Feld, Jan; Ip, Edwin; Leibbrandt, Andreas; Vecci, Joseph

Working Paper

Identifying and Overcoming Gender Barriers in Tech: A Field Experiment on Inaccurate Statistical Discrimination

CESifo Working Paper, No. 9970

Provided in Cooperation with:
Ifo Institute – Leibniz Institute for Economic Research at the University of Munich

Suggested Citation: Feld, Jan; Ip, Edwin; Leibbrandt, Andreas; Vecci, Joseph (2022) : Identifying and Overcoming Gender Barriers in Tech: A Field Experiment on Inaccurate Statistical Discrimination, CESifo Working Paper, No. 9970, Center for Economic Studies and ifo Institute (CESifo), Munich

This Version is available at:
https://hdl.handle.net/10419/266005

CESifo WORKING PAPERS

9970
2022

September 2022

Identifying and Overcoming Gender Barriers in Tech: A Field Experiment on Inaccurate Statistical Discrimination

Jan Feld, Edwin Ip, Andreas Leibbrandt, Joseph Vecci

CESifo

Impressum:

CESifo Working Papers
ISSN 2364-1428 (electronic version)
Publisher and distributor: Munich Society for the Promotion of Economic Research - CESifo GmbH
The international platform of Ludwigs-Maximilians University's Center for Economic Studies and the ifo Institute
Poschingerstr. 5, 81679 Munich, Germany
Telephone +49 (0)89 2180-2740, Telefax +49 (0)89 2180-17845, email office@cesifo.de
Editor: Clemens Fuest
https://www.cesifo.org/en/wp
An electronic version of the paper may be downloaded
· from the SSRN website: www.SSRN.com
· from the RePEc website: www.RePEc.org
· from the CESifo website: https://www.cesifo.org/en/wp

CESifo *Working Paper No. 9970*

Identifying and Overcoming Gender Barriers in Tech: A Field Experiment on Inaccurate Statistical Discrimination

Abstract

Women are significantly underrepresented in the technology sector. We design a field experiment to identify statistical discrimination in job applicant assessments and test treatments to help improve hiring of the best applicants. In our experiment, we measure the programming skills of job applicants for a programming job. Then, we recruit a sample of employers consisting of human resource and tech professionals and incentivize them to assess the performance of these applicants based on their resumes. We find evidence consistent with inaccurate statistical discrimination: while there are no significant gender differences in performance, employers believe that female programmers perform worse than male programmers. This belief is strongest among female employers, who are more prone to selection neglect than male employers. We also find experimental evidence that statistical discrimination can be mitigated. In two treatments, in which we provide assessors with additional information on the applicants' aptitude or personality, we find no gender differences in the perceived applicant performance. Together, these findings show the malleability of statistical discrimination and provide levers to improve hiring and reduce gender imbalance.

JEL-Codes: C930, J230, J710, J780.

Keywords: field experiment, discrimination, beliefs, gender.

Jan Feld
Victoria University / Wellington / New Zealand
jan.feld@vuw.ac.nz

Edwin Ip
University of Exeter / United Kingdom
e.ip@exeter.ac.uk

Andreas Leibbrandt
Monash University / Clayton / VIC / Australia
andreas.leibbrandt@monash.edu

Joseph Vecci
Gothenburg University / Sweden
Joseph.vecci@gu.se

This version: September 16, 2022

Leibbrandt acknowledges support from the Australian Research Council. Vecci acknowledges support from the Swedish Research Council (Project No. 2018-04793). We thank Andreas Drechsler, Laurent Faucheux and Brett Wilson for providing us with important insights on the programming profession. We thank Mallory Avery, Loukas Balafoutas and Derek Rury for their helpful feedback. We thank participants at conferences including AFE 2022, SABE 2022, ESA World Meeting 2022, and seminars at the University of Exeter, Google, Victoria University at Wellington, The Swedish Institute for Social Research, and Gothenburg University.

1. Introduction

Great strides have been made in many labor markets to reduce gender imbalances.[1] However, the technology sector (tech) is a notorious exception. In tech, women are substantially underrepresented and there are few encouraging signs for improvement. Over the last 10 years there has been a decrease in the number of women studying key tech degrees like computer science and if they do, they are less and less likely to work in tech than men.[2] These trends do not only manifest in gender inequalities in this important labor market, which represents more than half of all STEM jobs (Pew Research Center, 2021), but also likely undermine the efficient allocation of talent.

Several explanations have been proposed for why women remain underrepresented, including discrimination (Bertrand and Duflo, 2017, Neumark, 2018) and the idea that there may be average differences in skills (Aigner and Cain, 1977).[3] These two explanations intersect in the concept of statistical discrimination (Phelps, 1972; Arrow, 1973; Aigner and Cain, 1977) where uncertainty about skills causes employers to prefer hiring men due to beliefs about gender differences in skills. Importantly, these beliefs about gender differences can be accurate (i.e., men are actually more skilled than women) or inaccurate (e.g. there are no gender differences in skills) (Bordalo et al., 2016; Bohren et al. 2020; Mengel and Campos-Mercade, 2021; Lepage, 2021; Chan 2022).[4]

Identifying the role of these different sources of discrimination is crucial for evaluating existing policies and for developing new policies which aim to improve efficiencies and reduce imbalances. For instance, Gertsberg et al. (2022) show that share prices react negatively to gender quotas and Ip et al. (2020) show that gender quotas have little public support and backfire if women are believed to be less skilled than men. It is therefore important to study whether there are actual skill differences between men and women *and* simultaneously capture beliefs about gender differences in these skills.

