I Z A Institute
of Labor Economics
Initiated by Deutsche Post Foundation

DISCUSSION PAPER SERIES

IZA DP No. 17077

Decoding Gender Bias: The Role of Personal Interaction

Abdelrahman Amer
Ashley C. Craig
Clémentine Van Effenterre

JUNE 2024

IZA Institute of Labor Economics

Initiated by Deutsche Post Foundation

DISCUSSION PAPER SERIES

IZA DP No. 17077

Decoding Gender Bias: The Role of Personal Interaction

Abdelrahman Amer
University of Toronto

Ashley C. Craig
Australian National University and IZA

Clémentine Van Effenterre
University of Toronto and IZA

JUNE 2024

ISSN: 2365-9793

IZA – Institute of Labor Economics

Schaumburg-Lippe-Straße 5–9
53113 Bonn, Germany

Phone: +49-228-3894-0
Email: publications@iza.org

www.iza.org

ABSTRACT

Decoding Gender Bias: The Role of Personal Interaction*

Subjective performance evaluation is an important part of hiring and promotion decisions. We combine experiments with administrative data to understand what drives gender bias in such evaluations in the technology industry. Our results highlight the role of personal interaction. Leveraging 60,000 mock video interviews on a platform for software engineers, we find that average ratings for code quality and problem solving are 12 percent of a standard deviation lower for women than men. Half of these gaps remain unexplained when we control for automated measures of coding performance. To test for statistical and taste-based bias, we analyze two field experiments. Our first experiment shows that providing evaluators with automated performance measures does not reduce the gender gap. Our second experiment removed video interaction, and compared blind to non-blind evaluations. No gender gap is present in either case. These results rule out traditional economic models of discrimination. Instead, we show that gender gaps widen with extended personal interaction, and are larger for evaluators educated in regions where implicit association test scores are higher.

JEL Classification: C93, D83, J16, J71, M51

Keywords: discrimination, gender, coding, experiment, information

Corresponding author:
Clémentine Van Effenterre
University of Toronto
150 St. George Street
Toronto M5S3G7, ON
Canada
E-mail: c.vaneffenterre@utoronto.ca

* This paper greatly benefited from discussions and helpful comments from Iris Bohnet, Katherine Coffman, Rahul Deb, Nicole Fortin, Dylan Glover, Maria Guadalupe, Sara Heller, Peter Hull, Kory Kroft, Corinne Low, Michelle Lowry, Marion Monnet, Peter Morrow, Roland Rathelot, Alexandra Roulet, Nina Roussille, Heather Sarsons and Basit Zafar. We also thank seminar participants at the NBER Entrepreneurship SI, Ridge WELAC, SOLE, EALE, SEA, Sciences Po, CREST, INSEAD, SSE, Bocconi, the Queen’s & Toronto Workshop, LAGV, Monash, PSE and ANU. We are grateful to the Pivotal fund, the NBER Digitization Program and the Russell Sage Foundation for financial support. Matthew Jarvis-Cross and Sabrina Wang provided outstanding research assistance. This project received IRB approval at the University of Michigan, the University of Toronto, and the Australian National University. The second experiment was pre-registered on December 14, 2022, ID: AEARCTR-0009816, a pre-analysis plan was uploaded on the AEA RCT Registry website on January 3, 2023, and updated on February 17, 2023. A previous version of this paper was entitled “Does Better Information Reduce Gender Discrimination in the Technology Industry?”.

1 Introduction

With subjective judgment comes the potential for discriminatory bias. We study such bias in performance evaluations, which are essential determinants of hiring and promotion decisions in high-skilled industries. In these industries, the hiring process often involves multiple stages of screening, including aptitude tests and simulations of work tasks. Despite this rich information set, evaluators and decision-makers still have imperfect knowledge about the future performance of applicants. Ultimately, they evaluate those applicants by combining the data they have with judgment from in-person interactions. This leaves scope for many distinct types of bias.

Economists have long studied bias as a potential barrier preventing underrepresented groups from entering high-paying occupations (Bertrand and Duflo, 2017). They have proposed two main theories, which differ in their implications: information-based ("statistical") discrimination (Phelps, 1972; Arrow, 1973; Coate and Loury, 1993; Craig and Fryer, 2019) and taste-based discrimination (Becker, 1957). More recently, some work has attempted to focus specifically on the role of implicit bias, stereotypes, and incorrect beliefs (Bertrand et al., 2005; Bordalo et al., 2016; Bohren et al., 2023). The policy implications differ between these theories, yet it has proved hard both to quantify discrimination and identify the mechanisms that underlie it. Measuring discrimination requires us to hold fixed the performance of candidates, and compare decisions for individuals of different groups who perform objectively just as well. Separately identifying different types of biases further requires us to measure beliefs, or to observe changes in decisions as more or less information becomes available.

This paper provides evidence highlighting the crucial role of face-to-face interactions in triggering gender bias. Guided by a model of discrimination, we combine administrative data and two experiments in the context of coding evaluations in the technology sector. This is an industry where women are chronically underrepresented (Ashcraft et al., 2016). Our experiments allow us to control the information seen by evaluators—specifically, whether or not participants interact face-to-face, and whether gender is revealed via the first name of the coder—while holding constant the performance of a fixed set of candidates on real coding tasks in a natural labor market setting. We are therefore able to separately identify the different sources of gender discrimination, and contextual factors that trigger implicit gender bias.

We begin by analyzing administrative data from 60,000 mock interviews conducted on an online peer-to-peer platform based in the United States. The platform offers job applicants the opportunity to practice for technical interviews, during which they solve computer programming challenges. Mirroring real interviews, the evaluator on the platform can interact with the coder via video. These types of "coding interviews" are a common part of the recruitment of computer programmers (Behroozi et al., 2020).

We first document that female coders receive lower ratings than men. These gender gaps in assessments of coding ability and problem solving correspond to around 12 percent of a standard deviation. They are largely independent of the gender of the evaluator, and remain when we control for interviewees' and evaluators' levels of education, experience, and self-reported preparation.

We develop a model of discrimination in the spirit of Lundberg and Startz (1983) to help us understand the gender gaps we see in these performance evaluations. The model highlights four potential mechanisms, and allows us to identify tests of each mechanism that motivate two field experiments. First, evaluators may statistically discriminate against women if they believe them to be worse coders than men. Second, there may be differences in skills between men and women. Third, evaluators may engage in taste-based discrimination against women. Finally, implicit bias may manifest when evaluators and coders interact face-to-face.

Our first experiment asks whether evaluators incorrectly believe that women write worse code, and statistically discriminate against them based on this false belief. To evaluate this, we study the randomized roll-out of objective code quality measures, which were made available to pairs of participants before ratings were chosen. If voluntarily activated, these "unit tests" assessed whether the code executed without errors, and produced correct answers to test cases. We show that performance on these tests is predictive of future labor market outcomes. The availability of the tests increased ratings across the board without reducing the gender gap. This result allows us to reject the hypothesis that the gender gap in performance evaluations is driven by incorrect beliefs. Once available, the unit tests also allow us to show that there remains a gender gap even when we condition on this "objective" measure of code quality.

Our second experiment tests our remaining potential explanations for the gender gaps on the platform. First, we examine whether the gaps are explained by differences in code quality that were not measured by the unit tests. Second, we test for taste-based or rational statistical discrimination by revealing gender-disclosing names of applicants to evaluators. Third, we ask whether personal interactions themselves triggered bias. To test these hypotheses, we had a stratified random sample of code originally written on the platform reevaluated by computer science students. These new evaluators were not identical to platform users, but we ensured that the populations matched closely. We also show that there is a robust correlation between ratings by the two sets of evaluators. Video interaction was not included, so that evaluators could focus on evaluation of the code itself. The evaluation setting otherwise mirrored the platform. We randomized whether the coder's gender was revealed by their first name (the "non-blind" condition), or only initials were shown so that gender was masked (the "blind" condition). An important and novel feature of our experiment is that the same code blocks from the platform are evaluated in all evaluation contexts. This allows us to rule out differences in performance across conditions due to phenomena such as stereotype threat (Spencer et al., 2016).

To test for differences in code quality between men and women, we compare evaluations of code written by each gender in the "blind" condition. With the aim of isolating variation in quality which is not captured by the unit tests, we stratified on performance as measured by those tests. Because evaluators could not discern gender, their judgments could neither be affected by taste-based nor statistical discrimination. Thus, any gender gap in evaluations reflects unbiased assessments of code quality. However, we find no such gap in the gender-blind evaluations, despite there being a gender gap when the same set of code blocks were evaluated on the platform. This suggests that differences in code quality—or stylistic differences that are penalized for women (Vedres and Vasarhelyi, 2019)—do not explain the gender gap on the platform. It also implies that the gap cannot easily be explained by rational statistical discrimination, because this would rely on the existence of a true gender gap in quality.

We next test for taste-based discrimination. In the spirit of seminal work by Goldin and Rouse (2000), we do this by comparing "blind" to "non-blind" evaluations. Because treatment was randomized, and the set of scripts evaluated is precisely the same in each treatment, we can identify evaluator bias without confounding differences. We find no evidence that women are treated differently when gender was made visible and salient using their first names. This suggests that the gender gap on the platform is not explained by traditional taste-based discrimination, because revealing the coder's gender should suffice to trigger any such preference-based bias.

These findings suggest that the gender gap hinges on interpersonal dynamics. The difference between our "non-blind" condition and interviews on the platform is that participants on the platform interact via video. Eliminating this interaction eliminated the gender gap. Our experimental sample and the evaluation experience were otherwise chosen to closely match the platform. Two mechanisms could explain this result: Either a form of bias comes into play only during face-to-face interactions (e.g., implicit bias), or there are gender differences in coding duration or verbal performance which are not reflected in the written code but which nonetheless enter the ratings for code quality and problem solving. For example, women's communication styles could be perceived as less efficient. However, we find no gender differences in coding duration, and women do not receive lower ratings for communication.

We provide two analyses which provide more direct evidence that a form of implicit bias is at play. First, the gender gaps in ratings are twice as large among evaluators who graduated from an institution in geographic areas with more prejudice towards women in science, as measured by Implicit Association Tests (IAT). Second, the gender gap widens when personal interaction is longer, which provides more opportunity for gender differences in mannerisms to be noticed. Specifically, a fifteen minute increase in the duration of the overall session leads a widening of the gender gap by 4 percent of a standard deviation, controlling for the candidate's own objective performance and coding duration. This is consistent with implicit stereotypes becoming more pronounced as evaluators become more fatigued, as opposed to sustained contact reducing prejudice as suggested by contact theory (Allport et al., 1954). These two tests suggest that gender gaps arise specifically in settings where personal interaction is extended and evaluators may be predisposed to implicit bias.

This paper contributes to the literature on bias in hiring and promotion decisions. A relative strength of our approach is that we can identify underlying mechanisms. For example, audit studies are often used to measure bias because they can vary perceived group membership of candidates while holding fixed job-relevant characteristics of the individual (Neumark et al., 1996; Bertrand and Mullainathan, 2004; Neumark, 2012; Kroft et al., 2013; Farber et al., 2016; Kline et al., 2022). These studies generally cannot distinguish mechanisms such as statistical or taste-based discrimination, because they lack a way of measuring beliefs about group differences.[1] A rare exception is Bohren et al. (2023), who do so for an online Q&A forum by studying how discrimination changes as prior evaluations become available.

Our results provide a potential reason why correspondence studies on gender have found mixed results (Bertrand and Duflo, 2017), with recent papers showing that most firms exhibit negligible bias against female names (Kline et al., 2022, 2023). This result is in line with our blinding experiment, which suggests that revealing a candidate's gender via their first name does not trigger gender bias. However, our evidence on the importance of personal interaction suggests that this misses important biases which are only introduced when the evaluator and candidate interact face-to-face.

Other studies compare blind and non-blind evaluations of candidates (Goldin and Rouse, 2000; Breda and Ly, 2015; Breda and Hillion, 2016; Terrier, 2020; Lavy and Sand, 2018; Mocanu, 2023), but this design alone also fails to isolate the mechanisms that drive bias. Nor can it establish whether disparities stem from decision-maker bias at all, as opposed to candidates performing differently across evaluation conditions due to phenomena such as stereotype threat (Spencer et al., 2016).

We also contribute to recent literature specifically on the use of recruitment tools to address gender gaps in hiring. Mocanu (2023) finds that women's relative evaluation scores and the female share of new hires increased after "impartial" recruitment practices were mandated in the Brazilian public sector. In the technology sector, Feld et al. (2022) and Avery et al. (2023) show that providing recruiters with more information can reduce gender gaps in settings without live interaction: Feld et al. (2022) focus on skill measures which are not directly coding-related, while Avery et al. (2023) study the introduction of an AI hiring score. The results of our Experiment I contrast with these other papers, which suggests that the context and content of additional information about performance are critical to understanding its effect on bias.

Our ability to compare contexts with and without personal interaction is an important feature that distinguishes our study. Face-to-face interaction is a critical part of many hiring processes, and the fact that bias is more likely to emerge during such interactions opens the possibility that institutions could be redesigned to reduce bias. For example, personal interaction with candidates could be separated from code eval-

[1] An additional advantage relative to audit studies is that we are able provide real code excerpts to evaluators, which eliminates deception (Kessler et al., 2019, 2022).

uations. Our results align with work by Petrie and Greenberg (2023), who demonstrate that video interaction changes bargaining behavior more than text-based chat in a setting where there are gender gaps in bargaining outcomes that disappear when communication is disallowed. They are also in line with the literature on implicit discrimination and stereotypes, which emphasizes the role of unconscious mental associations and contextual factors in the formation of discriminatory behaviors (Bertrand et al., 2005; Reuben et al., 2014; Bordalo et al., 2016; Carlana, 2019; Hangartner et al., 2021; Dupas et al., 2021; Barron et al., 2022; Cunningham and de Quidt, 2022; Kessler et al., 2022; Bellemare et al., 2023; Handlan and Sheng, 2023). While IAT scores have been shown to predict bias in settings with sustained interaction (Carlana, 2019), we provide new evidence that awareness of a coder's gender via their first name is not enough to trigger bias, and that bias is amplified by longer interactions.

More broadly, we contribute toward understanding the factors limiting the progression of women in high-paying occupations (Bertrand et al. 2010, Goldin 2014, Roussille 2020), especially in the technology industry (Terrell et al., 2017; Murciano-Goroff, 2018; Miric and Yin, 2020; Boudreau and Kaushik, 2020; Feld et al., 2022; Avery et al., 2023). One contributing factor may be how information about performance is assessed. Empirical research faces the challenge that ability and performance are usually hard to quantify in high-skilled labor markets. Unlike previous studies, we have access to a problem-specific objective measure of performance for computer programmers, as well as code which we can have reevaluated in a blind setting. Combined with experimental variation, this helps us measure both performance and bias.

The remainder of the paper is structured as follows. We describe the institutional context and present our descriptive analysis of administrative data in Section 2. The model is presented in Section 3. The results of the two experiments are presented in Sections 4 and 5. We bring together all of our results in a discussion of all potential mechanisms in Section 6, and conclude in Section 7.

2 Administrative Data: Face-to-Face Coding Interviews

Technology companies such as Google and Atlassian conduct face-to-face coding interviews to screen job applicants. These interview questions are to a large degree standardized and aim to test applicants' understanding of basic coding concepts. Such hiring practice have led to the proliferation of test prep comapnies such as Coderbyte, HackerRank, and Pramp. Similar to test prep companies offering practice questions for the SAT, these companies offer a kit of coding interviews to prepare candidates during the screening process. Our data comes from one of several platforms that have been developed for this purpose. In particular, we leverage almost 60,000 mock interviews where users are paired to practice face-to-face coding interviews.

We use administrative data from the platform for both our experiments. The data allow us to observe a variety of metrics regarding coders' performance and evaluations. What distinguishes our data from other peer evaluation datasets is the ability to observe and link users' written code to their evaluations. This provides us with an unusual opportunity to hold fixed performance, and thereby rule out behavioral responses due to phenomena such as stereotype threat in our second experiment. We also link the platform data to individual-level labor market information from LinkedIn via Revelio labs. Figure A4 presents a detailed timeline of data coverage.

2.1 Interactions on the Platform

A user's experience on the platform begins when they sign up and provide information about their background and experience, including their proficiency with available programming languages. They then schedule an interview during one of many fixed time slots, with the platform suggesting slots which already have users with similar profiles. When the time arrives, users within the time slot are matched.[2]

The paired users interview each other in turn. Depending on the language, self-reported ability and experience of the users, one of 31 different coding problems is assigned. The interviewee solves the coding problem in an online text editor that both sides see while the users communicate via live video chat (see Figure A1). Once the interview finishes, the interviewer and interviewee swap roles. At the conclusion of their interaction, each user rates the other on their coding quality, communication, hireability, likability, and problem solving.

The platform therefore provides an environment where realistic time-constrained tasks are performed and evaluated. This allows the study of gender gaps in performance evaluations in a high-skilled labor market setting where face-to-face interac-

[2] Users are paired based on their similarity scores using Edmunds' Blossom algorithm, which chooses a matching that maximizes the total of similarity scores of paired users.

tions can be of high importance. In fact, users' online reviews underscore the importance of such interactions. For example, one user writes:

> *"I realized early that my biggest challenge wasn't the coding problems themselves: it was staying focused while solving them out loud in front of an interviewer with time pressure. [The platform] was perfect for practicing in an environment much more like the real interview."*

2.2 Description of The Platform Data

Our first experiment (Section 4) occurred during the period of covered by the first part of our dataset, which contains 60,513 interviews covering December 18, 2015 to April 18, 2018. Candidates participate in as many practice interviews as they like. Each time, they are paired with a different counterpart. During this period, users had participated in 12 sessions so far on average.

Descriptive statistics for the population of users are shown in Table D1. Participants are high-skilled, and the vast majority graduated in STEM fields. Almost 45 percent had Master's degrees, and nearly all others had a Bachelor's degree (see Figure A2). Two thirds of users had computer science degrees, with most of the rest spread across engineering, mathematics, statistics and the hard sciences (see Figure A3). Sixteen percent of users were female. Consistent with evidence from Murciano-Goroff (2018), we find that women declare lower levels of preparation on average.

Our second experiment (Section 5) uses platform data from a more recent period, from April 2018 to May 2021. Crucially, this more recent dataset contains the full code script written by interviewees on the platform. This allows us to provide real, user-written code for evaluation in Experiment II. For details about the sample and code block descriptive statistics, see Appendix Tables E1 and E2. In addition, we use first and last names from the more recent period to link the platform data with individual-level LinkedIn information using the Revelio labs database. This provides us with future labor market outcomes for participants. This is discussed further in Section 4.3.

2.3 Gender Gaps in Evaluations of Code Quality

Figure 1 and Table D2 show the gender gaps in evaluations on the platform between January 2016 to July 2017. This is before any interventions, so that the information that evaluators see about coders remains consistent throughout the period. Women received 12 percent of a standard deviation lower ratings for code quality and problem solving on average, with no difference in scores for communication.

These gender gaps remain largely unchanged when we control for the interviewee's and interviewer's level of education, years of experience and self-declared preparation level. They also persist when we add date fixed effects to take into account any changes in composition as the platform grew. They do not vary with the gender of the interviewer on average, consistent with prior evidence on the effect of matching female job candidates with interviewers of different genders (Rivera and Owens, 2015). Nor do they vary substantially by problem difficulty (see Figure D1).

3 A Guiding Model of Discrimination

The gender gaps we see on the platform are potentially explained by unmeasured differences in performance, multiple types of discrimination, or a combination of phenomena. Guided by a model of discrimination in the spirit of Lundberg and Startz (1983), we investigate these possibilities systematically.[3]

3.1 Our Guiding Theoretical Model

The role of an interviewer is to evaluate the ability of job candidate i. The candidate's true ability, y_i, is unobservable. However, the interviewer sees a noisy but informative signal of it, θ_i. In the context of these coding interviews, ability likely encompasses aspects captured by the subjective ratings for problem solving, coding and communication, but potentially also other dimensions of ability. We focus initially on coding ability, as measured by the code quality rating.

We consider a simple benchmark in which interviewers believe the performance of candidates of gender $g \in \{m, f\}$ is normally distributed in the population, with mean μ_g and variance σ_g^2.

$$y_i \sim \mathcal{N}\left(\mu_g, \sigma_g^2\right) \tag{1}$$

The evaluator may believe (correctly or incorrectly) that the mean, μ_g, and standard deviation, σ_g^2, differ between male and female candidates in the population.

The signal that an interviewer observes is unbiased, but noisy. Specifically, $\theta_i = y_i + \varepsilon_i$, where ε_i is normally distributed with mean zero and variance σ_ε^2, and is inde-

[3] See also Aigner and Cain (1977) for a related model, and Fang and Moro (2011) for a more general review of the literature on statistical discrimination.

pendent of both y_i and g. The unconditional distribution of θ_i is as follows.

$$\theta_i \sim \mathcal{N}\left(y_i, \sigma_g^2 + \sigma_\varepsilon^2\right) \tag{2}$$

This signal summarizes all of the information available to an interviewer when she assigns a rating, including verbal interaction, observation of the candidate as she performs the assigned coding task, and any objective measures of code quality.

3.2 Statistical Inference by Evaluators

Rational inference implies that the interviewer combines her belief about the population with the information in the signal. The interviewer's posterior belief, b_i about the candidate's performance is a weighted average of the signal and the group mean:

$$b_i = E\left[y_i \mid \theta_i, g\right] = s_g \theta_i + \left(1 - s_g\right) \mu_g \tag{3}$$

where $s_g = \frac{\sigma_g^2}{\sigma_g^2 + \sigma_\varepsilon^2} \in (0, 1)$ is the weight placed on the signal.

The role of the interviewer's *ex ante* belief is greater if the signal is less informative.[4] In the extreme case in which it is completely uninformative, the interviewer's estimate of every candidate's performance is simply her belief about the mean given the candidate's gender, μ_g. By contrast, the interviewer's beliefs about the population distribution of ability would be irrelevant if the signal had no noise.

3.3 Code Quality Evaluations

After forming a belief about candidate i's performance, the evaluator reports a code quality rating. This is a function of the evaluator's belief about i's performance but may also feature other biases. Specifically, we let the rating be a function:

$$r_i = R(b_i, g \mid \boldsymbol{c})$$

where b_i is the evaluator's belief about code quality, g is the candidate's gender, and $\boldsymbol{c}$ is a vector of parameters governing the evaluation environment (e.g., whether it is blind, non-blind or face-to-face).

[4] Alternatively, the interviewer will place more weight on her *ex ante* belief if he or she is confident of that opinion in the sense that σ_g^2 is small.

3.4 Types of Discrimination

3.4.1 Statistical Discrimination

Statistical discrimination arises when an interviewer's prior belief differs by gender. The rating assigned to a man will then differ from that assigned to a woman given the same interview performance and any other information seen by the evaluator.

As a benchmark, suppose that interviewers believe the variance of ability, σ_g^2, to be the same for both genders.[5] This implies that $s_m = s_f = s$. Then the gender difference in beliefs about code quality for a given signal realization, θ_i, is:

$$\begin{aligned} \text{Gender Gap in Beliefs} \mid \theta_i &= E\left[y_i \mid \theta_i, m\right] - E\left[y_i \mid \theta_i, f\right] \\ &= (1-s)\left(\mu_m - \mu_f\right). \end{aligned} \tag{4}$$

Equation (4) shows that beliefs—and thus interview ratings—will reflect the interviewer's preconceptions about the performance levels of men and women. Fixing the candidate's interview performance, this implies a gender gap in evaluations. The gap is larger if the signal is noisier so that σ_ε^2 is larger, or the interviewer's beliefs are more strongly held so that σ_g^2 is smaller.

Since the gender gap in Equation (4) is conditional on interview performance, it constitutes discrimination. Nonetheless, it is referred to as *rational* if interviewers' prior beliefs are correct. In this case, a prerequisite for such a gap to exist is that there is a true difference in average coding ability between men and women on the platform. However, it is also possible that the difference between μ_m and μ_f reflects a mistaken belief (a "bias"). This is *non-rational* statistical discrimination.

3.4.2 Non-Statistical Discrimination

Beyond statistical discrimination, it is also possible for there to be systematic bias in ratings that is not explained by differences in beliefs. In this case, ratings differ by gender even given the same posterior belief (b_i) about code quality:

$$\text{Bias}(b_i \mid \boldsymbol{c}) = R(b_i, g = m \mid \boldsymbol{c}) - R(b_i, g = f \mid \boldsymbol{c}). \tag{5}$$

One reason for such a bias to exist is that evaluators may be taste-based discrimi-

[5] We consider the implications of relaxing this assumption in Appendix D.1. Note that differing prior variances—holding fixed the mean—leads to lower ratings for the high-variance group at the high end (for the same signal) but higher ratings at the low end.

nators, who penalize women relative to men as in Becker (1957). In this case, simply knowing the coder's gender should suffice to drive bias.

An alternative possibility is that they unconsciously (or "implicitly") discriminate. Bias may then only arise or will be exacerbated when the environment ($\boldsymbol{c}$) is changed such that gender is made salient through profile photographs or extended personal interaction. If this is the case, $\text{Bias}(b_i \mid \boldsymbol{c})$ will increase in these situations. This type of context-dependency would be predicted by the literature on implicit discrimination and stereotypes (Bertrand et al., 2005).

3.5 Testing For Different Types of Discrimination

Each potential explanation for gender gaps on the platform has distinct implications for the effects of interventions that change the way interviews and evaluations work. We were therefore able to design experiments to shed light on each of them.

Mistaken Beliefs. A gender gap driven by non-rational statistical discrimination must be underpinned by incorrect perceptions that women write worse code than men. Providing additional information about performance would therefore be expected to reduce the gender gap. In Section 4 (Experiment I), we study an intervention which randomizes the provision of such additional information.

Gender Gaps in Quality. Another possibility is that the gender gaps reflect differences in performance, with no discrimination. In Section 5 (Experiment II), we use code written on the platform, ask other coders to evaluate these scripts separately in a "blind" setting in which gender is not revealed, and compare ratings for men and women. Any gender differences in evaluations would be expected to be maintained in this setting if they were driven by true differences in quality.

Taste-Based Discrimination. A third alternative highlighted by the model is that the gender gaps are driven by traditional taste-based statistical discrimination. We designed Experiment II to allow us to test for this possibility by comparing ratings in the blind condition, and a "non-blind" condition, which reveals gender using the first name of the interviewee, but holds all else equal.

Rational Statistical Discrimination. The fourth possibility is rational statistical discrimination. Even if beliefs are centered on the truth, evaluators use a gap in average performance to inform judgments about specific individuals. Given the same performance, they would therefore assign a lower score to a woman than a man.

Comparison of the blind and non-blind conditions in Experiment II allows us to detect rational statistical as well as taste-based discrimination. However, the model motivates additional tests which allow us to distinguish statistical discrimination specifically. First, statistical discrimination should only arise if evaluators believe women to be worse coders on average. Second, it relies on the presence of a true gap in performance between men and women. We can test both of these implications using information about gender differences in performance as assessed by blinded evaluators, and by analyzing priors which we collect as part of Experiment II.

Context-Dependent Discrimination. The final possibility is that bias is amplified by personal interaction. While similar to traditional taste-based discrimination which is driven by evaluator preferences, this mechanism would imply that evaluations in Experiment II (without video interaction) would exhibit less bias than evaluations on the platform (with video interaction).

We formalize these theoretical predictions in Sections 4 and 5 in the context of our experimental designs, and present detailed results.

4 Experiment I: Providing Objective Information

Starting on July 8, 2017, the platform rolled out a powerful new diagnostic to verify the quality of code written on the platform. Because the roll-out was randomized, we can use it to test whether the gender gap in code quality ratings is driven by incorrect beliefs that women are less competent coders than men.

4.1 Intervention

The new tool provided automated ("unit") tests which assessed whether the code ran without errors, and produced the correct answers for test cases. Figure E3 shows an example unit test, with the prompt shown in Figure E2 (Panel A) along with a sample answer (Panel B). Users could choose to activate the tests by pressing a button (see Figure A1) and run them at any time. The results of the unit tests were then visible to both the evaluator and interviewee before subjective ratings were chosen.

4.2 Theoretical Predictions

Our guiding model in Section 3 has concrete predictions for the effect of this intervention: The gender gap in ratings should narrow if the gap is driven by non-rational statistical discrimination based on incorrect beliefs that disfavor women.

