

# Does Better Information Reduce Gender Discrimination in the Technology Industry?\*

Abdelrahman Amer, Ashley C. Craig and Clémentine Van Effenterre

August 2023

## Abstract

Performance evaluation matters for hiring and promotion decisions. We combine experiments with administrative data to show that the presence of personal interactions affects the display of gender bias in performance evaluations. Leveraging 60,000 mock interviews from an online platform for software engineers, we document that women receive lower ratings for code quality and problem solving than men, even after controlling for an automated measure of performance which is predictive of future labor market outcomes. We analyze two field experiments, which vary the information seen by evaluators, to shed light on what drives these gaps. When interviews are conducted via video chat, our first experiment shows that providing evaluators with additional measures of performance does not reduce the gender gap in evaluations. This motivates a second experiment which removes video interaction, and compares blind to non-blind evaluations. There is no longer a detectable gender gap in either case. These results are hard to reconcile with traditional economic models of discrimination. Instead, the fact that the degree of bias depends on the context in which evaluation occurs is more consistent with a form of implicit bias that arises with personal interaction.

**JEL codes:** C93, D83, J16, J71, M51

**Keywords:** *Discrimination; Gender; Coding; Experiment; Information.*

---

\***Amer:** University of Toronto, 150 Saint George Street, Toronto ON M5S 3G7, Canada (e-mail: abdelrahman.amer@mail.utoronto.ca) **Craig:** Australian National University, Research School of Economics, Room 2094, LF Crisp Building, 25a Kingsley St, Acton ACT 2601 (e-mail: ashley.craig@anu.edu.au) **Van Effenterre:** University of Toronto, 150 Saint George Street, Toronto ON M5S 3G7, Canada (e-mail: c.vaneffenterre@utoronto.ca). This paper greatly benefited from discussions and helpful comments from Iris Bohnet, Katherine Coffman, Nicole Fortin, Dylan Glover, Maria Guadalupe, Sara Heller, Peter Hull, Kory Kroft, Corinne Low, Michelle Lowry, 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, and LAGV. We are grateful to the Pivotal fund, the NBER Digitization Program and the Russel Sage Foundation for continuous 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.

# 1 Introduction

Economists and policymakers have long dedicated attention to discrimination as a barrier to underrepresented groups in high-paying occupations (Bertrand and Duflo, 2017). Imperfect information has been seen as one way to rationalize such differential treatment (Phelps, 1972; Arrow, 1973; Coate and Loury, 1993; Craig and Jr, 2019). In many high-skilled industries, the hiring process is comprised of multiple stages: Recruiters learn about a candidate through resumes, referrals, test results, interviews, and simulations which ask the candidate to perform a task in a realistic work context. At every point of this process, evaluators and decision-makers have imperfect information about the skills and future performance of the applicants, and they have to rely on their own judgment to assess these applicants.

Focusing on evaluation of coding performance, a common stage during the recruitment process of computer programmers, we conduct two field experiments to show that the presence of personal interactions affects the display of gender bias in evaluations. In each experiment, we vary the information available to evaluators, while holding constant a realistic task (coding), a rating scale, and an objective measure of coding performance (a series of unit tests a code must pass). Our results show that evaluators display gender bias during face-to-face interviews. But once this interaction is removed, we find no evidence that women are treated differently. The discrimination we document in evaluations could have important consequences for labor market outcomes even if hiring managers do not themselves discriminate. Rather than showing up as direct discrimination, such bias would show up as “systemic” discrimination (Bohren et al., 2022), which could nonetheless perpetuate under-representation of women in the technology industry (Ashcraft et al., 2016).

Our study’s context is an online platform which lets job applicants in the technology industry practice their interview and coding skills. Mirroring real interviews for computer programmers, the evaluator can see and interact with the coder. Female coders receive lower ratings than men. These gender gaps in peers’ assessments of coding ability and problem solving correspond to around 12 percent of a standard deviation, are largely independent of the gender of the evaluator, and remain when we control for interviewees and interviewers’ level of education, experience, and preparation level. In the later years of our sample, we also see an objective measure of code

quality, which indicates whether the code produced correct answers. The gender gap remains even when we control for this measure of objective performance.

These persistent gender gaps in subjective ratings could be a combination of bias in evaluations, men and women writing code differently, and differences in how they talk about their code. Guided by a model of discrimination in the spirit of Lundberg and Startz (1983), we derive empirically testable implications and we use two randomized field experiments to shed light on the underlying mechanisms. Both experiments vary the amount of information seen by evaluators. First, we evaluate an experiment conducted by the platform, which retained the face-to-face component but provided the evaluator with objective information in real time about the candidate's performance before their rating is chosen. Second, we remove the face-to-face component by asking evaluators to assess the coding performance of a candidate based solely on the code written. Within the context of without face-to-face interaction, we compare evaluations when gender is visible or hidden, in the spirit of seminal work on blind evaluations (Goldin and Rouse, 2000).

Our first experiment focuses on the possibility that evaluators may incorrectly believe that women on the platform write worse code. If they can only imperfectly judge the quality of the code themselves, this would lead them to penalize women relative to men. To evaluate this, we study the randomized roll-out by the platform of objective code quality measures. These "unit tests" were made available to the evaluator before they chose their ratings, and assessed whether there were errors when the code was executed, and whether it produced correct answers to test cases. We find that the ability to better assess code quality changed evaluation behavior but not the gender gap, which is consistent with evaluators' beliefs being well-calibrated on average.

In principle, the small effect of providing these objective code quality measures to evaluators could stem from the fact that they are uninformative. However, we match the coders in our database to additional data from Revelio Labs, and show that differences in the objective performance measures correlate strongly with future labor market outcomes: A one standard deviation increase in the average objective score measure of platform participants is associated with a 6 percent higher starting salary.

Our second study aims to assess whether men and women write code that is evaluated differently even if gender is hidden, or if gender gaps only arise when gender

is visible. We answered this question by running a pre-registered randomized online experiment in which computer science students were asked to evaluate code taken from the platform itself. The experiment randomized whether gender was revealed by the first name of the code, or only initials were shown so that gender is masked. We find that there is no gender gap in *either* of these treatment arms, despite the code evaluated being identical to what the original evaluators saw. After arguing that evaluators did not simply ignore the names we provided, we conclude that these results have two implications. First, they imply that men and women write code that is similar in overall quality, as opposed to there being a gendered pattern in the code written that could explain the gaps in ratings on the platform (Vedres and Vasarhelyi, 2019). Second, they show that revealing gender does not by itself introduce bias.

The results are hard to reconcile with the traditional concepts of taste-based and statistical discrimination. Instead, they suggest the gender gaps we see hinge on personal interaction, even though the ratings we focus on are for code quality specifically. However, women do not receive lower scores for communication or likability, for which we have separate ratings. A plausible explanation is that “implicit” bias in quantitative skill assessment comes into play specifically when personal interaction makes gender very salient. This is in line with the literature on implicit discrimination and stereotypes (Bertrand et al., 2005; Bordalo et al., 2016; Carlana, 2019; Hangartner et al., 2021; Barron et al., 2022; Cunningham and de Quidt, 2022; Kessler et al., 2022). By eliciting preferences from performance evaluation data, our results complement previous approaches relying on the use of the implicit association test (IAT). In particular, these implicit biases are more likely to emerge during face-to-face interactions, compared to other contexts. Our preferred explanation, in line with the sociological literature, is that individuals are “doing gender” (West and Zimmerman, 1987) during face-to-face interactions. This intuition is also consistent with recent work documenting differential treatment of female candidates during in-person seminars in economics (Dupas et al., 2021; Handlan and Sheng, 2023).

In addition to our pre-registered analysis of gender bias, we conduct the same tests for racial bias. Coders who are not white or East Asian receive lower scores, conditional on the objective measures of code quality. Unlike for gender, however, we find that making race visible via the first name is enough to widen the racial gap in evalua-

tions. This suggests that traditional taste-based or statistical discrimination may be at play, without personal interaction being a necessary precursor for bias. It also implies that evaluators paid attention to the names they saw during our experiment.

This paper contributes to several strands of the literature. First, it connects to work on the role of information in the hiring process. Using methodology such as resume audit studies, previous authors have established the existence of discrimination in the labor market (Bertrand and Mullainathan, 2004; Neumark, 2012; Kroft et al., 2013; Farber et al., 2016). However, it has proven difficult to separate out rational statistical discrimination, statistical discrimination with incorrect beliefs, and taste-based discrimination. In a recent contribution, Bohren et al. (2019) do so for an online Q&A forum by studying how discrimination changes as prior evaluations become available. One feature which distinguishes our study from others in this vein is that we can compare contexts with and without personal interaction. This turns out to change how bias is expressed. We are also able to provide real code excerpts for external evaluators, which eliminates deception that is common in audit studies (Kessler et al., 2019, 2022). Finally, our study focuses on performance evaluations, as opposed to resume ratings, in a context where women are severely under-represented, and where we can show that our measures of skill are related to future labor market outcomes.

Another line of research has investigated factors behind the slow progression of women in high-paying occupations (Bertrand et al. 2010, Goldin 2014, Roussille 2020), and to a growing literature documenting potential causes of under-representation of women in the technology industry specifically (Terrell et al., 2017; Murciano-Goroff, 2018; Miric and Yin, 2020; Boudreau and Kaushik, 2020). Part of the explanation may lie in how information about past performance is assessed and interpreted in occupations that require different skills. However, empirical research in this area faces the challenge that ability and performance are usually hard to quantify in high-skilled labor markets. Unlike previous studies which rely on measures of performance such as billable hours for lawyers (Azmat and Ferrer, 2017) or patients' death for surgeons (Sarsons, 2022), we have access to a problem-specific objective measure of performance for computer programmers. Combined with experimental variation in evaluators' information sets, this helps us measure bias and understand its nature.

Finally, our paper complements a recent literature on gender gaps in performance

evaluations and hiring decisions. Mocanu (2023) finds that women’s relative evaluation scores and share of new hires both increased after a reform to hiring processes in the Brazilian public sector mandated the use of "impartial" recruitment practices. Consistent with our findings, the improvements she documents are largest for employers who switch from interviews and oral exams to a mixture of interviews and blindly written tests. Combining this with our results, and to the extent that the settings are comparable, this suggests that the removal of personal interaction might help explain these gains, as opposed to requiring a move to completely blind tests. Moreover, and in line with the interpretation of our results, is the new evidence by Brown (2023) showing that disparate outcomes by gender are possible even without traditional taste-based or statistical discrimination. Finally, both Feld et al. (2022) and Avery et al. (2023) show that providing recruiters with more information can reduce gender gaps—at least in their settings, which do not involve live personal interaction. Feld et al. (2022) focus on skill measures which are not directly coding-related. Avery et al. (2023) examine the introduction of an AI hiring score. Our paper shows that the context in which this additional information about candidates’ performance is provided is critical to understanding why it reduces bias in the selection of candidates.

## **2 Administrative Data from Face-to-Face Coding Interviews**

Recruiters of programmers are in the unusual position of being able to test a prospective employee’s ability to solve problems using skills they would require on the job. For many technology companies such as Google and Atlassian, this is achieved via live coding interviews which ask candidates to complete a realistic programming task. Our data come from one of several specialized platforms have been developed for this purpose. Examples include CoderPad, Coderbyte, HackerRank, Codility, and Pramp. These companies vary in their business models, ranging from interview practice platforms to those that actively source and screen candidates for specific employers. In our case, the company focuses on practice interviews.

As we describe below, we received multiple datasets from the platform, which span different periods. We then linked use these data in an experiment outside the platform, and the link the data to individual-level labor market information from Revelio labs.

Figure A5 presents the overall data infrastructure of the paper.

## 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 the available programming languages. They can then schedule an interview during one of many fixed time slots, with the platform suggesting slots which already have users with similar profiles. When that time arrives, users within the time slot are paired based on their similarity scores using Edmunds' Blossom algorithm.<sup>1</sup>

The paired users interview each other in turn. Depending on the language and self-reported ability and experience of the interviewee, one of 32 different coding problems is assigned. The interviewee then solves the coding problem in an online text editor that both sides see. 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, creativity, likability and overall performance.

Users' online reviews of their experience highlight several appealing features for the study of gender gaps in performance evaluations in a high skilled labor market, compared to a more traditional lab experiment. The platform provides an environment where realistic tasks are performed under time pressure by early-career computer programmers. 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."*

The platform also mimics the competitive environment in which the software developers are recruited, as they are potentially competing for the same jobs. However, the participants have clear incentives to cooperate, as one user writes:

*"Doing practice interviews with humans who talk to you was much more valuable than working with a review book or online lists of problems. And [the platform] users I paired with were consistently helpful, polite and professional."*

---

<sup>1</sup>This algorithm chooses a matching that maximizes the total of the similarity scores of paired users.

## 2.2 Description of the Data

The experiment we analyze first (Section 4) occurred during the period of covered by our first dataset, which covers 2015 to 2018. Between December 18, 2015, and April 18, 2018, users completed 60,513 interviews. The users are mainly from English-speaking countries (the US, UK and Australia) but also Europe, Brazil, Chile, India and Russia. The user base has grown rapidly, starting with a few users per day in early 2016 to 150 per day in mid-2018 (see Figure A4). Candidates participate in as many practice interviews as they like. Each time, they are paired with a different counterpart. From August 2016 to March 2018, users participated in 12 sessions on average.

Further descriptive statistics for the population of users are shown in Table 1. Participants are high-skilled, and the vast majority graduated in STEM fields. One third had Masters degrees, and nearly all others held a bachelor’s degree (see Figure A2). Two thirds of users had computer science degrees, with most of the remainder spread across engineering, mathematics, statistics and the hard sciences (see Figure A3). Eighteen percent of users were female. Consistent with evidence from Murciano-Goroff (2018), we find that women declare lower level of preparation on average.

Our second experiment in Section 5 uses data from a more recent period, from April 2018 to May 2021.<sup>2</sup> Crucial for our analysis, this second dataset contains first and last names. This allows us to link the data from interactions on the platform to a database from Revelio Labs which provides us with future labor market outcomes for participants. We discuss the Revelio dataset further in Section 4.8. The users’ names also allow us to predict the race and ethnicity of platform participants, which we use for a complementary analysis of racial discrimination in Section 7. Finally, this newer dataset contains the full code script written by each interviewee on the platform.

## 2.3 Gender Gaps in Evaluations of Code Quality

Figure 1 and Table 2 show the gender gaps in evaluations at the end of these interviews between January 2016 to July 2017, before any interventions. The information that evaluators see about coders is held constant in this 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.

---

<sup>2</sup>The construction of the sample and corresponding descriptive statistics are presented in Appendix Table C9, Table C10 and Table C11.



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 changes in composition as the platform grew. They do not vary with the gender of the interviewer, consistent with recent studies challenging the notion that female job applicants will be evaluated more favorably when they are paired with female versus male interviewers, consistent with prior evidence on the contrasted effect of matching female job candidates with female interviewers (Rivera and Owens, 2015). Nor do they vary substantially by problem difficulty (see Figure B7).

### 3 A Guiding Model of Discrimination

Without further evidence, the gender gaps we see are consistent with 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 throughout the rest of the paper.<sup>3</sup>

#### 3.1 Model Setup

The role of an interviewer is to estimate and provide an evaluation of the performance,  $y_i$ , of job candidate  $i$ . The candidate’s true performance is unobservable, but the interviewer sees an imperfect signal of it,  $\theta_i$ . 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 the rating of code quality.

For analytical simplicity, we assume that interviewers believe that the performance of candidates of gender  $g \in \{m, f\}$  is normally distributed in the population, with mean  $\mu_g$  and variance  $\sigma_g^2$ .

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

They may believe (correctly or incorrectly) that the mean,  $\mu_g$ , and standard deviation,  $\sigma_g^2$ , differ between male and female candidates in the population.

The signal that an interviewer observes is unbiased, but noisy. Specifically,  $\theta_i =$

---

<sup>3</sup>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.

$y_i + \varepsilon_i$ , where  $\varepsilon_i$  is normally distributed with mean zero and variance  $\sigma_\varepsilon^2$ , and is independent of both  $y_i$  and  $g$ . The unconditional distribution of  $\theta_i$  is as follows.

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

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:

$$b_i = E[y_i | \theta_i, g] = s_g \theta_i + (1 - s_g) \mu_g \quad (3)$$

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.<sup>4</sup> 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,  $\mu_g$ . By contrast, the interviewer's beliefs about the population distribution of ability would be completely irrelevant if the signal were perfect.

