Systemic Gender Inequities in Who Reviews Code

EMERSON MURPHY-HILL, Google, United States
JILLIAN DICKER, Google, United States
AMBER HORVATH, Carnegie Mellon University, United States
MAGGIE MORROW HODGES, Google, United States
CAROLYN D. EGELMAN, Google, United States
LAURIE R. WEINGART, Carnegie Mellon University, United States
CIERA JASPAN, Google, United States
COLLIN GREEN, Google, United States
NINA CHEN, Google, United States

Code review is an essential task for modern software engineers, where the author of a code change assigns other engineers the task of providing feedback on the author’s code. In this paper, we investigate the task of code review through the lens of equity, the proposition that engineers should share reviewing responsibilities fairly. Through this lens, we quantitatively examine gender inequities in code review load at Google. We found that, on average, women perform about 25% fewer reviews than men,¹ an inequity with multiple systemic antecedents, including authors’ tendency to choose men as reviewers, a recommender system’s amplification of human biases, and gender differences in how reviewer credentials are assigned and earned. Although substantial work remains to close the review load gap, we show how one small change has begun to do so.

CCS Concepts:
• Human-centered computing → Computer supported cooperative work; Empirical studies in collaborative and social computing.

Additional Key Words and Phrases: code review; equity

ACM Reference Format:

1 INTRODUCTION

Code review is a common software development practice for building and maintaining modern software systems. In communities and companies that practice code review, a software engineer proposes a concrete bug fix or feature implementation in the form of a code change, often called a pull request, patch, or changelist. As the author of that changelist, that software engineer assigns one or more other engineers the task of providing feedback on the changelist. Once the feedback

¹In this paper, we used preexisting gender data that contains only male and female gender labels, which may not reflect the gender identity of all individuals because it has insufficient support for trans individuals, and those of additional genders.

© 2023 Copyright held by the owner/author(s).
2573-0142/2023/4-ART94
https://doi.org/10.1145/3579527

is addressed, the author then submits that code into a shared code repository. Prior research has documented code review at various companies [5, 20] and in open source software development [62].

Google, a large multinational software company, is one such organization that regularly practices code review. Code review at Google was initially introduced, in part, for the purpose of knowledge transfer, to ensure "that more than one person would be familiar with each piece of code" [50]. Microsoft and open source engineers have also reported that knowledge transfer is "one of the primary purposes of code review" [12]. Based on surveys, Bosu and colleagues find that knowledge sharing is important because it can "help both authors and reviewers learn how to solve problems using new approaches" and "help socialize project details, e.g. architecture, common APIs, and existing libraries" [12].

Given the importance of knowledge transfer, we would expect a diverse range of engineers to have equitable opportunities to participate as code reviewers. Such equity would help ensure that code authors would benefit from reviewers’ knowledge and vice-versa. However, as we show in Section 2, there’s reason to believe that some groups are more likely to be asked to review than others.

In this paper, we examine code review using an equity lens, that is, that engineers should equitably share the load of performing code review as a mechanism to share knowledge. We are motivated by the management literature, which suggests negative consequences of inequitable knowledge sharing: for individuals, it decreases productivity, lowers work quality and signals lack of competence [26], and for organizations, it decreases performance and innovation [58]. In this paper we make the following contributions:

- A unique examination of code review load inequities in a major software company, demonstrating a 25% difference (16.8% when adjusting for confounding factors) between men’s and women’s code review loads (Section 4).
- A holistic examination of the antecedents of this inequity, including gender differences in how reviewer credentials, such as ownership, are assigned and earned (Section 5); changelist authors’ tendency to manually choose men as reviewers (Section 6.2); and a recommender system’s amplification of human biases (Section 6.5).
- An evaluation of an intervention designed to reduce code review load inequities (Section 7).

In the remainder of this paper, we will explore these contributions at a single company: Google. As we go, we will explain the particularities of code review at Google, and in Section 9, we will zoom out and explain how our results may generalize to other organizations that use code review.

2 RELATED WORK

2.1 Workplace Tasks and Gender

Demographic factors, such as gender and race, have been shown to influence the distribution of work among employees. Across a variety of jobs, industries, and workplaces, men are more likely to be engaged in more challenging, desirable, and visible work than are women. In contrast, women are more often tasked with work that supports the work of others, rather than advances their own careers. This phenomenon has been documented across a variety of professions, including law [60], investment banking [48], engineering [59], and academia [25], as well as among TSA agents [17] and grocery store clerks [56]. For example, white male engineers reported being assigned higher-profile work and having greater access to desirable assignments than their similarly qualified female and Black, Indigenous, and people of color (BIPOC) colleagues, and they were less likely to take on “office housework” like planning parties, taking notes, and organizing meetings [59]. Women and people of color have been shown to spend more time on service tasks that do not advance their
careers [57] and are often tapped to do more institutional service, especially related to diversity, equity, and inclusion efforts of their organizations [19, 23, 31].

These differences in the distribution of promotable versus non-promotable tasks can have serious implications for women and their organizations [3]. Tasks that are more promotable increase the likelihood of advancement, either directly or indirectly. Tasks that are highly promotable are often measured and included in performance evaluations. Tasks that are indirectly promotable may serve to increase an employee’s skill sets or status in the organization or profession in the future, but do not have an immediate payoff. Finally, tasks that are non-promotable (NPTs) have little to no payoff now or in the future, even though they may be important to the functioning of the organization. Research shows that a heavy load of NPTs interferes with time needed to promotable work [57] and lead to work overload, stress, and exhaustion when people try to do it all [37, 51].

Research suggests that women are more likely to perform NPTs than men because people expect them to [4]. These expectations result in women, more than men, volunteering, being asked to volunteer, and accepting requests to volunteer to perform NPTs to both fulfill those expectations and to avoid backlash if they do not. In contrast, men are more likely to be assigned more challenging, promotable tasks, including indirectly-promotable tasks that develop their skill sets and improve visibility, because people hold stereotyped perceptions of men being more skilled than women and therefore consider them a better “fit” for jobs that are viewed as higher status, and more competitive and challenging [29, 52].

This literature suggests that the direction of gender inequity in code review, if it were to exist, would depend on whether the task were promotable or non-promotable. In a preliminary survey we conducted with 178 software engineers at Google, when asked to sort 75 software engineering tasks into different categories including “low performance impact & high time spent,” the task most commonly placed into this non-promotable category was code review. Based on this finding and research on non-promotable tasks, women may be more likely to perform code reviews than men (see Appendix A). However, there is also reason to believe that engaging in code review might have some indirect benefits for software engineers because reviewing others’ code signals their expertise and allows them to demonstrate their ability to critique others’ work, even though they might not be aware of the benefits that might accrue. If it were the case that code review is an indirectly promotable task, then we might expect men to have greater opportunity, and to engage in more code review than women. In addition, because similarity bias suggests that people have more favorable opinions of people like themselves [45] and software engineers are primarily men (e.g. [53]), we might expect men to be requested, and to perform, more code reviews than women.

2.2 Code Review Equity

Code review has been studied extensively, and several researchers have explored how equity of code review outcomes are influenced by the demographics of participants. Terrell and colleagues found that on GitHub, when outsider women are perceptible as women, their pull request acceptance rate is lower than that of men, but when their gender is not perceptible, their pull request acceptance rate is higher than that of men [55]. In a laboratory study, Huang and colleagues found that participants spent less time inspecting women’s code than men’s during code review [32]. Nadri and colleagues found that perceptibly non-White GitHub contributors had lower odds of having their pull requests accepted than perceptibly White contributors [43]. Furtado and colleagues found that authors in countries with high human development indices (HDI) have higher pull request acceptance rates on GitHub than authors in countries with low HDI [21]. Rather than examining equity of code review outcomes, in this paper we instead look at review load equity.

Bosu and Sultana’s investigation of gender in open source found that in one of then 10 projects studied, women performed significantly fewer code reviews than men on the project [13]. Our
work builds on this by extending it to closed source software and exploring the antecedents of the review load gap.

2.3 Review Queue Length and Code Review Load

Prior empirical studies have touched on the issue of code review load, where review load is specifically defined as the number of changelists yet-to-be-reviewed at a given point in time, or review queue length. Baysal and colleagues found that the longer the review queue that an engineer has, the longer the delay an author can expect to get their code reviewed [8]. Ruangwan and colleagues found that an engineer’s review queue length is correlated with their likelihood to participate in an incoming review [49]. Kovalenko and colleagues at Microsoft [38] and Sadowski and colleagues at Google [50] found that authors sometimes take review queue into account when selecting reviewers.

Several code reviewer recommender systems have taken review queue length into account when recommending reviewers. Motivated by the above findings, Al-Zubaidi and colleagues created a system that uses a "multi-objective meta-heuristic algorithm to search for reviewers guided by two objectives, i.e., (1) maximizing the chance of participating in a review, and (2) minimizing the skewness of the review workload distribution among reviewers" [1]. Similarly, Rebai and colleagues’ system balances expertise, past collaboration, length of review queue, and number of changelists authored recently [46]. Chouchen and colleagues’ system balances review queue with review experience [18]. The notion of review load used in this paper – that is, the number of reviews performed in a fixed time window – differs from the length of review queue used in these papers. Indeed, our motivation for addressing review load inequities is different; in such prior work, "considering the workload of the developers is necessary for [reviewer recommenders] because the best candidate reviewers are not the best choices if they are not available for the review" [16]. In contrast, our motivation for understanding and improving review load equity is not to increase the likelihood that a recommendation will be accepted, but instead to increase knowledge sharing among engineers.

Two prior reviewer recommenders have similar motivations to our own. Mirsaeedi and Rigby argue that distributing knowledge among engineers is important and "speculate that code review can be effective in mitigating the turnover-induced knowledge loss" [41]. The authors thus created a reviewer recommender that uses review load as measured by the number of reviews performed in a three month period. Similarly, Strand and colleagues created and deployed a tool at Ericsson that attempts to balance review load across engineers, where review load is defined as the number of reviews completed in the last thirty days, as well as this number’s distance from the average across all reviewers [54]. A survey showed that engineers were evenly split about whether the system successfully distributed review loads. We anticipate that these two approaches could decrease the review load inequities described in the present paper.

3 METHODS

Given the large amount of data about code review at Google, we took a largely quantitative approach to understanding gender inequity in code review load and its antecedents (e.g. code ownership). We designed our analyses to be comprehensive, that is, throughout the research process, we tried to think broadly about what all the possible causes would be that we could evaluate with data. For each antecedent, we prioritized analyses that were feasible, that would enable plausible causal reasoning, and that we anticipated would yield practically actionable results. In this section, we describe the common aspects of the largely quantitative analyses we use in this paper.
We use pre-existing gender data that Google maintains as part of its annual diversity report [24]. The data is more than 99% complete for Google employees worldwide; we exclude missing data from our analyses. The reported gender categories are female or male.

We restrict our analysis to code reviews performed in Critique, the main code review tool used at Google [50]. Other code review tools like Gerrit and GitHub are used by some employees, so their experience is not captured here. We also restrict our analysis to submitted changelists, that is, merged into Google’s monolithic repository [36]. This excludes unsubmitted changes that were never given an LGTM (Looks Good to Me, similar to pull request approval on GitHub) from a reviewer, usually (in 93% of cases) because reviewers have not been assigned. In cases where reviewers are assigned, they can still be changed up until the code is submitted.

In our research, we followed the seven privacy principles for Google logs data described by Jaspan and colleagues [35], such as focusing only on tools and tasks used for work purposes. Analogous to an Institutional Review Board, the proposal for this research was reviewed by Google’s employee privacy working group.

We use regression analysis to predict a dependent variable of interest, such as the number of code reviews performed by a person. The main independent variable of interest is typically the gender of a person. We use cross-sectional data, where data is aggregated over a fixed period of time; for instance, the number of reviews performed per person over a three-month period.

Regression allows us to control for potential confounding factors. For instance, if tech leads do more reviews than individual contributors and men are more likely to be tech leads, then apparent differences in review load may partly be explained by job role. We use the following fixed-effect control variables drawn from prior quantitative code review research at Google [42]:

- **Role.** Either Individual Contributor, Tech Lead, Tech Lead Manager, or Manager. Individual contributors and tech leads tend to be more technical. This variable is related to “field of expertise”, which, according to Halvadia and Anvik’s survey, is one dimension of code reviewing expertise [27].
- **Tenure at Google.** As in prior work [42], to capture non-linear relationships we discretize tenure into either Less than a year, 1-2 years, 3-5 years, or 6+ years. This variable represents one dimension of “years of work experience”, which engineers believe is one dimension of code reviewing expertise [27].
- **Level.** Employees who are more senior have higher levels. We included employees with levels from entry level to senior staff level. Higher levels of seniority exist, but are rare and typically outliers in terms of job responsibilities. This variable represents an expertise dimension related to “years of work experience” [27].
- **Job Code.** Since there are many job codes at Google, we bucketed them into four categories. The most common job code for changelist authors and reviewers is Software Engineer (SWE). Another is Site Reliability Engineer (SRE). The third category is other types of engineers, such as Research Scientist Engineer. All other job codes are categorized as "Other." This variable represents an expertise dimension similar to “field of expertise” [27].

Distributions for these variables at Google can be found in the supplementary material of prior work [42]. When appropriate, our regressions include a random effect for team, making them mixed-effect models. The intuition is that some teams may behave differently than others; for instance, if women are more likely to be members of low review-load teams, then if women have apparently low review loads, the actual cause may be team placement.

We use three different types of regressions. The first type is an ordinary least squares regression, with a log transformed dependent variable whenever the dependent variable is tail skewed. The second type is a logistic regression, in cases where the outcome is binary. The third type is a linear
probability model, which is used when a logistic regression is not appropriate, specifically, when we wish to compare coefficients across different models. Logistic regression is not appropriate in this case due to their non-collapsibility. When reporting effect sizes from regression coefficients, we include 95% confidence intervals (CIs). We report adjusted $R^2$ for linear regressions, conditional $R^2 (R^2_c)$ for mixed-effect linear regressions, and McFadden’s pseudo $R^2$ for logistic regressions.

In different analyses, we sometimes use data over different time periods for two main reasons. First, when historical data was unavailable to us (e.g. Section 5.1), we could collect it only after we had formed a hypothesis. Since our hypotheses were not formed at the beginning of our research – but rather, over a multi-year research journey – some of our original data collection spanned different periods. Second, during our research journey, we shared our intermediate results with stakeholders, who later intervened (e.g. Section 7) to change how code review works. After such an intervention, some types of field data would be tainted by the intervention itself, limiting us to use only earlier data. The time period of data we analyzed is indicated in each analysis description.

4 GENDER INEQUITY IN REVIEW LOADS

In this section, we measure review load inequities by gender, first by examining the number of raw reviews performed in Section 4.1, then by adjusting for potential confounding factors in Section 4.2. Afterwards, we examine the antecedents to this gap: in Section 5, we examine gender differences in how reviewing credentials are assigned and earned, and in Section 6, we examine gender differences in how reviewers are chosen. Our final set of results in Section 7 evaluates an intervention designed to reduce review load inequities.

4.1 Raw Reviews Performed

To understand whether there was a gender difference in review loads, we began by calculating the number of reviews submitted by men and women.

In this section, we restricted our analysis to employees who were working full time and who had a Software Engineer job code. While other employees perform code reviews at Google, for this analysis, these restrictions allowed for a fair comparison of review loads.

Table 1 shows the median number of changelists (CLs) reviewed by men and women in 2019, showing a 25% gender gap in review load. This gap illustrates the issue intuitively, but not definitively; the raw difference conflates other effects, like the seniority of the reviewers and team-specific reviewing load.

<table>
<thead>
<tr>
<th>Women</th>
<th>Men</th>
</tr>
</thead>
<tbody>
<tr>
<td>162</td>
<td>215</td>
</tr>
</tbody>
</table>

Table 1. Review statistics in 2019 for full-time equivalent (FTE) software engineers at Google.

4.2 Adjusted Reviews Performed

To create a more controlled estimate of reviewing inequities, we adjusted for potential confounding factors by creating a mixed-effect linear regression. The regression’s dependent variable is the log of the number of reviews performed. The independent variable of interest is gender. For controls, we use a team random effect, and role, tenure, level and job code as fixed effects.

Our unit of analysis is a person during a fixed time period. One reason for using a fixed time period is that team members change; for instance, if a person joins a team months after the team is formed, it would be unreasonable to compare the total number of reviews performed by this person...
against that of people who joined at the team’s formation. We analyzed a three month period of time in the first half of 2019, including only people of the team who:

- Submitted at least one changelist (CL);
- Had only one primary team assignment;
- Were continuously employed during the entire period as a 100% FTE employee at Google with a SWE or SRE code, and between entry level and senior staff level, inclusive; and
- Did not change teams, gender, job codes, or roles during the study period. This restriction explains why we analyzed data over a period shorter than a full year: the longer the period of analysis, the smaller the team appears.

In total, the dataset for this analysis includes 45% of all reviewers who reviewed during the period and 85% of reviewed changelists. After running our analysis, the gender regression coefficient for women was -0.184. Since the dependent variable (number of reviews) was log transformed, exponentiating the gender coefficient yields a percentage change in the number of reviews, revealing that:

**Finding:** Women performed 16.8% fewer reviews than men (CI 13.7-19.8%, $R^2_c = 44\%$).

We replicated this finding for 2019-Q4 (19.2%, CI 16.2%-22.0%) and 2020-Q3 (17.6%, CI 14.7%-20.4%). This provides more robust evidence that women perform fewer code reviews than men at Google.

Why did women do fewer reviews? We answer this question in two parts: by examining gender gaps in reviewer credentials in Section 5 and then by examining gender gaps in how reviewers are selected in Section 6.

### 5 REVIEWER CREDENTIAL GAPS

Having shown that a review load gender gap exists, we next examine the credential gap, that is, gender differences in whether reviewers have been granted the requisite accreditation to perform certain kinds of reviews. We first consider gender differences in ownership credentials in Section 5.1, then differences in readability credentials briefly in Section 5.2.

#### 5.1 Ownership

Like many code repositories, Google’s codebase is access controlled by designating owners for different parts of the codebase. Owners are specified in OWNERS files, either by using usernames or by using permissions groups that typically include multiple people, such as a team of engineers. OWNERS files apply to the directory in which they are located, and ownership permissions are inherited from a directory’s parent directory.

Ownership is relevant to code review because if an author of a changelist is not an owner, the changelist must be approved (a special designation, apart from LGTM) by an owner. This ensures that code is never changed without the knowledge of a responsible employee. Thus, one hypothesis why women review fewer changelists than men is that women are less likely to be owners than men, as a consequence of similarity bias (men being more likely to choose men as owners). If women are less likely to be owners, then women are less likely to be chosen to review code when the author is not an owner.

Before we test this hypothesis, we next present some descriptive statistics about ownership. Because current ownership data is available but historical data is not, we ran a daily cron job to collect ownership data for changelists submitted between December 5, 2020 and April 20, 2021. We included all submitted CLs that are authored by a SWE or SRE. We excluded three types of CLs that are qualitatively different than the majority of CLs:
<table>
<thead>
<tr>
<th>Category</th>
<th>% of CLs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changelist does not require an owner’s approval (that is, the author is an owner)</td>
<td>69%</td>
</tr>
<tr>
<td>Changelist requires an owner’s approval that can be satisfied by someone on-team</td>
<td>16%</td>
</tr>
<tr>
<td>Changelist requires an owner’s approval that must be satisfied by someone off-team</td>
<td>14%</td>
</tr>
<tr>
<td>Changelist requires a combination of on-team and off-team approval</td>
<td>1%</td>
</tr>
</tbody>
</table>

Table 2. Ownership credential breakdown.

- CLs that are entirely changes to experimental code, because reviewers are optional.
- CLs that are entirely changes to open source code, because most changes are bulk transfers from public repositories like GitHub, so assigned reviewers are not inspecting the CLs for typical reasons, like knowledge sharing, correctness, or quality.
- Large scale changes (LSCs), which are a set of small but conceptually related CLs, where each CL is reviewed independently [61, Chapter 22]. For example, if a library owner is changing the name of a method, they may update all the references to that method across Google’s codebase with a set of LSC CLs. LSCs are typically very low risk changes.

We placed changelists into mutually exclusive and collectively exhaustive categories as shown in Table 2. From the table, we see that the majority of changelists do not require an owner’s approval, and, of those that do require an owner’s approval, they are split almost equally between needing off-team or on-team approval.

