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

Abdelrahman Amer, Ashley C. Craig and Clémentine Van Effenterre

July 2023

Abstract

Performance evaluation matters for hiring decisions. We use a series of field experiments to show that the context in which performance evaluation occurs affects the display of bias. We focus on gender bias in evaluations of the code written by computer programmers. Leveraging 60,000 mock interviews from an online platform for software engineers, we document that women receive lower ratings for code quality than men, controlling for individual characteristics and objective measures of code quality. We then ask what drives these gaps, and whether they can be reduced by changing what the evaluators see. We explore three different conditions in which we systematically vary the amount of information about a candidate's performance presented to evaluators. In blind evaluations of the code alone, there is no gender gap. No gap is introduced when gender is revealed to the evaluator via the coder's first name. However, there are large gender gaps when participants interact by video chat, even when evaluators can see whether the code produces the correct answers to test cases—a measure that is predictive of future labor market performance. These results are hard to reconcile with traditional economic models of discrimination. They are more consistent with a form of implicit bias that arises with personal interaction.

JEL codes: J16, M51, C93
Keywords: *Discrimination*; *Gender*; *Coding*; *Experiment*.

***Amer**: University of Toronto, 150 Saint George Street, Toronto ON M5S 3G7, Canada (e-mail: abman.amer@mail.utoronto.ca) **Craig**: Australian National University, Research School of Economics, Crisp Building 2094, 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, Roland Rathelot, Alexandra Roulet, Nina Roussille, Heather Sarsons, Basit Zafar, and from participants at numerous seminars. 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 and at the University of Toronto. 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 dedicated considerable attention to the possibility that discrimination may be an important barrier to underrepresented groups in high-paying occupations (Bertrand and Duflo, 2017). Imperfect information has been seen as one way to rationalize differential treatment of members of different groups such as men and women (Phelps, 1972; Spence, 1973; Arrow, 1973). In many industries, the typical hiring process comprises several stages, each of might occur under imperfect information. Recruiters extract information about a candidate through resumes, referrals, test results, interviews or simulation assessments to observe a candidate performing a task in a realistic work context.

We use a series of field experiments to show that the context in which performance evaluation occurs affects the display of bias. We focus on gender bias in evaluation of coding performance, a common step during recruitment process in the technology industry. Bias in evaluations could have consequences for labor market outcomes even when hiring managers are not themselves biased. Rather, prior bias of this kind would show up later as "systemic" discrimination (Bohren et al., 2022), which could perpetuate the under-representation of women in the technology industry (Ashcraft et al., 2016). We elicit preferences from performance evaluation data while going beyond resume ratings, and systematically vary the information available to evaluators.

Our study's context is an online platform which lets job applicants in tech practice their interview and coding skills in person. The evaluator can see and interact with the coder. This closely mirrors real interviews for computer programmers. Female coders on the platform receive lower ratings than men, holding fixed interviewees and interviewers' level of education, experience, and preparation level. The gaps are largely independent of the gender of the evaluator. They remain even when we control for an objective measure of code quality, which indicates whether the code ran and produced the correct answers for test cases.

These persistent gender gaps in subjective ratings could be a combination of men and women writing and talking about their code differently, along with bias in evaluations. Guided by a simple model of discrimination in the spirit of Lundberg and Startz (1983), we investigate the sources of these gaps. To shed light on the underlying mechanisms, we use two field experiments which vary the amount of information evaluators see. First, we evaluate a randomized experiment conducted by the platform, which retains the in-person component but provides objective information in real time to the evaluator about the candidate's performance before their rating is chosen (whether the code runs, and produces the correct answers for test cases). Second, we remove the in-person component by asking evaluators to assess the coding performance of a candidate based solely on the code script written. Finally, in the spirit of seminal work on blind evaluations (Goldin and Rouse, 2000), we remove any information about gender.

Our first study 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 a given coding solution themselves, this would lead them to penalize women relative to men. To evaluate this, we study the the randomized roll-out by the platform of objective code quality measures that assess whether the code ran and produced correct answers for test cases. These "unit tests" were made available to the evaluator before they chose their ratings. However, the ability to better assess code quality made little difference to the gender gaps in evaluations that we see, which is consistent with evaluators' beliefs being well-calibrated.

We show that differences in objective performance correlate strongly with future labor market outcomes. Matching our platform data with Revelio Lab data, we document that a one standard deviation-increase in the average objective score measure of platform participants is associated with a 5 percent higher starting salary, an effect driven entirely by male candidates. This indicates that this objective measure of performance correlates well with labor market performance.

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 coding solutions 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* case. After ruling out the possibility that evaluators simply ignored the names they saw, we argue that this implies two things. First, 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, revealing gender does not by itself introduce bias.

The results are also hard to reconcile with the traditional concepts of taste-based and statistical discrimination. Instead, they suggest that personal interaction drives the gender gaps we see, even though the ratings 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 comes into play when personal interaction makes gender very salient. This is in line 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). Our results complement previous approaches relying on the use of IAT measures using realized choices. In line with the sociological literature, they suggest that biases are more likely to emerge when individuals are "doing gender" (West and Zimmerman, 1987) during in-person interactions, rather than when gender is signaled indirectly, an intuition 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 also conduct the same tests for racial bias. We find that coders who are not white or East Asian receive lower scores, conditional on the objective measures of code quality. Unlike for gender, however, we do find that making race visible via the first name is enough to widen the racial gap in evaluations. This suggests that more traditional taste-based or statistical discrimination may be at play, without personal interaction being a necessary precursor for bias.

We contribute to several lines of the literature. First, we contribute to a long line of work focusing 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 for such studies to separate out rational statistical discrimination, statistical discrimination with incorrect beliefs, and taste-based discrimination. A recent contribution by Bohren et al. (2019b) conceptualizes this identification problem when isolating the source of discrimination, which has been tested experimentally by Barron et al. (2022). With respect to these papers, we investigate contexts in which different types of discrimination can emerge. Finally, by providing real code excerpts to external evaluators, we attempt to minimize deception prevalent in audit studies, and to make a methodological contribution to experimental studies investigating group-level labor market disparities (Kessler et al., 2019, 2022).

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 ability is interpreted in occupations that require different skills. However, ability and performance are usually hard to quantify in high-skilled labor markets. Compared to 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.

Closer to our paper are the contemporaneous studies by Feld et al. (2022) and Avery et al. (2023). Both studies show that providing recruiters with information about non-coding-related skills (aptitude, personality) of job applicants (Feld et al., 2022) and evaluation scores provided by the AI software (Avery et al., 2023) eliminates their perception of a gender gap in performance. 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 potential candidates.

Finally, we also contribute to the literature on digitization of labor markets. The global reach of online platforms enables employers to access a larger and potentially more diverse pool of workers (Brynjolfsson et al., 2003). Design choices relying on new technologies have been seen as a way to help mitigate systematic biases that occur in reviews and reputation systems at play in the hiring process (Cowgill, 2018; Bohnet, 2016). However, the increasing use of algorithms to automate decision-making has sparked concerns that these automated choices may produce discriminatory outcomes

(Lambrecht and Tucker, 2019; Chan and Wang, 2018; Edelman et al., 2017; Fisman and Luca, 2016). There is even some evidence that algorithmic tools can lead to worse hiring decisions (Hoffman et al. 2018). With this paper, we shed light on how automated evaluations of quality interact with in-person interviews to shape assessment of job applicants and hiring decisions.

2 Administrative Data from In-Person Coding Interviews

Recruiters of programmers are in the unique position of being able to test a prospective employee's ability to solve problems using skills they would require on the job. For many leading technology companies such as Google and Atlassian, interviews are comprised in part of coding challenges designed to test the relevant skills.

Our data come from one of several specialized platforms have been developed for this purpose, including 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 out case, the company focuses on practice interviews.

2.1 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.[1]

Each pair of users who are matched interview each other in turn. Depending on the language and self-reported ability and experience of the interviewee, a coding problem is assigned. Candidates can participate in as many different practice interviews as they like and each time, will be paired with a different counterpart. The interviewee then proceeds to solve the coding problem in an online text editor that both sides can see. At the same time, the users communicate via video chat (see Figure A1). Once the

[1] This algorithm chooses a matching that maximizes the total of the similarity scores of paired users.

interview finishes, the interviewer and interviewee swap roles. At the conclusion of their interaction, each of the two users rates the other on their coding quality, creativity, likability and overall performance.

Between December 18, 2015, and April 18, 2018, users on the platform engaged in 60,513 interviews. Eighteen percent of these users were female, and 81 percent were male.[2] The users mainly hail from English-speaking countries, the US, the UK and Australia but also from Europe, Brazil, Chile, India and Russia (Figure A2). The platform's user base has grown rapidly over time, starting with only a few users per day in early 2016 to around 150 per day in mid-2018 (see Figure A5). For the period of August 2016-March 2018, Table 1 shows that users were participated on average to 12 sessions. 32 different problems were assigned to the participants.

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 tasks are performed under time pressure. 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."

Additional 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 of participants had Masters degrees, and nearly all of the remaining users held a bachelor's degree (see Figure A3). Two thirds of users had computer science degrees, with most of the remainder spread between engineering, mathematics, statistics and the hard sciences (see Figure A4). Women represent about 17 percent of users on the platform. Consistent with evidence from Murciano-Goroff (2018), we find that on av-

[2] A small fraction of users could not be classified.

erage women declare lower level of preparation before the intervention.

We obtained a second dataset from the platform of 482,390 session-user pairs spanning from January 2018 to October 2022. Crucial for our analysis, this dataset contains first and last names. We were able to link the platform mock interviews data to the Revelio Labs database to retrieve participants' labor market outcomes, discussed in Section 4.8. This also allows us to predict the race and ethnicity of platform participants (see Section 6 for details). Finally, this dataset also contains the code scripts of users that we use in Experiment II. Figure A7 presents the overall data infrastructure of the paper.

2.2 Gender Gaps in Evaluations of Code Quality

We start by documenting gender gaps in evaluations at the end of these interviews. Figure 1 shows the average scores for men and women from January 2016 to July 2017. The information that evaluators see about coders is held constant in this period. Table 2 shows that during 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.

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 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). Finally, the gaps do not vary substantially by problem difficulty (see Figure B9). Finally, they persist when we add date fixed effects to take into account changes in composition as the platform grew.

3 A Guiding Model of Discrimination

Without further evidence, the gender gap in ratings we see is consistent with unmeasured differences in performance, discrimination, or a combination of phenomena. Guided by a simple model of discrimination in the spirit of Lundberg and Startz

(1983), we investigate the sources of these gaps throughout the rest of the paper.[3]

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, θ_i. In the context of the coding interviews we analyze, 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 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 μ_g and variance σ_g^2.

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

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

The signal that an interviewer observes is unbiased, but noisy. Specifically, $\theta_i = y_i + \varepsilon_i$, where ε_i is normally distributed with mean zero and variance σ_ε^2, and is independent of both y_i and g. The unconditional distribution of θ_i is therefore as follows.

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

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 Bayesian inference based on this noisy signal implies that the interviewer uses her belief about the population as well as the information contained in the signal. Specifically, the interviewer's belief, b_i about the candidate's performance is a simple

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

weighted average of the signal and the group mean:

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

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

The role of the interviewer's *ex ante* belief is greater if the signal is less informative.[4] In the extreme case in which the signal is completely uninformative, the interviewer's estimate of every candidate's performance is simply her belief about the mean given the candidate's gender, μ_g. In 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 the i's performance but may feature systematic bias as well. Specifically, we let the rating be a function:

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

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

3.4 Types of Discrimination

Statistical Discrimination

Rational belief formation may be consistent with statistical discrimination, which 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.

If we suppose that interviewers believe the *variance* of ability, σ_g^2, to be the same for both genders, then $s_m = s_f = s$.[5] In this case, the gender difference in beliefs about

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

[5] Differing prior variances—holding fixed the mean—leads to lower ratings for the high-variance

code quality given a fixed signal realization, θ_i, is given by equation 1.

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

Equation (1) 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 σ_ε^2 is larger, or the interviewer's beliefs are more strongly held so that σ_g^2 is smaller.

Since the gender gap in Equation (1) is conditional on interview performance, it constitutes discrimination. If interviewers' prior beliefs are correct, then a prerequisite for such a gap to exist is that there is a true difference in *average* coding ability between men and women on the platform. However, it is also possible that the difference between μ_m and μ_f reflects a mistaken belief (a "bias").

Non-Statistical Biases

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

$$\text{Bias} \mid b_i = R(b_i \mid g_i = m, \boldsymbol{c}) - R(b_i \mid g_i = f, \boldsymbol{c}). \tag{2}$$

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

group at the high end (for the same signal) but higher ratings at the low end.

4 Experiment I: Providing Objective Information

We begin by analyzing a natural experiment which allows is to test the hypothesis that the gender gaps in code quality ratings on this platform are driven by incorrect beliefs about gender. On July 8, 2017, the platform introduced direct tests of code quality in the form of a series of automated tests ("unit tests") during the interview. These unit tests evaluated whether the code ran without errors, and produced the correct answers for test cases. An example of such tests is shown in Figure C20 Panel C, together with the question (Panel A) and the code block (Panel B). Only a subset of users activated these tests, but they provide valuable information for the majority of candidates who did so. This additional information was provided to the evaluator before the subjective evaluation was chosen.

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, however, are consistent with interviewers' prior beliefs being well-calibrated on average.

Letting μ_g^* be the true average ability of gender g candidates, the unconditional gender gap in beliefs is given by Equation (3).

$$\text{Gender Gap} = s \underbrace{\left(\mu_m^* - \mu_f^*\right)}_{\textbf{True gap}} + (1-s) \underbrace{\left(\mu_m - \mu_f\right)}_{\textbf{Believed gap}} + \tau_m \tag{3}$$

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.[6] The effect of this depends on whether interviewers believe that the gender gap in coding ability is larger than it is in reality. If they do, then more information will shrink the gender gap. If they believe

[6] 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 more productive. See Craig (2019) for an analysis of this issue. In our setting, however, the set of coding solutions being evaluated is fixed.

it is small than it really is, the gap in beliefs would widen. In this sense, a finding that the unconditional gap in interview ratings narrows would simultaneously provide evidence of bias, and an effective solution to that bias.

4.2 Intervention

The change by the platform introduced direct tests of code quality via a series of automated tests (unit tests) during the interview. These unit tests evaluated whether the code ran without errors, and produced the correct answers for test cases. Users can choose to activate the tests by pressing a button (see Figure A1). The evaluation is then visible to both the interviewer and the interviewee. Finally, users can run the tests, and observe pass/fail outcomes. We view this as equivalent to increasing the precision of the signal, θ_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. As shown in Figure A6, the intervention was phased in gradually over time. The share of users treated at least once increased from July 2017 until all users were treated in October 27, 2017. During the staggered implementation (July-Oct 2017), we have data for all 6,401 sessions and 3,167 interviewees.

Figure A8 details how new users are assigned to treatment or control conditions as they enter the platform during the phase-in period. When a new user i is paired to another user j, one of two configurations arises. First, for pairs where both i and j are new users or who 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 all future interactions. Thus, any candidate matched with a partner in the treatment condition will automatically be treated as well. If i is matched to j who is already in the treatment condition, i therefore becomes treated without randomization. We deal with this potentially imperfect randomization in Section 4.6.

Baseline characteristics are reasonably balanced between the treated and the control groups, as shown in Table B3. However, 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 certain specifications control for propensity score of being treated.