Letting μ_g^* be the true average ability of gender g candidates, the unconditional gap in beliefs is obtained by taking the expectation of Equation (4) over the signal distribution.

$$\text{Gender Gap} = s \underbrace{\left(\mu_m^* - \mu_f^*\right)}_{\textbf{True gap}} + (1-s) \underbrace{\left(\mu_m - \mu_f\right)}_{\textbf{Believed gap}} \tag{6}$$

The effect of providing more information is that s increases. Holding fixed an interviewer's prior beliefs about the distributions of coding ability among men and women, the interviewer then places more weight on the signal they observe, which reduces the role for preconceptions about gender differences in ability.[6]

Put differently, weight shifts from the initially believed gender gap to any true gap in performance. The effect on the gender gap depends on whether interviewers believe that the gap in coding ability is larger or smaller than it is in reality. If they believe the gap is larger than in reality, more information will shrink it. If they believe it is smaller, the gap widens. Thus, a narrowing of the gap in interview ratings would simultaneously provide evidence of belief-based bias, and a solution to that bias.

Theoretical Prediction. If evaluators believe incorrectly that women are less skilled coders than men, then introduction of the unit tests should reduce the gender gap.

4.3 Labor Market Outcomes: Verifying the Value of the Unit Tests

Higher scores on these unit tests are strongly associated with future labor market performance. To establish this, we linked the interview data to labor market data from Revelio Labs. This includes data from hundreds of millions of LinkedIn profiles, combined with other sources.[7] For close to the universe of computer science (CS) graduates in the US labor market, we observe job titles, employers, and salary estimates.[8]

We describe the matching process and our analysis in Appendix B, but summarize

[6] The distribution of coding quality need not be invariant, since less precise information undermines the incentive to exert effort (Craig, 2023). In our setting, however, the set of coding solutions is fixed.

[7] More detail regarding the Revelio data database is available www.reveliolabs.com.

[8] One concern is that there may be some sample selection. However, we have reason to believe that coverage is high for CS graduates in the United States (US). See Appendix B for further discussion.

it here. We match platform participants with a Bachelor's or Master's degree to individuals in the Revelio data who attained a CS-related degree from a US institution. Matching is based on exact first and last name, and degree type. The final sample consists of 5,126 matched CS graduates from 2016 to 2023. Around half of the individuals in this sample participated in the period in which we have unit test results. The average starting salary of this sample is $81,000, which compares to data from Glassdoor indicating an average salary for CS graduates of $85,000 in 2023.[9]

From here, we use Mincer-type wage regressions of log earnings on individuals' unit test scores, and their characteristics such as gender, race, the highest degree obtained, institution-of-highest-degree, year-of-graduation, and location. Results are presented in Table 2. Going from the 25th to the 75th percentile of unit test scores is associated with a wage increase of 4.5 percent. This compares to a 6 percent residual gender gap in the first salaries of computer science graduates in the Revelio data.[10] We also find suggestive evidence that the return to higher unit test scores is higher for men than for women, although the estimate for women is imprecise. Full details of all aspects of this analysis are available in Appendix B.

4.4 Treatment Assignment

Treatment assignment was randomized by the platform. The share of users treated at least once increased from July 2017 until all users were treated in October 27, 2017. During this roll-out period, we have data for all 6,401 sessions and 3,167 interviewees.

Figure A5 details how new users were assigned to treatment or control as they entered the platform during the phase-in period. When a new user i was paired to another user j, there were two possibilities. First, if both i and j were new users or had only been in the control condition in the past, the pair was randomized into treatment with a 7 percent probability. Once treated, a user always remained in treatment for future interactions. Second, any candidate matched with a partner who was already in the treatment condition was themselves treated (without randomization).

[9] Computer science graduates sort into various occupations, but according to the BLS they primarily sort into Software Developers. Data from GlassDoor shows that the average entry level salary for Software Developers is around $85,000.

[10] This may be conservative: Because salaries are imputed from job roles, they do not capture within-role variation in pay. We also note that the gender pay gap reflects both supply and demand factors, such as gender differences in preferences for job amenities, job search (Le Barbanchon et al., 2021; Cortes et al., 2021), earning expectations, negotiation (Reuben et al., 2017; Roussille, 2020), or discrimination.

This nonstandard randomization motivates robustness tests in Section 4.8. However, we note that baseline characteristics are quite balanced between the treated and the control groups, as shown in Table D5. The main concern is that users' experience with the platform might differ between treatment and control, as treatment is an absorbing state. Therefore, in additional specifications, we control for date fixed effects, and in some specifications control for the likelihood of being treated.

4.5 Differences in Activation

Either the interviewer or interviewee could choose whether to activate the device during the interview, and not all did. We account for this using two-stage least squares (2SLS). We start with an Intention-to-Treat (ITT) model:

$$Y_{it} = \beta T_{it} + \theta_t + \epsilon_{it} \tag{7}$$

where Y_{it} is the score of individual i on date t, and θ_t are date fixed effects. $T_{it} = 1$ if the feature was enabled for a pair of users, and 0 otherwise.[11] The ITT is β from Equation (7). Standard errors are clustered at the date level.

Next, we estimate the treatment effect on the treated (TOT) by using treatment assignment as an instrument for actual treatment. Specifically, we estimate the following model using 2SLS:

$$Y_{it} = \delta D_{it} + \lambda_t + \eta_{it} \tag{8}$$

$$D_{it} = \pi T_{it} + \zeta_t + \nu_{it} \tag{9}$$

where Y_{it} is the outcome of user i at time t; D_{it} is a dummy for whether the user activated the tests; T_{it} is an indicator of whether the pair was assigned to treatment; and λ_t and ζ_t are time fixed effects. Standard errors are clustered at the date level.

4.6 Result: No Reduction In The Gender Gap

We begin our analysis studying the activation decision and the impact of the new information on gender gaps in subjective ratings. We then look at whether differences in objective performance are related to differences in ratings.

Estimates from Equation (7) and (8) are shown in Table 1. Panel A shows results for all users, then Panels B and C show results for men and women separately. For each

[11] Results are robust to the introduction of problem fixed effects.

outcome, the first column of the top sub-panel present ITT estimates of Equation (7). The second column presents 2SLS estimates. The first stages are summarized in the lower sub-panels. Appendix D.2 provides information about the compliers.

First Stage: Activation. 71 percent of users enabled the objective code quality tests, when available. This strong first stage suggests that the code quality ratings were observed and valued by participants. We observe a slightly weaker first stage for women (0.678, S.D=0.016) than for men (0.721, S.D=0.016). This is a small difference, but could reflect relative under-confidence of women (Mobius et al., 2022) or attention discrimination (Bartoš et al., 2016). We cannot distinguish these two hypotheses because we cannot observe whether the evaluator or interviewee activated the tests.

Treatment Effects on Subjective Ratings. Both men and women in the treated group receive higher ratings than their peers in the untreated group for all the ratings. The largest effects are on dimensions where the unit tests likely shed the most direct light, including the code quality and problem solving ratings. We also see improvements in communication ratings, which may reflect improvements in how participants talk about their code when more information about quality is available. Likeability ratings increase slightly. On net, we see an improvement in assessments of hireability.

Despite the increase in overall ratings, treatment did not disproportionately increase ratings for women. Instead, the increases in ratings are generally slightly larger for men, although our estimates are noisy. This is especially the case for coding and likability, where the effects are only marginally significant for women. As a result, gender gaps in subjective ratings persist following the introduction of the unit tests.

Why Would Ratings Increase? Our results indicate that the gender gaps persisted with more information, although ratings increased across the board. We evaluate alternative explanations for increase in ratings in Appendix D.1. Our leading explanation is that evaluators were unduly pessimistic for all coders, and potentially more about men than women. As we discuss in Section 5, we find some evidence consistent with this pattern when we collect information about prior beliefs in Experiment II.

4.7 Gender Gaps Controlling for Objective Code Quality

In the period following the introduction of the unit tests, we can also assess whether the gender gaps in subjective ratings are explained by gender differences in performance as measured by those tests. Our results suggest not. We first show that women are slightly underrepresented at the top of the performance distribution illustrated in Figure 2. Because performance on the unit tests is bimodal, we split the sample in two groups: users who passed all unit tests, and those who did not. For each of the two levels of performance, Figure 3 shows the average rating for each gender. Panel A plots average code quality ratings by objective performance, and Panel B shows ratings for problem solving. Large gender gaps remain, even conditional on objective performance. Although the gender gap in subjective ratings is halved for users at the top of the objective performance distribution, women receive lower subjective coding and problem solving ratings than men who perform equally well by this measure.

4.8 Robustness Checks

Alternative Samples and Empirical Designs. Table D7 provides robustness checks to probe the validity of our results. Panel A shows a baseline in which we estimate the ITT model interacted by gender.

In Panels B and C, we add month-of-interview, and then date-of-interview fixed effects. These adjust for changes in the share of users treated over time, and changes in user composition. The interaction of treatment with gender remains imprecisely estimated, still suggesting a slight widening of the gender gap. We control for individual characteristics in Panel D and find the same results. Including interviewee-fixed-effects in Panel H attenuates the treatment coefficients, with the interaction coefficient γ statistically insignificant. To ensure our results are not sensitive to the sample period, we expand our sample to include the pre-treatment period: The coefficients shrink slightly but the results are similar.

Endogenous Matching Between Users. The way in which treatment was randomizes means that treatment assignment may be contaminated by the matching process, in which case a naive comparison between treated and control users could provide a biased estimate. To address this threat, we control for a propensity score measuring the likelihood of being assigned to treatment.[12] The results are shown in Panel G of Table D7. Controlling for the propensity score does not affect our results.

[12] To estimate the propensity score, we use month-of-interview fixed effects and (for both the interviewer and interviewee) a dummy variable for each degree level, a dummy variable for each field of study, the number of years of experience, the self-declared level of preparedness, and gender.

Evaluator Assignment. We next ask whether women are more likely to be matched with harsh evaluators, defined as interviewers whose average coding ratings (excluding the focal session's rating) is below the median. Columns (3) and (4) of Table D4, show that female users are not more likely to be matched with a harsh evaluator.

User Composition. Conditional on an individual's covariates and their partner's, treatment assignment should be nearly as good as random, especially because the matching algorithm used by the platform uses the same characteristics. Nonetheless, we explore changes in user composition over time and in response to treatment. The results are reassuring. Our main specifications nonetheless control for date-of-interview fixed effects to minimize any concern that such changes could affect one gender more than the other.

Figure D5 shows that the gender composition of users did not change with the introduction of the unit tests. However, there could still be changes in which women select onto the platform. Figure D7 therefore confirms that there are no changes in the characteristics of first-time female users around time the tests were introduced in terms of work experience, educational background or field of study. Next, Figure D6 shows that other characteristics are also stable: We find no evidence of changes in the share who are US citizens, have a computer science degree, a graduate degree, or no working experience. Finally, we look at the share of high-performing users among first-time users, defined as those who passed all unit tests taken during their first interview. Figure D8 plots the shares of high-performing first-time female and male users and shows that they follow a parallel increase over time. Thus, the quality of first-time users increases over time, but not differentially by gender.

Gender Differences in Activation. Given the small gender differences in activation of the unit tests, we explore the possibility that there is differential selection by gender into activation. A potential reason for this to occur would be if one group were less likely to take the tests because they have lower self-confidence. We assess this in Figure D3, which shows the share of unit tests passed versus the number of tests taken, separately for male and female users. It shows that use of the tests varies similarly with objective performance for men and women.

5 Experiment II: Blind and Non-Blind Code Evaluation

The results so far established that there are gender gaps in evaluations even after controlling for unit test scores that measure code quality. These gaps are not reduced when evaluators are provided with the unit tests before choosing their rating.

Our theoretical model in Section 3 highlights three remaining explanations for the gender gaps. One is that women write code that is genuinely different in a way which is viewed by evaluators as lower quality on a dimension not captured by the unit tests. For example, there may be differences in efficiency, elegance, or portability. A second is that the gaps are driven by stable biases of a different kind, and not based on incorrect beliefs. For example, evaluators may engage in taste-based discrimination, in which case simply knowing the coder′s gender would suffice to drive bias. Finally, evaluators may be discriminating in a context-dependent way, with bias arising when gender is made more salient due to face-to-face interaction.

To distinguish these mechanisms we used coding solutions written by platform users in another randomized experiment. This experiment used a within-subject design, with new evaluators asked to assess code written by men and women in a "blind" setting where gender was masked, and a "non-blind" setting in which gender was revealed via the coder′s name. A novel feature of our experiment is that the same code blocks are evaluated in all three contexts: in-person on the platform, in our "blind" experimental arm, and in the "non-blind" arm. In contrast to other studies of blind evaluations (Goldin and Rouse, 2000), this lets us rule out differences in performance across conditions and contexts due to phenomena such as stereotype threat.[13]

We start by asking whether the gender gap on the platform can be explained by differences in unobservable code quality. To assess this mechanism we test for the presence of a gender gap in the blind condition. Next, we ask whether taste-based discrimination can explain the gender gap. To evaluate this possibility we compare the gender gap in ratings for the same code in the non-blind and blind conditions. Finally, to evaluate the importance of face-to-face interactions we compare ratings on the platform to non-blind experimental evaluations of the same code, which have no in-person component. We justify these comparisons theoretically below.

[13] Other studies have documented grading biases favoring women in male-dominated fields, by comparing results between written and oral examinations (Breda and Ly, 2015; Breda and Hillion, 2016), or in-class exams to blind evaluations by external graders (Terrier, 2020; Lavy and Sand, 2018).

Evaluators for this second experiment were mainly Bachelor's and Master's level computer science students with familiarity in the relevant programming languages. Descriptive statistics for our sample of experimental evaluators are available in Table E3. While the participants were not drawn from the set of users on the platform, they are similar in characteristics. We further discuss comparability between the experimental and platform contexts in Section 5.3, and show results when we rebalance the sample to match more exactly. A detailed description of the experiment's design is available in Appendix E. The RCT was pre-registered on December 14, 2022.[14]

5.1 Theoretical Predictions

In the blind condition when gender is masked, the evaluator could not see the gender of the coder. To form a belief about an individual's performance, the relevant prior is therefore the pooled distribution of ability among both men and women.

Letting λ_g be the fraction of participants of gender $g \in \{m, f\}$, and assuming that performance of each gender is normally distributed, the pooled belief is:

$$y_i \sim \mathcal{N}\left(\mu, \sigma^2\right) \tag{10}$$

where $\mu = \lambda_m \mu_m + \lambda_f \mu_f$ and $\sigma^2 = \lambda_m \sigma_m^2 + \lambda_f \sigma_m^2 + \left(\lambda_m \mu_m^2 + \lambda_f \mu_m^2 - \mu^2\right)$.

Conditional on the signal, θ, the posterior belief of a worker's performance is:

$$E\left[y_i \mid \theta_i, g\right] = \widetilde{s}\theta_i + (1 - \widetilde{s})\,\mu \tag{11}$$

where $\widetilde{s} = \frac{\sigma^2}{\sigma^2 + \sigma_\varepsilon^2} \in (0, 1)$ is the weight placed on the signal. Therefore the unconditional gender gap in beliefs is:

$$\text{Gender Gap} = \widetilde{s}\left(\mu_m - \mu_f\right). \tag{12}$$

This highlights that there cannot be a gender gap when evaluation is blind unless there are true differences in productivity between the groups. By comparison, the gender gap in the non-blind case in Equation (6) is a weighted average of the true gender gap in performance and the prior belief about that gap. These results suggest that we can jointly test for taste-based and rational statistical discrimination by comparing the gender gap observed for blind and non-blind evaluations of the same code.

As it turns out, we find no gender gap in blind ratings, which suggests that there is

[14] ID: AEARCTR-0009816. The pre-analysis plan is on the AEA RCT registry (updated: Feb 17, 2023).

no gap in skill between men and women. In turn, without a gap in skill, the model precludes rational statistical discrimination. The gender gap for non-blind experimental evaluations therefore reveals the extent of taste-based bias. We denote this by:

$$\text{Bias}(b_i \mid \boldsymbol{c} = \boldsymbol{c}_{NF}) = R(b_i \mid g_i = m, \boldsymbol{c} = \boldsymbol{c}_{NF}) - R(b_i \mid g_i = f, \boldsymbol{c} = \boldsymbol{c}_{NF}) \tag{13}$$

where $\boldsymbol{c}_{NF}$ indicates that there is no face-to-face interaction in the experiment.

Finally, we can compare bias in non-blind evaluations in the experiment to bias on the platform itself. We denote this by $\text{Bias}(b_i \mid \boldsymbol{c} = \boldsymbol{c}_F)$, where $\boldsymbol{c}_F$ indicates the presence of face-to-face interaction. Because this is the main way in which the two settings differ, comparing bias across the two contexts tells us about how bias is changed by face-to-face interaction. Specifically, one might expect that personal interaction makes gender more salient, and introduces "implicit" bias.

Theoretical Predictions. (1) If the gender gap is driven by differences between the code written by men and women, the gap will remain unchanged for a given code block when evaluation is made blind. (2) If the gender gap is driven by fixed taste-based bias, revealing gender should produce a wider gender gap. (3) If the gender gap is driven by context-dependent bias, in-person interactions as on the platform should change the gender gap relative to the non-blind experimental condition.

5.2 Empirical Design

5.2.1 Selecting Code Blocks from the Platform

To select code blocks for the experiment, we restrict to what we refer to as the experimental sample. We drop observations without unit test scores, keep only the most common programming languages (C++, Java, and Python), restrict to code with length no more than one standard deviation from the mean, and only consider the first attempt in cases where a given participant attempts the same problem twice. Finally, we exclude names that are uncommon or where gender is otherwise ambiguous.

Descriptive statistics from each step of the sample construction are presented in Tables E1 and E2. From this experimental sample, we select code blocks for the experiment. We do this in a stratified manner to maximize power for our statistical analyses. For each coding problem and language pair, we stratify by gender, race, and coding performance (whether the code passed all unit tests or not). Within each of these cells, we randomly picked one code block for the experiment. This yields a final sample of 456 code blocks. Table E2 presents summary statistics for this sample.

In the experiment, each evaluator sees code written by male and female users in each treatment arm. An example a code block and prompt is shown in Figure E2. Each evaluator is assigned four coding blocks in a random order. They evaluate these on the same Likert scales as on the platform, but without face-to-face interaction.

5.2.2 Treatment

Of the four blocks presented to an evaluator, two were "blind", and two "non-blind", with the order randomized. Within each of these arms, one code block was written by a man, and one by a woman. The order was again randomized. Table E4 confirms that the characteristics of evaluators are balanced across treatment orderings.

In the non-blind condition, gender was revealed via the given name of the coder. In addition, a box was shown an avatar that revealed gender without indicating any other aspect of a person's identity. In the blind condition, gender was hidden: Only the initial of the given name was seen, and no avatar was shown. An example of each treatment condition is presented in Figures E4-E7.

5.2.3 Key Outcomes And Additional Measures

Main Outcome. Evaluators judged the quality of code using the same Likert scales as on the platform, which range from 1 to 4. This is our main dependent variable.

Secondary Outcomes. We also asked the experimental evaluator for a prediction of: (1) the share of unit tests the code block passed; (2) whether a human evaluator judged that the coder passed or failed the interview; and (3) the percent chance that the candidate was later invited for a real interview for a role involving coding. This allows us to draw a more direct link between our findings and hiring outcomes.

Prior Beliefs. To measure participants' priors, we exposed them to three different vignettes before they performed their evaluation tasks. We asked them to predict the performance of three different hypothetical coders. We cross-randomized the first name (alternating gender) and the skill level for each vignette (see Appendix E).

Quality Measures. We measure how much time respondents spend on each question to measure fatigue and inattention, and how this varies over time. Our various measures of quality are presented in Table E7. We define our "high quality" sample as those passing the first attention check, and for whom the survey duration was between the first and last decile (more than 7 minutes, less than 4 hours), but we also check that our results are consistent with other measures of quality.

5.2.4 Incentives

Participants were incentivized in several ways. First, they were paid a participation fee of \$10, plus a piece rate of \$10 per script they evaluated. Second, they received bonus payments of \$2 for each accurate predictions they make for the objective code quality and hireability measures per code block. Third, the 10 best evaluators could earn a cash prize of \$500. Finally, we provided a non-financial but potentially powerful incentive by selecting a set of evaluators to participate in the Creative Destruction Lab 2023 Super Session. This brought real networking opportunities with world-class entrepreneurs, investors and scientists with high-potential startup founders.

5.2.5 Econometric Specifications

Analysis. Our primary aim is to test whether revealing gender changes the gender gap in ratings. To do so, we use the following specification.

$$\begin{aligned} Y_{ij} &= \beta_1 \times \text{Female\_Coder}_j + \beta_2 \times \text{NB}_{ij} + \beta_3 \times \text{NB}_{ij} \times \text{Female\_Coder}_j \qquad (14) \\ &+ \beta_4 \times \text{High\_Performer}_j + \beta_2 \times \text{Treatment\_Order}_i \\ &+ \sum_{k=1}^{4} \gamma_{jk} \mathbb{1}(\text{Script\_Order}_j = k) + \pi_{p(j)} + \delta_i + \epsilon_{ij} \end{aligned}$$

Here, we indicate treatment by defining $\text{NB}_j = 0$ for blind evaluation j, and $\text{NB}_j = 1$ for non-blind evaluation. Treatment_Order_i is an indicator for the randomly assigned treatment order ("non-blind then blind" condition versus "blind then non-blind"); and $\text{Script\_Order}_j = k$ is used to construct indicators that a given code block was the kth block the coder evaluated, to account for fatigue and learning. High_Performer_j indicates whether the code passed all unit tests or not. We include problem fixed effects, $\pi_{p(j)}$. In some specifications, we include evaluator fixed effects (δ_i) and additional controls. Standard errors are clustered at the evaluator level.

The coefficients of interest are: β_1, which measures the quality difference between male and female code in the blind condition; and β_3, which measures the differential effect of revealing the gender of the coder, depending on what that gender is.

As pre-specified, we also look at heterogeneity of effects. We do this with variants of Model (14) where treatment effects on gender bias are interacted with the gender of the evaluator, the difficulty and characteristics of the code, the coder's performance, and bias in the evaluator's beliefs as measured by their prior.

5.3 Results

No Gender Differences In Code Quality. Figure 5 presents our main results and Table 3 the corresponding estimates. The estimate of β_1 shows that in the blind condition, code blocks written by female coders do not receive lower ratings, predicted scores or interview chances. If anything, the coefficients are positive, although we cannot rule out zero or small negative coefficients. This rules out meaningful gender differences in coding styles that could drive gender disparities in the face-to-face interviews, but are not accounted for by the unit tests.

No Bias When Gender Is Revealed. Turning to the comparisons of treatments, our estimate of the effect of making evaluation non-blind (β_2 in Equation 14) is negative but the confidence interval includes zero, while the interaction with Female_Coder$_i$ (β_3) is positive though imprecisely estimated. In this sense, do not find evidence of systematic gender bias that arises when gender is revealed by the first name, as one would expect if the gender gap were driven by traditional taste-based discrimination.

Prior Beliefs. Experiment II allows us to explore participants' prior beliefs about the coding ability of men and women. Figure E1 shows the distributions of respondents' prior beliefs about performance on the unit tests. They split by gender and by the skill level reported in the vignette, ranging from a B.Sc to a Master's in computer science with various years of work experience. On average, prior beliefs tend to be similar for men and women, as reflected by the vertical continuous lines which show the mean reported prior. For comparison, the vertical dashed lines show the share of tests actually passed by coders of each gender.

There are two key lessons from Figure E1. First, participants tend to be too pessimistic across the board about the coders in the vignettes, despite having been told that 82 percent of all users pass the unit tests. This could help explain why the introduction of the unit tests in Experiment I increased ratings for both men and women. Second, priors for men and women are quite similar on average. In reality, men perform slightly better on these tests, although not nearly enough to explain the gap in ratings between male and female coders. Combined, these results are again consistent with the results of Experiment I, providing a reason why introducing the unit tests did not succeed in reducing the gender gap in evaluations.

Comparability of Contexts. Our experiment was constructed to closely mirror the platform experience, with the main difference being the removal of face-to-face interaction. For example, we chose both the participants and the rating scales to match the platform. Appendix Figure E9 verifies that there is a robust correlation for male users between ratings in Experiment I and platform ratings for the same code, despite the fact that the relationship is likely attenuated by noise. This is true for both blind and non-blind evaluations, and supports the idea that evaluators are answering the coding evaluation question in a similar way in both contexts for male coders. The correlation is weaker for female-written code. This may be explained by a reduction in bias, which we later argue arises when in-person interaction is removed in the experiment.

To further confirm that our results are not driven by differences between the samples, we explore how our results change if we re-weight our regressions to more exactly match the composition of users on the in terms of educational degree and gender (Table E8). Our results are qualitatively unchanged across these conditions, with nearly identical levels of bias in all specifications.

Alternative Samples. To test robustness, we explore whether our results hold in alternative samples. First, we restrict to our "high quality" sample (see Section 5.2.3). Table E5 provides balance tests within this sample. Results are then presented in Table E6, and point to similar effects as in the whole sample. We have also explored other restrictions using the measures of quality presented in Table E7, with qualitatively similar results (available on request). This includes restricting to participants who passed the first attention check, excluding participants whose reported ability with the language question is "basic", keeping only respondents who completed all evaluations assigned to them, and restricting to graduate student evaluators.

6 What Drives the Gender Gap in Code Ratings?

We began by documenting that there are gender gaps in evaluations of code quality on this platform, which remain even when we control for rich information about coders and their code. Our model motivated tests of potential mechanisms underlying this gap, and provides a useful lens through which to summarize and interpret our results, while also suggesting some final tests to support our conclusions.

6.1 Evidence on Potential Sources of Discrimination

Gender Differences in Code Quality. The results from the blind condition in Experiment II suggest that women do not write code that is of lower quality than men: For the set of coding solutions we ask experimental participants to evaluate, there was no clear gender gap in blind evaluations. This is despite a gender gap being observed for the same code on the platform where gender is observed and subjects interact.

Rational Statistical Discrimination. We see no gender gap in blind-evaluated code quality, and no gender gap in prior beliefs. This makes it hard to rationalize the gap in ratings we see with rational statistical discrimination. In the notation of the model: if $\mu_m = \mu_f$, then the gender gap in beliefs should be zero. Without some form of non-statistical bias, this would imply that there would be no gender gap in evaluations.

Non-Rational Statistical Discrimination. Can the gaps be explained by statistical discrimination with incorrect evaluator beliefs? Experiment I suggests otherwise. The experiment provided valuable additional information to evaluators, increasing the precision of the signal they saw of the coder's skill. However, we find no evidence that the gender gap falls, which would have been expected if the gender gap were driven by incorrect beliefs about the average skill levels of men and women.

Taste-Based Bias. Because there is no evidence of statistical discrimination, we can test for traditional tasted-based discrimination by comparing blind to non-blind evaluations of the same code. If blinding had eliminated or reduced gender gaps, this would have suggested taste-based discrimination, which arises when evaluators are made aware of the gender of the coder. Instead, blinding made little difference: Without gender being visible, there is no gender gap on average in evaluations of the code, and this does not change when gender is revealed via the coder's first name.

Gender Differences in Coding Time. Despite the similarities between the platform and experimental settings, there is one key difference: Participants interact via video on the platform. Bias only arises when such personal interaction is allowed. This raises the possibility that there are factors that affect the rating while not manifesting in the code itself. A clear example is that solving the coding problem might take longer for women than men. This could be the case if women are slower at solving a problem of given difficulty, or if they receive more help from their interview partner. However, we observe time use on the platform, and can verify that there are no significant gender differences in coding time (see Figure 6). As a result, controlling for interviewees' coding duration does not reduce the gender gap in ratings (Table 4). While there is a rating penalty for slow coders controlling for objective performance measures and problem fixed effects, this penalty does not vary significantly by gender.

Gender Differences in Communication Style. An alternative possibility is that men and women talk about their code differently. If women are less effective at communicating, this could introduce a gender gap that is not there when code is evaluated alone. We can test some versions of this hypothesis, because we observe ratings for communication and likeability. Figure 4 plots the average subjective ratings for communication (Panel A) and likeability (Panel B) by objective performance (share of unit tests passed), separately by gender. While both high and low performing women receive systematically lower subjective coding and problem solving ratings than men who perform equally well (Figure 3), the communication and likability ratings of men and women are similar across the objective performance distribution. This suggests that gender differences in communication styles are unlikely to explain the persistent gender gaps in coding subjective ratings.