To evaluate our hypothesis, we are primarily interested in the second case in the table above – where only part of a team has been granted ownership in a codebase they work on. In this case, it is plausible that the author should be granted ownership credentials for the codebase they are working on. If men are more likely to be granted ownership to the team’s codebase, that would explain (at least in part) why women would be less likely to do reviews. Thus, to evaluate our hypothesis, we categorized each person in 2021-Q1 who submitted a minimum number of changelists (n=10) as either frequently needing an on-team owners’ approval for them (50%+ of CLs) or not. This dataset contains 34% of all code authors in the period. We then created a logistic regression that predicted the frequent need for an owner’s approval based on gender, while controlling for tenure, job code, level, and role.

The regression reveals that women have higher odds than men (33.3%, CI 21.6%-46.0%, McFadden = 11%) of frequently needing an owner’s approval from someone else on their team, supporting the hypothesis.

**Finding:** Women have lower odds than men of having ownership of their codebase.

One practice used by some teams at Google is “whole-team” ownership, where all engineers on a team are granted ownership automatically, such as by using an access control list in an OWNERS file. We hypothesize that teams that use such an ownership strategy tend to have more equitable code review loads. To evaluate this hypothesis, we began by calculating what percent of a team’s authored CLs in 2021-Q1 required a team member’s approval. Defining a cutoff value of 10% for this number, we find about half of engineers are on teams above 10% and half are below, halves we label as “more restrictive ownership” teams and “less restrictive ownership” teams, respectively.

---

To examine the sensitivity of our chosen 50% threshold to define “frequent need for an owner’s approval”, we also ran the model with two other thresholds. At 25% and 75% of CLs needing an on-team owner’s approval, women had 28.2% (CI 18.7%-38.5%, McFadden = 9%) and 30.9% (CI 16.2%-47.2%, McFadden = 11%) higher odds than men, respectively.

---

We next created a linear regression to evaluate whether women on teams with more restrictive ownership did fewer reviews than those on less restrictive ownership teams. The dependent variable was the log of the number of reviews performed. The independent variable of interest was the interaction between gender and team ownership type. We controlled for role, tenure, job code, and level, and a random effect for the team. We used the same dataset in our previous ownership regression. Table 3 shows the results.

From Table 3, we define the baseline as men on less restrictive ownership teams. Compared to this baseline, women on teams with less restrictive ownership do 8% fewer reviews than men and on teams with more restrictive ownership do about 13% fewer reviews than men ($R^2_c = 42\%$).

**Finding:** The gender gap in review load is similar across teams with more and less restrictive ownership.

### 5.2 Readability

At Google, readability [61, Chapter 3] is a credential that any employee can earn for a specific programming language, such as Java, C++, and Python. To our knowledge, readability is practiced only at Google and not in any other organization. Thus, in this section we summarize our findings about readability, leaving the full details for the assiduous reader in Appendix B.

When a person is readability certified, commonly called having readability, it means that they have demonstrated a thorough understanding of Google’s style guide and best practices for a language. A person gets readability by going through the readability process in, for example, Java. During the readability process, the candidate person goes through their normal, day-to-day development work, where they create a changelist with a substantial amount of Java code, choose peers to review it, and address the comments made by those peer reviewers. After the peer reviewers have been satisfied, the candidate then requests that a readability reviewer – a volunteer who is already certified in Java – reviews the changelist. The readability reviewer provides feedback related to Java style and best practices, and the candidate addresses that feedback. Once the readability reviewer is satisfied, they approve the change, and the code is submitted. Additionally, the readability reviewer submits a survey that provides an evaluation of the changelist.

One or more readability administrators periodically examine readability reviewers’ evaluations of multiple changelists and decide whether the candidate has demonstrated the requisite understanding. The number of changelists that candidates submit to the readability process before they graduate varies, but the median number submitted to the Java process is 10. According to Winters and colleagues, around 20% of Google engineers are participating in a readability process at any given time [61].

---

3Here we do not give a specific number of reviews as the baseline, as the number differs depending on the engineers’ tenure, level, etc.
Employees are not required to get readability certification, but it is recommended and can be beneficial to do so. Jaspan and colleagues have shown that, compared to an author without C++ readability, an author with C++ readability will have their C++ changelists reviewed in 4.5% less time and spend 10% less time dealing with reviewer feedback [35].

As a credential for code review, readability is much like ownership, whereby having readability is beneficial because for a given changelist, either the author or at least one reviewer must have readability in the changelist’s language(s) before the changelist can be submitted.\(^4\) Thus, if an author does not have readability for a language in which they wrote a changelist, they must have at least one reviewer who does have readability in that language.

We first examined readability reviewers – since readability reviewers are volunteers that do extra reviews, we hypothesized that if women were underrepresented in the readability reviewer pool, then this might explain why women are doing fewer reviews overall. We found that women were indeed underrepresented in most languages (see Section B.1). However, we also found that when readability reviews are excluded, the review load gap is not substantially reduced between men and women from 16.8% to 16.3% (CI 13.2%-19.2%).

Given readability’s usefulness as a credential for code review, we next hypothesized that women may be less likely to have readability than men, which could explain why women do fewer code reviews. This hypothesis was confirmed; women were indeed less likely to have readability than men, across programming languages. We posed multiple follow-up hypotheses to uncover the antecedents, several of which we were unable to clearly confirm:

- We hypothesized that women might be disproportionately placed in teams where readability would not be useful as a reviewer (Section B.2). This hypothesis was false; women satisfied 4% fewer "team readability needs" than men (that is, uses of a readability language in a teammates’ changelist).
- We hypothesized that women might not be writing code where readability would be useful to them personally. This hypothesis was false; women satisfied 5% fewer readability needs than men for their own changelists (Section B.2).
- We hypothesized that women may not have submitted as much code before beginning readability certification as men. This hypothesis was false; women did not submit a significantly different amount of code before beginning the readability process (Section B.3.1).
- We hypothesized that women may have been held to a higher standard than men during the readability process. This hypothesis was false; women did not submit a significantly different amount of code during the readability process (Section B.3.2).
- We hypothesized that women might be holding themselves to a higher standard during readability process by declining to send some of their code to the readability process. This hypothesis was mixed; women did decline to send more changelists than men to the readability process, but they did not send a significantly different amount of code (Section B.3.2).

What we were able to confirm was that women were generally overrepresented as applicants to readability processes (Section B.3.3). However, they are overrepresented as "stalling" in the program, that is, still working in a programming language, but no longer sending changelists to the readability program for evaluation. This finding implies that the problem is not that women are not signing up to earn readability, but instead are disproportionately abandoning the process to get it. Indeed, we found that women were 7.5% less likely to complete the readability process than men (Section B.3.4).

---

\(^4\)Exceptions include when a readability process does not exist for the language (e.g. Markdown), when writing purely experimental code, or when the number of lines changed in the language is very small (typically 5 lines or fewer).
To provide insight as to why women are disproportionately stalling in readability program, we sent surveys to men and women who appeared to be stalled (Section B.3.5). 983 respondents answered questions about reasons for stalling in the readability process and about satisfaction levels of various dimensions of the readability process. Overall, men and women perceived readability similarly across most questions. The one question that appeared to plausibly explain why women may be more likely to stall in readability was that women were marginally less likely to report receiving respectful feedback than men (p=.046, 0.13 points lower on a 5-point scale). To address this gap, an unbiasing approach such as anonymous author code reviews [42] may be effective.

6 REVIEWER SELECTION GAPS

While the prior section examined credentials useful for code reviewers, we next examine gaps in how women and men are selected for review. We first look at how code reviews are selected in general (Section 6.1). We then examine gaps in manual selection (Section 6.2), automated selection (Section 6.3), and incomplete reviews (Section 6.4). Finally, we examine gaps in how reviewers are recommended automatically prior to selection (Section 6.5).

6.1 How Code Reviewers are Selected

A reviewer can be selected in multiple ways at Google, as we show in Table 4. Engineers can make manual selections in the following ways:

- The author of the changelist can select a reviewer. For example, an author may decide to select an engineer on their team who is aware of the goals and technical details of the change. According to the first row in the table, in 2020 we found that of reviewers who either LGTM’d or approved a CL, 84% of reviewers were selected by the author. The last column indicates that of reviewers initially selected by the author, 82% completed the review (the remaining 18% did not LGTM or approve the changelist).
- A reviewer can self-select to review. For example, a reviewer might notice a CL that’s relevant to their own work and assign themselves to review it.
- Other users, such as already assigned reviewers, occasionally select reviewers as well.

Less frequently, changelist reviewers can be selected automatically through the following tools:

- gwsq can select a reviewer [61]. gwsq is a tool that can be configured to perform a set of actions on code reviews, such as choosing a reviewer randomly from a queue.
- Other tools. A long tail of other tools exist for automated selection, which we will not discuss further.

6.2 Manual Selection

The top half of Table 4 shows that 89% of reviewers are selected manually, far outpacing any of the automated reviewer assignment tools. Manual selection of reviewers, as with any human
decision-making process, could be prone to human biases, such as similarity bias [45]. If women are less likely to be selected because of biases against them in engineering [39], then this could explain why women perform fewer reviews. Thus, we hypothesize that women’s manually-assigned review loads are lower than men’s manually-assigned review loads.

To evaluate this hypothesis, we examined equity in reviewing loads for changelists with manually selected reviewers. We created a linear regression whose dependent variable is the log of the number of reviews performed by a reviewer, where only manually-assigned reviews are included. We controlled for role, tenure, job code, and level, and a random effect for the team. We used the dataset from Section 4.2, and excluded people who had reviewed no manually-assigned CL. 45% of reviewers during this period are included in this analysis.

The regression ($R^2_c = 42\%$) revealed that women reviewed fewer changelists when reviews are assigned manually. In percentage terms, women review about 16.3% fewer manually-assigned CLs than men (CI 13.2%-19.2%), supporting the hypothesis:

**Finding:** Women are less likely to be selected manually for reviews than men.

### 6.3 Automated Selection

As Table 4 illustrates, gwsq is the dominant reviewer selection tool. In principle, use of automated reviewer selection tools, such as gwsq, could reduce or eliminate human biases that are at play when reviewers are selected manually. Indeed, according to its documentation, gwsq tries “to assign reviewers fairly via round-robin”. Thus, we hypothesize that review loads assigned with gwsq will be more equitable than review loads assigned manually.

To evaluate this hypothesis, we use the same regression and dataset we used in the previous section, examining the gwsq-assigned load for each person.

As a result, we found that women reviewed 6.5% fewer changelists than men when reviews are assigned by gwsq (CI = 2.6%-10.3%), a value lower than when manually assigned with non-overlapping confidence intervals. This suggests that the review load equity gap can be reduced – though not completely closed – with gwsq.

**Finding:** gwsq reviewing loads are more equitable than manually-assigned reviewing loads.

### 6.4 Incomplete Reviews

Based on the last column of Table 4, reviewers do not complete every review that they are selected to perform. Reviewers might not complete reviews when they are removed as reviewers or when they do not grant LGTM or approval. If women are less likely to complete reviews, that could explain their lower review load; we hypothesize that women are less likely to complete assigned reviews than are men.

We examine this hypothesis by collecting data for all of 2020, then calculating the percentage of reviews completed over the whole year for each person who was selected to perform a code review. This dataset includes 41% of reviewers in that period. We then take the weighted average for men and for women, weighting by the total number of reviews assigned so that people who are rarely selected for review have less of an impact on the average than people who are selected more often. Here we find the average percentage of reviews completed for men was 83.3% for men and 82.0% for women.
To increase the robustness of our analysis, we perform a linear probability regression predicting the percentage of reviews completed based on gender, controlling for role, level, tenure, and job code. The regression ($R^2 = 3.2\%$) suggests that women were 0.5% less likely to complete reviews than men ($p=.037$), providing weak support for the hypothesis. Thus, we judge:

**Finding:** Women are nearly equally likely to complete assigned reviews as men.

### 6.5 Reviewer Recommendation

For manual reviewer selection, reviewers may be selected with the assistance of an automated tool called Suggest Reviewers. Suggest Reviewers can be invoked from the command line, from different integrated development environments, or from Critique, as shown in Figure 1.\(^5\) Although a reviewer who uses the “Suggest reviewers” button in Critique may or may not actually use the suggestion, looking at data from May 2020, at least 17% of CLs used the Suggest Reviewers button in Critique.

While use of reviewer recommendation may reduce the effects of human biases, algorithmic biases may cause women’s lower review load as well. Thus, we hypothesize that Suggest Reviewers disproportionately recommends men.

To test this hypothesis, we performed the same regression as in Section 4.2 (also with 45% of all reviewers), but instead predicted the probability that an engineer would receive a large scale change (LSC) for review.\(^6\) We chose to study LSCs here because they are the most common type of CLs for which we can be certain that reviewers are chosen automatically using Suggest Reviewers, rather than manually. After controlling for role, job code, tenure, and level, the results supported the hypothesis, in that women had 26.8% (CI 21.5%-31.8%, McFadden = 11\%) lower odds of receiving an LSC compared to men.

**Finding:** Suggest Reviewers disproportionately recommends men.

For robustness, we also examined the initial LSC reviewer assignment. The data above describes the reviewers listed after submission of a CL, but not necessarily who was initially assigned to a

---

\(^5\)Although we were not able to get data about command line invocations, in the first four months of 2021, there were about 4.6 times as many Suggest Reviewer API calls from Critique, compared to Google’s most popular internal web-based development environment.

\(^6\)Suggest Reviewers is used slightly differently for LSCs than for other CLs; in particular, owners can specify a special set of usernames as “cleanup-approvers” to receive CLs like LSCs. This represents a threat to our analysis; it may be that women are disproportionately likely to be chosen to be cleanup-approvers, but other, non-LSC uses of Suggest Reviewers may recommend men and women proportionally. To give a sense of the magnitude of the threat, of the more than a half million OWNERS files, cleanup-approvers appear in just 0.6\% of them, as of July 15, 2021. It is reasonable to assume that this roughly corresponds to coverage of files, and thus represents a limited threat to the results here.

---
Thus, we also modeled the initial reviewer assignment(s) of LSCs using the same dataset. The results are consistent: the odds that an engineer was selected initially for an LSC review was 23.3% (CI 17.7%-28.5%, McFadden = 10%) lower for women than for men.

We next examine why Suggest Reviewers may be more likely to recommend men. First, we note that previous findings in this report will affect Suggest Reviewers, because Suggest Reviewers uses several signals for choosing reviewers:

- **Ownership.** Based on OWNERS files, this signal is based on who has been granted privileges on the files in a changelist.
- **Readability.** If the author of the CL requires readability approval, potential reviewers who have readability in the programming languages used in the changeset will be more likely to be recommended.

In the following subsections, we describe two other signals that Suggest Reviewers uses to decide who to recommend: authorship (Section 6.5.1) and prior reviews (Section 6.5.2).

### 6.5.1 Authorship Signal

An engineer will be more likely to be recommended to review a changelist by Suggest Reviewers if the engineer has previously modified one or more of the files in the changelist. A subtlety of the authorship signal is that it is independent of the size of the change to a file. For example, an engineer who authors two changelists, one that changes 2 lines and the other that changes 1, is considered twice as familiar as an engineer who authors a changelist that modifies 3 lines to the same file. This example is illustrated in Figure 2.

Reflecting on this subtlety, if women construct and sequence their changelists differently from men, this may explain why men are more likely to be recommended by Suggest Reviewers. In particular, we hypothesize that men are more likely to split their changes into multiple, smaller CLs than are women.

In 2019, the median woman authored 8.7% fewer changelists than the median man, but those changes were 4.8% larger. Like our prior analysis, such raw values may be impacted by confounding factors, such as level and tenure differences that correlate with gender. To adjust for these factors, we created two linear regressions with team as a random effect, and gender, tenure, level, role, and job code as fixed effects. The regression ($R^2_c = 42\%$ using data from 44% of authors) that predicts the log of the number of changelists authored shows that women submitted 16.9% (95% CI, 14.7-19.1%) fewer CLs than men. The regression that predicts the log of the median CL size for an engineer shows that the median size for women’s CLs was 7.3% (95% CI, 4.4-10.4%) larger than men’s.

**Finding:** Women submit fewer (but larger) changelists.

One potential explanation for this finding is that women may be less likely to use a version control tool that enables engineers to submit more and smaller CLs because dependent changelists
can be chained and reviewed independently. To investigate this explanation, we ran a similar logistic regression, predicting the tool’s usage. We found that women had about 51% lower odds of using the tool than men. Re-running our above models to control for tool usage shows that women write 11.1% fewer (CI 9.3-12.8%) and 10.3% larger (CI 7.2-13.4%) CLs than men. We conclude that use of this tool is one, but not the most substantial, driver of these differences. Nonetheless, closing the usage gap of this tool is worthwhile, perhaps by using the GenderMag method to root out gender inclusion bugs that it might have [15].

6.5.2 Reviews Signal. Suggest Reviewers increases an engineer’s likelihood to review if they have previously reviewed changes to the files in the changelist under review. It appears self-evident that the previous mechanisms for women’s lower review load – less likely to have readability, less likely to be an owner, and so on – will be perpetuated and amplified by Suggest Reviewers’ prior reviews signal. For instance, if a woman does not have ownership in 2021, then she is granted it in 2022, she is not suddenly equally likely to be a recommended reviewer. Rather, her relatively low review load in 2021 will reduce her likelihood of being recommended in 2022 and beyond.

What remains unclear is how much of a difference the reviews signal makes in terms of the gender gap. To find out, we recognized that an empirical approach like we have taken so far in this paper would not suffice, because the question is largely about a hypothetical – what would have happened if the reviews signal was different? Thus, we decided to run a simulation.

Our simulation goes through each CL in the order in which they were submitted and assigns a reviewer by either:

- (80% probability) the author assigning a reviewer manually, or
- (20% probability) using the first suggestion from Suggest Reviewers

We assigned these probabilities based on a rough approximation of Suggest Reviewers usage, described previously. The simulation assumes that:

- Exactly one reviewer is selected per changelist;
- Readability and owners’ approvals are not taken into account;
- At the beginning of each simulation run, the gender of each member of the team is randomly assigned with equal probability given to men and women; and
- When an author of a CL manually assigns a reviewer, they randomly select a member of their team, but are 25% more likely to select a man than a woman.

Since our simulation is computationally intensive, we were unable to run it on the full history of Google’s codebase. Instead, we ran it on the full change history of five directories that we selected. The first directory chosen was one maintained by a team that we are familiar with; this allowed us to debug the results. The other four directories were selected using a script that finds directories that have enough CLs to generate meaningful results, but not so many that the simulation would take an unreasonable amount of time to run. Table 5 shows the directories we used.

<table>
<thead>
<tr>
<th>Directory</th>
<th># CLs</th>
<th># authors on team</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directory 1</td>
<td>2134</td>
<td>10</td>
</tr>
<tr>
<td>Directory 2</td>
<td>412</td>
<td>5</td>
</tr>
<tr>
<td>Directory 3</td>
<td>604</td>
<td>9</td>
</tr>
<tr>
<td>Directory 4</td>
<td>720</td>
<td>8</td>
</tr>
<tr>
<td>Directory 5</td>
<td>319</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 5. Directories studied in simulation.
Average # of CLs Assigned To:

<table>
<thead>
<tr>
<th>Prior Reviews Signal</th>
<th>Men</th>
<th>Women</th>
<th>Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turned On</td>
<td>106</td>
<td>90</td>
<td>16</td>
</tr>
<tr>
<td>Turned Off</td>
<td>104</td>
<td>93</td>
<td>11</td>
</tr>
</tbody>
</table>

Table 6. Simulation results.

We ran the simulation 4,000 times for each directory, for a total of 20,000 simulation runs. For half of the runs, we ran Suggest Reviewers with the prior review signal turned on (as it is now in production), and half the runs we turned the signal off.

For analysis, we used a linear regression, predicting the number of reviews performed for each person. We modeled gender, review signal, and their interaction as fixed effects. We modeled the person and project as a single random effect. All coefficients were statistically significant ($p < .001$, $R^2 = 98\%$). Table 6 shows the average number of CLs that the model estimates men and women review when the prior review signal is turned on and off.

The table shows that men are assigned to review more CLs regardless of the prior reviews signal in the simulation. This is expected, because most CLs were assigned manually in our simulation and simulated that reviewers were more likely to choose men. However, with regard to our hypothesis, we find that turning the prior reviews signal off reduces the review load gap by 5 reviews, from a gap of 16 reviews to a gap of 11 reviews.