4.4 Identification

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 had to *choose* to activate the device during the interview, and not all did so. We therefore start with an Intention-to-Treat (ITT) model:

$$Y_{it} = \alpha + \beta T_{it} + \theta_t + \epsilon_{it} \tag{4}$$

where Y_{it} is the score of individual i for a session on date t, and θ_t are date fixed effects. T_{it} corresponds to a pair of users for whom the new feature was "enabled". The estimation of the β coefficient from Equation (4) corresponds to the ITT. Standard errors are clustered at the date level.

Next, we account for the fact that not all users activated the tests by using treatment assignment as an instrument for actual treatment. This allows us to estimate the treatment effect on the treated (TOT). Specifically, we estimate the following model using two-stage least squares (2SLS):

$$Y_{it} = \gamma + \delta D_{it} + \lambda_t + \eta_{it} \tag{5}$$

$$D_{it} = \mu + \pi T_{it} + \zeta_t + \nu_{it} \tag{6}$$

where Y_{it} is the outcome of user i at time t; D_{it} is a dummy variable indicating whether the user activated the objective tests; and T_{it} is an indicator of whether the pair was assigned to treatment; and λ_t and ζ_t are time fixed effects. Standard errors are again 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 of Equation (4) and Equation (5) 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 (4), and the second column presents 2SLS estimates Equation (5). The first stages Equation (6) are summarized in the lower sub-panels.

Activation. The first stage estimates indicate that 71 percent of users enabled the objective code quality tests when they were made available by the platform. This is a strong first stage, and suggests that the code quality ratings were observed and valued by participants on the platform. 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 difference in feedback aversion.

Complier Covariates. We characterize compliers by observable characteristics in Table B4.[7] As expected given the balance checks in Table B3, the treated and untreated complier estimates are very similar. Column (5) also presents mean characteristics for never-takers for comparison.[8] The comparisons in Table B4 reveal that for most subgroups, their representation among compliers is similar to the overall sample, although compliers do have slightly lower level of experience. However, the results confirm the gender gap in activation estimated from Equation 6: Compliers are less likely to be women than never-takers.

Impact on Subjective Ratings. The ITT and 2SLS estimates in Table 3 suggest that the availability of objective code quality measures increased subjective ratings, but did they did not disproportionately raise the ratings of women. 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. The increases in ratings are generally larger for men, particularly for coding and likability ratings, where the ef-

[7] 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_iD_i (Column 4) and $X_i(1-D_i)$, using T_i as an instrument for D_i.

[8] We estimate never-taker means by regressing $X_i(1-D_i)T_i$ on $(1-D_i)Z_i$. Note that there are no always-takers in this setting.

fects 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

Table 4 provides a battery of robustness checks to assess the validity of our results. We begin in Panel A with a baseline in which we estimate the Intention-to-Treat (ITT) model interacted by gender:

$$Y_{it} = \alpha + \beta T_{it} + \gamma T_{it} * \text{Woman} + \theta_t + \epsilon_{it} \tag{7}$$

where Y_{it} is the score of individual i for a session on date t, and θ_t are date fixed effects. T_{it} corresponds to a pair of users for whom the new feature was "enabled". The estimation of the β coefficient from Equation (4) corresponds to the ITT and γ to the effect on the gender gap in ratings.

Additional Covariates. In Panel B, we introduce month-of-interview fixed effects. The in Panel C we include date-of-interview fixed effects. These help account for the fact that the fraction treated changes over time, and so 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 tends to attenuate the treatment coefficient on most outcomes, and the interaction coefficient γ fails to be statistically significant.

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 size of the coefficients declines 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 and find consistent results.

$$Y_{it} = \alpha + \beta T_{it} * Post + \gamma T_{it} * Post * \text{Woman} + \theta_t + \epsilon_{it} \tag{8}$$

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 to the identification, we control our regression results with the propensity score obtained from a matching procedure in Panel G of Table 4. For the matching procedure, we control for month-of-interview fixed effects, and, for both the interviewer and the interviewer, 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. Additionally, 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 (results available upon request).

Changes in the Composition of Users Over Time. Conditional on individual's covariates and other users' covariates, treatment assignment should be as good as random. However, we also explore changes in the composition of users on the platform over time. First, the gender composition of the platform users didn't change drastically after the introduction of the new device, as shown in Figure B13. Second, Figure B14 presents the evolution of first-time users' characteristics on the platform of over time. We find no evidence of changes after the introduction of the device in terms of users' probability of being a US citizen, of having a computer science degree, a graduate degree, or no working experience.

Treatment-Induced Selection. A natural questions is whether the treatment increased the pool of qualified women to choose from. We investigate here whether the treatment led to a changes in how female interviewees selected into the platform. Figure B15 confirms that there are no changes in the characteristics of first-time female users around the date of the introduction of the device on the platform in terms of work experience, educational background or field of study. Second, we look at the evolution of the share of high-performing users among first-time users, defining high-performing first-time users as those who passed all unit tests taken for a given problem during their first interview experience on the platform. Figure B16 plots the shares of high-performing first-time female and male users and shows that they follow a parallel increase over time. While the quality of all first-time users increases over time, it does not increase differentially by gender. As our main specification controls for date-of-interview fixed effects, our results are cannot be explained by positive selection that would affect only one group.

Gender Differences in Device Use. If men and women were characterized by different abilities to adopt the device, it could potentially explain why the gender gap is not closed after the implementation of the device, but that this effect could take some time to materialize. We explore this possibility by looking at the dynamics of adoption of the device by gender. In Figure B11, we plot the learning curve of both male and female users, measured the number of test passed over time. We dis-aggregate both by number of days and by number of sessions, to account for the fact that women might not be using the platform as frequently as men. Overall, we observe that the learning curves are remarkably similar, and that, if anything, the curve is steeper in Panel B of Figure B11. This suggests that a slower adoption rate cannot in itself explain our results. Additionally, we explore the possibility that the use of the device could be interpreted differently, if users take tests a lot because they have low level of self-confidence for instance, or to the contrary if they want to signal their ability by using the device a lot. Figure 3 shows the average objective coding performance (number of tests completed over test passed) according to the number of tests taken, separately for male and female users, and rejects the hypothesis that men and women's device uses correlate with differences in objective performance.

Problem Assignment and Evaluator Type. Additionally, we explore characteristics of the match between participants and problems. If women were systematically assigned easier problems, this could potentially explain why the updating differs by gender. To explore this possibility, we compute the average objective performance of users for a given problem (a high average performance corresponds to a low-difficulty problem). We show in Figure B17 (Panel A) that problems substantially differ in difficulty levels. Table B5 confirms that, with the exception of interviewer's years of experience, participants' characteristics are reasonably balanced across problem's average difficulty, split by the median ratio of tests solved over tests passed. We also rank problems by the standard deviation of the performance, as shown in Figure B17 Panel B. We define a problem with a high standard deviation of performance as a proxy for a less informative signal about an individual's performance. As presented in Table B1 columns (1) and (2), the gender of the interviewee does not predict the type of problem assigned, both in terms of difficulty and standard deviation. Additionally, we explore the possibility that the ranking of problems' difficulty vary by gender. Figure B12 shows the relative ranking of problems' difficulty by gender. The ranking is proxied by the average performance of users of the same gender for each problem. The orange vertical lines indicate any positive (negative) deviation upward (downward) of female users' ranking compared to male users' ranking. We conclude that the two rankings are overall similar. Finally, we explore whether women are more likely to be matched with harsh evaluators. We define a harsh evaluator as an interviewer whose average coding ratings (excluding the session's rating) is below the median. As presented in Table B1 columns (3) and (4), female users are not more likely to be matched with a harsh evaluator.

Problem Difficulty and Precision of the Signal. Finally, we investigate whether the persistence of the gender gaps can be explained by belief updating. We first investigate whether our results vary by problems' characteristics. We use the quasi-random assignment of the 31 coding problems to investigate how gender gaps in ratings vary depending on the difficulty and ambiguity of the problem solved. We compute again the average objective performance of users for a given problem (a high average performance corresponds to a low-difficulty problem). Following Bohren et al. (2019a), we explore variations in results across the level of precision of the signal, proxied the standard deviation of performance. To gain precision, we group problems by difficulty and precision levels, and estimate again Equation (5) separately for each group and each gender. Results are presented in Figure B18. 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 across problems of various difficulty levels. We provide 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. Overall, these results are consistent with previous studies looking at differences in updating by group (Sarsons 2022). Figure B18 Panel B confirms that for low standard-deviation problems (when the signal is more precise), the treatment effect on subjective ratings is higher for both genders, despite being consistently lower and imprecisely estimated for women. These results also provide an indirect test for inattention: if the users were not paying 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

Figure 4 shows that gender gaps in subjective ratings are not fully explained by objective performance differences between men and women, as measured by these tests. Panel A of the plots the average subjective ratings in coding by objective performance (ratio of tests completed over tests passed at 100 or less), and Panel B shows ratings for problem solving. The plots are separated by gender.[9] 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 5). These residual gaps amount to about 6 percent of a standard deviation.

To test for the role of learning in correcting potential 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 B6.

[9] 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.

The gender gap in subjective ratings does not vanish when we account for the interviewer's learning on the platform, 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 learning plays a significant role in this context.

4.8 Ratings of Code Quality and Labor Market Performance

To investigate the extent to which our objective measure of coding ability relates to actual differences in the labor market, we link the platform mock interviews data with the Revelio Labs database. Revelio offers a standardized database of hundreds of millions of publicly available employment records from websites such as LinkedIn and job posting boards. It allows us to observe the near universe of Computer Science (CS) related job spells, and graduates in the US labor market. Although salaries are not directly reported, they are imputed using job posting data, H1B-visa records and the Current Population Survey.[10] A possible concern is that data from websites such as LinkedIn is susceptible to a degree of sample selection, for example only high achieving CS graduates will show up. However, we have reason to believe that this is unlikely to be the case in our setting. One, our mock interview participants are actively seeking employment in a CS related position, thus having an online presence is necessary. Two, the US produces almost 60,000 computer science baccalaureates annually. Data from Revelio shows that we have an average number of annual degrees close to that from 2016 to 2026.[11]

Using our platform data, we select US residing participants with either a bachelors or masters degree. We then match this sample with the universe of individuals who attained a CS related degree in a US institution from Revelio. We use first and last name, and degree type to find *exact* matches between participants on the platform and Revelio. Observations matched to multiple Revelio profiles are dropped.[12] The final sample consists of 5,126 matched CS graduates from 2016 to 2023. For 50 percent of this

[10] More detail regarding the Revelio data database is available here

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

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

sample, we have data on their objective performance on the platform. The outcome variable of interest is the first salary post graduation, although we also look at average salary post graduation.[13]

Results are presented in Table 6. In column (1), we start by documenting a 8 percent residual gender pay gap for computer science graduates for the (log) first salary post graduation. We use a Mincer-type wage regression in which we control for individuals' characteristics such as race, the highest degree obtained, institution-of-highest-degree, year-of-graduation and city fixed effects. Without more information, this 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 are agnostic about the sources of this pay gap, as our goal is to validate the objective measure of coding quality and investigate different 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 sessions on the platform and whether the participant had already graduated when they took sessions on the platform.[14] We find a positive and statistically significant coefficient (0.055, SD=0.022) for the standardized objective score measure. Going from the 25th to the 75th percentile of standardized score is associated with a wage increase of 5 percent. While this objective measure is potentially correlated with the quality of training received by participants, this exercise validates this variable as predictive of labor market outcomes. Including the subjective coding and communication ratings to the regression does not affect the magnitude of the objective measure of coding quality: the returns of these measures of skills is small and imprecisely estimated for the overall sample. Interestingly, we find heterogenous returns of skills by gender in columns 3 to 6: we find a precise zero return of the objective measure of coding performance for women.[15] Additionally, the interaction between the female dummy and the subjective coding measure is negative (although imprecisely estimated), and is positive with the subjective com-

[13] 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.

[14] To reduce noise, we re-weight the regression for the number of sessions each user had on the platform. The results are qualitatively similar when we add weights.

[15] See Table B7 for estimation on separate samples.

munication measure. These results complement previous studies documenting the growing returns to social skills (Deming, 2017; Deming and Kahn, 2018; Edin et al., 2022), by showing differentials returns of cognitive and non-cognitive skills by gender in a math-intensive field.

5 Experiment II: Blind and Non-Blind Code Evaluation

Having seen no evidence that gender gaps shrink with the provision of additional information, we now 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. The experiment asked computer scientists to evaluate code written on the platform. Following settings in which blind evaluations occurred (Goldin and Rouse, 2000), we compared these evaluations in a "blind" setting to those when gender was revealed via the name of the code. The aim of this was to establish whether residual gender gaps in subjective ratings are due to unmeasured differences in code quality as assessed by evaluators, or gender bias.

The RCT was pre-registered on December 14, 2022.[16] The participants are 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 C11. To complement the discussion here, full detail 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 λ_g be the fraction of participants of gender $g \in \{m, f\}$, and assuming

[16] ID: AEARCTR-0009816. The pre-analysis plan is available on the AEA RCT registry website (updated version: February 17, 2023).

that performance of each gender is normally distributed, the pooled belief is:

$$y_i \sim \mathcal{N}\left(\mu, \sigma^2\right) \tag{9}$$

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

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

$$E\left[y_i \mid \theta_i, g\right] = \widetilde{s}\theta_i + \left(1 - \widetilde{s}\right)\mu \tag{10}$$

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

$$\text{Gender Gap} = \widetilde{s}\left(\mu_m - \mu_f\right). \tag{11}$$

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. To add value beyond the experiment itself, we can also examine particular dimensions of performance. To this end, we pre-registered different dimensions of the code on which to examine gender differences (length, time of program execution, number of comments, maintainability of code).[17]

5.2 Empirical Design

5.2.1 Code Blocks from the Platform

We use de-identified code blocks written by a set 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 C20 Panel B. For each code block, we have access to the platform's objective measures of performance including sub-test results. Descriptive statistics and demographics of each step of the sample construction are presented in Table C8 and in Table C9. In Table C10, we replicate the results

[17] We found that an AI tool (Chat GPT) was not able to predict the gender of the coder of a code when the first name was not displayed.

of Table 5 for this sample of codes and find an even larger gender gap in subjective ratings when we control for objective performance. Our final sample is stratified 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.[18]

5.2.2 Treatment

Each evaluator i is assigned a set of four code scripts in a random order. We use a within-subject design. 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 is a high-performing coder. 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. An example of each treatment condition is presented in Figure C21. 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 C12 confirms that the characteristics of evaluators are balanced across each treatment order.

5.2.3 Outcomes

We asked evaluators to judge the quality of the code using the same Likert scales as on the platform. The evaluators were recruited among students computer science who had familiarity with the relevant programming languages.

Main Outcome. Our primary outcome is evaluators' subjective ratings of the code quality. For each block of code, respondents will be asked to rate problems on a scale from 1 to 4. For all primary hypotheses, we will 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

[18] We choose to stratify by race to keep a representative population of coders for our experiment. We further discuss racial bias in Section 6.

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, and we focus on an unusually high-skill segment of the labor market.[19]

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 ask 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 C16. 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.[20]

To measure participants' priors, we exposed them to three different vignettes before the 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.

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

[19] 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).

[20] Table C13 confirms that the characteristics of evaluators are also balanced across each treatment order for the quality sample.

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 excepts will reinforce the credibility of our design and encourage participants to exert effort in the evaluation process.