### 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:

$$r_i = R(b_i | g_i, \mathbf{c})$$

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

<sup>4</sup>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  $\sigma_\varepsilon^2$  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,  $\sigma_g^2$ , to be the same for both genders, which implies that  $s_m = s_f = s$ .<sup>5</sup> Then the gender difference in beliefs about code quality for a given signal realization,  $\theta_i$ , is:

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

Equation (4) shows that beliefs—and thus interview ratings—will reflect the interviewer’s preconceptions about the performance levels of men and women. This implies a gender gap that (in this example) is constant and independent of the candidate’s interview performance. This gender gap is larger if the signal is noisier so that  $\sigma_\epsilon^2$  is larger, or the interviewer’s beliefs are more strongly held so that  $\sigma_g^2$  is smaller.

Since the gender gap in Equation (4) is conditional on interview performance, it is 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  $\mu_m$  and  $\mu_f$  reflects a mistaken belief (a “bias”). This is *non-rational* statistical discrimination.

### 3.4.2 Non-Statistical Biases

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 differs by gender even given the same posterior belief ( $b_i$ ) about code quality:

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

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

---

<sup>5</sup>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 knowingly penalize women relative to men. In this case, simply knowing the coder’s gender is enough to drive bias in evaluations. An alternative possibility is that they unconsciously (or “implicitly”) discriminate. Bias may then only arise or will be exacerbated when gender is made salient through photos or extended personal interaction. We examine all of these possibilities below.

## 4 Experiment I: Providing Objective Information

Starting from July 8, 2017, the platform provided a natural experiment which lets us test the hypothesis that the gender gap in code quality ratings is driven by incorrect beliefs: Direct measures of code quality began to be rolled out in the form of automated (“unit”) tests assessed whether the code ran without errors, and produced correct answers. These tests were visible to both the evaluator and interviewee before the subjective rating was chosen. Figure C19 shows an example unit test, along with the prompt (Panel A) and a sample answer (Panel B). Not all users activated these tests, but they provide valuable information for the majority of candidates who did.

### 4.1 Theoretical Prediction

The model in Section 3 has concrete predictions for the effect of this intervention: The gender gap in ratings should narrow in response if the gap is driven by non-rational statistical discrimination based on incorrect beliefs. Our results are, however, more consistent with interviewers’ prior beliefs being well-calibrated on average.

Letting  $\mu_g^*$  be the true average ability of gender  $g$  candidates, the unconditional gender gap in beliefs is given by Equation (6).

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

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 on the signal they observe, and reduces the role for preconceptions about gender differences in ability.<sup>6</sup>

---

<sup>6</sup>The distributions of coding ability need not be invariant to the information structure in equilibrium, since less precise information undermines the incentive for an individual to become productive. Craig (2023) focuses on this issue. In our setting, however, the set of coding solutions is fixed.

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, more information will shrink it. If they believe it is smaller than in reality, the gap would widen. In this sense, a finding that the gap in interview ratings narrows would simultaneously provide evidence of bias, and an effective solution to that bias.

## 4.2 Intervention

The automated unit tests introduced by the platform evaluated whether the code ran without errors, and produced the correct answers for test cases. Users could choose to activate the tests by pressing a button (see Figure A1). The evaluation was then visible to both the interviewer and the interviewee. Once they are activated, users could run the tests any time, and observe pass/fail outcomes. We view this as equivalent to increasing the precision of the signal,  $\theta_i$ , in our theoretical model.

## 4.3 Treatment Assignment

Treatment assignment was randomized by the platform, but evaluation is complicated by non-random matching between users. Availability of the unit tests was phased in gradually. 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 A6 details how new users are assigned to treatment or control as they enter the platform during the phase-in period. When a new user  $i$  is paired to another user  $j$ , there are two possibilities. First, if both  $i$  and  $j$  are new users or have only been in the control condition in the past, the pair is randomized into treatment with a 7 percent probability. Once treated, a user always remains in treatment for future interactions. Second, any candidate matched with a partner who is already in the treatment condition will themselves be treated (without randomization).

This imperfect randomization motivates some of our robustness tests in Section 4.6. However, we note that baseline characteristics are quite balanced between the treated and the control groups, as shown in Table B3. 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.

## 4.4 Incomplete Take-Up

If all users had activated the objective code quality measures when they were available, our design would have allowed us to directly estimate average treatment effects by comparing outcomes between users in the treatment and control groups. However, users could *choose* whether to activate the device during the interview, and not all did. We account for this with two-stage least squares (2SLS).

We start with an Intention-to-Treat (ITT) model:

$$Y_{it} = \alpha + \beta T_{it} + \theta_t + \epsilon_{it} \quad (7)$$

where  $Y_{it}$  is the score of individual  $i$  on date  $t$ , and  $\theta_t$  are date fixed effects.  $T_{it} = 1$  if the feature was enabled for a pair of users, and 0 otherwise. The ITT is  $\beta$  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 two-stage least squares (2SLS):

$$Y_{it} = \gamma + \delta D_{it} + \lambda_t + \eta_{it} \quad (8)$$

$$D_{it} = \mu + \pi T_{it} + \zeta_t + v_{it} \quad (9)$$

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  $\lambda_t$  and  $\zeta_t$  are time fixed effects. Standard errors are clustered at the date level.

## 4.5 Results: A Persistent Gender Gap in Evaluations

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 Equation (8) are shown in Table 3. Panel A shows results for all users, then Panels B and C show results for men and women separately. For each outcome, the first column of the top sub-panel present ITT estimates of Equation (7), and the second column presents 2SLS estimates Equation (8). The first stages Equation (9) are summarized in the lower sub-panels.

**Activation.** 71 percent of users enabled the objective code quality tests when they were made available. This strong first stage suggests that the code quality ratings were observed and valued by participants. Additionally, we observe a lower first stage for women (0.678, S.D=0.016) than for men (0.721, S.D=0.016), consistent with evidence of gender differences in response to feedback (Coffman et al., 2023).

**Complier Characteristics.** We show the observable characteristics of compliers in Table B4.<sup>7</sup> As expected given the balance checks in Table B3, the treated and untreated complier estimates are very similar. Column (5) also presents characteristics for never-takers. The comparisons in Table B4 reveal that the representation of most subgroups among compliers is similar to the overall sample, although compliers do have slightly less experience. However, the results confirm the gender gap in activation: Compliers are less likely to be women than never-takers.

**Impact on Subjective Ratings.** The estimates in Table 3 suggest that both men and women in the treated group receive higher ratings than their peers in the untreated group in problem solving, communication, and hireability ratings. However, treatment did not disproportionately increase ratings of women. Instead, the increases in ratings are generally larger for men, particularly for coding and likability, where the effects are only marginally significant for women. As a result, gender gaps in subjective ratings persist or even increase following the introduction of the device.

## 4.6 Robustness Checks and Mechanism Checks

Table B5 provides robustness checks to assess the validity of our results. Panel A shows a baseline in which we estimate the ITT model interacted by gender:

$$Y_{it} = \alpha + \beta T_{it} + \gamma T_{it} \times \text{Woman}_i + \theta_t + \epsilon_{it} \quad (10)$$

with variables defined as in equation 7. The coefficient  $\beta$  is the overall ITT, and  $\gamma$  is the additional effect of being assigned to treatment on the gender gap in ratings.

**Adding Covariates.** In Panel B, we introduce month-of-interview fixed effects. Then in Panel C we include date-of-interview fixed effects. These help account for the fact

---

<sup>7</sup>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  $D_i$ .

that the share of users treated changes over time, as does the composition of users (an issue we discuss more below). The treatment coefficient shrinks slightly but stays highly statistically significant. The interaction with gender remains imprecisely estimated and, if anything, suggests that treatment widened the gender gap. We control for individual characteristics in Panel D and find virtually the same results, while the inclusion of interviewee fixed effects in Panel H attenuates the treatment coefficient on most outcomes, with the interaction coefficient  $\gamma$  statistically insignificant.

**Alternative Samples and Empirical Designs.** 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. In Panel F, we also exploit the staggered introduction of the objective quality measures in a difference-in-differences framework over the whole period, including month-of-interview fixed effects.

$$Y_{it} = \alpha + \beta T_{it} \times \text{Post}_t + \gamma T_{it} \times \text{Post}_t * \text{Woman}_i + \theta_t + \epsilon_{it} \quad (11)$$

The results in are very similar to those on the post-treatment period only.

**Endogenous Matching Between Users.** Since the treatment condition is potentially contaminated by the matching process, a naive comparison between treated and control users could provide a biased estimate. To address this threat, we control our regression results with the propensity score obtained from a matching procedure in Panel G of Table B5. For the matching procedure, we control for month-of-interview fixed effects, and, for both the interviewer and the interviewee, by 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, as well as gender. Reassuringly, controlling our regressions for the propensity score matching does not affect our results. In an alternative specification, we estimate the propensity score by logistic regression and the Conditional Average Treatment Effect (CATE) directly using a single-equation lasso and find consistent results (available upon request).

**Changes in 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 uses the same characteristics. Nonetheless, we explore changes in user composition over time and in response to treatment, which could mat-



ter because the share of users treated increases over time. Even though the results are reassuring, our main specification controls for date-of-interview fixed effects, so that our results cannot be explained by selection that affects only one group.

First, Figure B12 shows that the gender composition of users was stable after the introduction of the unit tests. However, there could still be changes in which women select onto the platform. Figure B14 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 B13 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 also 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 B15 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 evidence that scores increase for both men and women when unit tests are available, they could bolster learning over time. In turn, because men are more likely to activate the tests, differences in the rate of learning could conceivably explain why the tests do not help close the gender gap in ratings. We find no evidence to support this hypothesis. To test it, we plot the number of tests passed over time in Figure B9. We provide two versions of these learning curves, one over calendar time (Panel A), and one by number of sessions completed (Panel B), to account for the fact that women might not be using the platform as frequently as men. The rate of learning over calendar time is remarkably similar for men and women, with only a level shift down in the number of tests passed by women. If anything, the curve is steeper for women in terms of sessions completed.

Additionally, we explore the possibility that unit tests could be interpreted differently for women relative to men. This could occur if one group were more likely to activate the tests because they have lower self-confidence, or if they want to signal their ability. We assess this in Figure B10, 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 vary similarly with objective performance for men and women.

**Problem Assignment.** We next test whether women were systematically assigned different problems. Although our regressions have problem fixed effects, evaluators may learn more from differences in performance on harder problems, or problems where performance is highly variable. To explore this, we construct a measure of problem difficulty: the average objective performance of both men and women on that problem. As Figure B16 (Panel A) shows, problems differ substantially in difficulty.<sup>8</sup> We also construct a measure of the precision of the signal that evaluators see by ranking problems by the standard deviation of performance (Figure B16 Panel A). Table B1 verifies that the gender of the interviewee does not predict the type of problem assigned, both in terms of difficulty and standard deviation. More broadly, Table B6 shows that, with the exception of interviewer’s years of experience, participants’ characteristics are reasonably balanced across problem’s average difficulty, split at the median ratio of tests solved over tests passed.

**Evaluator Assignment.** We also 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 B1, show that female users are not more likely to be matched with a harsh evaluator.

**Problem Difficulty and Precision of the Signal.** Finally, we test whether evaluators update differently for men and women even for the same problem, which could help explain the persistence of the gender gaps. To assess this, we examine the effect of treatment on gender gaps for different problems. Splitting problems into groups at the median level and standard deviation of difficulty to ensure we have enough power, we estimate Equation (8) separately for each group and each gender.

The results are presented in Figure B17. Like Sarsons (2022), we do find some evidence of differential updating. Evaluators appear to update less from the provision of unit tests for women than for men, especially for low difficulty problems. In Panel A, we document an asymmetric updating pattern by gender and problem difficulty. For men, the improvement in ratings is larger for low-difficulty problems than for high-difficulty problems, although we cannot formally reject that the effects are equal

---

<sup>8</sup>The ranking of problems’ difficulty do not seem to vary by gender. Figure B11 shows the relative ranking of problems by gender, proxied by the average performance of users of each gender for each problem. The orange vertical lines indicate any positive (negative) deviation upward (downward) of female users’ ranking compared to male users’ ranking. The rankings are very similar.

across problems of various difficulty levels. There is suggestive evidence of a reversed effect for women: the treatment effects are imprecisely estimated for both groups of problems, but the magnitude of the effect is larger for high-difficulty problems. Panel B confirms that the treatment effect on subjective ratings is higher for both genders for low standard-deviation problems (when the signal is more precise), with lower but imprecisely estimated point estimates for women.

These results also provide a test for inattention: If evaluators had not paid attention to the introduction of the device, they would not have adjusted their beliefs about users' performance differently according to the precision of the signal.

#### **4.7 Persistent Gender Gaps Controlling for Objective Measures of Code Quality**

The gender gaps in ratings are not fully explained by objective performance differences, as measured by these tests. Figure 3 (Panel A) plots average subjective ratings in coding by objective performance (share of tests passed), and Panel B shows ratings for problem solving. The plots are separated by gender.<sup>9</sup> Women receive systematically lower subjective coding and problem solving ratings than men who perform equally well, although the gender gap in subjective ratings is halved for users at the top of the objective performance distribution. These results are confirmed when we control for sociodemographic characteristics of the interviewer and the interviewee, as well as date-of-interview fixed effects (see Table 4). The residual gaps amount to about 6 percent of a standard deviation.

To test whether less experienced participants are more likely to hold inaccurate beliefs, we look at how the gender gap conditional on the objective measures of performance vary with the interviewer's experience on the platform. The results are shown in Table B7. The gender gap in subjective ratings does not vanish when we account for interviewer experience, proxied by the number of past interviews, the number of interviews with female users, or whether the previous interview was with a top performer female users, defined as a female user who performed above the median. Hence our empirical investigation doesn't support the hypothesis that more experience on the platform or exposure to female coders plays a significant role in this context.

---

<sup>9</sup>We split the sample in two groups: users who passed all unit tests, and those who didn't, given the bimodal distribution of the objective performance measure (see Figure 2).

## 4.8 Measures of Code Quality and Future Labor Market Outcomes

As a way of validating the platform’s measure of code quality, we linked our interview data to future labor market outcomes from Revelio Labs. Revelio offers a database of hundreds of millions of 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.<sup>10</sup>

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.<sup>11</sup>

From the set of interviewees on the platform, we select those residing in the US who have a bachelors or masters 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.<sup>12</sup> The final sample consists of 5,126 matched CS graduates from 2016 to 2023. For 50 percent of this sample, we have data on their objective performance on the platform. The outcome variable of interest is the first salary after graduation, although we also look at average salary after graduation. Data from Glassdoor indicates that the average salary for CS graduates in 2023 is \$85,000, our matched sample has an average starting salary of \$81,000.

From here, we use a Mincer-type wage regression of log earnings on individuals’ characteristics such as gender, race, the highest degree obtained, institution-of-highest-degree, year-of-graduation and city fixed effects.

Results are presented in Table 5. Column (1) shows that there is a 6 percent residual gender gap for computer science graduates in their first salary after graduation. This

---

<sup>10</sup>More detail regarding the Revelio data database is available [www.reveliolabs.com](http://www.reveliolabs.com).

<sup>11</sup>See Loyalka et al. (2019) for a cross-country analysis of CS university graduates.

<sup>12</sup>This follows the same matching method adopted by Abramitzky et al. (2012), Abramitzky et al. (2014) and Abramitzky and Boustan (2017).

residual gender pay gap could reflect both supply and demand factors, such as the role of gender differences in preferences for job amenities, gender differences in job search (Le Barbanchon et al., 2021; Cortes et al., 2021), in earning expectations and negotiation (Reuben et al., 2017; Roussille, 2020), or discrimination. We remain agnostic about the ultimate sources of the gap, focusing instead on validating the objective measure of coding quality, as well investigating the returns to skills in this labor market.

In column (2), we add controls for the average objective measure of coding quality across all sessions on the platform, the number of past sessions on the platform and whether the participant had graduated at the time of their interview session.<sup>13</sup> We find a positive and statistically significant coefficient (0.068, SD=0.032) for the standardized objective score measure: Going from the 25th to the 75th percentile of standardized score is associated with a wage increase of 6 percent. While this objective measure may be correlated with the quality of training received by participants, this exercise validates the unit tests as predictive of labor market outcomes. Including the subjective coding, communication and problem solving ratings to the regression has little effect on the magnitude of the coefficient on objective code quality.

Finally, we note that there is suggestive evidence of heterogenous returns of skills by gender in columns 3 to 6, with little zero return of the objective measure of coding performance for women.<sup>14</sup> However, the estimate for women is imprecise.

## 5 Experiment II: Blind and Non-Blind Code Evaluation

Having seen no evidence that gender gaps shrink with the provision of additional information, we turn to a second experiment. Using coding solutions taken directly from interactions on the platform itself, we ran an online randomized experiment with computer science students who had familiarity with the relevant programming languages. The experiment asked these computer scientists to evaluate the solutions. We compared these evaluations in a “blind” setting to those when gender was revealed via the name of the code, in the same vein as many other studies in which blind evaluations occurred (Goldin and Rouse, 2000). The aim of this was to establish whether residual gender gaps in subjective ratings are due to gender bias, or unmeasured dif-

---

<sup>13</sup>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 when we add weights.