[1] For instance, around 25 per cent of all national parliamentarians are women, up from 11 per cent in 1995 (UN Women, 2021). While 19.7% of corporate board seats are occupied by women up from 15% in 2015 (Deloitte, 2021)

[2] In the United States, 19% who earned a B.S. in computer science in 2016 are women, down from 27% in 1997 (NSF, 2019); Women as compared to men with computer science degrees are less likely to work in the field (38% vs. 53%) (NSF, 2019).

[3] Other explanations include motherhood (Petit, 2007; Correll et al, 2007).

[4] In addition, there are other types of discrimination that can cause inefficiencies and explain why there are gender barriers in tech, for example, taste-based (Becker, 1957) and attention-based (Bartos et al, 2016). There are also more indirect forms of discrimination such as stickiness in the belief updating process (Sarsons, 2017), and systemic discrimination (Bohren et al., 2022).

We present a labor market field experiment with over 2,000 job applicants in tech and over 600 professionals with significant hiring experience in tech. Our experimental design makes it possible to both measure actual programming skills of job applicants and identify the beliefs of professionals about the skills of applicants. Our pre-registered experiment consists of two stages. In the first stage, we advertised two programmer jobs on major job sites in the United States and comprehensively measured job applicants' actual skills using standardized and expert-validated programming tasks and evaluation methods. In addition, applicants were randomized into two treatments following the programming tasks, where they either completed a commonly used personality assessment or an aptitude test for programmers.

In the second stage, we incentivized professionals to report their beliefs about the skills of these job applicants. We recruited programmers and HR professionals, almost 90% of whom were involved in hiring programmers in their regular jobs, as participants ("employers") and provided them with information about the job advertisement, the programming tasks and how the programming task was scored. We then elicited each employer's beliefs about applicants' scores in the coding task based on a profile consisting of basic information from the applicants' resumes (first name, education, years of experience, etc.). Employers were incentivized to accurately guess applicants' scores: the closer their guesses were to the actual scores, the higher their chance of receiving a large monetary bonus.

Employers were randomized into three treatments. In the baseline treatment, they only had access to basic information from the applicants' resume; in the aptitude treatment they also received information on applicants' aptitude assessment; and in the personality treatment they received information on applicants' personality assessment. Despite the popularity of these assessments with employers, it is not clear how predictive they are of an applicant's skills. Aptitude tests are designed to be informative about the applicant's skills whereas personality assessment may be informative as programmers are perceived to have certain personality traits (Ehrlinger et al. 2018). As a result, an applicant's aptitude may offer a *direct* predictor of skills and an applicant's personality traits may offer an *indirect* predictor of skills. Therefore, these assessments should provide employers with different types of potentially productivity-relevant information to reduce uncertainty about job applicants' skills. This information could affect gendered beliefs and mitigate statistical discrimination against women, as there would be a reduced need to rely on gender to form beliefs. Finally, we used incentivized procedures to ask employers about their beliefs about the distribution of

skills by gender in the applicant sample and in the general population, which allow us to explore potential reasons for differences in beliefs.

This study makes three main contributions. *First*, we use a field experiment that captures both actual and perceived job skills of professionals who have selected into an important, male-dominant industry. Thus, we study actual actors in a labor market, which reduces the risk of drawing misleading inferences from behavior in convenience samples, where no job specific selection took place and it is unclear how selection affects behavior.[5] By identifying actual and perceived job skills within the same group of job applicants, we can also provide direct evidence on the extent of inaccurate beliefs about gender differences after selection into a profession, which then renders it possible to study the basis of inaccurate statistical discrimination in one of the most important labor markets with gender disparities. *Second*, we show that inaccurate beliefs can be corrected by supplementing resumes with information on applicants' personality or aptitude. *Third*, we investigate the source of inaccurate beliefs by testing the role of the representative heuristic (Bordalo et al., 2016), selection neglect (Fiedler, 2000) and attention discrimination (Bartos et al., 2016).

Our findings reveal a striking difference between actual and perceived skills that disadvantages women. We do not find significant differences between male and female applicants in their actual coding skills. However, in our baseline, we do find that employers believed that female applicants are significantly less skilled than male applicants. Employers believed that female applicants' score in the programming task is 0.12 deviations (SD) worse than their male counterparts ($p<0.001$) and after controlling for applicant characteristics this gap increases up to 0.39 SD. We also find that these gender differences in beliefs are larger for female than male employers.

There is a silver lining. We show that beliefs about gender differences in coding skills disappear in our two information treatments, which suggests that employers used gender to statistically discriminate in the absence of sufficient information. We observe that giving employers information on the applicants' personality or aptitude assessment led to less gender differences in skill assessments: the perceived difference between male and female applicants' programming performance is only 0.01 SD when employers received information

[5] There are at least two reasons why there could be important differences between subjects in convenience samples and actual applicants and employers. First, job applicants have selected into a profession based on their skills and interests. While it easy to imagine that there are gender differences in programming skills among, for example, students in a laboratory, it is less clear there would be gender differences in programming skills among women and men who have decided to become programmers. Second, employers are regularly exposed to female and male programmers. Such exposure might allow them to learn about the true gender differences in programming skills. In contrast, most students in the laboratory did not have the opportunity for such learning.

on applicants' personality scores; and the sign reverses when employers received information on applicants' aptitude with employers believing that women are on average 0.05 SD better programmers (both not significant).

