PR is closed without merging, its ‘merged_at’ is set as NULL. Therefore, for the PRs with ‘merged_at’ = NULL, we set \(\text{isAccepted} = 0\). For the remaining PRs, we set \(\text{isAccepted} = 1\).

5.2.15 Review Interval (reviewInterval)

For each PR, we computed the number of seconds between the time when the PR was created (i.e. created_at) and the time when that PR was closed (i.e., closed_at). Similarly, for Gerrit-based reviews, we used differences between created and last updated. By dividing the review interval by 3600, we computed the review interval in terms of hours.

5.2.16 Code Review Participation (avgReviewPerMonth)

For each reviewer, we computed the average number of CRs over their tenure with a project. This variable is the dependent variable for RQ3. We take monthly snapshots of this variable and five independent variables (i.e., participant characteristics) over the tenure of a participant. Such snapshots are tuples listing \(\text{tenure, gender, isGenderNeutral, totalCommit, reviewExp, avgReviewPerMonth}\). We took monthly snapshots
instead of one single average number since the former allows us to investigate whether women had different CR participation trajectories during their tenure than men. For example, women may start slowly in terms of review participation but, once recognized, may catch up.

To illustrate our computation, let us assume that Alice has been contributing to the Go project for 4 months. She is a woman with a gender-neutral profile. During the months of her tenure, she committed 5, 3, 7, and 9 changes, respectively. Her number of CR participation during these months is 2, 4, 3, and 7, respectively. Therefore, her average CRs are 2, 3, 3, and 4, respectively. To build our regression model, we include four data points for Alice, which are: (1, Woman, Yes, 5, 2, 2), (2, Woman, Yes, 8, 6, 3), (3, Woman, Yes, 15, 9, 3), (4, Woman, Yes, 24, 16, 4).

5.3 Regression Model Training and Analysis

We use multivariate regression modeling techniques to evaluate our hypotheses. Linear regression and logistic regression are commonly used to inspect relations among one dependent variable and one or more independent variables when the dependent variable is scalar or binary, respectively (Boston University SoPH 2016). To develop Code acceptance models, we use logistic regression since the dependent variable is Accepted is dichotomous. To analyze Code Review Interval and Code Review Participation, we use linear regressions, since the dependent variables (i.e., reviewInterval and avgReviewPerMonth) are scalars. We have grouped the independent variables into two categories: (i) contributor characteristics and (ii) changeset characteristics. While both groups of independent variables are used to train Code acceptance and Code Review Interval models, we use only the ‘contributor characteristics’ group to train Code Review Participation models. Since changeset characteristics are specific to a particular code change, those may not be influential in deciding how many reviews a person participates in during a period. Table 3 describes the variables for our models.

Recent SE studies (McIntosh et al. 2015; Bosu and Carver 2014) inspired us to adopt the models’ construction and analysis approach proposed by Harrell Jr. to validate our proposed hypotheses (Harrell 2015). Harrell’s approach enables us to formulate non-linear relationships among variables while being aware of the possibility of overfitting (i.e., where the model performs very well for the training dataset but poorly otherwise) (Harrell 2015). We detail our five-step model development and analysis approach in the following.

5.3.1 Step 1: Normality Adjustment

We found skewed distributions for three independent variables (i.e., totalCommit, revExp, and patchSize) and one dependent variable (i.e., reviewInterval. Following recommendations from prior CR studies (McIntosh et al. 2015; Thongtanunam et al. 2016) to include such variables, we apply log transformations (i.e., log_e(x)).

5.3.2 Step 2: Correlation and Redundancy Analysis

Before training regression models, we must assess multicollinearity, which occurs when two or more independent variables are highly correlated with one another in a regression model. Without such an assessment, a model may not be able to reliably determine the effects of individual independent variables (Mansfield and Helms 1982). We conduct correlation analysis among the independent variables of the models to identify highly correlated variables. We execute the Spearman rank correlation tests between all pairs of independent variables (ρ).
to construct hierarchical overviews of the correlated variables. Among the variables residing in the same sub-hierarchy and having $|\rho| < 0.7$, we select only one for the final regression model. Prior studies in software engineering also considered 0.7 as the threshold value for identifying redundant variables (McIntosh et al. 2015; Thongtanunam et al. 2016; Bosu and Carver 2014). We repeated this step for each of the $14 \times 6 = 84$ regression models that we trained. Figure 1 shows our variable clustering for our acceptance model from the Go dataset. The red trend line indicates our cutoff correlation (i.e., 0.7). If the cluster meeting point for two or more variables is below this trend line, we select only one variable (i.e., the one with the highest correlation with the dependent variable) from that cluster. In this example, `fileCount` and `dirCount` are highly correlated. Among these two, we took only `dirCount` for training this project’s regression model, since it has a higher correlation with the dependent variable `isAccepted`. Similarly, from the correlated pair (`logRevExp`, `logTotalCommit`), we took only `logRevExp`. With the surviving variables, we computed the variance inflation factor (VIF). None of the predictors have a VIF greater than 2.5, suggesting that multicollinearity is not a substantial threat.

5.3.3 Step 3: Modeling Non-Linear Relationships

Restricted cubic splines (RCS) are often used during regression modeling to model non-linear relationships between explanatory variables and outcome (Durrleman and Simon 1989). In this transformation strategy, the range of values of an independent variable is split up, with “knots” defining the end of one segment and the start of the next. During regression, if more degrees of freedom or independent variables are used than the dataset can afford, regression models may overfit. An overfitted model fails to show actual relationships among dependent and independent variables. We calculate the budget for total degrees of freedom for our code review dataset, following Harrell Jr.’s recommendations of at most $n/15$ total degrees of freedom, where $n$ is the number of rows in a dataset (Harrell et al. 1984, 1985). For example, our smallest dataset (i.e., Whamcloud) includes 21,555 rows after excluding unresolved participants. Therefore, we can allocate at most 1,437 degrees of freedom without overfitting. The maximum number of degrees of freedom allocated for model training was 27. Therefore, our models do not have overfitting threats.

![Fig. 1 Variable clustering for the Go project for training code acceptance model](image_url)
However, we cannot allocate additional degrees of freedom to dichotomous variables such as Gender, isGenderNeutral, and isBugFix. For the remaining independent variables that survived multicollinearity analysis, we allocated four degrees of freedom using the rcs method from the rms R package.

5.3.4 Step 4: Training Regression Models

Equations (1), (2), and (3) denote regression models for Code acceptance, Code review Interval, and Code review participation respectively. To test the interaction between Gender and isGenderNeutral (i.e., $H1.2$, $H2.2$, and $H3.2$), we replaced Gender and isGenderNeutral factors with Gender * isGenderNeutral, a four-level interaction factor, in each of the following three equations. We use the lm method from the R stats package to fit the models. Since the Code acceptance model has binary outcomes, we use binomial linking function. For the remaining two models, we use gaussian aka ‘identity’ linking. However, these equations differed among the projects if some variables were found to be redundant during Step 2.

1. \( \text{isAccepted} \sim rcs(\log(\text{totalCommit}), 4) + rcs(\log(\text{patchSize}), 4) + rcs(\log(\text{revExp}), 4) + rcs(\text{tenure}, 4) + rcs(\text{cyCmplx}, 4) + rcs(\text{numPatch}, 4) + is\text{BugFix} + rcs(\text{dirCount}, 4) + rcs(\text{cmtVolume}, 4) + rcs(\text{ratioNew}, 4) + rcs(\text{fileCount}, 4) + \text{Gender} + is\text{GenderNeutral} \) (1)

2. \( \log(\text{reviewInterval}) \sim rcs(\log(\text{totalCommit}), 4) + rcs(\log(\text{patchSize}), 4) + rcs(\log(\text{revExp}), 4) + rcs(\text{tenure}, 4) + rcs(\text{cyCmplx}, 4) + rcs(\text{numPatch}, 4) + is\text{BugFix} + rcs(\text{dirCount}, 4) + rcs(\text{cmtVolume}, 4) + rcs(\text{ratioNew}, 4) + rcs(\text{fileCount}, 4) + \text{Gender} + is\text{GenderNeutral} \) (2)

3. \( \text{avgReviewPerMonth} \sim rcs(\text{tenure}, 4) + rcs(\log(\text{totalCommit}), 4) + \text{Gender} + rcs(\log(\text{revExp}), 4) + is\text{GenderNeutral} \) (3)

5.3.5 Step 4: Assessment of Model Performance

There are several metrics to assess the goodness of fit for regression models. In this study, we use Veall and Zimmerman Pseudo $R^2$ (Veall and Zimmermann 1994) for the logistic regression models, since prior research (Smith and McKenna 2013) found this measure having closer correspondence to ordinary least square $R^2$. Veall and Zimmerman’s index is defined as follows:

\[
R^2_{ve} = \frac{2[LL(\text{null}) - LL(\text{full})]}{2[LL(\text{null}) - LL(\text{full})] + N} - \frac{2LL(\text{null}) - N}{2LL(\text{full})}
\]

Where $LL(\text{full})$ is the log-likelihood of the full model and $LL(\text{null})$ represents the same for a model eliminating all the independent variables (i.e., intercept-only model).

For linear regression models, we measure Adjusted $R^2$ (Yin and Fan 2001) to assess goodness of fit.
5.3.6 Step 5: Interpretation of Results

For the logistic regression model (i.e., H1), we compute the Odds Ratio (OR) to estimate the impact of our variables of interest (i.e., Gender, and isGenderNeutral). OR represents the odds that an outcome will occur with the presence of a factor, compared to the odds of the same outcome occurring in its absence (Fulton et al. 2012). OR is very useful in estimating the impact if a variable of interest is dichotomous. OR = 1 indicates no impact of an independent variable on the dependent variable. A value of OR > 1 indicates that, with the increase of that particular independent variable, the dependent variable has higher odds of occurring, and OR < 1 indicates the opposite.