<sup>14</sup>See Table B8 for estimation results for men and women separately.

ferences in code quality as assessed by evaluators.

The RCT was pre-registered on December 14, 2022.<sup>15</sup> The participants were predominantly Bachelors and Masters level computer science students with familiarity in the relevant programming languages. Full descriptive statistics for the participants are available in Table C13. To complement the discussion here, a comprehensive description of the experiment’s design is available in Appendix C.

## 5.1 Theoretical Prediction

In the analysis below, we compare blind to non-blind evaluations. In the blind condition when gender is unobservable, the evaluator can no longer condition her belief on the gender of the applicant. To form a belief about his or her performance, the relevant belief is therefore the interviewer’s perception of the pooled ability of men and women. Letting  $\lambda_g$  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:

$$y_i \sim \mathcal{N}(\mu, \sigma^2) \quad (12)$$

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

Conditional on the signal,  $\theta$ , the posterior belief of a worker’s performance is:

$$E[y_i | \theta_i, g] = \tilde{s} \theta_i + (1 - \tilde{s}) \mu \quad (13)$$

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

$$\text{Gender Gap} = \tilde{s} (\mu_m - \mu_f). \quad (14)$$

This highlights that there cannot be a gender gap when evaluation is blind unless there are true differences in productivity between the two groups; and thus that comparing blind and non-blind evaluations of the same code reveals the extent of gender bias.

We note that any true differences in productivity would have to be beyond what is captured by our objective measures of code quality, since there is a gender gap even conditional on these measures.

---

<sup>15</sup>ID: AEARCTR-0009816. The pre-analysis plan is available on the AEA RCT registry website (updated version: February 17, 2023).

## 5.2 Empirical Design

### 5.2.1 Selecting Code Blocks from the Platform

We use a stratified random sample of de-identified code blocks written by a subset of men and women on the platform, which span coders of different skill levels and problems of different levels of difficulty. An example of such a code block is shown in Figure C19 Panel B. For each block, we have the platform's objective measures of performance, including sub-test results. Descriptive statistics from each step of the sample construction are presented in Table C9, Table C10 and Table C11.

For comparison, Table C12 presents estimates of the gender gap in the experimental sample, controlling for objective performance. The methodology is the same as for Table 4. Within this sample, we find an even larger gender gap in subjective ratings. Finally, we stratify by gender, race and coding performance, i.e whether the coder's performance (unit tests) is below or above the median for any given problem.<sup>16</sup>

### 5.2.2 Treatment

Each evaluator  $i$  is assigned a set of four coding solutions in a random order. We stratify the experiment by gender and performance: Out of four code scripts, each evaluator sees two code scripts written by female coders, among which one of each is high-performing according to the objective tests.

We use a within-subject design, in which each evaluator encounters two conditions. In the "non-blind" condition, gender is revealed via the given name of the coder. In the blind condition, gender is hidden because only the initial of the given name is seen. An example of each treatment condition is presented in Figure C20. For each evaluator  $i$ , the gender of the coder will be revealed for half of the problems. To account for potential priming effect, we randomized whether the gender of the coder is revealed in the first or in the second half of the study. Table C14 confirms that the characteristics of evaluators are balanced across treatment orderings.

### 5.2.3 Outcomes

**Main Outcome.** For our primary outcome, we asked evaluators to judge the quality of the code using the same Likert scales as on the platform. This scale ranged from

---

<sup>16</sup>We choose to stratify by race to keep a representative population of coders for our experiment. We further discuss racial bias in Section 7.

1 to 4. For all primary hypotheses, we use these responses as our main dependent variable. We note that this outcome differs significantly call-back rates, which are often used in correspondence studies. First, as discussed by Kessler et al. (2019), call-back rates depend on employers' interest in a candidate, but also the likelihood that the candidate will accept the job: an employer will not pursue candidates who will be unlikely to accept a position if offered. Second, callback rates only identify preferences at one point in the quality distribution. In our setting, we will learn about evaluators' preferences at various levels of the performance distribution.<sup>17</sup>

**Additional Measures.** We also have a secondary outcome: evaluators' prediction of the candidate's score from the automated tool. Specifically we ask them how many unit tests out of 10 unit tests do they think were passed. A third outcome is evaluators' prediction of whether a human evaluator decided that this coder passed or failed the interview. Finally, we asked evaluators what is the percent chance that the candidate was later invited for an interview for a role involving coding. This allows us to draw a more direct link between our findings and hiring outcomes.

Additionally, 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 C18. We define our quality sample as those passing the first attention check, and for whom the survey duration is comprised 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.<sup>18</sup>

To measure participants' priors, we exposed them to three different vignettes before they perform their evaluation tasks. We ask them to predict the potential performance of three different hypothetical coders. We cross-randomize the first name (alternating gender) and the skill level for each vignette (see Appendix C).

#### 5.2.4 Incentives

Incentives in our experiment differ from traditional correspondence studies. In part, this is due to our effort to reproduce the incentives and environment faced by participants on the platform. However, it also presents other advantages.

---

<sup>17</sup>While our study models only part of the hiring process, bias at an earlier stage such as the coding interview would show up as structural bias in subsequent rounds (Pincus, 1996; Bohren et al., 2022).

<sup>18</sup>Table C15 confirms that the characteristics of evaluators are also balanced across each treatment order for the quality sample.



First, we do not rely on deception. Participants were clearly informed that these code blocks had been written by real software developers without manipulation, despite the fact that we would not necessarily reveal all information. A drawback of this design is that we had to inform subjects that responses would be used in research, which could potentially have led to experimenter demand effects (De Quidt et al. 2018), but we think providing real code excerpts reinforced the credibility of our design and encouraged participants to exert effort in the evaluation process.

One concern is that we ask evaluators to provide subjective ratings on several code blocks, which could have lowered effort and attention over time. To address this, we included incentivized questions where individuals are asked to predict the unit tests passed by the code. Additionally, we provided a non-financial but potentially powerful incentive selecting a set of evaluators to the Creative Destruction Lab (CDL) 2023 Super Session which brought together world-class entrepreneurs, investors and scientists with high-potential startup founders. CDL Super Session days provided real networking opportunities and exposure to key players in the industry. We expect this increased the incentive for participants to accurately evaluate the code blocks.

Finally, we note that the university student evaluators were not hiring workers or co-workers. Therefore, any residual gender gap in ratings across the blind and non-blind conditions cannot be attributed to homophily, but will reflect valuations of a candidate's performance only. It is therefore a lower bound for overall discrimination in settings where evaluators have ongoing interactions with workers they hire.

### **5.2.5 Hypotheses Tested**

Our pre-analysis plan specified the following hypotheses to be tested.

#### **Primary**

- H1: Code blocks are evaluated differently if the gender of the coder is known.
- H2: Code blocks written by women are evaluated differently when we reveal the gender of the coder, with the gender gap increasing.
- H3: Individual gender bias varies significantly across evaluators.

#### **Secondary**

- H4: The gender identity of the evaluator affects their bias.
- H5: The difficulty of a given coding problem affects evaluator bias.

- H6: The level of the coder's performance affects the degree of bias.
- H7: Prior bias as assessed by vignettes correlates with the evaluator's bias in ratings.
- H8: The characteristics of a given coding problem affects the evaluator's bias.
- H9: The race of the coder affects the degree of gender bias.

## 5.2.6 Econometric Specifications

To test these hypotheses, we proceed as follows. First, we define  $NB_j = 0$  for a blind evaluation  $j$ , and  $NB_j = 1$  for a non-blind evaluation. The variable  $Treatment\_Order_i$  is an indicator for the randomly assigned treatment order ("non-blind then blind" condition versus "blind then non-blind") that the evaluator sees; while  $Script\_Order_j = k$  is used to construct indicators that a given code block was the  $k$ th block the coder evaluated, which we include to account for fatigue and learning.

To test H1, we use the following specification:

$$Y_{ij} = \beta_0 + \beta_1 \times NB_{ij} + \beta_2 \times Treatment\_Order_i + \beta_3 \times Strata_j + \sum_{k=1}^4 \gamma_{jk} \mathbb{1}(Script\_Order_j = k) + \pi_{p(j)} + \delta_i + \epsilon_{ij} \quad (15)$$

where  $Y_{ij}$  is a discrete variable from 1 to 4 which captures the ratings of evaluator  $i$  for code block  $j$ ;  $NB_{ij}$  is an indicator for whether gender is revealed to the evaluator;  $strata_j$  are the four strata fixed effects (female  $\times$  high\_performer);  $\pi_{p(j)}$  are problem fixed-effects. In some specifications, we include evaluator fixed effects ( $\delta_i$ ) and additional controls.<sup>19</sup> Standard errors are clustered at the evaluator level.

In Equation (15), the coefficient of interest is  $\beta_1$ . It measures the average difference in ratings for code blocks where the gender of the coder is revealed, controlling for the treatment order. This does not test for differences across gender, but rather whether non-blind code blocks are evaluated more harshly regardless of gender.

To test H2, we will use the following specification, which is very similar to Equation (15) but interacts the key variables with gender indicators:

$$Y_{ij} = \beta_0 + \beta_1 \times Female\_Coder_j + \beta_2 \times NB_{ij} + \beta_3 \times NB_{ij} \times Female\_coder_j + \beta_4 \times High\_Performer_j + \sum_{k=1}^4 \gamma_{jk} \mathbb{1}(Script\_Order_j = k) + \pi_{p(j)} + \delta_i + \epsilon_{ij} \quad (16)$$

---

<sup>19</sup>Since code blocks characteristics are randomly drawn, including these variables in the analysis should not affect our estimates but could increase precision.

The coefficients of interest are:  $\beta_1$ , which measures the quality difference between male and female code in the blind condition; and  $\beta_3$ , which measures the differential effect of revealing the gender of the coder, depending on what that gender is.

To test H4 to H9, we use variant of Model (16) where treatment effect on gender bias is interacted with, respectively, the gender of the evaluator, the difficulty and characteristics of the code, the coder's performance, the evaluator's bias measured through their priors, and the race of the coder.

### 5.3 Results

Table 6 presents our main results, which center on hypothesis H2. The estimate of  $\beta_1$  shows that in the blind condition, code blocks written by female coders do not receive systematically different lower ratings, unit tests prediction or interview predictions. If anything, the coefficients are positive, although we cannot rule out zero or small negative coefficients. This rules out any systematic meaningful gender differences in coding styles that could drive gender disparities in the face-to-face interviews, but which are not accounted for by the unit tests (Vedres and Vasarhelyi, 2019).

Turning to the effect of treatment, our estimate of the effect of making evaluation non-blind ( $\beta_2$  in Equation 16) is negative but noisy, while the interaction with gender ( $\beta_3$ ) is positive but imprecisely estimated. In this sense, do not find evidence of uniform gender bias that arises when gender is revealed by the first name. However, we note that there are effects for racial subgroups, as we discuss more in Section 7.

Table C16 explores Hypothesis H1. Overall, code evaluated in the non-blind condition tends to receive lower ratings, predicted unit test scores and predicted interview chances. The table also reveals that code blocks seen at the beginning of the task are evaluated more harshly.<sup>20</sup> Finally, we note that we do not find support for H5, H6, H7 and H8, namely that the difficulty and characteristics of the code, the coder's performance and the evaluator's bias measured through their priors affect the evaluators' gender bias in ratings and outcome predictions.<sup>21</sup>

**Priors.** Experiment II also allows us to explore participants' prior beliefs about differences in ability between men and women. Figure C18 shows the distributions of re-

---

<sup>20</sup>Results on the "quality sample" are presented in Table C17 and point to similar effects.

<sup>21</sup>Results available upon request.

spondents' prior beliefs by gender and skill level of the vignette. The continuous lines represent the mean prior for each gender. The dashed lines show the actual performance for each gender. Overall, 82 percent of users pass all unit tests, and evaluators do not systematically underestimate women's performance.

## 6 What Drives the Gender Gap in Code Ratings?

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

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 when gender is not observed. 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.** The lack of a gender gap in blind-evaluated code quality makes it hard to rationalize the gap in evaluations 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 close to zero. Without some form of non-statistical bias in rating behavior, this would also 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 that this is not the case. The experiment provided more 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.** Another possibility is taste-based discrimination. Because there is no evidence of statistical discrimination, we can test for taste-based bias by comparing blind to non-blind evaluations of the same code. If statistical discrimination is not at play, and blinding eliminated or reduced gender gaps, this would suggest taste-based discrimination. Instead, we show that blinding makes 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.

While inattention could drive these results, we think it is unlikely for two reasons. First, there is a high correlation between unit test scores and ratings provided by evaluators, which suggests that evaluators exerted effort and attention during the task. Second and more importantly, we do see an effect of blinding on the dimension of race and evidence of explicit racial discrimination consistent with correspondence studies (Bertrand and Mullainathan, 2004; Bertrand and Duflo, 2017; Kline et al., 2022). This indicates that the null result for gender cannot simply be explained by inattention.

**Gender Differences in Communication Style.** We are left with the conclusion that bias only arises in our data when personal interaction is allowed while the code is written. One explanation for this could be that men and women talk about their code in different ways. If women are less effective at communicating along the way, this could introduce a gender gap that is not there when code is evaluated on its own.

While it is hard to test this directly, we do observe ratings for communication. Figure 4 plots the average subjective ratings in communication (Panel A) and likability (Panel B) by objective performance (share of tests passed), separately by gender. While both high and low performing women received systematically lower subjective coding and problem solving ratings than men who perform equally well (Figure 3), Figures 4 shows that the communication and likability ratings of men and women are comparable across the objective performance distribution. This suggests that for a given objective performance, gender differences in communication styles are unlikely to explain persistence in gender gaps in coding subjective ratings.

**Implicit Bias.** An alternative explanation, which is more compatible with the similarity in communication ratings, is that the gaps stem from a type of “implicit” bias (Bertrand et al., 2005; Carlana, 2019; Hangartner et al., 2021; Barron et al., 2022; Cunningham and de Quidt, 2022). Specifically, gender and differences in mannerisms and behavior become much more salient with personal interaction. This introduces a phenomenon that could perhaps be referred to as a form of “taste-based” bias but might better be referred to as “implicit” bias.

## 7 Racial Discrimination

We were also able to predict race and ethnicity. We first did this based on both the first and last names of the coders.<sup>22</sup> We had two goals. First, we aimed to proxy for the "true" race and ethnicity as observed by participants on the platform during the face-to-face interviews. Second, we sought to test for productivity differences between groups in the blind condition of Experiment II. However, participants in Experiment II were exposed to first names only. We therefore supplemented the measure based on both names by asking two reviewers to provide their best guess of the race of each coder on the basis on their first name only.<sup>23</sup>

We start with ratings from face-to-face interactions on the platform. As Table C19 (Panel A) shows, there is a racial penalty for coders who are not white or East Asian, controlling for objective test scores.<sup>24</sup> To gain power, we group "white" and "East Asian" together because separate point estimates have similar sign.<sup>25</sup> The penalty is robust to the inclusion of evaluator fixed effects for the sample of male coders, but becomes statistically insignificant for the smaller sample of female coders. The gender penalty does not vary substantially when we interact it with racial group (Panel B).

Our results are similar in the experimental sample. We investigate the interaction between race and gender in the context of blinding or revealing the first name of coders. Results are presented in Table 7. In columns 1 to 4, we present results using the two human categorizations of race and ethnicity using first names only; and in columns 5 to 6 the algorithmic categorization using first and last names as benchmark. We again find a penalty for non-white non-East Asian coders, but see that this is especially the case for men.<sup>26</sup> The coefficients are stable across the different categorizations of race (across reviewers 1 and 2), and the magnitude increases with the inclusion of evaluator fixed effects. Overall, these results suggest an explicit bias against non-white non-East Asian men, triggered by distinctively non-white names.

---

<sup>22</sup>We used the Python `ethnicolr` that exploits the US census data, the Florida voting registration data, and the Wikipedia data collected by Skiena and colleagues.

<sup>23</sup>We ensured that the two reviewers had different genders and races. Reviewers' assessment are correlated but not identical. Most first names over which reviewers' assessments differ are white or East Asian according to our predictive algorithm.

<sup>24</sup>According to the racial and ethnicity classification of the predictive algorithm, this includes "Asian, Indian Sub Continent", "Greater African, African" and "Greater African, Muslim". This group constitutes 48 percent of the sample.

<sup>25</sup>Disaggregated results are available upon request.

<sup>26</sup>This category includes coders classified as either South Asian, Black, Latinx or Other.

The different results for race and gender suggest that different mechanisms could be at play for racial and gender discrimination in this context. Unlike for gender, revealing race by displaying the first name does introduce racial bias, as measured by the difference in the racial evaluation gaps in the blind and non-blind conditions. Personal interaction does not appear to be required to see this effect.