6.2 Implicit Bias With Face-to-Face Interaction

We argue that these gender gaps reflect bias that is triggered when gender differences in mannerisms and behavior become significantly more noticeable during face-to-face interaction, aligning with the concept of "implicit" bias (Bertrand et al., 2005; Carlana, 2019; Hangartner et al., 2021; Barron et al., 2022; Cunningham and de Quidt, 2022).

In addition to the comparison of the platform and experimental settings in Section 5, we now supplement this with more direct evidence of implicit bias. By harnessing the linkage between the platform data and individual-level LinkedIn information (see Section 4.3), we can see evaluators' higher education institutions. This allows us to compare ratings assigned by evaluators who attended an institution in geographic areas with high Implicit Association Test (IAT) scores—indicating more prejudice towards women in science (measured from Harvard's Project Implicit)—to those educated in areas with lower IAT scores. Figure 7 shows that the gender gap in coding ratings is significantly larger for interviewers educated in high-IAT regions.[15]

A further reason to think that implicit bias is at play is that the gender gap in coding ratings is larger when the duration of face-to-face interaction is longer. Such longer duration plays two roles. One, provides more time for gender differences in mannerisms to manifest. Two, increases evaluators' fatigue levels when making judgement decisions. With gender being more salient, and evaluators being more fatigued, reliance on implicit biased associations becomes more likely. Specifically, Table 4 shows that the gender gap increases when their partner's coding duration is longer: A fifteen-minute increase in the length of the session is associated with a gender gap that is 4 percent of a standard deviation wider. The reason we focus on the duration for an individual's *partner* is that this lengthens the interaction without being confounded with the individual's own coding proficiency.

These results are not driven by systematic differences in evaluators or the evaluator's own experience. We verify this by adding controls for the evaluator's own objective performance, and evaluator fixed effects. The results are similar. We also test whether the gender gap reduces as interviewers on the platform complete more interviews, becoming increasingly experienced. Table C2 shows that this is not the case. In summary, bias in the assessment of quantitative skills manifests specifically when a long interaction with a given person makes gender salient at the time. This is a novel contribution: While IAT scores are known to predict bias in settings with sustained interaction as well as in snap decisions as in the IAT itself (Carlana, 2019), our results suggest that longer interactions exacerbate this bias. This may be explained by more reliance on stereotypes when evaluators are under higher cognitive load, combined with gender being salient because interaction is face-to-face.

This type of context-dependent bias explains all the patterns we see. Specifically, the gender gaps in ratings on the platform are persistent and not explained by performance on the unit tests. Yet revealing the gender of the coder without interpersonal

[15] We define a high IAT area as a metropolitan statistical area with an average IAT Gender-Science score above the US median of 0.31. Estimates by subgroup are presented in Table C1. The distribution of IAT scores across geographic areas in our sample is provided in Figure C1.

communication is not enough to replicate these gaps among our very similar experimental evaluators. These results also align with work by Petrie and Greenberg (2023), who demonstrate that video interaction changes bargaining behavior more than text-based chat in a setting where communication introduces gender gaps in bargaining outcomes. They are also in line with recent work suggesting that images have the potential to serve as a more impactful medium for the perpetuation of gender bias than text alone (Guilbeault et al., 2024). Finally, the apparent increased reliance on stereotypical heuristics when evaluators are fatigued from long sessions is consistent with recent results by Doyle et al. (2024), which show that teachers are more likely to be biased towards underrepresented groups when they are multitasking.

7 Conclusion

We present two field experiments studying coding evaluation in the technology industry, which is a sector where women are systematically underrepresented. Across these experiments, we evaluate three treatments which systematically vary the information about a candidate's performance presented to evaluators.

Our results show that gender bias in performance evaluation is context-specific. Face-to-face interaction appears to be a precursor for gender bias. Without such interaction, we rule out traditional taste-based and statistical discrimination in this context. We also rule out differences in performance. Our results are most consistent with the literature on implicit discrimination and stereotypes. Put differently, in line with the sociology literature, biases are more likely to emerge when individuals are "doing gender" (West and Zimmerman, 1987) during personal interaction, rather than when gender is merely revealed by a person's name. This conclusion is further supported by the fact that longer interactions are associated with larger gender gaps, and the association between gender gaps and IAT scores where the evaluator was educated.

It remains an important question for future research precisely which settings and modes of interaction lead to such bias. Some have argued that inter-group contact can reduced biases (Pettigrew and Tropp, 2006), yet implicit bias persists even in settings with extensive contact (Carlana, 2019; Alesina et al., 2023). We go further, and find that sustained interaction with a given individual appears to amplify bias. However, more work is needed to understand the effects of the mode of interaction. For example, watching a recording of an interview may suffice to reveal gender differences in mannerisms. Alternatively, synchronous interaction may be required.

Our analysis suggests concrete ways to mitigate bias in performance evaluation. The gender gap in our setting is eliminated when personal interaction is removed. Decoupling coding evaluations from face-to-face interviews may therefore provide a way to reduce biases in the evaluation of cognitive skills, because the technical evaluations will not themselves involve face-to-face interaction. We caution that it could be more problematic to remove face-to-face interaction entirely: This could harm female candidates who have relatively strong social skills, which are becoming increasingly valued in the labor market (Deming, 2017).

References

Abadie, Alberto, "Semiparametric instrumental variable estimation of treatment response models," *Journal of Econometrics*, 2003, *113* (2), pp. 231–263.

Abramitzky, Ran and Leah Boustan, "Immigration in American Economic History," *Journal of Economic Literature*, 2017, *55* (4), pp. 1311–1345.

_, Leah Platt Boustan, and Katherine Eriksson, "Europe's Tired, Poor, Huddled masses: Self-Selection and Economic Outcomes in the Age of Mass Migration," *American Economic Review*, 2012, *102* (5), pp. 1832–1856.

_, _, and _, "A Nation of Immigrants: Assimilation and Economic Outcomes in the Age of Mass Migration," *Journal of Political Economy*, 2014, *122* (3), 467–506.

Aigner, Dennis J. and Glen G. Cain, "Statistical Theories of Discrimination in Labor Markets," *Industrial and Labor Relations Review*, 1977, *30* (2), 175–187.

Alesina, Alberto, Michela Carlana, Eliana La Ferrara, and Paolo Pinotti, "Revealing Stereotypes: Evidence from Immigrants in Schools," *American Economic Review*, 2023, *forthcoming*.

Allport, Gordon Willard, Kenneth Clark, and Thomas Pettigrew, *The Nature of Prejudice*, Addison-wesley publishing company Cambridge, MA, 1954.

Ashcraft, Catherine, Brad McLain, and Elizabeth Eger, *Women in tech: The facts*, National Center for Women & Technology (NCWIT), 2016.

Avery, Mallory, Andreas Leibbrandt, and Joseph Vecci, "Does Artificial Intelligence Help or Hurt Gender Diversity? Evidence from Two Field Experiments on Recruitment in Tech," *Evidence from Two Field Experiments on Recruitment in Tech (February 14, 2023)*, 2023.

Barbanchon, Thomas Le, Roland Rathelot, and Alexandra Roulet, "Gender Differences in Job Search: Trading off Commute against Wage," *The Quarterly Journal of Economics*, 2021, *136* (1), 381–426.

Barron, Kai, Ruth Ditlmann, Stefan Gehrig, and Sebastian Schweighofer-Kodritsch, "Explicit and Implicit Belief-Based Gender Discrimination: A Hiring Experiment," Technical Report, CESifo Working Paper 2022.

Bartoš, Vojtěch, Michal Bauer, Julie Chytilová, and Filip Matějka, "Attention Discrimination: Theory and Field Experiments with Monitoring Information Acquisition," *American Economic Review*, 2016, *106* (6), 1437–75.

Becker, Gary S, "The Economics of Discrimination (chicago: University of chicago)," 1957.

Behroozi, Mahnaz, Shivani Shirolkar, Titus Barik, and Chris Parnin, "Debugging hiring: What went right and what went wrong in the technical interview process," in "Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Society" 2020, pp. 71–80.

Bellemare, Charles, Marion Goussé, Guy Lacroix, and Steeve Marchand, "Physical Disability and Labor Market Discrimination: Evidence from a Video Résumé Field Experiment," *American Economic Journal: Applied Economics*, 2023, *15* (4), 452–476.

Bertrand, Marianne and Esther Duflo, "Field experiments on discrimination," in "Handbook of Economic Field Experiments," Vol. 1, Elsevier, 2017, pp. 309–393.

_ and Sendhil Mullainathan, "Are Emily and Greg more employable than Lakisha and Jamal? A field experiment on labor market discrimination," *American Economic Review*, 2004, *94* (4), 991–1013.

_ , Claudia Goldin, and Lawrence F Katz, "Dynamics of the Gender Gap for Young Professionals in the Financial and Corporate Sectors," *American Economic Journal: Applied Economics*, 2010, *2* (3), 228–55.

_ , Dolly Chugh, and Sendhil Mullainathan, "Implicit Discrimination," *The American Economic Review*, 2005, *95* (2), 94–98.

Bohren, J Aislinn, Kareem Haggag, Alex Imas, and Devin G Pope, "Inaccurate statistical discrimination: An identification problem," *Review of Economics and Statistics*, 2023, pp. 1–45.

Bordalo, Pedro, Katherine Coffman, Nicola Gennaioli, and Andrei Shleifer, "Stereotypes," *Quarterly Journal of Economics*, 2016, *131* (4), pp. 1753–1794.

Boudreau, Kevin and Nilam Kaushik, "The Gender Gap in Tech & Competitive Work Environments? Field Experimental Evidence from an Internet-of-Things Product Development Platform," Technical Report, National Bureau of Economic Research 2020.

Breda, Thomas and Mélina Hillion, "Teaching accreditation exams reveal grading biases favor women in male-dominated disciplines in France," *Science*, 2016, *353* (6298), 474–478.

_ and Son Thierry Ly, "Professors in core science fields are not always biased against women: Evidence from France," *American Economic Journal: Applied Economics*, 2015, *7* (4), 53–75.

Carlana, Michela, "Implicit Stereotypes: Evidence from Teachers' Gender Bias*," *Quarterly Journal of Economics*, 03 2019, *134* (3), 1163–1224.

Coate, Stephen and Glenn Loury, "Antidiscrimination Enforcement and the Problem of Patronization," *American Economic Review*, 1993, *83* (2), pp. 92–98.

Cortes, Patricia, Jessica Pan, Ernesto Reuben, Laura Pilossoph, and Basit Zafar, "Gender Differences in Job Search and the Earnings Gap: Evidence from the Field and Lab," Technical Report, National Bureau of Economic Research 2021.

Craig, Ashley C., "Optimal Taxation with Spillovers from Employer Learning," *American Economic Journal: Economic Policy*, 2023, *14* (2), pp. 82–125.

_ and Roland G. Fryer, "Complementary Bias: A Model of Two-Sided Statistical Discrimination," 2019.

Cunningham, Tom and Jonathan de Quidt, "Implicit Preferences," Technical Report, CEPR Discussion Paper 2022.

Deming, David J, "The Growing Importance of Social Skills in the Labor Market," *The Quarterly Journal of Economics*, 2017, *132* (4), 1593–1640.

Doyle, Lewis, Peter R Harris, and Matthew J Easterbrook, "Quality and quantity: How contexts influence the emergence of teacher bias," *Social Psychology of Education*, 2024, pp. 1–28.

Dupas, Pascaline, Alicia Sasser Modestino, Muriel Niederle, Justin Wolfers et al., "Gender and the dynamics of economics seminars," Technical Report, National Bureau of Economic Research 2021.

Fang, Hanming and Andrea Moro, "Theories of Statistical Discrimination and Affirmative Action: A Survey," in Jess Benhabib, Matthew O. Jackson, and Alberto Bisin, eds., *Handbook of Social Economics*, Elsevier, 2011, chapter 5, pp. 133–200.

Farber, Henry S, Dan Silverman, and Till Von Wachter, "Determinants of callbacks to job applications: An audit study," *American Economic Review*, 2016, *106* (5), 314–18.

Feld, Jan, Edwin Ip, Andreas Leibbrandt, and Joseph Vecci, "Identifying and Overcoming Gender Barriers in Tech: A Field Experiment on Inaccurate Statistical Discrimination," Technical Report, CESifo Working Paper 2022.

Goldin, Claudia, "A grand gender convergence: Its last chapter," *American Economic Review*, 2014, *104* (4), 1091–1119.

_ and Cecilia Rouse, "Orchestrating impartiality: The impact of "blind" auditions on female musicians," *American Economic Review*, 2000, *90* (4), pp. 715–741.

Guilbeault, Douglas, Solène Delecourt, Tasker Hull, Bhargav Srinivasa Desikan, Mark Chu, and Ethan Nadler, "Online images amplify gender bias," *Nature*, 2024, pp. 1–7.

Handlan, Amy and Haoyu Sheng, "Gender and Tone in Recorded Economics Presentations: Audio Analysis with Machine Learning," Technical Report 2023.

Hangartner, D., D. Kopp, and M. Siegenthaler, "Monitoring Hiring Discrimination through Online Recruitment Platforms," *Nature*, 2021, *589*, 572—576.

Kenneth, J Arrow, "The Theory of Discrimination," *Discrimination in Labor Markets*, 1973, *3.*

Kessler, Judd B, Corinne Low, and Colin D Sullivan, "Incentivized resume rating: Eliciting employer preferences without deception," *American Economic Review*, 2019, *109* (11), 3713–44.

_ , _ , and Xiaoyue Shan, "Lowering the playing field: Discrimination through sequential spillover effects," Technical Report, mimeo 2022.

Kline, Patrick, Evan K Rose, and Christopher R Walters, "Systemic Discrimination Among Large US Employers," *Quarterly Journal of Economics*, 2022, *137* (4), pp 1963–2036.

_ , _ , and _ , "A Discrimination Report Card," Technical Report, NBER 2023.

Kroft, Kory, Fabian Lange, and Matthew J Notowidigdo, "Duration dependence and labor market conditions: Evidence from a field experiment," *The Quarterly Journal of Economics*, 2013, *128* (3), 1123–1167.

Lavy, Victor and Edith Sand, "On the Origins of Gender Gaps in Human Capital: Short- and Long-Term Consequences of Teachers' Biases," *Journal of Public Economics*, 2018, *167(C)*, 263–269.

Loyalka, Prashant, Ou Lydia Liu, Guirong Li, Igor Chirikov, Elena Kardanova, Lin Gu, Guangming Ling, Ningning Yu, Fei Guo, Liping Ma et al., "Computer science skills across China, India, Russia, and the United States," *Proceedings of the National Academy of Sciences*, 2019, *116* (14), 6732–6736.

Lundberg, Shelly J. and Richard Startz, "Private Discrimination and Social Intervention in Competitive Labor Market," *American Economic Review*, 1983, *73* (3), 340–347.

Miric, Milan and Pai-Ling Yin, "Population-Level Evidence of the Gender Gap in Technology Entrepreneurship," 2020.

Mobius, Markus M., Muriel Niederle, Paul Niehaus, and Tanya S. Rosenblat, "Managing Self- Confidence: Theory and Experimental Evidence," *Management Science*, 2022, *68* (11), 7793–8514.

Mocanu, Tatiana, "Designing Gender Equity: Evidence from Hiring Practices and Committees," 2023.

Murciano-Goroff, Raviv, "Missing Women in Tech: The Role of Self-Promotion in the Labor Market for Software Engineers," 2018.

Neumark, David, "Detecting discrimination in audit and correspondence studies," *Journal of Human Resources*, 2012, *47* (4), 1128–1157.

_ , Roy J Bank, and Kyle D Van Nort, "Sex Discrimination in Restaurant Hiring: An Audit Study," *Quarterly Journal of Economics*, 1996, *111* (3), pp 915–941.

Petrie, Ragan and Adam Eric Greenberg, "Communication Architecture Affects Gender Differences in Negotiation," Technical Report, SSRN Discussion Paper 2023.

Pettigrew, Thomas F and Linda R Tropp, "A meta-analytic test of intergroup contact theory," *Journal of Personality and Social Psychology*, 2006, *90* (5), 751–783.

Phelps, Edmund S, "The statistical theory of racism and sexism," *The American Economic Review*, 1972, *62* (4), 659–661.

Reuben, Ernesto, Matthew Wiswall, and Basit Zafar, "Preferences and biases in educational choices and labour market expectations: Shrinking the black box of gender," *The Economic Journal*, 2017, *127* (604), 2153–2186.

_ , Paola Sapienza, and Luigi Zingales, "How stereotypes impair women's careers in science," *Proceedings of the National Academy of Sciences*, 2014, *111* (12), 4403–4408.

Rivera, Lauren A and Jayanti Owens, "Glass Floors and Glass Ceilings: Sex Homophily and Heterophily in Job Interviews," *Social Forces*, 2015.

Roussille, Nina, "The central role of the ask gap in gender pay inequality," 2020.

Spencer, Steven J, Christine Logel, and Paul G Davies, "Stereotype threat," *Annual review of psychology*, 2016, *67*, 415–437.

Terrell, Josh, Andrew Kofink, Justin Middleton, Clarissa Rainear, Emerson Murphy-Hill, Chris Parnin, and Jon Stallings, "Gender differences and bias in open source: Pull request acceptance of women versus men," *PeerJ Computer Science*, 2017, *3*, e111.

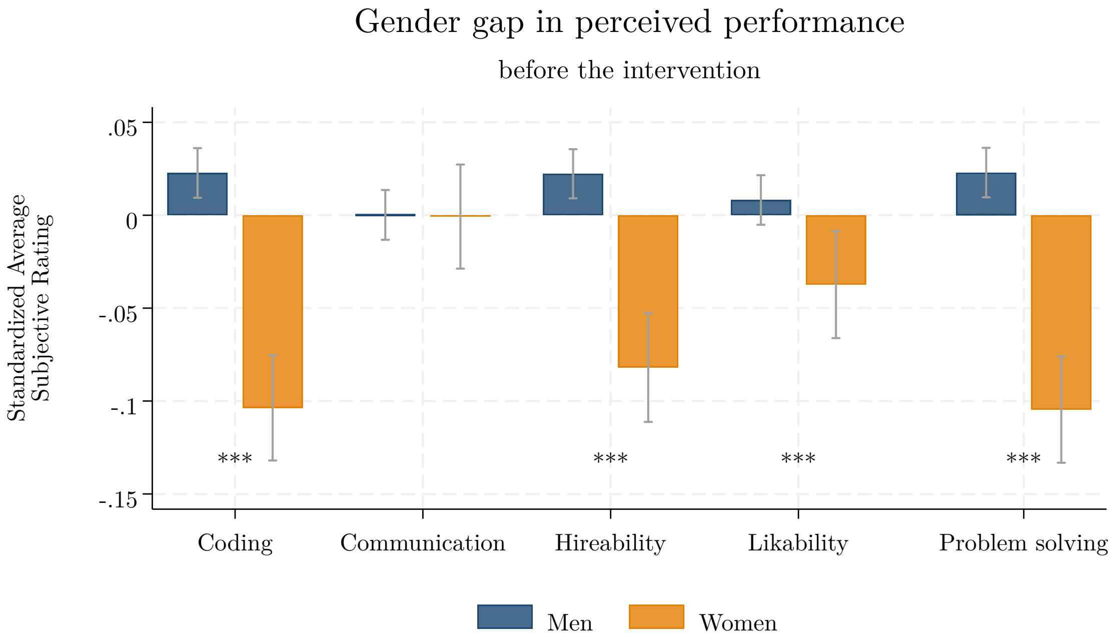
Terrier, Camille, "Boys Lag Behind: How Teachers' Gender Biases Affect Student Achievement," *Economics of Education Review*, 2020, *77.*

Vedres, Balazs and Orsolya Vasarhelyi, "Gendered behavior as a disadvantage in open source software development," *EPJ Data Science*, 2019, *8* (1), 25.

West, Candace and Don H. Zimmerman, "Doing Gender," *Gender and Society*, 1987, *1* (2), 125–151.

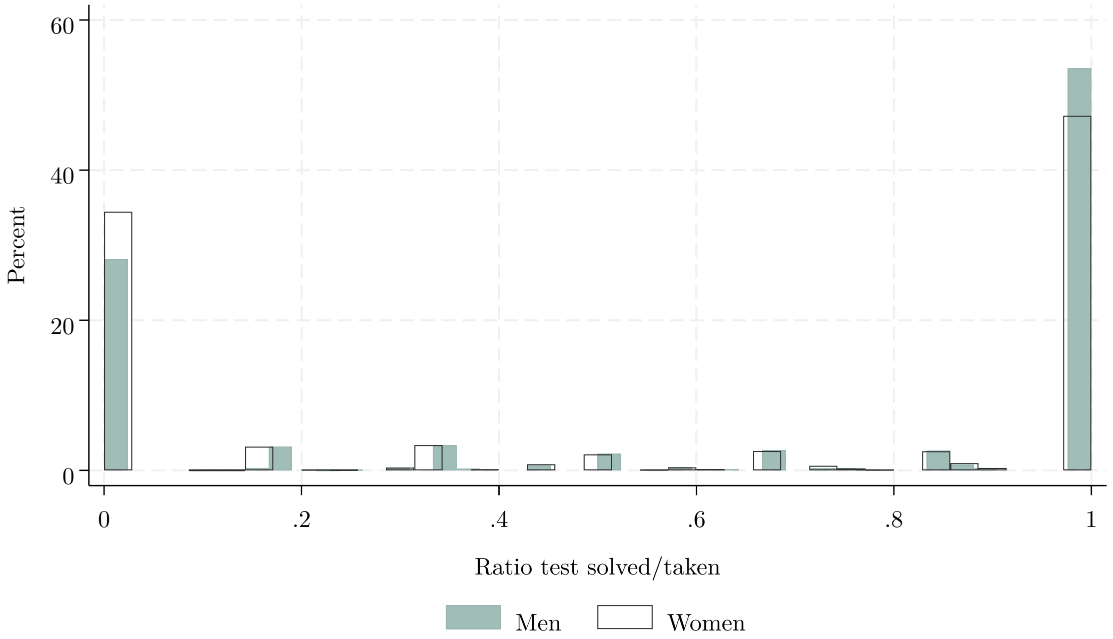
Tables and Figures

Figure 1: Pre-intervention Gender Gaps – Whole Sample



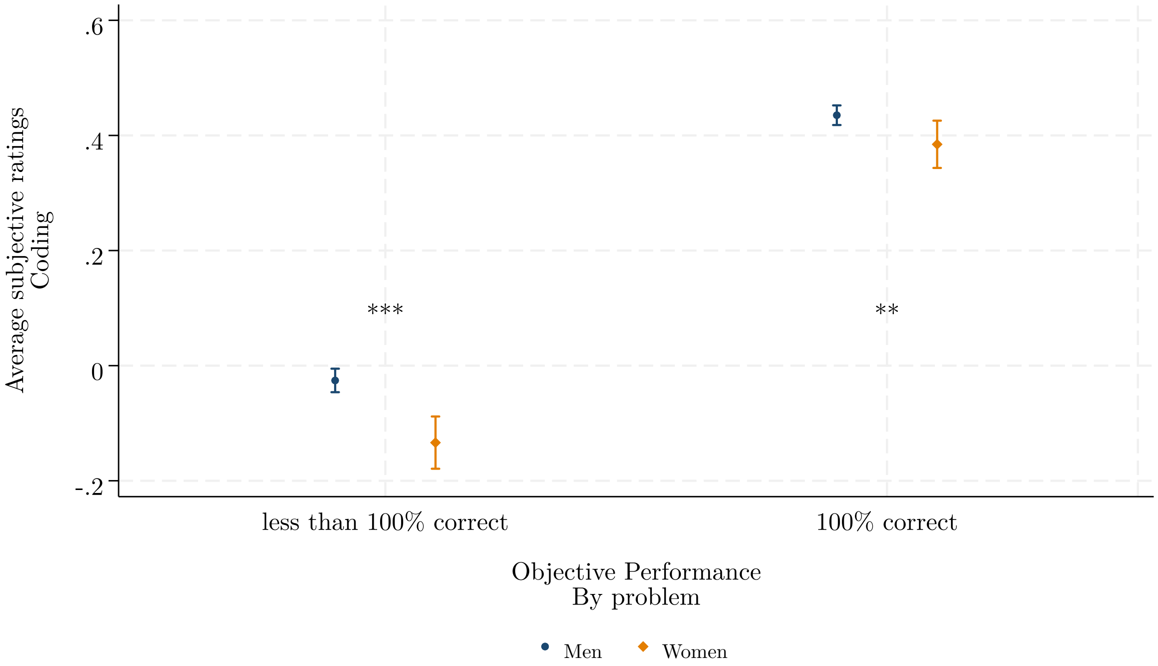
Notes: This figure shows the gender gap in peer-rated performance in five categories for standardized variables: coding, communication, hirability, likability and problem solving, for the whole sample. Stars above a category indicate statistical significance of the gap at the one percent level, and the 95-percent confidence intervals of each bar are shown in gray.

Figure 2: Distribution of Objective Performance by Gender

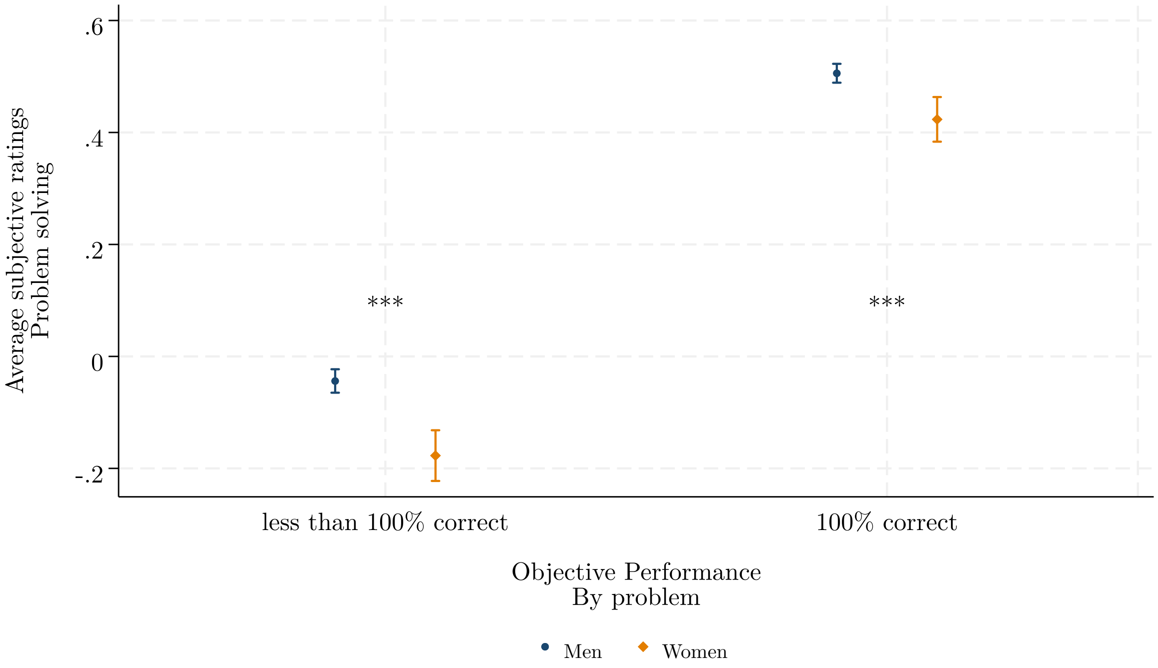


Notes: The figure presents the distribution of our objective performance measure (share of tests passed) by gender. As we describe in Section 4, these "unit tests" indicate whether the code ran and produced the correct answers to pre-defined test cases.