For the linear regression models (i.e., H2 and H3), interpretation is more straightforward. For these models, the regression coefficient of an independent variable indicates the number of changes to the dependent variable with 1 unit of change. A positive coefficient indicates positive impacts on the dependent variable, and vice versa. Moreover, a larger absolute value for an independent variable’s coefficient indicates a larger impact.

6 Results

We present the results of our study in this section for the proposed research questions. For each question, we address the analysis results and their interpretation in terms of evaluation metrics. As our study is limited to men and women, bias can work in two directions. If men are favored, we colored that attribute with blue, and if women are favored, we colored the cell with pink.

6.1 Gender vs. Code Acceptance (RQ1)

Table 5 shows the average acceptance rates for groups formed based on gender, GIP, and GNP. Acceptance rates between men and women differ by 1-9% among the projects. We noticed the largest difference in the LibreOffice project. Similar to Terrell et al. (2017), we noticed between 2-5% higher acceptance rates for women across three of the four GitHub project groups. For the GNP groups, 11 out of the 14 cases indicate women with GNPs having lower acceptance rates than women with GIPs; while three projects (i.e., Chromium OS, Couchbase, and Go) indicate the opposite. However, in contrast to Terrell et al., we did not find women with GNPs having higher acceptance rates on GitHub. However, this difference may be due to our different heuristics to determine whether a profile is a GNP.

Table 6 shows the results of our code acceptance logistic regression models. The ‘Pseudo $R^2$‘ column in Table 6 shows the performances of the models estimated using Veall and Zimmermann index (Veall and Zimmermann 1994). Columns 3-7 in Table 6 indicate the measure of association between our factors of interest and code acceptance based on OR. We also mark the statistical significance of those associations using three markers, where *** $\implies p < 0.001$, ** $\implies p < 0.01$, and * $\implies p < 0.05$. Since we code men with 0’s and women with 1’s, an OR > 1 (i.e., Gender column) implies that the probability of a CR’s acceptance increases when gender changes from 0 to 1 (i.e., Man to woman) while other factors remain unchanged, and vice versa. Similarly, an OR > 1 for Neutral profile
Table 5  Acceptance rates for men, women with gender identifiable profiles (GIP) and gender-neutral profiles (GNP)

| Project      | Overall    |       | GIP       |       | GNP       |       |
|--------------|------------|-------|-----------|-------|-----------|-------|
|              | Men        | Women | Men       | Women | Men       | Women |
| Android      | 86.48      | 81.26 | 84.80     | 86.67 | 87.98     | 77.98 |
| Chromium OS  | 86.34      | 87.40 | 87.32     | 85.26 | 86.06     | 87.79 |
| Couchbase    | 87.15      | 86.35 | 88.02     | 79.28 | 86.31     | 86.82 |
| Go           | 88.36      | 85.94 | 91.40     | 75.02 | 85.85     | 91.12 |
| Libreoffice  | 92.67      | 83.55 |          |       |           |       |
| oVirt        | 86.85      | 85.23 | 87.28     | 88.63 | 86.69     | 83.65 |
| Qt           | 83.95      | 87.77 |          |       |           |       |
| Typo3        | 91.33      | 90.20 | 91.87     | 91.40 | 90.02     | 80.12 |
| Whamcloud    | 75.12      | 75.86 |          |       |           |       |
| WikiMedia    | 91.07      | 88.52 |          |       |           |       |
| Github(S)    | 86.34      | 86.14 | 86.78     | 86.86 | 85.36     | 85.55 |
| Github(M)    | 78.46      | 82.40 | 77.92     | 83.35 | 79.57     | 81.21 |
| Github(L)    | 80.04      | 82.81 | 79.54     | 84.79 | 81.02     | 79.76 |
| Github(XL)   | 69.98      | 73.65 | 71.20     | 76.52 | 67.76     | 71.19 |

indicates higher odds of a CR’s acceptance with GNPs. To investigate interactions between the two key independent factors (i.e., gender and GNP), we trained a second regression model for each project, where we replace Gender and isGenderNeutral with Gender * isGenderNeutral. This interaction factor has four levels, indicating i) men with GIPs, ii) women with GIPs, iii) men with GNPs, and iv) women with GNPs. The second regression model for a project has almost identical goodness of fit as the first one, as we are only replacing two dichotomous variables with a four-level product variable while preserving other factors. These interaction models help us identify if GNPs /GIPs have different associations with code acceptance for men and women. For these interaction models, we consider men with GIPs as the reference group. Therefore, the OR values reported in columns 5-7 of Table 6 indicate odds of code acceptance with respect to men with GIPs. For example, for the Android project, the odds of code acceptance for women with GIPs and women with GNPs are 1.42 and 0.66, respectively, with respect to men with GIPs, if other factors remain unchanged.

6.1.1 Results of H1.1

Gerrit dataset: Results from Table 6 indicate that significant gender biases exist among nine out of the ten projects, with Typo3 being the only exception. Hence, nine projects support H1.1a, and the null hypothesis (H1.10) cannot be rejected for the remaining one. Based on OR values, seven out of those nine projects favor men. Three (i.e., Couchbase, Go, and LibreOffice) out of those seven projects indicate that women have less than three-fourths the odds of getting code accepted with respect to men, even if all other factors remain unchanged. On the other hand, two projects (i.e., Android and Qt) show significantly higher odds favoring women.

GitHub dataset: Our results also suggest significant gender biases among all four GitHub groups (i.e., support for H1.1a). The OR values suggest significantly higher odds of acceptance for PRs submitted by women on GitHub.
Table 6  Do gender/gender-neutral profiles of the contributors influence acceptance of their code changes?

| Project            | Pseudo R² | Gender Neutral profile | Women with GIP | Men with GNP | Women with GNP |
|--------------------|-----------|------------------------|----------------|-------------|---------------|
| Android            | 0.172     | 0.85***                | 1.42***        | 0.87***     | 0.66***       |
| Chromium OS        | 0.419     | 0.96***                | 0.84***        | 0.95***     | 1.17***       |
| Couchbase          | 0.319     | 0.73***                | 0.52***        | 0.83***     | 1.41*         |
| Go                 | 0.552     | 0.52***                | 1.09*          | 0.17***     | 0.75***       | 8.73***       |
| LibreOffice        | 0.459     | 0.81***                | 1.03           | 1.02        | 1.05*         | 0.72***       |
| oVirt              | 0.133     | 1.05***                | 0.99           | 0.99        | 1             | 0.83          |
| Qt                 | 0.060     | 1.05***                |                |             |               |
| Typo3              | 0.269     | 0.84 *                 |                |             |               |
| WikiMedia          | 0.380     | 0.84*                  |                |             |               |
| Github (S)         | 0.107     | 1.15**                 | 0.88***        | 1.23**      | 0.89***       | 0.87          |
| Github (M)         | 0.077     | 1.24***                | 1.12***        | 1.38***     | 1.15***       | 0.79***       |
| Github (L)         | 0.087     | 1.22***                | 1.06***        | 1.41***     | 1.09***       | 0.71***       |
| Github (XL)        | 0.052     | 1.27***                | 0.83***        | 1.34***     | 0.84***       | 0.9***        |

Cells in blue backgrounds represent significant biases favoring men and cells in pink backgrounds represent biases favoring women. ***, **, and * represent statistical significance at $p < 0.001$, $p < 0.01$, and $p < 0.05$ respectively. No values among columns indicate statistically insignificance. For the gender neutral hypotheses test (i.e., H1.2), gray background suggests a significant higher odds of women with neutral profiles getting their code accepted.

### 6.1.2 Results of H1.2

**Gerrit dataset:** As we have already indicated, we computed the isGenderNeutral attribute only for six projects, since the remaining four projects do not support avatars. OR values under the ‘Neutral profile’ column suggest that developers with GNPs have significantly lower odds of getting code accepted in three out of those six projects, while only one project (i.e., Go) indicates the opposite. OR values under Woman with GIP indicate that women from this group have significantly lower odds of acceptance than men with GIPs in three projects (i.e., Chromium OS, Couchbase, and Go), while Android shows the opposite correlation.

All six projects, except Typo3, indicate significant differences in acceptance rates between men with GIPs and men with GNPs (i.e., supporting $H1.2_a$). Men with GNPs have decreased odds of code acceptance in Android, Chromium OS, Couchbase, and Typo3, but have increased odds in oVirt. Our comparisons between women with GIPs and women with GNPs indicate significant differences (i.e., support for $H1.2_a$) in five projects. In three projects (i.e., Chromium OS, Couchbase, and Go), women with GNPs had significantly higher odds of acceptance with respect to women with GIPs. Women with GNPs also have significantly higher acceptance rates than women with GIPs among these three projects (Table 5). Therefore, GNPs increase the odds of acceptance for women in all three projects. Surprisingly, the odds are almost eight times higher in Go. Our further investigation found that most of the
experienced women in Go prefer using GNPs, while less experienced women use GIPs. This particular difference between experienced and less-experienced women in avatar selection results in surprisingly higher odds of acceptance for women with GNPs. While we do not speculate on the underlying reason for selecting GNPs, interviews of women from Go may shed more light. On the other hand, we noticed the opposite association in two projects (i.e., Android, and oVirt), as women with GNPs have lower odds of acceptance than both men and women with GIPs. In summary, five⁶ Out of the projects, some show significant differences in acceptance rates between men/women with and without gender-identifiable profiles and therefore support H₁₂ₐ.