A risk 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 incentive compatible questions where individuals are asked to predict the unit tests passed by the code. Additionally, we provided a symbolic but potentially powerful incentive selecting a set of evaluators to the Creative Destruction Lab 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 to have increased the incentive for participants to accurately evaluate the code blocks.

Finally, university student evaluators were not in the position to hire 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 the evaluator will have ongoing interactions with workers they hire.

5.2.5 Hypotheses Tested

We present here the hypothesis from our pre-analysis plan

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 the 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 H1, we use the following specification:

$$\begin{aligned} Y_{ij} &= \beta_0 + \beta_1 \times NB_{ij} + \beta_2 \times treatment\_order_i + \beta_3 \times strata_j \\ &+ \sum_{k=1}^{4} \gamma_{jk} \left(order \times \mathbb{1}_k \right) + \pi_{p(j)} + \delta_i + \epsilon_{ij} \end{aligned} \tag{12}$$

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; $treatment\_order_i$ is an indicator for the randomly assigned treatment order ("non-blind then blind" condition versus "blind then non-blind") that the evaluator sees; $strata_j$ are the four strata fixed effects (female×high_performer); γ_{jk} are script-number fixed effects to account for fatigue and learning; $\pi_{p(j)}$ are problem fixed-effects. In some specifications, we include δ_i evaluator fixed effects, and controls. Since code blocks characteristics are randomly drawn, including these variables in the analysis should not affect our estimates but could increase precision. Standard errors are clustered at the evaluator level. In Equation (12), the coefficient of interest in β_1, which measures the average differences in subjective ratings for code blocks where the gender of the coder is revealed or not, controlling for the treatment order. This does not test for differences across gender, but rather whether non-blind codes are evaluated more harshly regardless of the gender.

To test H2, we will use the following specification, which is very similar to Equation (12) but interacts the key variables with gender indicators:

$$\begin{aligned} Y_{ij} &= \beta_0 + \beta_1 \times female\_coder_j + \beta_2 \times NB_{ij} + \beta_3 \times NB_{ij} \times female\_coder_j \\ &+ \beta_4 \times high\_performer_j + \sum_{j=1}^{4} \gamma_{jk} \left(order \times \mathbb{1}_k \right) + \pi_{p(j)} + \delta_i + \epsilon_{ij} \end{aligned} \tag{13}$$

The coefficients of interest are β_1, which measures productivity differences between male and female codes in the blind condition, and β_3, which measures the differential effect of revealing the gender of the coder on subjective ratings, depending on what that gender is. The addition of evaluator fixed effects provides us with an indirect test of H3.

To test H4 to H9, we will use variant of Model (13) 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 7 (H2) presents our main results. The estimate of β_1 shows that in the blind condition, codes written by female coders don't receive systematically different ratings, unit tests prediction or interview predictions. If anything, the coefficients are positive but imprecisely estimated. This rules out any systematic gender differences in coding styles that could drive gender disparities in the in-person interviews and that are not accounted for by the unit tests results (Vedres and Vasarhelyi, 2019). Second, we find negative and noisy coefficients for the non-blind codes (β_2 in Equation 13), and positive but imprecisely estimated coefficients for the interacted effect with female-written codes (β_3 in Equation 13). We further discuss these results when we decompose them by race in Section 6. Overall, we don't provide evidence of a uniform gender bias in the non-blind condition. Table C14 (H1) reveals that codes evaluated in the non-blind condition tend to receive lower ratings, unit test prediction and interview prediction, and that codes seen at the beginning of the task are evaluated more harshly.[21] Finally, we don't 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.[22]

Priors. Experiment II also allows us to explore participants' prior beliefs about differences in ability between men and women. Figure C19 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. Overall, we find that evaluators don't systematically underestimate women's performance.

[21] Results on the "quality sample" are presented in Table C15 and point to similar effects.

[22] Results available upon request.

6 Racial Discrimination

To measure coders' race, we first predicted race and ethnicity based on first and last names of the coders.[23] This measure has two goals: first, it allows us to proxy for the "true" race and ethnicity as observed by participants on the platform during the in-person interviews. Second, this measure allows us to reject productivity differences between groups in the blind condition of Experiment II. However, participants in Experiment II were exposed to first names only, which makes this categorization imperfect to capture potential racial bias. We therefore asked two external reviewers to provide their best guess of the race of each coder on the basis on their first name only.[24]

First, using the in-person interactions data, we document a racial penalty for coders who are not white or East Asian, controlling for objective performance measure. Results are presented in Table C17 (Panel A).[25] To gain power, we group "white" and "East Asian" together because the separate point estimates have similar sign.[26] The penalty is robust to the inclusion of evaluator fixed effects for the sample of male coders, but becomes statistically insignificant for the sample of female coders. Additionally, results in Panel B show that the gender penalty does not vary substantially when we interact it with this racial group.

Second, we turn to our 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 C18. In columns 1 to 4, we present results using the two independent 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 find that the male penalty for non-blind codes documented previously is entirely driven by non-white non-East Asian men, consistent with the results from the in-person interactions.[27] The coefficients are stable across the different categorizations

[23] 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.

[24] We ensured that the two reviewers had different genders and races. While reviewers' assessment are relatively well correlated, the vast majority of first names over which reviewers' assessments differ are white or East Asian names according to our predictive algorithm.

[25] 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.

[26] Disaggregated results are available upon request.

[27] This category includes coders classified by reviewers as either South Asian, Black, Latinx or Other.

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.

7 What Drives the Gender Gap in Code Ratings?

In Section 2.2, we showed that there are gender gaps in evaluations of code quality which remain even when we control for rich information about both coders and the code they write. 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 evaluations in blind evaluations where 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 as well. Without some form of non-statistical bias in rating behavior, this would also imply that there would be no gender gap in evaluations of code quality.

Non-Rational Statistical Discrimination

Can the gaps be explained by non-rational statistical discrimination, with evaluator beliefs that are incorrect? 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 in Evaluations

Taste-based discrimination remains a possibility. Because there is little evidence of statistical discrimination, we can test for tasted-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. But 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 actual 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 a failure to pay attention.

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 being written. One possible 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, and how they vary across the objective performance distribution. Figure 5 plots the average subjective ratings in communication (Panel A) and likability (Panel B) by objective performance (ratio of tests completed over tests passed at 100 or less), 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 4), Figures 5 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.

8 Conclusion

We use a series of field experiments to show that the display of gender bias in performance evaluation is context-specific. We explore three different conditions in which we systematically vary the amount of information about a candidate's performance presented to evaluators. We focus on gender discrimination in evaluation of coding performance, a common step during recruitment process in the technology industry.

In line with recent work, we show that gender discrimination can take different forms beyond the traditional distinction of taste-based and (accurate) statistical discrimination, depending on the context of the evaluation of performance. During the in-person interviews, women receive lower ratings than men, holding fixed interviewees and interviewers' characteristics, and when we control for an objective measure of code quality.

The random provision of the objective measure of code quality to the interviewer does not change this gap, which suggests that statistically accurate discrimination is unlikely to drive our results. Matching our interview data to the Revelio Lab database for graduates in computer sciences, we document that the objective measure of code quality is highly predictive of higher first salary after graduation for men but not for women.

When we remove in-person interactions, gender gaps in subjective coding quality are closed, both in the blind and non-blind (when gender is revealed) conditions, which rules out explicit gender discrimination. We measure evaluators' prior beliefs about candidates' coding ability and we find that evaluators don't systematically underestimate women's performance. In the blind condition, we show that there are no gender gaps in the rated quality of blinded code, rejecting gendered pattern of behavior in coding. Our results are more in line 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). In line with the sociology literature, we show that biases are more likely to emerge when individuals are "doing gender" (West and Zimmerman, 1987) during in-person interactions, rather than when gender is signaled indirectly. They also relate to recent papers documenting differential treatment of female candidates during in-person seminars (Dupas et al., 2021; Handlan and Sheng, 2023).

Our analysis delivers two key insights on tools to mitigate discrimination in performance evaluation. First, while women received lower coding ratings than men only during in-person interactions, they receive equal ratings in communication. Hence, decoupling the coding task from the in-person interview might help mitigate biases in the evaluation of cognitive skills. Removing all in-person interactions might potentially harm female candidates, as our analysis of labor market data reveals that women experience lower returns to (objective) coding skills and higher returns to social skills (communication) than men, even in a math-intensive field. Future research could explore the mechanisms behind these differential returns and how they contribute to the gender pay gap.

Second, women and underrepresented minority 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 in-person interactions are removed, suggesting 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 explicit 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 Vecci, "Does Artificial Intelligence Help or Hurt Gender Diversity? Evidence from Two Field Experiments on Recruitment in Tech," *Evidence from Two Field Experiments on Recruitment in Tech (February 14, 2023)*, 2023.

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 Ditlmann, 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.

Bohnet, Iris, *What works*, Harvard University Press, 2016.

Bohren, J Aislinn, Alex Imas, and Michael Rosenberg, "The dynamics of discrimination: Theory and evidence," *American Economic Review*, 2019, *109* (10), 3395–3436.

_ , 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.

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.

Brynjolfsson, Erik, Yu Hu, and Michael D Smith, "Consumer surplus in the digital economy: Estimating the value of increased product variety at online booksellers," *Management Science*, 2003, *49* (11), 1580–1596.

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

Chan, Jason and Jing Wang, "Hiring preferences in online labor markets: Evidence of a female hiring bias," *Management Science*, 2018, *64* (7), pp. 2973–2994.

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.

Cowgill, Bo, "Bias and productivity in humans and algorithms: Theory and evidence from resume screening," Technical Report, Columbia Business School, Columbia University 2018.

Craig, Ashley C., "Optimal Taxation with Spillovers from Employer Learning," 2019.

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

Deming, David and Lisa B Kahn, "Skill requirements across firms and labor markets: Evidence from job postings for professionals," *Journal of Labor Economics*, 2018, *36* (S1), S337–S369.

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

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.

Edelman, Benjamin, Michael Luca, and Dan Svirsky, "Racial discrimination in the sharing economy: Evidence from a field experiment," *American Economic Journal: Applied Economics*, 2017, *9* (2), 1–22.

Edin, Per-Anders, Peter Fredriksson, Martin Nybom, and Bjoern Ockert, "The Rising Return to Noncognitive Skill," *American Economic Journal: Applied Economics*, April 2022, *14* (2), 78–100.

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.

Fisman, Raymond and Michael Luca, "Fixing discrimination in online marketplaces," *Harvard business review*, 2016, *94* (12), 88–95.

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.

Hoffman, Mitchell, Lisa B Kahn, and Danielle Li, "Discretion in hiring," *Quarterly Journal of Economics*, 2018, *133* (2), pp. 765–800.

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.

Lambrecht, Anja and Catherine Tucker, "Algorithmic bias? An empirical study of apparent gender-based discrimination in the display of STEM career ads," *Management Science*, 2019, *65* (7), 2966–2981.

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.

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.

Spence, M., "Job market signaling," *Quarterly Journal of Economics*, 1973, *87* (3), 355–374.

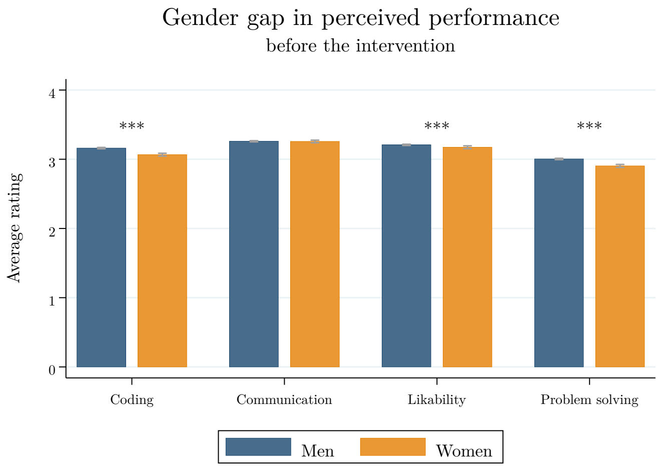
Terrell, Josh, Andrew Kofink, Justin Middleton, Clarissa Rainear, Emerson Murphy-Hill, Chris Parnin, and Jon Stallings, "Gender differences and bias in open source: Pull request acceptance of women versus men," *PeerJ Computer Science*, 2017, *3*, e111.

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.

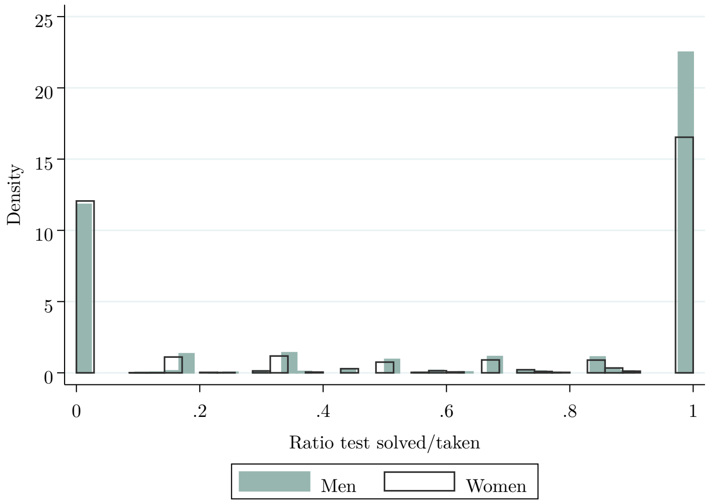
Tables and Figures

Figure 1: Pre-intervention gender gaps – Whole sample



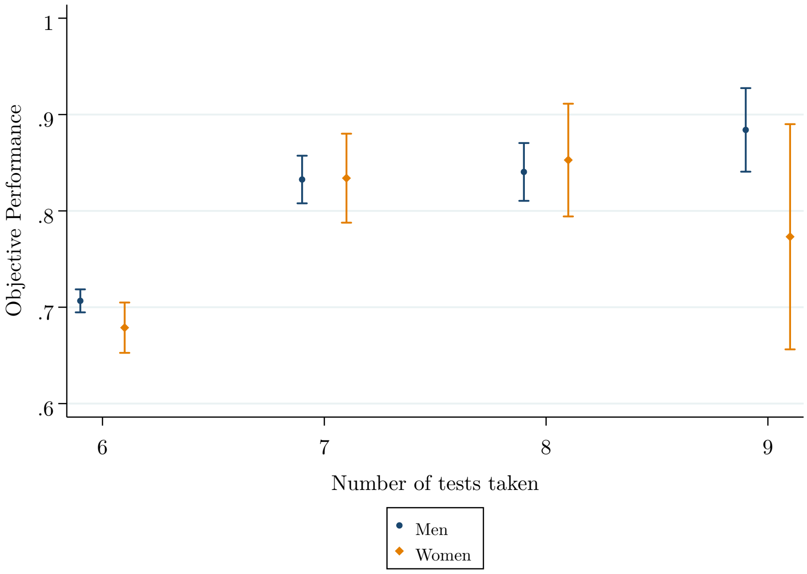
Notes: This figure shows the gender gap in peer-rated performance in five categories for normalized variables: coding, communication, hirability, likability and problem solving, for the whole sample. Stars above a category indicate statistical significance of the gap at the one percent level, and the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

Figure 2: Distribution of Objective Performance by Gender



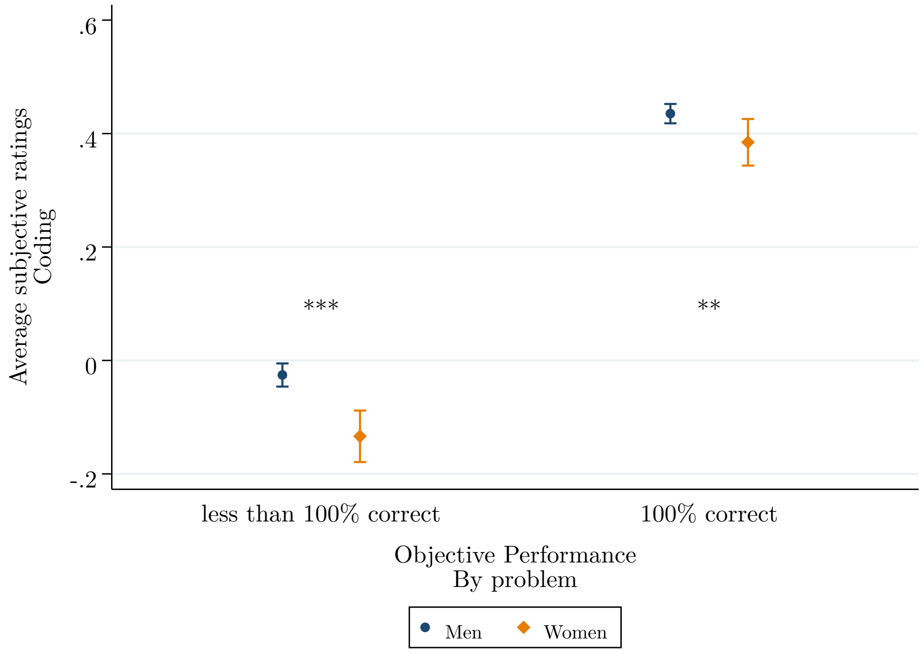
Notes: The figure presents the distribution of the objective performance measure (ratio of test solved / tests taken) by gender.