## 8 Conclusion

We present two field experiments which show that gender bias in performance evaluation is context-specific. Our focus is on evaluation of coding performance, a common step during recruitment process in the technology industry. Within that context, we evaluate three treatments which systematically vary the amount of information about a candidate’s performance presented to evaluators.

In line with recent work, we show that gender discrimination can take forms beyond the traditional distinction of taste-based and statistical discrimination. Specifically, face-to-face interaction appears to be a precursor without which gender bias does not arise in this context. We argue that this is most consistent with the literature on implicit discrimination and stereotypes (Bertrand et al., 2005; Carlana, 2019; Hangartner et al., 2021; Barron et al., 2022; Cunningham and de Quidt, 2022; Kessler et al., 2022). 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 signaled indirectly.

Our analysis suggests ways to mitigate discrimination in performance evaluation. First, women received lower coding ratings than men only during face-to-face interactions, but equal ratings for communication. Hence, decoupling the coding task from the face-to-face interview might help mitigate biases in the evaluation of cognitive skills. We note that it may be more problematic to remove face-to-face interaction entirely. This could potentially harm female candidates, given that labor market data suggest higher returns to social skills (as measured by communication ratings) are higher for women than men. Future research could explore these apparent differential returns and how they might contribute to the gender pay gap.

Second, women and underrepresented minority coders would both benefit from blind coding reviews, but particularly non-white non-East Asian male candidates. Our

analysis of gender and racial biases reveals that bias against non-white non-East Asian men is robust across all evaluation conditions, including when the face-to-face interactions are removed. This suggests that more traditional taste-based or statistical discrimination may be at play, without personal interaction being a necessary precursor for bias. Further research is needed to better understand the contexts in which biases are triggered and could be mitigated.



## 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.
- 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 Vecchi**, “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.
- Azmat, Ghazala and Rosa Ferrer**, “Gender gaps in performance: Evidence from young lawyers,” *Journal of Political Economy*, 2017, 125 (5), pp. 1306–1355.
- 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 Ditzmann, Stefan Gehrig, and Sebastian Schweighofer-Kodritsch**, “Explicit and Implicit Belief-Based Gender Discrimination: A Hiring Experiment,” Technical Report, CESifo Working Paper 2022.
- 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.

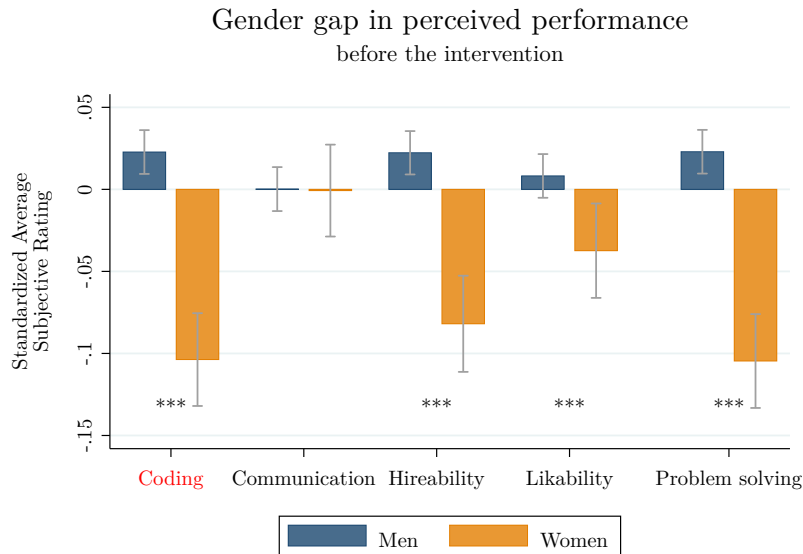
- Bohren, J Aislinn, Kareem Haggag, Alex Imas, and Devin G Pope**, “Inaccurate statistical discrimination: An identification problem,” Technical Report, National Bureau of Economic Research 2019.
- , **Peter Hull, and Alex Imas**, “Systemic discrimination: Theory and measurement,” Technical Report, National Bureau of Economic Research 2022.
- 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.
- Brown, Christina**, “Understanding Gender Discrimination by Managers,” Technical Report, mimeo 2023.
- 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.
- Coffman, Katherine, Manuela Collis, and Leena Kulkarni**, “Stereotypes and Belief Updating,” Technical Report, Harvard Business School Cambridge 2023.
- 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 Jr**, “Complementary Bias: A Model of Two-Sided Statistical Discrimination,” 2019.
- Cunningham, Tom and Jonathan de Quidt**, “Implicit Preferences,” Technical Report, CEPR Discussion Paper 2022.
- 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.
- 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,” *The Quarterly Journal of Economics*, 2022, 137 (4), 1963–2036.
- 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.
- 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.
- 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.
- Phelps, Edmund S**, “The statistical theory of racism and sexism,” *The American Economic Review*, 1972, 62 (4), 659–661.
- Pincus, Fred L**, “Discrimination comes in many forms,” *American Behavioral Scientist*, 1996, 40 (2), 186–194.

- Quidt, Jonathan De, Johannes Haushofer, and Christopher Roth**, “Measuring and bounding experimenter demand,” *American Economic Review*, 2018, 108 (11), 3266–3302.
- 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.
- 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.
- Sarsons, Heather**, “Interpreting Signals in the Labor Market: Evidence from Medical Referrals,” 2022.
- 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.
- 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.

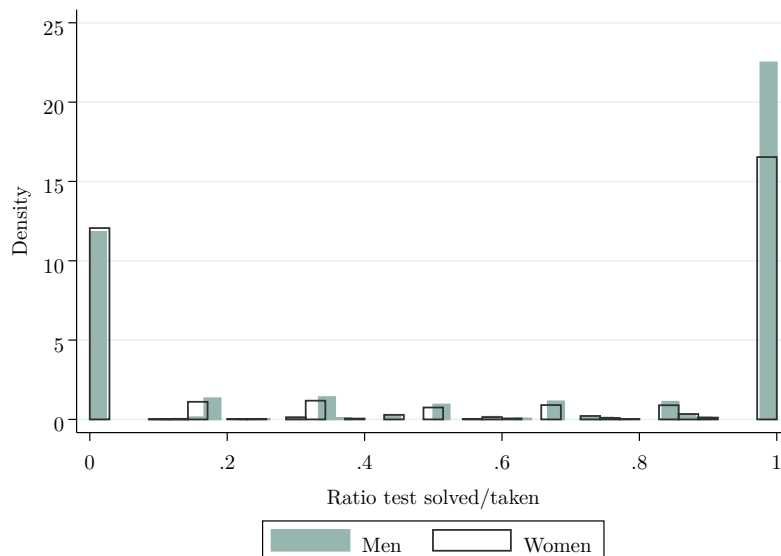
# Tables and Figures

**Figure 1: Pre-intervention gender gaps – Whole sample**



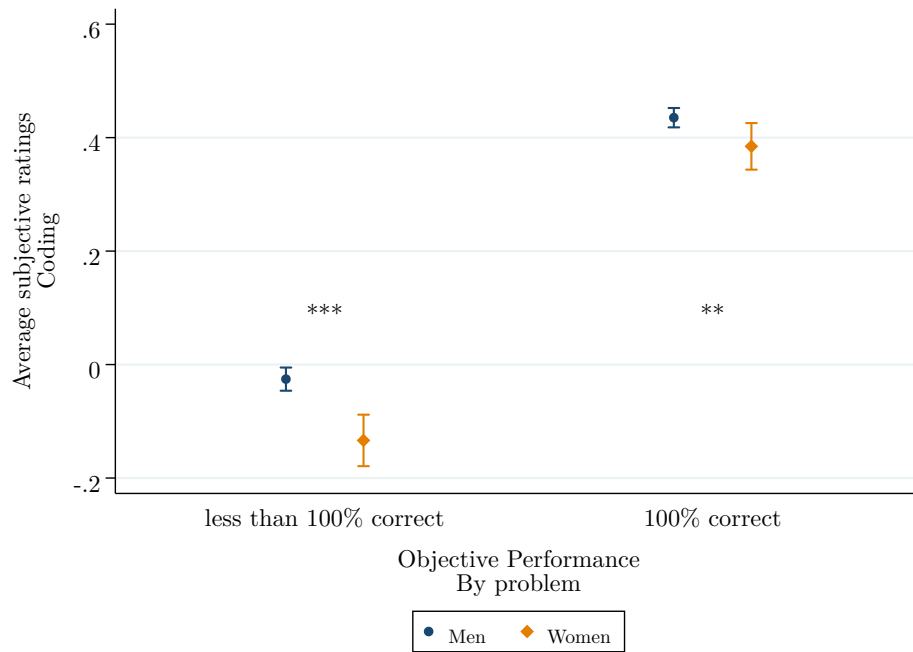
*Notes:* This figure shows the gender gap in peer-rated performance in five categories for standardized variables: coding, communication, hirability, likability and problem solving, for the whole sample. Stars above a category indicate statistical significance of the gap at the one percent level, and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

**Figure 2: Distribution of Objective Performance by Gender**

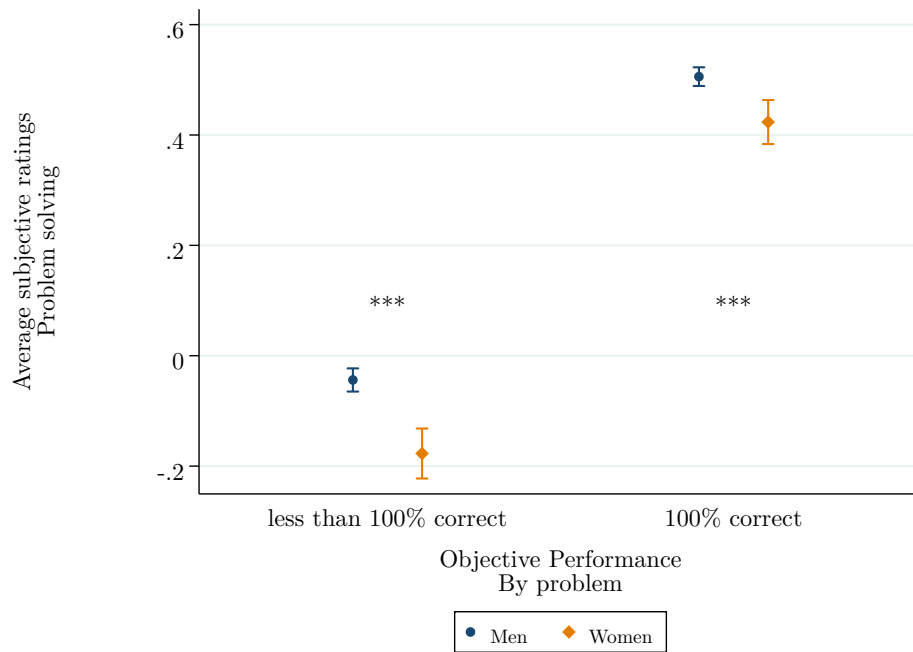


*Notes:* The figure presents the distribution of our objective performance measure (share of tests passed) by gender. As we describe in Section 4, these “unit tests” indicate whether the code ran and produced the correct answers to pre-defined test cases.

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



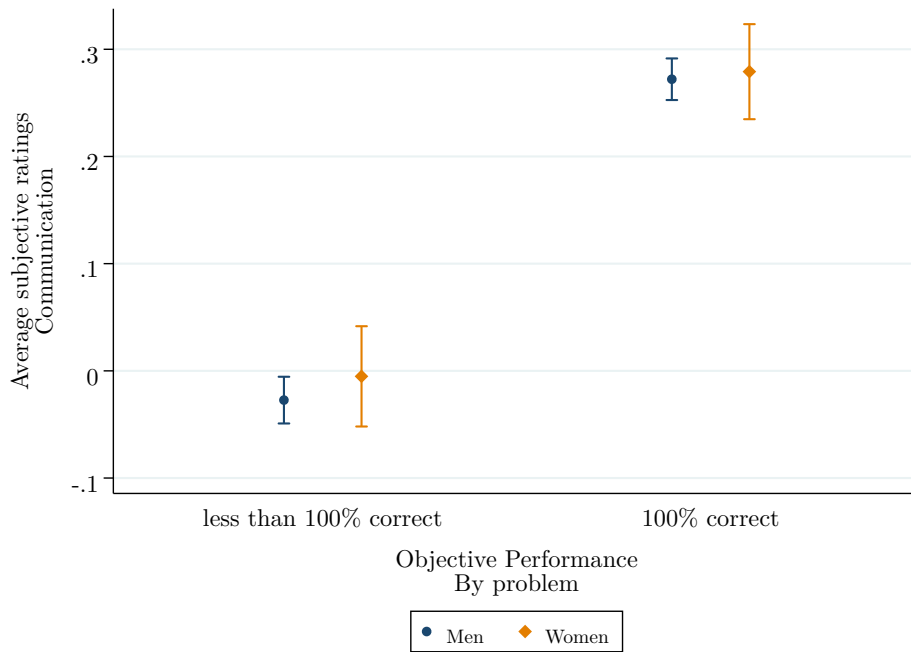
**(a) Coding**



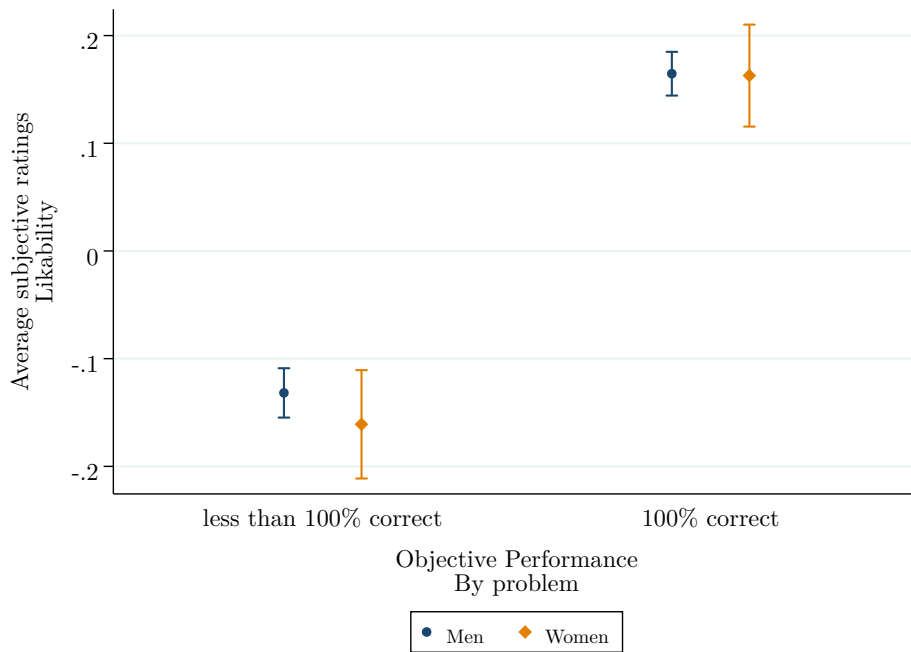
**(b) Problem solving**

*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.

**Table 1: Descriptive Statistics — August 2016-March 2018**

Number of sessions	25,036
Number of interviewees	10,441
Number of interviewers	10,232
Number of problems	31
Share of female interviewees	17.82
Share of female interviewers	17.81

*Panel A: All*

<b>Variable</b>	<b>Mean</b>	<b>Std. Dev.</b>	<b>Min.</b>	<b>Max.</b>	<b>N</b>
Country: USA	0.715	0.452	0	1	49,733
Interviewee's deg.: computer science	0.669	0.471	0	1	49,731
Interviewee without working experience	0.273	0.445	0	1	49,732
Interviewee with a graduate degree	0.451	0.498	0	1	49,733
Interviewee Preparation Level	2.904	0.798	1	5	49,661

*Panel B: Women*

<b>Variable</b>	<b>Mean</b>	<b>Std. Dev.</b>	<b>Min.</b>	<b>Max.</b>	<b>N</b>
Country: USA	0.802	0.399	0	1	8,861
Interviewee's degree : computer science	0.650	0.477	0	1	8,861
Interviewee without working experience	0.304	0.46	0	1	8,861
Interviewee with a graduate degree	0.516	0.5	0	1	8,861
Interviewee Preparation Level	2.784	0.792	1	5	8,855

*Panel C: Men*

<b>Variable</b>	<b>Mean</b>	<b>Std. Dev.</b>	<b>Min.</b>	<b>Max.</b>	<b>N</b>
Country: USA	0.696	0.46	0	1	40,872
Interviewee's deg.: computer science	0.673	0.469	0	1	40,870
Interviewee without working experience	0.266	0.442	0	1	40,871
Interviewee with a graduate degree	0.437	0.496	0	1	40,872
Interviewee Preparation Level	2.930	0.797	1	5	40,806

*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 2: Gender Gap in Subjective Ratings Pre-Intervention**

	<b>Coding</b>				
	(1)	(2)	(3)	(4)	(5)
Interviewee female	-0.127*** (0.016)	-0.121*** (0.016)	-0.121*** (0.016)	-0.121*** (0.018)	-0.118*** (0.019)
	<b>Problem Solving</b>				
	(1)	(2)	(3)	(4)	(5)
Interviewee female	-0.126*** (0.016)	-0.110*** (0.016)	-0.110*** (0.016)	-0.111*** (0.018)	-0.117*** (0.018)
	<b>Communication</b>				
	(1)	(2)	(3)	(4)	(5)
Interviewee female	-0.000 (0.016)	0.000 (0.016)	-0.000 (0.016)	-0.001 (0.019)	0.006 (0.019)
Observations	26,306	25,952	25,952	25,932	25,952
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 January 2016 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 3: Impact of the Introduction of the Automated Measure of Code Quality**

<i>Panel A: All</i>										
	<b>Coding</b>		<b>Problem solving</b>		<b>Likeability</b>		<b>Communication</b>		<b>Hirability</b>	
	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								
<i>Panel B: Women</i>										
	<b>Coding</b>		<b>Problem solving</b>		<b>Likeability</b>		<b>Communication</b>		<b>Hirability</b>	
	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								
<i>Panel C: Men</i>										
	<b>Coding</b>		<b>Problem solving</b>		<b>Likeability</b>		<b>Communication</b>		<b>Hirability</b>	
	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.