GitHub dataset: We also noticed significant associations between GNP and code acceptance among all four GitHub groups. However, OR values under ‘Neutral profile’ suggest that among the projects belonging to S and XL groups, participants with GNPs have lower odds of acceptance than participants with GIPs. On the contrary, projects belonging to M and L groups show the opposite. Results from our interaction regression models suggest that women with GIPs have significantly higher odds of acceptance than men with GIPs across all four GitHub groups. Women with GNPs have lower odds than men with GIPs in projects belonging to three of the GitHub groups (i.e., M, L, and XL). We also noticed significant differences between men/women with GIPs and GNPs across all four groups, which therefore supports H₁₂ₐ. For women, GNPs significantly decreased the odds of code acceptance across all four groups. However, for men, GNP is associated with significantly increased odds in two groups (i.e., M and L) but has opposite odds in the two remaining groups.

We accept H1₁ₐ (there are significant differences in code acceptance rates for men and women) for nine out of the ten Gerrit projects and fail to reject the null hypothesis H1₁₀ for the remaining one (i.e., Typo3). We tested H₁₂ for six Gerrit projects. Among those H₁₂ₐ (there are significant differences in code acceptance rates between men/women with and without gender-identifiable profiles) is supported for five projects but not for the remaining one (i.e., Typo3). We also found support for both H₁₁ₐ and H₁₂ₐ across all four GitHub project groups.

Code acceptance likelihood differs based on gender in 13 out of the 14 tested cases. While all four Github groups show biases favoring women, most of the Gerrit projects show the opposite. Therefore, although gender biases in code acceptance are prevalent, the direction of such bases is not universal and depends on the project context.

6.2 Gender vs. Code Review Interval (RQ2)

RQ2: Do the genders of the contributors influence code review intervals for their code changes?

Figure 2 shows the distribution of review interval hours for men and women using bean plots. We log-transformed review interval hours to account for skewed distributions. Table 7

⁶ For our second sub-hypotheses (i.e., H₁₂ₐ, H₂₂ₐ, and H₃₂ₐ), we consider a project as supporting if either of the two gender groups indicates differences between members with and without GIPs.
shows correlations between gender and GNPs with code review intervals based on our linear regression models. Since the dependent variable for these models is log-transformed, exponentiating the coefficients yields the multiplication factor for that variable. For example, in the Android project, the coefficient for the Gender = 0.112. Therefore, when Gender changes from 0 to 1 (i.e., from man to woman), code review intervals increase $e^{0.112} = 1.12$ times. In Table 7, a Delay > 1 indicates higher code review intervals with the presence of a factor and vice versa. We also expressed the fit of the model for each project with Adjusted $R^2$. Similar to H1, we trained a second regression model for each project, where Gender and isGenderNeutral is replaced with Gender * isGenderNeutral. As men with GIPs are the reference group in these models, the values reported in columns 5-7 of Table 7 indicate a delay multiplier with respect to men with GIPs. For example, for the Chromium OS project, the delay multipliers for women with GIPs and women with GNPs are 1.07 and 0.77, respectively, with respect to men with GIPs (Table 7).

6.2.1 Results of H2.1

**Gerrit dataset:** Seven out of the ten projects show significant differences in CR intervals based on gender, thereby supporting $H2.1_a$. For the remaining three projects (i.e., Chromium OS, oVirt, and Wikimedia), the null hypotheses ($H2.1_0$) cannot be rejected. CRs submitted by women encounter significantly higher delays than men among three projects (Android, LibreOffice, and Typo3). On the other hand, the opposite is true for Couchbase, Go, Qt, and Whamcloud.

**GitHub dataset:** Our results for all four GitHub datasets support $H2.1_a$, with women having significantly lower CR intervals than men in all cases.
Table 7 Do gender/gender-neutral profiles of the contributors influence code review intervals for their code changes?

| Project       | Adj. \( R^2 \) | Gender Neutral profile | Women with GIP | Men with GNP | Women with GNP |
|---------------|----------------|------------------------|----------------|-------------|----------------|
| Android       | 0.433          | 1.12***                | 1.04***        | 1.11***     | 1.04***        |
| Chromium OS   | 0.235          | 1                      | 1.07***        | 1.24***     | 1.1***         |
| Couchbase     | 0.312          | 0.85***                | 1.16***        | 2.02***     | 1.17***        |
| Go            | 0.287          | 0.73***                | 1.13***        | 0.72***     | 1.13***        |
| LibreOffice   | 0.186          | 2***                   | —              | —           | —              |
| oVirt         | 0.398          | 1.01                   | 0.96**         | 0.78***     | 0.93***        |
| Qt            | 0.107          | 0.54**                 | —              | —           | —              |
| Typo3         | 0.183          | 1.13*                  | 0.76***        | 1.05        | 0.75***        |
| Whamcloud     | 0.268          | 0.77***                | —              | —           | —              |
| Wikimedia      | 0.265          | 1.01                   | —              | —           | —              |
| Github (S)    | 0.091          | 0.8***                 | 1.47***        | 0.72***     | 1.45***        |
| Github (M)    | 0.092          | 0.89***                | 1.03**         | 0.93**      | 1.04***        |
| Github (L)    | 0.099          | 0.85***                | 1.15***        | 0.74***     | 1.12***        |
| Github (XL)   | 0.109          | 0.97***                | 1.15***        | 0.89***     | 1.14***        |

Cells in blue backgrounds represent men with significantly lower review intervals and cells in pink backgrounds represent the same for women. For the \( p \) value columns, *** , **, and * represent values < 0.001, < 0.01, and < 0.5 respectively. No values among \( p \) columns indicate statistically insignificance. For the gender neutral hypotheses test (i.e., H2.2), gray background suggests significant lower review intervals for women with gender neutral profiles.

6.2.2 Results of H2.2

**Gerrit dataset:** Our results also suggest significant differences between developers with and without GNPs. Four of the projects indicate additional delays for authors with GNPs than for authors with GIPs. On the other hand, two projects (i.e., oVirt and Typo3) indicate shorter CR intervals for GNP groups. Delay values under Woman with GIP indicate that women from this group have significantly longer CR intervals than men with GIPs in three projects (i.e., Android, Chromium OS, and Couchbase), while Go and Ovirt show the opposite association. All six projects indicate significant differences in CR intervals between men with GIPs and men with GNPs (i.e., supporting \( H^2.2_a \)). For men, GNP is associated with significantly longer delays in four projects and significantly the opposite in the remaining two. Interestingly, although GNP is associated with shorter review intervals for men from oVirt and Typo3, this factor shows the opposite association for women from the same projects. Comparisons between women with GIPs and women with GNPs suggest significantly lower CR intervals for the GNP group in two projects (i.e., Chromium OS, and Couchbase) and the opposite in two (i.e., Typo3 and oVirt). Therefore, \( H^2.2_a \) is supported for women from four projects. In summary, all six projects support \( H^2.2_a \), with either men or women showing significant differences in CR intervals based on GIP selection.
GitHub dataset: We also noticed significant associations between GNP and CR intervals among all four GitHub groups. Participants with GNPs were more likely to encounter longer CR intervals than participants with GIPs. Similarly, women with GNPs had significantly longer CR intervals than men with GIPs among three of the four GitHub groups. The remaining GitHub group (i.e., M) shows significantly shorter CR intervals for such women. Men with GNPs had significantly longer CR intervals than men with GIPs in all four groups (i.e., support for $H2.2_a$). Comparisons between women with GIPs and GNPs suggest significantly longer CR intervals for the GNP group in all cases except M. In aggregate, $H2.2_a$ is supported by all four GitHub groups.

Our results support $H2.1_a$ (there are significant differences in code review intervals between men and women) for seven out of the 10 Gerrit projects. However, we fail to reject the null hypothesis $H2.1_0$ for three such projects (i.e., Typo3, oVirt, and Wikimedia). We found support for $H2.2_a$ (there are no significant differences in code review intervals between men/women with and without gender-identifiable profiles) in all six Gerrit-based projects allowing GIPs. We also found support for both $H2.1_a$ and $H2.2_a$ across all four GitHub project groups. GNPs are associated with significantly longer code review intervals in eight out of the 10 cases (i.e., 6 Gerrit-based, and 4 GitHub-based groups). However, women with GNPs had significantly shorter CR intervals in three of the cases: – Chromium OS, Couchbase, and GitHub (M).

6.3 Gender vs. Code Review Participation (RQ3)

Table 8 shows correlations between gender and GNPs with CR participation based on our linear regression models. Column Adj. $R^2$ reports the goodness of fit for the trained models. For these models, linear regression coefficients represent how CR participation per month changes if a factor changes from 0 to 1. A positive coefficient (Z) value indicates a positive correlation with a factor, and vice versa. For example, in the Android project, the coefficient for the Gender is -1.60. Therefore, if we change Gender from 0 to 1 (i.e., from man to woman) while keeping all other factors the same, monthly CR participation reduces...
Similar to H1 and H2, we trained a second regression model for each project by replacing Gender and isGenderNeutral with the interaction terms Gender * isGenderNeutral.