Figure 3: Objective Performance by Number of Tests Taken

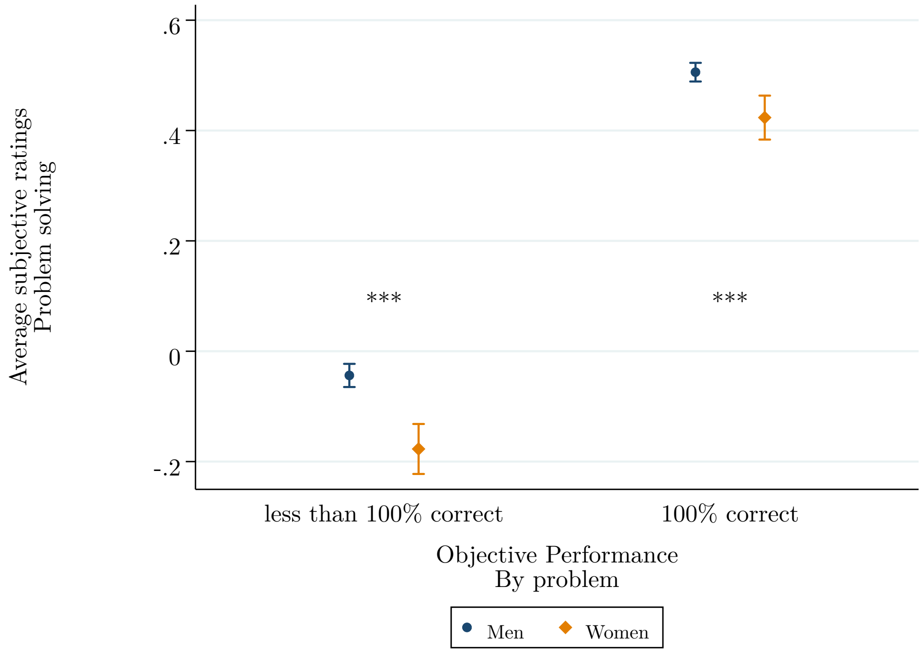


Notes: This figure shows the average objective coding performance (number of tests completed over test passed) by how many tests were taken, separately for male and female users.

Figure 4: Subjective Measure by Objective Performance — Coding and Problem Solving



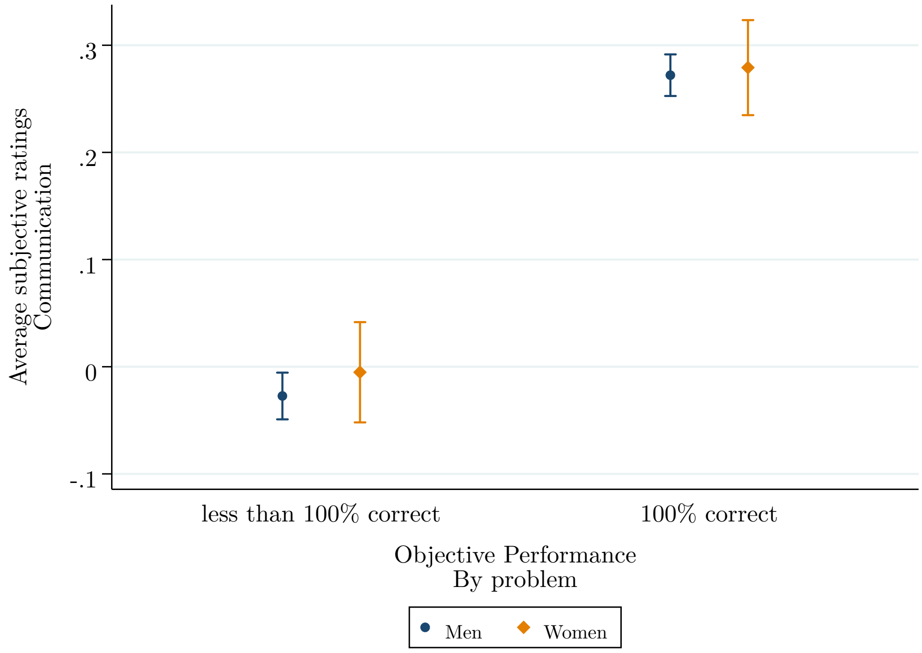
(a) Coding



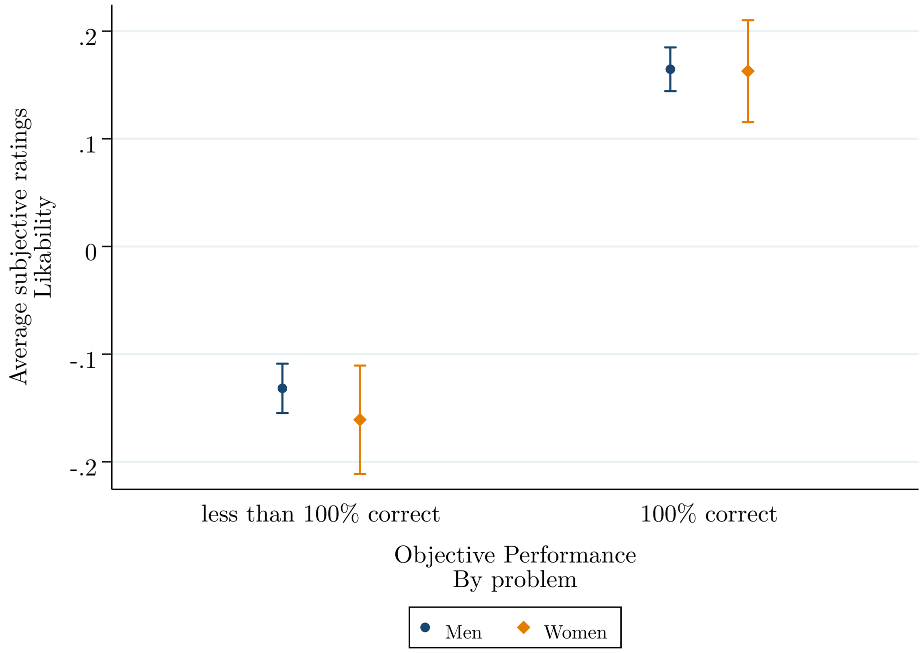
(b) Problem solving

Notes: This figure shows the average subjective ratings in coding (Panel A) and problem solving (Panel B) by objective performance (ratio of tests completed over tests passed at 100 or less), separately by gender.

Figure 5: Subjective Measure by Objective Performance — Communication and Likability



(a) Communication



(b) Likability

Notes: This figure shows the average subjective ratings in communication (Panel A) and likability (Panel B) by objective performance (ratio of tests completed over tests passed at 100 or less), separately by gender.

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

Variable	**Mean**	**Std. Dev.**	**Min.**	**Max.**	**N**
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

Variable	**Mean**	**Std. Dev.**	**Min.**	**Max.**	**N**
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

Variable	**Mean**	**Std. Dev.**	**Min.**	**Max.**	**N**
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

Table 2: Gender Gap in Subjective Ratings Pre-Intervention

	Coding				
	(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)
	Problem Solving				
	(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)
	Communication				
	(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. 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

Panel A: All

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

Panel B: Women

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

Panel C: Men

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

Notes: This table shows results for the ITT and 2SLS models using the whole sample. For each of the five dimensions on which users are rated, the coefficient on treatment in each model is shown in each of the top subpanels, for the whole sample, women and men respectively. The first stages are shown in the lower subpanels. Standard errors are clustered at the date level.

Table 4: Robustness Checks

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

Notes: This table shows results a series of robustness checks. Panel A presents the results of the baseline ITT specification (Treatment) and the interaction with a categorical variable equal to one when the interviewee is a woman. In Panel B we add month-of-interview fixed effects, and date-of-interview fixed effects in Panel C. In Panel D, we control for socio-demographic characteristics. In Panel E we expand our sample to include pre-treatment introduction interviews with month-of-interview fixed effects. In Panel F, we 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 5: Gender Gap in Subjective Coding Ratings, Controlling for Objective Performance

	Subjective Coding Ratings			
	(1)	(2)	(3)	(4)
Interviewee female	-0.0812***	-0.0638***	-0.0645***	-0.0610***
	(0.0172)	(0.0173)	(0.0173)	(0.0197)
Objective performance	0.485***	0.456***	0.457***	0.479***
	(0.0141)	(0.0141)	(0.0141)	(0.0171)
Interviewer female			0.0320*	0.0298
			(0.0165)	(0.0189)
Interviewee's sociodemographic controls	No	Yes	Yes	Yes
Interviewer's sociodemographic controls	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, 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 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 date-of-interview fixed effects.

Table 6: Labor Market Outcomes

	Ln(first salary post graduation)					
	(1)	(2)	(3)	(4)	(5)	(6)
Female	-0.084***	-0.098**	-0.102**	-0.098***	-0.106***	-0.110**
	(0.032)	(0.043)	(0.043)	(0.037)	(0.037)	(0.044)
Non white	-0.061*	-0.077**	-0.073*	-0.071**	-0.069**	-0.072*
	(0.033)	(0.039)	(0.039)	(0.033)	(0.033)	(0.039)
Masters Degree	0.150***	0.236***	0.233***	0.237***	0.241***	0.244***
	(0.034)	(0.037)	(0.037)	(0.032)	(0.033)	(0.038)
Objective Performance		0.056***	0.073***			0.075***
		(0.022)	(0.024)			(0.025)
Objective Performance × Female			-0.100*			-0.111**
			(0.055)			(0.056)
Subjective Coding Rating				0.030		-0.005
				(0.024)		(0.038)
Coding Rating × Female				-0.053		-0.084
				(0.052)		(0.089)
Communication Rating					0.023	0.019
					(0.023)	(0.035)
Communication Rating × Female					0.024	0.136*
					(0.051)	(0.082)
City FE	Yes	Yes	Yes	Yes	Yes	Yes
Institution FE	Yes	No	No	No	No	No
Graduation Year FE	Yes	Yes	Yes	Yes	Yes	Yes
Controls	No	Yes	Yes	Yes	Yes	Yes
Observations	3,709	2,352	2,352	3,121	3,121	2,339

Notes: This table presents 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 7: 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 a test for H2. The even columns include evaluator fixed effects. Standard errors are clustered at the evaluator level.

Table 8: Blinding Experiment — Main Results Racial Gaps

	Subjective Coding Ratings					
	Reviewer 1		Reviewer 2		Algorithmic Prediction	
	(1)	(2)	(3)	(4)	(5)	(6)
Non-blind code	-0.123	-0.152*	-0.149*	-0.162**	-0.170**	-0.171*
	(0.079)	(0.085)	(0.077)	(0.081)	(0.086)	(0.094)
Non-blind code×Female code	0.143	0.147	0.097	0.107	0.129	0.099
	(0.104)	(0.111)	(0.097)	(0.101)	(0.112)	(0.119)
White or East Asian	0.087	0.097	0.017	0.052	-0.006	0.014
	(0.060)	(0.070)	(0.062)	(0.068)	(0.061)	(0.072)
Non-blind code	0.081	0.127	0.139	0.160	0.146	0.144
×White or East Asian	(0.104)	(0.121)	(0.062)	(0.068)	(0.103)	(0.120)
Non-blind code	-0.174	-0.161	-0.098	-0.089	-0.137	-0.071
×White or East Asian×Female	(0.119)	(0.138)	(0.116)	(0.135)	(0.119)	(0.138)
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 investigates gender and racial disparities on final ratings, where the main racial category is white or East Asian. 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?

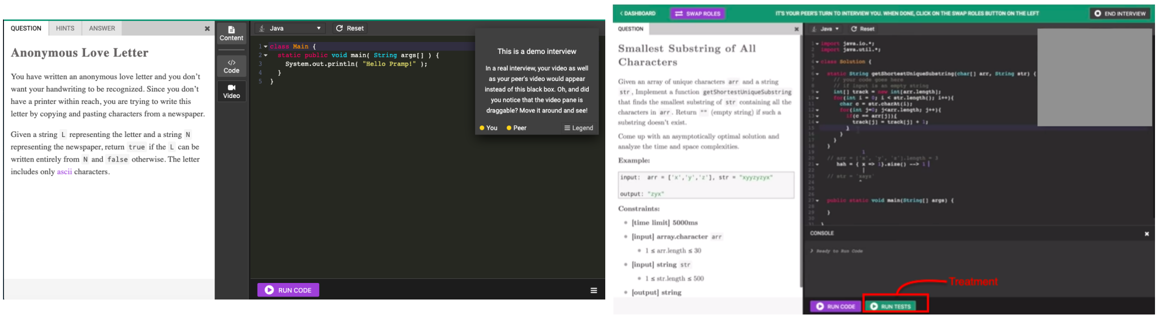
Abdelrahman Amer, Ashley C. Craig and Clémentine Van Effenterre

July 2023

List of Appendices

Appendix A Institutional details

Figure A1: Environment of the platform and treatment



Anonymous Love Letter

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

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

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

This is a demo interview

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

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

Smallest Substring of All Characters

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

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

Example:

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

output: "zyx"
```

Constraints:

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

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

class Solution {

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


  public static void main(String[] args) {

  }
}
```

CONSOLE

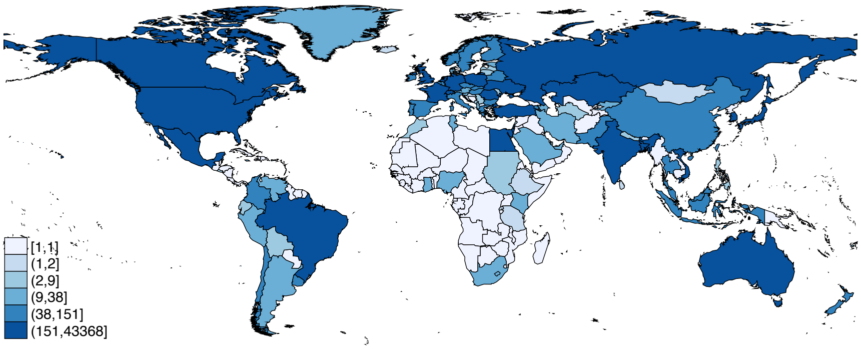
```
> Ready to Run Code
```

(a) Control

(b) Treatment

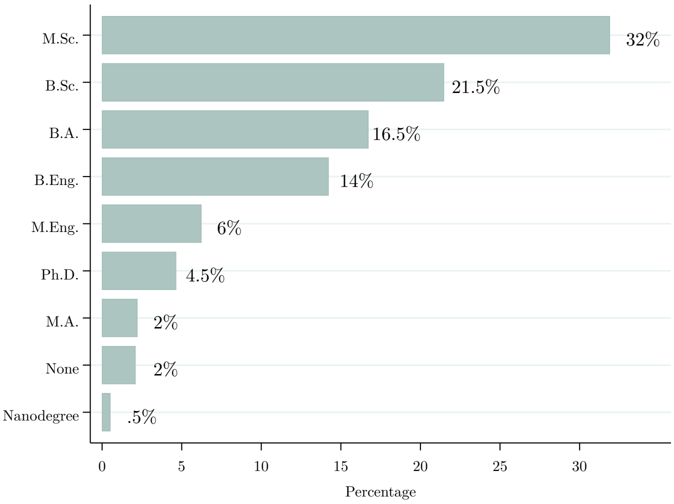
Notes: Figure A1(a) presents the website layout for a mock interview on the platform in the control condition. Figure A1(b) represents the treatment condition.