**Table 4: Gender Gap in Coding Ratings, Controlling for Objective Scores**

	Subjective Coding Ratings			
	(1)	(2)	(3)	(4)
Interviewee female	-0.0812*** (0.0172)	-0.0638*** (0.0173)	-0.0645*** (0.0173)	-0.0610*** (0.0197)
Objective performance	0.485*** (0.0141)	0.456*** (0.0141)	0.457*** (0.0141)	0.479*** (0.0171)
Interviewer female			0.0320* (0.0165)	0.0298 (0.0189)
Interviewee sociodemographics	No	Yes	Yes	Yes
Interviewer sociodemographics	No	Yes	Yes	Yes
Date FE	No	No	No	Yes
Observations	19,559	19,551	19,551	19,551

*Notes:* This table shows the estimation of the gender gap in subjective ratings in a linear regression model in which we progressively add controls, as described in Section 4.7. We control for objective performance (proxied by the share of unit tests that are correct). In column 2, we add socio-demographic 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 date-of-interview fixed effects.

**Table 5: Labor Market Outcomes**

	Ln(first salary post graduation)					
	(1)	(2)	(3)	(4)	(5)	(6)
Female	-0.063*	-0.074*	-0.063	-0.069*	-0.064	-0.070
	(0.036)	(0.043)	(0.041)	(0.041)	(0.042)	(0.047)
Non White	-0.040	-0.070	-0.064	-0.061	-0.065	-0.068
	(0.035)	(0.046)	(0.043)	(0.043)	(0.043)	(0.045)
Masters Degree	0.126***	0.200***	0.203***	0.212***	0.201***	0.214***
	(0.030)	(0.031)	(0.033)	(0.035)	(0.033)	(0.034)
Objective Performance		0.068**				0.063*
		(0.032)				(0.033)
Objective Performance $\times$ Female		-0.057				-0.058
		(0.054)				(0.061)
Subjective Coding Rating			0.041			-0.027
			(0.027)			(0.069)
Subjective Rating $\times$ Female			-0.027			0.041
			(0.055)			(0.115)
Communication Rating				0.041*		0.038
				(0.023)		(0.042)
Communication Rating $\times$ Female				0.014		0.079
				(0.052)		(0.075)
Prob. Solv. Rating					0.050*	0.028
					(0.026)	(0.052)
Prob. Solv. Rating $\times$ Female					-0.021	-0.068
					(0.064)	(0.149)
City FEs	Yes	Yes	Yes	Yes	Yes	Yes
Institution FEs	Yes	No	No	No	No	No
Observations	3,625	2,297	3,051	3,051	3051	2,284

*Notes:* This table presents our analysis of labor market outcomes in Section 4.8. 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.

**Table 6: Blinding Experiment — Main Results Gender Gaps**

	Coding subjective rating		Unit tests prediction		Interview prediction	
	(1)	(2)	(3)	(4)	(5)	(6)
Female code	0.029 (0.058)	0.028 (0.058)	0.202 (0.178)	0.217 (0.180)	0.028 (0.050)	0.027 (0.050)
Non-blind code	-0.082 (0.058)	-0.085 (0.058)	-0.284 (0.188)	-0.269 (0.189)	-0.158** (0.051)	-0.056 (0.050)
Non-blind code × Female code	0.046 (0.083)	0.057 (0.083)	0.209 (0.255)	0.219 (0.257)	0.039 (0.069)	0.035 (0.069)
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	2,323	2,323	2,323	2,704	2,704

*Notes:* This table provides results from Experiment II (see Section 5), testing pre-registered Hypothesis H2. The regression specification is as described in Equation (6). The even columns include evaluator fixed effects, while the odd columns do not. Standard errors are clustered at the evaluator level.

**Table 7: Blinding Experiment — Main Results On Racial Gaps**

	Subjective Coding Ratings					
	Reviewer 1		Reviewer 2		Algorithmic Prediction	
	(1)	(2)	(3)	(4)	(5)	(6)
Non-blind × Male × Non White/Non East Asian	-0.117 (0.079)	-0.152* (0.085)	-0.148* (0.077)	-0.162** (0.081)	-0.166* (0.085)	-0.172* (0.094)
Non-blind × Female × White/East Asian	-0.068 (0.070)	-0.036 (0.075)	-0.009 (0.082)	0.017 (0.090)	-0.030 (0.066)	0.003 (0.070)
Non-blind × Female × Non White/Non East Asian	0.013 (0.083)	-0.009 (0.089)	-0.053 (0.071)	-0.055 (0.074)	-0.045 (0.089)	-0.077 (0.096)
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	2,323	2,292

*Notes:* This table provides results from Experiment II (see Section 5). In this case, we investigate gender and racial disparities in final ratings. The omitted racial category is white or East Asian men. The even columns include evaluator fixed effects. Standard errors are clustered at the evaluator level.

(For Online Publication)

Appendix to

# Does Better Information Reduce Gender Discrimination in the Technology Industry?

Abdelrahman Amer, Ashley C. Craig and Clémentine Van Effenterre

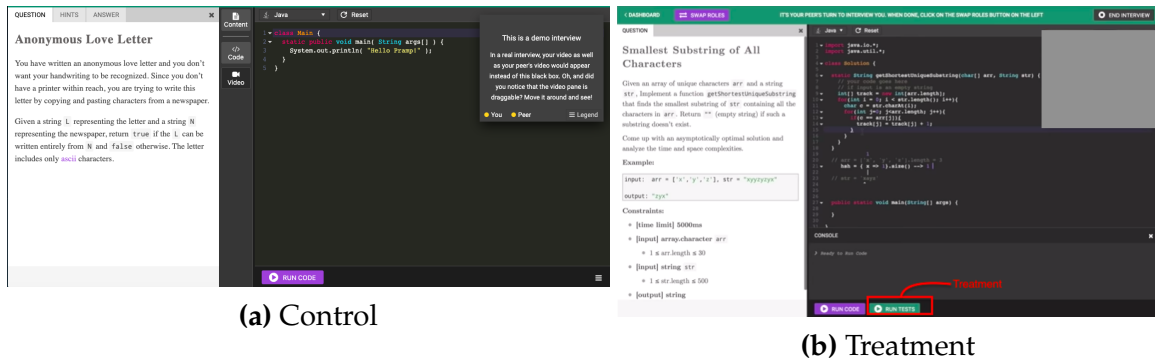
August 2023

## List of Appendices

<b>Appendix A: Institutional details</b>	<b>A-2</b>
<b>Appendix B: Additional Results</b>	<b>A-5</b>
<b>Appendix C: Follow-up Experiment</b>	<b>A-18</b>
<b>Appendix D: Questionnaire</b>	<b>A-32</b>

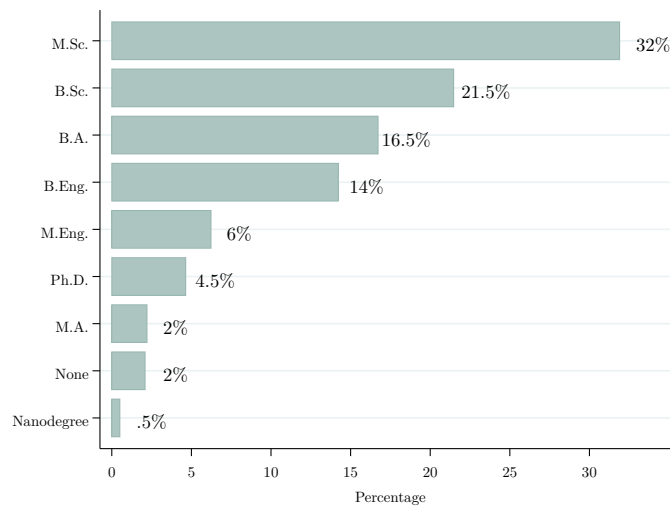
# Appendix A Institutional details

Figure A1: Environment of the platform and treatment



Notes: Figure A1(a) presents the website layout for a mock interview on the platform in the control condition. Figure A1(b) represents the treatment condition.

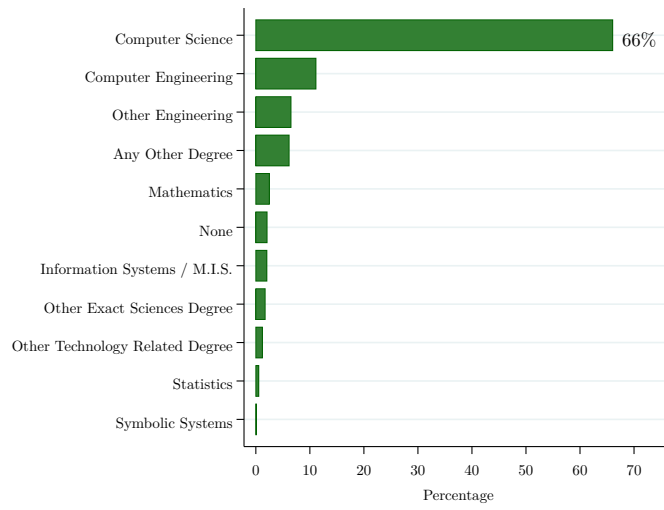
Figure A2: Users' level of education



Notes: The figure presents the average level of education of users.

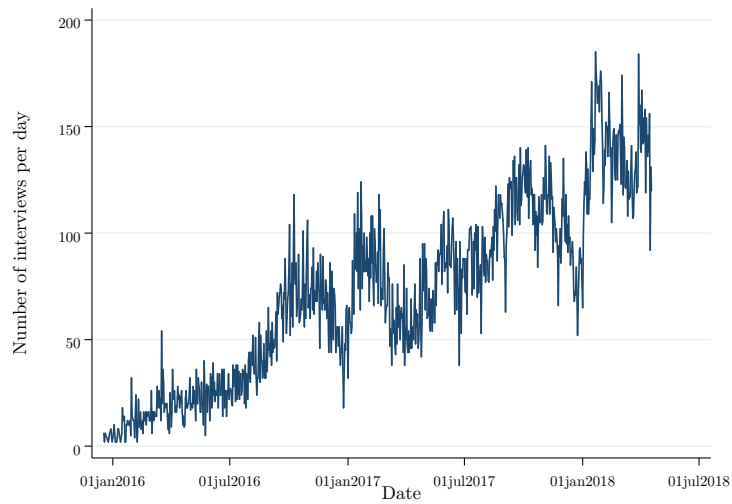


**Figure A3: Users' field of education**



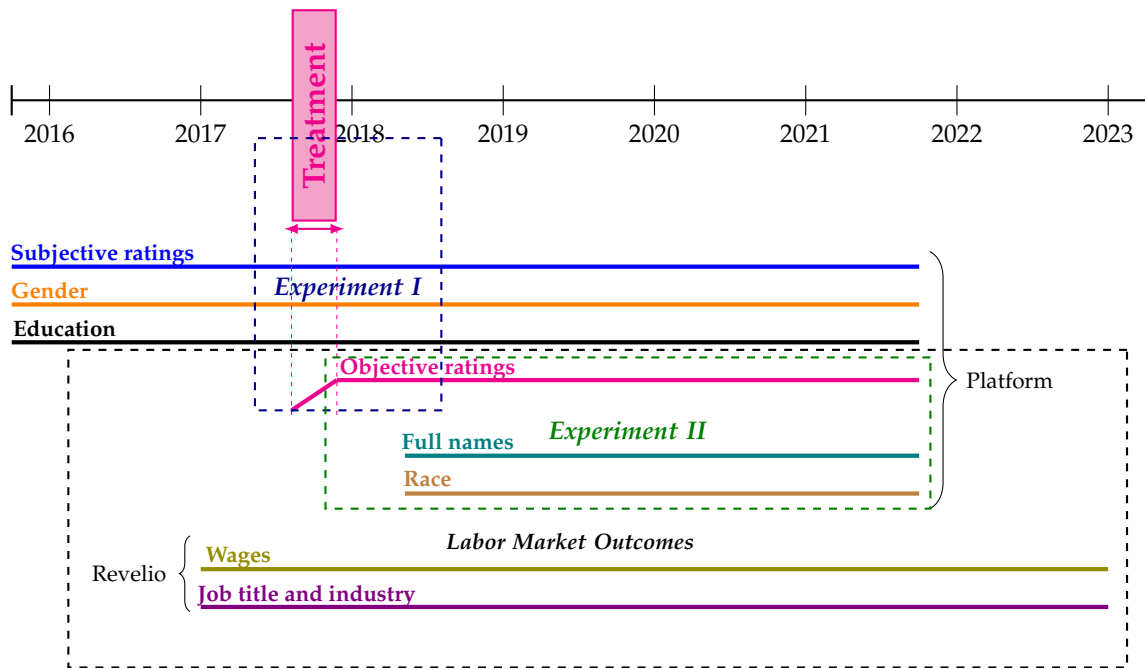
Notes: The figure presents the field of education of users.

**Figure A4: Growth of the platform**



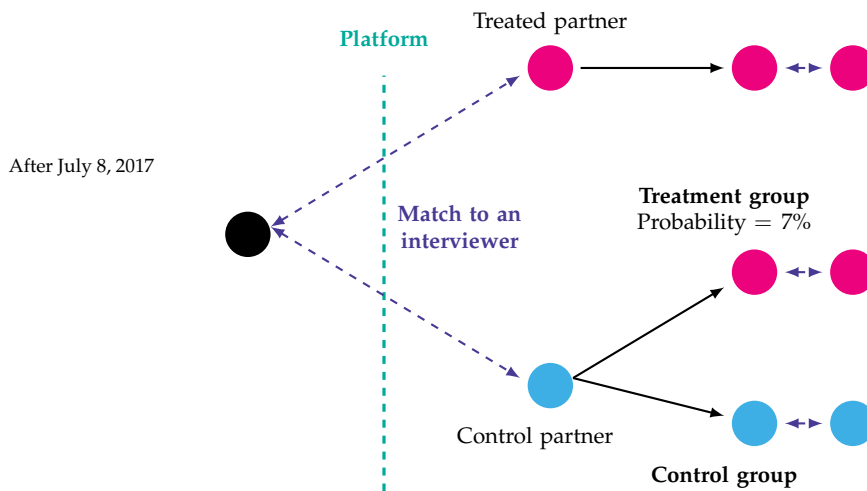
Notes: The figure shows the evolution of the number of users on the platform from January 2016 to January 2018.

**Figure A5: 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.

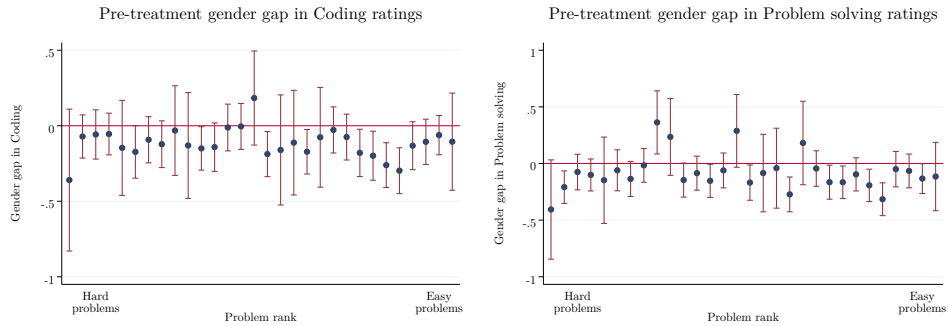
**Figure A6: Treatment assignment**



Notes: This diagram shows how users are assigned to the treatment or to the control conditions when then enter the platform.