7 A SMALL INTERVENTION AND ITS EVALUATION

Given that our analysis so far suggests that a gender gap in review loads exist, and that credential and selection differences are antecedents to that gap, we next turn to what we can do to close the gap. While our analysis suggests that a wide variety of interventions may be effective, in this section we report on one modest change to existing infrastructure as a first step. In Section 6.5.2, in a simulation we found that Suggest Reviewers disproportionately recommended men, compared to women. In part, this was because the algorithm uses prior reviewing experience, where prior reviewers are most often selected manually with a tendency to disproportionately choose men.

Motivated by this finding, the team at Google responsible for Suggest Reviewers reduced the relative impact of reviewer and approver familiarity signals in the Suggest Reviewers algorithm. Although our simulation suggests that this change should reduce the gender gap, we wanted to test the impact of the change in practice and at scale:

**RQ1:** Did the change increase the likelihood that Suggest Reviewers suggests women?

Since Suggest Reviewers is a tool used in production thousands of times a day, we also need to know whether we’ve substantially changed the quality of review recommendations:

**RQ2:** Did the change decrease the quality of suggested reviewers?

7.1 Method

As in Section 6.5, we examined reviewer assignments to large scale changes (LSCs), because in these changelists we know Suggest Reviewer was used for reviewer assignments. In particular, we examine assignments during the period after the change was rolled out to production (April 1, 2021 to September 9, 2021). To account for any potential seasonal effects, we compare this data to the same period the prior year (April 1, 2020 to September 9, 2020). We then examine all employees
who did at least one code review during each period (encompassing 62% of all reviewers), and for each reviewer, determine whether they received an LSC.

To answer RQ1, we calculate the raw probability an engineer will receive an LSC broken down by gender, and perform a regression analysis to control for confounding factors. The regression uses a linear probability model, controlling for role, tenure, job code, and level, with team as a random effect.

To answer RQ2, we measured reviewer quality by examining whether the reviewer assigned by Suggest Reviewers in LSCs actually LGTM’d the CL, as opposed to another reviewer LGTMing. If the initially assigned reviewer did not LGTM – either because they were removed from the review or didn’t complete their review – this implies that the suggestion was of low quality. We use this data in a linear probability model, with whether the CL was LGTM’d by the initially-assigned reviewer as the dependent variable, and the period (before or after the change) as the independent variable of interest. We included the size of the changelist and the main programming language used as control variables. When inspecting the data, we noticed that CLs were less likely to be LGTM’d by their first-assigned reviewer on US holidays and weekends, so we included another control variable indicating whether the reviewer was assigned on a weekday, weekend, or US holiday.

7.2 Results

Figure 3 shows the raw results for RQ1. The percentage of people who were assigned LSC CLs grew from 35% to 43% for women and from 48% to 53% for men. Comparing the differences, the growth was unequal; 5% more men received an LSC CL after the change, while 8% more women did. This suggests that the answer to RQ1 is “yes”, that the change was beneficial to close the gap between men and women. Our regression model ($R^2_c = 27\%$) substantiates these raw results. After adjusting for confounding variables, the model indicates that the probability that a man would receive an LSC rose by 5.3%, while women’s rose by 7.4%, a statistically significant gender difference ($p=.003$). However, it is notable that a gap remains, likely due to other contributing factors explained previously in this paper.

For RQ2, our modeling found little to no change in review quality. As a concrete baseline, before the change, a very small C++ LSC CL sent out on a US workday had an 85.1% chance of being LGTM’d by the originally assigned reviewer. In the after period, the model indicates that rate is
reduced to 84.8%. However, the model indicates that the difference in probabilities is not statistically significant (p=0.061) by typical standards.

8 LIMITATIONS
A variety of limitations should be considered when interpreting the results of this study:

• **One code review system in one company.** Due to the relative ease with which we could access data, we chose to study code reviews performed with Critique at Google, so findings may not generalize to other ecosystems where engineers do their reviews, such as GitHub. We discuss generalizability in Section 9.1.

• **External influence.** As suggested by Bardzell and Bardzell’s guidelines for feminist human-computer interaction research [7], we recognize that our ability to investigate the questions and publish the findings in this paper are influenced by the context in which we did our work, most notably Google. Compared to our experience doing purely academic research on publicly available data, we believe we were given sufficient autonomy by Google without undue influence. However, both the present research and any future research we might conduct using Google’s resources is contingent on maintaining good relationships with Google employees, leadership, infrastructure teams, and data stewards. We recognize that some external influence, even if unintentional or unconscious, was unavoidable.

• **Causal Inference.** Our ability to make causal inferences is limited to examining correlations while controlling for covariates, because gender is generally not a modifiable exposure. Nonetheless, some reasonable causal inferences can be made by relying on structural relationships (e.g., adding an owner will cause an increase in likelihood that the person will be recommended by Suggest Reviewers). Our choice of controls also limited causal inference. Some factors we were unable to control for, such as “code quality expertise [or] understanding of the project architecture” [27], which are difficult to reliably measure at scale. Other factors are controllable, such as collaboration and code files in prior code reviews [18], but we purposefully avoided controlling for such behavioral factors, especially those that may interact with gender. For instance, we found that on average women made fewer changes than men in the same amount of time (Section 6.5.1); had we controlled for files changed in prior reviews [18], the gender difference in review load may have been muted.

• **Team Identification.** We used data on which teams employees belonged to, but this data can be inaccurate or misleading. The data source we used was the canonical source of teams data at Google, but that information is largely based on how individual team managers decide to organize information about their teams. Inaccuracies can arise when, for instance, a team is not specified for an employee (e.g., in December 2021, we found 3% of software engineers had no primary teams listed) or an employee is listed under multiple primary teams (32% of engineers defined more than one primary team). In the latter case, we used the first specified primary team.

• **Generalizing from LSCs.** In our analysis of Suggest Reviewers, we analyzed only LSCs, expecting that our findings about Suggest Reviewers generalizes to other types of changelists. While LSCs and non-LSCs differ in some important ways – LSCs tend to be lower risk and made by authors who have less knowledge of the part of the codebase being changed – those differences don’t theoretically matter to how Suggest Reviewers works. What matters to Suggest Reviewers is, for instance, who has worked with the files being changed and who has ownership of those files, exactly like non-LSC CLs, supporting the argument for generalizability. However, LSCs may not be representative in ways that we did not anticipate; if so, generalization is limited.
9 DISCUSSION

In this paper, we have demonstrated gender review load inequities, and uncovered a variety of underlying systemic antecedents. In this section, we show how the these issues may generalize beyond Google, and what can be done about them.

9.1 Beyond Google

In Section 5.1, we found that women are less likely to have ownership of the codebases that they work on, and that teams using a whole-team ownership policy tend to have more equitable review loads. The use of ownership is common to every code repository of which we are aware. For instance, GitHub repositories use access permissions\(^7\) and CODEOWNERS files\(^8\) to control which engineers approve pull requests for merging.

In Section 5.2, we found that women were less likely to have readability credentials than men, which appears to be a consequence of women’s higher rate of stalling in the readability process, which in turn may be driven by a perception of women receiving less respectful feedback than men. While Jaspan and colleagues have found that readability credentials have beneficial effects on engineer productivity\(^9\), our findings suggest that other companies who wish to implement readability should be cognizant of and create strategies for mitigating biases that may cause inequitable outcomes.

Beyond this paper, while readability is specific to Google, the broader notion of code review credentials is not. For example, Bozorgzadeh advocates for the policy that “technical leaders should certify people for review. A review certificate shows that a developer has mastered both the technical skills and business aspects of the product.”\(^9\) As another example, GitLab’s code review documentation says that reviewers must be domain experts.\(^10\)

In Section 6.2, we showed that when reviewers are manually selected, people tend to disproportionately select men. We expect that this finding generalizes well beyond Google; in every code review system of which we are aware, the default option – and sometimes the only option – is to manually select a person to review a changelist. On one hand, this design decision is sensible – the author often knows someone who is well-qualified to review their code. On the other hand, a system that asks authors to select reviewers manually will inevitably be influenced by authors’ conscious and unconscious biases.

In Section 6.3, we showed that use of gwsq – a tool that automatically assigns reviews – results in a more equitable review load than manual assignment. We also expect this result to hold in other contexts. Other automated review assignment tools include GitHub’s auto assignment,\(^11\) GitLab’s Reviewer Roulette,\(^12\) Bitbucket’s reviewer assigner,\(^13\) and Gerrit’s reviewer plugin.\(^14\)

In Section 6.5, we showed how the Suggest Reviewers tool can perpetuate manual reviewer selection bias by using two problematic signals: past review history and past authorship history. The first signal – being more likely to recommend reviewers who reviewed the same code in the past – is used in all reviewer recommenders of which we are aware, including Microsoft’s cHRev [63]

\(^{7}\) https://docs.github.com/en/get-started/learning-about-github/access-permissions-on-github

\(^{8}\) https://docs.github.com/en/repositories/managing-your-repositories/settings-and-features/customizing-your-repository/about-code-owners

\(^{9}\) https://www.infoq.com/articles/practices-better-code-reviews/

\(^{10}\) https://about.gitlab.com/topics/version-control/what-is-code-review/

\(^{11}\) https://docs.github.com/en/github-ae@latest/organizations/managing-members-into-teams/managing-code-review-settings-for-your-team

\(^{12}\) https://about.gitlab.com/blog/2018/06/28/play-reviewer-roulette/

\(^{13}\) https://bitbucket.org/atlassianlabs/bitbucket reviewer assigner

\(^{14}\) https://gerrit.googlesource.com/plugins/reviewers/
and WhoDo [2], VMWare’s Review Bot [6], and Mirsaeedi and Rigby’s Sophia [41]. Thus, these systems likely also perpetuate bias from manual reviewer selection. The second signal – being more likely to recommend reviewers who have authored past changelists involving the same files – has more nuanced generalizability. Approaches like Sofia’s [41] use file-level authorship data, so such approaches may have the same challenge as Google’s Suggest Reviewers. In contrast, approaches like Review Bot [6] that use line-based authorship data may not have the same challenge, because women writing more lines of code per changelist (Section 6.5.1 and on GitHub [55]) may act as a counterbalance to the fewer changelists they write per unit time (Section 6.5.1).

9.2 Interventions

In Section 7, we showed how a small change to an existing system can reduce the review load gender gap. However, a wide variety of interventions are possible to mitigate the issues described in this paper. Throughout, we will use GitHub as a publicly-visible and widely-used point of comparison to show how these interventions could be implemented.

To address the issue of inequitable review loads generally, a grassroots approach to self-correct is for teams to have regular conversations about task equity in code review within their team. For example, a team may wish to look back at review loads every six months. Questions for reflection can include: Have some team members been assigned a disproportionately high or low number of reviews? What’s causing the disproportionality? Is the disproportionality desirable? If not, what steps might the team take to even out the review load? At Google, data on an individual’s reviews is easily queryable, but aggregating that data by review type and assignment source is inconvenient. Such review load information is also inconvenient to access on GitHub. Thus, a turnkey solution that provides actionable review load information may help teams and organizations redistribute review loads more equitably.

To better understand the rationale for restrictive ownership policies, we informally discussed it with two engineers (both tech leads) who were currently or had previously been on teams with such policies. Reasons for restrictive ownership policies included:

- Significant influx of new team members (especially new employees) may allow an inexperienced engineer to LGTM another inexperienced engineer’s code.
- A prior incident where an engineer mistakenly submitted a hard-to-retract changelist, which resulted in their ownership being revoked.
- Outages caused by breaking changes can affect paying customers.

When asked whether the choice of who gets ownership may be influenced by biases, both discussants said ‘yes’. In particular, one acknowledged that they may be unlikely to grant ownership to engineers who do not behave like existing owners, where ‘like behavior’ is defined as producing similar feedback on the teams’ changelists.

To address the issue of women being less likely to be granted ownership privileges, we have three suggestions. First, teams should consider granting whole-team ownership whenever possible, where a newcomer is automatically granted ownership when they join a team. The concern expressed by the two engineers above – that broad ownership enables inexperienced team members to LGTM each others’ code – may be addressed through other means, like using tooling to ensure that this does not occur at the time of reviewer assignment or to practice code review shadowing as a task-specific form of job shadowing [30]. Second, when whole-team ownership cannot be used, teams should consider defining ownership-granting criteria up-front so as to avoid bias. Such unbiasing techniques are
considered best practice in other workplace domains like hiring\textsuperscript{16} and promotion;\textsuperscript{17} we argue that unbiasing techniques should likewise be used when considering ownership. Moreover, as tech companies hire an increasingly diverse workforce, revisiting who is granted ownership at regular intervals could help ensure that ownership reflects this increasing diversity. Third, documentation on ownership within Google and on GitHub\textsuperscript{8} explain only the mechanics of granting ownership; readers may be more likely to implement unbiasing if this documentation frames ownership as a potentially biased process and provides resources for unbiasing.

To address the issues with automated reviewer recommendation, there are both immediate and broader implications. To start, we would not advise that reviewer recommenders directly utilize gender or other demographic data to make recommendations. Disadvantages of taking this approach include the danger of overburdening marginalized engineers with too many reviews, as well as the difficulty in being able to practically identify every group that faces inequitable review loads when building the recommender. However, several practical approaches are feasible. Since we found direct evidence that decreasing the weight of prior review experience improves equity, doing so in other contexts seems advisable. For instance, the toolsmiths who created WhoDo evaluated their tool with a 1 to 1 reviewer to authorship experience ratio, but noted that changing this ratio was simple [2]. Likewise, since we found counting authored changelists disadvantaged women when recommending reviewers, a remediation is for reviewer recommenders to factor in the size of authored changes. More broadly, we urge toolsmiths and researchers to consider fairness when designing and evaluating their reviewer recommenders. We also encourage broader thinking about what the purpose is of a reviewer recommender. Prior research has largely defined the purpose as recommending the \textit{most knowledgeable} reviewer, and to a lesser extent, distributing review load. But another purpose of these tools could be to recommend reviewers who are \textit{complementary} to the author. That is, when the author is already knowledgeable about the codebase, consider recommending a less contextually knowledgeable reviewer who can bring a fresh perspective.

To address the issue of men being disproportionately selected manually for review, several potential solutions exist. One potential solution is to use implicit bias training to help engineers choose reviewers more objectively, but implicit bias training’s effectiveness on decision making degrades over time [11]. Rather, we would suggest more systemic solutions, such as greater adoption and use of round-robin-style automated reviewer assignment. While development teams may choose to adopt such tools themselves, platforms could do more to make adoption easier; for example, adoption of Google’s gwsq and GitHub’s auto assignment could be spurred by making these features enabled by default, more findable, or easier to configure. At the same time, round-robin-style assignment is not a panacea, because review load inequities will continue to exist as long as credential inequities do. Moreover, automated assignment tools like gwsq allow complex customizations such as manually upweighting or downweighting reviewers, which can also lead to inequitable review loads.

More broadly, we believe that it is time to rethink the design of code review systems. In modern code review, a user \textit{must} select a reviewer, perhaps by selecting a team alias or assisted by a reviewer suggestion tool, if such a thing was configured and the author knows how to use it. But this design – to force a human to choose a reviewer – is not the only design possible. At the extreme other end of the choice spectrum, a code review system could remove the human from the loop, instead choosing reviewers automatically. As a middle ground, yet another design would be to nudge authors into choosing automatically-selected reviewers, such as by pre-filling the reviewer box. In

\textsuperscript{16}https://hbr.org/2017/06/7-practical-ways-to-reduce-bias-in-your-hiring-process
\textsuperscript{17}https://rework.withgoogle.com/print/guides/5443632811212800/
short, when it comes to the design of reviewer selection in code review systems, the way it is is not the way it must be.

10 CONCLUSION

In this paper, we applied an equity lens to understand code review loads in a large software development company. Through this lens, we saw how gender inequities can unintentionally be built into software engineering ecosystems. Such inequities that may have negative downstream impacts in terms of knowledge sharing. We demonstrated that equity can be increased with a small change to an existing reviewer recommender, but this is just one piece of a multi-faceted problem for which systemic solutions are needed. While the journey towards equitable software development requires thoughtful research and design, we look forward to an inclusive future where a diverse range of engineers can build software for a diverse range of users.

ACKNOWLEDGMENTS

Thanks to Alice Merrick, Andreas Eckleder, Annie Jean-Baptiste, Asim Husain, Cheryl Simon Retzlaff, Daniel Jasper, Danny Berlin, David Baird, Ingo Walther, James Dennett, Jason Roberts, Jeanine Banks, Jeff Cox, Jen Kozenski Devins, Kevin Storer, Liz Kammer, Matthew Bruggeman, Matthew Jorde, Mekka Okereke, Melonie Parker, Michael DePasquale, Míchéal Ó Foghlú, Michael Ricci, Ole Rehmsen, Rodrigo Damazio Bovendorp, Sameer Ajmani, Zachary Mccaw, and our anonymous reviewers.

REFERENCES


[31] Laura E Hirshfield and Tiffany D Joseph. 2012. ‘We need a woman, we need a black woman’: Gender, race, and identity taxhion in the academy. Gender and Education 24, 2 (2012), 213–227.


Robert Kevin Toutkoushian and Marcia L Bellas. 1999. Faculty time allocations and research productivity: Gender, race.

Anton Strand, Markus Gunnarson, Ricardo Britto, and Muhmmad Usman. 2020. Using a context-aware approach to


Joan C. Williams, Su Li, Roberta Rincon, and Peter Finn. 2016. Climate control: Gender and racial bias in engineering? Center for Worklife Law & Society of Women Engineers.


A Appendix: Deriving Non-Promotable Software Engineering Tasks

Before we began the quantitative analysis of code review in this paper, we began with a broader investigation into what non-promotable tasks [3] existed for software engineering.

A.1 Assembling a Non-Promotable Task List

We first prepared a list of tasks that software engineers perform at Google based on prior work. In creating this list, we consolidated tasks listed from four sources:

- the Software Engineering Body of Knowledge (SWEBOK) [14];
- a Google-internal document listing “impact” tasks that software engineers should perform to benefit the organization;
- a list of the common software engineering tasks retrieved from a Google-internal quarterly survey; and
- a set of tasks based on brainstorming with Google subject-matter experts.

To ensure we we had not missed any tasks, we distributed two versions of a survey. The first version – the “regret” survey – asked participants what tasks they spent a lot of time on and wished they had said “no” to doing. The second version – “performance appraisal” survey – asked participants what tasks they spent a lot of time on that are unlikely to noticeably improve their performance appraisal. We chose two framings because non-promotable tasks are multi-dimensional; a non-promotable task may be non-promotable because it does not benefit the organization in a quantifiable, easy-to-communicate way (thus, unlikely to help a performance appraisal score) or because the task is undesirable and, while beneficial to the larger organization, takes away time from clearly promotable work (thus, regrettable). The survey respondents could list 0 to 3 tasks and provide supplemental descriptions of the tasks. We also provided an open-response question to collect any other thoughts respondents had about low-promotability software engineering work.

We randomly sampled 400 software engineers that had gone through at least one performance appraisal at Google and received 33 responses (8.25% response rate). The 33 respondents reported 62 total tasks (17 from the “regret” survey, 45 from the “performance appraisal” survey), with 18 unique responses that were not already covered by other sources. Between the surveys and the prior work list, we collected 75 total tasks, including organizing an off-site event with your team, refactoring code, or monitoring system health and performance.

A.2 Ranking Non-Promotable Software Engineering Tasks

With our consolidated list of 75 tasks, we designed a card sort activity where participants sorted tasks based upon (a) their perceived high or low impact on a performance appraisal and (b) whether the tasks took a high or low amount of time. Each combination of high or low impact, and high or low time spent, was a category in which participants could sort the 75 tasks. Participants were instructed to only sort tasks that they have done themselves and, upon completing the activity, were asked if they had any additional low promotability tasks and how they personally defined “high and low” time spent and “high and low” promotability.

We deployed the card sorting activity in two phases. In the first phase, we performed a stratified sample where we took the four lowest promotion levels of software engineers, and randomly sampled 100 engineers from each level. The second phase included 600 engineers randomly sampled across all levels, without stratification. Among the 1,000 sampled engineers, 178 completed the
task (17.8% response rate). After performing a Fisher’s Exact test between the two card sort phases, we found no significant difference in how any of the tasks were sorted, thus all results will report counts that are merged between the two card sort phases. Participants were offered compensation for completing the task in the form of credits good for a massage.

A.3 Results
To maximize confidence in our results, we classify a task with at least 50% of respondents sorting it as "Low perf impact and high time spent" as non-promotable. Table 7 displays the resulting six non-promotable tasks, where categories are shown as columns and tasks are shown as rows. For instance, 111 participants said that they spent a high amount of time on code reviews tasks, but also that code review had a low amount of impact on their performance assessments.