### Table 8: Do gender/gender-neutral profiles of the contributors influence their participation in code reviews?

| Project          | Adj. \(R^2\) | Values indicate \{regression coefficient\} | \(p-value\) |
|------------------|---------------|-------------------------------------------|-------------|
| Gender Neutral profile | Women with GIP | Men with GNP | Women with GNP |
| Android          | 0.244         | -1.6*** | 0.58*** | -1.64*** | 0.57*** | 0.07 |
| Chromium OS      | 0.307         | -2.14*** | -0.05 | -3.43*** | -0.23 | 1.57*** |
| Couchbase        | 0.386         | -3.17*** | 1.81*** | 5.14*** | 2.23*** | -9.49*** |
| Go               | 0.419         | -1.63*  | -1.06** | -5.27*** | -1.26** | 4.35*  |
| LibreOffice      | 0.343         | -0.85   | 0.6  | 6.2***  | 0.86  | -5.69* |
| oVirt            | 0.429         | -3.5*** | -0.8*  | -14.05*** | -2.25*** | 13.7*** |
| Qt               | 0.574         | -1.77*  | -0.8*  | -14.05*** | -2.25*** | 13.7*** |
| Typo3            | 0.273         | 4.57*** | 0.6  | 6.2***  | 0.86  | -5.69* |
| Whamcloud        | 0.224         | -2.15*  | -0.84*** | -1.61*** | -0.89*** | 0.68*  |
| Github (S)       | 0.078         | -0.63*  | 0.62*** | -0.73 | 0.61*** | 0.2   |
| Github (M)       | 0.094         | -0.05   | -0.42** | 0.86*  | -0.25 | -1.92*** |
| Github (L)       | 0.171         | -0.7*** | -0.45*** | -1.22*** | -0.54*** | 1.14*** |
| Github (XL)      | 0.138         | -1.25*** | -0.84*** | -1.61*** | -0.89*** | 0.68*  |

Cells in blue backgrounds represent men with higher code review participation and cells in pink backgrounds represent the same for women. For the \(p\) value columns, *** , **, and * represent values < 0.001, < 0.01, and < 0.05 respectively. No values among \(p\) columns indicate statistically insignificance. For the gender neutral hypotheses test (i.e., H3.2), gray background suggests a significant higher number of code review participation from women with neutral profiles.
6.3.1 Results of H3.1

_Gerrit dataset:_ Nine out of the ten projects indicate significant gender biases in CR participation and therefore support $H_{3.1a}$. LibreOffice is the only exception, where the null hypotheses ($H_{3.10}$ cannot be rejected. Eight out of those nine projects indicate women review significantly fewer CRs than men from the same project. Women had significantly higher CR participation only in Typo3.

_GitHub dataset:_ We observe that men review significantly higher numbers of PRs than women in three out of the four groups. Only the M group is an exception, where the difference is not statistically significant. Therefore, $H_{3.1a}$ is supported for three groups, and the null hypothesis ($H_{3.10}$) cannot be rejected for the remaining one.

6.3.2 Results of H3.2

_Gerrit dataset:_ Our results also indicate a significant association between CR participation and GNPs in four out of the six custom avatar-supporting projects. While GNP has a positive association with CR participation in Android and Couchbase, the opposite association is seen in Go and oVirt. Four out of the six projects also suggest that women with GIPs have significantly lower CR participation than men with GIPs belonging to the same project. The remaining two projects (i.e., Couchbase and Typo3) indicate the opposite, where women with GNPs had significantly higher CR participation. Four out of the six projects, excluding Chromium OS and Typo3, indicate significant differences in CR participation between men with GIPs and men with GNPs (i.e., supporting $H_{3.2a}$). While GNP is associated with lower CR participation for men in Go and oVirt, opposite associations are found in Android and Couchbase. Comparisons between women with GIPs and women with GNPs indicate significant differences in CR participation (i.e., support for $H_{3.2a}$) among five projects, where Android is the only exception. Women with GNPs have significantly higher CR participation than women with GIPs in three projects (i.e., Chromium OS, Go, and oVirt), and the remaining two projects (i.e., Couchbase and Typo3) indicate the opposite. In summary, all six projects support $H_{3.2a}$ with either men or women showing significant differences in CR participation based on GIP selection.

_GitHub dataset:_ Three GitHub groups except the S group indicate that participants with GNPs were more likely to participate in a significantly lower number of code reviews than participants with GIPs. While women with GIPs had significantly lower CR participation than men with GIPs in two groups (i.e., L and XL), the opposite trend is seen in the S group. Men with GIPs participate in significantly higher numbers of CRs than men with GNPs among projects belonging to the L and XL groups, while projects belonging to the S group show an opposite trend. In contrast, women with GIPs had significantly lower CR participation than women with GNPs in the L and XL groups, while the M group showed an opposite association. In summary, all four GitHub groups show significant differences in CR participation between men/women with and without gender-identifiable profiles and therefore support $H_{3.2a}$.

---

**Our analyses found support for $H_{3.1a}$ (there are significant differences in code review participation between men and women) in nine out of the ten Gerrit-based projects with Libreoffice being an exception. Three GitHub groups excluding the M group also support $H_{3.1a}$. All six GIP-supporting Gerrit projects as well as all four GitHub groups also support $H_{3.2a}$ (there are significant differences in code review participation between men/women with and without gender identifiable profiles).**
6.4 Results Summary

Table 9 provides an overview of the gender biases identified among our datasets. According to these results, gender biases are common during CRs/PRs. The review interval (i.e., H2) is the metric that shows the lowest number of gender-biased cases. Even so, 11 out of the 14 cases show biases in terms of review intervals. However, the directions of bias vary based on both the metric and the project. In general, among the Gerrit projects, men have favorable measures in the majority of the cases. Interestingly, women on GitHub have favorable measures in terms of both code acceptance and review interval. However, we find an opposite picture for GitHub women in terms of code review participation. Table 10 summarizes the biases due to the gender neutrality of profiles. These types of biases are also present in most cases. Although persons with GIPs are more likely to have favorable measures, GNP-favoring projects are also common.

7 Discussion

Comparison with Terrell et al. (2017): We investigated two key results from Terrell et al.’s study regarding patch acceptance and appearance in women. First, supporting their findings, we found significantly higher odds of PR acceptance for women on GitHub across all four groups. However, the same finding applies to only two out of the ten Gerrit-based projects, while seven projects suggest the contrary.

Second, we found three Gerrit-based projects (i.e., Chromium OS, Couchbase, and Go), where women with GNPs have significantly higher odds of getting code accepted than women with GIPs. This finding is similar to Terrell et al.’s. However, contrasting theirs, our findings suggest that GNPs significantly lower the odds of acceptance for women’s code on GitHub as well as for two Gerrit-based projects (i.e., Android, and oVirt). However, we would like to restate that these differences may be due to a different heuristic adopted by this study than Terrell et al.’s to identify GNPs.

| Hypothesis          | Gerrit favoring |               | Github favoring |               |
|---------------------|-----------------|---------------|-----------------|---------------|
|                     | # biases | Men | Women | #total | Men | Women |
| Code acceptance: H1.1a | 9      | 7   | 2    | 4      | 0   | 4     |
| Review interval: H2.1a | 7      | 3   | 4    | 4      | 0   | 4     |
| Review participation: H3.1a | 9      | 8   | 1    | 3      | 3   | 0     |

The numbers under a gender indicate the number of groups that had biases favoring that gender.
Comparison with Bosu and Sultana (2019): Bosu et al. (2019) and our study have nine common projects, and our study partially replicates their study, some of our results differ from their findings. First, they found less than 10% women across all the projects. However, we noticed four projects, including three common to both studies, exceeded that threshold.

Their results found significantly lower acceptance rates for women in Android, Chromium OS, and LibreOffice and the opposite in oVirt, Qt, and Typo3. While we obtained similar results for Chromium OS and LibreOffice, where women had lower odds of acceptance, we notice the opposite in Android. Similarly, although our results for Qt are similar to those of Bosu and Sultana (2019), the opposite is seen for oVirt. Our results from Typo3 are inconclusive.

For our second hypothesis (H2.1), Bosu and Sultana (2019) found biases favoring men in Android and Couchbase, whereas women are favored in Qt and Typo3. On the contrary, we found two projects (i.e., Couchbase and Typo3) flipping, with women being favored in Android and Typo3 but men being favored in Couchbase and Qt. However, these differences may be due to two study factors. First, we have added four additional years of data. Second, unlike Bosu and Sultana, we have used regression models to account for various confounding variables. Regardless, we comply with Bosu and Sultana’s hypotheses (Bosu and Sultana 2019) that gender biases are not universal and do not work in the same direction across all projects. The prevalence of gender bias may depend on various contextual factors, such as the culture of a community, its sponsor, its governance, and the size of the project.

Women are more likely to use gender-neutral profiles: Table 4 shows that the majority of women (52%) prefer GNPs. On Gerrit-based projects, more than three-quarters of women use GNPs. While the ratio of women using GNP is lower on GitHub than on Gerrit, almost half of the women on GitHub use GNPs. In comparison, almost two-thirds of men use GIPs on GitHub. Prior studies have found women using GNPs to avoid unwanted attention (Frluckaj et al. 2022; Menking et al. 2019) as well as biases (Terrell et al. 2017). While using GNP is a personal choice for a person, disproportionally higher ratios of GNP among women hinder promoting diversity and inclusion initiatives, as prior studies have found a lack of mentors and women role models as a barrier to encouraging more women joining computing (Bosu and Sultana 2019; Wang and Weinberger 2020; Lockwood 2006).

Do gender-neutral profiles help avoid biases? Terrell et al. hypothesized that GNPs may help women avoid unwanted attention as well as biases on GitHub. While we found evidence supporting this hypothesis in three Gerrit-based projects (i.e., Chromium OS, Couchbase, and Go), we also found contradictory evidence in two projects and among all four GitHub groups. Women with GNPs had between 1-5% lower PR acceptance rates than women with GIPs from the same GitHub group. On the contrary, men with GNPs had significantly higher odds of acceptance than men with GIPs in two GitHub groups (i.e., M, and L). Therefore, GNP may not be a consequential factor, especially on GitHub. However, GNPs may be helpful for women in some projects. For example, women with GNPs from the Go project.
had +16.1% higher acceptance rates than women with GIPs from the same project. On the contrary, the difference in acceptance rate between the Go project’s men with and without GNPs is -5.55%. Therefore, while GNPs are positively associated with women’s acceptance of Go, the opposite association is found for men. Regardless, GNPs do not solve underlying biases and help promote diversity and inclusion. Therefore, if a project manager finds similar differences in another project, they should investigate possible causes and their remedies.