Figure A2: Users across the world



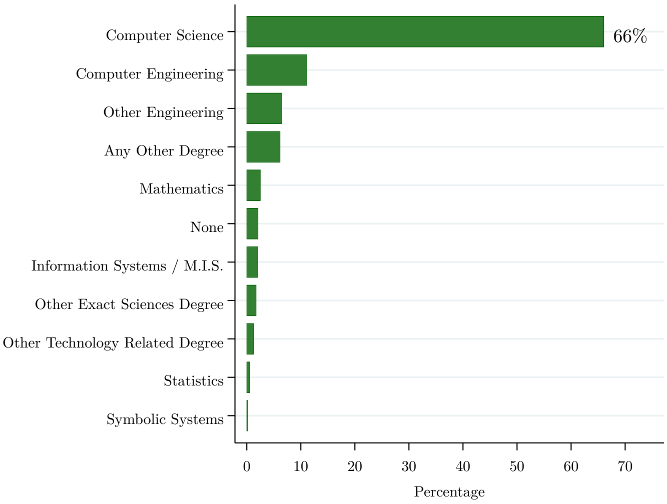
Notes: The map shows the distribution of users across the world.

Figure A3: Users' level of education



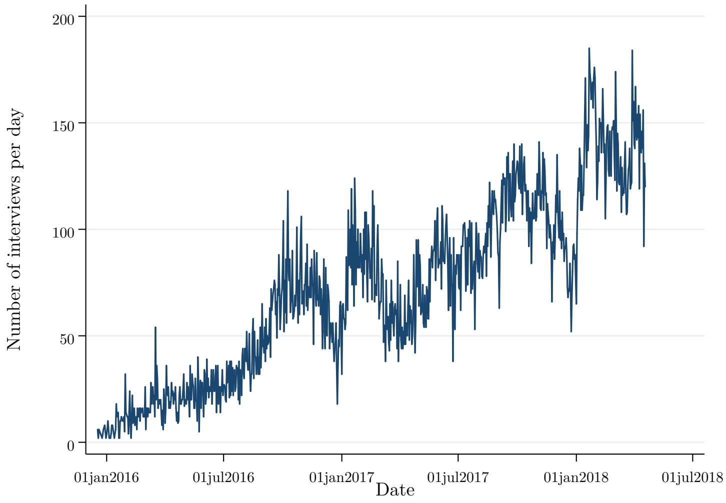
Notes: The figure presents the average level of education of users.

Figure A4: Users' field of education



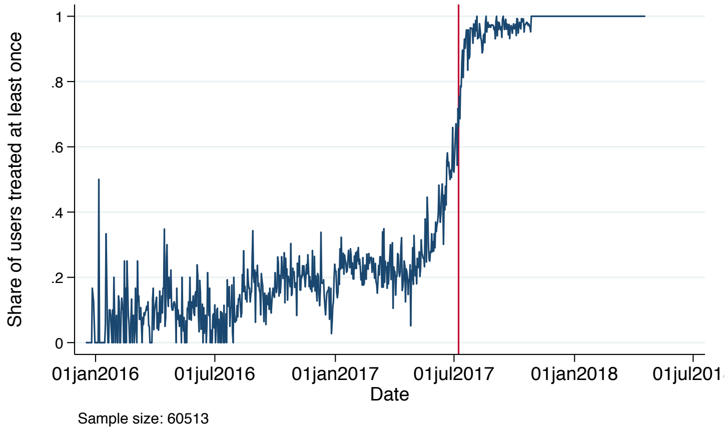
Notes: The figure presents the field of education of users.

Figure A5: Growth of the platform



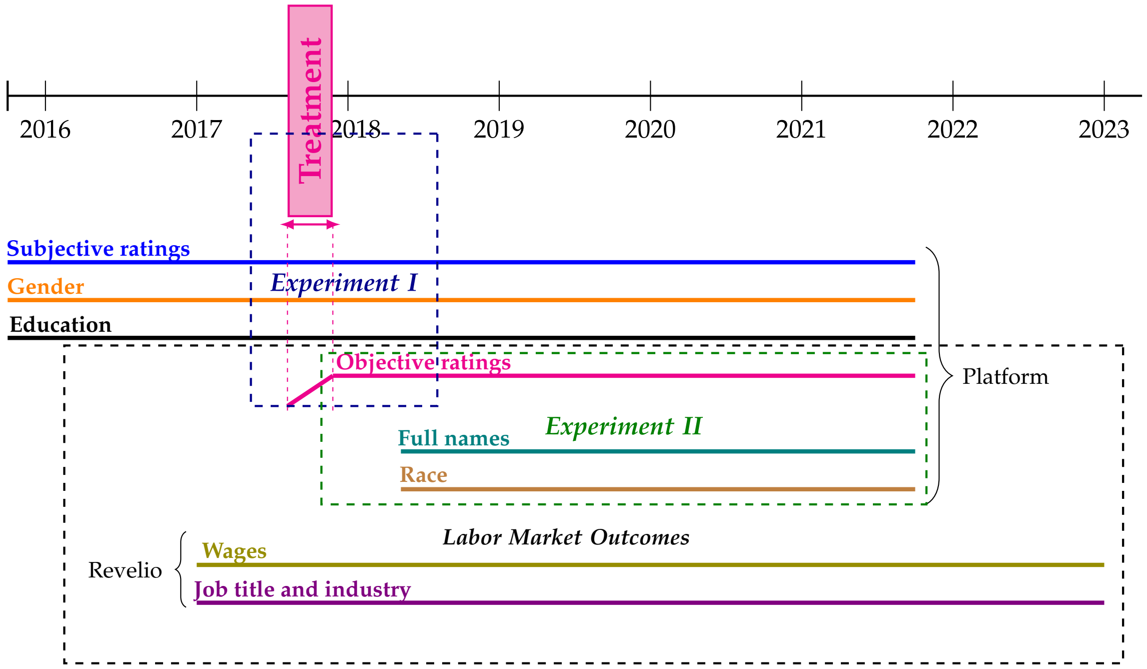
Notes: The figure shows the evolution of the number of users on the platform from January 2016 to January 2018.

Figure A6: Share of treated users on the platform over time



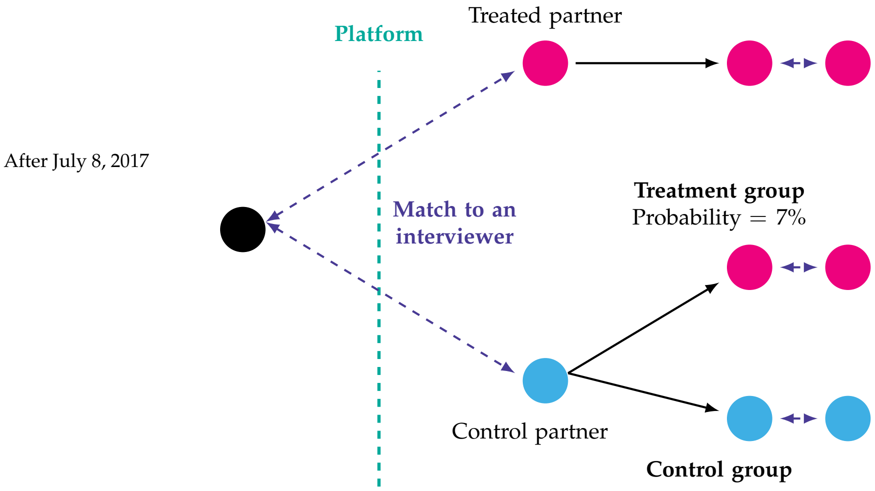
Notes: This figure shows the evolution over time of the share of users who have been treated at least once on the platform. The red line corresponds to the introduction of the new device on the platform.

Figure A7: Data Infrastructure



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.

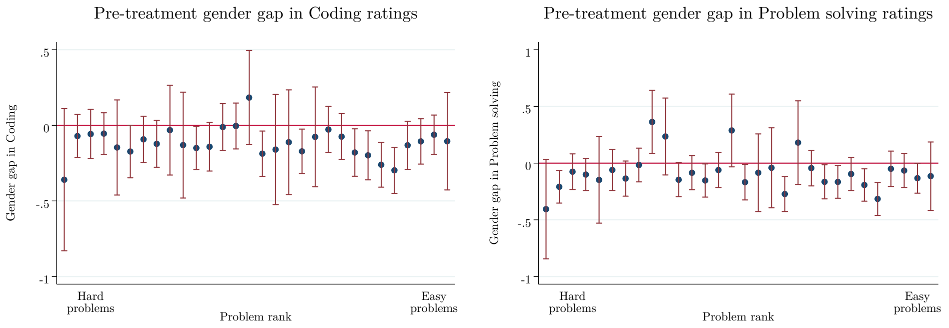
Figure A8: Treatment assignment



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

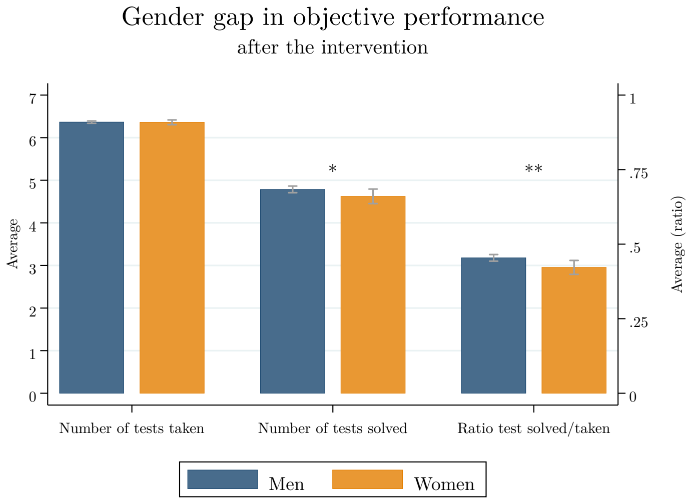
Appendix B   Additional Results

Figure B9: Pre-treatment gender gaps by problem difficulty



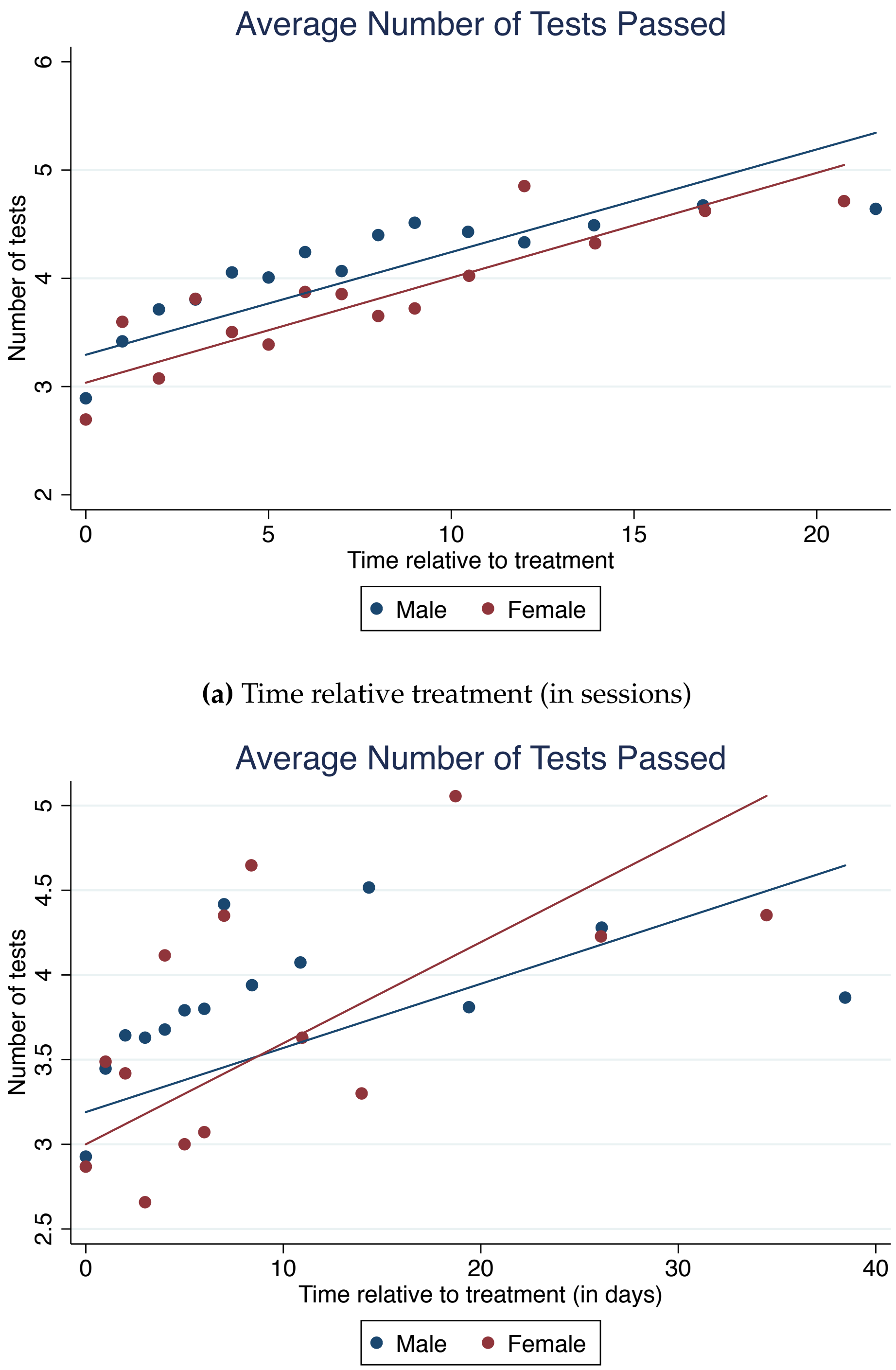
Notes: This figure plots gender gaps in subjective ratings for coding and problem solving by problem difficulty in the pre-intervention period. Problem difficulty is computed using the average objective performance of users in the post-intervention period.