Figure 3: Subjective Ratings by Objective Score — Coding and Problem Solving



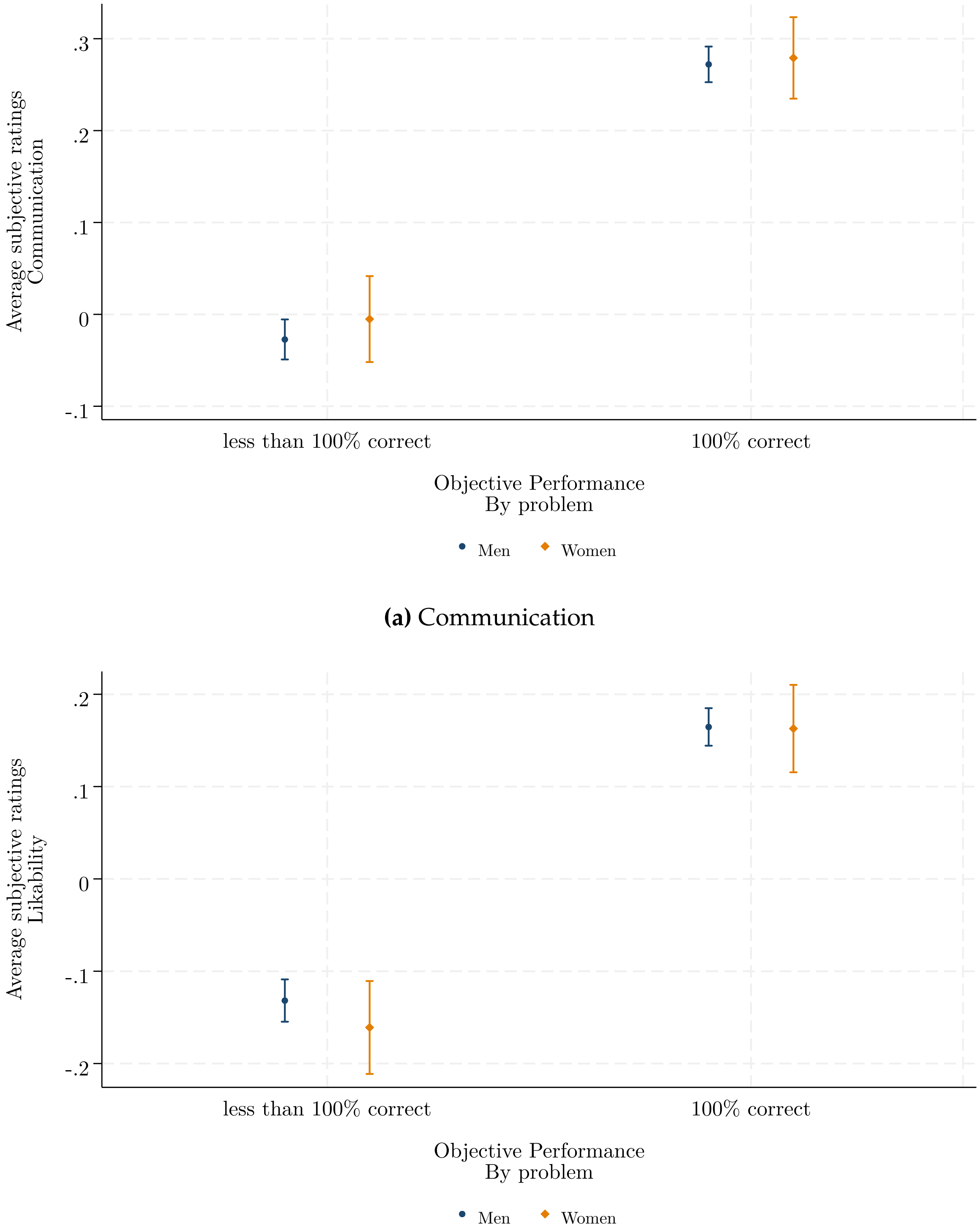
(a) Coding



(b) Problem solving

Notes: This figure shows the average subjective ratings for coding (Panel A) and problem solving (Panel B) for high and low quality code blocks. Reflecting the bimodal distribution of objective performance, we define high quality as passing all tests. Results for men are in blue, and results for women are in orange.

Figure 4: Subjective Ratings by Objective Score — Communication and Likability

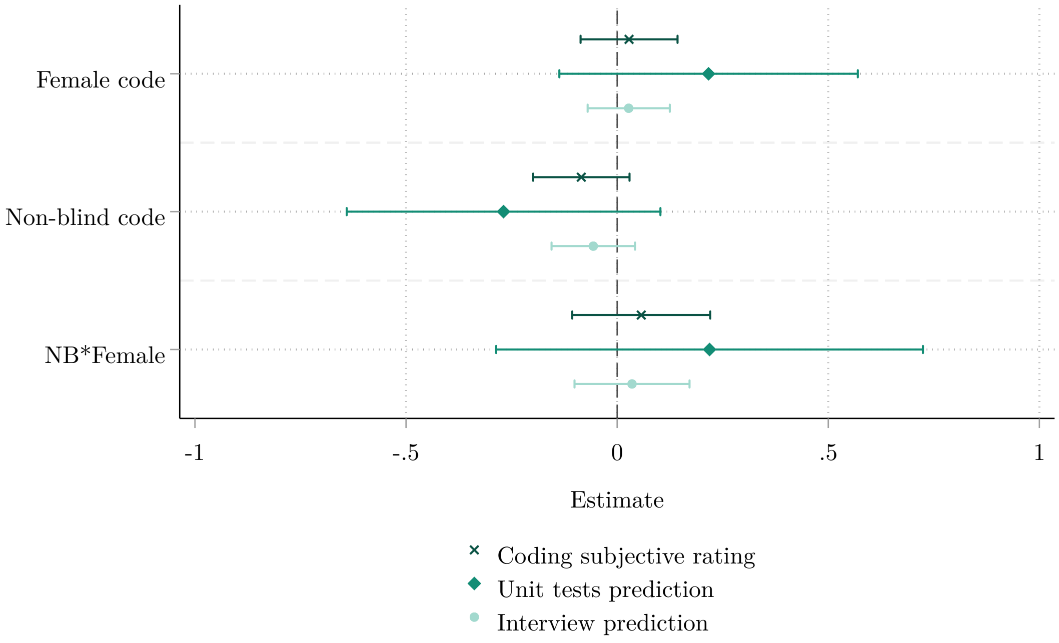


(a) Communication

(b) Likability

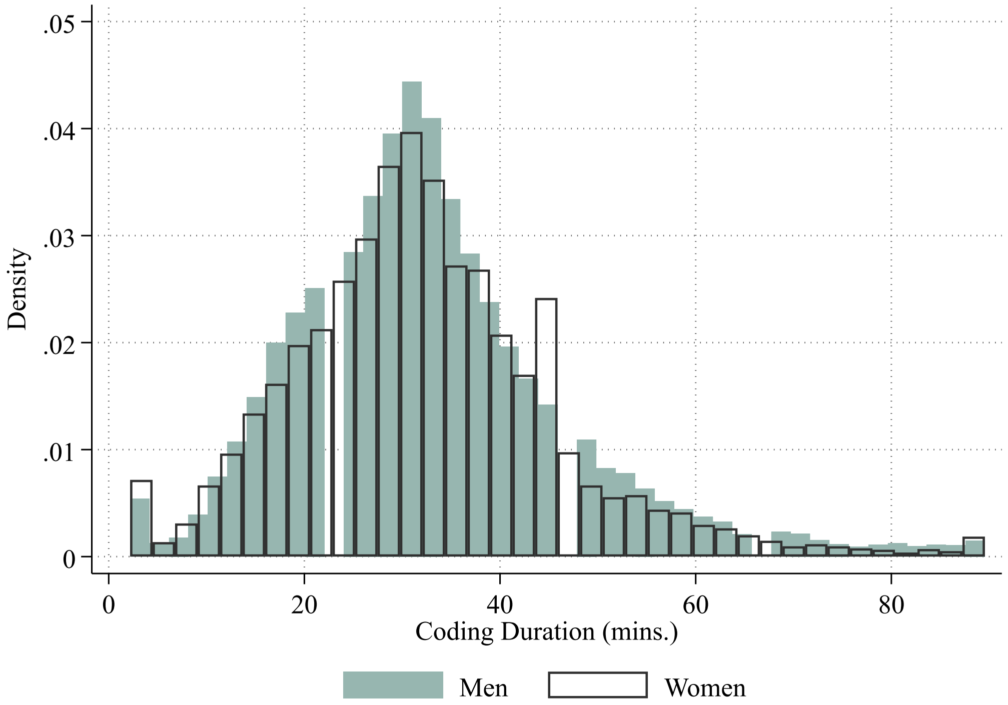
Notes: This figure shows the average subjective ratings for communication (Panel A) and likability (Panel B) for high and low quality code blocks. Reflecting the bimodal distribution of objective performance, we define high quality as passing all tests. Results for men are in blue, and results for women are in orange.

Figure 5: Blinding Experiment — Effect Of Blinding On Gender Gaps



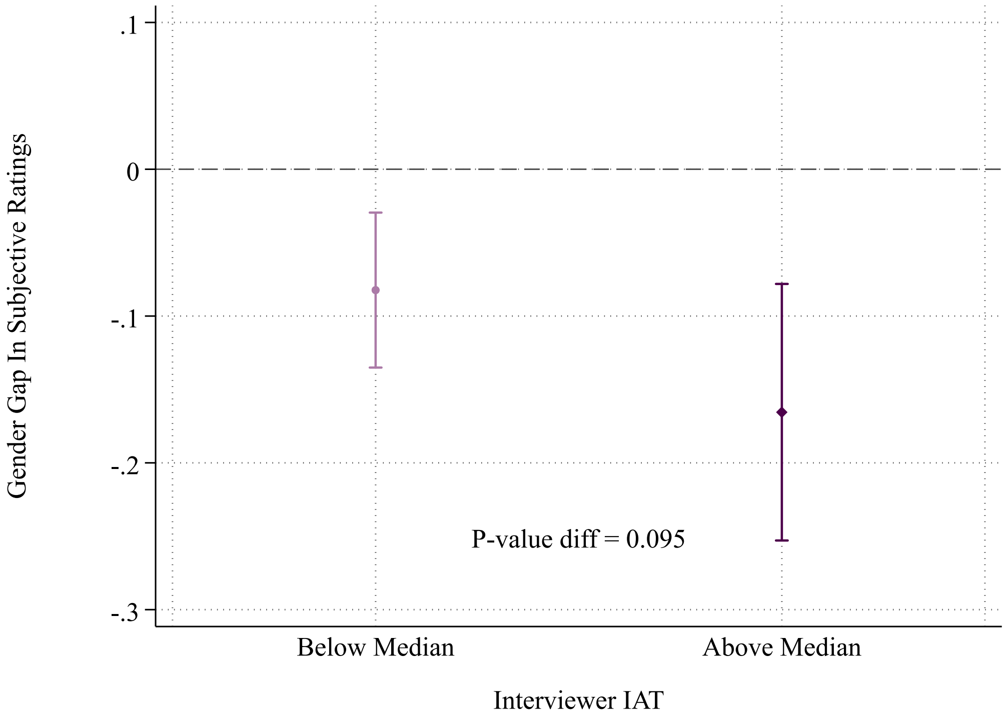
Notes: This figure shows the results from Experiment II (see Section 5). The regression specification is as described in Equation (3), controlling for evaluator fixed effects. The dependent variables are the (standardized) subjective coding ratings, participants' prediction of the unit tests passed by the code script and their prediction of the coder's probability of passing the interview. The 95-percent confidence intervals shown are based on standard errors clustered at the evaluator level. Corresponding estimates are presented in Table 3.

Figure 6: Coding Duration By Gender



Notes: This figure shows the coding duration in minutes by gender in the experimental sample.

Figure 7: Gender Gap By Evaluator IAT



Notes: This figure shows the gender gap in ratings by evaluator's IAT. Average IAT score is calculated at the MSA level. MSAs are then classified as having either below or above median IAT score relative to other geographic areas. The distribution of IAT score is presented in Figure C1. Evaluators' graduating institutions are matched to their MSA allowing us to classify evaluators to below (above) median if they graduated from an institution located in an MSA with a below (above) median IAT score. Evaluators' institutions are obtained from LinkedIn data as described in Section 4.3. IAT scores are from the Gender-Science IAT module for the years 2018 and 2019 of the Harvard Implicit Project. Corresponding estimates are presented in Table C1.

Table 1: Impact of the Introduction of the Automated Measure of Code Quality

Panel A: All

	Coding		**Problem solving**		**Likeability**		**Communication**		**Hirability**	
	ITT	2SLS	ITT	2SLS	ITT	2SLS	ITT	2SLS	ITT	2SLS
Treatment	0.147	0.205	0.211	0.295	0.086	0.120	0.198	0.277	0.169	0.237
s.d	(0.031)	(0.043)	(0.030)	(0.041)	(0.033)	(0.046)	(0.039)	(0.005)	(0.028)	(0.039)
P-value	0.000	0.000	0.000	0.000	0.012	0.010	0.000	0.000	0.000	0.000
N	11,029	11,029	11,029	11,029	11,029	11,029	11,029	11,029	11,049	11,049
First stage		0.714								
s.d		(0.009)								
P-value		0.000								
N		11,591								
F-stat		6084.30								

Panel B: Women

	Coding		**Problem solving**		**Likeability**		**Communication**		**Hirability**	
	ITT	2SLS	ITT	2SLS	ITT	2SLS	ITT	2SLS	ITT	2SLS
Treatment	0.092	0.135	0.188	0.276	0.054	0.080	0.183	0.269	0.175	0.257
s.d	(0.081)	(0.114)	(0.073)	(0.103)	(0.080)	(0.114)	(0.073)	(0.104)	(0.080)	(0.113)
P-value	0.258	0.239	0.012	0.008	0.497	0.482	0.013	0.010	0.030	0.024
N	2,049	2,049	2,049	2,049	2,049	2,049	2,049	2,049	2,055	2,055
First stage		0.678								
s.d		(0.016)								
P-value		0.002								
N		2,151								
F-stat		2069.16								

Panel C: Men

	Coding		**Problem solving**		**Likeability**		**Communication**		**Hirability**	
	ITT	2SLS	ITT	2SLS	ITT	2SLS	ITT	2SLS	ITT	2SLS
Treatment	0.162	0.225	0.218	0.302	0.093	0.129	0.199	0.276	0.168	0.234
s.d	(0.032)	(0.045)	(0.033)	(0.046)	(0.039)	(0.054)	(0.044)	(0.061)	(0.033)	(0.046)
P-value	0.000	0.000	0.000	0.000	0.019	0.016	0.000	0.000	0.000	0.000
N	8,980	8,980	8,980	8,980	8,980	8,980	8,980	8,980	8,994	8,994
First stage		0.721								
s.d		(0.016)								
P-value		0.000								
N		9,440								
F-stat		4392.79								

Notes: This table shows the main results from Experiment I (see Section 4). Both ITT and 2SLS models are shown, using the whole sample and splitting by gender. For each of the five dimensions on which users are rated, the coefficient on treatment in each model is shown from left to right in the upper subpanels. The first stages are shown in the lower subpanels. Standard errors are clustered at the date level, and shown in parentheses.

Table 2: Automated Measure of Code Quality and Future Labor Market Outcomes

	Ln(first salary post graduation)		
	(1)	(2)	(3)
Female	-0.063*	-0.073*	-0.074*
	(0.036)	(0.044)	(0.043)
Non white	-0.040	-0.071	-0.070
	(0.035)	(0.046)	(0.046)
Masters Degree	0.126***	0.202***	0.200***
	(0.030)	(0.032)	(0.031)
Objective Score		0.052**	0.068**
		(0.024)	(0.032)
Objective Score × Female			-0.057
			(0.054)
City FE	Yes	Yes	Yes
Higher Education Institution FE	Yes	No	No
Observations	3,625	2,297	2,297

Notes: This table presents our analysis of labor market outcomes discussed in Section 4.3 and Appendix B. The coefficients come from Mincer-type regressions where the dependent variable is the (log) first salary post graduation using observations from participants of the platform data matched with the Revelio Lab database. Controls include the number of session on the platform and whether the participant had already graduated when they took sessions on the platform. Standard errors are clustered at the city-of-residence level, and shown in parentheses.

Table 3: Blinding Experiment — Effect Of Blinding On Gender Gaps

	Subjective coding rating		Unit test prediction		Interview prediction	
	(1)	(2)	(3)	(4)	(5)	(6)
Female code	0.027	0.023	0.192	0.198	0.025	0.023
	(0.059)	(0.059)	(0.180)	(0.182)	(0.050)	(0.050)
Non-blind code	-0.075	-0.080	-0.261	-0.252	-0.153**	-0.054
	(0.059)	(0.059)	(0.192)	(0.193)	(0.051)	(0.051)
Non-blind code × Female code	0.036	0.049	0.173	0.192	0.037	0.035
	(0.084)	(0.085)	(0.261)	(0.263)	(0.070)	(0.070)
Treatment order control	Yes	Yes	Yes	Yes	Yes	Yes
Order of scripts FE	Yes	Yes	Yes	Yes	Yes	Yes
Problem FE	Yes	Yes	Yes	Yes	Yes	Yes
Evaluator FE	No	Yes	No	Yes	No	Yes
Observations	2,323	2,292	2,323	2,292	2,704	2,704

Notes: This table provides results from Experiment II (see Section 5), testing the pre-registered hypothesis that revealing gender introduces a gender gap that penalizes women. The regression specification is as described in Equation (3). The dependent variables are the (standardized) subjective coding ratings (columns 1-2), participants' prediction of the unit tests passed by the code script (columns 3-4) and their prediction of the coder's probability of passing the interview (columns 5-6). The even columns include evaluator fixed effects. Standard errors are clustered at the evaluator level, and shown in parentheses.

Table 4: Interaction Duration & Gender Gaps

	Subjective Coding Ratings					
	(1)	(2)	(3)	(4)	(5)	(6)
Female	-0.132***	-0.114***	-0.141***	-0.120***	-0.143***	-0.124***
	(0.017)	(0.013)	(0.023)	(0.016)	(0.023)	(0.016)
Coding Duration	-0.077***	-0.073***			-0.103***	-0.088***
	(0.009)	(0.006)			(0.012)	(0.008)
Coding Duration x Female	0.002	0.001			-0.013	-0.007
	(0.017)	(0.013)			(0.024)	(0.017)
Partner Coding Duration			-0.092***	-0.065***	-0.112***	-0.075***
			(0.011)	(0.007)	(0.012)	(0.007)
Partner Coding Duration x Female			-0.037	-0.038**	-0.036	-0.038**
			(0.023)	(0.016)	(0.024)	(0.016)
Partner Obj Score			0.037	-0.017	0.006	-0.040**
			(0.025)	(0.017)	(0.025)	(0.017)
Obj Score	Yes	Yes	Yes	Yes	Yes	Yes
Problem FE	Yes	Yes	Yes	Yes	Yes	Yes
Evaluator FE	Yes	No	Yes	No	Yes	No
Observations	26,593	36,680	15,345	23,472	15,345	23,472

Notes: This table provides results for the gender gap in subjective ratings testing for the hypothesis that longer interviews are associated with a higher gender gap. Columns (1)-(2) show the effect of coding duration on subjective ratings and allows for differences by gender. Columns (3)-(4) show the effect of partners' coding duration on ratings, allowing for differences by gender. Columns (5)-(6) further control for partners' objective performance to account for retaliation. The odd columns include evaluator fixed effects. All specifications control for the number of lines of code, and the number of lines written per minute.

(For Online Publication)

Appendix to

Deconding Gender Bias: The Role of Personal Interaction

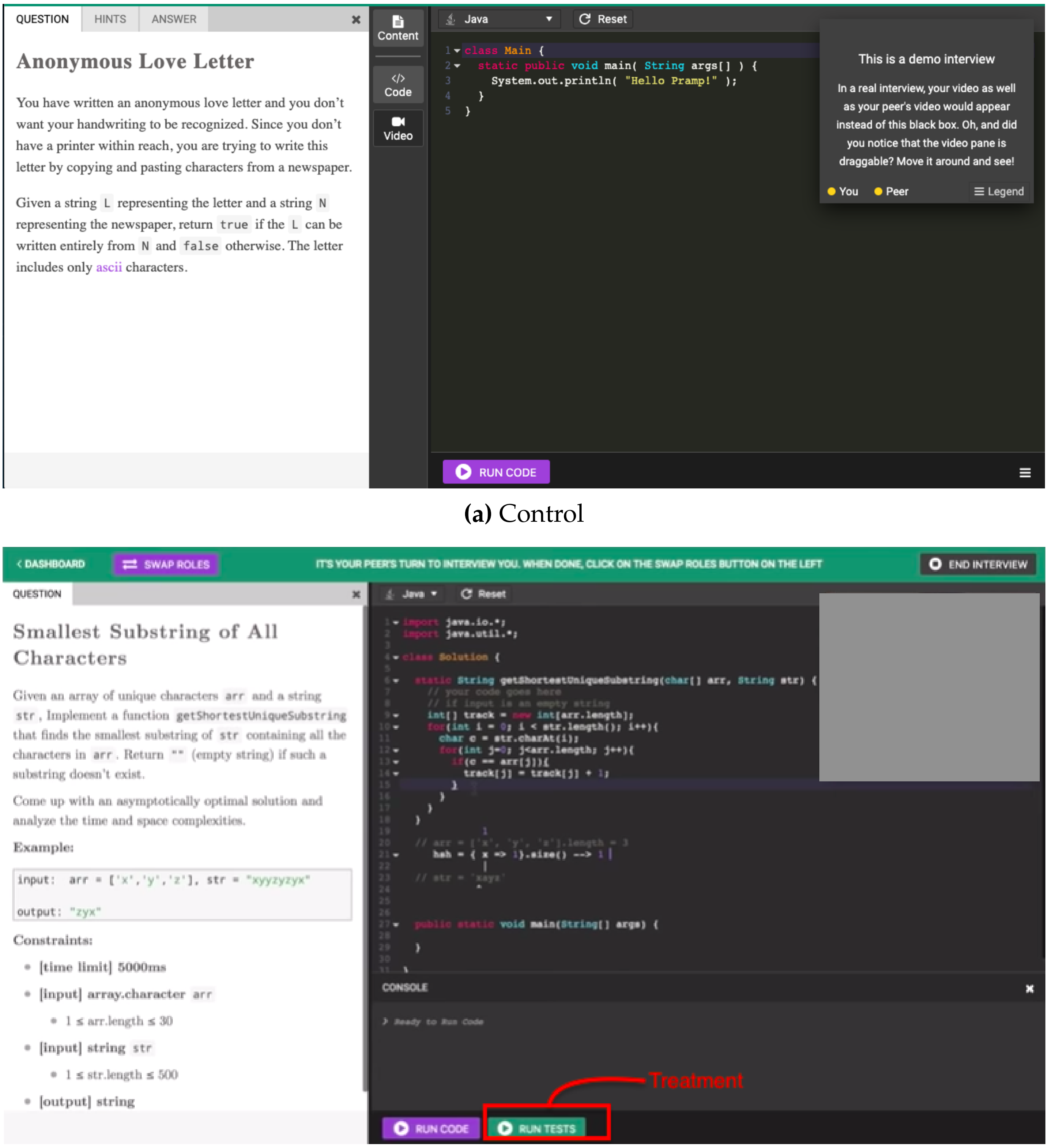
Abdelrahman Amer, Ashley C. Craig and Clémentine Van Effenterre

June 2024

List of Appendices

Appendix A Institutional details

Figure A1: Environment of the Platform (Treatment vs. Control)



Anonymous Love Letter

You have written an anonymous love letter and you don't want your handwriting to be recognized. Since you don't have a printer within reach, you are trying to write this letter by copying and pasting characters from a newspaper.

Given a string `L` representing the letter and a string `N` representing the newspaper, return `true` if the `L` can be written entirely from `N` and `false` otherwise. The letter includes only ascii characters.

```
class Main {
  static public void main( String args[] ) {
    System.out.println( "Hello Pramp!" );
  }
}
```

This is a demo interview

In a real interview, your video as well as your peer's video would appear instead of this black box. Oh, and did you notice that the video pane is draggable? Move it around and see!

(a) Control

IT'S YOUR PEER'S TURN TO INTERVIEW YOU. WHEN DONE, CLICK ON THE SWAP ROLES BUTTON ON THE LEFT

Smallest Substring of All Characters

Given an array of unique characters `arr` and a string `str`, Implement a function `getShortestUniqueSubstring` that finds the smallest substring of `str` containing all the characters in `arr`. Return `""` (empty string) if such a substring doesn't exist.

Come up with an asymptotically optimal solution and analyze the time and space complexities.

Example:

```
input:  arr = ['x','y','z'], str = "xyyzyzyx"

output: "zyx"
```

Constraints:

- [time limit] 5000ms
- [input] array.character `arr`
  - 1 ≤ arr.length ≤ 30
- [input] string `str`
  - 1 ≤ str.length ≤ 500
- [output] string

```
import java.io.*;
import java.util.*;

class Solution {

  static String getShortestUniqueSubstring(char[] arr, String str) {
    // your code goes here
    // if input is an empty string
    int[] track = new int[arr.length];
    for(int i = 0; i < str.length(); i++){
      char c = str.charAt(i);
      for(int j=0; j<arr.length; j++){
        if(c == arr[j]){
          track[j] = track[j] + 1;
        }
      }
    }
  }
             1
  // arr = ['x', 'y', 'z'].length = 3
     hsh = { x => 1}.size() --> 1 |
             |
  // str = 'xzyz'
           *


  public static void main(String[] args) {

  }

}
```

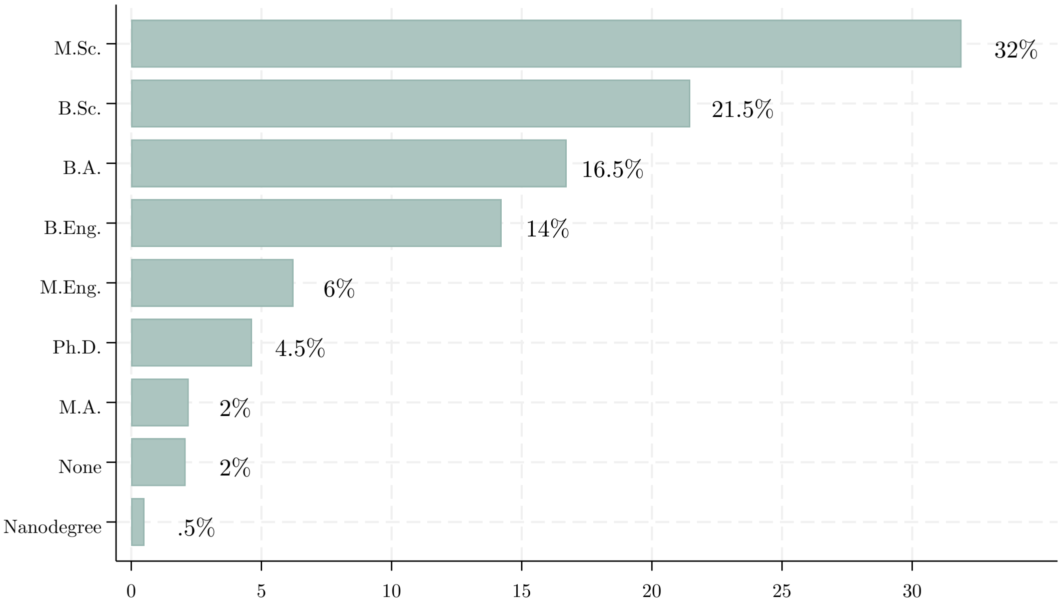
CONSOLE

```
> Ready to Run Code
```

(b) Treatment

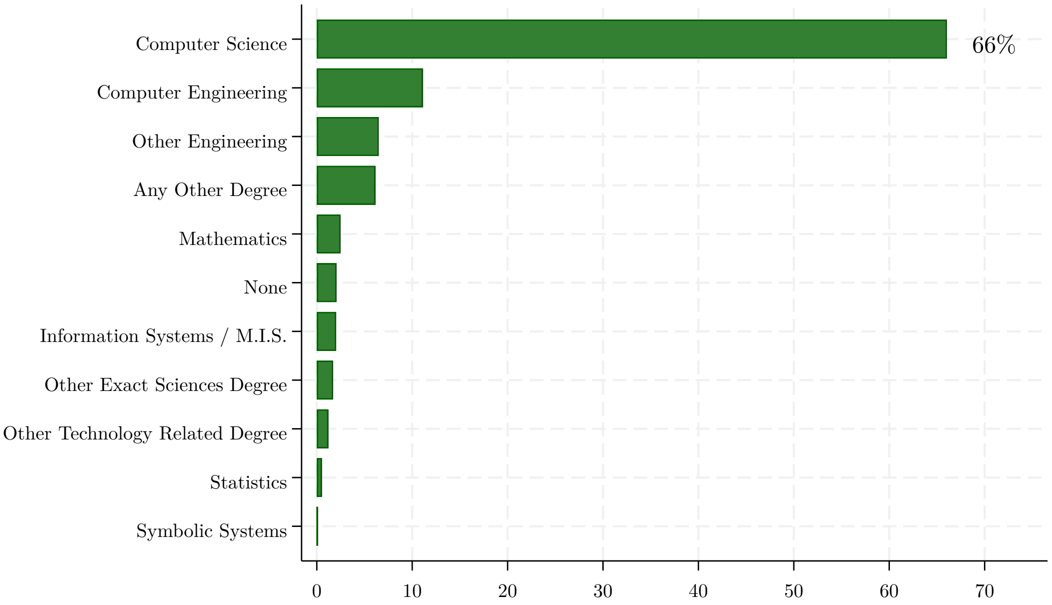
Notes: figure shows the platform layout for a mock interview. Panel (a) shows the control condition, where the code can be run but there are no build in "unit tests" to verify code quality. Panel (b) shows the treatment condition, in which a button is added to run the diagnostic tests.