We also provide evidence on the role of different potential sources for the perceived gender skill differences. We show that our findings square well with selection neglect. Consistent with selection neglect, we show that employers insufficiently distinguished between gender skill differences in the population (which likely exists in programming) and in the applicant sample. Interestingly, we find that female employers did not adjust for selection at all. Selection neglect may therefore explain why female employers are more likely to believe that female applicants are less skilled than male applicants. We do not find evidence for attention discrimination or representativeness heuristics in our context.

Our field experimental approach complements the empirical literature studying gender discrimination, which is largely based on observational data (Bertrand and Duflo, 2017, Neumark, 2018), audit studies (e.g. Neumark, et al., 1996; Kübler et al. 2018; Kline et al., 2021), laboratory experiments (e.g., Lane, 2016; Holm, 2000; Fershtman and Gneezy, 2001; Slonim and Guillen 2010), and some field experiments (e.g. List, 2004; Delfino, 2021). These studies have identified the existence and extent of gender discrimination in many labor markets but often struggle to disentangle different types of discrimination and only few studies provide evidence for the role of beliefs (e.g. Bohren et al., 2019; Coffman et al., 2021). Bohren et al. (2020) provide a notable exception. Using a Mturk experiment, they show that inaccurate statistical discrimination can easily be mis-specified as taste-based discrimination.[6] Accounting for the main drivers and types of discrimination in the field is important to avoid policies and practices that further disadvantage women (Hoogendoorn and Van Praag, 2012; Hoogendoorn et al. 2013; Besley et al., 2017; Ip et al., 2020). We use a design which renders it possible to uncover the foundation and type of statistical discrimination of hiring professionals in key labor markets for women.

Our paper also contributes to the literature on gendered beliefs by studying possible solutions to inaccurate beliefs that disadvantage women. Research has shown that an increase

[6] Bohren et al. (2020) presented theoretical evidence for the role of inaccurate statistical discrimination and tested their theoretical predictions in an online experiment on MTurk. They investigated discrimination by nationality and gender in an experiment where some participants were assigned the role of employers and others were assigned the role of workers performing a simple math task. Their results are consistent with inaccurate statistical discrimination. For example, they showed that "employers" were less likely to hire American than Indian "workers" although Americans and Indians performed equally well on the task. Under the assumption of accurate beliefs, such a result would show evidence for taste-based discrimination. However, the authors further showed that participants wrongly believed that Indian workers were better and that what appeared to be taste-based discrimination is to a large extent statistical discrimination based on inaccurate beliefs.

in contact between groups can reduce negative beliefs (Paluck et al. 2019; Lowe, 2021; Rao, 2019; Scacco and Warren, 2018), encouraging role models can change perceptions (Dee, 2005; Fairlie et al., 2014; Olivetti et al., 2020) and that social recategorization can decrease animosity (Kawakami et al., 2007; Forbes and Schmader, 2010). As statistical discrimination is based on uncertainty about productivity, providing relevant information should reduce the reliance on (possibly inaccurate) beliefs about gender. Our experiment allows to test whether information provided to employers from aptitude and personality tests actually reduces the role of inaccurate beliefs in hiring decisions.

2. Experimental Design

Our experimental design allows us to provide insights into current key questions on the role and type of statistical discrimination for gender gaps in labor markets. We measure both actual and perceived gaps in programming skills between male and female programmers while minimizing the role of taste-based discrimination. The experiment consists of two stages. In stage 1, we comprehensively measure the actual programming skills of female and male programmers. In stage 2, we measure employers' perceptions of the programming skills of these programmers and conduct a comprehensive experiment to investigate these perceptions. We pre-registered the experiment at the AEA registry (AEARCTR-0004227) and received ethics approval.[7]

For stage 1, we advertised a Python programming job across major job sites in the United States, including general job sites (e.g. indeed.com) and specialized tech job sites (e.g. Dice, Crunchboard, Github). The job consisted of 80 hours of programming work over 2 months at US$40 per hour and was open to anyone who was based in the United States.[8] To apply, applicants had to upload their resume and fill out a short form asking, among other things, about their demographic and contact information as well as how they learned to program. We advertised the same job twice (September 2019 and February 2020), using

[7] This project received ethics approval from Monash University in 2018 (Project ID: 14985).
[8] Contract work is common in the tech industry from small start-ups to large corporations. For instance, contract workers have outnumbered direct employees at Google since 2018 (e.g. Sheng, 2018). Research conducted by Upwork (2020) finds that 35% of Americans take up freelance work in 2019, with 45% of them providing skilled services such as programming and IT. The 2021 version of the study finds that 53% of Computers/Mathematics professionals carry out freelance work (Upwork, 2021a). The market for contract tech workers has become even more important recently, with 80% of hiring managers increasing their use of tech freelancers since the onset of COVID (Upwork 2021b).

identical protocols. Each time, we posted the job ad for one month. We hired one programmer in each wave.[9]

We received 2,183 applicants who met our criteria for inclusion in our analysis. The criteria were that they filled in a survey with their email address, lived in the United States, indicated that they know how to program in Python, were either female or male, and were not excluded for other reasons (e.g. they did not apply for job 1 *and* job 2). Some applicants that we excluded from our data analysis (e.g. non-binary applicants) were still considered for the job. As specified in the pre-analysis plan, we invited all eligible female applicants (n=310) and a random sample of all eligible male applicants (n=1,298) to complete an online skill assessment.