Figure B10: 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 B11: Gender differences in learning

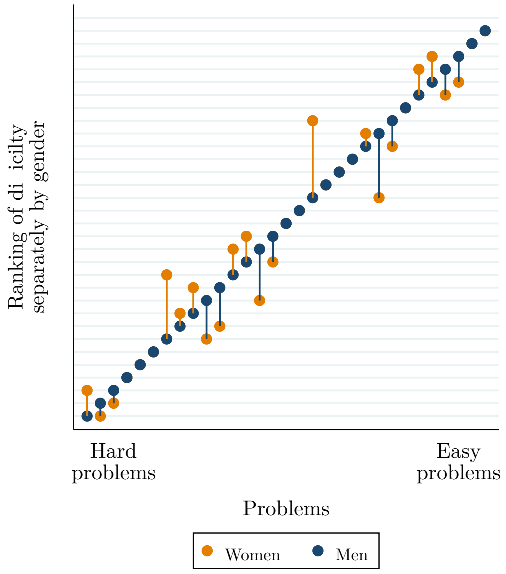


(a) Time relative treatment (in sessions)

(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 B12: Ranking of problems by gender



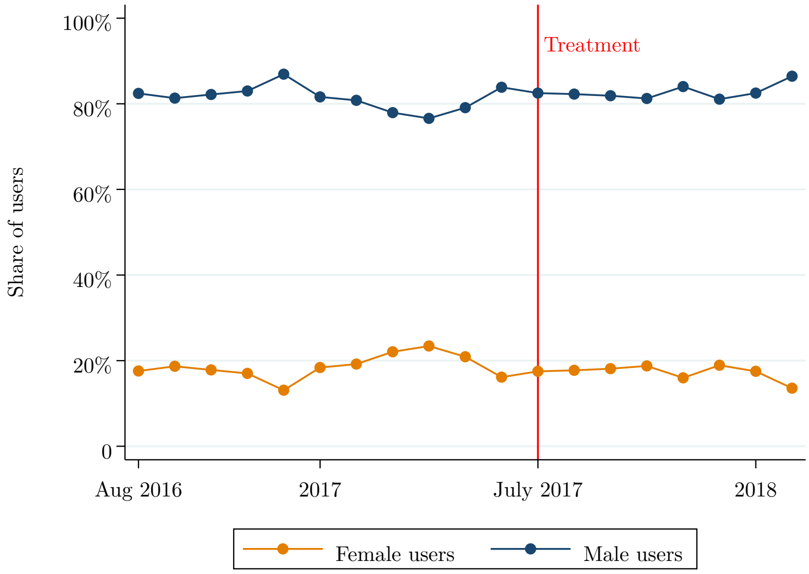
Notes: This figure shows the relative ranking of problems' difficulty by gender. The ranking is proxied by the average performance of users for each problem. The orange vertical lines show any positive or negative deviation of female users' ranking compared to male users' ranking.

Table B1: Problems' and Evaluators' Characteristics

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

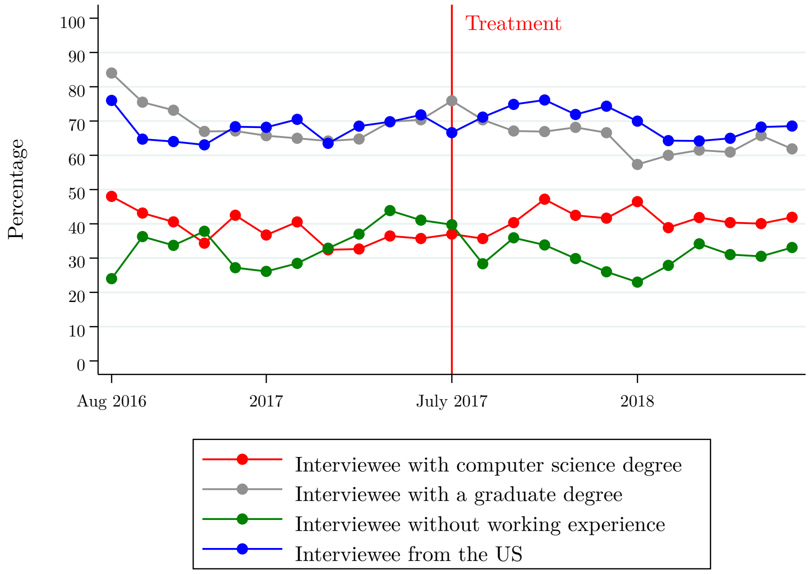
Notes: The regression TBC

Figure B13: Share of male and female users over time



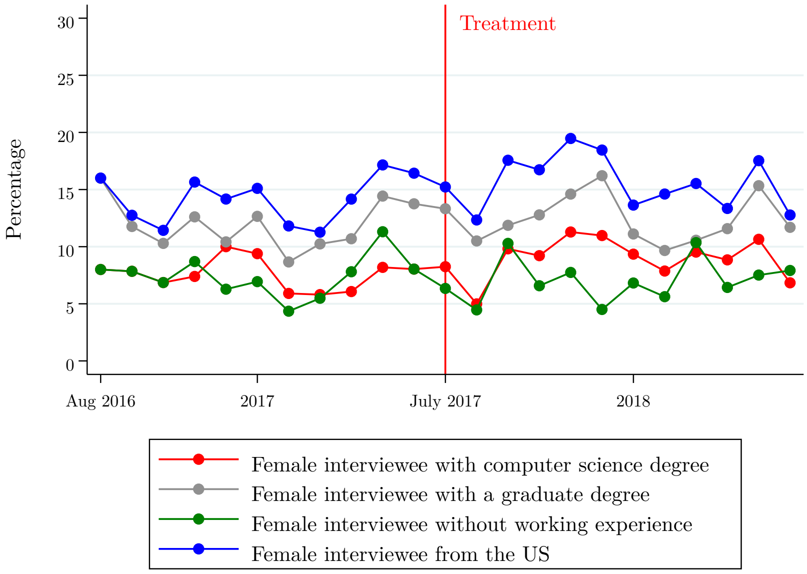
Notes: This figure shows the evolution of the shares of female and male users on the platform before and after the introduction of the device.

Figure B14: Evolution of First-Time Users' Characteristics



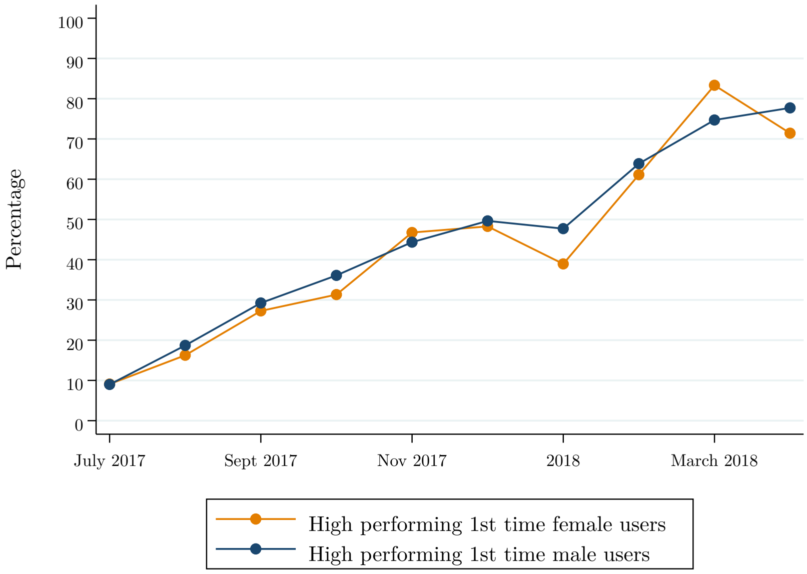
Notes: The figure presents the evolution of first-time users' characteristics averaged by month around the date of the introduction of the device on the platform.

Figure B15: 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 B16: 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

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

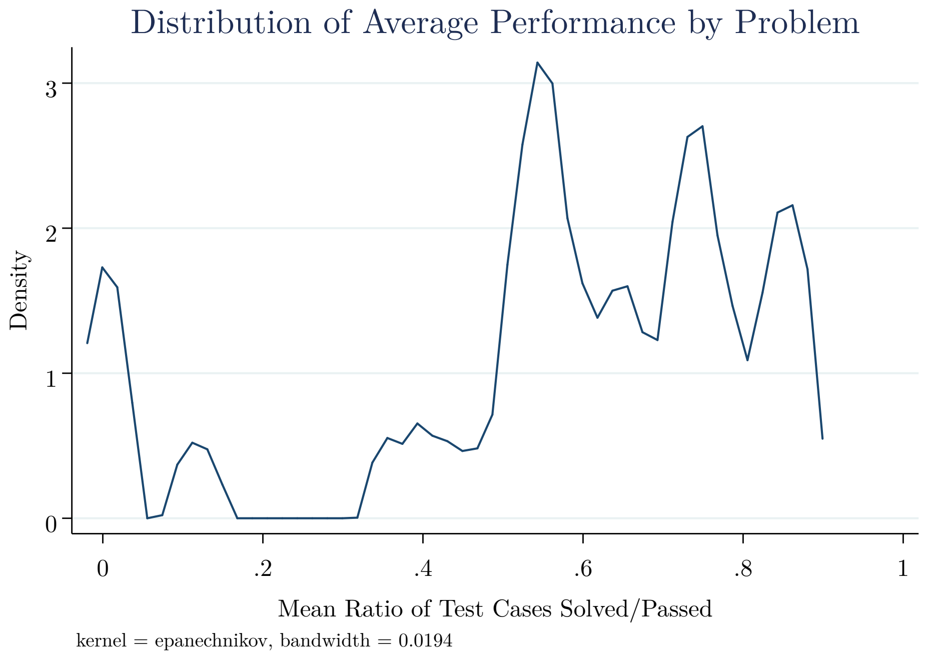
Panel B: Women

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

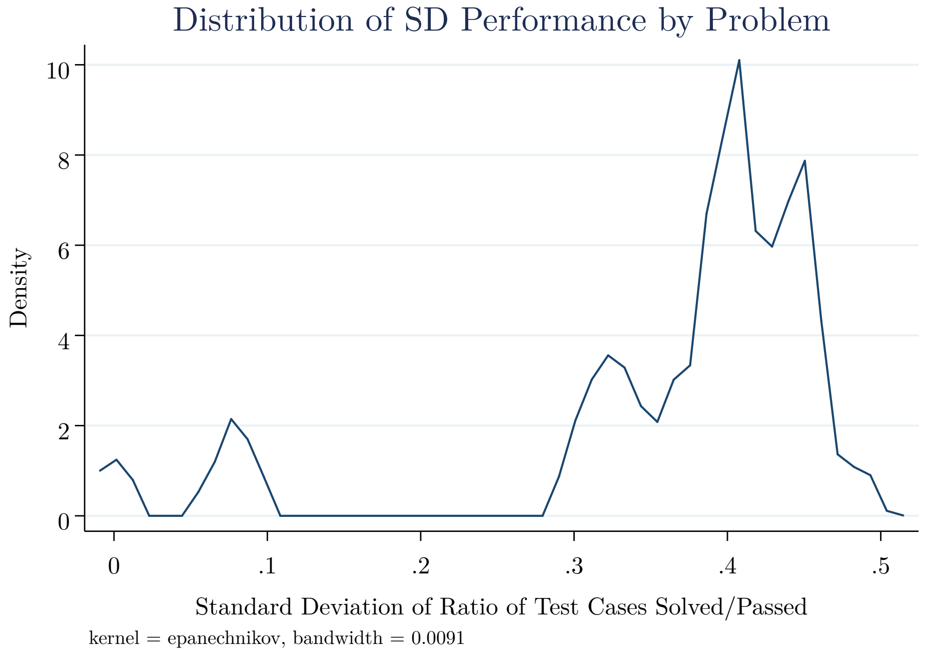
Panel C: Men

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

Figure B17: Variations across Problems



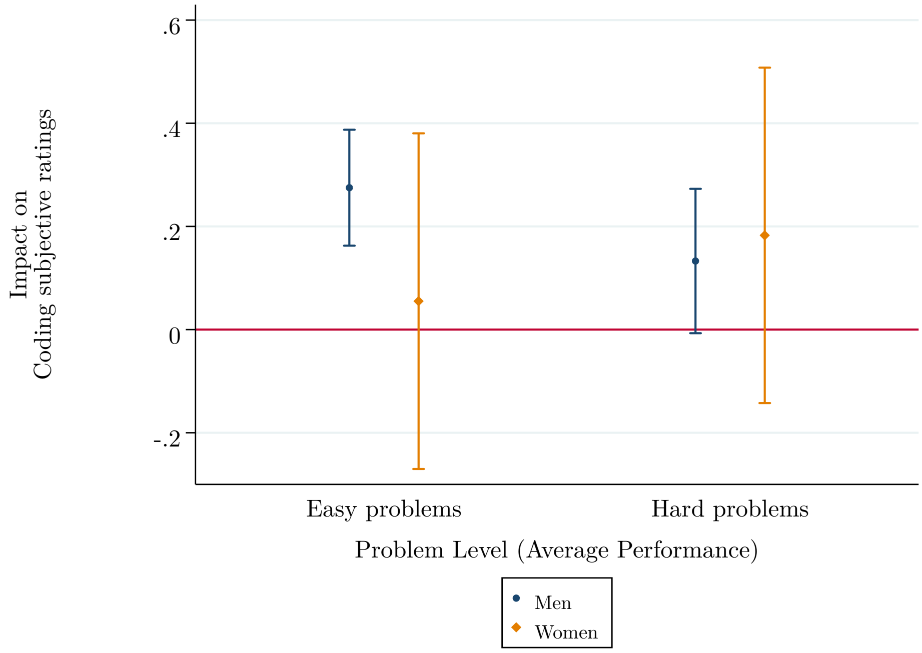
(a) Problem Average Difficulty



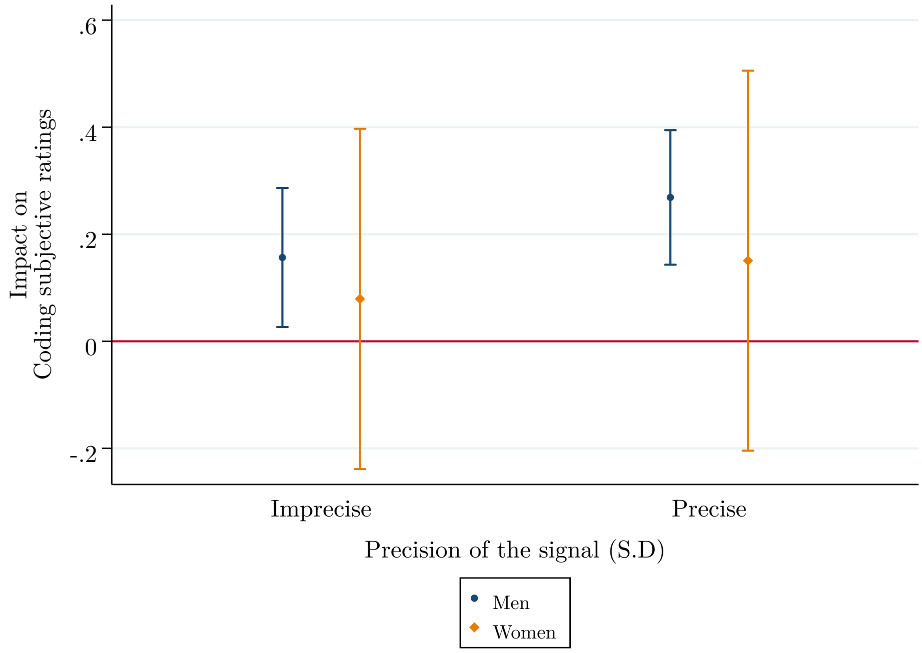
(b) Precision of the Signal

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 B18: 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 (5) 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
N	1,587	10,004		
Test of joint significance	F-stat: 1.100 (p-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***	0.186	0.177	0.166	0.212
	(0.015)		(0.007)	(0.016)	(0.008)
Country: USA	0.718***	0.684	0.681	0.684	0.693
	(0.010)		(0.008)	(0.021)	(0.010)
Interviewee's deg.: computer science	0.709***	0.652	0.660	0.649	0.663
	(0.011)		(0.008)	(0.021)	(0.009)
Interviewee's deg.: postgraduate	0.726***	0.431	0.434	0.450	0.424
	(0.011)		(0.008)	(0.021)	(0.009)
Interviewee's years of experience	0.736***	3.067	3.061	2.859	3.225
	(0.021)		(0.045)	(0.159)	(0.062)
Interviewee Preparation Level (self-declared on 1-5 scale)	0.621***	2.880	2.928	2.768	2.816
	(0.049)		(0.013)	(0.034)	(0.017)