Figure A2: Users' Level of Education



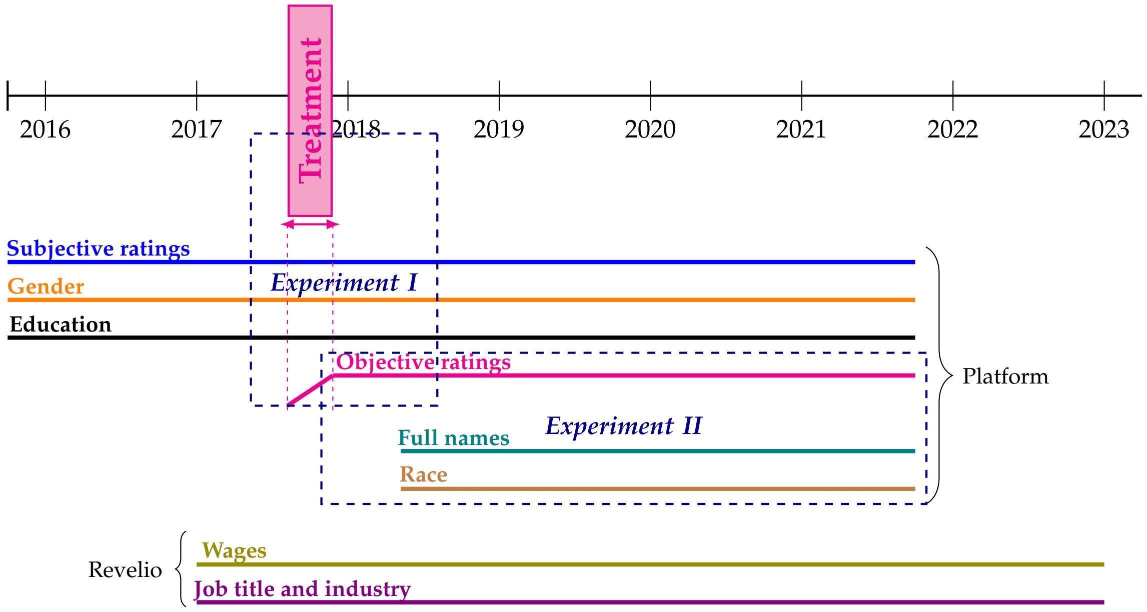
Notes: The figure presents the distribution of the level of education of users on the platform in the period covered by our first dataset (from 2015 to 2018, as described in Section 2.2).

Figure A3: Users' Field of Education



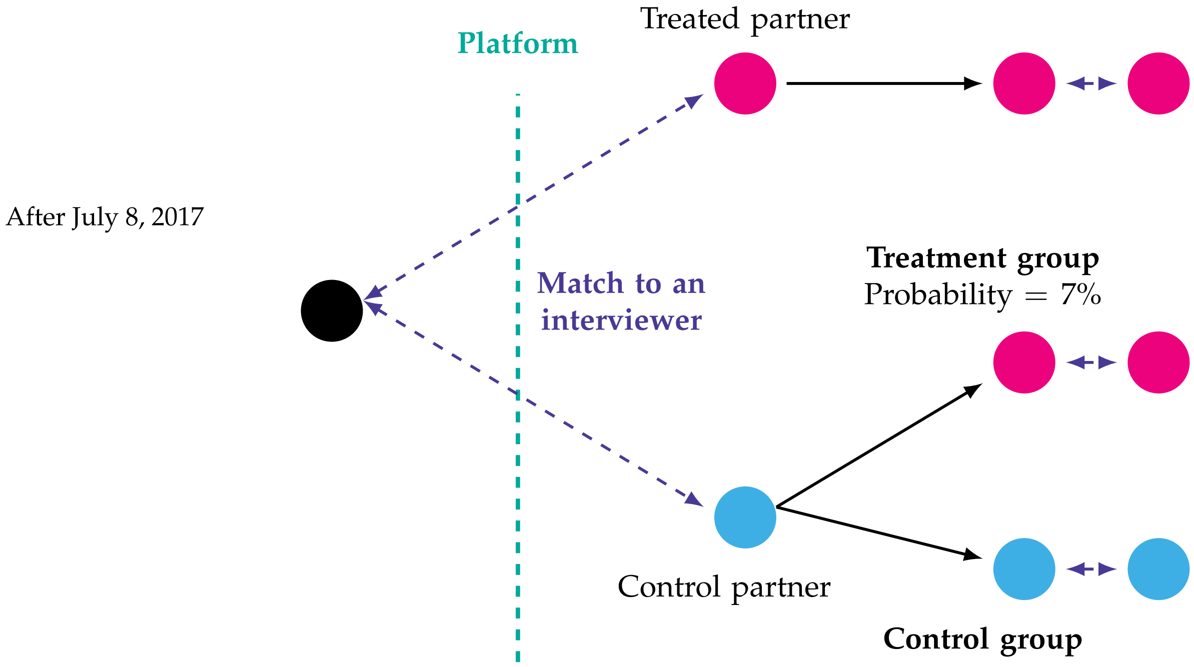
Notes: The figure presents the distribution of the field of education of users on the platform in the period covered by our first dataset (from 2015 to 2018, as described in Section 2.2).

Figure A4: Summary of Data Availability



Notes: This diagram shows the data infrastructure we use to build Experiment I and II and the validation exercise using labor market outcomes from Revelio Lab. Experiment 1 is described and analyzed in Section 4. Experiment 2 is described and analyzed in Section 5. The Revelio data are described in Section 4.3, with further discussion in Appendix B.

Figure A5: Treatment Assignment Diagram



Notes: This diagram shows how users were assigned to the treatment or to the control conditions when they sign up for an interview. If they and their partner are new users, they were randomized into treatment with 7% probability. However, if they or their partner had previously interacted on the platform as part of the treatment group, they remained in treatment.

Appendix B Labor Market Data

In this Appendix, we describe how we link our data to labor market outcomes from Revelio labs, and analyze the merged dataset. The Revelio data contain information from publicly available LinkedIn profiles, and job posting boards. These data contain close to the universe of Computer Science (CS) graduates in the US labor market, and their job spells. We also observe an estimate of their salaries imputed using job posting data, H1B-visa records and the Current Population Survey.[A.1]

One concern with such data is that there may be some degree of sample selection. For example, only high achieving graduates might have profiles. However, we have two reasons to believe that this is less of a problem in our setting than others. First, participants on the platform are actively seeking employment in a CS related position, making an online presence highly desirable if not unavoidable. Second, the US produces around 60,000 computer science baccalaureates annually, and there are about this many such degrees in the Revelio data from 2016 to 2026.[A.2]

From the set of interviewees on the platform, we select those residing in the US who have a Bachelor's or Master's degree. We then match this sample to the universe of individuals in the Revelio data who attained a CS-related degree from a US institution. We use only exact matches based on their first and last name, and degree type. Observations matched to multiple Revelio profiles are dropped.[A.3] The final sample consists of 5,126 matched CS graduates from 2016 to 2023. We have unit test data for about 50 percent of this sample.

We use a Mincer-type wage regression of log earnings on individuals' unit test scores, their characteristics, year-of-graduation and city fixed effects. The main outcome is the first salary after graduation, although we also look at average salary after graduation. Results are presented in Table 2. Column (1) shows that there is a 6.3 percent residual gender gap for computer science graduates in their first salary after graduation. In column (2), we add the average objective measure of coding quality across all sessions on the platform, the number of past sessions on the platform and

[A.1] More detail regarding the Revelio data database is available www.reveliolabs.com.

[A.2] See Loyalka et al. (2019) for a cross-country analysis of CS university graduates.

[A.3] This follows the same matching method adopted by Abramitzky et al. (2012), Abramitzky et al. (2014) and Abramitzky and Boustan (2017).

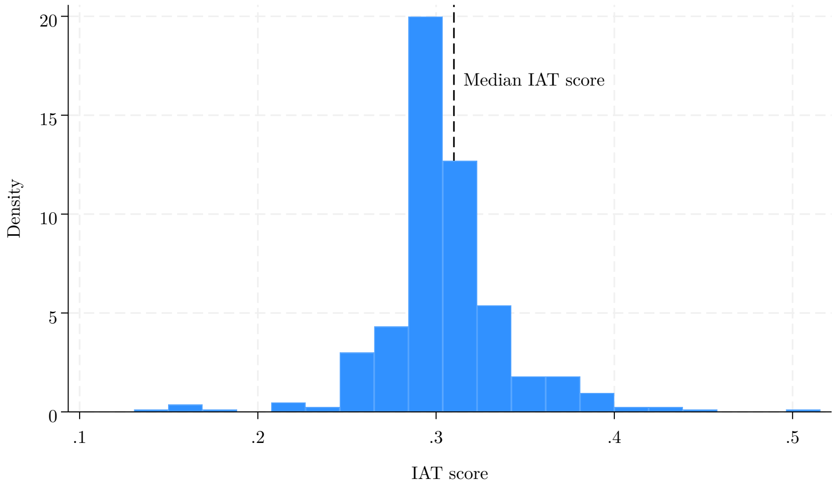
whether the participant had graduated at the time of their interview session.[A.4] We find a positive and statistically significant coefficient (0.052, SD=0.024) for the standardized objective score measure, which implies that going from the 25th to the 75th percentile of standardized score is associated with a wage increase of 4.5 percent.

Finally, we note that there is suggestive evidence of heterogenous returns of skills by gender in column 3, with little return of the objective measure of coding performance for women. However, the estimate for women is imprecise.

[A.4] To reduce noise, we also tried re-weighting the regression for the number of sessions each user had on the platform. The results are qualitatively similar.

Appendix C Implicit Bias Results

Figure C1: Distribution of IAT Scores



Notes: This figure presents the distribution of IAT scores of evaluators' metropolitan statistical areas (MSA) of graduation in our sample described in Section 4.3. The dash line indicates the US median.

Table C1: Gender Gap By Evaluator IAT

	Subjective Coding Ratings		
	Low Bias	High Bias	All
Female	-0.085***	-0.151***	-0.082***
	(0.027)	(0.050)	(0.027)
Female x High IAT			-0.083*
			(0.050)
High Score	0.475***	0.579***	0.498***
	0.028	(0.053)	(0.025)
Observations	5,730	1,672	7,402

Notes: This tables shows the gender gap in (standardized) subjective ratings for two groups. Column (1) presents the gender gap when evaluators graduated from a higher education institution located in an MSA with below-median IAT score (i.e less prejudice against women in science). Column (2) presents results when evaluators graduated from a higher education institution located in an MSA with above-median IAT score (i.e more prejudice against women in science). Column (3) tests for statistical differences in the gender gap between both groups. Objective score is controlled for in all specifications. Evaluators' institutions are obtained from LinkedIn data as described in Section 4.3. IAT scores are from the Gender-Science IAT module for the years 2018 and 2019 of the Harvard Implicit Project.

Table C2: Gender Gap in Subjective Coding Ratings — Evaluator Learning

	Subjective Coding Ratings		
	(1)	(2)	(3)
Female	-0.108***	-0.097***	-0.106***
	(0.017)	(0.021)	(0.014)
High Score	0.635***	0.636***	0.634***
	(0.014)	(0.014)	(0.014)
Coding Duration	-0.003***	-0.003***	-0.003***
	(0.000)	(0.000)	(0.000)
Evaluator Cumulative Sessions	0.002***		
	(0.001)		
Evaluator Cumulative Sessions x Female	-0.000		
	(0.001)		
Evaluator Total Sessions		0.000	
		(0.000)	
Evaluator Total Sessions x Female		-0.001	
		(0.001)	
Evaluator First Session			-0.060***
			(0.015)
Evaluator First Session x Female			-0.025
			(0.036)
Problem FE	Yes	Yes	Yes
Observations	38,256	38,256	38,256

Notes: This table shows the gender gap when controling for evaluators' experience on the platform. Column (1) controls for evaluators' cumulative number of sessions, Column (2) controls for total number of sessions on platform, and Column (3) accounts for whether this is the evaluator's first session on platform. Evaluator First Session is a dummy indicating this. Interactions with female dummy tests for whether these characteristics are associated with a different gender gap in subjective ratings.

Appendix D Experiment I: Additional Results

D.1 Explaining a Persistent Gender Gap

Our results indicate that gender gaps did not decrease with more information. While this may be due to statistical chance, it suggests that evaluators may be unduly pessimistic about men relative to women. Experiment I could not shed more direct light on prior beliefs, but we later collected information about beliefs in Experiment II. As we discuss in Section 5, we do find evidence that is consistent with evaluators discounting slightly the performance of men relative to women, compared to the true gender gap in performance as measured by the unit tests.

We can also evaluate other possibilities, one of which is that the unit tests were more informative for men than women.[A.5] To see why this could conceivably explain our results, consider an extension of the model in Section 3. Rather than the weight on the signal being the same for men and women ($s_m = s_f$), let the signal be more informative for one gender. In this case, the gender gap given signal realization θ_i is:

$$\text{Gender Gap} \mid \theta_i = \overbrace{s_m \mu_m^* + (1 - s_m)\, \mu_m}^{\textbf{Male Belief}} - \overbrace{\left[s_f \mu_f^* + \left(1 - s_f\right) \mu_f \right]}^{\textbf{Female Belief}} \tag{A.1}$$

where $s_g = \frac{\sigma_g^2}{\sigma_g^2 + \sigma_\varepsilon^2} \in (0, 1)$ is the weight placed on the signal for gender $g \in \{m, f\}$. The impact of more information on the gender gap is then:

$$d\text{Gap} = ds_m \underbrace{(\mu_m^* - \mu_m)}_{\textbf{Male Pessimism}} - ds_f \underbrace{\left(\mu_f^* - \mu_f\right)}_{\textbf{Female Pessimism}} \tag{A.2}$$

where ds_g is the marginal impact of information on s_g.

This highlights the two reasons why the gender gap could persist with more information. First, $\mu_m^* - \mu_m$ may larger than $\mu_f^* - \mu_f$, which would imply that evaluators are unduly pessimistic about men compared to women, relative to the true performance.

Second, the impact on the signal may be larger for men than for women, ($ds_m > ds_f$). This could occur for example if men are assigned problems which are more informative. However Table D4 shows that men and women face similar problems. This is true in terms of difficulty, as measured by average performance of others on

[A.5] Beyond these two explanations, the differential impact could be due to a non-linear mapping between beliefs and ratings, or to statistical chance.

those problems. It is also true for problems with different cross-sectional variances in performance, which could indicate that some tests are more discerning than others. Furthermore, Figure D9 shows that the gender difference in impact is present even when we strict to high-variance or low-variance problems.

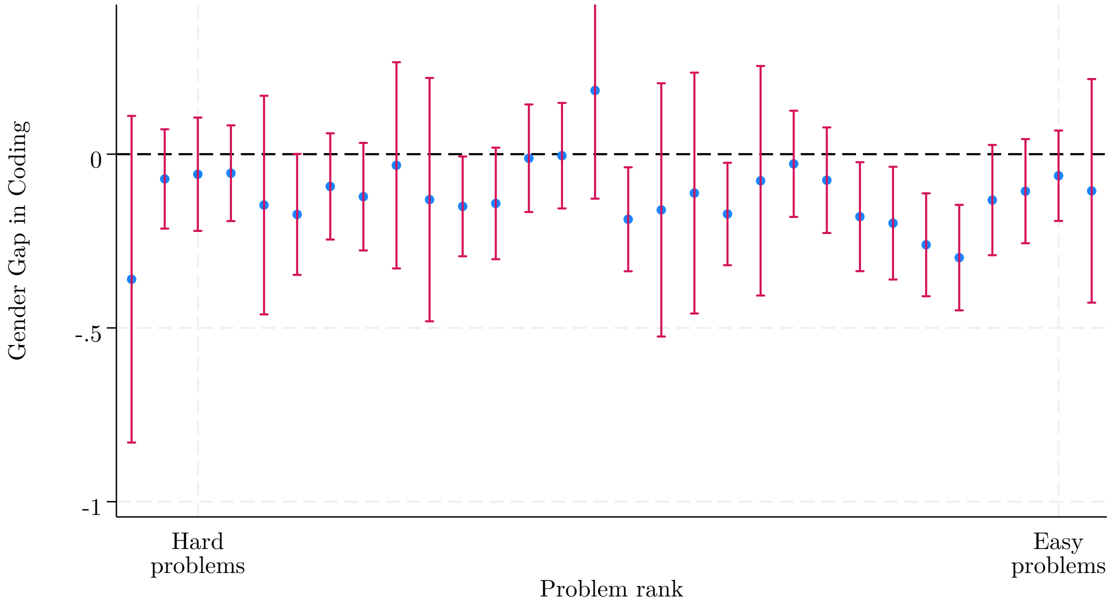
D.2 Complier Characteristics.

We show observable characteristics of compliers in Table D6.[A.6] Characteristics are similar between treated and untreated compliers. Column (5) presents characteristics for never-takers. The comparisons in Table D6 reveal that the representation of most subgroups among compliers is similar to the overall sample, although compliers do have slightly less experience. However, the gender gap in activation translates into under-representation of women among the compliers.

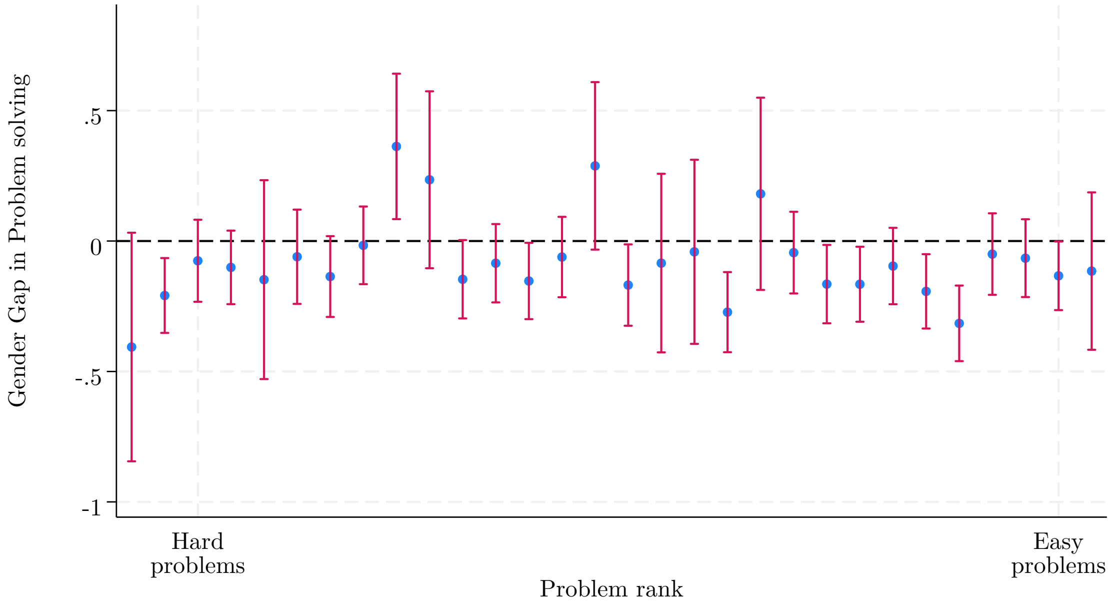
[A.6] Following Abadie (2003), these characteristics are recovered by calculating the fraction of compliers in different subsamples. The results come an IV procedure where the dependent variable is $X_i D_i$ (Column 4) and $X_i(1 - D_i)$, using T_i as an instrument for D_i.

D.3 Additional Figures and Tables

Figure D1: Pre-treatment Gender Gaps by Problem Difficulty



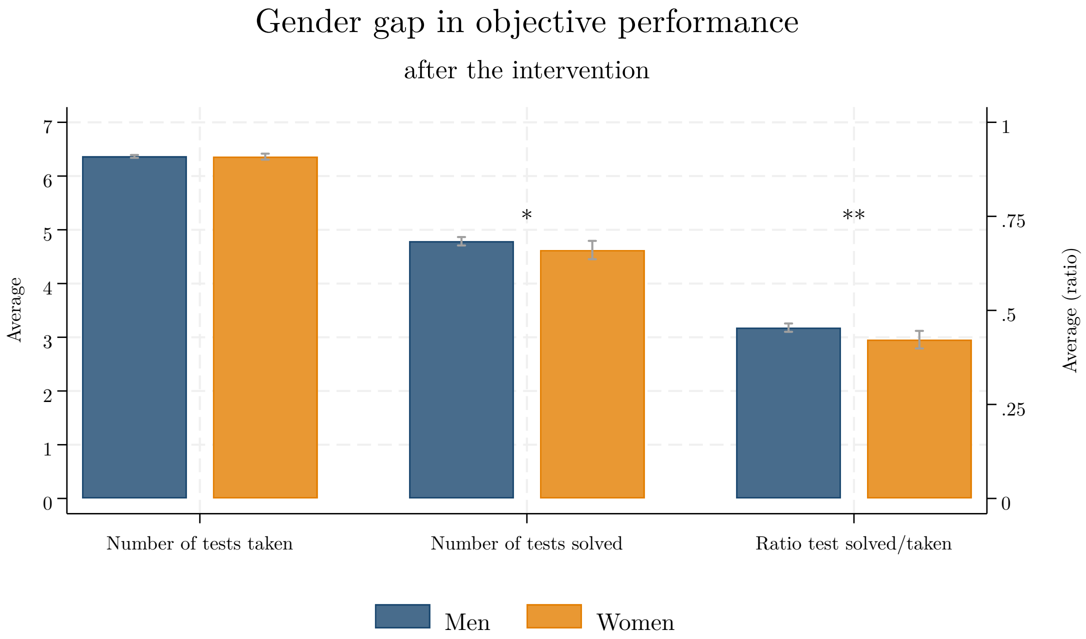
(a) Pre-treatment gender gap in coding ratings



(b) Pre-treatment gender gap in problem solving ratings

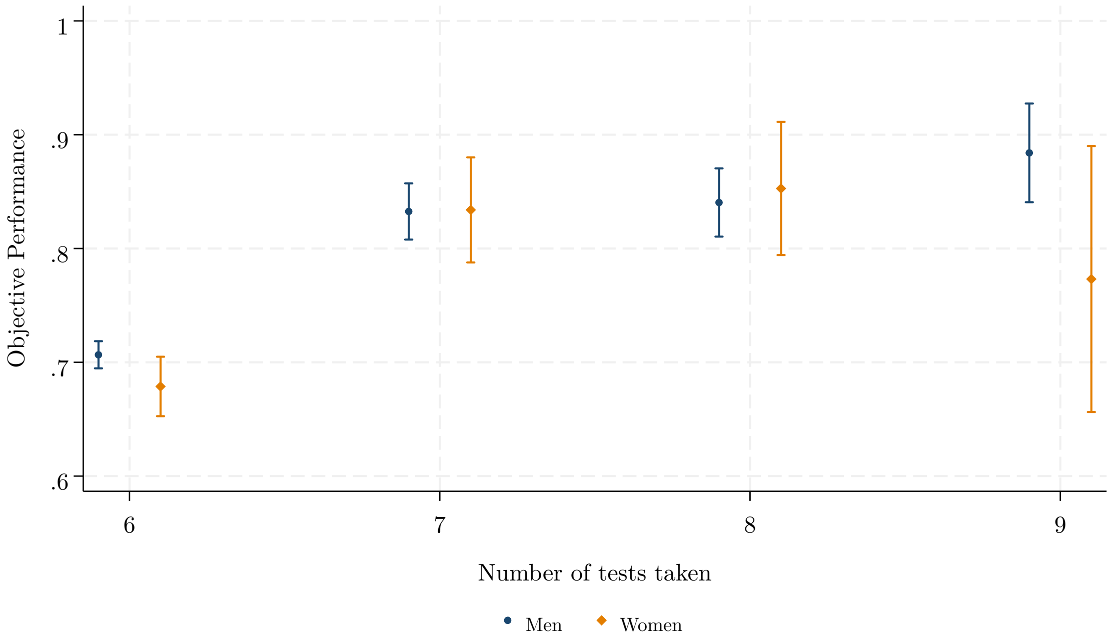
Notes: This figure plots gender gaps in subjective ratings for coding and problem solving by problem difficulty in the pre-intervention period of Experiment I (2015 to 2018 as described in Section 2.2). Problem difficulty is computed using the average objective performance of users in the post-intervention period.

Figure D2: Gender Gap In Objective Performance After The Intervention



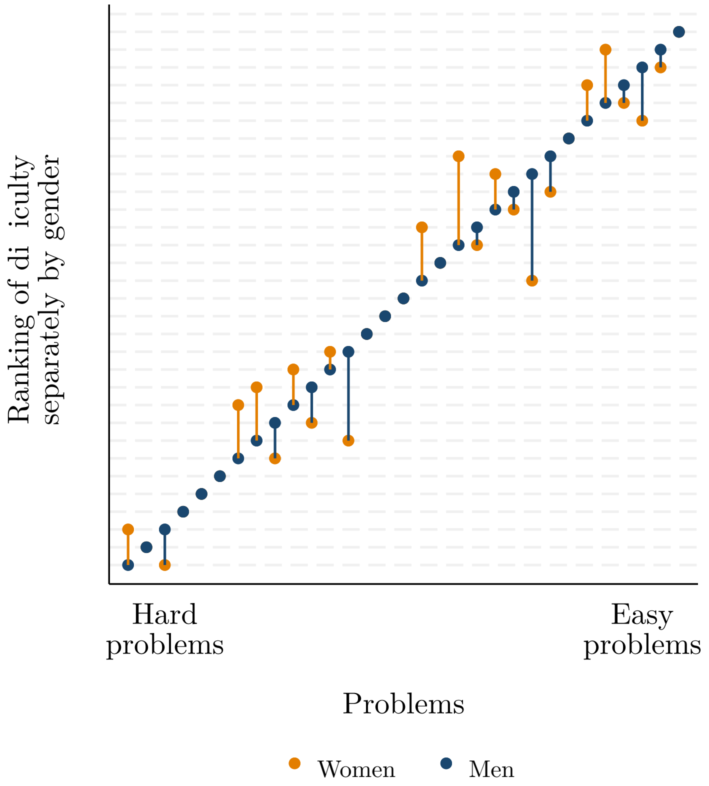
Notes: This figure presents the level of objective performance for men and women after the intervention in terms of number of tests taken, number of tests solved or failed (right y-axis), and the share of unit tests passed (right y-axis).