All invited applicants were invited to complete a Python test. In addition, they were randomized into completing either an aptitude test or a personality test. For a separate, pre-registered study, we randomize all applicants into one of two treatments, one where applicants were *not* offered a financial incentive for completing the tests, and one where they were offered a flat fee for completing the test and a performance incentive.[10] In this paper, as outlined in our pre-analysis plan, we will only consider the applicants who were unincentivized, which represents the most natural job application environment. However, we will show that our results are robust to the inclusion of data from the incentivized treatment.[11]

In total, we invited 816 applicants to perform the Python test. Of those, 331 attempted the test. Of those who attempted, we excluded 13 applicants for other reasons (e.g. we could not match the applicant and skills assessment data because applicants used different email addresses when applying at these stages, see appendix A1.1). Our main applicant sample therefore consists of 318 applicants who attempted the Python test and whose profiles are used in the stage 2 experiment. Not completing a job assessment is a natural part of the job process and therefore representative of what happens in the field.[12] Table 1 shows the summary statistics of all applicants from our main sample.[13] More than half the sample are employed and 48% have finished a 4-year college degree and 28% are still studying. Further,

[9] As outlined in an amendment to our pre-analysis plan, we posted a second job as we had fewer female applicants than we specified as a goal in the pre-registration. We decided to post a second job (and we also made the pre-analysis plan amendment) before inviting any applicants to take the skill assessment. This procedure ensures our data collection efforts are not driven by our results but by concerns about low sample size.
[10] This study has been separately pre-registered at AEARCTR-0004625.
[11] We show in Table A2 that our first stage findings are robust to the inclusion of the incentive treatment.
[12] For instance, some people who apply for the job may not have sufficient skills to complete the assessment. While data on applicant attrition during the job application process is rare, recent research from over 200,000 job applicants suggests similar rates of dropout (Hardy et al, 2017, Hartwell et al, 2020).
[13] Summary statistics of all invited applicants can be found in Table A1 in the Appendix.

on average applicants have 5.9 years of coding experience, with most learning to code at university.

Table 1: Summary Statistics Main Applicant Sample

	(1) N	(2) mean	(3) sd	(4) min	(5) max
Female	318	0.220	0.415	0	1
Currently studying	318	0.267	0.443	0	1
Currently employed	318	0.531	0.500	0	1
Education					
High school without graduation	318	0.0157	0.125	0	1
High school graduate	318	0.0440	0.205	0	1
Some college	318	0.182	0.387	0	1
2-year college	318	0.0440	0.205	0	1
4-year college	318	0.487	0.501	0	1
Postgrad	318	0.226	0.419	0	1
Years coding exper.	318	5.902	5.982	0	38
Learned coding					
in university	318	0.852	0.355	0	1
in online course	318	0.280	0.450	0	1
self-taught	318	0.544	0.499	0	1
other way	318	0.211	0.408	0	1

2.1 The Skill Assessment

The skill assessment was implemented by Mettl, a global leader in online assessments. Their assessment tools have been used by many organizations, such as Accenture, Barclays and Pepsico. The assessment included two parts. The first part was a Python programming test. The second part was randomly selected to be either an aptitude test or a personality test.

The Python test is the basis for our measure of programming skills. It was implemented in Mettl's online coding simulator and consisted of two tasks which applicants had to complete in 115 minutes.

Mettl has a large data bank of Python programming tasks. To help us with the selection of the tasks, we surveyed 15 professional Python programmers. Each of these programmers was shown our job description, 3 commonly used long tasks and 4 commonly used short tasks. Our Python test contained the two programming tasks (one short one and

one long one) which were rated as being most useful for predicting a job applicant's Python programming skills on the job.

Applicants' codes were assessed by a bespoke program to determine their scores. The overall score of the Python test can range from 0 to 100 and it is the average of the following five sub-scores. The *test cases* score measures if the programs perform their functions correctly. The *efficiency score measures* how fast a code finishes a test case (conditional on it giving the correct answer). The *complexity score* measures how complex the code is, with less complex code generally being considered better as it is easier to test and maintain. The *coding convention score* measures to what extent the code follows popular coding conventions. Following coding conventions makes code easier to understand by other programmers. The *frequency of errors score* flags likely coding errors. Each of these sub-scores can range from 0 to 100. We calculated the total score as weighted average of all five sub-scores with the weights being determined by the 15 professional programmers in the survey. The details on measurement and the weight of each sub-score can be found in Table 2. We pre-registered how we would score the test and did not deviate from our procedure. We describe the Python programmer survey and scoring in greater detail in Appendix A2.

Table 2: Scoring of Python Programming Test

Component	Measurement	Weight
Test case score	Each code is run over 10 test cases. Code that correctly solves more test cases is given a higher score. This score is provided by Mettl.	28.5%
Efficiency score	The time to complete each test case (conditional on it giving the correct answer). This is automatically measured by Mettl.	21.4%
Complexity score	A program (Pylint) is used to analyse the applicant's code and calculate the McCabe's Cyclomatic complexity score.	21.1%
Coding convention score	A program (Pylint) is used to analyse whether the applicant's code follows coding conventions as outlined in the style guide for Python code PEP 8.	11.5%
Frequency of error score	A program (Pylint) is used to measure the number of programming errors in the applicant's code.	17.5%

To assess the perceived relevance of the python test by employers, we asked participants in stage 2 of our experiment, which consist of programmers and human resource (HR) professionals (see next subsection), whether they thought the programming test was "a good measure of applicants' programming ability". We find that 93.44% answered yes

(91.03% for programmers and 95.85% for HR). This suggests that the assessment is perceived as a useful proxy of programming skill.