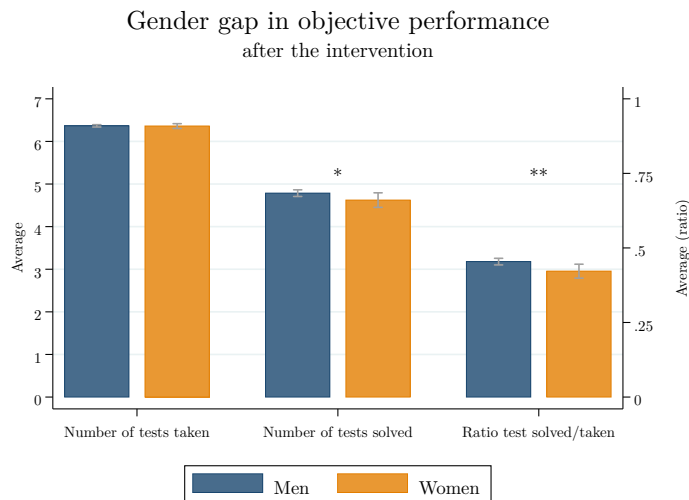
# Appendix B Additional Results

**Figure B7: Pre-treatment gender gaps by problem difficulty**



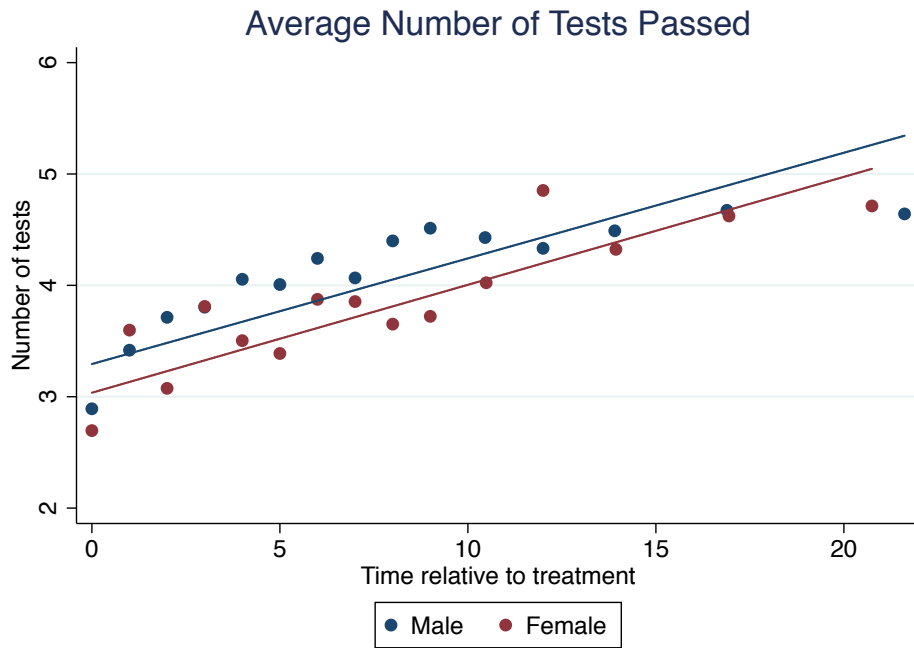
*Notes:* This figure plots gender gaps in subjective ratings for coding and problem solving by problem difficulty in the pre-intervention period. Problem difficulty is computed using the average objective performance of users in the post-intervention period.

**Figure B8: Gender gap in objective performance after the intervention**

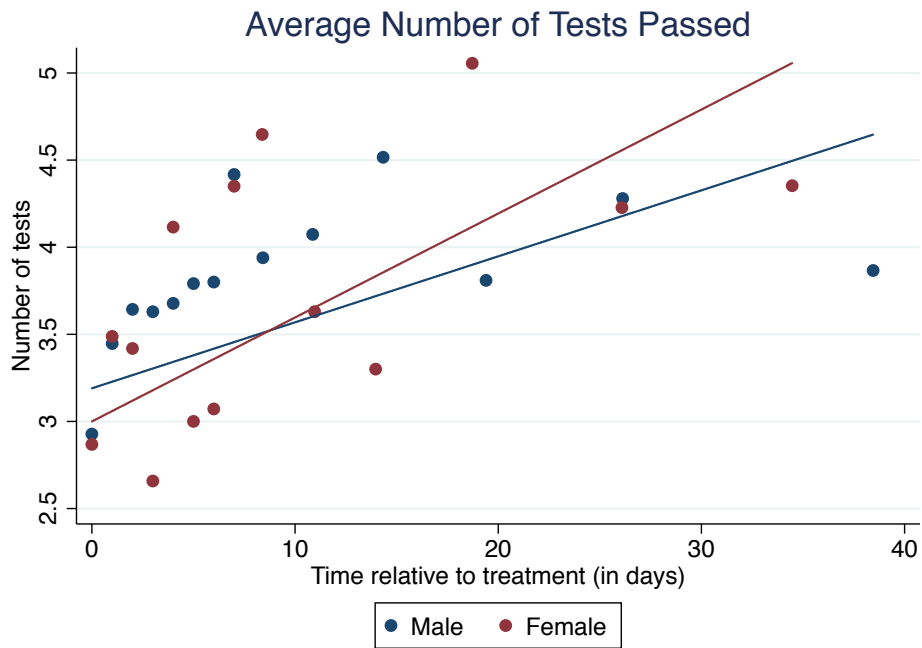


*Notes:* This figure presents the gender gap in objective performance after the intervention in terms of number of tests taken, number of tests solved or failed (right y-axis), and the ratio test solved/passed (right y-axis).

**Figure B9: Gender differences in learning**



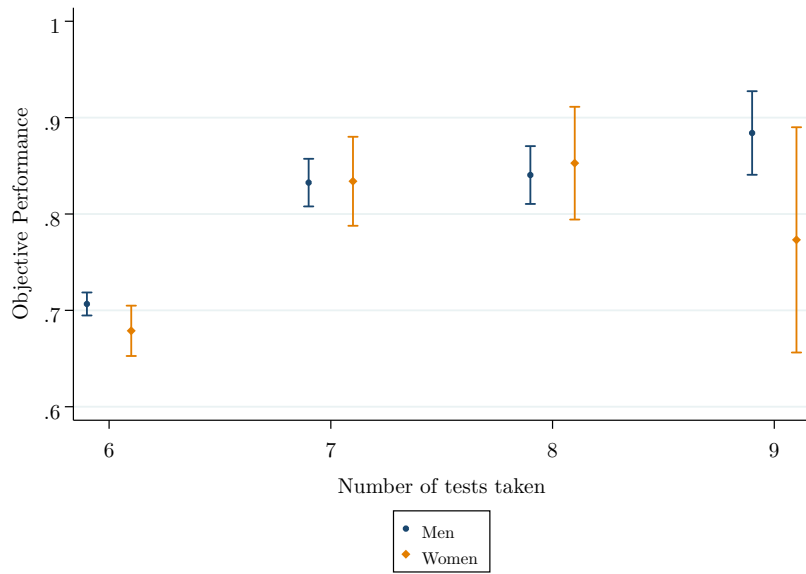
**(a) Time relative treatment (in sessions)**



**(b) Time relative treatment (in days)**

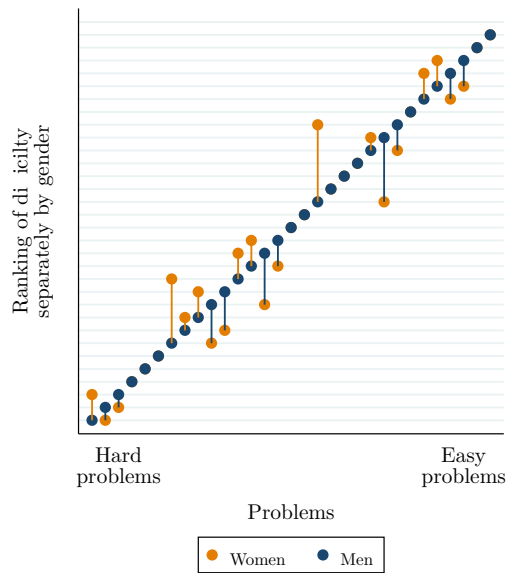
*Notes:* This figure shows the evolution over time in days (Panel A) and over sessions (Panel B) of the objective coding performance (number of tests completed) of male and female users.

**Figure B10: 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 B11: Ranking of problems by gender**



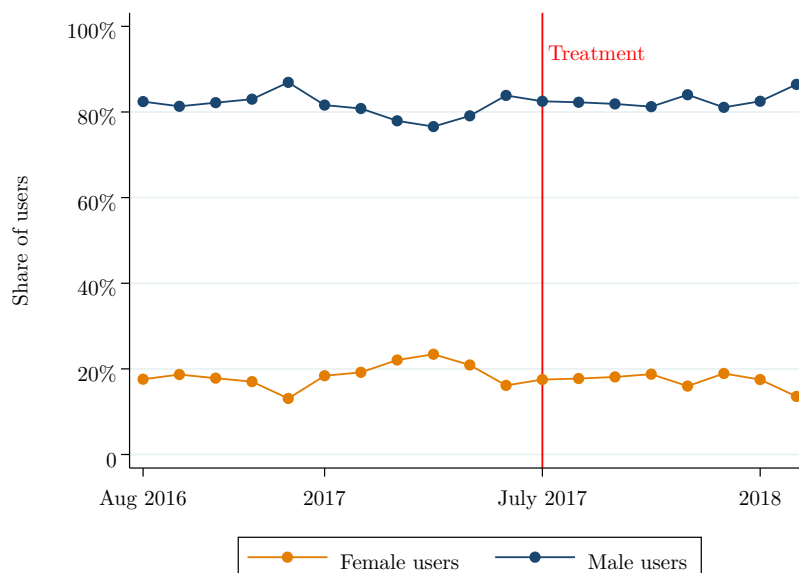
*Notes:* This figure shows the relative ranking of problems' difficulty by gender. The ranking is proxied by the average performance of users for each problem. The orange vertical lines show any positive or negative deviation of female users' ranking compared to male users' ranking.

**Table B1:** Problems' and Evaluators' Characteristics

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

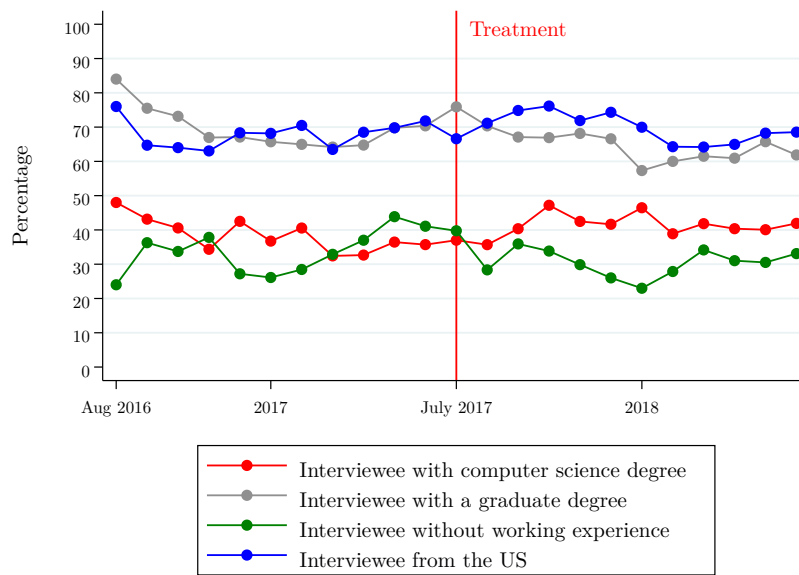
Notes: The regression TBC

**Figure B12:** Share of male and female users over time



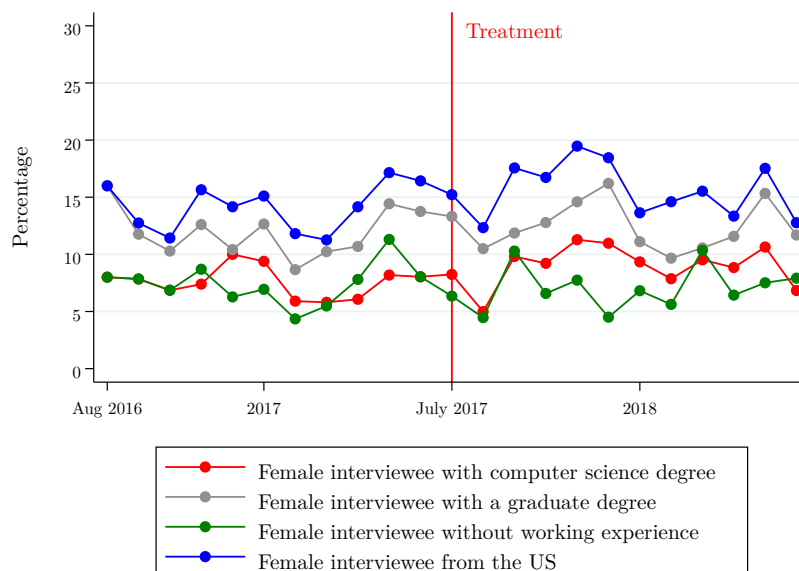
Notes: This figure shows the evolution of the shares of female and male users on the platform before and after the introduction of the device.

**Figure B13: Evolution of First-Time Users' Characteristics**



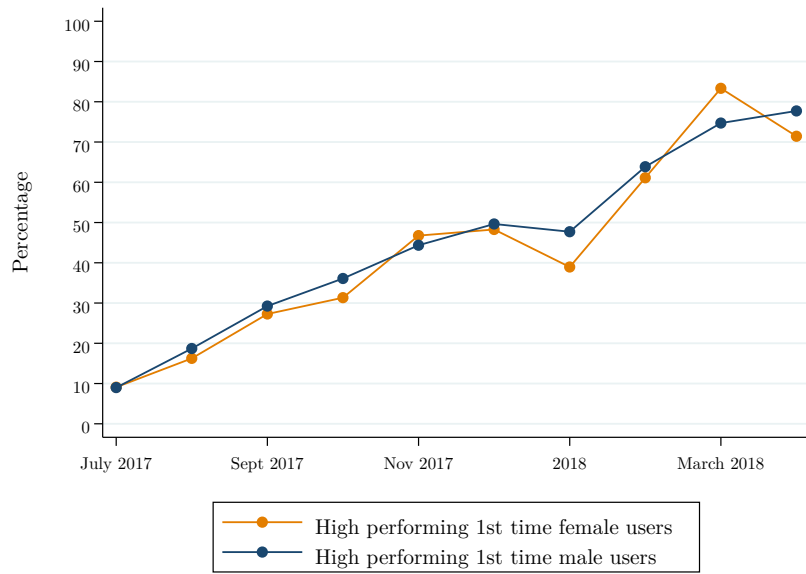
Notes: The figure presents the evolution of first-time users' characteristics averaged by month around the date of the introduction of the device on the platform.

**Figure B14: 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 of the introduction of the device on the platform.

**Figure B15: 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 introduction of the device on the platform. High-performing users are defined as those passing all unit tests taken for a given problem.