Figure D3: Objective Performance by Number of Tests Taken



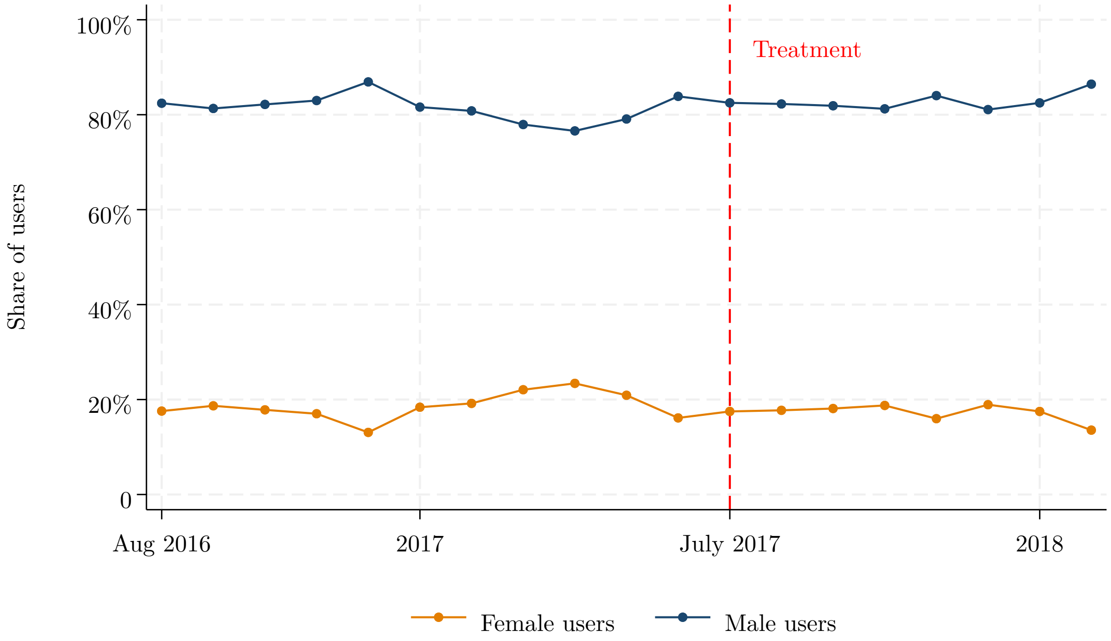
Notes: This figure shows the average objective coding performance (number of tests completed over test passed) by how many tests were taken, separately for male and female users.

Figure D4: Ranking of problems by gender



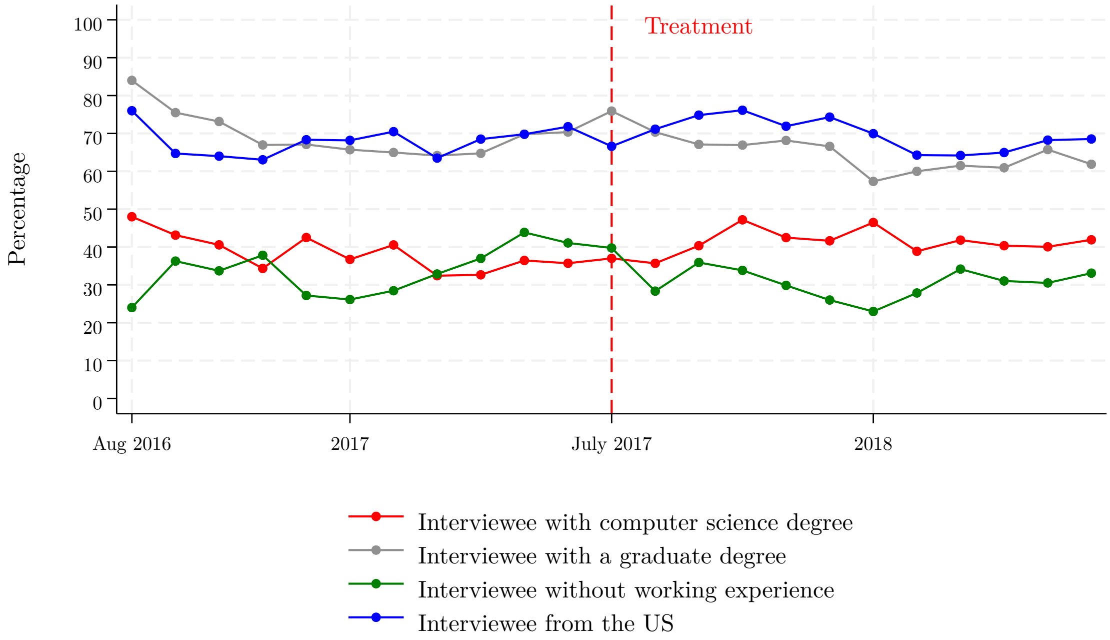
Notes: This figure shows the relative ranking of problems' difficulty by gender. The ranking is proxied by the average performance of users for each problem on the unit tests. The orange vertical lines show any positive or negative deviation of female users' ranking compared to male users' ranking.

Figure D5: Share of male and female users over time



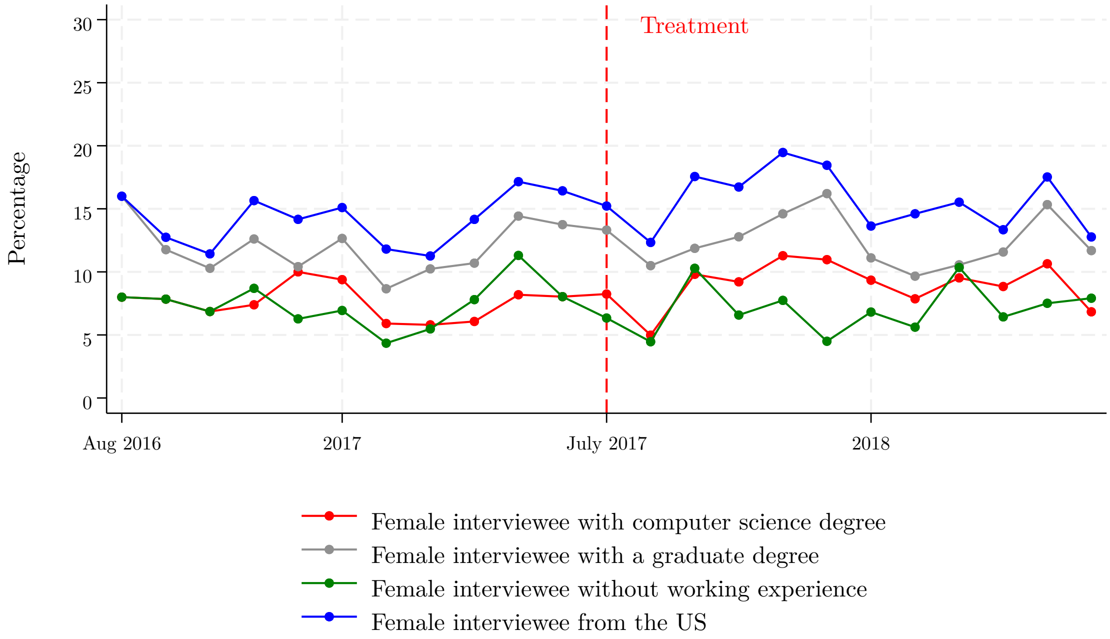
Notes: This figure shows the evolution of the shares of female and male users on the platform before and after the unit tests began to be introduced. The vertical red line shows when the introduction started.

Figure D6: Evolution of First-Time Users' Characteristics



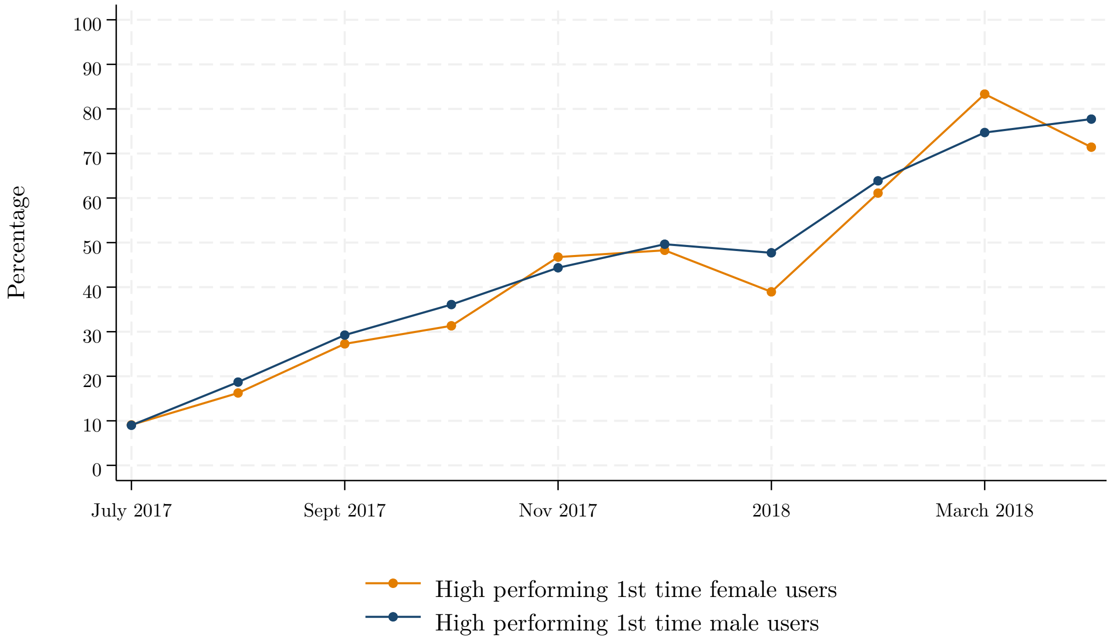
Notes: The figure presents the evolution of first-time users' characteristics averaged by month around the date that the unit tests began to be introduced. The vertical red line shows when the introduction started.

Figure D7: Evolution of First-Time Female Users' Characteristics



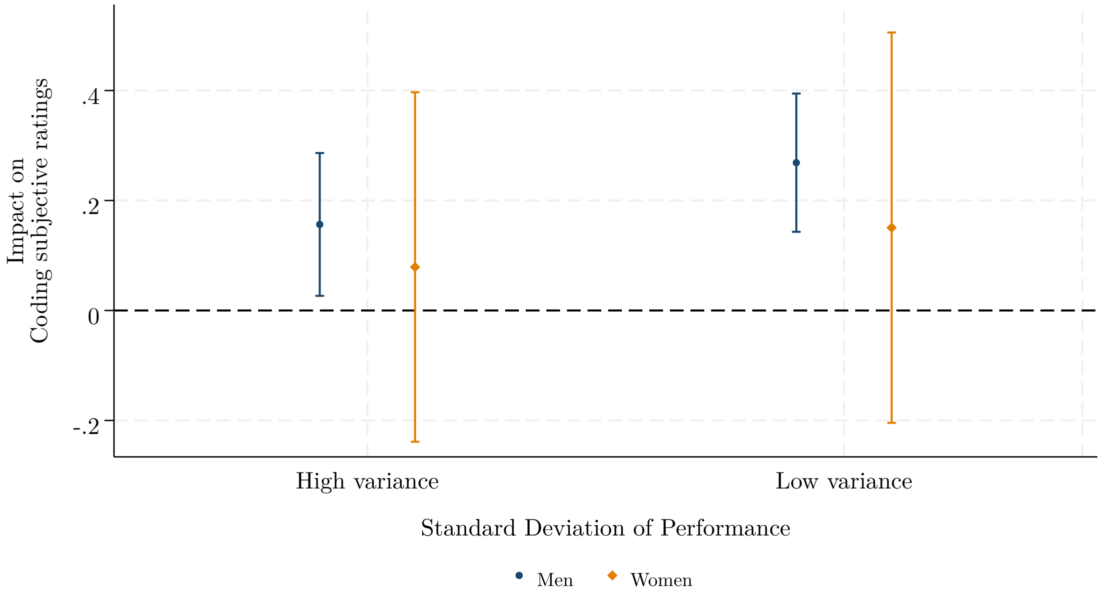
Notes: The figure presents the evolution of first-time female users' characteristics averaged by month around the date that the unit tests began to be introduced. The vertical red line shows when the introduction started.

Figure D8: Share of High-Performing First-Time Female and Male Users



Notes: The figure presents the evolution of the share of high-performing first-time female and male users by month after the unit tests began to be introduced. High-performing users are defined as those passing all unit tests taken for a given problem.

Figure D9: Men's and Women's Treatment Effects on Subjective Rating by Problem — Variation of Performance



Notes: This figure shows the estimates of Equation (8) where the dependent variable is the subjective rating in coding, separately by problem type (with high and low cross-sectional variance in performance) and gender.

Table D1: Descriptive Statistics — Dec 2015-April 2018

Number of sessions	30,466
Number of interviewees	12,960
Number of interviewers	12,707
Number of problems	31
Share of female interviewees	16.46
Share of female interviewers	16.44

Panel A: All

Variable	**Mean**	**Std. Dev.**	**Min.**	**Max.**	**N**
Country: USA	0.716	0.451	0	1	60,513
Interviewee's deg.: computer science	0.661	0.473	0	1	60,483
Interviewee without working experience	0.267	0.442	0	1	60,508
Interviewee with a graduate degree	0.45	0.497	0	1	60,513
Interviewee Preparation Level	2.897	0.798	1	5	60,307

Panel B: Women

Variable	**Mean**	**Std. Dev.**	**Min.**	**Max.**	**N**
Country: USA	0.796	0.403	0	1	9,959
Interviewee's degree : computer science	0.652	0.476	0	1	9,959
Interviewee without working experience	0.309	0.462	0	1	9,957
Interviewee with a graduate degree	0.514	0.5	0	1	9,959
Interviewee Preparation Level	2.779	0.786	1	5	9,940

Panel C: Men

Variable	**Mean**	**Std. Dev.**	**Min.**	**Max.**	**N**
Country: USA	0.701	0.458	0	1	50,554
Interviewee's deg.: computer science	0.662	0.473	0	1	50,524
Interviewee without working experience	0.259	0.438	0	1	50,551
Interviewee with a graduate degree	0.437	0.496	0	1	50,554
Interviewee Preparation Level	2.92	0.799	1	5	50,367

Notes: This table shows descriptive statistics for the sample of interviews we analyze in Section 2.3, before the introduction of objective code quality measures. The top panel shows key aggregate statistics. The lower three panels present summary statistics for interviewee characteristics overall, for men and for women respectively.

Table D2: Gender Gap in Subjective Ratings Pre-Intervention

	Coding				
	(1)	(2)	(3)	(4)	(5)
Interviewee female	-0.127***	-0.121***	-0.121***	-0.121***	-0.118***
	(0.016)	(0.016)	(0.016)	(0.018)	(0.019)
Observations	26,306	25,952	25,952	25,932	25,952
	Problem Solving				
	(1)	(2)	(3)	(4)	(5)
Interviewee female	-0.126***	-0.110***	-0.110***	-0.111***	-0.117***
	(0.016)	(0.016)	(0.016)	(0.018)	(0.018)
Observations	26,306	25,952	25,952	25,932	25,952
	Likability				
	(1)	(2)	(3)	(4)	(5)
Interviewee female	-0.042***	-0.042***	-0.042***	-0.043**	-0.045**
	(0.015)	(0.015)	(0.015)	(0.017)	(0.018)
Observations	26,306	25,952	25,952	25,932	25,952
	Communication				
	(1)	(2)	(3)	(4)	(5)
Interviewee female	-0.000	0.000	-0.000	-0.001	0.006
	(0.016)	(0.016)	(0.016)	(0.019)	(0.019)
Observations	26,306	25,952	25,952	25,932	25,952
	Hireability				
	(1)	(2)	(3)	(4)	(5)
Interviewee female	-0.104***	-0.101***	-0.101***	-0.102***	-0.095***
	(0.016)	(0.016)	(0.016)	(0.019)	(0.019)
Observations	26,264	25,911	25,911	25,911	25,911
Interviewee's controls	No	Yes	Yes	Yes	Yes
Interviewer's controls	No	Yes	Yes	Yes	Yes
Problem FE	No	No	No	Yes	No
Date FE	No	No	No	No	Yes

Notes: This table shows the estimation of the gender gap in subjective ratings pre-intervention from December 2015 to July 2017, using a linear regression model in which we progressively add controls (see Section 2.3). In column 2, we add sociodemographic controls, such as interviewer's and interviewee's years of experience, a dummy variable for each level area of education and highest educational level, and self-reported level of preparedness. In column 3 to 5, we control for the gender of the interviewer. In columns 4, we add problem fixed effects. In columns 5, we add date-of-interview fixed effects.

Table D3: Subjective Ratings Pre-Intervention

Panel A: All

Variable	Mean	Std. Dev.	Min.	Max.	N
Score in coding	-0.048	1.003	-2.981	1.12	26,306
Score in problem solving	-0.047	0.984	-2.62	1.264	26,306
Score in likability	0.075	0.932	-2.738	1.095	26,306
Score in communication	-0.055	0.992	-3.413	1.042	26,306
Score in hireability	0.004	0.998	-3.042	1.046	26,334

Panel B: Women

Variable	Mean	Std. Dev.	Min.	Max.	N
Score in coding	-0.152	0.995	-2.981	1.12	4,731
Score in problem solving	-0.15	0.987	-2.62	1.264	4,731
Score in likability	0.041	0.940	-2.738	1.095	4,731
Score in communication	-0.056	0.975	-3.413	1.042	4,731
Score in hireability	-0.082	1.029	-3.042	1.046	4,736

Panel C: Men

Variable	Mean	Std. Dev.	Min.	Max.	N
Score in coding	-0.026	1.003	-2.981	1.12	21,575
Score in problem solving	-0.024	0.982	-2.62	1.264	21,575
Score in likability	0.083	0.93	-2.738	1.095	21,575
Score in communication	-0.055	0.996	-3.413	1.042	21,575
Score in hireability	0.022	0.991	-3.042	1.046	21,598

Notes: This table shows summary statistics for the rating variable for the sample period before Experiment 1. See Section 2.2 for more information about the sample. The first panel is for all users, while the following two panels split by gender.

Table D4: Problems' and Evaluators' Characteristics

	Problem Difficulty	Variation of the Performance	Harsh Evaluator	
	(1)	(2)	(3)	(4)
Interviewee female	-0.003	0.006	0.005	0.005
	(0.008)	(0.008)	(0.010)	(0.010)
Interviewer Gender	Yes	Yes	Yes	Yes
Date FE	Yes	Yes	Yes	Yes
Problem FE	No	No	No	Yes
N	26,667	26,667	22,582	19,635

Notes: This table shows the coefficient on gender from regressions with dependent variables of problem difficulty, within-problem variation in performance, and whether or not the evaluator was historically harsh as measured by whether the ratings they chose in the past were lower than the median.

Table D5: Balancing Test – Whole Sample

Variables	Control	ITT	Difference	P-value
Interviewee female	0.179	0.187	0.007	0.549
Interviewer female	0.178	0.187	0.008	0.504
Gender interviewer missing	0.049	0.048	-0.001	0.873
Country: USA	0.686	0.684	-0.002	0.923
Interviewee's deg.: computer science	0.645	0.653	0.008	0.635
Interviewer's deg.: computer science	0.643	0.653	0.009	0.578
Interviewer's deg.: postgraduate	0.437	0.431	-0.006	0.700
Interviewee's deg.: postgraduate	0.441	0.430	-0.012	0.498
Interviewee's years of experience	2.943	3.087	0.144	0.224
Interviewer's years of experience	2.958	3.090	0.132	0.271
N	1,587	10,004		
Test of joint significance	*F*-stat: 1.100 (*p*-value: 0.377)			

Notes: This table shows descriptive statistics for the control and ITT samples for Experiment I (see Section 4), along with p-values which test whether differences are significant.

Table D6: Baseline Characteristics of Compliers and Never-Takers

	First Stage	Sample mean	Compliers		Never-takers
	(1)	(2)	(3)	(4)	(5)
			Treated	Untreated	
Interviewee female	0.678***	0.186	0.177	0.166	0.212
	(0.015)		(0.007)	(0.016)	(0.008)
Country: USA	0.718***	0.684	0.681	0.684	0.693
	(0.010)		(0.008)	(0.021)	(0.010)
Interviewee's deg.: computer science	0.709***	0.652	0.660	0.649	0.663
	(0.011)		(0.008)	(0.021)	(0.009)
Interviewee's deg.: postgraduate	0.726***	0.431	0.434	0.450	0.424
	(0.011)		(0.008)	(0.021)	(0.009)
Interviewee's years of experience	0.736***	3.067	3.061	2.859	3.225
	(0.021)		(0.045)	(0.159)	(0.062)
Interviewee Preparation Level (self-declared on 1-5 scale)	0.621***	2.880	2.928	2.768	2.816
	(0.049)		(0.013)	(0.034)	(0.017)

Notes: Column 1 corresponds to the first stage regression for each specific group. Column 2 is the frequency of the group in the estimation sample. Columns 4 and 5 correspond to the estimation of the characteristic in the complier sample, following Abadie (2003) and corresponds to a 2sls regression where the dependent variable corresponds to the endogenous variable multiplied by the indicator of the group.
* p<0.10, ** p<0.05, *** p<0.01

Table D7: Robustness Checks for Experiment I

	Coding	**Problem solving**	**Likeability**	**Communication**	**Hireability**
Panel A: Baseline					
Treatment	0.166***	0.222***	0.099**	0.197***	0.178***
S.E	0.032	0.032	0.039	0.044	0.033
Treatment*Woman	-0.099	-0.056	-0.074	0.006	-0.045
S.E	0.066	0.061	0.084	0.069	0.076
N	11029	11029	11029	11029	11049
Panel B: with Month FE					
Treatment	0.140***	0.212***	0.079**	0.161***	0.150***
S.E	0.029	0.029	0.036	0.042	0.030
Treatment*Woman	-0.109*	-0.067	-0.066	0.013	-0.044
S.E	0.064	0.059	0.082	0.067	0.074
N	11029	11029	11029	11029	11049
Panel C: with Controls					
Treatment	0.168***	0.226***	0.104***	0.199***	0.180***
S.E	0.032	0.032	0.038	0.044	0.033
Treatment*Woman	-0.093	-0.061	-0.074	0.003	-0.044
S.E	0.066	0.060	0.084	0.070	0.076
N	11029	11029	11029	11029	11049
Panel D: no Date FE					
Treatment	0.160***	0.221***	0.100***	0.167***	0.149***
S.E	0.028	0.028	0.033	0.041	0.029
Treatment*Woman	-0.106	-0.066	-0.067	0.014	-0.044
S.E	0.064	0.059	0.082	0.067	0.074
N	11029	11029	11029	11029	11049
Panel E: Including pre-treatment period					
Treatment	0.146***	0.213***	0.082**	0.197***	0.162***
S.E	0.031	0.031	0.034	0.040	0.028
Treatment*Woman	0.011	-0.009	0.025	0.007	0.041*
S.E	0.023	0.024	0.023	0.021	0.024
N	54077	54077	54077	54077	51533
Panel F: Controlling for Propensity Score Matching					
Treatment	0.165***	0.221***	0.099**	0.195***	0.177***
S.E	0.032	0.033	0.039	0.044	0.033
Treatment*Woman	-0.099	-0.055	-0.073	0.008	-0.045
S.E	0.066	0.061	0.084	0.068	0.076
N	11029	11029	11029	11029	11049
Panel G: with Individual FE					
Treatment	-0.005	0.082**	0.028	0.079*	0.060
S.E	0.036	0.033	0.044	0.047	0.037
Treatment*Woman	-0.031	-0.026	-0.169*	0.023	-0.036
S.E	0.092	0.090	0.097	0.111	0.093
N	9797	9797	9797	9797	9816

Notes: This table shows results a series of robustness checks. Panel A presents the results of the baseline ITT specification (Treatment) and the interaction with a categorical variable equal to one when the interviewee is a woman. In Panel B we add month-of-interview fixed effects, and date-of-interview fixed effects in Panel C. In Panel D, we control for socio-demographic characteristics. In Panel E we expand our sample to include pre-treatment introduction interviews with month-of-interview fixed effects. In Panel F, we control for propensity score matching. In Panel G, we control for interviewee fixed effects. Standard errors are clustered at the date level.

Appendix E Experiment II: Additional Results

E.1 Experimental Design

Recruitment Our subject population is comprised of recent graduates or students currently enrolled in computer science programs. We recruited evaluators through universities' undergraduate and graduate programs. Our recruitment email disclosed that we were studying how evaluators judge the performance of software developers, but did not mention gender.

Randomization We used a within-subject design in which each evaluator is assigned 4 coding problems. Two are blind, and two of which are non-blind. Within each treatment arm, evaluators were presented with a code block written by a man and another by a woman, the order of which is randomized. We also randomized the order of treatment: For half of evaluators, evaluation is blind, then non-blind; For the other half, evaluation was non-blind, then blind.

Stratification We constructed the pool of code blocks to be randomly assigned to participants as follows. We stratified the experimental sample on gender, race and performance (i.e dummy for passing all unit tests). This was carried for each coding question and coding language pair. More precisely, for each coding question-language pair (e.g., list sorting in Python) we randomly selected a *single* code from each gender, race, performance cell. This procedure produced a pool of 456 code blocks for the experiment. This stratification procedure means that for each treatment arm and gender pair (e.g. Non-Blind male) all participants have probability $\frac{1}{4}$ of being assigned a script from each race, performance cell. Finally, each selected code block had a blind and a non-blind version. We ensured that if a participant saw a code block in the blind arm they could not see it in the non-blind arm, and vice versa.

Testing the salience of treatment In the piloting phase of the experiment, we asked a random sample of online participants (“evaluators”) on Prolific to predict the gender of a participant (“worker”) after evaluating a task they completed, mimicking the layout of the first name and avatar of our main experiment. While some “evaluators” did not pay attention to the gender of the "workers", neither the evaluators' characteristics nor the workers' characteristics (including gender, race, and how racially distinctive the first name) are predictive of the accuracy of the gender prediction. Additionally, we tested whether an AI tool (Chat GPT) was able to predict the gender of the coder of a code when the first name is not displayed, and it was not able to form that prediction.

Measure of Priors To measure participants' priors, we exposed them to three different vignettes before they perform their evaluation tasks. We asked them to predict the performance of three different hypothetical coders. We cross-randomized the first name (alternating gender) and the skill level for each vignette. The vignetted are constructed as follows:

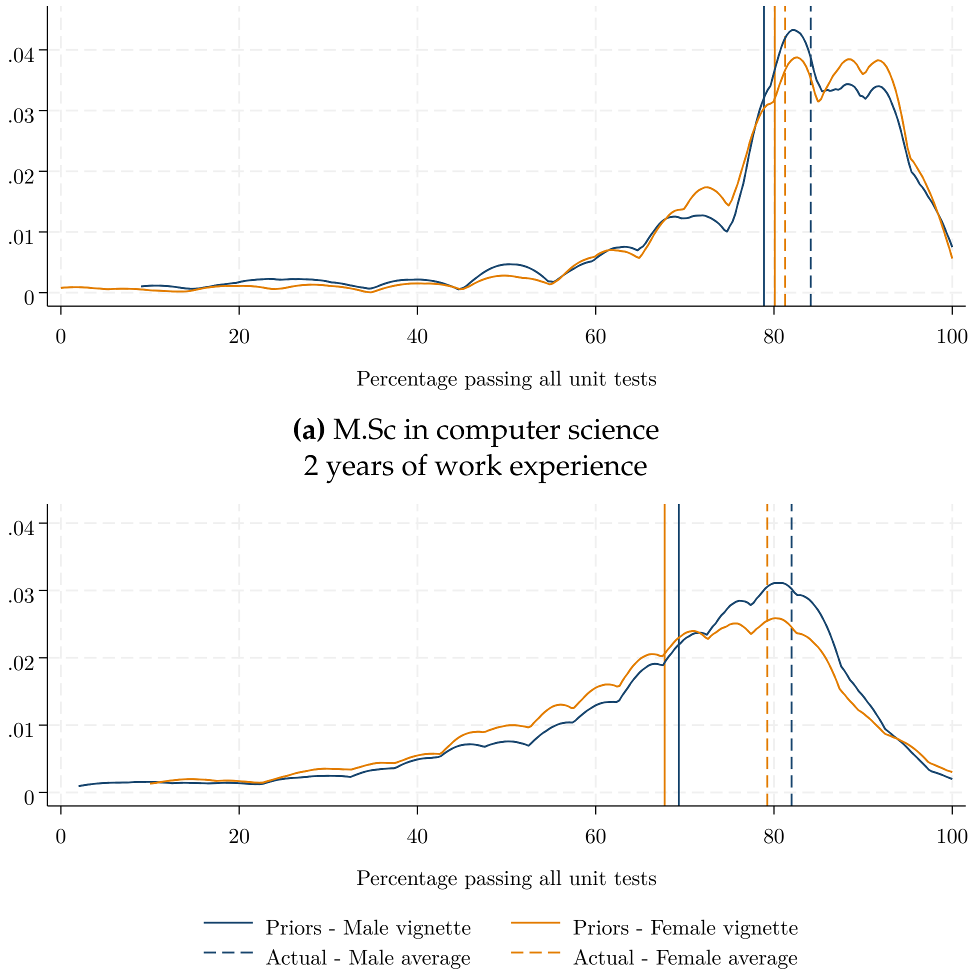
> *82% of the codes you will potentially see resulted in a perfect score and passed all the unit tests. We ask your opinion about the potential performance of different hypothetical coders. If your guess is within 5% of the truth, we will send you an additional reward!*
>
> *"[First Name] holds [Skills]. According to you, what is the percent chance that [First Name]'s code passed all the unit tests?"*

The names and skills shown in the vignettes are as follows.