The aptitude test consists of five sub-scores: cognitive abilities, abstract reasoning, critical thinking, reasoning ability and attention to detail. It contained 43 questions in total and applicants took on average 14 minutes to complete this test. The aptitude test was the standard aptitude test for programmers used by Mettl and it was directly scored by them.

The personality test consists of an 86-item Big Five personality test that captured applicants' agreeableness, conscientiousness, emotional stability, extraversion and openness to experience. This is a standard personality test used by employers and it was scored directly by Mettl. Applicants took on average 15 minutes to complete this test.

2.2 Stage 2: Measuring Perceived Programming Skill

In stage 2, we measured how a sample of employers perceived the skills of the applicants from stage 1. We collected data from 625 employers consisting of 311 programmers and 314 HR professionals who work in tech.[14] Nearly 90% of them have been involved in hiring programmers in their jobs. We selected both these samples as these are the two groups generally responsible for hiring, yet they have different skills and experiences which may translate into different beliefs. Both types of employers were recruited using a panel service provided by Qualtrics. We paid each employer $20 to complete the 30-minute experiment plus bonuses from various tasks within the experiment. On average, employers earned $33.

The stage 2 experiment consists of three parts. The first part showed a screenshot of the exact job ad (see Figure A1 in appendix) and precisely described where it was posted, the programming test, how the test score is calculated, and a description of the employers' task. The experiment included 10 comprehension questions and one attention check to make sure the employers had a good understanding of the job and could form expectations about the potential applicant pool. The second part elicited employers' beliefs about the applicants' ability. We describe this part below. The third part consists of an experiment, which includes various tasks and questions to better understand the nature of employer beliefs.

In the second part of stage 2, employers saw profiles of 10 randomly selected applicants (5 females and 5 males). As the employers read each profile, they were asked to guess the applicant's score on the Python test (task 1) and the score guessed by a randomly

[14] We preregistered and requested a sample of 600, however, Qualtrics oversampled the number of employers. We include all employers for full transparency.

selected participant with a similar occupation to themselves (task 2). We use employers' guesses on task 1 to measure their beliefs about applicants' Python programming skills.

Each employer made 10 incentivized guesses for task 1 and 10 incentivized guesses for task 2. One of the 20 guesses was randomly selected for potential payment. For this selected profile, employers were paid based on the binarized scoring rule, where the closer they got to the correct scores, the more likely they received an additional $10 (Hossain and Okui, 2013; Danz et al., 2022). As is common, we emphasized that the more accurate a guess, the more likely they are to earn a bonus.[15]

In our baseline treatment, employers (n = 240) saw applicants' profiles which contained the first name and the initial of the last name. This information allowed the employer to infer the gender of the applicant without it being explicitly signaled, reducing concerns around social desirability bias. Like the information provided in a resume, the profiles also included information on the highest level of education completed, whether applicants are currently studying or working, if working, their occupation, years of programming experience, and where they learnt to program. Table 3 shows an example of a profile from the baseline treatment.

Table 3: Example of a Profile from Baseline Treatment

Name	Benjamin C.
Highest Education level	Graduated 4-year college
Currently Studying	No
Currently Working	Part Time in Software Development
Years of Programming Experience	8
Learned Programming from	University and self-taught

Our design minimizes the influence of dislike for a particular group which may affect hiring decisions in real life for three reasons. First, employers were informed that the hiring had already taken place and therefore knew that their guesses would not affect the applicants. Second, employers had a financial incentive to accurately guess applicants' performance. Third, there would be no potential relationship between the applicants and the employers. Our design therefore minimizes the main channels for taste-based discrimination and allows us to study beliefs that could lead to (different types of) statistical discrimination.

After reviewing 10 profiles, employers saw one of two bonus profiles. The characteristics were identical for both bonus profiles except for the name, where one showed

[15] Employers could read the technical details of the binarized scoring rule if they wish by opening a pop-up page.

a female name and the other a male name. Their beliefs about the performance for these profiles allow us to cleanly compare beliefs about male and female applicants holding profile characteristics constant. We chose the bonus profiles from our pool of applicants to be representative of the modal characteristics.[16] The guesses for the bonus profile were also incentivized. Employers could earn an additional $2, with payment based on the binarized scoring rule.

In the third part of the experiment, employers are asked to answer a few questions about themselves and complete a few tasks that allow us to study the sources behind any gender difference in beliefs. We describe these in greater details in Section 4.5, where we discuss the sources of gendered beliefs.

Columns (1) to (5) Table 4 shows summary statistics on the employer sample in the baseline treatment. The average age was 35 with women making up 42% of employers. 88% were responsible for hiring programmers. On average, employers correctly answered 8 out of 10 comprehension checks at their first attempt and 99% passed the attention checks. [17]This suggests understanding of the tasks was high.

2.3 Field Experiment on Changing Perceived Programming Skill

We investigate the assessment of applicants' skills for a job in a male-dominated labor market where female applicants might suffer from statistical discrimination. However, statistical discrimination is only possible if employers are uncertain about applicants' skills. Because of this uncertainty employers may use gender as a signal for skills. One way to mitigate statistical discrimination is therefore to provide further information about the applicant that may be relevant. This information should reduce uncertainty and reliance on gender as a signal.