**Table B2: Subjective Ratings Pre-Intervention**

---

*Panel A: All*

<b>Variable</b>	<b>Mean</b>	<b>Std. Dev.</b>	<b>Min.</b>	<b>Max.</b>	<b>N</b>
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*

<b>Variable</b>	<b>Mean</b>	<b>Std. Dev.</b>	<b>Min.</b>	<b>Max.</b>	<b>N</b>
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

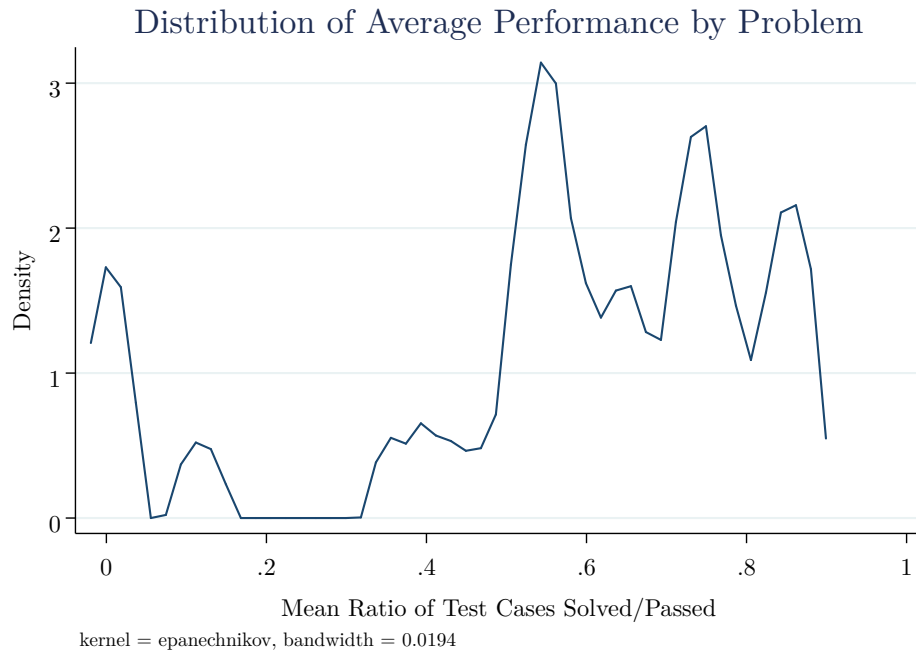
---

*Panel C: Men*

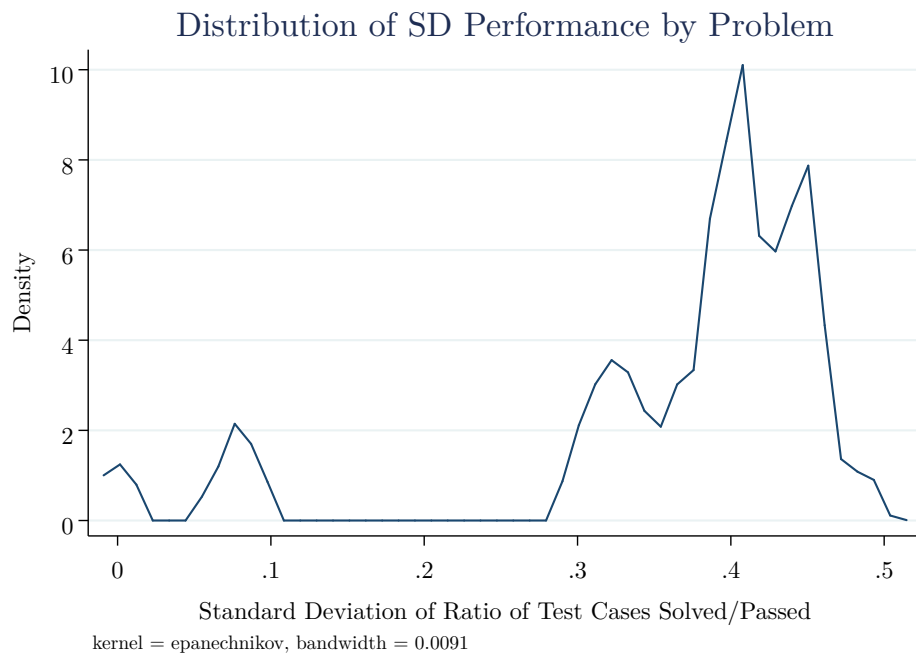
<b>Variable</b>	<b>Mean</b>	<b>Std. Dev.</b>	<b>Min.</b>	<b>Max.</b>	<b>N</b>
Score in coding	-0.026	1.003	-2.981	1.12	21,575
Score in problem solving	-0.024	0.982	-2.62	1.264	21,575
Score in likability	0.083	0.93	-2.738	1.095	21,575
Score in communication	-0.055	0.996	-3.413	1.042	21,575
Score in hireability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

---

**Figure B16: Variations across Problems**



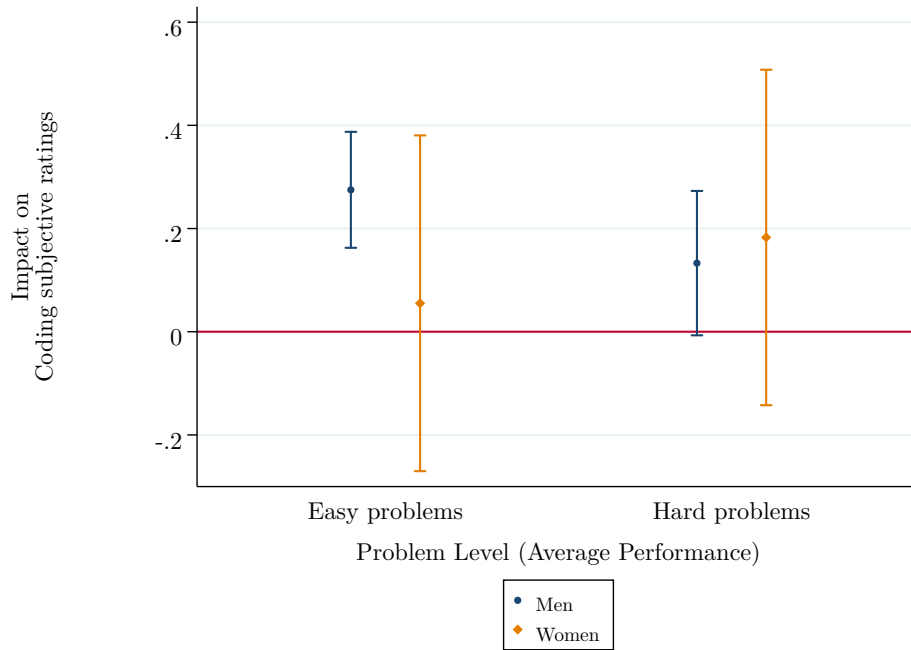
**(a) Problem Average Difficulty**



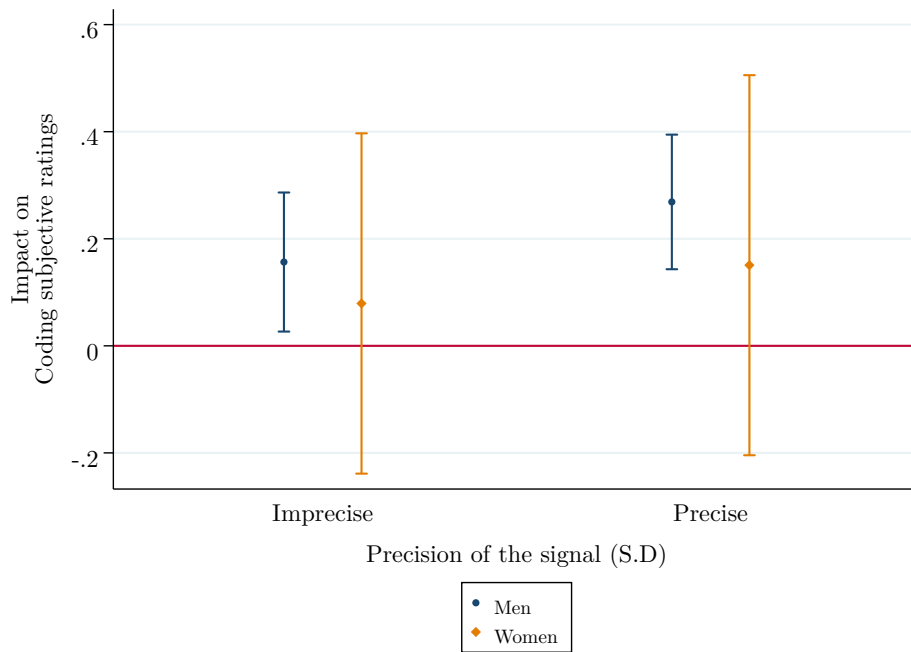
**(b) Precision of the Signal**

*Notes:* This figure shows the distribution of average performance by Problem (Panel A) and the distribution of standard deviation by problem (Panel B) measured by the mean and standard deviation the objective coding performance (ratio of tests completed over tests passed).

**Figure B17: Men's and Women's Treatment Effects on Subjective Rating by Problem**



**(a) Problem Average Difficulty**



**(b) Precision of the Signal**

*Notes:* This figure shows the estimates of Equation (8) where the dependent variable is the subjective rating in coding, separately by problem type and gender.

**Table B3: 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
<i>N</i>	1,587	10,004		
Test of joint significance	<i>F</i> -stat: 1.100 ( <i>p</i> -value: 0.377)			

**Table B4: Baseline characteristics**

	First Stage	Sample mean	Compliers		Never-takers
	(1)	(2)	(3)	(4)	(5)
			Treated	Untreated	
Interviewee female	0.678*** (0.015)	0.186	0.177 (0.007)	0.166 (0.016)	0.212 (0.008)
Country: USA	0.718*** (0.010)	0.684	0.681 (0.008)	0.684 (0.021)	0.693 (0.010)
Interviewee's deg.: computer science	0.709*** (0.011)	0.652	0.660 (0.008)	0.649 (0.021)	0.663 (0.009)
Interviewee's deg.: postgraduate	0.726*** (0.011)	0.431	0.434 (0.008)	0.450 (0.021)	0.424 (0.009)
Interviewee's years of experience	0.736*** (0.021)	3.067	3.061 (0.045)	2.859 (0.159)	3.225 (0.062)
Interviewee Preparation Level (self-declared on 1-5 scale)	0.621*** (0.049)	2.880	2.928 (0.013)	2.768 (0.034)	2.816 (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 B5: Robustness Checks**

	Coding	Problem solving	Likeability	Communication	Hireability
<i>Panel A: Baseline</i>					
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
<i>Panel B: with Month FE</i>					
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
<i>Panel C: with Controls</i>					
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
<i>Panel D: no Date FE</i>					
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
<i>Panel E: Including pre-treatment period</i>					
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
<i>Panel F: Difference-in-Difference</i>					
Treatment	0.131***	0.199***	0.075**	0.160***	0.143***
S.E	0.029	0.029	0.035	0.041	0.030
Treatment*Woman	-0.070	-0.008	-0.047	0.022	-0.010
S.E	0.062	0.056	0.076	0.063	0.070
N	54077	54077	54077	54077	51533
<i>Panel G: Controlling for Propensity Score Matching</i>					
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
<i>Panel H: with Individual FE</i>					
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 implement a difference-in-differences with month-of-interview fixed effects. In Panel G, we control for propensity score matching. In Panel H, we control for interviewee fixed effects. Standard errors are clustered at the date level.

**Table B6:** Balancing test by problem difficulty – whole sample

Variables	Hard	Easy	Difference	P-value
Interviewee female	0.173	0.176	0.003	0.583
Interviewer female	0.175	0.173	-0.002	0.625
Gender interviewer missing	0.079	0.073	-0.006	0.057
Country: USA	0.699	0.702	0.003	0.556
Interviewee's deg.: computer science	0.641	0.639	-0.001	0.818
Interviewer's deg.: computer science	0.642	0.636	-0.006	0.370
Interviewer's deg.: postgraduate	0.477	0.469	-0.007	0.302
Interviewee's deg.: postgraduate	0.471	0.471	-0.000	0.978
Interviewee's years of experience	3.230	3.286	0.056	0.186
Interviewer's years of experience	3.321	3.193	-0.128	0.002
<i>N</i>	11,984	12,080		
Test of joint significance	<i>F</i> -stat: 1.800 ( <i>p</i> -value: 0.078)			

**Table B7:** Gender gap in Subjective Coding Ratings and Interviewer's Experience on the Platform

	Subjective Coding Ratings			
	(1)	(2)	(3)	(4)
Interviewee female	-0.081*** (0.018)	-0.081*** (0.018)	-0.084*** (0.021)	-0.0757*** (0.021)
Interviewer's total # of sessions	Yes			
Interviewer's # of past sessions		Yes		
Interviewer's total # of female interviewees			Yes	
Past top female performer				Yes
Objective performance	Yes	Yes	Yes	Yes
Interviewer gender	Yes	Yes	Yes	Yes
Interviewee's sociodemographic controls	Yes	Yes	Yes	Yes
Interviewer's sociodemographic controls	Yes	Yes	Yes	Yes
Date FE	No	No	No	Yes
Observations	19,551	19,551	14,677	13,541

*Notes:* This table shows the estimation of the gender gap in subjective ratings, controlling for objective performance measure (proxied by the ratio of test solved over passed by problem), using a linear regression model in which we progressively add controls. In column 1, we add a control for the interviewer's total number of sessions, in column 2 we control for the number of previous sessions, in column 3 control for the interviewer's total number of sessions with a female user, and in column 4 we control for whether the interviewer faced a top female performer during the previous session. All specifications include controls for interviewer's and interviewee's years of experience, a dummy variable for each level area of education and highest educational level and for the gender of the interviewer, problem fixed-effects and date-of-interview fixed effects.

**Table B8: Labor Market Outcomes by Gender**

	Ln(first salary post graduation)									
	Male					Female				
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
Objective Performance	0.057 (0.035)	0.053 (0.033)	0.052 (0.033)	0.050 (0.035)	0.050 (0.035)	0.023 (0.047)	0.021 (0.051)	0.014 (0.051)	0.013 (0.055)	0.013 (0.054)
Subjective Coding Rating		0.010 (0.037)	-0.019 (0.049)	-0.035 (0.070)	-0.036 (0.071)		0.039 (0.086)	-0.040 (0.098)	-0.052 (0.083)	-0.046 (0.084)
Communication Rating			0.052 (0.045)	0.050 (0.042)	0.048 (0.042)			0.139* (0.080)	0.136 (0.087)	0.158* (0.082)
Prob. Solv Rating				0.023 (0.053)	0.022 (0.053)				0.018 (0.117)	0.023 (0.124)
Collab. Rating					0.005 (0.029)					-0.048 (0.080)
Sociodemographic controls	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
City FEs	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Observations	1,700	1,691	1,691	1,691	1,691	535	531	531	531	531

*Notes:* This table presents Mincer-type regression where the dependent variable is the (log) first salary post graduation observations from participants of the platform data matched with the Revelio Lab database, separately for men and women. 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.

## Appendix C Follow-up Experiment

### C.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 discloses that we are studying how evaluators judge the performance of software developers but does not explicitly mention gender.

**Sample** To construct the sample of code blocks, we leverage the more recent dataset obtained from the platform we partnered with, spanning observations from April 2018 to May 2021. Like our previous dataset, this dataset contains the subjective ratings and objective measure of coding quality. From this sample, we use first names to identify gender using predictions from genderize.io. This leaves us with 38,322 session-participant pairs, and 10,380 unique participants.<sup>A.1</sup> Of these, 21 percent are probabilistically identified as female. A novel feature of our dataset is that we can link this information to the code blocks written by each participant in each session. Our final sample is stratified by gender, race, and coding performance.

**Randomization** Let  $N$  be the number of evaluators and  $P$  the number of problems by evaluator. Our experiment is stratified by gender and performance, such that  $\frac{P}{2}$  code blocks are written by women, among which  $\frac{P}{4}$  are high-score codes according to the platform objective device. Each evaluator  $i$  is assigned a set of  $P$  problems in a random order. We use a within-subject design. We define  $NB_j = 0$  for a blind problem  $j$  (if the gender of the coder is not revealed),  $NB_j = 1$  for a non-blind problem  $j$  (if the gender of the coder is revealed). For each evaluator  $i$ , the gender of the coder will be revealed for half of the problems. To account for potential priming effect, we plan to randomize whether the gender of the coder is revealed in the first or in the second half of the study:

1. For half of evaluators, problems will be blind, then non-blind.

---

<sup>A.1</sup>See Table C11.



$$\forall i = 1, \dots, \frac{N}{2} \begin{cases} \text{for } j = 1, \dots, \frac{P}{2} & , NB_{ij} = 0 \\ \text{for } j = \frac{P}{2}, \dots, P & , NB_{ij} = 1 \end{cases}$$

2. For the other half, problems will be non-blind, then blind.

$$\forall i = \frac{N}{2}, \dots, N \begin{cases} \text{for } j = 1, \dots, \frac{P}{2} & , NB_{ij} = 1, \\ \text{for } j = \frac{P}{2}, \dots, P & , NB_{ij} = 0 \end{cases}$$

**Testing the salience of the main treatment** In the piloting phase of the experiment, we asked a random sample of online participants ("evaluator") on Prolific to predict the gender of a participant ("worker") after evaluating a task they completed, mimicking the lay-out of the first name and avatar of our main experiment. While a non-trivial fraction of "evaluators" didn't 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 ask them to predict the potential performance of three different hypothetical coders. We cross-randomize the first name (alternating gender) and the skill level for each vignette. The vignette 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?"*