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

* p<0.10, ** p<0.05, *** p<0.01

Table B5: 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
N	11,984	12,080		
Test of joint significance	F-stat: 1.800 (p-value: 0.078)			

Table B6: Gender gap in Subjective Coding Ratings and Interviewer's Learning

	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 B7: Labor Market Outcomes by Gender

	Ln(first salary post graduation)					
	Male			Female		
	(1)	(2)	(3)	(4)	(5)	(6)
Objective Performance	0.066**	0.067**	0.067**	0.002	0.001	-0.012
	(0.031)	(0.032)	(0.032)	(0.059)	(0.060)	(0.062)
Subjective Coding Rating		0.008	-0.008		-0.001	-0.117
		(0.038)	(0.044)		(0.084)	(0.096)
Communication Rating			0.029			0.199*
			(0.039)			(0.104)
Sociodemographic controls	Yes	Yes	Yes	Yes	Yes	Yes
City FE	Yes	Yes	Yes	Yes	Yes	Yes
Graduation Year FE	Yes	Yes	Yes	Yes	Yes	Yes
Controls	Yes	Yes	Yes	Yes	Yes	Yes
Observations	1,873	1,862	1,862	419	419	419

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 January 2018 to May 2022 (see Table **??**). 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. Of these, 18 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.

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

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

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

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.

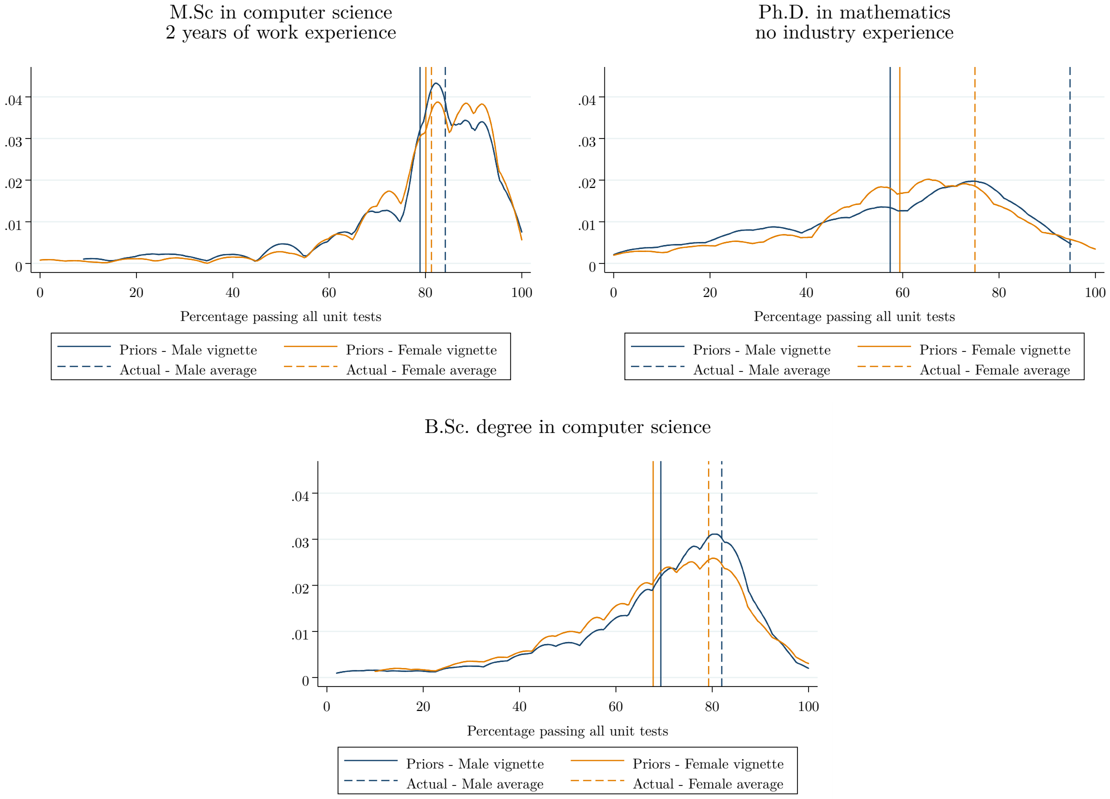
Measure of Priors To measure participants' priors, we exposed them to three different vignettes before they perform their evaluation tasks. We ask them to predict the potential performance of three different hypothetical coders. We cross-randomize the first name (alternating gender) and the skill level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

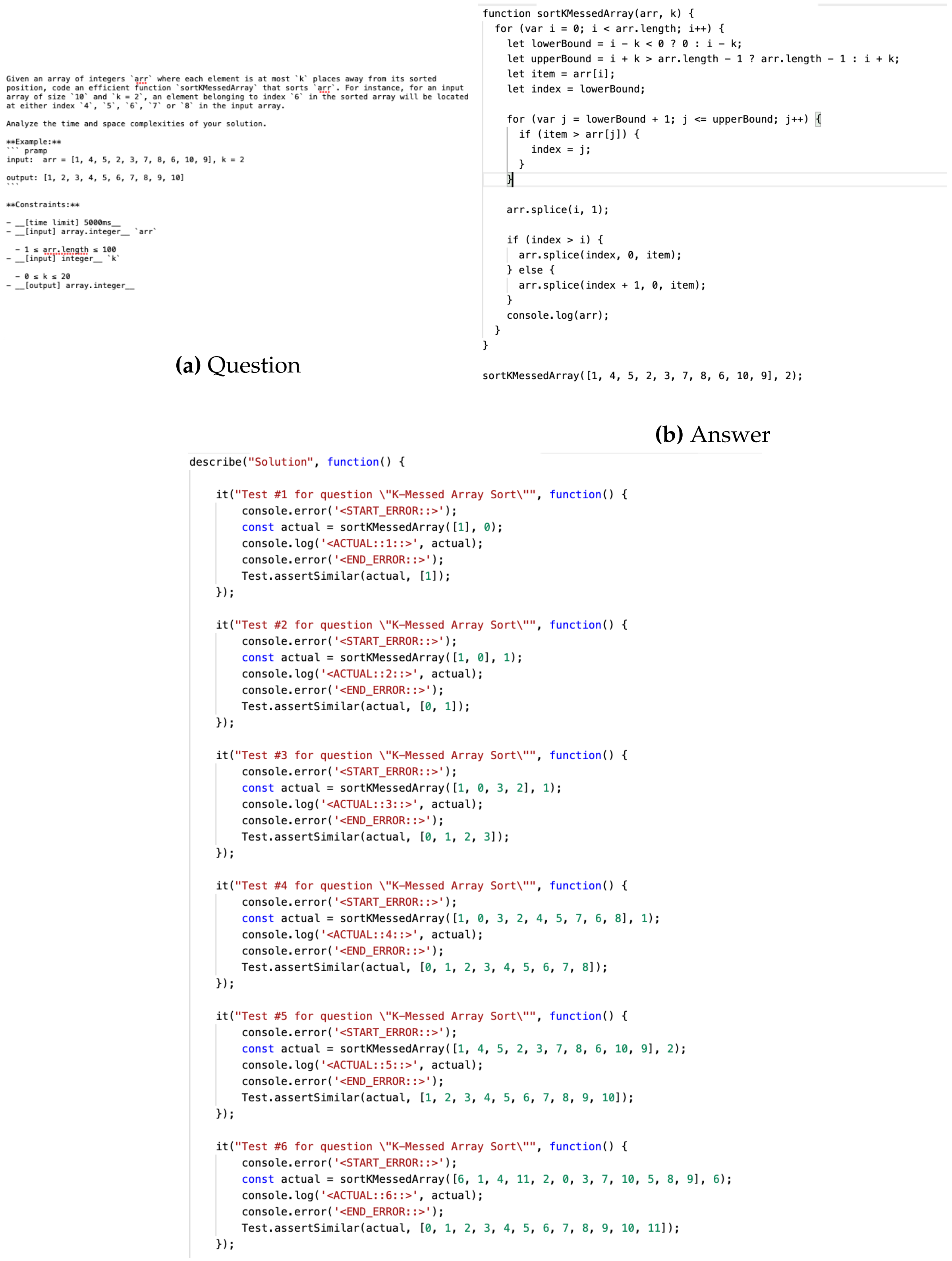
Skills	First names
a M.Sc in computer science and has 2 years of work experience	Katie/Tom
a Ph.D. in mathematics and has no industry experience	Alexa/Mickael
a B.Sc. degree in computer science	Corinne/Matt

Figure C19: 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 C20: 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 `sortKMessedArray` that sorts `arr`. For instance, for an input
array of size `10` and `k = 2`, an element belonging to index `6` in the sorted array will be located
at either index `4`, `5`, `6`, `7` or `8` in the input array.

Analyze the time and space complexities of your solution.

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

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

Constraints:

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

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

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

(a) Question

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

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

    arr.splice(i, 1);

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

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

(b) Answer

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

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

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

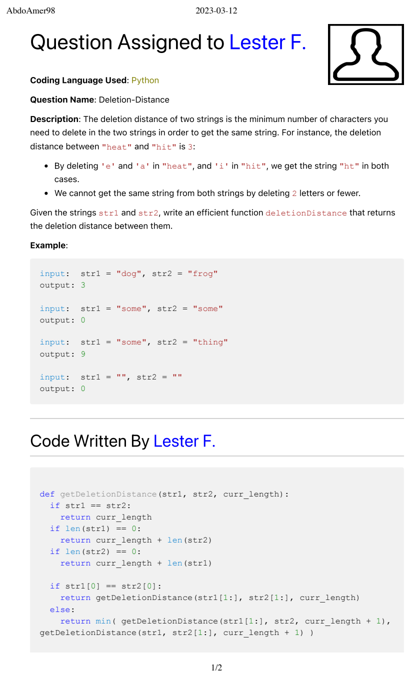
    it("Test #3 for question \"K-Messed Array Sort\"", function() {
        console.error('<START_ERROR::>');
        const actual = sortKMessedArray([1, 0, 3, 2], 1);
        console.log('<ACTUAL::3::>', actual);
        console.error('<END_ERROR::>');
        Test.assertSimilar(actual, [0, 1, 2, 3]);
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

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

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

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

(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 C21



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 "heat" and "hit" is 3:

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

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

**Example**:

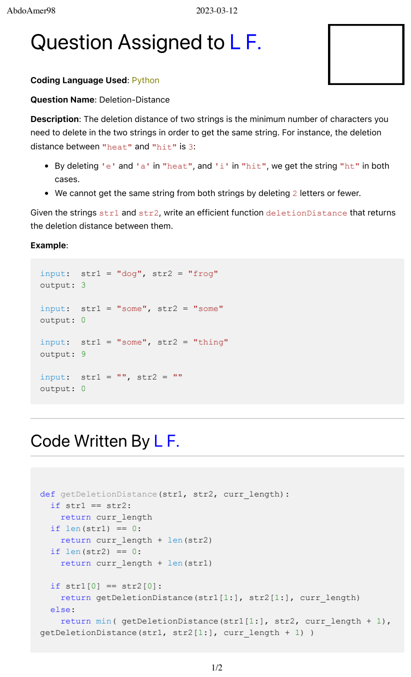
```
input:  str1 = "dog", str2 = "frog"
output: 3

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

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

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

Code Written By Lester F.

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

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

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 "heat" and "hit" is 3:

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

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

Example:

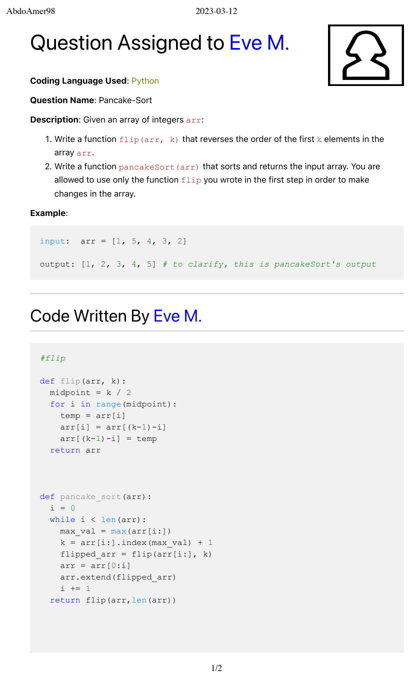
```
input:  str1 = "dog", str2 = "frog"
output: 3

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

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

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

Code Written By L F.

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

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

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:

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

Example:

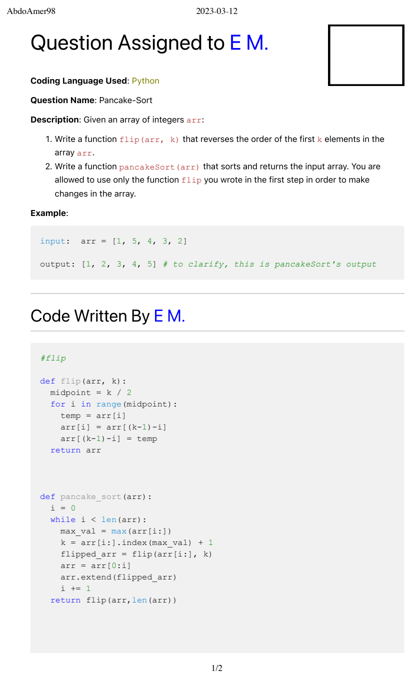
```
input:  arr = [1, 5, 4, 3, 2]

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

Code Written By Eve M.