We therefore conduct a controlled experiment in stage 2 to provide causal evidence on the role of information for job skill assessments. Employers were randomized into three treatments: *baseline*, *aptitude* and *personality*. In *baseline*, employers only received basic information typically shown on applicants' resumes (Table 3). In *aptitude*, employers

[16] The following are the modal characteristics: graduated from 4-year college; not studying or employed; had 5 years of coding experience and learnt coding through university and self-taught.

[17] For example, comprehension questions included having employers correctly identify the definition of each programming test component. As attention check, we asked employers to select the option "Strongly Disagree" to a question item that appeared in the third part of the experiment. The item reads "Thank you for answering the prior questions. You will now be asked a number of demographic questions. To proceed please select strongly disagree"

received additional information on the applicants' scores from the aptitude test (cognitive abilities, abstract reasoning, critical thinking, reasoning ability and attention to detail). In *personality*, employers received additional information on the applicants' scores on the Big Five personality traits (conscientiousness, extraversion, agreeableness, openness, and emotional stability). Examples of these profiles can be found in the Appendix (Figures A2 and A3). Columns (6) to (8) of Table 4, report a balance test comparing employer sample characteristics in the control, aptitude and personality variants showing no significant differences in observable characteristics other than in the number of comprehension checks passed at first attempt (7.72 in baseline, 7.93 in aptitude, 8.23 in personality, p=0.041).[18]

Table 4: Summary Statistics Employer Sample

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Sample	Baseline (n = 240)				Aptitude (n = 218)	Personality (n = 162)	
Variables	mean	sd	min	max	Mean	Mean	p-value F-test
age in years	35.12	9.93	18	69	34.88	35.83	0.6564
female	0.42	0.49	0	1	0.36	0.41	0.4026
completed 4+ years college	0.71	0.46	0	1	0.70	0.68	0.7983
full time employed	0.79	0.41	0	1	0.81	0.74	0.2831
programmer	0.51	0.50	0	1	0.52	0.45	0.2889
years in current role	7.13	6.22	0	50	6.89	7.79	0.4258
hires programmer	0.88	0.33	0	1	0.88	0.86	0.7728
# of comprehension checks passed	7.72	2.09	2	10	7.93	8.23	0.0408
attention check passed	0.98	0.13	0	1	0.99	0.99	0.5873

Note: This table shows summary statistics of the employer sample, separately for employers in the control treatment (Columns 1-4), aptitude treatment (Column 5), and personality treatment (Column 6). The n in the headers of Columns 3, 6 and 7 refer to the number of employers in each treatment. Across all three treatments, we miss age for 2 employers, years in current role for 3 employers, and hiring for 47 employers. Female is a dummy variable which is 1 if the employer is female and 0 if the employer is male. This variable is missing for 7 employers who had missing or other genders. "Hires programmers" is a dummy variable is equal to 1 for programmers who indicated that they are regularly, sometimes or rarely involved in hiring programmers and 0 for programmers who indicated they are not involved in hiring programmers. "# of comprehension checks passed" refers to the number of questions participants passed at first attempt, these exclude information treatments-specific control questions. All employers must answer all comprehension questions correctly before proceeding. To test if the characteristics of the three treatments differ by treatment, we regressed each variable on an aptitude treatment dummy and a personality treatment dummy (leaving the baseline treatment as base group) and ran an F-test for joint significance of those two dummies. Column 8 shows the p-values of those F-tests. We discuss the reasons for the differences in employers in the different treatments and their implications in Appendix A1.2.

[18] The Personality treatment has a higher rate of attrition (i.e., lower rate of experiment completion) than the other treatments. However, we do not believe this to be a concern, as we show in Table 4 there is little difference in characteristics across treatments suggesting treatments are balanced. We discuss attrition in detail in Appendix A1.2.

We selected these two information treatments because aptitude and personality tests are common in job applications.[19] However, there is no clear evidence on their ability to predict skill levels. There are reasons to believe that these tests may be useful: a programming aptitude test is designed to be informative about programming skills and programmers are perceived to on average have certain personality traits (Ehrlinger et al. 2018). If these tests are predictive of programming skills, then providing employers with their results should limit the reliance on gender to infer applicants' skills.

3. Empirical Strategy and Interpretation of Results

Gender differences in actual coding skill

We start by testing raw gender differences in coding skills amongst our applicants by estimating the following model:

$$Python\ Score_i = \alpha_0 + \alpha_1 Female_i + \varepsilon_i, \quad (1)$$

where $Python\ Score_i$ is the Python score of applicant i, $Female_i$ is a dummy variable equal to one if the applicant is female and zero if the applicant is male, and ε_i is an error term. We estimate Equation (1) with ordinary least squares (OLS) regressions. Without any additional controls, OLS estimates of α_1 show us the average gender gap in the Python score of the applicants in our sample.

We then extend the model by including the following control variables: education level (high school graduate, some college, graduated 2-year college, graduated 4-year college, postgrad; base group: less than high school), a dummy variable indicating if the applicant is currently studying, a dummy variable indicating if the applicant is currently employed, applicants' stated coding experience in years, as well as four dummy variables indicating how the applicant learned to code (in university, in an online course, self-taught, other). OLS estimates of α_1 from this specification show us the gender gap in the Python score conditional on these variables.