Skills	First names
a M.Sc in computer science and has 2 years of work experience	Katie/Tom
a Ph.D. in mathematics and has no industry experience	Alexa/Mickael
a B.Sc. degree in computer science	Corinne/Matt

Our results regarding prior beliefs using the resulting data are discussed briefly in Section 5. The accompanying figures follow below.

Figure E1: Respondents' Priors Beliefs About Performance by Gender

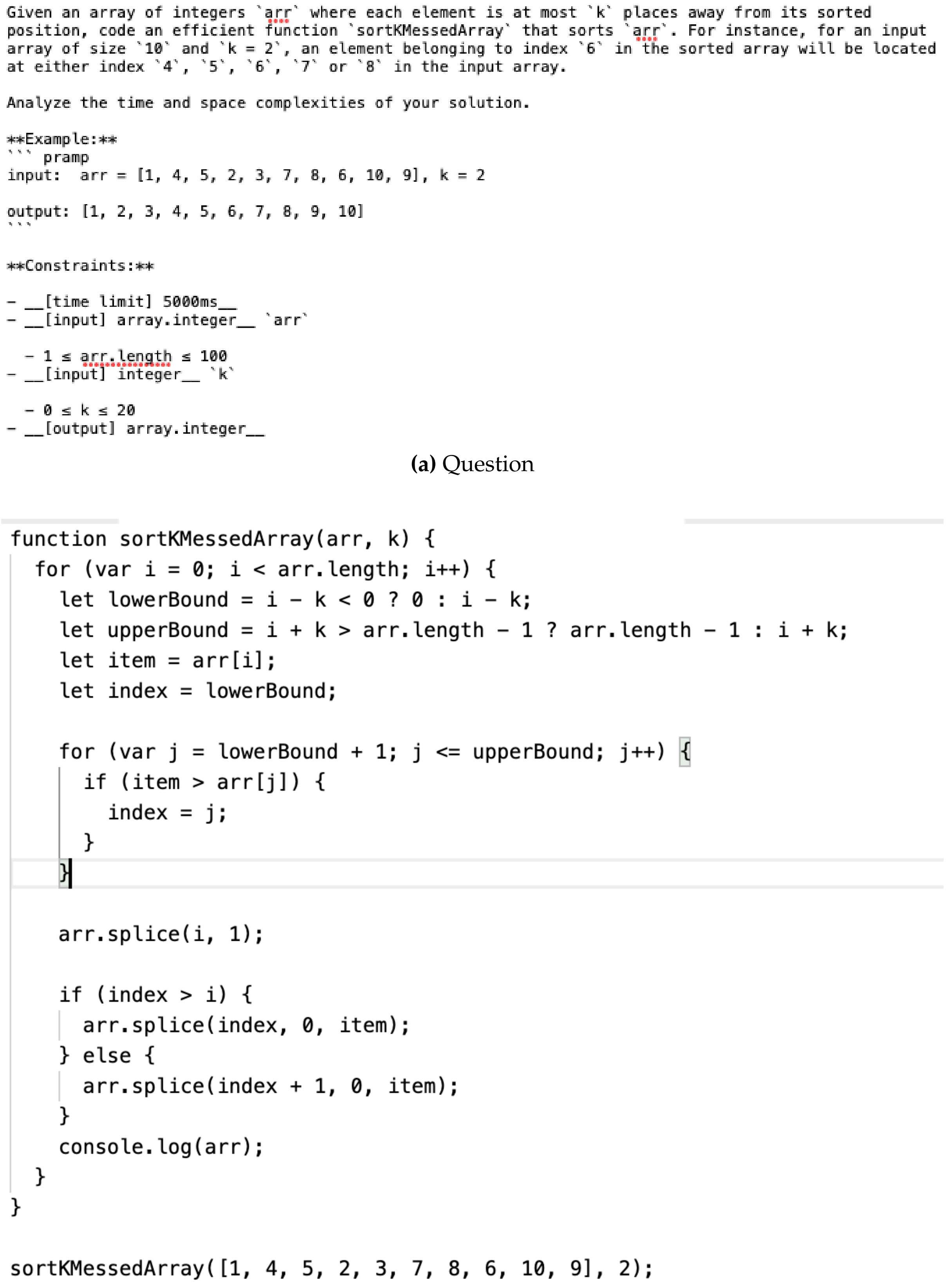


(a) M.Sc in computer science
2 years of work experience

(b) B.Sc. degree in computer science

Notes: This figure shows the distributions of respondents' prior beliefs by gender and skill level of the vignette. The continuous lines represent the mean prior for each gender. The dash lines represent the actual performance for each gender calculated from the sample of codes from the experimental sample. In the overall sample of codes, 82 percent of users pass all unit tests.

Figure E2: Question and Answer for Example Problem — K-Messed Array Sort



~~~
Given an array of integers `arr` where each element is at most `k` places away from its sorted
position, code an efficient function `sortKMessedArray` that sorts `arr`. For instance, for an input
array of size `10` and `k = 2`, an element belonging to index `6` in the sorted array will be located
at either index `4`, `5`, `6`, `7` or `8` in the input array.

Analyze the time and space complexities of your solution.

**Example:**
``` pramp
input:  arr = [1, 4, 5, 2, 3, 7, 8, 6, 10, 9], k = 2

output: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
~~~

Constraints:

- __[time limit] 5000ms__
- __[input] array.integer__ `arr`

  - 1 ≤ arr.length ≤ 100
- __[input] integer__ `k`

  - 0 ≤ k ≤ 20
- __[output] array.integer__
```

(a) Question

```
function sortKMessedArray(arr, k) {
  for (var i = 0; i < arr.length; i++) {
    let lowerBound = i - k < 0 ? 0 : i - k;
    let upperBound = i + k > arr.length - 1 ? arr.length - 1 : i + k;
    let item = arr[i];
    let index = lowerBound;

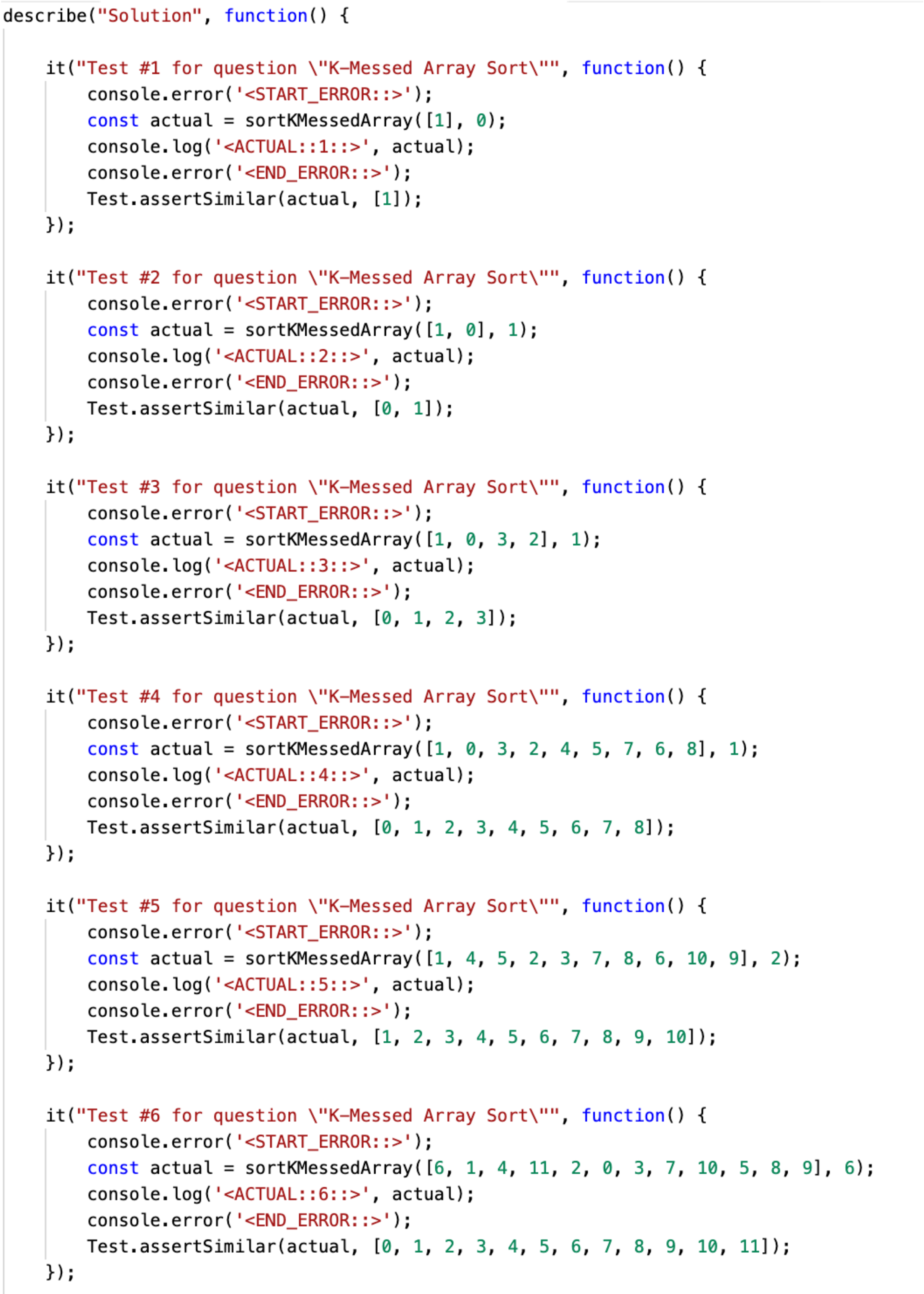
    for (var j = lowerBound + 1; j <= upperBound; j++) {
      if (item > arr[j]) {
        index = j;
      }
    }

    arr.splice(i, 1);

    if (index > i) {
      arr.splice(index, 0, item);
    } else {
      arr.splice(index + 1, 0, item);
    }
    console.log(arr);
  }
}

sortKMessedArray([1, 4, 5, 2, 3, 7, 8, 6, 10, 9], 2);
```

(b) Answer

Notes: This figure presents an example code block that was used in Experiment II. Panel A displays the question, and Panel B the written code block.
```

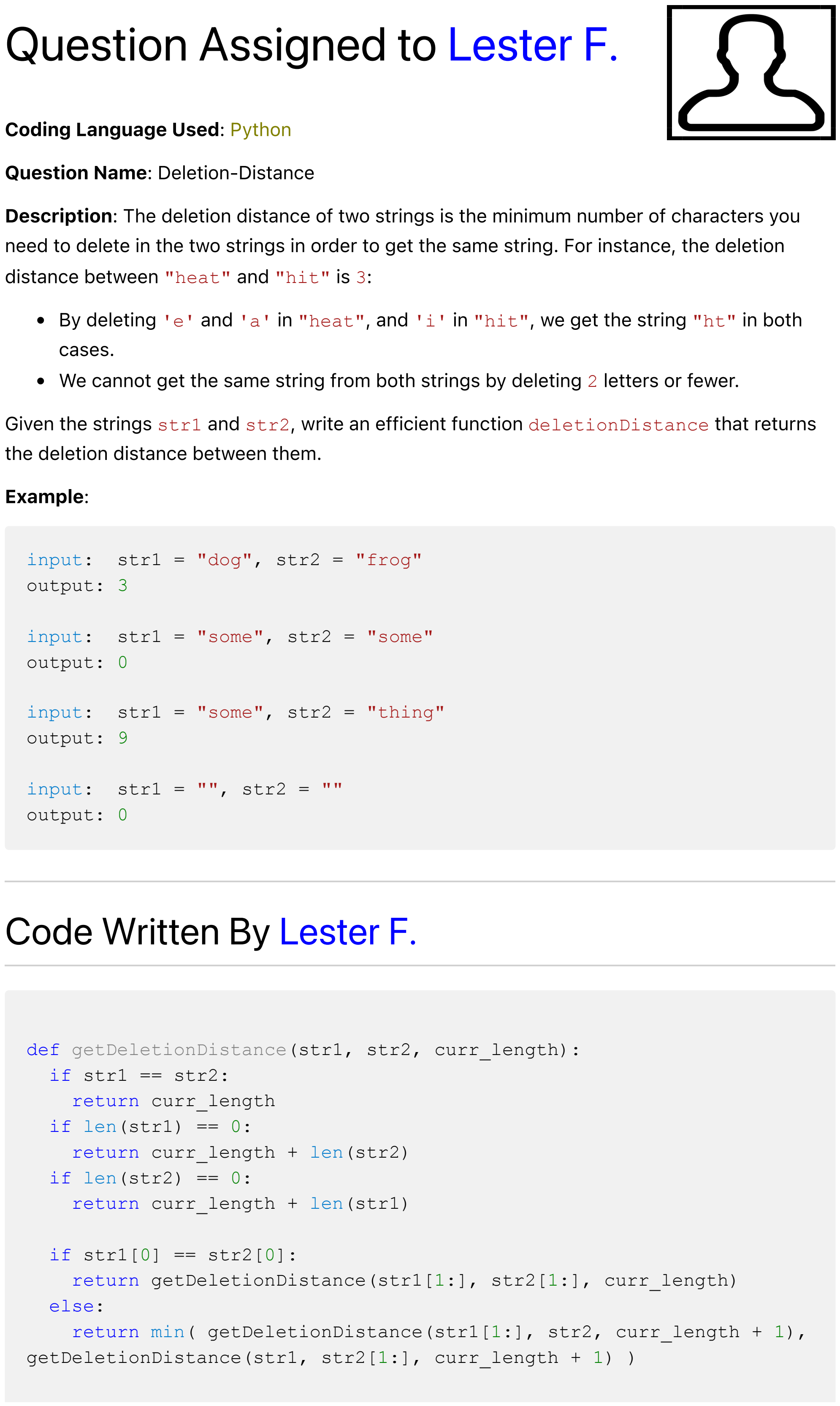
**Figure E3:** Tests for Example Problem — K-Messed Array Sort



```
describe("Solution", function() {

    it("Test #1 for question \"K-Messed Array Sort\"", function() {
        console.error('<START_ERROR::>');
        const actual = sortKMessedArray([1], 0);
        console.log('<ACTUAL::1::>', actual);
        console.error('<END_ERROR::>');
        Test.assertSimilar(actual, [1]);
    });

    it("Test #2 for question \"K-Messed Array Sort\"", function() {
        console.error('<START_ERROR::>');
        const actual = sortKMessedArray([1, 0], 1);
        console.log('<ACTUAL::2::>', actual);
        console.error('<END_ERROR::>');
        Test.assertSimilar(actual, [0, 1]);
    });

    it("Test #3 for question \"K-Messed Array Sort\"", function() {
        console.error('<START_ERROR::>');
        const actual = sortKMessedArray([1, 0, 3, 2], 1);
        console.log('<ACTUAL::3::>', actual);
        console.error('<END_ERROR::>');
        Test.assertSimilar(actual, [0, 1, 2, 3]);
    });

    it("Test #4 for question \"K-Messed Array Sort\"", function() {
        console.error('<START_ERROR::>');
        const actual = sortKMessedArray([1, 0, 3, 2, 4, 5, 7, 6, 8], 1);
        console.log('<ACTUAL::4::>', actual);
        console.error('<END_ERROR::>');
        Test.assertSimilar(actual, [0, 1, 2, 3, 4, 5, 6, 7, 8]);
    });

    it("Test #5 for question \"K-Messed Array Sort\"", function() {
        console.error('<START_ERROR::>');
        const actual = sortKMessedArray([1, 4, 5, 2, 3, 7, 8, 6, 10, 9], 2);
        console.log('<ACTUAL::5::>', actual);
        console.error('<END_ERROR::>');
        Test.assertSimilar(actual, [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]);
    });

    it("Test #6 for question \"K-Messed Array Sort\"", function() {
        console.error('<START_ERROR::>');
        const actual = sortKMessedArray([6, 1, 4, 11, 2, 0, 3, 7, 10, 5, 8, 9], 6);
        console.log('<ACTUAL::6::>', actual);
        console.error('<END_ERROR::>');
        Test.assertSimilar(actual, [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]);
    });
```

Notes: This figure presents the unit tests for the example code block in Figure E2, which was used in Experiment II. Figure E2 shows the question and answer.

Figure E4: Example Code Interface for Experiment II (Non-Blind Male)



Question Assigned to Lester F.

Coding Language Used: Python

Question Name: Deletion-Distance

Description: The deletion distance of two strings is the minimum number of characters you need to delete in the two strings in order to get the same string. For instance, the deletion distance between `"heat"` and `"hit"` is `3`:

- By deleting `'e'` and `'a'` in `"heat"`, and `'i'` in `"hit"`, we get the string `"ht"` in both cases.
- We cannot get the same string from both strings by deleting `2` letters or fewer.

Given the strings `str1` and `str2`, write an efficient function `deletionDistance` that returns the deletion distance between them.

Example:

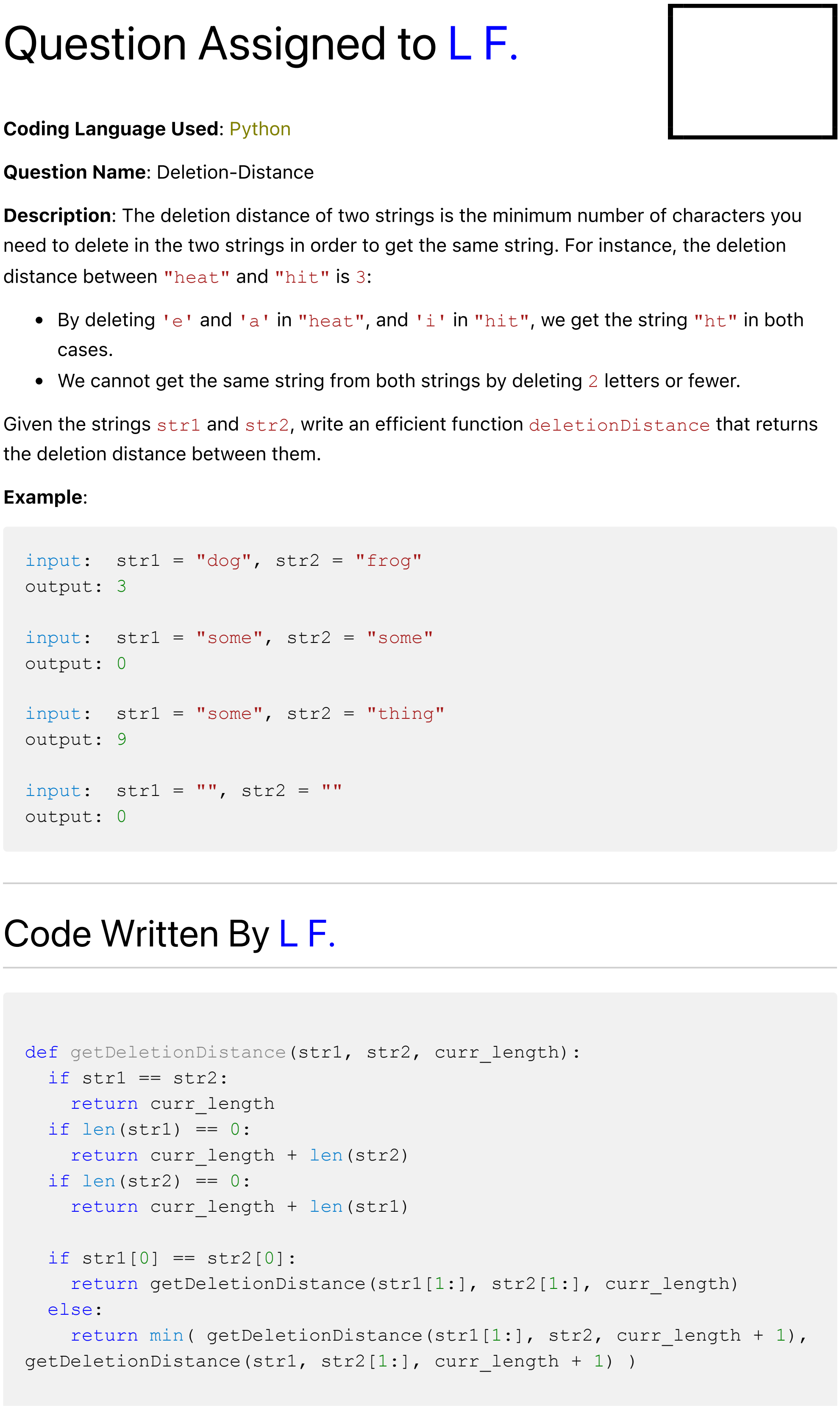
```
input:  str1 = "dog", str2 = "frog"
output: 3

input:  str1 = "some", str2 = "some"
output: 0

input:  str1 = "some", str2 = "thing"
output: 9

input:  str1 = "", str2 = ""
output: 0
```

Code Written By Lester F.

```
def getDeletionDistance(str1, str2, curr_length):
  if str1 == str2:
    return curr_length
  if len(str1) == 0:
    return curr_length + len(str2)
  if len(str2) == 0:
    return curr_length + len(str1)

  if str1[0] == str2[0]:
    return getDeletionDistance(str1[1:], str2[1:], curr_length)
  else:
    return min( getDeletionDistance(str1[1:], str2, curr_length + 1),
getDeletionDistance(str1, str2[1:], curr_length + 1) )
```

Notes: This figure presents an example code block written by a male coder as it is presented in the non-blind condition.

Figure E5: Example Code Interface for Experiment II (Blind Male)



Question Assigned to L F.

Coding Language Used: Python

Question Name: Deletion-Distance

Description: The deletion distance of two strings is the minimum number of characters you need to delete in the two strings in order to get the same string. For instance, the deletion distance between `"heat"` and `"hit"` is `3`:

- By deleting `'e'` and `'a'` in `"heat"`, and `'i'` in `"hit"`, we get the string `"ht"` in both cases.
- We cannot get the same string from both strings by deleting `2` letters or fewer.

Given the strings `str1` and `str2`, write an efficient function `deletionDistance` that returns the deletion distance between them.

Example:

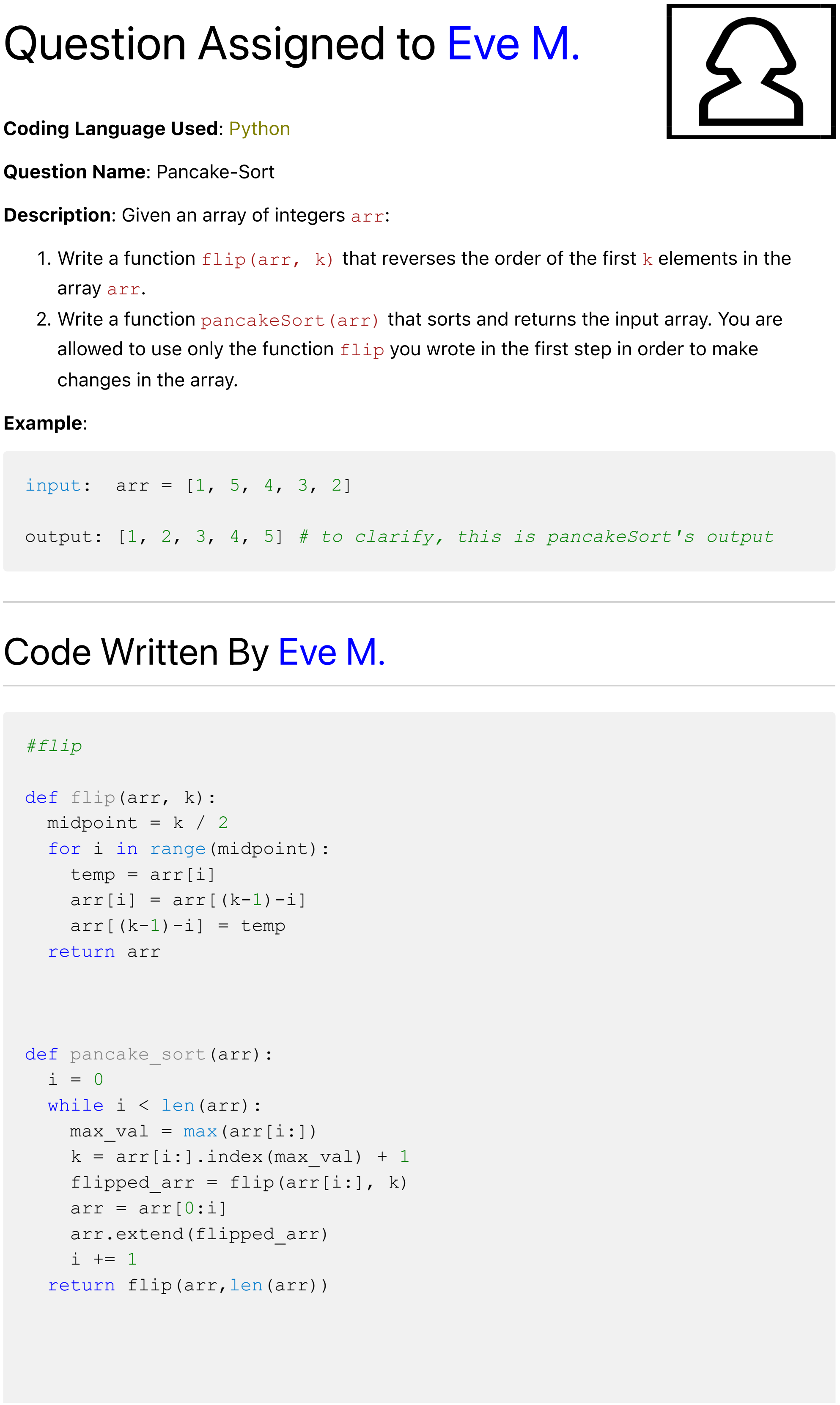
```
input:  str1 = "dog", str2 = "frog"
output: 3

input:  str1 = "some", str2 = "some"
output: 0

input:  str1 = "some", str2 = "thing"
output: 9

input:  str1 = "", str2 = ""
output: 0
```

Code Written By L F.