**Gender bias exists in different directions** From the results of three research questions, we see that gender bias exists in FOSS communities, but the direction of the bias varies. In terms of code acceptance, we found seven Gerrit projects significantly favoring men, but Android, Qt, and four GitHub groups favored women. In terms of CR intervals, men are favored in three Gerrit projects, and women are favored in four Gerrit projects as well as the four GitHub groups. Finally, in terms of CR participation, men had even larger advantages, with eight Gerrit projects as well as three GitHub groups in their favor. Therefore, gender biases may manifest not only in different directions in another project but also differently across other CR aspects in the same project. *One size fits all* approach may not be the best strategy to mitigate gender biases during CRs.

**Women participate less in code review** Women encounter the most biases during CR participation, with eight Gerrit projects and three GitHub groups being against their favor. While a person may self-assign to participate in a CR, most of the CR assignments are based on invitations. Therefore, women are less likely to be invited to participate than men. Since lower CR participation from women is a trend common across almost all the projects, this observation may be explained by affinity bias—the tendency of people to collaborate with others who share similar interests, experiences, and backgrounds. Moreover, a recent study also suggests affinity biases during pull request acceptance on GitHub (Nadri et al. 2021).

We also noticed another interesting trend among several projects. Although men with GNPs from Chromium OS, Go, oVirt, GitHub (L), and GitHub (XL) are less likely to participate in CRs than men with GIPs, we see the opposite trend for women with GNPs having higher participation. This observation further supports our hypotheses regarding affinity biases. Since CRs have other benefits, such as knowledge sharing and building relationships (Bosu et al. 2016), lower CR participation puts women in disadvantageous positions compared to men counterparts. Therefore, to promote diversity and inclusion, FOSS projects should take initiatives to ensure the equitable participation of women in the CR process.

**Ratio of women increasing among FOSS projects** Since nine out of the ten projects were also included in Bosu and Sultana’s study (Bosu and Sultana 2019), we compare the ratios of women reported for each of these projects in their study against those found in this
study (Fig. 4). Eight out of the nine projects indicate higher ratios of women in April 2022 than in September 2017; Go is the only exception. Android and Chromium OS indicate the highest increment.

**Recommendations for project managers** Project managers can play an important role in fostering diversity and inclusion. Many organizations use platforms and dashboards to improve software development practices including code reviews (Hasan et al. 2021; Bird et al. 2015; Augustine et al. 2017; Guerrouj et al. 2016). These platforms can provide them with insights into code review activities and uncover potential biases. For example, if a woman is among the top contributors based on code commits but ranks way lower in terms of code review participation, or if a woman is considered an expert for a project area but is not added as a reviewer for relevant changes, those may hint at potential biases against them. Project managers should look into such cases and delve further to identify possible causes. If biases are found, a manager should take steps such as educating developers, arranging team-building activities, or creating policies to mitigate biases.

**Directions for researchers** Diversity and the impact of gender bias in FOSS projects are well researched. Researchers also suggested numerous strategies to mitigate such biases. Though gender bias exists among many projects, the direction and amplitude of that bias vary based on project size, community, and culture. Similar strategies may not work across all projects; the characteristics of biases and their underlying causes differ. While prior research has proposed various bias mitigation strategies (detailed in Section 8.3), the extent to which various contextual factors influence each strategy’s relevance as well as effectiveness lacks empirical validation.

Women are subject to not only biased feedback during code reviews (Paul et al. 2019; Imtiaz et al. 2019), but are also less likely to have equitable participation in the CR process. Results of our study suggest that CR participation is one of the most gender-inequitable aspects of most of the projects. Therefore, future work should focus on the identification of barriers to the equitable participation of women in the CR process and strategies to mitigate those barriers.

### 8 Related Works

This section briefly presents existing research on gender bias in open-source communities.

#### 8.1 Characteristics of Gender Bias in FOSS

Several prior studies have found the existence of gender bias in FOSS projects (Terrell et al. 2017; Bosu and Sultana 2019). Bosu and Sultana found that women make up only 6.70% of the non-casual developers in 10 popular Gerrit projects. While Vascilescu et al. found 9% women among approximately 873k GitHub contributors (Vasilescu et al. 2015), in a later study Terrell et al. found only 5.2% among 4 million GitHub users (Terrell et al. 2017). These studies suggest gender unbalance among FOSS communities. Wajcman shows a connection between masculinility and technology and suggests that this connection results from the historical and cultural construction of gender and produces gender bias in tech (Wajcman 2007). Biases favoring one gender over another can be explicit or implicit. Prior research shows that implicit gender biases are dominant among professional software developers and impact hiring decisions and contribution evaluation. Bosu and Sultana (2019) found explicit biases against women in the form of a lower code acceptance rate, a longer first feedback
interval, a longer code review interval, and higher scrutiny during code review. They also suggest the existence of implicit biases regarding participation in code review.

Women also encounter bias while communicating with other contributors. Nafus (2012) found that men monopolize authorship of code in the FOSS community and withdraw the necessary social ties that are necessary to build a gender-inclusive environment. Canedo et al. (2021) found that a lack of gender-equal communication impedes the participation of women. Women participants in the study of Lee and Carver (2019) reported negative interactions from males towards females. Also, women shared that men are aggressive, hold stereotypical beliefs that women are less competent, devalue women’s input, and try to humiliate women during interactions. Moreover, while investigating how FOSS contributors interact with opposite genders, they found both positive and negative responses. Negative responses express ideas related to stereotypes about women, like women getting easily offended or creating drama. One respondent shared that "men are usually more used to being rejected, so there is less drama to be afraid of" when interacting with women contributors. On the contrary, female contributors shared that they were required to prove themselves repeatedly or were made to feel less comfortable around men. As contemporary software development organizations are dominated by males, occurrences of misogynistic and sexist remarks are abundant in many communities (Squire and Gazda 2015).

Along with communities and organizations, software products or tools can be biased toward one group of people while neglecting others (Burnett et al. 2017). Gender inclusiveness issues have also been found in software tools. Researchers have developed a tool, Gender-Mag (Burnett et al. 2016) to identify gender bias in software tools. Mendez et al. (2018a) found gender issues in from 7%-71% of the use cases for tools and infrastructure used in OSS projects and showed that because of biased infrastructure, women newcomers are more disadvantaged than men. It is proven that tools are a contributing factor in gender disparity in the open-source community (Mendez et al. 2018a,b) so it is high time we focus on building gender neutral tools for software development.

8.2 Consequences of Gender Bias

Vandana et al. Singh and Bongiovanni (2021) investigated the situation of women working in OSS communities and shared that due to the very low representation of women, they feel isolated and burdened to represent all women who are being excluded. Moreover, women often feel their contributions are not valued enough because of their gender, and they have to work harder to get recognition. Surprisingly, one participant in her study shared that women are expected to write better code than men because they have to be outstanding to reach a higher level of projects, whereas men with average skillsets can make it to those positions. Nasif and his colleagues (Imtiaz et al. 2019) conducted a study to investigate the effect of Prove-it-again where a member of a non-stereotypical group has to meet stricter standards to prove his/her competence. They did not find that women’s pull requests take longer, in general, to get accepted, but women concentrate their contributions on fewer projects and fewer organizations than men.

According to Edna et al. (Canedo et al. 2021), lack of gender-equal communication hinders the participation of women in OSS communities. In this case, some men do not interact or talk with their female colleagues equally. To express their opinions, women have to interrupt others, even if they are project managers. Paul et al. (2019) researched sentiment during code reviews for 6 popular OSS projects (Android, Chromium OS, Couchbase, OmapZoom, Ovirt, Qt) and found that in three of these projects, women receive negative reviews more
often. Moreover, men withhold positive encouragement while reviewing women’s codes. Women tend to express their sentiments less than men and also are more neutral to men counterparts than other women contributors. Additionally, the interaction between men and women was found to be flirtatious, whereas woman-to-woman communication was found supportive (Thelwall et al. 2010) in MySpace comments. Nasif and his colleagues (Imtiaz et al. 2019) also investigated if women have a narrow band of socially acceptable behavior while contributing to OSS projects. They shared that women avoid using profane words more than men; they are more neutral when showing sentiments or stereotypically masculine or feminine traits.

8.3 Strategies to Mitigate Bias

Researchers also mentioned multiple initiatives, e.g., promoting women to leadership roles, adopting codes of conduct (CoC), maintaining a transparent organizational culture, promoting gender-inclusive language, de-biasing tools, and so on, that can be taken to mitigate bias in the OSS platform and increase the participation of women. Canedo et al. (2020) and Prana et al. (2021) suggest assigning women to senior roles to lead them to empowerment. Leadership increases the visibility of their accomplishments, shows respect, and encourages others to join the team (Singh 2019). Catolino et al. (2019) show in their study that the presence of women in the community reduces community smells that are sub-optimal patterns in the culture of software organizations by mediating discussions and increasing communications among members. By avoiding the segregation of tasks as masculine and feminine, treating women developers as developers, we can dismantle stereotypical views and create a gender-inclusive environment (Calvo 2020; Vedres and Vasarhelyi 2019; Canedo et al. 2020). Participants in Vandana’s study (Singh 2019) suggested including women in different conferences and activities instead of isolating them. Singh mentions letting women lead where it is appropriate (Singh 2019).