We estimate Equation (1) in our main sample, as well as a sample where we exclude applicants who scored 0 on the Python test. For ease of interpretation, we standardize the Python test score for both samples by subtracting the mean and dividing by the standard

[19] Psychometric testing in a process known as "assessment center" is common in job applications in countries such as the United States and the United Kingdom. See e.g. Ballantyne and Povah (2017), Crawley et al. (1990), Spychalski (1997).

deviation of Python score *in our main sample*. This way of standardizing our outcome variable makes the size of the coefficients in both samples comparable.

Gender differences in perceived coding skill

We test if there is a perceived gender gap in programming skills, by estimating the following model:

$$Python\ Score\ Guess_{ep} = \beta_0 + \beta_1 Female\ Profile_p + \varepsilon_{ep}, \quad (2)$$

where $Python\ Score\ Guess_{ep}$ is the test score guess of employer *e* about profile *p,* $Female\ Profile_p$ is a dummy variable which is equal to one if profile *p* was from a female applicant and 0 if it was from a male applicant. In this specification without control variables, OLS estimates of β_1 show the average difference in employers' test score guesses of female profiles compared to male profiles.

Any gender difference in guesses could be driven by differences in other characteristics shown on the profiles which are correlated with gender. In additional specifications, we therefore control for applicant characteristics that were shown on the profile. Those are the same control variables we include when estimating the gender gap in applicants' actual coding skills. In these specifications, OLS estimates of β_1 show differences in employers' test score guesses of female profiles compared to male profiles, conditional on the other information available on the profiles.

We cluster standard errors at the employer level because we observe for each employer 10 guesses of regular profiles. For all specifications, we standardize the Python test score guesses by subtracting the mean guess and dividing by the standard deviation of *all regular profile guesses in the baseline treatment*. This approach allows us to compare the effect sizes across all specifications.

We estimate Equations (2) with different samples. We begin by estimating models using all profiles shown to employers in the baseline treatment. However, since employers observe multiple profiles there is potential for order or learning effects. To estimate perceived skill gaps without such learning effects, we also estimate a model restricting our sample to the first profile employers judged. Finally, to completely control for profile characteristics we estimate a model using only the bonus profile—where profiles were identical except for the name of the applicant. These three variants allow us to better understand the dynamics of beliefs about gender differences in programming skills.

4. Results

4.1: No significant gender differences in actual programming skill

Table 5 shows no significant differences in Python coding test scores between female and male applicants.[20] Female applicants in our sample scored similarly to their male peers (p-value=0.675). This statistically insignificant difference holds when we additionally control for applicants' level of education, student status, employment status, and coding experience (p-value=0.395).

Table 5: Gender differences in applicants' Python test scores

	(1)	(2)	(3)	(4)
	Dep. Variable: Std. Coding Skill			
female	-0.061	-0.137	0.064	0.009
	(0.146)	(0.159)	(0.111)	(0.122)
Constant	0.014	0.124	0.301***	0.753***
	(0.062)	(0.509)	(0.046)	(0.179)
Observations	318	318	272	272
R-squared	0.001	0.033	0.001	0.050
Controls	No	Yes	No	Yes
Include Zeros	Yes	Yes	No	No

Note: Additional control variables in Columns (2) and (4) are four indicators of applicants' level of education (high school graduate, some college, graduated 2-year college, graduated 4-year college, postgrad; base group: less than high school), one dummy variable indicating if the applicant is currently studying, one dummy variable indicating if the applicant is currently employed, applicants' stated coding experience in years, coding experience-squared,. Heteroskedasticity robust standard errors in parentheses, as well as four dummy variables indicating how the applicant learned to code (in university, in an online course, self-taught, in another way). We standardize the Python test score for both samples by subtracting the mean and dividing by the standard deviation of Python score in our main sample. *** p<0.01, ** p<0.05, * p<0.1

The lack of a statistically signification gender difference may hide important differences in the distribution of the applicants' test scores. Figure A4 in the appendix shows that this is not the case. The distributions of test scores are very similar for female and male applicants. A Kolmogorov–Smirnov test confirms that these distributions are statistically indistinguishable (p = 0.810). These distributions also reveal that substantial shares of female

[20] Our finding of an insignificant gender difference is consistent with indirect evidence from Terrell et al. (2017) who compare acceptance rates of contributions to GitHub projects by female and male programmers.

and male applicants score zero points on the Python test. Scoring zero points might be an indication of applicants not trying hard or not being qualified in the first place.[21] Columns (3) and (4) of Table 5 shows that without these zero-scores, the sign flips with female applicants now slightly outperforming male applicants. However, once again, there are no statistically significant differences between female and male performance in both models without controls (p =0.564) or with control variables (p =0.944).[22] We re-estimate the specifications shown in Table 5 adding data from applicants from the incentive treatment. Table A2 in the appendix shows that including those additional observations leads to the same conclusions: there are no significant gender differences in any specification.

FINDING 1: *There are no significant gender differences in actual programming task performance*.

4.2 Employers Perceive Female Applicants to Have Lower Python Skills

Table 6 shows that employers believed that female applicants performed significantly worse on the programming test than their male counterparts. Without any additional control variables, employers expected female applicants to score 0.12 SD lower. This difference is statistically significant at the 1 percent level and equivalent to 2.66 points on the Python test. The perceived gender skill gap remains virtually unchanged when we control for applicants' information shown on the profile (Column 2).