B APPENDIX: EXTENDED READABILITY ANALYSIS
B.1 Equity in Readability Reviewers Load
As part of the readability process, engineers can volunteer to review and evaluate readability candidates’ changelists – so-called “readability reviewers”. If women are less likely to be readability reviewers, that may be responsible for their overall lower review load. Thus, we next evaluate the hypothesis that women are less likely to be readability reviewers.

To evaluate this hypothesis, we first compare the percentage of readability reviewers who are women against the readability percentages from the beginning of this section. In seven out of nine readability programs, we found that women are underrepresented as readability reviewers. For example, for the C++ program, we found that 17.4% of Submitters are women, but only 8.2% of readability reviewers are women. Overall, however, we judge that the hypothesis is largely supported:

**Finding:** Women are less likely to be readability reviewers than men in most languages.
Although men tend to disproportionately be readability reviewers and those reviewers are volunteering for higher review loads, this does not explain why women do fewer reviews overall. The overlapping confidence interval suggests that the reduced gap is not substantial.

Given the finding, would it be advisable to try to increase representation of women as readability reviewers? At this point, we do not believe that we have enough information to say. While improving representation of women in readability reviewing is a worthwhile goal, asking more women to volunteer as readability reviewers may amount to a minority tax for women [47], since we would be asking them to solve the problem without understanding whether deeper equity issues exist. Future qualitative research can help us better understand why there are few women readability reviewers, and whether closing that gap may yield benefits.

It is notable that the process of a person becoming a readability reviewer requires the person to apply, be evaluated, and be accepted. Our analysis above only looks at the outcome of this process. Biases are likely to play a role in the evaluation process as well, though whether women might be favored or disfavored remains future work. One way to reduce such biases is to use unbiasing methods used elsewhere at Google, such as anonymous author code review [42], when evaluating candidate readability reviewers.

It is also notable that some readability languages use self and peer nominations as a way of recruiting qualified readability reviewers to apply. Analogously, prior research shows that:

- Women may be less likely to self-nominate, due to social gender norms: “behaviours such as self-nomination are interpreted differently when enacted by men and women… ‘what appears assertive, self-confident and entrepreneurial in a man often looks abrasive, arrogant or self-promoting in a woman’” [44].

- In job referrals, a kind of peer nomination, people tend to refer people similar to themselves, which “exacerbates inequality among workers” [22].

Although we have no direct evidence here, self and peer nominations for readability may suffer from the same problem. One way to reduce the impact of these types of biases would be to shift to more structured and uniform nomination mechanisms. For instance, periodic emails could be sent to all potential candidates, encouraging them to apply to be a readability reviewer, including statistics about other successful applicants (e.g. number of CLs submitted in the language, prior to application), and then include custom statistics about how the recipient in particular stacks up against other successful applicants.

### B.2 Who Has Readability

In this section we examine the hypothesis that women are less likely to have readability than men, which would explain (at least in part) why women review fewer changelists than men.

To evaluate this hypothesis, we first investigated whether readability differences exist by roughly examining what percent of reviewers are women for different languages. More precisely, for a given programming language with a readability process, we calculated the following metrics on a quarterly basis:

- **Total Number of Submitters** represents the number of people submitting at least 10 CLs for which readability was required during the quarter in the programming language. This metric is intended to estimate a baseline of the number of users of the language, as well as the total addressable market of people who could benefit from having readability in the language.

- **% Women Submitters** represents the percentage of the Total Number of Submitters who identify as women.

18As we show in Section 6.5.1, on average women submit fewer CLs over time; consequently, using this fixed cutoff may undercount women language users, compared to men.
- **Total Number of People Readability Certified** represents the number of people with readability.
- **% Women with Readability** represents, out of all the people with readability in the language, what percent identify as women.

We analyze data only for languages for which there are at least 50 people with readability for 2020-Q3. Overall, we found that women are underrepresented in each readability language. For example, for the C++ program, we found that 17.4% of Submitters are women, but only 13.7% of people who have readability are women. However, a weakness of this analysis is that it does not control for covariates like tenure, and also does not capture whether engineers actually need readability in each programming language to do their work.

To investigate whether readabilities are needed, we introduce a unit we call *readability-need*. A *readability-need* is a readability language and a changelist. For instance, a changelist that modifies 100 lines of Java, 50 lines of C++, and 1 line of javascript constitutes two *readability-needs*: one for Java, one for C++, and none for javascript since the change in that language is small enough that readability approval is not required. A *readability-need* can be satisfied if the author or at least one reviewer has readability in that language.

Using this unit, we can calculate two metrics:

- Percent of team *readability-needs* satisfied. For an engineer, this is the number of *readability-needs* authored by other engineers on their team that the engineer can satisfy, divided by the total number of team *readability-needs*. If, on average, women satisfy a lower percentage of team *readability-needs* than men, then our hypothesis that women are less likely to have (practically useful) readabilities is supported.
- Percent of own *readability-needs* satisfied. For an engineer, this is the percent of *readability-needs* for the changelists they authored that they themselves can satisfy. If, on average, women satisfy a lower percentage of *readability-needs* as author than men, then increasing the number of women who have readability will not only be beneficial for their team, but also for their own work, since having readability increases an engineer’s own velocity [35].

We calculated these metrics for changelists authored between February and April 2020. To control for covariates, we created linear regressions predicting each metric, with the team as the random effect and gender, tenure, role, level, and job code as fixed effects. We restricted analysis to engineers with a SWE or SRE job code; between entry level and senior staff level, inclusive; having only one primary team assignment; and were 100% FTE employees at Google on April 30, 2020. We restricted metric calculations to engineers who had at least 10 *readability-needs* as reviewer or 10 *readability-needs* as author. In sum, we analyzed data for 67% of reviewers during the period.

We found that engineers could satisfy an average of 35% of team *readability-needs* as a reviewer. However, the regression \( R^2_c = 42\% \) indicated that women had 4% lower satisfaction of team *readability-needs* \( (p < .0001) \). To illustrate this more concretely, consider engineers who can satisfy none (0%) of their team’s *readability needs*. Examining just mid-career engineers (Software engineers with an Individual Contributor role who have been at Google between 3 and 5 years and are one level above entry-level) in our dataset, 30% of such male engineers can satisfy none of the team’s...
readability needs, whereas 37% of such female engineers can satisfy none of the team’s readability needs. This data provides further evidence to support our hypothesis:

Finding: Women are less likely to have readability than men.

We also found that engineers could satisfy an average of 43% of their own readability-needs. The regression ($R^2 = 37\%$) indicates that women had 5% lower satisfaction than men ($p < .0001$) of those readability-needs, suggesting that increasing the number of women who have readability will be personally useful, not just useful to the team.

### B.3 Equity in the Readability Process

To determine why women are less likely to have readability than men, we examine the process of obtaining readability in three phases: before the readability process (Section B.3.1), during the readability process (Section B.3.2), and exiting the readability process (Section B.3.3).

#### B.3.1 Pre-Readability

For people who decide to engage in the readability process, we next examine behavioral differences between genders with regard to readability. We hypothesize that women submit more code than men before beginning the readability process.

This hypothesis may be true if women are holding themselves to a higher standard [34] or have lower self-efficacy [10, 28, 33, 40]. Such gaps could be reduced with clearer messaging on when a person is ready for readability.

We examined the hypothesis by using the following metrics:

- **CLs Submitted in Language** is the number of changelists a person submits that used the language prior to beginning the readability process (a median of 38 CLs for all people studied) and

- **Lines of Code in Language** is the number of lines of code they changed in changelists submitted that used the language before beginning the readability process (median: 6166).

We ran two linear regressions predicting the log of each of these metrics, controlling for the readability language, as well as the level, job code, role, and tenure of the person. Since the same person can get readability for multiple languages, we included a random effect for the applicant. Each regression included the main independent variable of interest – the gender of the person who began the readability process. We analyzed data for people who completed the readability certification, representing 9% of authors during the period.

Overall, the regressions did not show a statistically significant difference in terms of CLs ($p = .363, R^2 = 50\%$) or lines of code submitted ($p = .098, R^2 = 46\%$) before beginning the readability process. Thus, our findings do not support the hypothesis:

Finding: Of the people who have readability, women and men do not differ significantly in terms of the amount of code submitted before beginning the readability process.

#### B.3.2 During Readability

To explore potential explanations of why women are less likely to have readability, we examined two hypotheses regarding the readability process itself:

Hypothesis 1: Women submit more code than men during the readability process.

One potential explanation for women being less likely to satisfy their own readability needs would be that their CLs have more readability needs. This is not the case: women and men have nearly identical mean readability-needs per CL: 1.039493 and 1.039415, respectively, a non-significant difference ($p = 0.360$, Mann–Whitney U test).
This hypothesis may be true if readability reviews are holding women to a higher standard when they are being evaluated as ready for graduating.

**Hypothesis 2:** Women are less likely to send their CLs to readability reviewers during the readability process.

This hypothesis may be true if women are holding themselves to a higher standard or have lower self-efficacy – that is, they may be deciding not to send some changes for readability evaluation. We examine each hypothesis below.

**Do women submit more code than men during the readability process?** We examined Hypothesis 1 by using the same metrics as the prior section (CLs and lines of code), but examined only the CLs submitted by people who completed the readability process, during that process. Also like the prior analysis, we use regressions to control for covariates.

Overall, the regressions did not show a statistically significant difference in terms of CLs ($p = 0.988, R^2_c = 40\%$) or lines of code ($p = 0.938, R^2_c = 25\%$) submitted during the readability process. Thus, we conclude:

**Finding:** Of the people who have readability, women and men do not differ significantly in terms of the amount of code submitted during the readability process.

**Are women less likely to send their CLs to readability reviewers?** We evaluated Hypothesis 2 by examining the readability process for each person who has completed readability, and counting the number of CLs (median: 29) and total lines (median: 3353) submitted that they decided not to send for readability review. Like the prior analyses, we used a regression to control for covariates.

While there was no significant difference in the total number of lines of code that were not sent for readability ($p = 0.191, R^2_c = 26\%$), women declined to send 8% more CLs ($p = 0.008, R^2_c = 35\%$) during the readability process. This provides partial support for Hypothesis 2:

**Finding:** Of the people who have readability, women decline to send more CLs to readability reviewers during the readability process than men, though the total lines of code they decline to send are not significantly different.

### B.3.3 Exiting Readability

To examine another angle on why women are less likely to have readability, we next examine gender differences in stalling in the readability process.

**Are women more likely to stall in the readability process?** Figure 4 below shows data for the C++ language over time – chosen here because it is the readability language with the most readability reviewers, and is reasonably representative of other languages – with the percentages indicating what percent of each series women make up:

- The orange line (■) indicates the percent of Submitters in the language that are women, as defined previously. For example, we see that in C++, the percentage of submitters that are women has gradually risen from around 11% to around 17%. This can be considered a baseline for “representation”.
- The green line (■) indicates the percent of Readability Certified people who are women, as defined previously. For example, we see that in C++, women’s representation in having readability has risen over time, but has stayed consistently below the baseline Submitters rate.
Fig. 4. Percent of women per quarter among four populations for the C++ readability program. Data is included only for a quarter when at least 10 women are represented.

- The purple line (■) indicates the percent of readability applicants who are women. Applicants are defined as people who submitted their first CL for a language to that language’s readability queue. For most languages, women have consistently been overrepresented in the applicant pool relative to Submitters, yet comparing this to the green line, are underrepresented in the Readability Certified pool. This implies that women are not completing the certification process. We next examine this directly.

- The pink line (●) indicates the percent of people with Incomplete Readability who are women. A person has Incomplete Readability if they were an Applicant that quarter, but have not been granted readability since that time. The last quarter shown in the chart is 2020-Q2, meaning that significant time has elapsed since this analysis was run (March 2021) for engineers to complete the readability process.

When we inspect the figures for all languages that have readability programs, we observe the following:

- Women generally are underrepresented as having readability (green line), compared to those using the language (orange).
- Women are, surprisingly, usually overrepresented in terms of beginning the readability process (purple).
- For many languages, the pink line (those not finishing readability) is generally higher than the purple line (those starting readability), suggesting that women are less likely to finish the readability process than men. Higher rates of readability abandonment are a threat to the goal of improving representation in readability.

In what follows, we seek to better understand the “when” and “why” of readability abandonment by looking at people who have “stalled” in the readability process, that is, ceased submitting CLs. We first examine at what point women stall in the readable process, then examine stalling more robustly with statistical methods. As stated in a prior footnote, it is difficult to distinguish between people not intending to complete the process (what we might call “real abandonment”) from people who intend to continue but are currently stalled (what we might call “abandonment false positives”).

The largest tenure cohort of readability applicants are those who have been at Google less than a year. Since gender diversity among tech employees has been increasing over time [24], this helps explain why women tend to be overrepresented among applicants. However, that’s not all that’s driving this trend; for instance, considering only people with 3-5 years of tenure in 2021Q1, 20% of C++ users were women (CI 19.1%-21.1%), yet 33% of C++ readability applicants were women (CI 22.7%-44.4%).
Fig. 5. Outcomes of sequential phases of the readability process. Labels for groups of fewer than 10 people or representing less than 1% of the total are omitted.

B.3.4 When are women stalling in the readability process? Are women stalling early or late in the readability process? To answer this question, we divided the readability process into different phases, examining stalling at each phase, as shown in Figure 5.

Each pair of bars in Figure 5 represents a sequence of CLs submitted to the readability process. The first pair represents the 1st CL, then the 2nd and 3rd CL, then the 4th through 7th, 8th through 15th, 16th through 31st, and finally 32nd through 63rd. Due to low numbers, people who continued to submit CLs to the readability process after 63 CLs are not included in Figure 5. Within the pairs, the bar on the left represents women, and the bar on the right men. Each bar is color coded to indicate what occurred to each participant on or during that period, either:

- **Stalled, then authored many CLs** in the same language as readability was being sought. Here, “many CLs” means submitting at least 10 CLs of at least 10 lines of code in the language; we chose the CL count threshold because it is the median number of CLs that Java graduates submit to the readability process, and chose the lines of code threshold, conservatively, as twice the typical number of lines of code for which readability is required (Section 5.2). We constructed this group to represent people who did not complete the process yet would have benefitted from continuing.

- **Stalled, then authored few CLs.** This represents people who have not finished the readability process, but also did not appear to use it much after the last time they submitted a CL to the process. They may not use it much because they switched programming languages, left the company, or so recently submitted their last CL to readability that they have not had the opportunity to submit another.

- **Continued the readability process.** Unlike other categories, anyone who shows up in this category will also show up in the next phase.

- **Graduated, then submitted few CLs.** These people were granted readability during this phase, but did not appear to write a lot of code in the language afterwards.

- **Graduated, then submitted many CLs.** We constructed this group to represent people who realized the most benefit from earning readability.
As an example interpretation of Figure 5, the leftmost pair of bars indicate that after the 1st CL is submitted to readability, 3.6% of women stalled in the process but continued to write CLs in the language, compared to 3.2% of men.

From Figure 5, we see that women tend to disproportionately stall in the readability process across all phases, with no discernible tendency to stall earlier or later. This result is robust to the context of stalling, whether continuing to write CLs in the language or not.

**Why are women more likely to stall in the readability process?** From the prior section, we observed that women are more likely to stall in the readability process. We next ask two questions:

- Are women still more likely to stall in readability, controlling for confounding factors?
- What accounts for women’s increased likelihood of stalling?

To answer the first question, we created a linear probability model to predict whether an engineer who began the process either completed the process or did not complete the process. Independent variables were the engineer’s level, job code, role, and tenure, as well as the readability language to account for differences in completion rates between languages. We excluded people who “authored few CLs” (as defined previously) after their last CL submitted to the readability process. This analysis includes 15% of code authors.

This regression indicates women are 7.5% less likely than men to complete the readability process (CI 4.5%-10.6%, $R^2_c = 6\%$).

In this model, we also included an interaction between language and gender, to determine whether the gender effect depended on the language. Most interactions were insignificant, meaning that the 7.5% figure above is similar across languages.

**Finding:** Women are about 7.5% less likely than men to complete the readability process.

To answer the second question, we created another linear regression, adding in three additional variables that we hypothesized might explain why women were less likely to complete the readability process. The three variables we added were:

- **Pre-readability preparation.** We hypothesized that people who submit more CLs in a language before beginning the readability process are more prepared to begin the process. Thus, we added a variable that was the log of the number of prior CLs submitted.

- **Many comments on readability CLs.** We hypothesized that participants who got more comments from readability reviewers would be more likely to stall in the process. Thus, we added the median number of estimated comments received from the readability reviewer during the readability process. Here we required an estimate, because existing data sources were insufficiently detailed to obtain the actual number of comments from readability reviewers. Since we did have the actual number of comments received from readability reviewers for recent CLs, we validated that the estimate was reasonable by correlating the estimate to the actual number of comments; we found the two had a Pearson’s r of 0.9, indicating a large correlation.

- **Team switches.** We hypothesized that people who switch teams after beginning the process are more likely to stall in the process, as their new team may be less likely to have a need for someone with readability in a language that was useful in the old team. We operationalized this notion by capturing team assignments on one of the following CLs:
  - The 10th CLs in a language after their last CL submitted to readability in that language;
  - If there are fewer than 10 in the language after their last CL submitted to readability, then the most recent one in the language; or
– If no CLs in a language were written after the last one submitted to readability, then the last CL submitted to readability.

We then compared the primary team assignment(s) of a participant on these two CLs. If the participant continued membership in all of the original primary teams, we considered them as staying on the same team. Otherwise, if they continued membership in at least one of the primary teams, we considered them “mostly” staying on the same team. Otherwise, we considered them switching teams.

The three new variables in the regression ($R^2 = 7\%$) showed interesting results:

- Writing more CLs prior to beginning the readability process correlated significantly with higher rates of stalling ($p < .001$).
- Receiving more comments correlated with higher rates of stalling ($p < .001$).
- Those who switched teams were about 3.6% more likely to stall.

In this regression with these three new variables, women were 6.5% less likely to complete the readability process (CI 3.5%-9.4%). Comparing this to the original 7.5% estimate (CI 4.5%-10.6%), considering the overlapping confidence intervals, we infer that these factors account for little, if any, of women’s disproportionate stalling in the readability process.

B.3.5 Readability Survey. We concluded that qualitative research was necessary to uncover what accounts for women’s disproportionate likelihood of not completing the readability process. We ran a survey among engineers who had stalled in the readability process to understand a) the primary reasons engineers stop submitting CLs for readability approval, thus blocking their completion of the process, and b) whether engineers’ reasons for stalling the readability process, pain points with the readability process, and attitudes toward the readability process differ by gender. Consistent with our prior analyses, we defined stalling as having not submitted their last 10 CLs written in the language in which they were pursuing readability certification for consideration in that language’s readability process.

We sampled engineers who started the readability certification process in the last year but had not submitted their last 10 CLs in the language for readability review (n=3009). We prioritized those who had stalled the process during the most recent year to ensure pain points identified were relevant to the current implementation of the process. Our study was broken into two phases. First, we invited a subset of our sample (n=899) to take a two question survey with a close-ended question asking about their likelihood to resume the process (“Very likely” to “Not at all likely”) and an open-ended question gathering reasons they’ve stalled the process (“Please describe any reasons you’ve recently submitted CLs in [embedded field: language] without submitting them to readability”). One author thematically coded the open responses to this open-ended question (n=285) using an inductive approach.