Skills

First names

*a M.Sc in computer science and has 2 years of work experience*

Katie/Tom

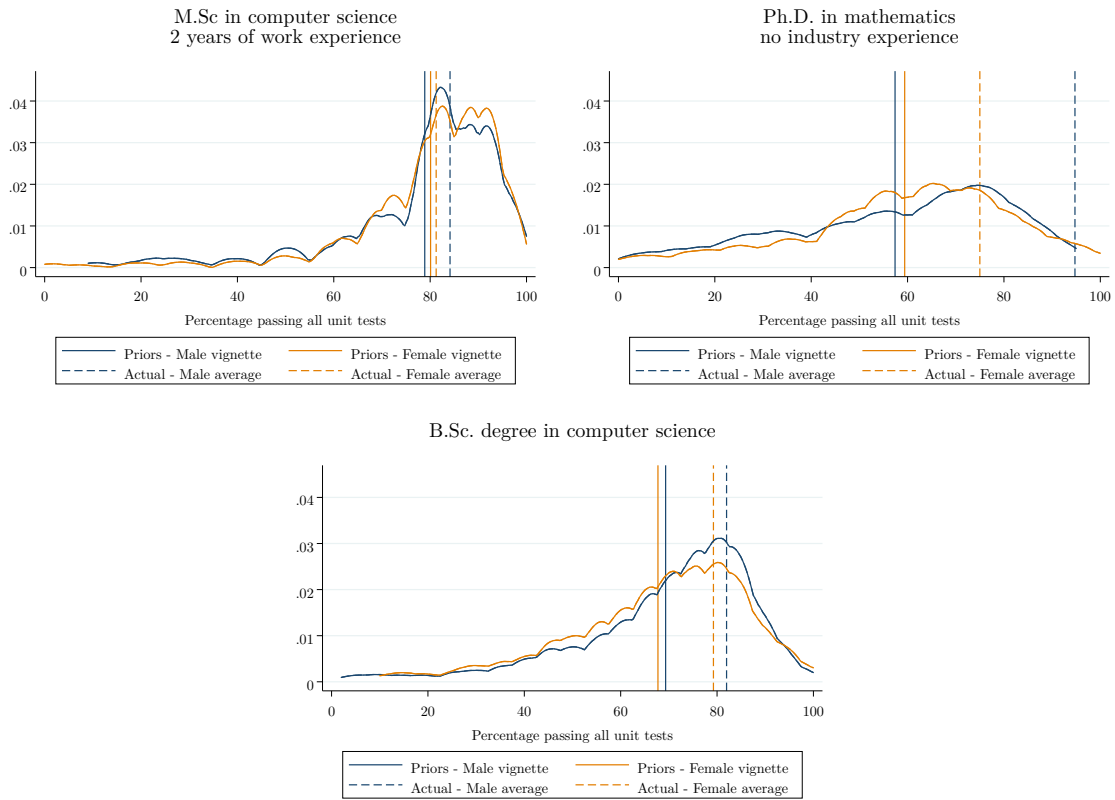
*a Ph.D. in mathematics and has no industry experience*

Alexa/Mickael

*a B.Sc. degree in computer science*

Corinne/Matt

**Figure C18: Respondents' Priors Beliefs about Performance by Gender**



*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 C19: Example of Code — 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 'sortKMessdArray' 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:**
--- prompt
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__
    
```

```

function sortKMessdArray(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);
  }
}

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

(a) Question

(b) Answer

```

describe("Solution", function() {

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

(c) Tests

Notes: This figure presents an example of code excerpt that will be used in the experiment. Panel A displays the question, Panel B the written code block, and Panel C the series of unit tests that generate the objective measure of performance.

# Figure C20

AbdoAmer98 2023-03-12

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 "beat" and "bit" is 3:

- By deleting 'e' and 'a' in "beat", and 'i' in "bit", we get the string "bt" 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) )
```

1/2

**(a) Non-Blind Male**

AbdoAmer98 2023-03-12

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 "beat" and "bit" is 3:

- By deleting 'e' and 'a' in "beat", and 'i' in "bit", we get the string "bt" 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) )
```

1/2

**(b) Blind Male**

AbdoAmer98 2023-03-12

Question Assigned to **Eve M.** 

**Coding Language Used:** Python

**Question Name:** Pancake-Sort

**Description:** Given an array of integers `arr`:

- Write a function `flip(arr, k)` that reverses the order of the first `k` elements in the array `arr`.
- 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] + flipped_arr
        i += 1
    return flip(arr, len(arr))
```

1/2

**(c) Non-Blind Female**

AbdoAmer98 2023-03-12

Question Assigned to **E M.** 

**Coding Language Used:** Python

**Question Name:** Pancake-Sort

**Description:** Given an array of integers `arr`:

- Write a function `flip(arr, k)` that reverses the order of the first `k` elements in the array `arr`.
- 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] + flipped_arr
        i += 1
    return flip(arr, len(arr))
```

1/2

**(d) Blind Female**

Notes: This figure presents an example of code in the blind and non-blind conditions for both male and female coders.

## C.2 Descriptive Statistics: Sample of Codes

Table C9: Descriptive Statistics — Follow-up Experiment

	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	42.24%	56.47%	100%
Share of Python scripts	29.76%	37.29%	43.10%
Share of Java scripts	35.14%	34.91%	44.72%
Share of C++ scripts	16.89%	9.22%	12.16%

Note: 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 USA only.

**Table C10: Descriptive Statistics — Sample Construction — January 2018-May 2022**

	Obs.	Mean	Median	S.D
<b>Raw Data</b>				
Full score	203,769	0.34	1.00	0.39
Num code lines	482,390	44.12	40.00	37.45
Female	-	-	-	-
Non-white	-	-	-	-
<b>Clean Data</b>				
Full score	100,933	0.81	1.00	0.40
Num code lines	178,717	55.25	48.00	31.89
Female	-	-	-	-
Non-white	-	-	-	-
<b>Experimental Sample</b>				
Full score	38,322	0.82	1.00	0.38
Num code lines	38,322	45.18	44.00	13.55
Num code lines - male	31,245	45.23	44.00	13.63
Num code lines - female	7,077	44.97	44.00	13.17
Female	38,322	0.18	0.00	0.39
Non-white	38,322	0.61	1.00	0.49

**Table C11:** Descriptive Statistics — Sessions used for the Experimental Data — April 2018-May 2021

Number of sessions	38,322
Number of interviewees	10,380
Number of interviewers	18,339
Number of problems	38
Share of female interviewees	21.43

*Panel A: All*

<b>Variable</b>	<b>Mean</b>	<b>Std. Dev.</b>	<b>Min.</b>	<b>Max.</b>	<b>N</b>
Interviewee's deg.: computer science	0.703	0.457	0	1	10,196
Interviewee without working experience	0.285	0.451	0	1	10,380
Interviewee with a graduate degree	0.497	0.500	0	1	10,197
Interviewee Preparation Level	2.997	0.873	1	5	10,378

*Panel B: Women*

<b>Variable</b>	<b>Mean</b>	<b>Std. Dev.</b>	<b>Min.</b>	<b>Max.</b>	<b>N</b>
Interviewee's deg.: computer science	0.696	0.460	0	1	2,207
Interviewee without working experience	0.330	0.470	0	1	2,225
Interviewee with a graduate degree	0.588	0.492	0	1	2,207
Interviewee Preparation Level	2.805	0.849	1	5	2,225

*Panel C: Men*

<b>Variable</b>	<b>Mean</b>	<b>Std. Dev.</b>	<b>Min.</b>	<b>Max.</b>	<b>N</b>
Interviewee's deg.: computer science	0.705	0.456	0	1	7,989
Interviewee without working experience	0.273	0.445	0	1	8,155
Interviewee with a graduate degree	0.472	0.499	0	1	7,990
Interviewee Preparation Level	3.050	0.872	1	5	8,153

*Notes:* This table shows descriptive statistics for the sample of interviews from April 2018 to May 2021 that we use for Experiment II in Section 5. 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 C12:** Gender gap in subjective coding ratings, controlling for objective performance — Experimental Sample

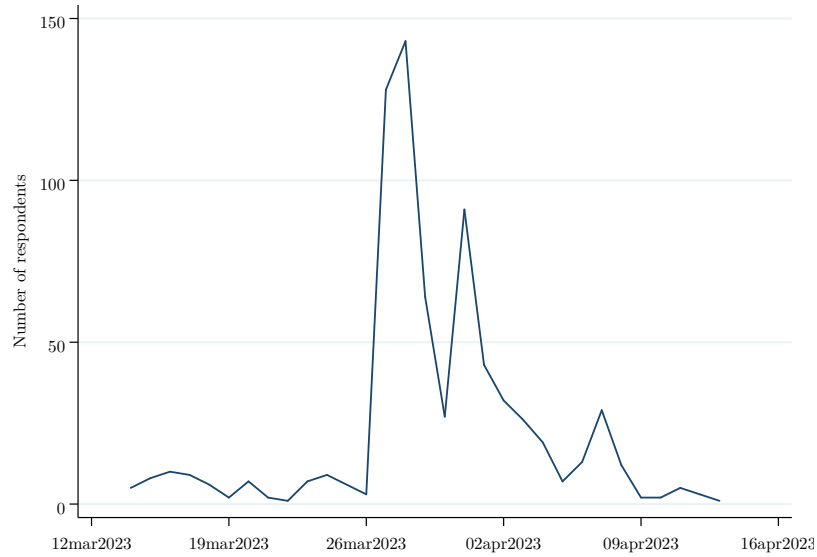
	Subjective Coding Ratings			
	(1)	(2)	(3)	(4)
Interviewee female	-0.123 *** (0.0131)	-0.108*** (0.0127)	-0.126*** (0.0161)	-0.126 *** (.0160)
Objective performance		1.092*** (0.0216)	1.157*** (0.0290)	1.155*** (0.0303)
Interviewer's FE	No	No	Yes	Yes
Problem FE	No	No	No	Yes
Observations	38,322	38,322	38,322	38,322

Note: This table shows the estimation of the gender gap in subjective ratings, controlling for objective performance measure (proxied by the ratio of test solved over passed by problem), using a linear regression model in which we progressively add controls. We progressively add interviewer and problem fixed effects.

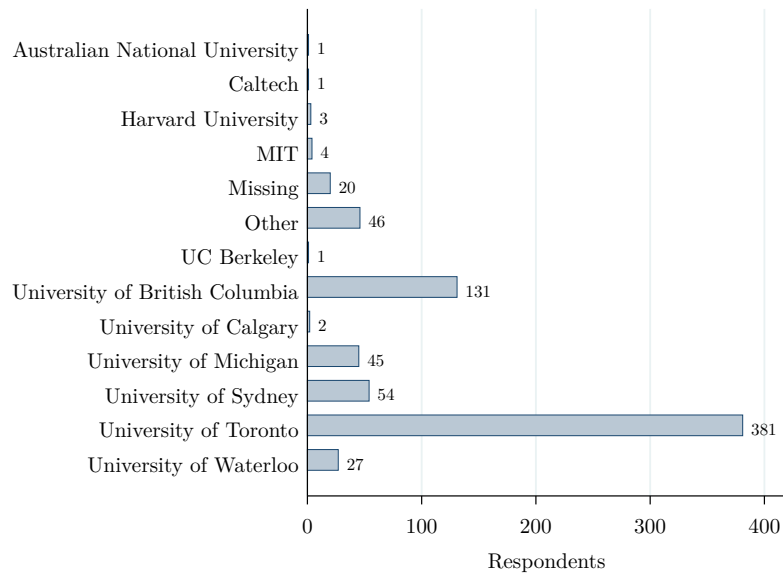


### C.3 Descriptive Statistics: Evaluators

Figure C21: Number of respondents over time



(a) Number of respondents



(b) Number of respondents by institutions

**Table C13: Descriptive Statistics — Participants**

	Mean	Std. Dev.	N
<b>Gender</b>			
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
<b>Recoded race</b>			
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
<b>Current situation</b>			
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
<b>Highest degree completed</b>			
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
<b>Experience with Python</b>			
Basic	0.221	0.415	707
Intermediate	0.448	0.498	707
Advanced	0.331	0.471	707
<b>Experience with Java</b>			
Basic	0.536	0.499	676
Intermediate	0.361	0.481	676
Advanced	0.104	0.305	676
<b>Experience with C++</b>			
Basic	0.643	0.479	673
Intermediate	0.272	0.445	673
Advanced	0.085	0.279	673
<b>Preferred language</b>			
C++	0.089	0.285	716
Java	0.141	0.348	716
Python	0.77	0.421	716

**Table C14: Treatment-Control Balance — Whole sample**

	Non-blind to Blind (1)	Blind to Non-blind (2)	Difference (3)	<i>p</i> -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

*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 C15: Treatment-Control Balance — Quality sample**

	Non-blind to Blind (1)	Blind to Non-blind (2)	Difference (3)	<i>p</i> -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

*Notes:* This table presents balancing checks for the quality 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 C16: Blinding Experiment — Main Results whole sample**

	Coding subjective rating		Unit tests prediction		Interview prediction	
	(1)	(2)	(3)	(4)	(5)	(6)
Non-blind code	-0.059 (0.040)	-0.057 (0.040)	-0.181 (0.131)	-0.161 (0.132)	-0.139*** (0.037)	-0.039 (0.036)
Treatment order	-0.001 (0.041)		0.056 (0.138)		-0.115** (0.044)	
Script 1	-0.262*** (0.057)	-0.260*** (0.058)	-0.368* (0.179)	-0.346 (0.183)	-0.156** (0.051)	-0.012 (0.051)
Script 2	-0.098 (0.058)	-0.092 (0.058)	-0.206 (0.181)	-0.170 (0.181)	-0.199*** (0.049)	-0.054 (0.048)
Script 3	-0.022 (0.059)	-0.019 (0.059)	-0.267 (0.184)	-0.276 (0.185)	-0.080 (0.050)	-0.067 (0.050)
Evaluator FE	No	Yes	No	Yes	No	Yes
Problem FE	Yes	Yes	Yes	Yes	Yes	Yes
Observations	2,323	2,292	2,323	2,292	2,704	2,704

Notes: This table provides a test for H1 for the whole sample. The even columns include evaluator fixed effects. Standard errors are clustered at the evaluator level.

**Table C17: Blinding Experiment — Main Results quality sample**

	Coding subjective rating		Unit tests prediction		Interview prediction	
	(1)	(2)	(3)	(4)	(5)	(6)
Non-blind code	-0.071 (0.045)	-0.065 (0.045)	-0.230 (0.150)	-0.201 (0.151)	-0.078 (0.044)	-0.038 (0.043)
Treatment order	-0.013 (0.046)		0.094 (0.153)		-0.030 (0.045)	
Script 1	-0.242*** (0.063)	-0.239*** (0.064)	-0.350 (0.204)	-0.327 (0.207)	-0.121 (0.062)	-0.061 (0.062)
Script 2	-0.090 (0.065)	-0.080 (0.065)	-0.184 (0.204)	-0.153 (0.204)	-0.129* (0.059)	-0.064 (0.059)
Script 3	0.019 (0.066)	0.022 (0.067)	-0.164 (0.209)	-0.175 (0.209)	-0.046 (0.059)	-0.037 (0.059)
Evaluator FE	No	Yes	No	Yes	No	Yes
Problem FE	Yes	Yes	Yes	Yes	Yes	Yes
Observations	1,852	1,835	1,852	1,835	1,946	1,946

Notes: This table provides a test for H1 for the quality sample. The even columns include evaluator fixed effects. Standard errors are clustered at the evaluator level.

**Table C18: 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: basic	0.138	0.345	716
Evaluated all codes	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

**Table C19: Racial Gap in Subjective Coding Ratings, Controlling for Objective Performance**

<i>Panel A</i>	Subjective Coding Ratings					
	Whole sample		Male coders		Female coder	
	(1)	(2)	(3)	(4)	(5)	(6)
White or East Asian	0.074*** (0.010)	0.067*** (0.018)	0.073*** (0.011)	0.065*** (0.021)	0.072*** (0.024)	0.039 (0.108)
Objective performance	Yes	Yes	Yes	Yes	Yes	Yes
Evaluator FE	No	Yes	No	Yes	No	Yes
Problem FE	Yes	Yes	Yes	Yes	Yes	Yes
Observations	38,322	38,322	31,245	31,245	7,077	7,077
<i>Panel B</i>	Whole sample					
	(1)	(2)				
White or East Asian	0.072*** (0.011)	0.066*** (0.020)				
Female	-0.122*** (0.033)	-0.109*** (0.019)				
White or East Asian × Female	0.002 (0.027)	-0.006 (0.047)				
Objective performance	Yes	Yes				
Evaluator FE	No	Yes				
Problem FE	Yes	Yes				
Observations	38,322	38,322				

*Notes:* This table provides descriptive evidence of racial gaps in ratings for in-person interviews on the platform, controlling for objective performance and problem fixed effects. The even columns include evaluator fixed effects, with standard errors clustered at the evaluator level.

# Appendix D: 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](mailto: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](mailto: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](mailto: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](mailto: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.

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 ]

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?  
Basic-Intermediate-Advanced
  - Python
  - Java
  - C++

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?

#### 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 2 to 4 — Repeat**

*FOR PILOT ONLY* What is your prediction of the percent chance that the last candidate was a woman?

- Cursor between 0 and 100

## Follow-up questions

1. In which country do you currently reside?
  - Canada
  - USA
  - Other (choose)
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
6. On a scale of 1-4 how prepared do you believe you are able to evaluate others' code?
  - 1
  - 2
  - 3
  - 4

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

Congratulations, you completed the main portion of the experiment! Once you have completed the questionnaire, you will reach the end of the experiment and learn about your total payment.

END of Questionnaire