The CoC should enlist norms of unacceptable behavior in the community and the consequences of violations of those codes. Singh et al. (2019) investigated websites for 355 projects in OSS; only 28 of them had Codes of Conduct or similar guidelines. Prior studies show that introducing a code of conduct can reduce tightrope effects and foster a culture of collaboration (Imtiaz et al. 2019; Singh 2019; Prana et al. 2021; Calvo 2020; Singh and Brandon 2019). A study conducted by Robson (Robson 2018) shows that it is not enough only to introduce the code of conduct in the community; proper enforcement is also required to make the environment inclusive for all (Singh 2019; Tourani et al. 2017). Moderators should invigilate to see if the rules are being followed, and punishment must be ensured if necessary (Forte et al. 2012). If organizations that host the open source project remain transparent about their work environment and culture, women can decide beforehand whether to make a contribution. It can increase the level of affinity between personal and professional values (Qiu et al. 2010).

Women newcomers to OSS face disproportionately more barriers than others. Among different types of biases, gender bias in tools and infrastructures used in open-source media disadvantages women (Mendez et al. 2018a). Researchers have developed a tool: GenderMag that can be used to de-bias tools. GenderMag uses gendered personas and a specialized Cognitive Walkthrough (CW) to evaluate software and identify if there is any gender-inclusiveness issue with the tool for a wide range of people, from designers to software developers (Mendez et al. 2018a,b; Padala et al. 2020). Researchers have also mentioned

7 https://gendermag.org/
a few humanitarian open-source projects where women might be interested in Parra et al. (2016). Women developers can start contributing to these projects and gain confidence and experience. Such types of projects can be launched in greater numbers to attract more women to OSS. In a nutshell, researchers suggested initiating projects that are specifically preferred by women. Qiu et al. (2010) investigated the signals that contributors notice before starting to work on a project. Few participants shared that the openness of the community or the language used by the contributors played an important role in deciding to contribute to a project. One participant in their study found that one project included “nice guys” in their documentation. OSS contributors also raised concern about gender-inclusive language sharing, saying the project should avoid words like “guys” or gendered pronouns, assuming contributors are mostly men or hold one demographic background (Canedo et al. 2020; Qiu et al. 2010). Nafus (2012) also indicated the use of inadequate language in OSS. So, it is mandatory to use gender-inclusive language in documentation or during communication with each other.

9 Validity Threats

**Internal validity** We examined our hypotheses for a total of 1010 FOSS projects, where 1000 projects are using pull-request-based code reviews and the remaining 10 are using Gerrit-based code reviews. Project selection is the primary threat to the internal validity of our study. Since we partially replicated the studies of Bosu and Sultana’s (2019) study, we adopted their project list for the Gerrit dataset. As one of those projects, OmapZoom, was inactive, we replaced it with another project, Wikimedia that fits their project selection criteria. We also updated their dataset with code reviews until April 30, 2022, so that we could get the most recent status of gender bias.

Furthermore, we are replicating the study of Terrell et al. (2017). As their dataset is not publicly available, we selected 1000 projects from GHTorrent exported in March 2021. Being an open-source platform, GitHub hosts different types of personal and group projects that may or may not be based on programming languages. Following Kalliamvakou et al.’s recommendation, Kalliamvakou et al. (2016), we selected only those projects that use one of the popular programming languages, have at least 20 contributors, and have had 20 pull requests during the last three months. We followed the criteria to avoid any discontinued or too small projects that may not add any valuable information to our study. Moreover, we selected GitHub projects by stratified sampling so that we could have insights about projects of different sizes and conducted data analysis separately. Despite our carefully designed sampling strategy, the characteristics of our sample may differ from the entire GitHub ecosystem.

The GHTorrent 2019 dataset had a data hole from July to December 2019. Though GHTorrent did not confirm if the issue has been fixed GHTorrent MySQL export from March 2021, we found pull requests created during that period in this dataset. However, we cannot confirm if all the pull requests during that period are included in our analyses.

**External validity** We can observe from the results that the direction and degree of bias differ for different projects. Results might be different if we replicate the study in different settings or on different projects. The scenario for communication and collaboration among developers may not be the same for other projects in FOSS communities. A project’s culture also depends on various factors, such as the nature and size of the project, the number of contributors, and project governance. While we have used data from a large number of projects in this study, our sample may not adequately represent the entire FOSS spectrum.
**Construct validity** One major construct that poses a threat to the validity of our study is the gender resolution of the users. First, we did not attempt to identify non-binary genders, since it is almost impossible to identify those developers without their input. Second, for the persons with GNPs, we assign GenderComputer’s resolution. While GenderComputer has been used in several prior SE studies, including the two original ones, name-to-gender resolution tools are prone to misclassifications (Santamaría and Mihaljević 2018). Therefore, between 5-15% of the 85,772 persons with GNPs are miscategorized in our dataset. Finally, for the users with GIPs, we assigned genders based on their avatars. If we identified a conflict between the gender assigned by GenderComputer and our avatar-based assignment, we manually checked the profile and searched on social media to make a resolution. However, if GenderComputer assigns either ‘Unisex’ or ‘None’ and a user is using an avatar suggestive of a different gender, he/she would be mislabeled. While we believe such cases would be negligible, such misclassifications are another threat to our gender resolution strategy.

We used the Python library *lizard* to calculate source code metrics, which have been starred more than a thousand times on GitHub and have been forked more than 200 times. While we have not evaluated its accuracy, it may be subject to some errors.

**Conclusion validity** We have regression-based models for analysis to account for several confounding variables. We have used well-known and mature libraries such as *stats*, *rms* to train our models. To estimate effects, we have used standard metrics such as the odds ratio for logistic regression models and the co-efficient (Z) for the linear models. Therefore, it is unlikely that any threats to validity will arise from the evaluation metrics, library selection, or evaluation of the dependent variables.

10 Conclusion

Gender bias in FOSS communities is well-researched. But how this bias varies for different aspects of OSS project development has not been studied. To bridge this gap, we studied the existence of bias in both explicit (pull request acceptance, code review interval) and implicit bias (code review participation). We found that gender bias exists in different directions. The level of bias in the above-mentioned aspects differs based on the size of the projects, the community, and the culture of the projects. For GitHub projects, women have a higher pull request acceptance rate and more code review participation (except for extra large projects), but a longer code review interval. The gender of the contributors has a mixed impact on Gerrit projects. So, a single strategy to mitigate bias in all projects might not be fruitful. We also investigated the effect of using gender-neutral profiles on the three aspects and provided recommendations based on our results. Prospective newcomers in FOSS communities, project managers, and researchers may gain some insights about the status of gender bias from our study and take appropriate steps to avoid biased occurrences and create a gender-inclusive environment.

11 Disclaimer

Our study primarily follows the proposed design published in the Registered Report (Sultana and Bosu 2021), but there are a few deviations from the original study. We have listed those below:
We have created both datasets based on more recent data. Originally, we proposed to use the GHTorrent MySQL dump from June 2019. Instead, we have used the GHTorrent data dump from March 2021. Also, for the Gerrit dataset, we have updated Bosu et al.’s dataset to include all CRs completed through April 30th 2022, to get the most recent picture of gender bias on these platforms. We have also identified bot accounts for exclusion.

A criterion for selecting GitHub projects was changed from at least 10 contributors to ‘at least 20 contributors’, since we found that a random selection with lower thresholds increases selection space by orders of magnitude and returns projects that have low development activities.

We primarily planned to identify the gender of a user from the GenderComputer tool, a GitHub/Gerrit profile, a social network (LinkedIn or Facebook), and the contact search feature on Gmail. However, since the GitHub profile avatar is public and gender can be identified from those avatars using face recognition algorithms and human raters, we added this step: identification of gender from avatars, to be more confident in gender identification.

There is a change in the computation of the gender-neutral profile attribute. We proposed in the study design that we would consider a profile gender-neutral if the gender of the user is not identifiable from the avatar, display name, full name, or human ratings. But, we considered profiles gender-neutral if the gender of the profile owner was not identifiable from the avatar. We discarded inferring gender neutrality from a name or using a human rater since it requires familiarity with the origin of the name. For example, most people from the Indian subcontinent may easily infer the name ‘Sayma’ belongs to a woman. However, people of different origins may have difficulties. Moreover, even though a name-to-gender inference tool can reliably guess gender from a name, CR participants are less likely to use such tools during their collaborations. Therefore, to avoid such limitations, we considered only avatars as sources to identify gendered profiles, i.e., we marked a profile as GIP only if we could reliably identify its owner’s gender using the associated avatar.

Moreover, we proposed that we would estimate the power of each independent variable using Wald Chi-Square statistics ($\chi^2$). During study execution, we used ‘Odds Ratio’ (OR) for the logistic regression models and regression coefficients for the linear regression models instead. There are benefits from both ends in using one over the other. Wald $\chi^2$ shows the influence of an independent variable over the trained model, (i.e., how much explanatory power a model would lose if we removed this variable. On the other hand, OR shows how an independent variable’s change influences the outcome variable. Specifically, for dichotomous variables, OR indicates how the odds of an outcome change if the factor is present in contrast to its absence. Since our variables of interest (i.e., gender and isGenderNeutral) are dichotomous, we think using OR has clear advantages in terms of interpreting the results. Similarly, for linear regression models, the coefficient shows how the dependent variable changes if the independent variable changes by one unit. Again, due to the dichotomous nature of our factors such as gender, the regression coefficient shows how the dependent changes if we change gender from ‘0’ - ‘man’ to ‘1’ -‘woman’. Therefore, our modified choices have clear advantages over our prior selection in terms of interpretability.

**Funding** Work conducted by for this research is partially supported by the US National Science Foundation under Grant No. 1850475. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.
Data availability  Our data mining and analysis scripts and aggregated dataset are publicly available at https://doi.org/10.5281/zenodo.760853. Due to privacy concerns, we do not make the full dataset, which is more than 1TB publicly available. The authors would be happy to share the dataset with researchers upon contact.