We then created a close-ended survey question based on our thematic analysis to send to the remaining sample (n=2110) within the second survey. Respondents were able to select multiple reasons for stalling. In this second survey, we also asked the respondents the same close-ended question about their likelihood of resuming the process, as well as a subset of the survey questions that are sent by the readability program managers to those who complete the process. The aim of including these additional questions was to gather more information about the respondents’ attitudes toward the readability process and their pain points. These additional questions measured satisfaction with topics including the quality of feedback provided, the consistency of feedback across reviewers, documentation of the readability process, the time required for readability review, clarity of their progress toward obtaining readability certification, and the length of the overall certification process. Table 8 lists all questions in both surveys.
### Survey 1

1. How likely is it that you will submit one of your next 10 CLs in [embedded: language] to readability?  
   (Very likely, Likely, Somewhat likely, Slightly likely, Not at all likely)
2. Please describe any reasons you’ve recently submitted CLs in [embedded field: language] without submitting them to readability. (Open Ended)

### Survey 2

1. How likely is it that you will submit one of your next 10 CLs in [embedded: language] to readability?  
   (Very likely, Likely, Somewhat likely, Slightly likely, Not at all likely)
2. Please select any reasons you’ve recently submitted CLs in [embedded field: language] without submitting them for readability approval. Select all that apply.
   - The additional time required for review is too long given my deadlines
   - My recent CLs have not been good candidates for readability because of their size or content
   - My teammates have readability
   - My team or projects’ standards differ from the readability standards (e.g., due to framework constraints)
   - The feedback I’ve gotten is inconsistent, nitpicky, and/or not helping me grow
   - I forgot
   - The codebase I’m making changes within is not up to readability standards (e.g., legacy code)
   - I don’t understand my progress or don’t seem to be making any
   - The overall certification process takes too long and doesn’t seem worth it
   - I am no longer or not currently working in the programming language
   - Other: (Open ended)
3. Please rate your level of agreement with each of the following statements based on your experience so far pursuing readability certification in [embedded: language]. (Matrix: Strongly disagree, Disagree, Neither agree nor disagree, Agree, Strongly Agree)
   - My readability experience has been positive overall.
   - I believe that the readability process is worthwhile.
   - I understand the criteria for achieving readability.
4. Please select how frequently each of the following occurred during your experience so far pursuing readability certification in [embedded: language]. (Matrix: Never, Rarely, Sometimes, Frequently, Always)
   - Readability reviewers responded promptly during the review.
   - Readability reviewers provided helpful feedback.
   - Readability reviewers were respectful when providing feedback.
5. How consistent have comments been across different reviewers for [embedded: language]? (Very consistent, Consistent, Somewhat consistent, Slightly consistent, Not at all consistent)
6. The documentation for the [language] process clearly defines what changelists should be submitted as part of the process? (Strongly disagree, Disagree, Neither agree nor disagree, Agree, Strongly Agree)
7. The documentation for the [language] process clearly defines what changelists should be submitted as part of the process? (Very consistent, Consistent, Somewhat consistent, Slightly consistent, Not at all consistent)
8. The length of the process/number of reviews during the readability process in [embedded: language] is: (Much too long, Too long, About right, Too short, Much too short, Not sure)
9. Please share any other sources of dissatisfaction with the readability process in [embedded language]. (Open ended)

Table 8. Readability survey questions.
38% of those invited responded to the first survey (n=346) and 30% of those invited responded to the second survey (n=637). The open-ended responses from the first survey that described reasons for stalling the readability process were coded with categories that aligned with the close-ended answer options in the second survey, and the phrasing of the close-ended questions about their likelihood to resume the readability process were identical in both surveys, so we merged the responses to these questions across the two surveys into one dataset. The remaining responses we analyzed were to questions only asked within the second survey, which was longer. Across both surveys, there was not a substantial difference between the gender breakdown of those engineers who responded and those engineers who were invited, indicating the respondents are representative of the group of engineers who had not submitted their last 10 CLs to readability at the time of the study with regard to gender.

Of the 10 reasons for stalling in the readability process and 14 other satisfaction questions, responses were overall similar for women and men. To examine statistical differences, we applied logistic regressions for reasons for stalling (one for each reason, predicting whether a participant selected that reason) and a linear regressions for satisfaction questions (one for each question, predicting the ordinal response to that question), controlling for level, tenure, job code, role, and readability language. Gender was the predictor of interest. A generalized variance inflation factor (GVIF) test revealed GVIF values smaller than 1.5, suggesting that multicollinearity was not a substantial threat.

Of the 24 models, two showed significant gender differences:

- Women reported being more satisfied than men (p=.041, 0.22 points higher on a 5-point scale) with the length of time they had to wait for the readability process to start, that is, the gap between when they enrolled in the program and when they were allowed to start sending CLs to readability reviewers. This difference does not seem related to women’s higher likelihood of stalling in the readability program.
- Women reported receiving less respectful feedback than men (p=.046, 0.13 points lower on a 5-point scale) from readability reviewers. This difference seems plausibly related to women’s higher likelihood of stalling in the readability program.

Given that only two out of 24 items showed gender differences, that the p-values are nearly statistically insignificant, that a false discovery correction (e.g. [9]) may yield no statistical differences, and that the effect sizes were small, suggests that overall, men and women who have stalled may not perceive the readability process substantially differently. Nonetheless, an unbiasing approach such as anonymous author code review [42] may be effective in ensuring that readability candidates receive uniform feedback, independent of their demographics.

**B.4 Regression Results**

*Regressions for "Section 4.2 Adjusted Reviews Performed"*

Summary of regression model:
Linear mixed model fit by REML. t-tests use Satterthwaite’s method ['lmerModLmerTest']
Formula: review_count_formula
Data: reviewer_cls_complete

REML criterion at convergence: 81471.1

Scaled residuals:
Min 1Q Median 3Q Max
-4.2483 -0.5081 0.1251 0.6326 4.2249

Random effects:
Groups  Name   Variance  Std.Dev.
team_name (Intercept) 0.6149 0.7842
### Systemic Gender Inequities in Who Reviews Code

#### Summary of regression model

Linear mixed model fit by REML. *t*-tests use Satterthwaite's method ['lmerModLmerTest']

**Formula:** review_count ~ formula

**Data:** reviewer_cls_complete

**REML criterion at convergence:** 77494

**Scaled residuals:**

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>1Q</th>
<th>Median</th>
<th>3Q</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-4.5142</td>
<td>-0.5084</td>
<td>0.1124</td>
<td>0.6362</td>
<td>4.8252</td>
</tr>
</tbody>
</table>

**Random effects:**

<table>
<thead>
<tr>
<th>Groups</th>
<th>Name (Intercept)</th>
<th>Variance Std.Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>team_name</td>
<td>0.6031</td>
<td>0.7766</td>
</tr>
</tbody>
</table>

**Residual:**

|                  | 1.0413 | 1.0204 |

**Fixed effects:**

|             | Estimate | Std. Error | df  | t value | Pr(>|t|) |
|-------------|----------|------------|-----|---------|----------|
| (Intercept) | 2.58104  | 0.02107    | 122 | 122.508 | < 0.0000000000000002 |
| roleM       | 0.04666  | 0.05532    | 2238 | 0.843   | 0.399    |
| roleTL      | 0.55244  | 0.02751    | 23974 | 20.231  | < 0.0000000000000002 |
| roleTLM     | 0.06018  | 0.03775    | 23357 | 1.594   | 0.111    |
| tenure1-2 years | 0.69890  | 0.02085    | 22969 | 33.526  | < 0.0000000000000002 |
| tenure3-5 years | 0.85244  | 0.02353    | 23175 | 36.227  | < 0.0000000000000002 |
| tenure6+ years | 0.80202  | 0.02780    | 23432 | 28.845  | < 0.0000000000000002 |
| job_codeENG_RES | -0.57863 | 0.07656    | 24976 | -7.558  | 0.0000000000000024 |
| job_codeENG_SRE_SWE | 0.04966  | 0.10613    | 11527 | 0.468   | 0.640    |

**Signif. codes:**  0 ‘***’  0.001 ‘**’  0.01 ‘*’  0.05 ‘.’  1

**Correlation matrix not shown by default, as p = 15 > 12.**

Use `print(summary(model), correlation=TRUE)` or `vcov(summary(model))` if you need it.

---

**2019-Q4**

**Summary of regression model:**

Linear mixed model fit by REML. *t*-tests use Satterthwaite's method ['lmerModLmerTest']

**Formula:** review_count ~ formula

**Data:** reviewer_cls_complete

**REML criterion at convergence:** 77494

**Scaled residuals:**

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>1Q</th>
<th>Median</th>
<th>3Q</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-4.5142</td>
<td>-0.5084</td>
<td>0.1124</td>
<td>0.6362</td>
<td>4.8252</td>
</tr>
</tbody>
</table>

**Random effects:**

<table>
<thead>
<tr>
<th>Groups</th>
<th>Name (Intercept)</th>
<th>Variance Std.Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>team_name</td>
<td>0.6031</td>
<td>0.7766</td>
</tr>
</tbody>
</table>

**Residual:**

|                  | 1.0413 | 1.0204 |

**Fixed effects:**

|             | Estimate | Std. Error | df  | t value | Pr(>|t|) |
|-------------|----------|------------|-----|---------|----------|
| (Intercept) | 2.58104  | 0.02107    | 122 | 122.508 | < 0.0000000000000002 |
| roleM       | 0.04666  | 0.05532    | 2238 | 0.843   | 0.399    |
| roleTL      | 0.55244  | 0.02751    | 23974 | 20.231  | < 0.0000000000000002 |
| roleTLM     | 0.06018  | 0.03775    | 23357 | 1.594   | 0.111    |
| tenure1-2 years | 0.69890  | 0.02085    | 22969 | 33.526  | < 0.0000000000000002 |
| tenure3-5 years | 0.85244  | 0.02353    | 23175 | 36.227  | < 0.0000000000000002 |
| tenure6+ years | 0.80202  | 0.02780    | 23432 | 28.845  | < 0.0000000000000002 |
| job_codeENG_RES | -0.57863 | 0.07656    | 24976 | -7.558  | 0.0000000000000024 |
| job_codeENG_SRE_SWE | 0.04966  | 0.10613    | 11527 | 0.468   | 0.640    |
job_codeENG_SRE_SYSENG  -0.20621  0.12697  15842.17421  -1.624  0.104
level4  0.25475  0.02080  22770.63803  12.249 < 0.0000000000000002 ... 
level5  0.46788  0.02542  23183.70424  18.409 < 0.0000000000000002 ... 
level6  0.39433  0.03730  23679.92475  10.571 < 0.0000000000000002 ... 
level7  -0.06477  0.05951  24114.78577  -1.126  0.260
genderFEMALE  -0.21282  0.01846  22696.34216  -11.527 < 0.0000000000000002 ... 

---

Correlation matrix not shown by default, as p = 15 > 12.
Use print(summary(model), correlation=TRUE) or 
vcov(summary(model)) if you need it

2020-Q3

"Summary of regression model:"
Linear mixed model fit by REML t-tests use Satterthwaite’s method ['lmerModLmerTest']
Formula: review_count_formula
Data: reviewer_cls_complete

REML criterion at convergence: 87667.3

Scaled residuals:
     Min      1Q Median      3Q     Max
-4.8851 -0.5112  0.1178  0.6331  4.4826

Random effects:
Groups     Name     Variance     Std.Dev.   Residual
team_name (Intercept)  0.6148  0.7841

Fixed effects:
        Estimate     Std. Error     t value     Pr(>|t|)
(Intercept)  2.61867  0.02237  17528.04461  117.078 < 0.0000000000000002 ... 
roleM         0.20219  0.06170   24787.74046   3.277  0.001051 ... 
roleTL        0.60947  0.02501  26714.14535  24.370 < 0.0000000000000002 ... 
roleTLM       0.11075  0.03487  25862.46163   3.176  0.001496 ... 
tenure<2 years 0.72667  0.02005  25523.21849  36.241 < 0.0000000000000002 ... 
tenure<5 years 0.93272  0.02285  25833.37763  40.823 < 0.0000000000000002 ... 
tenure<6 years 0.90920  0.02649  26090.71293  34.522 < 0.0000000000000002 ... 
job_codeENG_RES  -0.65399  0.07819  27274.00017  -8.364 < 0.0000000000000002 ... 
job_codeENG_SRE_SWE  0.24844  0.06398  6122.20534   3.883  0.000104 ... 
job_codeENG_SRE_SYSENG  0.04719  0.07148  9604.67811   0.660  0.509095 ... 
level4         0.19549  0.01895  25266.25133  10.316 < 0.0000000000000002 ... 
level5         0.28152  0.02454  25767.85461  11.567 < 0.0000000000000002 ... 
level6         0.15408  0.03662  26449.41656   4.207  0.000025954770 ... 
level7        -0.41174  0.06058  27263.77043  -6.796  0.000000000000011 ... 
genderFEMALE  -0.21282  0.01846  22696.34216  -11.527 < 0.0000000000000002 ... 

---

Signif. codes:  0 ‘***’  0.001 ‘**’  0.01 ‘*’  0.05 ‘.’  0.1 ‘ ’  1

Correlation matrix not shown by default, as p = 15 > 12.
Use print(summary(model), correlation=TRUE) or 
vcov(summary(model)) if you need it

[1] "Confidence interval for women:"
[1] "FEMALE estimate:"
[1] "-0.194" 
[1] "17.6"
Computing profile confidence intervals ...
Regressions for “5.1 Ownership”

Call:
```
glm(formula = formula, family = "binomial", data = results)
```

Deviance Residuals:
```
        Min       1Q    Median       3Q      Max
-1.2320 -0.8274  -0.5901  1.1238  3.1203
```

Coefficients:
```
         Estimate Std. Error z value Pr(>|z|) 
(Intercept) -0.12124   0.04262  -2.845  0.00445 **
roleM        -0.55982   0.29969  -1.868  0.06176 .
roleTLM      -1.24306   0.18868  -6.698  < 0.0001  ***
tenure1−2 years -0.37891   0.29969  -1.264  0.20748 
tenure3−5 years -0.54763   0.05370  -10.199  < 0.0001  ***
tenure6+ years -0.86277   0.18868  -4.565  0.0000  ***
tenure1−2 years -0.37891   0.29969  -1.264  0.20748 
tenure3−5 years -0.54763   0.05370  -10.199  < 0.0001  ***
tenure6+ years -0.86277   0.18868  -4.565  0.0000  ***
job_codeENG_RES 0.39217   0.30230   1.297  0.19454 
job_codeENG_SRE_SWE 1.79762   0.13664  13.156  < 0.0001  ***
job_codeENG_SRE_SYSENG 1.78688   0.25134   7.109  0.0000  ***
level4       -0.39599   0.04115  -9.622  < 0.0001  ***
level5       -0.67565   0.05635  -11.990  < 0.0001  ***
level6       -0.83587   0.11274  -7.414  0.0000  ***
level7       -2.57834   0.58860  -4.380  0.0000  ***
genderFEMALE 0.24874   0.03917   6.351  0.0000  ***
```

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 1

(Dispersion parameter for binomial family taken to be 1)

AIC: 22924

Number of Fisher Scoring iterations: 6

[1] "Confidence interval for women:"

[1] "FEMALE estimate:"

[1] "0.249"

[1] "−28.2" "−18.7" "−38.5"

[1] "Marginal and conditional R squared for model:"

R2m R2c
theoretical 0.2167955 0.2167955
delta 0.1470785 0.1470785

[1] "Percent of observations analyzed:"

[1] 34.22

[1] "Summary of regression model:"

Call:
```
glm(formula = formula, family = "binomial", data = results)
```

Deviance Residuals:
```
     Min       1Q    Median       3Q      Max
-1.0175 -0.5921  -0.4662  -0.1713  3.2962
```
Coefficients:

|                      | Estimate | Std. Error | z value | Pr(>|z|)    |
|----------------------|----------|------------|---------|------------|
| (Intercept)          | -1.30865 | 0.05618    | -23.293 | 0.0000000000000002 |
| roleM                | -0.49326 | 0.72246    | -0.683  | 0.4948     |
| roleTL               | -1.42156 | 0.24524    | -5.797  | 0.000000000000067363153 |
| tenure1–5 years      | -1.01193 | 0.08848    | -11.437 | 0.0000000000000002 |
| job_codeENG_RES      | 0.65753  | 0.47704    | 1.383   | 0.1804     |
| job_codeENG_SRE_SYSENG | -3.15395 | 0.50285    | -6.270  | 0.000000003682233 |
| genderFEMALE         | 0.26918  | 0.06036    | 4.460   | 0.000000082007892554 |

Signif. codes:  < 0.001 '***'  0.001 '*'  0.05 '.'  1 ' '  1

(Dispersion parameter for binomial family taken to be 1)

AIC: 16117

Number of Fisher Scoring iterations: 7

[1] 'Confidence interval for women:'
[1] 'FEMALE estimate:'
[1] "0.287"  
[1] "-33.3"  

Waiting for profiling to be done...

2.5 % 97.5 %
"-21.6" "46.0"

[1] 'Marginal and conditional R squared for model:'

R2m  R2c
theoretical 0.3349661 0.3349661
delta 0.1675830 0.1675830
[1] 'Percent of observations analyzed:'
[1] 34.22
[1] "Pseudo-R2:

fitting null model for pseudo-r2

LH LHnull G2 McFadden r2ML r2CU
-8043.67973886 -9831.75014200 1976.14088788 0.16933966 0.08534491 0.15307853

[1] "
[1] 'Raw odds of very_frequently_needs_team_owner_approval'
[1] 'Summary of regression model.'

Call:
glm(formula = formula, family = "binomial", data = results)

Deviance Residuals:

Min 1Q Median 3Q Max
-0.7782 -0.4110 -0.3398 -0.1515 3.7142

Coefficients:

|                      | Estimate | Std. Error | z value | Pr(>|z|)    |
|----------------------|----------|------------|---------|------------|
| (Intercept)          | -1.30865 | 0.05618    | -23.293 | 0.0000000000000002 |
| roleM                | -0.49326 | 0.72246    | -0.683  | 0.4948     |
| roleTL               | -1.42156 | 0.24524    | -5.797  | 0.000000000000067363153 |
| tenure1–5 years      | -1.01193 | 0.08848    | -11.437 | 0.0000000000000002 |
| job_codeENG_RES      | 0.65753  | 0.47704    | 1.383   | 0.1804     |
| job_codeENG_SRE_SYSENG | -3.15395 | 0.50285    | -6.270  | 0.000000003682233 |
| genderFEMALE         | 0.26918  | 0.06036    | 4.460   | 0.000000082007892554 |

Systemic Gender Inequities in Who Reviews Code

---

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

AIC: 10338

Number of Fisher Scoring iterations: 16

[1] ‘Confidence interval for women:’
[1] ‘FEMALE estimate:’
[1] ‘0.269’
[1] ‘−30.9’
[1] ‘16.2’

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Waiting for profiling to be done...

[1] ‘Marginal and conditional R squared for model:’

R2m  R2c

[1] theoretical 0.7074460 0.7074460
delta 0.3511061 0.3511061

[1] ‘Percent of observations analyzed:’
[1] 34.22

[1] ‘Pseudo-R2:’

<table>
<thead>
<tr>
<th>model</th>
<th>r2ML</th>
<th>r2CU</th>
</tr>
</thead>
<tbody>
<tr>
<td>fitting null model for pseudo-r2</td>
<td>0.35041</td>
<td>0.09826</td>
</tr>
<tr>
<td>job_codeENG_RES</td>
<td>-0.35041</td>
<td>0.09826</td>
</tr>
<tr>
<td>job_codeENG_SRE_SWE</td>
<td>0.26803</td>
<td>0.04488</td>
</tr>
<tr>
<td>job_codeENG_SRE_SYSENG</td>
<td>0.06500</td>
<td>0.05558</td>
</tr>
</tbody>
</table>

[1] ‘Summary of regression model:’

Linear mixed model fit by REML. t-tests use Satterthwaite’s method

Formula: formula

Data: results

REML criterion at convergence: 57739.9

Scaled residuals:

<table>
<thead>
<tr>
<th>Q</th>
<th>Median</th>
<th>3Q</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min 5.0439</td>
<td>0.5355</td>
<td>0.0899</td>
<td>3.7730</td>
</tr>
</tbody>
</table>

Random effects:

<table>
<thead>
<tr>
<th>Groups</th>
<th>Name</th>
<th>Variance</th>
<th>Std.Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>team_name (Intercept)</td>
<td>0.2045</td>
<td>0.4522</td>
<td></td>
</tr>
<tr>
<td>Residual</td>
<td>0.6697</td>
<td>0.8184</td>
<td></td>
</tr>
</tbody>
</table>

Fixed effects:

| Estimate | Std. Error | df  | t value | Pr(>|t|) |
|----------|------------|-----|---------|----------|
| (Intercept) | 2.90828 | 0.02186 | 16389.05453 | 133.064 | < 0.0000000000000002 |
| roleM | 0.34923 | 0.07905 | 20264.77944 | 4.418 | 0.0000000000000000 |
| roleTL | 0.47968 | 0.02133 | 21776.26036 | 22.491 | < 0.0000000000000002 |
| tenure1–2 years | 0.65076 | 0.01868 | 21377.07730 | 34.845 | < 0.0000000000000002 |
| tenure3–5 years | 0.82753 | 0.02149 | 21552.05859 | 38.507 | < 0.0000000000000002 |
| tenure6+ years | 0.74570 | 0.02426 | 21618.39557 | 30.742 | < 0.0000000000000002 |
| job_codeENG_RES | -0.35041 | 0.09826 | 21694.99341 | -3.566 | 0.000363 |
| job_codeENG_SRE_SWE | 0.26803 | 0.04488 | 4552.33017 | 5.971 | 0.0000000000000023 |
| job_codeENG_SRE_SYSENG | 0.06500 | 0.05558 | 9334.05584 | 1.169 | 0.242310 |
| level1 | 0.28612 | 0.01671 | 20742.91642 | 17.127 | < 0.0000000000000002 |
| level5 | 0.57240 | 0.02138 | 21318.55353 | 26.775 | < 0.0000000000000002 |
| level6 | 0.71572 | 0.03335 | 21620.95466 | 21.462 | < 0.0000000000000002 |
| level7 | 0.63763 | 0.06588 | 21391.98981 | 9.679 | < 0.0000000000000002 |
| genderFEMALE | -0.08034 | 0.02234 | 21239.71758 | -3.596 | 0.000324 |
| more_restrictive_ownershipTRUE | -0.09895 | 0.01975 | 4272.65838 | -5.010 | 0.00000565647 |
| genderFEMALE: more_restrictive_ownershipTRUE | -0.06955 | 0.02999 | 20865.12027 | -2.319 | 0.020412 |

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation matrix not shown by default, as p = 17 × 12.