```
def getDeletionDistance(str1, str2, curr_length):
  if str1 == str2:
    return curr_length
  if len(str1) == 0:
    return curr_length + len(str2)
  if len(str2) == 0:
    return curr_length + len(str1)

  if str1[0] == str2[0]:
    return getDeletionDistance(str1[1:], str2[1:], curr_length)
  else:
    return min( getDeletionDistance(str1[1:], str2, curr_length + 1),
getDeletionDistance(str1, str2[1:], curr_length + 1) )
```

Notes: This figure presents an example code block written by a male coder as it is presented in the blind condition.

Figure E6: Example Code Interface for Experiment II (Non-Blind Female)



Question Assigned to Eve M.

Coding Language Used: Python

Question Name: Pancake-Sort

Description: Given an array of integers `arr`:

1. Write a function `flip(arr, k)` that reverses the order of the first `k` elements in the array `arr`.
2. Write a function `pancakeSort(arr)` that sorts and returns the input array. You are allowed to use only the function `flip` you wrote in the first step in order to make changes in the array.

Example:

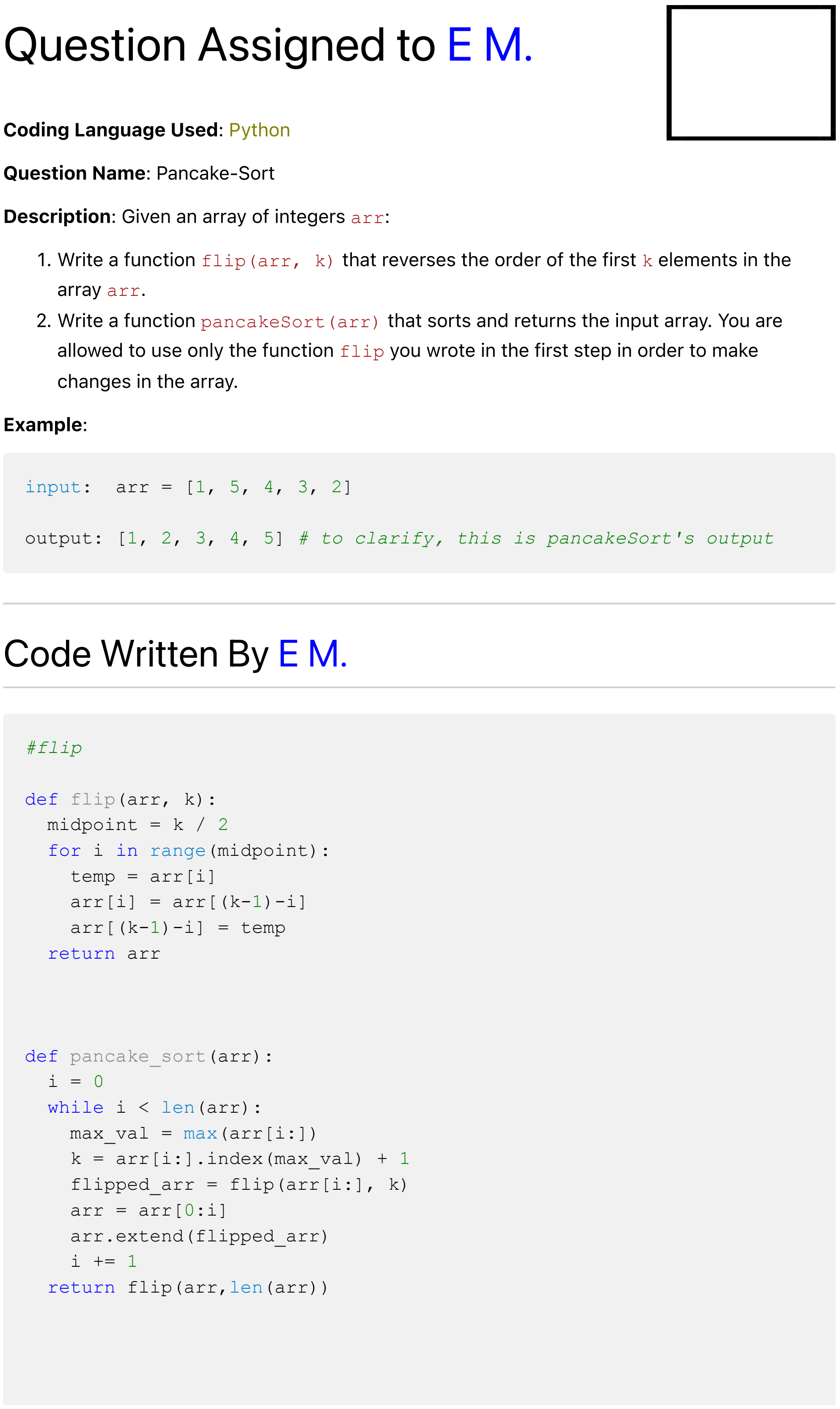
```
input:  arr = [1, 5, 4, 3, 2]

output: [1, 2, 3, 4, 5] # to clarify, this is pancakeSort's output
```

Code Written By Eve M.

```
#flip

def flip(arr, k):
  midpoint = k / 2
  for i in range(midpoint):
    temp = arr[i]
    arr[i] = arr[(k-1)-i]
    arr[(k-1)-i] = temp
  return arr



def pancake_sort(arr):
  i = 0
  while i < len(arr):
    max_val = max(arr[i:])
    k = arr[i:].index(max_val) + 1
    flipped_arr = flip(arr[i:], k)
    arr = arr[0:i]
    arr.extend(flipped_arr)
    i += 1
  return flip(arr,len(arr))
```

Notes: This figure presents an example code block written by a female coder as it is presented in the non-blind condition.

Figure E7: Example Code Interface for Experiment II (Blind Female)



Question Assigned to E M.

Coding Language Used: Python

Question Name: Pancake-Sort

Description: Given an array of integers `arr`:

1. Write a function `flip(arr, k)` that reverses the order of the first `k` elements in the array `arr`.
2. Write a function `pancakeSort(arr)` that sorts and returns the input array. You are allowed to use only the function `flip` you wrote in the first step in order to make changes in the array.

Example:

```
input:  arr = [1, 5, 4, 3, 2]

output: [1, 2, 3, 4, 5] # to clarify, this is pancakeSort's output
```

Code Written By E M.

```
#flip

def flip(arr, k):
  midpoint = k / 2
  for i in range(midpoint):
    temp = arr[i]
    arr[i] = arr[(k-1)-i]
    arr[(k-1)-i] = temp
  return arr



def pancake_sort(arr):
  i = 0
  while i < len(arr):
    max_val = max(arr[i:])
    k = arr[i:].index(max_val) + 1
    flipped_arr = flip(arr[i:], k)
    arr = arr[0:i]
    arr.extend(flipped_arr)
    i += 1
  return flip(arr,len(arr))
```

Notes: This figure presents an example code block written by a female coder as it is presented in the blind condition.

E.2 Descriptive Statistics: Sample of Code Blocks

Table E1: Descriptive Statistics — Follow-up Experiment— January 2018-May 2022

	Raw Data	Clean Data	Experimental Data
Number of session-participant pairs	482,390	178,717	38,322
Number of unique participants	97,614	30,633	10,380
Number of unique problems	39	39	38
Share non-missing unit score	0.42	0.56	1.00
Share of Python scripts	0.30	0.37	0.43
Share of Java scripts	0.35	0.35	0.45
Share of C++ scripts	0.17	0.09	0.12
Share Female			0.18
Share Nonwhite			0.62
Share Full Score			0.82

Notes: This table presents basic characteristics for the code blocks in the sample used in Experiment II (see Sections 2.2 and 5). The raw data are as received from platform. The clean data correspond to scripts with non-missing interviewer rating, feedback and question type. The final sample corresponds to scripts with identified gender and race, and non-missing unit-test score. Participants restricted for those in the United States.

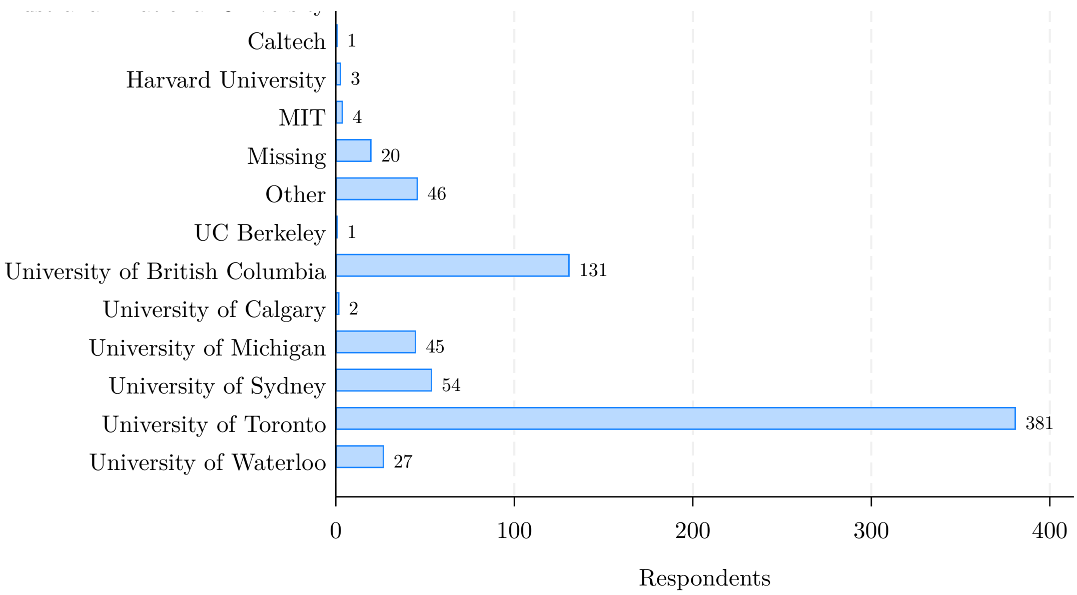
Table E2: Descriptive Statistics — Coding Blocks

	Mean	**Std. Dev.**
Female Users	0.500	0.501
Objective score	0.744	0.314
Passed all unit tests	0.500	0.501
Subjective Rating	3.379	0.713
Num. lines	47.14	13.70
C++	0.088	0.283
Java	0.544	0.499
Python	0.368	0.483
Master degree or more	0.520	0.500
Major in CS	0.827	0.379
Years of FT work experience	3.055	3.143
N	456	

Notes: This table provides summary statistics for the final set of code blocks on which Exp II was conducted. These blocks were obtained via the stratification process explained in 5.2.1.

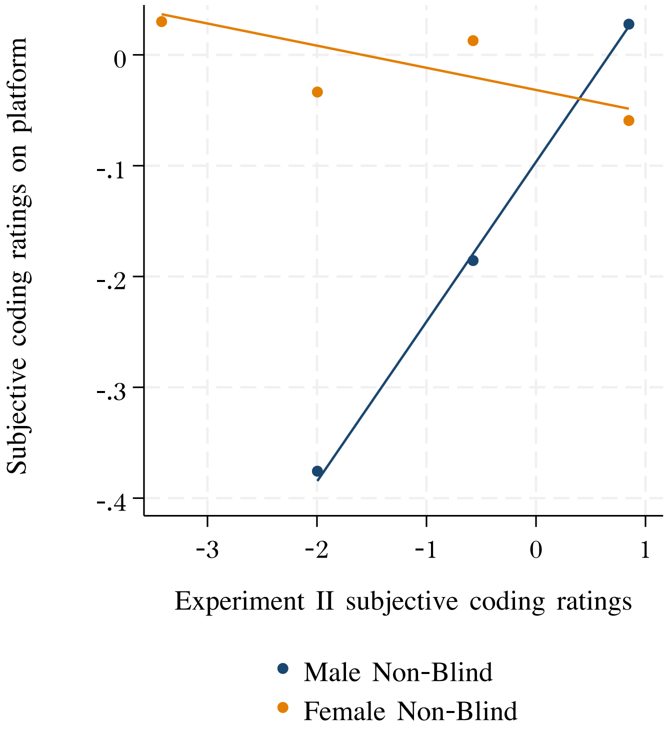
E.3 Descriptive Statistics: Evaluators

Figure E8: Respondents by Institutions



Notes: This figure shows the locations of the evaluators in Experiment II .

Figure E9: Comparability of Coding Ratings between Experiments I and II



Notes: This figure shows the average relationship between ratings in Experiment II and ratings on the platform for the same code block, for men and women.

Table E3: Descriptive Statistics — Participants

	Mean	Std. Dev.	N
Gender			
Female	0.278	0.448	565
Male	0.658	0.475	565
Non-binary / third gender	0.03	0.171	565
Prefer not to say	0.03	0.171	565
Prefer to self-describe	0.004	0.059	565
Recoded race			
White	0.164	0.371	603
South Asian	0.216	0.412	603
Chinese	0.526	0.5	603
Black	0.005	0.07	603
Latinx	0.018	0.134	603
Other	0.071	0.258	603
Unknown	0.158	0.365	716
Current situation			
Currently a student	0.828	0.377	705
Completed at least one degree	0.166	0.372	705
Didn't complete a degree	0.006	0.075	705
Highest degree completed			
Associates or technical degree	0.004	0.065	704
Bachelor's degree	0.736	0.441	704
High School diploma or GED	0.021	0.145	704
MA, MSc or MEng	0.151	0.358	704
PhD	0.047	0.212	704
Some college, but no degree	0.034	0.182	704
Prefer not to say	0.007	0.084	704
Experience with Python			
Basic	0.221	0.415	707
Intermediate	0.448	0.498	707
Advanced	0.331	0.471	707
Experience with Java			
Basic	0.536	0.499	676
Intermediate	0.361	0.481	676
Advanced	0.104	0.305	676
Experience with C++			
Basic	0.643	0.479	673
Intermediate	0.272	0.445	673
Advanced	0.085	0.279	673
Preferred language			
C++	0.089	0.285	716
Java	0.141	0.348	716
Python	0.77	0.421	716

Notes: This table shows descriptive statistics participants in Experiment II (see Section 5).

Table E4: Treatment-Control Balance — Whole Sample

	Non-blind to Blind (1)	Blind to Non-blind (2)	Difference (3)	p-value of diff. (4)
Female	0.278	0.278	-0.000	0.992
Male	0.662	0.655	-0.008	0.850
White respondent	0.158	0.170	0.011	0.714
South Asian	0.205	0.227	0.022	0.510
Chinese	0.554	0.497	-0.057	0.161
Black	0.007	0.003	-0.003	0.569
Latinx	0.020	0.017	-0.003	0.776
Other	0.056	0.087	0.030	0.149
Unknown	0.146	0.169	0.024	0.387
Currently a student	0.827	0.830	0.003	0.927
Completed at least one degree	0.164	0.168	0.003	0.908
Didn't complete a degree	0.008	0.003	-0.006	0.303
Bachelor's degree	0.708	0.764	0.056	0.090
MA, MSc or MEng	0.170	0.131	-0.039	0.144
PhD	0.059	0.034	-0.025	0.115
C++	0.082	0.097	0.015	0.479
Java	0.161	0.122	-0.039	0.137
Python	0.758	0.781	0.024	0.455
Observations	1,420	1,444		

Notes: This table presents balancing checks for the whole sample. The p-values are obtained from a linear regression on each covariate with strata fixed effect. Standard errors are clustered at the evaluator level.

Table E5: Treatment-Control Balance — High Quality sample

	Non-blind to Blind (1)	Blind to Non-blind (2)	Difference (3)	p-value of diff. (4)
Female	0.260	0.260	0.000	0.994
Male	0.683	0.683	-0.000	0.992
White respondent	0.171	0.178	0.008	0.831
South Asian	0.175	0.244	0.069	0.079
Chinese	0.553	0.465	-0.088	0.068
Black	0.005	0.005	0.000	0.990
Latinx	0.028	0.014	-0.014	0.322
Other	0.069	0.094	0.025	0.353
Unknown	0.135	0.141	0.006	0.856
Currently a student	0.841	0.823	-0.018	0.588
Completed at least one degree	0.155	0.177	0.022	0.505
Didn't complete a degree	0.004	0.000	-0.004	0.317
Bachelor's degree	0.705	0.774	0.070	0.075
MA, MSc or MEng	0.179	0.117	-0.063	0.048
PhD	0.052	0.044	-0.007	0.706
C++	0.088	0.109	0.021	0.421
Java	0.167	0.137	-0.030	0.346
Python	0.745	0.754	0.009	0.821
Observations	1,004	992		

Notes: This table presents balancing checks for the quality sample, namely restricting to participants who passed the first attention check question, and excluding respondents whose survey completion time falls within the bottom 10th (less than 8 minutes) and top 90th percentiles (4 hours or more). The p-values are obtained from a linear regression on each covariate with strata fixed effect. Standard errors are clustered at the evaluator level.

Table E6: Effect Of Blinding On Gender Gaps — Quality Sample

	Subjective coding rating		Unit test prediction		Interview prediction	
Female code	-0.030	-0.022	0.072	0.081	0.022	0.024
	(0.066)	(0.065)	(0.207)	(0.207)	(0.060)	(0.059)
Non-blind code	-0.120	-0.116	-0.364	-0.357	-0.112	-0.073
	(0.066)	(0.067)	(0.219)	(0.220)	(0.062)	(0.062)
Non-blind code×Female code	0.105	0.107	0.290	0.335	0.072	0.073
	(0.094)	(0.095)	(0.299)	(0.299)	(0.086)	(0.086)
Treatment order control	Yes	Yes	Yes	Yes	Yes	Yes
Order of scripts FE	Yes	Yes	Yes	Yes	Yes	Yes
Problem FE	Yes	Yes	Yes	Yes	Yes	Yes
Evaluator FE	No	Yes	No	Yes	No	Yes
Observations	1,852	1,835	1,852	1,835	1,946	1,946

Notes: This table provides results from Experiment II (see Section 5), testing the pre-registered hypothesis that revealing gender introduces a gender gap that penalizes women for the quality sample, for the quality sample, namely restricting to participants who passed the first attention check question, and excluding respondents whose survey completion time falls within the bottom 10th (less than 8 minutes) and top 90th percentiles (4 hours or more). The regression specification is as described in Equation (3). The dependent variables are the (standardized) subjective coding ratings (columns 1-2), participants' prediction of the unit tests passed by the code script (columns 3-4) and their prediction of the coder's probability of passing the interview (columns 5-6). The even columns include evaluator fixed effects. Standard errors are clustered at the evaluator level.

Table E7: Alternative Quality Measures

	Mean	**Std. Dev.**	**N**
Passed 1st attention check	0.852	0.355	716
Passed 2nd attention check	0.327	0.469	716
Self-reported ability: intermediate/advanced	0.862	0.345	716
Evaluated all code blocks	0.793	0.405	716
Graduate student	0.194	0.396	716
Survey time: less than 8 minutes	0.101	0.301	716
Survey time: 4 hours or more	0.099	0.299	716

Notes: This table provides alternative quality measures for our responses to Experiment II. The first two rows show the shares of individuals who passed our easier and harder attention checks (see the survey in Appendix E). The third row shows the fraction of evaluators whose ability with the chosen coding language is intermediate or advanced. The fourth row shows the share of respondents who completed all evaluations assigned to them. Row 5 is the share who are graduate students. The final two rows show the shares of respondents who spent an unusually large or small amount of time on the survey.

Table E8: Blinding Experiment — Main Results (Reweighted)

	Coding subjective rating		Unit tests prediction		Interview prediction	
	(1)	(2)	(3)	(4)	(5)	(6)
Female code	0.090	0.089	0.311	0.312	0.088	0.087
	(0.074)	(0.070)	(0.217)	(0.213)	(0.065)	(0.061)
Non-blind code	-0.040	-0.034	-0.338	-0.307	-0.121*	-0.044
	(0.071)	(0.070)	(0.241)	(0.242)	(0.066)	(0.066)
Non-blind code × Female code	0.006	-0.001	0.237	0.211	0.000	-0.004
	(0.100)	(0.099)	(0.319)	(0.318)	(0.089)	(0.088)
Treatment order control	Yes	Yes	Yes	Yes	Yes	Yes
Order of scripts FE	Yes	Yes	Yes	Yes	Yes	Yes
Problem FE	Yes	Yes	Yes	Yes	Yes	Yes
Evaluator FE	No	Yes	No	Yes	No	Yes
Observations	2,314	2,284	2,314	2,284	2,664	2,664

Notes: This table provides results from Experiment II (see Section 5), testing the pre-registered hypothesis that revealing gender introduces a gender gap that penalizes women. The regression specification is as described in Equation (3).The even columns include evaluator fixed effects, while the odd columns do not.Standard errors are clustered at the evaluator level. Results are weighted by gender and education composition of users on the platform. Weights are equal the inverse predicted probability of being in the experiment relative to the Platform.

Appendix F Experiment II: Questionnaire

Informed Consent

Overview. You are being asked to take part in a research study being done by a group of researchers from the University of Michigan and the University of Toronto. This is a survey for academic research in social sciences. Your participation is invaluable for our research. If you choose to participate and to complete the survey, you will be financially compensated with a minimum of $50. As a participant, you will be asked to evaluate pieces of code written by others, and answer a short follow-up questionnaire. We expect that participation will take around 60 minutes. In each part, you will receive clear instructions and will be told how your decisions in that part will influence your earnings in the study. You will also have the opportunity to learn about your performance as evaluator.

Non-Deception Statement. This study does not deceive you by providing misleading or incorrect information. All our communications are truthful, but we may not always reveal all information. Specifically, there are different versions of this study. While you will be fully informed about the version of this study that you have been randomly assigned to, you will not be informed about different versions of this study that other participants are in.

Voluntary Participation, Privacy, and Point of Contact. Your participation is completely voluntary. You can agree to take part and later change your mind. Your decision will not be held against you. Note that the data you provide in this study will be anonymized prior to analysis. Your information will be kept entirely confidential and accessed only by the research team, and only as necessary to conduct the research. In the future, this non-identifiable data may be shared with other researchers or published. All information identifying you as a study participant will be destroyed upon the conclusion of the study. However, the anonymized information you provide may be maintained indefinitely.

The principal investigator of this study is Ashley C. Craig from University of Michigan. If you have any questions, concerns, or complaints, or think this research hurt you, talk to the research team at `ash@ashleycraig.com`. If you have questions about your rights as participants, you can contact the Research Oversight and Compliance Office—Human Research Ethics Program at ethics.review@utoronto.ca or 416-946-3273. You can also contact the University of Michigan IRB (Health Sciences and Behavioral Sciences) at 734-936-0933 or `irbhsbs@umich.edu`, quoting eResearch #HUM00204184.

The research study you are participating in may be reviewed for quality assurance to make sure that the required laws and guidelines are followed. If chosen, (a) representative(s) of the Human Research Ethics Program (HREP) may access study-related data and/or consent materials as part of the review. All information accessed by the HREP will be upheld to the same level of confidentiality that has been stated by the research team. If you would like a summary of the results of this research (once the study has been completed), please email `ash@ashleycraig.com`.

Compensation. You will receive $10 if you complete the survey and an additional $10 for each code segment you evaluate. Additionally, we will ask you to make a series of predictions. You will have the opportunity to gain $2 for each accurate prediction. Your total earnings will be distributed within one week after the completion of the survey. If you are interested, you can receive individualized feedback about the quality of your performance as an evaluator.

Based on their performance, the best ten evaluators win a $500 prize. The three best evaluators will also be invited to the Creative Destruction Lab 2023 Super Session in Toronto, which brings together world-class entrepreneurs, investors and scientists with high-potential startup founders. Organized in June 2023, the CDL Super Session days will give you with meaningful networking opportunities and exposure to key players in the industry. If there are ties in evaluation performance, the recipients of the prize and these invitations will be chosen randomly from among the set of evaluators with equal best accuracy scores. You may print a copy of this information for your records.

> Yes, I would like to voluntarily participate in this experiment. [Signature]
>
> I am interested in receiving individualized feedback on my performance as an evaluator. □ Yes □ No

---

For the purposes of payment and the $500 cash prize, and to be considered for an invitation to the Creative Destruction Lab, please type your email below. We will not use your email for any purposes other than the provision of these rewards.

[Type here]

In what currency would you like to receive your payment?

○ AUD
○ CAD
○ USD

Please make sure you are willing and ready to sit through this study uninterruptedly and undistractedly before starting it. We ask you to please focus on the tasks of this study and thank you for your cooperation.

General Roadmap

This study consists of 4 evaluation tasks, followed by a few questions. The evaluation parts will ask you to give a score from 1 to 4 for scripts, both of which are solutions to a given coding question. The coding question will be outlined before the script.

Attention Checks. Note that this experiment contains attention checks. These questions are there to ensure you are paying attention as you take this survey. The answers to those attention check questions will not be ambiguous, will not be a trick question, and will not be timed. If you answer an attention check incorrectly or not within the provided time, you may be dismissed without pay.

Here is your first attention check. In the space below, please spell the word "human" backwards. Please use all lowercase letters and insert no space between the letters.

[Type here]

---

1. What best describes your present situation regarding your education?

   ○ I am currently a student
   ○ I have completed at least one degree
   ○ I was previously enrolled in a degree program but did not complete it

2. What is your highest level of education (including enrolled)?

□ High School diploma or GED

□ Some college, but no degree

□ Associates or technical degree

□ Bachelor's degree

□ MA, MSc or MEng

□ PhD

□ Prefer not to say

3. What is or are the area(s) of your highest degree? (multiple answers are allowed)

□ Computer Science

□ Computer Engineering

□ Mathematics

□ Information Systems / M.I.S.

□ Statistics

□ Other Exact Sciences Degree (e.g. physics, chemistry, astronomy)

□ Other Technology Related Degree

□ None

□ Other

4. What is the institution where you received or will receive your highest degree?

[Drop down menu]

5. How would you describe your knowledge of these programming languages?

Python ○ Basic ○ Intermediate ○ Advanced

Java ○ Basic ○ Intermediate ○ Advanced

C++ ○ Basic ○ Intermediate ○ Advanced

6. During this study, you will be asked to evaluate a series of human written code blocks. Please select the coding language you are most proficient in.

○ Python

○ C++

○ Java

Before you start, we want to ask you a series of quick questions. The code excerpts were automatically subjected to a series of unit tests. These determined whether the code ran, and produced correct answers in pre-defined test cases.

Overall, 52% of the code blocks you will potentially see resulted in a perfect score and passed all the unit tests. We ask your opinion about the potential performance of different hypothetical coders. If your guess is within 5% of the truth for coders like those described, you will receive an additional reward!

- Katie/Tom holds a M.Sc in computer science and has 2 years of work experience. According to you, what is the percent chance that Katie's code passed all the unit tests?
- Alexa/Michael holds a Ph.D. in mathematics and has no industry experience. According to you, what is the percent chance that Alexa's code passed all the unit tests?
- Corinne/Matt holds a B.Sc. degree in computer science. According to you, what is the percent chance that Matt's code passed all the unit tests?

[***Note***: *Names and characteristics were randomized as described in Section 5.*]

[BEGINNING OF TASK]

We are now going to ask you to evaluate a series of codes. These codes were written by actual software developers. We will provide you with the initial question and their written answers.

For each piece of code, we ask you to give your personal opinion about the quality of code, by providing a rating between 1 (lowest) and 4 (highest). At the end of all code evaluation, we will ask you to explain how you decided on your rating. You will gain a $10 additional bonus for each code you evaluate.

Additionally, we will ask you to make a series of predictions. You will have the opportunity to gain $2 for each accurate prediction.

Code Block 1

1. How would you rate the quality of the code (1 lowest, 4 highest)?

   ○ 1 (lowest)

   ○ 2

   ○ 3

   ○ 4 (highest)

2. Can you let us know why you gave this score to the code ?

[Text Box]

3. A series of unit tests were used to evaluate this code. How many out of 10 unit tests do you think were passed? If your guess is within 5 percentage points of the truth, you will gain $2 and will increase your chances of participating to the Creative Destruction Lab Meeting and winning one of the $500 prizes.

    [Drop Down Menu]

4. How confident are you about this prediction?

    ○ Not confident at all
    ○ Not confident
    ○ Somewhat confident
    ○ Confident
    ○ Very confident

5. Another human evaluator assessed whether this coder passed or failed based on this coding performance and other factors. We ask you to guess whether that evaluator decided that this coder passed or failed. Please note that 85% of all coders pass. If you guess correctly, you will gain $2 USD, and will increase your chances of participating in the Creative Destruction Lab meeting and winning one of the $500 USD prizes. Based on this code that they wrote, do you think the code passed or failed?

    ○ Failed
    ○ Passed

6. How confident are you about this prediction?

    ○ Not confident at all
    ○ Not confident
    ○ Somewhat confident
    ○ Confident
    ○ Very confident

According to you, what is the percent chance that the candidate was later invited for an interview for a role involving coding?

[Cursor Between 0 and 100]

People often consult internet sites to learn about employment opportunities in tech. We want to know which sites you use. We also want to know if you are paying attention, so please select Glassdoor and Crunchbase regardless of which sites you use. When looking for employment opportunities, which is the one website you would visit first? (Please only choose one).

□ LinkedIn

□ Hired

□ Glassdoor

□ Crunchbase

□ ZipRecruiter

□ TripleByte

□ Underdog

□ Angel

[Code Block 2 to 4 — Repeat The Above With Different Details]

Follow-up questions

1. In which country do you currently reside?

   ○ Canada

   ○ USA

   ○ Other: [Type]

2. How do you describe yourself?

   ○ Male

   ○ Female

   ○ Non-Binary / third gender

   ○ Prefer to self-describe: [Type]

   ○ Prefer not to say

3. What is your year of birth?

   [Drop Down Menu]

4. What best describes your employment status of the last three months?

   ○ Working full-time

   ○ Working part-time

   ○ Unemployed and looking for work

   ○ A homemaker or stay-at-home parent

   ○ Student

   ○ Retired

   ○ Other

5. How many year of working experience do you have?

   [Drop Down Menu]

---

1. In the box below, explain how you made your decisions today. Please answer in one or more full sentences.

   [Text Box]

2. If you had to guess, what do you think was this study about? Please answer in one or more full sentences.

   [Text Box]

3. Do you have any comments or feedback related to this study? (optional)

   [Text Box]

4. Was there anything confusing about this study? (optional)

   [Text Box]

Thank you very much for participating in this study!

Your response has been recorded and your total earnings will distributed within one week. If you have any questions or if you experienced any problems, please feel free to reach out to Ashley Craig at [email omitted].

You may now close this window.

[END OF QUESTIONNAIRE]