Declarations

Conflicts of interests/Competing interests  The authors have no competing interests to declare that are relevant to the content of this article.

References

Alba B (2018) To achieve gender equality, we must first tackle our unconscious biases. http://theconversation.com/to-achieve-gender-equality-we-must-first-tackle-our-unconscious-biases-92848
Augustine V, Hudopoli J, Marcinczak P, Snipes W (2017) Deploying software team analytics in a multinational organization. IEEE Softw 35(1):72–76
Bacchelli A, Bird C (2013) Expectations, outcomes, and challenges of modern code review. In: 2013 35th International Conference on Software Engineering (ICSE), IEEE, pp 712–721
Barnett M, Bird C, Brunet J, Lahiri SK (2015) Helping developers help themselves: Automatic decomposition of code review changesets. In: 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, vol 1. IEEE, pp 134–144
Beard C (2018) Diversity and inclusion at mozilla. https://blog.mozilla.org/careers/diversity-and-inclusion-at-mozilla/. Accessed 2023/04/01
Beneschott B (2014) Is open source open to women? https://www.toptal.com/open-source/is-open-source-open-to-women. Accessed 2023/04/01
Bertagnoli L (2021) How tech can get more women into software engineering. https://builtin.com/software-engineering-perspectives/women-in-engineering. Accessed 2023/04/01
Bird C, Carnahan T, Greiler M (2015) Lessons learned from building and deploying a code review analytics platform. In: 2015 IEEE/ACM 12th Working Conference on Mining Software Repositories, IEEE, pp 191–201
Boston University SoPH (2016) Correlation and regression with R. https://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/R/R5_Correlation-Regression/R5_Correlation-Regression4.html. Accessed 29 May 2021
Bosu A, Carver JC (2014) Impact of developer reputation on code review outcomes in oss projects: An empirical investigation. In: Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, Association for Computing Machinery, New York, NY, USA, ESEM ’14. https://doi.org/10.1145/2652524.2652544
Bosu A, Sultana KZ (2019) Diversity and inclusion in open source software (oss) projects: Where do we stand? In: Proceedings of the 2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)
Bosu A, Carver JC, Hafiz M, Hilley P, Janni D (2014) Identifying the characteristics of vulnerable code changes: An empirical study. 22nd ACM SIGSOFT International Symposium on the Foundations of Software Engineering, China, Hong Kong, pp 257–268
Bosu A, Carver JC, Bird C, Orbeck J, Chockley C (2016) Process aspects and social dynamics of contemporary code review: Insights from open source development and industrial practice at microsoft. IEEE Trans Softw Eng 43(1):56–75
Bourke J (2017) Diversity and inclusion: The reality gap. https://www2.deloitte.com/us/en/insights/focus/human-capital-trends/2017/diversity-and-inclusion-at-the-workplace.html. Accessed 29 June 2022
Built-in (2021) Diversity + inclusion:What is the meaning of diversity & inclusion? a 2021 workplace guide. https://builtin.com/diversity-inclusion. Accessed 29 May 2021
Burnett M, Stumpf S, Macbeth J, Makri S, Beckwith L., Kwan I, Peters A, Kernigan W (2016) Gendermag: A method for evaluating software’s gender inclusiveness. Interact Comput 28(6):760–787
Burnett M, Counts R, Lawrence R, Hanson H (2017) Gender hl and microsoft: Highlights from a longitudinal study. In: 2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), pp 139–143. https://doi.org/10.1109/VLHCC.2017.8103461
Calvo D (2020) The (in)visible barriers of free software: Inequalities of online communities in spain. Stud Commun Sci 21. https://doi.org/10.24434/j.scoms.2021.01.011
Empirical Software Engineering

Canedo E, Bonifacio R, Okimoto M, Serebrenik A, Pinto G, Monteiro E (2020) Work practices and perceptions from women core developers in OSS communities. In: Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), pp 1–11

Canedo ED, Mendes F, Cerqueira A, Okimoto M, Pinto G, Bonifacio R (2021) Breaking one barrier at a time: How women developers cope in a men-dominated industry. In: Brazilian Symposium on Software Engineering, Association for Computing Machinery, New York, NY, USA, SBES ’21, p 378–387. https://doi.org/10.1145/3474624.3474638

Catolino G, Palomba F, Tamburri D, Serebrenik A, Ferrucci F (2019) Gender diversity and women in software teams: how do they affect community smells? In: Proceedings - 2019 IEEE/ACM 41st International Conference on Software Engineering, ACM/IEEE, pp 11–20. https://doi.org/10.1109/ICSE-SEIS.2019.00010. https://2019.icse-conferences.org/home

Ciceri F (2021) Diversity statement. https://www.debian.org/intro/diversity. Accessed 2023/04/01

David PA, Shapiro JS (2008) Community-based production of open-source software: What do we know about the developers who participate? Inf Econ Policy 20(4):364–398. https://doi.org/10.1016/j.infoecopol.2008.10.001. https://2019.icse-conferences.org/home

Durrleman S, Simon R (1989) Flexible regression models with cubic splines. Stat Med 8(5):551–561

Eidinger E, Enbar R, Hassner T (2014) Age and gender estimation of unfiltered faces. IEEE Trans Inf Forensic Secur 9(12):2170–2179

Fan Y, Xia X, Lo D, Li S (2018) Early prediction of merged code changes to prioritize reviewing tasks. Empir Softw Eng 23(6):3346–3393

Forte A, Antin J, Bardzell S, Honeywell L, Riedl J, Stierch S (2012) Some of all human knowledge: Gender and participation in peer production. In: Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work Companion, Association for Computing Machinery, New York, NY, USA, CSCW ’12, p 33–36. https://doi.org/10.1145/2141512.2141530

Foundation (2008) Geek feminism wiki. http://geekfeminism.wikia.com/wiki/FLOSS#Discussion_of_issues. Accessed 2023/04/01

Frluckaj H, Dabbish L, Widder DG, Qiu HS, Herbsleb J (2022) Gender and participation in open source software development. Proc ACM Hum-Comput Interaction 6(CSCW2):1–31

Fulton LV, Mendez FA, Bastian ND, Musal RM (2012) Confusion between odds and probability, a pandemic? J Stat Educ 20(3)

Ghosh RA (2005) Understanding free software developers: Findings from the FLOSS study. Perspect Free Open Source Softw 28:23–47

Gousios G (2013) The ghtorrent dataset and tool suite. In: Proceedings of the 10th Working Conference on Mining Software Repositories, IEEE Press, Piscataway, NJ, USA, MSR ’13, pp 233–236. http://dl.acm.org/citation.cfm?id=2487085.2487132. Accessed 2023/04/01

Gousios G, Pinzger M, Deursen Av (2014) An exploratory study of the pull-based software development model. In: Proceedings of the 36th International Conference on Software Engineering, Association for Computing Machinery, New York, NY, USA, ICSE’2014, p 345–355. https://doi.org/10.1145/2568225.2568260

Goyal K, Agarwal K, Kumar R (2017) Face detection and tracking: Using opencv. In: 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), vol 1. IEEE, pp 474–478

Guerrouj L, Baysal O, Lo D, Khomh F (2016) Software analytics: challenges and opportunities. In: Proceedings of the 38th International Conference on Software Engineering Companion, pp 902–903

Harrell F (2015) Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis. Springer Series in Statistics, Springer International Publishing. https://books.google.com/books?id=94RgCgAAQBAJ. Accessed 2023/04/01

Harrell F, Lee K, Califf R, Pryor D, Rosati R (1984) Regression modelling strategies for improved prognostic prediction. Stat Med 3(2):143–152. https://doi.org/10.1002/sim.4780030207

Harrell F, Lee K, Matchar D, Reichert T (1985) Harrell jr fe, lee kl, matchar db, reichert taregression models for prognostic prediction: advantages, problems, and suggested solutions. cancer treat rep 69: 1071–1077. Cancer Treat Rep 69:1071–77

Hasan M, Iqbal A, Islam MRU, Rahman AI, Bosu A (2021) Using a balanced scorecard to identify open source developers to improve code review effectiveness: An industrial experience report. Empir Softw Eng 26:1–34

Imtiaz N, Middleton J, Chakraborty J, Robson N, Bai G, Murphy-Hill E (2019). Investigating the effects of gender bias on github. https://doi.org/10.1109/ICSE.2019.00079

Jeong G, Kim S, Zimmermann T, Yi K (2009) Improving code review by predicting reviewers and acceptance of patches. Res Softw Anal Error-Free Comput Center Tech-Memo (ROSAEC MEMO 2009-006) 1:1–18