Use print(summary(model), correlation=TRUE) or
vcov(summary(model)) if you need it

[1] ‘Summary of regression model:’
Regressions for “6.2 Manual Selection”

For manual model:
Formula: update(review_count_formula, log(manual_count) ~ .)
Data: reviewer_cls_complete[manual_count > 0]

REML criterion at convergence: 80886.9

Scaled residuals:
Min 1Q Median 3Q Max
−4.1522 −0.5249 0.1234 0.6417 4.0496

Random effects:

Groups   Name     Variance     Std. Dev.
team_name (Intercept) 0.5619 0.7496
Residual               1.1066 1.0520

Fixed effects:

(Intercept) 2.54313 0.02168 14706.91035 117.280 < 0.00000000000000002 ... 0.0266 0.235
roleM       0.12632 0.05696 22944.19963 2.218
roleTL      0.61576 0.02702 24881.10504 22.790 < 0.00000000000000002 ...
roleTLM     0.17703 0.03741 23973.53071 4.731 0.00000000000000002 ...
tenure1−2 years 0.67302 0.02088 23844.26007 32.236 < 0.00000000000000002 ...
tenure3−5 years 0.81557 0.02381 24068.37592 34.257 < 0.00000000000000002 ...
tenure6+ years 0.81059 0.02807 24312.42251 28.873 < 0.00000000000000002 ...
job_codeENG_RES −0.56775 0.07874 25490.82663 −7.210 < 0.00000000000000002 ...
job_codeENG_SRE_SWE 0.14765 0.11311 10753.16383 1.305 0.1919
job_codeENG_SRE_SYSENG −0.11263 0.12647 13684.73600 −0.891 0.3732
level4      0.20603 0.02040 23625.11108 10.098 < 0.00000000000000002 ...
level5      0.38921 0.02557 24034.88175 15.224 < 0.00000000000000002 ...
level6      0.25970 0.03794 24542.34912 6.845 < 0.000000000000007852 ...
level7      −0.25283 0.06089 24739.21820 −4.152 < 0.00000000000000002 ...
genderFEMALE −0.17743 0.01855 23475.64767 −9.568 < 0.00000000000000002 ...

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation matrix not shown by default, as p = 15 > 12.
Use print(summary(model), correlation=TRUE) or vcov(summary(model)) if you need it

1 ‘Confidence interval for women.’
1 ‘FEMALE estimate.’
1 ‘−0.177’
1 ‘16.3’
Computing profile confidence intervals ...
2.5 % 97.5 %
genderFEMALE "19.2" "13.2"
1 ‘Marginal and conditional R squared for model.’

Rm  Rc
[1] 0.1308819 0.4235623
1 ‘Percent of observations analyzed.’
1 45.23

Regressions for “6.3 Automated Selection”

For gwsq model:
Formula: update(review_count_formula, log(gwsq_count + 1) ~ .)
Data: reviewer_cls_complete[team_gwsq_percent > 0]

REML criterion at convergence: 51972.8

Scaled residuals:

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>1Q</th>
<th>Median</th>
<th>3Q</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>-4.903</td>
<td>-0.553</td>
<td>-0.122</td>
<td>0.430</td>
<td>4.553</td>
</tr>
</tbody>
</table>

Random effects:

- Groups: Name
  - Variance
  - Std. Dev.

  - `team_name (Intercept)` 1.1020 1.0498
- Residual 0.977 0.9886

Fixed effects:

| Estimate  | Std. Error | df | t value | Pr(>|t|) |
|-----------|------------|----|---------|----------|
| (Intercept) | 0.94669 | 0.03081 | 5530.05933 | 0.00000000000000002 2 |
| roleM | 0.01311 | 0.00672 | 15113.49200 | 0.195 | 0.845122 |
| roleTL | 0.31461 | 0.03242 | 15878.06645 | 9.704 | 0.00000000000000002 2 |
| roleTLM | 0.18592 | 0.04490 | 15478.49664 | 4.141 | 0.0000347835 |
| tenure~2 years | 0.34818 | 0.02458 | 15316.61719 | 14.281 | 0.00000000000000002 |
| tenure~3 years | 0.46297 | 0.02797 | 15398.91699 | 16.549 | 0.00000000000000002 |
| tenure~4 years | 0.44001 | 0.03308 | 15462.72682 | 13.301 | 0.00000000000000002 |
| job_codeENG_RES~2 years | -0.27482 | 0.13866 | 15500.03789 | -1.982 | 0.047498 2 |
| job_codeENG_RES~3 years | 0.21209 | 0.15454 | 15132.10662 | 1.372 | 0.170001 |
| job_codeENG_RES~4 years | 0.18224 | 0.16744 | 7489.49880 | 1.088 | 0.276461 |
| job_codeENG_RES~5 years | 0.08483 | 0.02372 | 15264.89375 | 3.577 | 0.000349 2 |
| job_codeENG_RES~6 years | 0.16172 | 0.03005 | 15365.15539 | 5.381 | 0.000000752 2 |
| job_codeENG_RES~7 years | 0.15656 | 0.04574 | 15583.52060 | 2.986 | 0.002834 2 |
| job_codeENG_RES~8 years | 0.02168 | 0.07296 | 15748.60887 | 0.297 | 0.766402 |
| genderFEMALE~2 years | -0.07081 | 0.00797 | 15753.36916 | 0.297 | 0.766402 |

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation matrix not shown by default, as p = 15 > 12.

Use `print(summary(model), correlation=TRUE)` or `vcov(summary(model))` if you need it.

### Regressions for “6.4 Incomplete Reviews”

Call:

```
lm(formula = formula, data = results)
```

Residuals:

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>1Q</th>
<th>Median</th>
<th>3Q</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>-0.3900</td>
<td>-0.0584</td>
<td>0.0498</td>
<td>0.1052</td>
<td>0.3853</td>
</tr>
</tbody>
</table>

Coefficients:

| Estimate  | Std. Error | df | t value | Pr(>|t|) |
|-----------|------------|----|---------|----------|
| (Intercept) | 0.809147 | 0.00243 | 328.819 | 0.00000000000000002 2 |
| tenure~2 years | 0.006306 | 0.00282 | 2.232 | 0.02663 2 |
| tenure~3 years | 0.004117 | 0.00359 | 4.131 | 0.00003561308 2 |
| tenure~4 years | 0.0018116 | 0.00368 | 4.923 | 0.00000485759 2 |
| roleM | -0.065433 | 0.00699 | -9.357 | 0.00000000000000002 2 |
| roleTL | 0.016889 | 0.00349 | 4.837 | 0.0000132296 2 |
| roleTLM | -0.035689 | 0.00492 | -10.990 | 0.00000000000000002 2 |
| level1 | 0.004967 | 0.00280 | 1.674 | 0.09414 |
| level2 | 0.002846 | 0.00384 | 0.841 | 0.40029 |
| level3 | -0.031085 | 0.00483 | -6.429 | 0.00000000013 2 |
| level4 | -0.107535 | 0.00765 | -14.040 | 0.00000000000000002 2 |
| job_codeENG_RES | -0.070816 | 0.00797 | -8.881 | 0.00000000000000002 2 |

### Summary of regression model:

#### Call:

```r
glm(formula = update(review_count$formula, did_local_rosie_review - . - (1 | team_name)), family = "binomial", data = reviewer_cls_complete)
```

#### Deviance Residuals:

<table>
<thead>
<tr>
<th>Min</th>
<th>1Q</th>
<th>Median</th>
<th>3Q</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.7464</td>
<td>-1.0771</td>
<td>-0.6409</td>
<td>1.0725</td>
<td>2.0804</td>
</tr>
</tbody>
</table>

#### Coefficients:

| Estimate | Std. Error | z value | Pr(>|z|) |
|----------|------------|---------|---------|
| (Intercept) | -1.47833 | 0.03896 | -37.943 < 0.0000000000000002 *** |
| roleM | 0.02728 | 0.10384 | 0.263 | 0.79276 |
| roleTL | 0.66147 | 0.15038 | 13.135 < 0.0000000000000002 *** |
| roleTLM | 0.16040 | 0.06801 | 2.359 | 0.01834 |
| tenure2−2 years | 0.98847 | 0.04215 | 23.451 < 0.0000000000000002 *** |
| tenure5−3 years | 1.34005 | 0.04642 | 28.866 < 0.0000000000000002 *** |
| tenure6+ years | 1.62874 | 0.05346 | 30.428 < 0.0000000000000002 *** |
| job_codeENG_SRE_SWE | -0.25109 | 0.12602 | -1.993 | 0.04432 |
| job_codeENG_SRE_SYSENG | -0.5284 | 0.16651 | 3.229 | 0.07419 |
| job_codeENG_RES | -0.30836 | 0.17927 | -1.675 | 0.09384 |
| level4 | 0.24915 | 0.03958 | 6.294 | 0.0000000000309 *** |
| level5 | 0.41651 | 0.04769 | 8.735 < 0.0000000000000002 *** |
| level6 | 0.24039 | 0.06937 | 3.465 | 0.00053 |
| level7 | -0.08790 | 0.10902 | -0.806 | 0.42008 |
| genderFEMALE | -0.31251 | 0.03567 | -8.741 < 0.0000000000000002 *** |

#### Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 1

### Marginal and conditional R squared for model:

<table>
<thead>
<tr>
<th>R2m</th>
<th>R2c</th>
</tr>
</thead>
<tbody>
<tr>
<td>theoretical</td>
<td>0.1515667</td>
</tr>
<tr>
<td>delta</td>
<td>0.127086</td>
</tr>
</tbody>
</table>

#### Percent of observations analyzed:

| 45.42 |

#### Fitting null model for pseudo-R2:

| lihNull | G2 | McFadden | r2ML | r2CU |

---

**Significant codes:**

- **0** ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 1
And with initial assignments:

Summary of regression model:

Call:
glm(formula = update(review_count_formula, initially_assigned_a_rosie ~ ., (1 | team_name)), family = "binomial", data = reviewer_cls_complete)

Deviance Residuals:
Min 1Q Median 3Q Max
-1.7443 -1.0330 -0.5966 1.0700 2.1840

Coefficients:

| Estimate | Std. Error | z value | Pr(>|z|) |
|----------|------------|---------|----------|
| (Intercept) | -1.63605 | 0.04056 | -40.335 < 0.0000000000000002 |
| roleM | 0.03010 | 0.10429 | 0.289 | 0.77285 |
| roleTL | 0.60323 | 0.04996 | 12.075 < 0.0000000000000002 |
| tenure~2 years | 1.02929 | 0.04347 | 23.476 < 0.0000000000000002 |
| tenure~5 years | 1.45321 | 0.04753 | 30.147 < 0.0000000000000002 |
| tenure~6+ years | 1.71535 | 0.05433 | 31.571 < 0.0000000000000002 |
| job_codeENG_RES | -0.01528 | 0.12639 | -1.209 | 0.22663 |
| job_codeENG_SRE_SWE | 0.13093 | 0.16207 | 0.808 | 0.41918 |
| job_codeENG_SRE_SYSENG | 0.65225 | 0.18376 | 3.550 < 0.0000000000000002 |
| level4 | 0.25723 | 0.04039 | 6.369 | 0.0000000000000001 |
| level5 | 0.46130 | 0.04829 | 9.553 < 0.0000000000000002 |
| level6 | 0.39701 | 0.06986 | 5.683 | 0.000000001326196 |
| level7 | 0.23660 | 0.11032 | 2.145 | 0.03197 |
| genderFEMALE | -0.26499 | 0.03599 | -7.362 | 0.0000000000000001 |

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

AIC: 32002

Number of Fisher Scoring iterations: 4

"Confidence interval for women:"

"FEMALE estimate:

"- 0.265"

"23.3"

Waiting for profiling to be done...

2.5 % 97.5 %

"28.5" "17.7"

"Marginal and conditional R squared for model:"

\[ \text{R}^2 \text{ (theoretical)} = 0.1673333, \text{R}^2 \text{ (delta)} = 0.1402371 \]

"Percent of observations analyzed:"

45.42

"Pseudo-R2:"

<table>
<thead>
<tr>
<th>fitting null model for pseudo-r2</th>
<th>r2ML</th>
<th>r2CU</th>
</tr>
</thead>
<tbody>
<tr>
<td>null model</td>
<td>0.15986</td>
<td>0.15986</td>
</tr>
<tr>
<td>lH</td>
<td>0.34696426526</td>
<td>0.09789787</td>
</tr>
<tr>
<td>lHNull</td>
<td>0.12578225</td>
<td>0.16845416</td>
</tr>
</tbody>
</table>

Regressions for "6.5.1 Authorship Signal."

CLS AUTHORED

Predict the log of the number of CLs authored by an engineer.

"Summary of regression model:"

Linear mixed model fit by REML. t-tests use Satterthwaite’s method ['lmerModLmerTest']

Formula: author_count_formula

Data: author_cls_complete

REML criterion at convergence: 93101

Scaled residuals:
Min 1Q Median 3Q Max
Random effects:
Groups   Name Variance Std.Dev.  
  team_name (Intercept) 0.4794 0.6924  
  Residual             0.7446 0.8629  

Fixed effects:  
  Estimate Std. Error   df t value Pr(>|t|)  
  (Intercept)  3.4436 0.0165 11297.33 05294 204.875 < 0.0000000000000002  
  roleM        -0.5116 0.0450 0856 0.57411 -11.349 < 0.0000000000000002  
  roleTL       0.1061 0.0399 928 03280.6920986 5.307 0.000000112  
  roleTLM      -0.5241 0.0291 071 3206.8816625 -18.028 < 0.0000000000000002  
  tenure\text{–}2 years -0.2499 0.014 834 32112.0236273 16.852 < 0.0000000000000002  
  tenure\text{–}5 years 0.2403 0.0170 846 32345.166089 14.071 < 0.0000000000000002  
  tenure\text{+} years 0.2028 0.0203 736 32533.6621882 9.957 < 0.0000000000000002  
  job\_codeENG\_RES -0.5170 0.0623 445 33554.9769890 -8.294 < 0.0000000000000002  
  job\_codeENG\_SWE 0.2450 0.0592 307 9659.5298891 2.753 0.0059195  
  job\_codeENG\_SRE\_SYSENG -0.0897 0.1011 231 13262.5674224 -0.839 0.374168  
  level4         -0.0005 0.0144 836 31829.5776216 -0.035 0.972067  
  level5        -0.0467 0.0184 948 32281.0901007 -2.520 0.0004311  
  level6        -0.2758 0.0282 547 32789.2859018 -9.860 < 0.0000000000000002  
  level7        -0.6921 0.0486 769 33048.8567975 -14.219 < 0.0000000000000002  
  genderFEMALE  -0.1856 0.0137 163 31577.0353374 -13.537 < 0.0000000000000002  

Signif. codes:  0 ‘***’  0.001 ‘**’  0.01 ‘*’  0.05 ‘.’  0.1 ‘ ’  1

Correlation matrix not shown by default, as p = 15 > 12.  
Use print(summary(model), correlation=TRUE) or vcov(summary(model)) if you need it

[1] 'Confidence interval for women:'
[1] 'FEMALE estimate:'
[1] '-0.186'
[1] '16.9'

Computing profile confidence intervals ... 
  2.5 % 97.5 %  
  genderFEMALE "19.1" "14.7"

[1] 'Marginal and conditional R squared for model:'
  R2m  R2c
  [1] 0.4256727 0.4175371
[1] 'Percent of observations analyzed:'
[1] 43.85

Same regression, but include whether the user a fig user as a factor[1] 'Summary of regression model:'
Linear mixed model fit by REML. t-tests use Satterthwaite's method ['.lmerModLmerTest']
Formula: update(author_count_formula, 
  . ~ fig_user)
  Data: author_clis_complete

REML criterion at convergence: 70537.5

Scaled residuals:  
  Min 1Q Median 3Q Max
  -3.1432 -0.6358 0.0401 0.6709 4.8554

Random effects:  
Groups   Name Variance Std.Dev.  
  team_name (Intercept) 0.1284 0.3583  
  Residual             0.4076 0.6384  

Fixed effects:  
  Estimate Std. Error   df t value Pr(>|t|)  
  (Intercept)   3.6605 0.0117 61765.595830 311.045 < 0.0000000000000002  
  roleM         -0.1853 0.0330 36 31930.987428 -5.597 0.00000000022012  
  roleTL        0.0849 0.0144 24 33861.724086 5.893 0.0000000038338  
  roleTLM       -0.2052 0.0212 06 33251.503745 -9.679 < 0.0000000000000002  
  tenure\text{–}2 years 0.1877 0.0108 03 33369.852236 17.376 < 0.0000000000000002  
  tenure\text{–}5 years 0.2017 0.0124 50 33532.890331 16.203 < 0.0000000000000002  
  tenure\text{+} years 0.1727 0.0148 05 33646.129452 11.667 < 0.0000000000000002  
  job\_codeENG\_RES -0.2079 0.0440 59 31032.930353 -4.720 0.000002568095  
  job\_codeENG\_SWE 0.1349 0.0563 89 8891.043955 2.393 0.0167
Systemic Gender Inequities in Who Reviews Code

job_code ENG_SRE_SYSENG 0.047389 0.065951 13273.031468 0.719 0.4724
level4 0.021654 0.010545 33097.867309 2.054 0.0400
level5 0.005074 0.013368 33458.720927 0.380 0.7043
level6 -0.090431 0.020477 33703.137830 -4.592 0.0000004440817
level7 -0.224461 0.035306 33655.860220 -6.358 0.0000000001280
genderFEMALE -0.117528 0.010057 32821.215726 -11.686 <0.0000000000000002
fig_userUnknown -2.306546 0.013325 33839.708410 -173.104 <0.0000000000000002
fig_userUser 0.126258 0.008685 33879.908016 14.538 <0.0000000000000002

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation matrix not shown by default, as p = 17 > 12.
Use print(summary(model), correlation=TRUE) or vcov(summary(model)) if you need it

[1] 'Confidence interval for women.'
[1] 'FEMALE estimate.'

Computing profile confidence intervals ...

2.5 % 97.5 %
genderFEMALE "12.8" "9.3"

[1] 'Marginal and conditional R squared for model.'
  Rm  R2c
[1] 0.5100387 0.6274416

[1] 'CLS SIZES-----------------------------------------------'

Predict the log of an engineer's median CL size.

[1] 'Summary of regression model.'

Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']

Data: author_cls_complete

REML criterion at convergence: 95272.3

Scaled residuals:

  Min 1Q Median 3Q Max
-4.5256 -0.5731 0.0048 0.5865 7.5617

Random effects:

  Groups   Name     Variance Std.Dev.
           team_name (Intercept) 0.2414  0.4913
           Residual             0.8531  0.9236

Fixed effects:

  Estimate Std. Error df t value Pr(>|t|)
(Intercept)  3.54594  0.01561  15113.94879 227.206 <0.0000000000000002
roleM        -0.37026  0.04768  31978.95833  -7.766 0.0000000000008350
roleTLM     -0.24808  0.02077  33881.54759  -11.942 <0.0000000000000002
tenure1-2 years -0.17473  0.01555  33470.75940  -11.238 <0.0000000000000002
tenure3-5 years -0.27203  0.01788  33628.42002  -15.218 <0.0000000000000002
tenure6+ years  -0.31222  0.02129  33724.43026  -14.667 <0.0000000000000002
job_codeENG_RES  0.51218  0.06328  30227.26214    8.104 0.000000000000000553
job_codeENG_SWE  0.63293  0.07984  8435.53570   -7.928 0.000000000000002514
job_codeENG_SRE_SYSENG  -0.69125  0.09379  12829.49285  -7.370 0.0000000000000008002
level4       -0.01935  0.01521  33209.24501   -1.272  0.2033
level5       -0.08698  0.01927  33565.19816   -4.514 0.000000638379996310
level6       -0.02447  0.02948  33773.19388   -0.830   0.4066
level7      -0.10193  0.05076  33701.35666   -2.008  0.0447
genderFEMALE  0.07092  0.01443  32928.62537    4.914 0.0000000896277184974

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation matrix not shown by default, as p = 15 > 12.
Use print(summary(model), correlation=TRUE) or vcov(summary(model)) if you need it

[1] 'Confidence interval for women.'
[1] 'FEMALE estimate.'
[1] '0.071'
Computing profile confidence intervals ... 2.5 % 97.5 %
genderFEMALE '−4.4' '−10.4'

'Marginal and conditional R squared for model:'

R2m  R2c

0.05157682 0.2607703

'Percent of observations analyzed:' 43.85

Same regression, but include whether the user a fig user as a factor | Summary of regression model:

Linear mixed model fit by REML. t-tests use Satterthwaite’s method ['lmerModLmerTest']

Formula: update(size_formula, . ~ . + fig_user)

Data: author_clsc_complete

REML criterion at convergence: 94968.3

Scaled residuals:

Min  1Q Median  3Q Max
−4.4791 −0.5760 −0.0046 0.5727 7.6841

Random effects:

Groups   Name    Variance  Std.Dev.
          team_name (Intercept) 0.2342 0.4839
          Residual          0.8467 0.9201

Fixed effects:

                   Estimate Std. Error   df  t value Pr(>|t|)
(Intercept)       3.471826  0.016679 17751.303027 208.154  < 2e-16
roleM             −0.339085  0.047556 32006.651500 −7.145  0.000e+00
roleTL            −0.240657  0.020869 33881.603424 −11.632  0.000e+00

roleTLM           −0.525137  0.030460 33351.661888 −17.240  0.000e+00

tenure1−2 years   −0.164700  0.015515 33501.587325 −10.730  0.000e+00

tenure3−5 years   −0.251428  0.017875 33650.572560 −14.066  0.000e+00

tenure6+ years    −0.292812  0.021253 38391.797088 −13.739  0.000e+00

job_codeENG_RES   0.546225  0.062929 30130.901680  8.680  0.000e+00
job_codeENG_SRE_SWE −0.648881  0.079180 8391.797088 −8.195  0.000e+00

job_codeENG_SRE_SYSENG −0.668998  0.093115 12809.679464 −7.185  0.000e+00

level4            −0.020587  0.015149 33233.108846 −1.359  0.174
level5            −0.013050  0.019195 33580.203167 −6.436  0.000e+00
level6            −0.004769  0.029388 33781.449328 −0.162  0.871
level7            −0.058268  0.050680 33710.297779 −1.150  0.250

genderFEMALE      0.097600  0.014453 32947.590069  6.753  0.000e+00
fig_userUnknown   −0.095453  0.019098 33771.538638 −4.998  0.000e+00
fig_userUser      0.194453  0.012451 33826.608382 15.616  0.000e+00

---

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation matrix not shown by default, as p = 17 > 12.

Use print(summary(model), correlation=TRUE) or vcov(summary(model)) if you need it

'Confidence interval for women:'

'FEMALE estimate:'

'0.098'

'−10.3'

Computing profile confidence intervals ... 2.5 % 97.5 %
genderFEMALE '−7.2' '−13.4'

'Marginal and conditional R squared for model:'

R2m  R2c

0.06047314 0.2604384

'Percent of observations analyzed:' 43.85

Regressions for “6.5.2 Reviews Signal”

Linear mixed model fit by REML. t-tests use Satterthwaite’s method ['lmerModLmerTest']

Formula: formula

Data: data

Regression for “7 A Small Intervention and Its Evaluation”

**Regressions for “7 A Small Intervention and Its Evaluation”**

**Systemic Gender Inequities in Who Reviews Code**

**REML criterion at convergence:** 1342850

**Scaled residuals:**

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>1Q</th>
<th>Median</th>
<th>3Q</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-10.1069</td>
<td>-0.5418</td>
<td>-0.0093</td>
<td>0.5183</td>
<td>8.6983</td>
</tr>
</tbody>
</table>

**Random effects:**

- **user** (Intercept) 7734.7 87.95
- **gender** FEMALE 0.002

**Fixed effects:**

**Estimate** | **Std. Error** | **df** | **t value** | **Pr(>|t|)**
---|---|---|---|---|
(Intercept) 106.43078 | 13.57066 | 40.84630 | 7.843 | 0.0000000000000112 ...

genderFEMALE | -16.92217 | 0.09072 | 167954.99979 | -186.537 | < 0.0000000000000002 ...
reviewer_weightOFF | -2.62367 | 0.09072 167954.99979 | -28.921 | < 0.0000000000000002 ...
genderFEMALE: reviewer_weightOFF | 3.15817 | 0.12829 167954.99977 | 24.617 | < 0.0000000000000002 ...

**Signif. codes:** 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

**Correlation of Fixed Effects:**

- **(Intr)**
- **gender** FEMALE rv_OFF
- **role** M
- **job_code** ENG_SRE
- **tenure** 6 + years
- **pericodic reviewer assigned rosie after**
- **periodic reviewer assigned rosie after: gender** FEMALE

**Formula:**

- Linear mixed model fit by REML
- **Formula:**
- **data:** melted

**REML criterion at convergence:** 133767.6

**Scaled residuals:**

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>1Q</th>
<th>Median</th>
<th>3Q</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-2.46732</td>
<td>-0.86066</td>
<td>0.03415</td>
<td>0.83702</td>
<td>2.91873</td>
</tr>
</tbody>
</table>

**Random effects:**

- **Groups** Name  Variance  Std. Dev.
- **team_id** (Intercept) 0.02762 0.1662
- **user** (Intercept) 7734.7 87.95

**Estimate** | **Std. Error** | **df** | **t value** | **Pr(>|t|)**
---|---|---|---|---|
(Intercept) 0.152865 | 0.021680 | 100879.72410 | 7.342 | 0.00000000000005098 ...
roleM | -0.003487 | 0.009784 | 108465.99927 | -0.356 | 0.72153|
roleTL | 0.002269 | 0.004509 | 106167.90467 | 18.831 | < 0.0000000000000002 ...
roleTLM | 0.014228 | 0.005873 | 108299.23992 | 2.423 | 0.01538 |
tenure<2 years | 0.161790 | 0.021466 | 108895.64967 | 7.360 | < 0.0000000000000002 ...
tenure3-5 years | 0.306425 | 0.021466 | 108996.30492 | 14.284 | < 0.0000000000000002 ...
tenure>5 years | 0.355099 | 0.021534 | 108874.86271 | 14.518 | < 0.0000000000000002 ...
job_codeENG_OTHER | -0.320698 | 0.004824 | 708122.90395 | -67.783 | < 0.0000000000000002 ...
job_codeENG_SRE | 0.197814 | 0.009908 | 15954.36220 | 1.996 | 0.04590 |
level4 | 0.107053 | 0.004776 | 109017.13127 | 22.414 | < 0.0000000000000002 ...
level5 | 0.146129 | 0.005478 | 108837.87106 | 26.750 | < 0.0000000000000002 ...
level6 | 0.155657 | 0.007145 | 107325.21835 | 18.896 | < 0.0000000000000002 ...
level7 | 0.867913 | 0.008659 | 106101.55420 | 7.031 | < 0.0000000000000002 ...
periodic reviewer assigned rosie after | 0.081764 | 0.002901 | 100354.86634 | 17.845 | < 0.0000000000000002 ...
genderFEMALE | -0.084574 | 0.008459 | 107262.14921 | -17.406 | < 0.0000000000000002 ...
periodic reviewer assigned rosie after: genderFEMALE | 0.019850 | 0.008635 | 100334.86635 | 2.992 | 0.00278 |

**Signif. codes:** 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

**Correlation matrix not shown by default, as p > 17 > 12.**

**Use print(summary(model), correlation=TRUE) or vcov(summary(model)) if you need it**

[1] "Marginal and conditional R squared for model:"
Regressions for “B.1 Equity in Readability Reviewers Load”

Linear mixed model fit by REML, t-tests use Satterthwaite’s method ['lmerModLmerTest']
Formula: update(review_count_formula, log(non_rp_review_count) ~ .)  
Data: reviewer_cls_complete[non_rp_review_count > 0]

Regression coefficients:

| Estimate | Std. Error | df | t value | Pr(>|t|) |
|----------|------------|----|---------|----------|
| (Intercept) | 2.66670 | 0.02193 | 14207.39767 | 121.607 | < 0.0000000000000002 ... |
| roleM | 0.16840 | 0.05669 | 22886.29463 | 2.971 | 0.00297 .. |
| roleTL | 0.59118 | 0.02695 | 24834.43390 | 21.939 | < 0.0000000000000002 ... |
| roleTLM | 0.16806 | 0.03727 | 23910.75413 | 4.509 | 0.00000655175638 ... |
| tenure1−2 years | 0.69254 | 0.02077 | 23746.56241 | 33.342 | < 0.0000000000000002 ... |
| tenure3−5 years | 0.84619 | 0.02370 | 23964.10122 | 35.706 | < 0.0000000000000002 ... |
| tenure6+ years | 0.85465 | 0.02796 | 24214.20369 | 30.570 | < 0.0000000000000002 ... |
| job_codeENG_RES | −0.53788 | 0.07880 | 25696.43593 | −6.826 | < 0.0000000000000002 ... |
| job_codeENG_SRE_SWE | 0.14579 | 0.11458 | 10964.36475 | 1.272 | 0.20325 |
| job_codeENG_SRE_SYSENG | −0.12362 | 0.12800 | 13827.90685 | −0.966 | 0.33415 |
| level4 | 0.17988 | 0.02030 | 23535.00818 | 8.862 | < 0.0000000000000002 ... |
| level5 | 0.35150 | 0.02546 | 23943.66349 | 13.805 | < 0.0000000000000002 ... |
| level6 | 0.23314 | 0.03784 | 24470.96531 | 6.162 | < 0.000000000000729822 ... |
| level7 | −0.26944 | 0.06058 | 24732.54484 | −4.447 | < 0.0000000872693674 ... |
| genderFEMALE | −0.17755 | 0.18454 | 23399.79204 | −9.620 | < 0.0000000000000002 ... |

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation matrix not shown by default, as p = 15 > 12.
Use print(summary(model), correlation=TRUE) or vcov(summary(model)) if you need it

Regressions for “B.2 Who Has Readability”

Linear mixed model fit by REML, t-tests use Satterthwaite’s method ['lmerModLmerTest']
Formula: formula  
Data: per_engineer

R2m R2c
[1] 0.1608784 0.2695136

"Percent of observations analyzed:"
[1] 62.1
REML criterion at convergence: 18026

Scaled residuals:

<table>
<thead>
<tr>
<th>Min</th>
<th>Q1</th>
<th>Median</th>
<th>Q3</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3.3157</td>
<td>-0.7036</td>
<td>-0.0566</td>
<td>0.7281</td>
<td>3.2473</td>
</tr>
</tbody>
</table>

Random effects:

- Groups Name Variance Std.Dev.
  - team_name (Intercept) 0.02155 0.1468
  - Residual 0.08917 0.2986

Fixed effects:

- Estimate Std. Error df t value Pr(>|t|)
  - (Intercept) 0.019438 0.005341 22273.70140 3.639 0.000274 ***
  - tenure<2 years 0.191198 0.005281 31960.87959 36.761 < 0.00000000000000002 ***
  - tenure<5 years 0.349752 0.005862 32139.88283 59.666 < 0.00000000000000002 ***
  - tenure>=5 years 0.472042 0.006792 32272.066329 69.497 < 0.00000000000000002 ***
  - roleM 0.000200 0.013508 30872.287885 0.742 0.458216
  - roleTL 0.001320 0.009085 31754.427652 0.147 0.883461
  - level4 0.064705 0.014952 31634.231073 13.066 < 0.00000000000000002 ***
  - level5 0.107370 0.016189 32050.55974 17.315 < 0.00000000000000002 ***
  - level6 0.131747 0.019123 32390.033925 14.440 < 0.00000000000000002 ***
  - level7 0.133217 0.014513 32496.463087 9.163 < 0.00000000000000002 ***
  - job_codeENG_RES -0.160247 0.017667 28885.698678 -9.066 < 0.00000000000000002 ***
  - job_codeENG_SRE_SWE -0.117974 0.015641 8349.108254 -7.543 < 0.00000000000000002 ***
  - job_codeENG_SRE_SYSENG -0.180316 0.018461 4203.518315 -9.765 < 0.00000000000000002 ***
  - genderFEMALE -0.035816 0.044557 31520.206610 -0.785 0.438460

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation matrix not shown by default, as p = 15 > 12.
Use print(summary(model), correlation=TRUE) or vcov(summary(model)) if you need it.

[1] "Confidence interval for women:
FEMALE estimate : 
-0.036
[1] "3.5"

Computing profile confidence intervals ...
2.5 % 97.5 %
genderFEMALE "4.4" "2.7"
[1] "Marginal and conditional R squared for model:
R2m R2c
1.0 0.4714947
[1] "Percent of observations analyzed:"
67.04
[1] "For a large slice of engineers:
gender can satisfy no readability"
1: FEMALE 0.3704071
2: MALE 0.2953972
[1] "******** Ability to satisfy own readability needs "
[1] "Mean satisfaction:"
[1] 0.4304607
[1] "Summary of regression model:"
Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
Formula: formula
Data: per_engineer

REML criterion at convergence: 24598.5

Scaled residuals:

<table>
<thead>
<tr>
<th>Min</th>
<th>Q1</th>
<th>Median</th>
<th>Q3</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2.8641</td>
<td>-0.7461</td>
<td>-0.0178</td>
<td>0.7623</td>
<td>2.9976</td>
</tr>
</tbody>
</table>

Random effects:

- Groups Name Variance Std.Dev.
  - team_name (Intercept) 0.0146 0.1208
  - Residual 0.1266 0.3558

Fixed effects:
Estimate Std. Error df t value Pr(>|t|)
\((\text{Intercept})\)
0.027320 0.006319 22876.482148 4.523 0.000015449053514
\(\text{tenure}_{-2} \text{ years}\)
0.250521 0.006484 28706.423325 38.637 < 0.0000000000000002
\(\text{tenure}_{-5} \text{ years}\)
0.434927 0.007339 28779.622198 59.260 < 0.0000000000000002
\(\text{tenure}_{+} \text{ years}\)
0.564051 0.008612 28815.031377 65.495 < 0.0000000000000002
\(\text{roleM}\)
0.003734 0.019621 27810.971371 0.190 0.849
\(\text{roleTL}\)
0.075676 0.007995 28834.933604 9.466 < 0.0000000000000002
\(\text{roleTLM}\)
0.130935 0.007746 28748.157573 16.904 < 0.0000000000000002
\(\text{level4}\)
0.168221 0.011784 28823.994708 14.275 < 0.0000000000000002
\(\text{level7}\)
0.184010 0.021450 28795.091983 8.578 < 0.0000000000000002
\(\text{job\_codeENG\_RES}\)
-0.168995 0.023521 22184.365682 -7.185 < 0.0000000000000069
\(\text{job\_codeENG\_SRE\_SWE}\)
-0.112025 0.017418 6859.964269 -6.432 < 0.0000000000134792
\(\text{job\_codeENG\_SRE\_SYSENG}\)
-0.138220 0.022837 14475.184193 -6.052 < 0.0000000001462100
\(\text{genderFEMALE}\)
-0.050641 0.005701 28325.697040 -8.883 < 0.000000000000002

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation matrix not shown by default, as p = 15 > 12.
Use print(summary(model), correlation=TRUE) or vcov(summary(model)) if you need it

[1] "Confidence interval for women:'
[1] "FEMALE estimate:'
[1] "-0.051"
[1] "4.9"

Computing profile confidence intervals ...
2.5 % 97.5 %
genderFEMALE "6.0" "3.9"

[1] "Marginal and conditional R squared for model:'
R2m R2c
[1 , ] 0.296246 0.368977

[1] "Percent of observations analyzed:'
67.04

[1] "For a large slice of engineers' gender can_satisfy_no_readability"
1: MALE 0.31050692
2: FEMALE 0.3974359

[1] "Do the number of readability languages used per CL differ?" gender V1
1: FEMALE 1.039493
2: MALE 1.039415

Wilcoxon rank sum test with continuity correction

data: nlangs by gender
W = 339768875571, p-value = 0.3601
alternative hypothesis: true location shift is not equal to 0

Regressions for "B.3 Equity in the Readability Process"

"........... Regressions predicting pre-process outcomes.............."

log(cls_since_first_cl) ~ (1 | username) + readability_lang + level + job_code + role + tenure + gender
<environment: 0x17536e2d8f40>

[1] "Summary of regression model:
Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
Formula: formula

Data: data[get(dv) > 0]

REML criterion at convergence: 34524.6

Scaled residuals:
Min 1Q Median 3Q Max
-3.3896 -0.5417 0.0417 0.6104 3.7451

Random effects:
Groups Name Variance Std. Dev.
username (Intercept) 0.236 0.4858

**Systemic Gender Inequities in Who Reviews Code**

**Random effects:**
- job_code: OTHER
- role: M 0.2056 0.1316 1.1035 3.8438 0.1182 0.0493

**Formula:**
- log (loc_since_first_cl) ~ (1 | username) + readability_lang + level + job_code + role + tenure + gender

**Residual**
- 1.018 1.0088

**Fixed effects:**

| Estimate | Std. Error | df | t value | Pr(>|t|) |
|----------|------------|----|---------|----------|
| (Intercept) | -3.5846 | 0.02986 | 11109.72010 | 120.178 | <0.000000000000002 |
| readability_langdart | -1.52549 | 0.06230 | 11210.38275 | -24.487 | <0.000000000000002 |
| readability_langgo | -2.37062 | 0.04245 | 11177.31909 | -55.847 | <0.000000000000002 |
| readability_langjava | -0.1046 | 0.02670 | 11138.02751 | -3.918 | 0.000087972745754 |
| readability_langjavascript | -0.22876 | 0.05593 | 10860.84222 | -4.090 | 0.0000437591602 |
| readability_langkotlin | -2.28379 | 0.10817 | 11157.49664 | -21.112 | <0.000000000000002 |
| readability_langobjc | 0.63683 | 0.08847 | 11233.79317 | 7.198 | 0.000000000000064 |
| readability_langpython | -0.08100 | 0.04121 | 10776.91151 | -1.966 | 0.04935 |
| readabilitylangswig | -3.05574 | 1.10148 | 7224.00363 | -2.774 | 0.00555 |
| readability_langtypescript | -1.97674 | 0.09772 | 11057.35528 | -49.773 | <0.000000000000002 |
| level4 | -0.05520 | 0.02629 | 10399.23231 | -2.100 | 0.03579 |
| level5 | -0.07039 | 0.03661 | 10289.38371 | -1.923 | 0.05453 |
| level6 | -0.11281 | 0.07615 | 10452.64873 | -1.481 | 0.13855 |
| level7 | 0.43436 | 0.22319 | 10828.55413 | 1.946 | 0.05166 |
| job_codeENG_OTHER | -0.22834 | 0.05414 | 10347.10472 | -4.218 | 0.000024906749124 |
| job_codeENG_SRE | -0.35212 | 0.06390 | 10174.30747 | -5.197 | 0.0000000206261104 |
| job_codeOTHER | -0.05067 | 0.08381 | 10945.61809 | -0.714 | 0.000000001034 |
| roleM | 0.20560 | 0.13161 | 11035.84380 | 1.562 | 0.11828 |
| roleTL | 0.03836 | 0.04592 | 10636.33260 | 0.835 | 0.40356 |
| roleTLM | -0.04695 | 0.10404 | 10588.74375 | -0.451 | 0.65180 |
| tenure=2 years | 0.26843 | 0.02738 | 11166.60262 | 9.803 | <0.000000000000002 |
| tenure=5 years | 0.57780 | 0.03516 | 18757.31624 | 16.433 | <0.000000000000002 |
| tenure=+ years | 0.81961 | 0.04947 | 10621.11790 | 16.569 | <0.000000000000002 |
| genderFEMALE | -0.02604 | 0.02864 | 9984.95354 | -0.909 | 0.36328 |

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation matrix not shown by default, as p = 24 > 12.