```
#flip

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



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

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:

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

Example:

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

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

Code Written By E M.

```
#flip

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



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

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 C8: 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 C9: Descriptive Statistics — Sample Construction — January 2018-May 2022

	Obs.	Mean	Median	S.D
Raw Data				
Full score	203,769	0.34	1.00	0.39
Num code lines	482,390	44.12	40.00	37.45
Female	-	-	-	-
Non-white	-	-	-	-
Clean Data				
Full score	100,933	0.81	1.00	0.40
Num code lines	178,717	55.25	48.00	31.89
Female	-	-	-	-
Non-white	-	-	-	-
Experimental Sample				
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

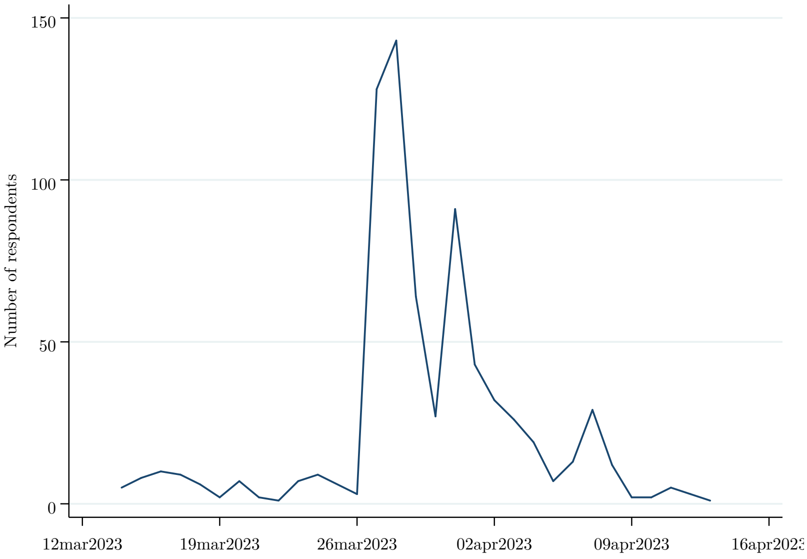
Table C10: Gender gap in subjective coding ratings, controlling for objective performance — Experimental Sample

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

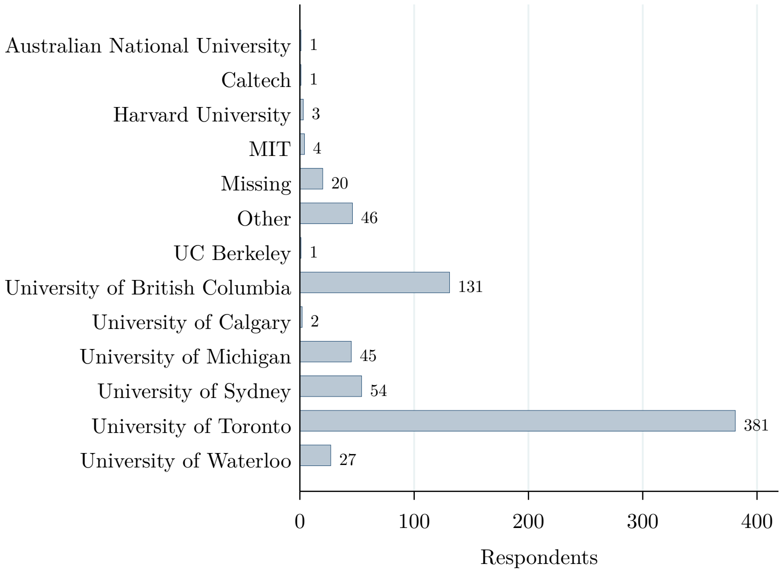
Note: This table shows the estimation of the gender gap in subjective ratings, controlling for objective performance measure (proxied by the ratio of test solved over passed by problem), using a linear regression model in which we progressively add controls. We progressively add interviewer and problem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

C.3 Descriptive Statistics: Evaluators

Figure C22: Number of respondents over time



(a) Number of respondents



(b) Number of respondents by institutions

Table C11: Descriptive Statistics — Participants

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

Table C12: Treatment-Control Balance — Whole sample

	Non-blind to Blind	Blind to Non-blind	Difference	*p*-value of diff.
	(1)	(2)	(3)	(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 C13: Treatment-Control Balance — Quality sample

	Non-blind to Blind	Blind to Non-blind	Difference	*p*-value of diff.
	(1)	(2)	(3)	(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 C14: 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	-0.057	-0.181	-0.161	-0.139***	-0.039
	(0.040)	(0.040)	(0.131)	(0.132)	(0.037)	(0.036)
Treatment order	-0.001		0.056		-0.115**	
	(0.041)		(0.138)		(0.044)	
Script 1	-0.262***	-0.260***	-0.368*	-0.346	-0.156**	-0.012
	(0.057)	(0.058)	(0.179)	(0.183)	(0.051)	(0.051)
Script 2	-0.098	-0.092	-0.206	-0.170	-0.199***	-0.054
	(0.058)	(0.058)	(0.181)	(0.181)	(0.049)	(0.048)
Script 3	-0.022	-0.019	-0.267	-0.276	-0.080	-0.067
	(0.059)	(0.059)	(0.184)	(0.185)	(0.050)	(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 C15: 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	-0.065	-0.230	-0.201	-0.078	-0.038
	(0.045)	(0.045)	(0.150)	(0.151)	(0.044)	(0.043)
Treatment order	-0.013		0.094		-0.030	
	(0.046)		(0.153)		(0.045)	
Script 1	-0.242***	-0.239***	-0.350	-0.327	-0.121	-0.061
	(0.063)	(0.064)	(0.204)	(0.207)	(0.062)	(0.062)
Script 2	-0.090	-0.080	-0.184	-0.153	-0.129*	-0.064
	(0.065)	(0.065)	(0.204)	(0.204)	(0.059)	(0.059)
Script 3	0.019	0.022	-0.164	-0.175	-0.046	-0.037
	(0.066)	(0.067)	(0.209)	(0.209)	(0.059)	(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 C16: Quality Measures

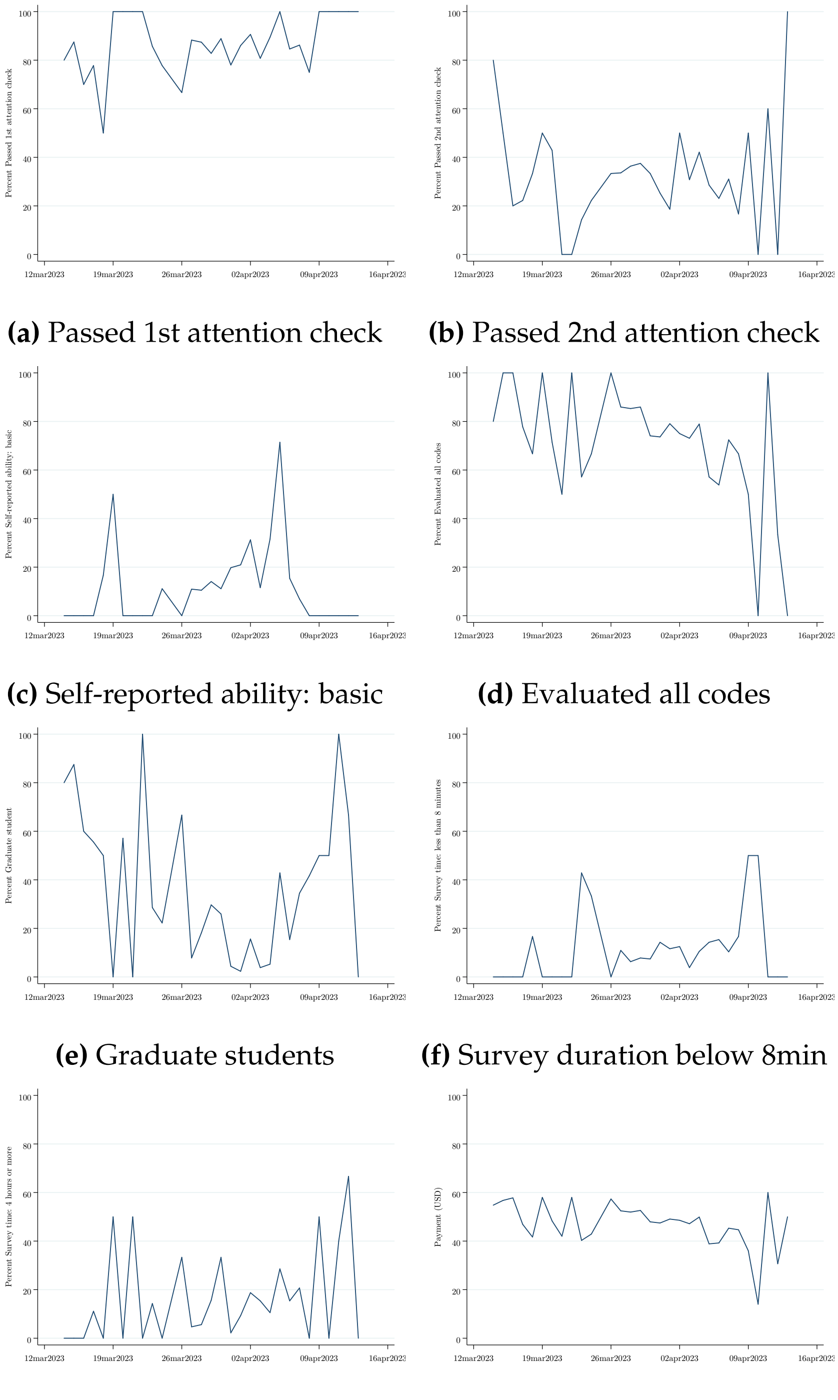
	Mean	**Std. Dev.**	**N**
Passed 1st attention check	0.852	0.355	716
Passed 2nd attention check	0.327	0.469	716
Self-reported ability: basic	0.138	0.345	716
Evaluated all codes	0.793	0.405	716
Graduate student	0.194	0.396	716
Survey time: less than 8 minutes	0.101	0.301	716
Survey time: 4 hours or more	0.099	0.299	716

Table C17: Racial Gap in Subjective Coding Ratings, Controlling for Objective Performance

	Subjective Coding Ratings					
Panel A	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
Panel B	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.

Figure C23: Quality Checks



(a) Passed 1st attention check **(b)** Passed 2nd attention check

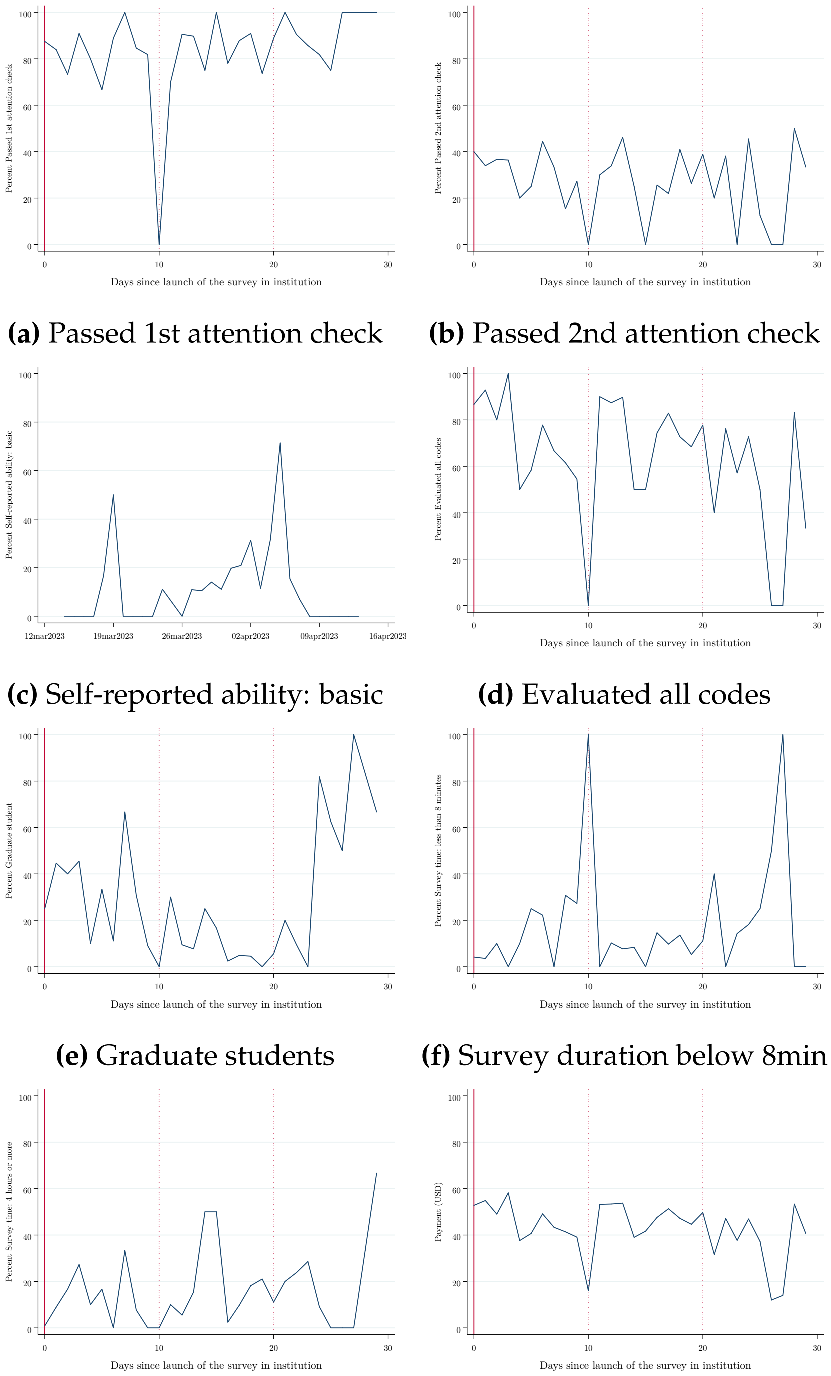
(c) Self-reported ability: basic **(d)** Evaluated all codes

(e) Graduate students **(f)** Survey duration below 8min

(g) Survey duration above 4H **(h)** Final payment (USD)

Notes: This figure presents trends of several quality check measures over time.

Figure C24: Quality Checks since Launch Date in Institution



(a) Passed 1st attention check **(b)** Passed 2nd attention check

(c) Self-reported ability: basic **(d)** Evaluated all codes

(e) Graduate students **(f)** Survey duration below 8min

(g) Survey duration above 4H **(h)** Final payment (USD)

Notes: This figure presents trends of several quality check measures over time since the start of the survey experiment in each institution.

Table C18: Blinding Experiment — Main Results Racial Gaps

	Subjective Coding Ratings					
	Reviewer 1		Reviewer 2		Algorithmic Prediction	
	(1)	(2)	(3)	(4)	(5)	(6)
Non-blind code	-0.123	-0.152*	-0.149*	-0.162**	-0.170**	-0.171*
	(0.079)	(0.085)	(0.077)	(0.081)	(0.086)	(0.094)
Non-blind code×Female code	0.143	0.147	0.097	0.107	0.129	0.099
	(0.104)	(0.111)	(0.097)	(0.101)	(0.112)	(0.119)
White or East Asian	0.087	0.097	0.017	0.052	-0.006	0.014
	(0.060)	(0.070)	(0.062)	(0.068)	(0.061)	(0.072)
Non-blind code	0.081	0.127	0.139	0.160	0.146	0.144
×White or East Asian	(0.104)	(0.121)	(0.062)	(0.068)	(0.103)	(0.120)
Non-blind code	-0.174	-0.161	-0.098	-0.089	-0.137	-0.071
×White or East Asian×Female	(0.119)	(0.138)	(0.116)	(0.135)	(0.119)	(0.138)
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 investigates gender and racial disparities on final ratings, where the main racial category is white or East Asian. The even columns include evaluator fixed effects. Standard errors are 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. If you have questions about your rights as participants, you can contact the Research Oversight and Compliance Office — Human Research Ethics Program at ethics.review@utoronto.ca or 416-946-3273. You can also contact the University of Michigan IRB (Health Sciences and Behavioral Sciences) at 734-936-0933 or irbhsbs@umich.edu, quoting eResearch #HUM00204184.

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

Compensation

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

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

Yes, I would like to voluntarily participate in this experiment.

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