Jiang Y, Adams B, German DM (2013) Will my patch make it? and how fast? case study on the linux kernel. In: 2013 10th Working Conference on Mining Software Repositories (MSR), IEEE, pp 101–110
Kalliamvakou E, Gousios G, Blincoe K, Singer L, German DM, Damian D (2016) An in-depth study of the promises and perils of mining github. Empir Softw Eng 21(5):2035–2071
Krieger B, Leach J, Nafus D (2006) Gender integrated report of findings. European Union Sixth Framework Programme, Free/Libre/Open Source Software: Policy Support I(1)
Kononenko O, Baysal O, Guerrouj L, Cao Y (2015) Investigating code review quality: Do people and participation matter? pp 111–120. https://doi.org/10.1109/ICSM.2015.7332457
Laura Sherbin RR (2017) Diversity doesn’t stick without inclusion. https://www.vernamyers.com/2017/02/04/diversity-doesnt-stick-without-inclusion/. Accessed 2023/04/01
Lee A, Carver JC (2019) Floss participants’ perceptions about gender and inclusiveness: A survey. In: 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE), pp 677–687. https://doi.org/10.1109/ICSE.2019.00077
Lenarduzzi V, Nikkola V, Saarimäki N, Taibi D (2021) Does code quality affect pull request acceptance? an empirical study. J Syst Softw 171:110806
Lin B, Serebrenik A (2016) Recognizing gender of stack overflow users. In: Proceedings of the 13th International Conference on Mining Software Repositories, pp 425–429
Lockwood P (2006) Someone like me can be successful: Do college students need same-gender role models? Psychol Women Q 30(1):36–46
Mansfield ER, Helms BP (1982) Detecting multicollinearity. Am Stat 36(3a):158–160
McIntosh S, Kamey Y, Adams B, Hassan AE (2015) An empirical study of the impact of modern code review practices on software quality. Empir Softw Eng 21. https://doi.org/10.1007/s10664-015-9381-9
Mendez C, Padala HS, Steine-Hanson Z, Hilderbrand C, Horvath A, Hill C, Simpson L, Patil N, Sarma A, Burnett M (2018a) Open source barriers to entry, revisited: A sociotechnical perspective. In: Proceedings of the 40th International Conference on Software Engineering, Association for Computing Machinery, New York, NY, USA, ICSE ’18, p 1004–1015. https://doi.org/10.1145/3180155.3180241
Mendez C, Sarma A, Burnett M (2018b) Gender in open source software: what the tools tell. pp 21–24. https://doi.org/10.1145/3195570.3195572
Menking A, Erickson I, Pratt W (2019) People who can take it: how women wikipedians negotiate and navigate safety. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp 1–14
Mirsaeedi E, Rigby PC (2020) Mitigating turnover with code review recommendation: balancing expertise, workload, and knowledge distribution. In: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering, pp 1183–1195
Nadri R, Rodríguez-Pérez G, Nagappan M (2021) On the relationship between the developer’s perceptible race and ethnicity and the evaluation of contributions in oss. IEEE Trans Softw Eng 48(8):2955–2968
Nafus D (2012) Patches don’t have gender: What is not open in open source software. New Media Soc 14(4):669–683. https://doi.org/10.1177/1461444811422887
Padala SH, Mendez CJ, Dias LF, Steinmacher I, Steine Hanson Z, Hilderbrand C, Horvath A, Hill C, Simpson LD, Burnett M, Gerosa M, Sarma A (2020) How gender-biased tools shape newcomer experiences in oss projects. IEEE Trans Softw Eng 1. https://doi.org/10.1109/TSE.2020.2984173
Parra E, Haiduc S, James R (2016) Making a difference: An overview of humanitarian free open source systems. In: 2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C), pp 731–733
Paul R, Bosu A, Sultana KZ (2019) Expressions of sentiments during code reviews: Male vs. female. pp 26–37. https://doi.org/10.1109/2019.8667987
Paul R, Türrko AK, Bosu A (2021) Why security defects go unnoticed during code reviews? a case-control study of the chromium os project. In: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), pp 1373–1385
Pourhoseingholi MA, Baghestani AR, Vahedi M (2012) How to control confounding effects by statistical analysis. Gastroenterol Hepatol Bed Bench 5(2):79
Prana GAA, Ford D, Rastogi A, Lo D, Purandare R, Nagappan N (2021) Including everyone, everywhere: Understanding opportunities and challenges of geographic gender-inclusion in oss. pp 1. https://doi.org/10.1109/TSE.2021.3092813
Qiu Y, Stewart KJ, Bartol KM (2010) Joining and socialization in open source women’s groups: An exploratory study of kde-women. In: Ågerfalk P, Boldyreff C, González-Barahona JM, Madye GR, Noll J (eds) Open Source Software: New Horizons. Springer, Berlin Heidelberg, Berlin, Heidelberg, pp 239–251
Rigby P, German D, Storey MA (2008) Open source software peer review practices. In: 2008 ACM/IEEE 30th International Conference on Software Engineering, pp 541–550. https://doi.org/10.1109/ICSE.2008.1368162
Robson N (2018) Diversity and decorum in open source communities. In: Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2018, pp 986–987
Santamaría L, Mihaljević H (2018) Comparison and benchmark of name-to-gender inference services. PeerJ Comput Sci 4:e156
Santos A, Vegas S, Oivo M, Juristo N (2019) A procedure and guidelines for analyzing groups of software engineering replications. IEEE Trans Softw Eng PP:1. https://doi.org/10.1109/TSE.2019.2935720
Shull FJ, Carver JC, Vegas S, Juristo N (2008) The role of replications in empirical software engineering. Empir Softw Eng 13(2):211–218
Singh V (2019) Women participation in open source software communities. In: Proceedings of the 13th European Conference on Software Architecture - Volume 2, Association for Computing Machinery, New York, NY, USA, ECSA'19, pp 94–99. https://doi.org/10.1145/3344948.3344968
Singh V, Brandon W (2019) Open Source Software Community Inclusion Initiatives to Support Women Participation, pp 68–79. https://doi.org/10.1007/978-3-030-20883-7_7
Singh V, Bongiovanni B (2021) Motivated and capable but no space for error women’s experiences of contributing to open source software. In: The international Journal of information, diversity and inclusion, vol 5
Smith TJ, McKenna CM (2013) A comparison of logistic regression pseudo r2 indices. Mult Linear Regression Viewpoints 39(2):17–26
Squire M, Gazda R (2015) Floss as a source for profanity and insults: Collecting the data. In: 2015 48th Hawaii International Conference on System Sciences, IEEE, pp 5290–5298
Sultana S, Bosu A (2021) Are code review processes influenced by the genders of the participants? https://doi.org/10.48550/ARXIV.2108.07774. https://arxiv.org/abs/2108.07774
Sultana S, Turzo AK, Bosu A (2023) Replication package for Code Reviews in Open Source Projects?: How Do Gender Biases Affect Participation and Outcomes? Zeonodo. https://doi.org/10.5281/zenodo.7608539
Tao Y, Han D, Kim S (2014) Writing acceptable patches: An empirical study of open source project patches. In: 2014 IEEE International Conference on Software Maintenance and Evolution, IEEE, pp 271–280
Team FD (2019) Diversity and inclusion in fedora. https://docs.fedoraproject.org/en-US/diversity-inclusion/. Accessed 2023/04/01
Terrell J, Kofink A, Middleton J, Raineer C, Murphy-Hill E, Parnin C, Stallings J (2017) Gender differences and bias in open source: Pull request acceptance of women versus men. PeerJ Comput Sci 3:e111
Thelwall M, Wilkinson D, Uppal S (2010) Data mining emotion in social network communication: Gender differences in myspace. J Am Soc Inf Sci Technol 61(1):190–199
Thongtanunam P, Tantithamthavorn C, Kula RG, Yoshida N, Iida H, Matsumoto Ki (2015) Who should review my code?: a file location-based code-reviewer recommendation approach for modern code review. In: 2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER), IEEE, pp 141–150
Thongtanunam P, McIntosh S, Hassan A, Iida H (2016) Review participation in modern code review. Empir Softw Eng 22:768–817
Tourani P, Adams B, Serebrenik A (2017) Code of conduct in open source projects. In: 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER), pp 24–33. https://doi.org/10.1109/SANER.2017.7884606
Vasilescu B, Capipluppi A, Serebrenik A (2014) Gender, representation and online participation?: a quantitative study. Interact Comput 26(5):488–511
Vasilescu B, Posnett D, Ray B, van den Brand MG, Serebrenik A, Devanbu P, Filkov V (2015) Gender and tenure diversity in github teams. In: Proceedings of the 33rd annual ACM conference on human factors in computing systems, pp 3789–3798
Veall MR, Zimmermann KF (1994) Evaluating pseudo-r2’s for binary probit models. Qual Quant 28(2):151–164
Vedres B, Vasarhelyi O (2019) Gendered behavior as a disadvantage in open source software development. EPJ Data Sci 8(1):25
Wajcman J (2007) From women and technology to gendered technoscience. Inf Community Soc 10(3):287–298
Wang L, Weinberger K (2020) Reasons for lack of diversity in open source: The case Katie Bouman. Free and Open Technologies
Xia X, Lo D, Wang X, Yang X (2015) Who should review this change?: Putting text and file location analyses together for more accurate recommendations. In: 2015 IEEE International Conference on Software Maintenance and Evolution (ICSME), IEEE, pp 261–270
Yin P, Fan X (2001) Estimating r 2 shrinkage in multiple regression: A comparison of different analytical methods. J Exp Educ 69(2):203–224
Zafar S, Malik MZ, Walia GS (2019) Towards standardizing and improving classification of bug-fix commits. In: 2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), IEEE, pp 1–6
Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Sayma Sultana is a Computer Science Ph.D. student at Wayne State University. She received her BS in Computer Science from the Bangladesh University of Engineering & Technology in 2017. Her primary research interests lie in identifying and developing interventions to reduce gender bias among software development organizations. Her works also span the area of code review, social network analysis, software security, and empirical software engineering.

Asif Kamal Turzo is a Ph.D. Candidate of Computer Science at Wayne State University. His research interests include Empirical Software Engineering, Software Security, and Code Review. He received his B.S. in Computer Science and Engineering from the Bangladesh University of Engineering and Technology (BUET). Before joining Wayne State University, he worked as a Software Engineer in multiple software development organizations in Bangladesh. He has received several national and international awards for Mobile Application Development projects.

Amiangshu Bosu is an Assistant Professor of Computer Science at the Wayne State University. He completed his Ph.D. dissertation work at the University of Alabama in 2015. His research spans empirical software engineering, peer code review, software security, diversity and inclusion, mining software repositories, and social network analysis. His research appears in prestigious conferences and journals in the areas of software engineering such as ICSE, FSE, EMSE, TSE, TOSEM, ASE, JSS, ESEM, SANER, and MSR.