Use print(summary(model), correlation=TRUE) or vcov(summary(model)) if you need it

1. 'Marginal and conditional R squared for model.'
2. Rm
3. [ ] 0.378681 0.4956445
4. [ ] 'Percent of observations analyzed.'
5. [ ] 9.15

log (loc_since_first_cl) ~ (1 | username) + readability_lang + level + job_code + role + tenure + gender

<environment: 0x1757710d3210>
1. 'Overall outcome median: 5663'
2. 'Summary of regression model'.
3. Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']

**Formula:**
- Data: data[get(dv) > 0]

**REML criterion at convergence:** 43192.2

**Scaled residuals:**
- Min
- 3Q
- Median
- 0.4255
- 0.0735
- 0.5233
- 4.0058

**Random effects:**
- Groups
- Name
- Variance
- Std. Dev.
- username (Intercept) 0.4831 0.6951
- Residual 2.2528 1.5009

**Fixed effects:**

| Estimate | Std. Error | df | t value | Pr(>|t|) |
|----------|------------|----|---------|----------|
| (Intercept) | 8.74653 | 0.04413 | 11091.42036 | 198.205 | <0.000000000000002 |
| readability_langdart | -2.26592 | 0.09218 | 11193.74902 | -24.582 | <0.000000000000002 |
| readability_langgo | -3.35350 | 0.06322 | 11167.57495 | -52.760 | <0.000000000000002 |
| readability_langjava | -0.13249 | 0.03947 | 11130.54894 | -3.357 | 0.000791 |
| readability_langjavascript | -0.49707 | 0.08269 | 10919.98251 | -6.011 | 0.00000000019 |
| readability_langkotlin | -3.08689 | 0.15987 | 11171.45774 | -19.390 | <0.000000000000002 |
| readability_langobjc | 0.31910 | 0.13073 | 11205.74519 | 2.440 | 0.014682 |
| readability_langpython | -0.33036 | 0.06895 | 10833.98206 | -5.420 | 0.0000000607 |
Recomputing criterion at convergence: 18791.4

Scaled residuals:
Min  Q1 Median  Q3 Max
-4.8955 -0.4895 0.0475 0.5704 3.6542

Random effects:
Groups Name Variance Std.Dev.
username (Intercept) 0.06565 0.2562
Residual 0.24368 0.4936

Fixed effects:

| Estimate   | Std. Error | df | t value | Pr(>|t|) |
|------------|------------|----|---------|---------|
| (Intercept)| 2.459887   |    | 165.834| <2.2e-16|
| readibility_langdagart| 0.3712360 | 0.0309192 | 11187.6639277 | 12.007 < 0.0000000000000002 |
| readibility_langgo| 0.873653 | 0.0210646 | 11161.4951234 | 41.476 < 0.0000000000000002 |
| readibility_langjava| -0.1932065 | 0.0132471 | 11126.5744106 | -14.585 < 0.0000000000000002 |
| readibility_langjavascript| 0.2565029 | 0.0277371 | 10777.0633503 | 9.248 < 0.0000000000000002 |
| readibility_langkotlin| -0.379017 | 0.0536832 | 11175.8284000 | -7.061 < 0.00000000000000175 |
| readibility_langobjc| 0.4096941 | 0.0439210 | 11233.9811598 | 9.328 < 0.0000000000000002 |
| readibility_langpython| 0.222909 | 0.0264315 | 10728.1173535 | 10.912 < 0.0000000000000002 |
| readibility_langswig| -1.7542696 | 0.5443516 | 6747.4596220 | -3.223 < 0.00128 |
| readibility_langtypescript| 0.2925637 | 0.0197016 | 11012.7447204 | 14.840 < 0.0000000000000002 |

level1 0.1939520 0.030732 | 10467.8444203 | -1.067 < 0.25859
level5 0.0074958 0.0182058 | 10363.2941805 | -0.412 < 0.68055
level6 0.5521289 0.0541313 | 80359.59027 | 1.233 < 0.21748
level7 0.2646650 0.0647674 | 10497.0739971 | 1.377 < 0.16847

job_codeENG_OTHER| 0.0012663 | 0.0269237 | 10412.9885431 | 0.047 < 0.96249
job_codeENG_SRE| 0.0563626 | 0.0317846 | 10266.1733684 | 1.773 < 0.07621
job_codeOTHER| 0.0765518 | 0.0414470 | 10987.0499295 | 1.842 < 0.06548
roleM 0.1482247 0.0653920 | 11036.7794711 | -2.267 < 0.02343
roleTL 0.2037870 0.0228269 | 10711.5168601 | -1.042 < 0.29741
**Systemic Gender Inequities in Who Reviews Code**

---

### Formula

Log of `log_in_process` - `(1 | username) + readability_lang + level + job_code + role + tenure + gender

**Environment:** 0x17536deab678

**Fixed effects:**

| Estimate       | Std. Error | df | t value | Pr(>|t|) |
|----------------|------------|----|---------|----------|
| (Intercept)    | 7.9714558  | 0.017906 | 11133.5883394 | 445.095 < 0.00000000000000002 ... |
| readability_langdart | 0.3235197  | 0.0373540 | 11203.0162155 | 8.656 < 0.00000000000000002 ... |
| readability_langgo  | 0.6215197  | 0.0255258 | 11178.5532187 | 24.349 < 0.00000000000000002 ... |
| readability_langjava | 0.0133565  | 0.0160138 | 11152.5192719 | 0.833 0.40491 |
| readability_langjavascript | 0.2346714  | 0.0335345 | 10941.7540827 | 6.998 0.00000000000000275 ... |
| readability_langkotlin | -0.3332202 | 0.0648579 | 11187.2150382 | -5.138 0.00000002821532 ... |
| readability_langobjc | 0.1072183  | 0.0504464 | 11221.9053578 | 2.021 0.04327 ... |
| readability_langpython | 0.0005041  | 0.0247116 | 10894.847223 | 3.258 0.00113 ... |
| readability_langswig | -3.4587697 | 0.6600619 | 7948.3006609 | -5.240 0.00000001461263 ... |
| readability_langtypescript | -0.0466077 | 0.0238138 | 11089.6397088 | -1.957 0.05035 ... |
| level4          | 0.0014004  | 0.0157725 | 10615.9875008 | 0.089 0.92925 |
| level5          | -0.0183938 | 0.0219607 | 10531.0009751 | -0.837 0.40242 ... |
| level6          | 0.0838269  | 0.0456767 | 10665.2549137 | 1.835 0.06650 ... |
| level7          | -0.1924733 | 0.1338451 | 10926.8244785 | -1.438 0.15045 ... |
| job_codeENG_OTHER | -0.0008942 | 0.0327251 | 10585.3614579 | -0.027 0.97820 ... |
| job_codeENG_SRE  | 0.0138467  | 0.0383745 | 10443.2217184 | 0.361 0.71823 ... |
| job_codeOTHER   | 0.1247987  | 0.0500171 | 11016.6013937 | 2.495 0.01261 ... |
| roleM           | -0.1301442 | 0.0789257 | 11078.4895685 | -1.649 0.09919 ... |
| roleTL          | -0.0283463 | 0.0275561 | 10791.1700008 | -1.029 0.30366 ... |
| roleTLM         | -0.1935935 | 0.0623962 | 10753.3818372 | -3.103 0.00192 ... |
| tenure1~2 years | 0.0927199  | 0.0164245 | 11175.7126572 | 5.645 0.00000001689809 ... |
| tenure3~5 years | 0.0951305  | 0.0210929 | 10877.7954441 | 4.510 0.000000654862174 ... |
| tenure6+ years  | 0.0038202  | 0.0297016 | 10776.1180387 | 0.129 0.89766 ... |
| genderFEMALE    | -0.003186  | 0.0171856 | 10306.4362923 | -0.077 0.93884 ... |

**Signif. codes:** 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 1

### Correlation matrix not shown by default, as p = 24 > 12.

Use `print(summary(model), correlation=TRUE)` or `vcov(summary(model))` if you need it.

---

**Proc. ACM Hum.-Comput. Interact., Vol. 7, No. CSCW1, Article 94. Publication date: April 2023.**
REML criterion at convergence: 34763.4

Scaled residuals:

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>Q1</th>
<th>Median</th>
<th>Q3</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-3.2938</td>
<td>-0.5382</td>
<td>0.0575</td>
<td>0.5972</td>
<td>4.7906</td>
</tr>
</tbody>
</table>

Random effects:

- Groups Name Variance Std. Dev.
- username (Intercept) 0.3763 0.6134
- Residual 0.9505 0.9749

Fixed effects:

|                     | Estimate | Std. Error | df | t value | Pr(>|t|) |
|---------------------|----------|------------|----|---------|----------|
| (Intercept)         | 2.7293   | 0.03891    | 11039| 15848 | 88.303   |
| readabilty_langdart| 0.42239  | 0.06401    | 10885| 61051 | 6.599    |
| readabilty_langgo  | 0.28404  | 0.04360    | 10936| 82492 | 6.514    |
| readabilty_langjava | 0.18581 | 0.02745    | 10896| 67541 | 6.768    |
| readabilty_langjavascript | 0.32994 | 0.05732    | 10128| 9729  | 5.756    |
| readabilty_langkotlin | -0.93942 | 0.11179    | 10950| 93616 | -8.040   |
| readabilty_langobjc | 0.88793  | 0.09080    | 11116| 15290 | 9.779    |
| readabilty_langpython | 0.39994 | 0.04237    | 10172| 9987  | 9.440    |
| readabilty_langswig | 0.69133  | 1.10612    | 4354| 6986  | 0.625    |
| level4              | -0.0621  | 0.02727    | 10304| 14120 | -2.061   |
| level5              | -0.1749  | 0.03801    | 10130| 60964 | -4.598   |
| level6              | -0.14564 | 0.07899    | 10217| 8198  | -1.844   |
| level7              | -0.04869 | 0.23025    | 10547| 1245  | -0.211   |
| job_codeENG_OTHER   | -0.32146 | 0.05655    | 10203| 77918 | -5.685   |
| job_codeENG_SRE     | -0.11522 | 0.06632    | 10064| 29194 | -1.737   |
| job_codeOTHER       | -0.63170 | 0.08652    | 10197| 15504 | -7.301   |
| roleM               | -0.02412 | 0.13810    | 10834| 89396 | -0.175   |
| roleTL              | 0.08133  | 0.04756    | 10624| 63962 | 1.710    |
| roleTLM             | -0.19332 | 0.10828    | 10506| 59924 | -1.785   |
| tenure1-2 years     | 0.46983  | 0.02830    | 11102| 98759 | 22.965   |
| tenure3-5 years     | 0.84180  | 0.03644    | 10676| 59070 | 23.103   |
| tenure6+ years      | 0.77139  | 0.05130    | 10490| 64995 | 15.036   |
| genderFEMALE        | 0.07960  | 0.02979    | 9749| 57338 | 2.672    |

Signif. codes:  < 0.001 ‘***’  0.001 ‘**’  0.01 ‘*’  0.05 ‘.’  1 ‘ ’  0

Correlation matrix not shown by default, as p = 24  12.

Use print(summary(model), correlation=TRUE) or vcov(summary(model)) if you need it
Percent of observations analyzed:

\[
\text{log}(\text{loc\_unsent}) - (1 | \text{username}) + \text{readability\_lang} + \text{level} + \text{job\_code} + \text{role} + \text{tenure} + \text{gender}
\]

Overall outcome median: 3353

Summary of regression model: Linear mixed model fit by REML. t-tests use Satterthwaite’s method ‘lmerModLmerTest’

Formula: formula

Data: data[get(dv) > 0]

REML criterion at convergence: 43536.8

Scaled residuals:

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>1Q</th>
<th>Median</th>
<th>3Q</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-4.775</td>
<td>-0.505</td>
<td>0.0765</td>
<td>0.5869</td>
<td>5.7974</td>
</tr>
</tbody>
</table>

Random effects:

<table>
<thead>
<tr>
<th>Groups</th>
<th>Name</th>
<th>Variance</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>(Intercept)</td>
<td>0.6256</td>
<td>0.7909</td>
</tr>
<tr>
<td>Residual</td>
<td></td>
<td>2.2920</td>
<td>1.5139</td>
</tr>
</tbody>
</table>

Fixed effects:

| Estimate | Std. Error | df | t value | Pr(>|t|) |
|----------|------------|----|---------|----------|
| (Intercept) | 7.37990 | 0.04588 | 11005.52425 | 160.832 | < 0.0000000000000002 |
| readability\_langdart | 0.35637 | 0.09525 | 11057.12233 | 3.741 | 0.000184 |
| readability\_langgo | 0.03461 | 0.06500 | 11038.15435 | 0.533 | 0.594390 |
| readability\_langjava | 0.35337 | 0.04087 | 10992.38079 | 8.645 | < 0.0000000000000002 |
| readability\_langjavascript | 0.31206 | 0.08551 | 10643.27641 | 3.649 | 0.000264 |
| readability\_langkotlin | -1.10777 | 0.16634 | 11040.81288 | -6.600 | 0.00000000000267 |
| readability\_langobjc | 0.75536 | 0.13492 | 11114.87221 | 5.599 | 0.0000000000021099 |
| readability\_langpython | 0.34717 | 0.06321 | 10567.34142 | 5.493 | 0.0000000000045226 |
| readability\_langswig | -0.63555 | 1.67099 | 6292.89573 | -0.379 | 0.704594 |
| readability\_langtypescript | -0.41248 | 0.06107 | 10889.57062 | -6.755 | 0.00000000000115 |
| level4 | -0.01663 | 0.04035 | 10304.60245 | -0.462 | 0.644262 |
| level5 | -0.12830 | 0.05620 | 10183.36220 | -2.279 | 0.022675 |
| level6 | 0.03458 | 0.11682 | 10324.76147 | 0.296 | 0.757238 |
| level7 | 0.24263 | 0.34087 | 10677.57628 | 0.712 | 0.476600 |
| job\_codeENG\_OTHER | -0.42583 | 0.08397 | 10255.96786 | -5.071 | 0.00000000001640 |
| job\_codeENG\_SRE | -0.34789 | 0.09814 | 10098.11986 | -3.545 | 0.000395 |
| job\_codeOTHER | -0.75755 | 0.12824 | 10876.89587 | -5.907 | 0.000000003579 |
| roleM | 0.22588 | 0.20467 | 10859.48921 | 1.094 | 0.274031 |
| roleTL | 0.01767 | 0.07841 | 10561.34591 | 0.251 | 0.501873 |
| roleTLM | -0.44936 | 0.16026 | 10577.77940 | -2.804 | 0.005059 |
| tenure\(\leq\)2 years | 0.67579 | 0.04202 | 11060.66911 | 16.081 | < 0.0000000000000002 |
| tenure\(>\)2 \(\leq\)5 years | 0.84917 | 0.05398 | 10650.70037 | 15.732 | < 0.0000000000000002 |
| tenure\(>\)=6 years | 0.77376 | 0.07595 | 10533.23785 | 10.188 | < 0.0000000000000002 |
| genderFEMALE | 0.05752 | 0.04403 | 9862.06967 | 1.306 | 0.191489 |

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation matrix not shown by default, as p = 24 > 12.

Use print(summary(model), correlation=TRUE) or vcov(summary(model)) if you need it

Confidence interval for women:

FEMALE estimate: 0.058

Confidence interval for men:

FEMALE estimate: -0.59

Computing profile confidence intervals ... 2.5 % 97.5 %

Marginal and conditional R squared for model:

R2m R2c
[1,] 0.05936492 0.2610516

Percent of observations analyzed:

9.05

Regressions for “B.3.4 When are women stalling in the readability process?”
**Linear mixed model fit by REML**

**t-tests use *Satterthwaite’s method* ['lmerModLmerTest']

**Formula** update(formula ~ . + readability_long + gender)

**Data** per_person_summary

**REML criterion at convergence**: 27976.4

---

### Fixed effects:

|                          | Estimate | Std. Error | df | t value | Pr(>|t|) |
|--------------------------|----------|------------|----|---------|----------|
| (Intercept)              | 0.592046 | 0.008633   | 19618 | 68.370 | < 0.000000000000002 |
| readability_longdart     | -0.073765 | 0.021250 | 19782 | -3.471 | 0.000519 |
| readability_longgo       | -0.080617 | 0.015357 | 19427 | -5.199 | 0.000000024400370 |
| readability_longjava      | -0.072425 | 0.010855 | 19352 | -6.728 | 0.000000000000001 |
| readability_long knotlin  | -0.189497 | 0.022731 | 19695 | -8.337 | 0.000000000000002 |
| readability_longpython    | -0.073558 | 0.027416 | 19804 | -2.683 | 0.007502 |
| readability_longtype script | -0.099550 | 0.032693 | 19823 | -3.102 | 0.002125 |
| level4                   | 0.010381 | 0.008218 | 17401 | 1.265 | 0.206548 |
| level5                   | -0.006080 | 0.012252 | 17334 | -0.295 | 0.768372 |
| level6                   | 0.054553 | 0.027193 | 18543 | 2.006 | 0.044858 |
| level7                   | -0.038651 | 0.077385 | 19212 | -0.496 | 0.617962 |
| job_codeENG_OTHER        | 0.039513 | 0.017322 | 17474 | 2.281 | 0.025318 |
| job_codeENG_SRE          | -0.041489 | 0.021808 | 17506 | -1.919 | 0.055066 |
| job_codeOTHER            | 0.118307 | 0.025514 | 15367 | 4.657 | 0.000000035315199 |
| roleM                    | -0.055232 | 0.051866 | 19027 | -1.026 | 0.304748 |
| roleTL                   | 0.036467 | 0.019760 | 19194 | 1.846 | 0.064977 |
| roleTLM                  | -0.005084 | 0.042754 | 18919 | -0.119 | 0.905546 |
| tenure < 2 years         | -0.122994 | 0.080202 | 19775 | -13.737 | 0.000000000000002 |
| tenure <= 5 years        | -0.098824 | 0.011798 | 18584 | -8.376 | 0.000000000000002 |
| tenure > 6 years         | -0.111231 | 0.017516 | 18579 | -6.350 | 0.0000000219988 |
| genderFEMALE             | -0.074723 | 0.015334 | 18298 | -4.873 | 0.000000010669047 |
| readability_longdart:genderFEMALE | -0.019098 | 0.046626 | 19793 | -0.410 | 0.682108 |
| readability_longgo:genderFEMALE | 0.016600 | 0.038053 | 19781 | 0.436 | 0.646267 |
| readability_longjava:genderFEMALE | 0.023941 | 0.021759 | 19359 | 1.010 | 0.271240 |
| readability_long knotlin:genderFEMALE | 0.038752 | 0.046616 | 19715 | 0.252 | 0.792902 |
| readability_longpython:genderFEMALE | -0.080601 | 0.060945 | 19182 | -1.340 | 0.246143 |
| readability_longtype script:genderFEMALE | 0.016664 | 0.047482 | 19806 | -0.233 | 0.819502 |
| readability_longtype script:genderFEMALE | 0.059519 | 0.033816 | 19223 | 1.759 | 0.078657 |
| readability_longtype script:genderFEMALE | 0.013641 | 0.035194 | 19407 | 0.427 | 0.695747 |

---

**Signif. codes:**  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

**Correlation matrix not shown by default, as p = 32 > 12.**

**Use print(summary(model), correlation=TRUE) or vcov(summary(model)) if you need it**

**Fit warnings:**

fixed-effects model matrix is rank deficient so dropping 1 column / coefficient

[1] "Confidence interval for women:
[1] "FEMALE estimate:"
[1] "-0.075"

**Computing profile confidence intervals ...**

2.5 % 97.5 %

genderFEMALE - 10.5 - 4.5

[1] "Marginal and conditional R squared for model:"

R2m  R2c

[1, ] 0.0284533 0.0435003
[1] "Percent of observations analyzed:" 14.97
[1] "Summary of regression model:" Linear mixed model fit by REML t-tests use Satterthwaite’s method ['lmerModLmerTest']

**Formula** update(formula ~ . + log(cls_since_first_cl) + team_overlap + median_comments + readability_long + gender)

**Data** per_person_summary

**REML criterion at convergence**: 27976.4

---

### Scaled residuals:

<table>
<thead>
<tr>
<th>Min 1Q Median 3Q Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.5109 -0.9280 -0.5659 1.0102 1.7791</td>
</tr>
</tbody>
</table>

**Random effects:**
Received January 2022; revised July 2022; accepted November 2022