This gender difference in perceived skills is even more pronounced when we look at the first applicant profile and the bonus profile. Looking only at the first profile, this gender gap increases to 0.19 SD without any controls (Column 3), to 0.39 SD when we control for information shown on the profile (Column 4).[23] When only looking at the bonus profile at the end of the assessments, employers guessed a 0.23 SD lower score for the profile with the female name.[24] Taken together, these results suggest one consistent and highly robust finding: employers believe women programmers are less skilled at coding.

[21] We did not anticipate gender differences in zero points scores and as such did not specify the need for an analysis excluding zeros in the pre-analysis plan.
[22] Another concern is that the lack of gender differences in scores is driven by the overall scoring of the Python test. Table A3 in the appendix shows that this is not the case. We see no significant (conditional or unconditional) gender differences for any of the sub scores.
[23] This large increase in effect size is partly driven by the education controls. For the first profile, female applicants were particularly well educated and high levels of education predict high coding test scores. Without controlling for applicants' level of education, the female coefficient reduces to 0.23 SD.
[24] We do not include controls as profile characteristics are identical except for the name.

Table 6: Beliefs about Gender Differences in Coding Skill

	(1) Std. Guess	(2) Std. Guess	(3) Std. Guess	(4) Std. guess	(5) Std. guess
female profile	-0.118***	-0.124***	-0.188*	-0.390***	-0.234**
	(0.031)	(0.042)	(0.102)	(0.125)	(0.119)
Constant	0.059	-0.497*	0.034	-0.479	0.351**
	(0.049)	(0.267)	(0.073)	(1.079)	(0.173)
Observations	2,400	2,400	240	240	240
R-squared	0.003	0.085	0.014	0.105	0.017
Controls	No	Yes	No	Yes	No
Sample	All profiles	All profiles	1st profile	1st profile	Bonus profile

Note: Controls are: five dummies for educational achievement, currently studying, currently working, three dummies for how applicant learnt to code (at university, online course, self-taught, other), years of programming experience, and programming experience squared. All dependent variables are standardized by subtracting the mean and dividing by the standard deviation of all guesses made in the baseline treatment. Columns 1-2 report standard errors clustered at the employer level. Standard errors reported in all other columns are heteroskedasticity robust. *** p<0.01, ** p<0.05, * p<0.1

FINDING 2: *Female applicants are perceived to have a significantly lower performance in the programming task than male applicants.*

Firms often use multiple assessors to evaluate each job application. In such cases, assessors' beliefs about one's colleagues' beliefs might matter as well (see, for example Bursztyn et al., 2020). We therefore estimate the perceived gender gap in second-order beliefs, that is, guesses about the guess of a randomly selected fellow employer who works in the same profession. Table A4 in the appendix shows that employers also believed that their peers expected female applicants to perform worse on the Python test. In our main specification using second order guesses of all 10 regular applicant profiles, we see that employers expected their peers to guess 0.10 SD worse on the Python test and this gap increases to 0.13 SD once we control for information shown on applicants' profiles. Moreover, employers' guesses of applicants' test scores and their guesses of other employer's guesses are highly correlated (r=0.84). These results suggest that employers believed that their colleagues held a similar view.

Table 7: Heterogeneity by Employers' Occupation and Gender

(1) Subsample	(2) Female profile coef.	(3) Std. err.	(4) p-value
Panel A: Programmer vs HR, female employer vs male employer			
Programmer (n = 123)			
raw	-0.114***	0.043	0.009
conditional on controls	-0.100*	0.051	0.054
HR (n = 117)			
raw	-0.122***	0.046	0.009
conditional on controls	-0.148**	0.069	0.034
Female employer (99)			
raw	-0.187***	0.050	0.000
conditional on controls	-0.214***	0.068	0.002
Male employer (139)			
raw	-0.067*	0.040	0.098
conditional on controls	-0.065	0.054	0.229
Panel B: four subgroups			
Female Programmer (n= 32)			
raw	-0.212**	0.079	0.012
conditional on controls	-0.124	0.093	0.189
Male Programmer (n = 90)			
raw	-0.079	0.051	0.123
conditional on controls	-0.086	0.061	0.163
Female HR (n = 67)			
raw	-0.176***	0.064	0.008
conditional on controls	-0.249***	0.086	0.005
Male HR (n = 49)			
raw	-0.045	0.066	0.504
conditional on controls	-0.007	0.115	0.951

Note: The female profile coefficients shown in column (2) are from regressions of standardized Python score guess on a female profile dummy. Rows denoted with 'conditional on controls' report the same coefficient from regressions that additionally control for the following information shown on the applicants' profiles: five dummies for educational achievement, currently studying, currently working, three dummies for how applicant learnt to code (at university, online course, self-taught, other), years of programming experience, and programming experience squared. Panel A shows results separately for programmers, HR managers, women and men. Panel B shows results separately for female programmers, male programmers, female HR managers, male HR managers. All regressions are estimated with data from our main experiment only. Standard errors are clustered at the employer level. p<0.01, ** p<0.05, * p<0.1

The characteristics of the employers could be important in understanding beliefs. Table 7 shows how beliefs about gender differences in coding performance differ by key employer characteristics. Panel A shows results from regressions with and without profile controls estimated in four samples with employers who are 1) female, 2) male, 3) programmers and 4) HR professionals. We find that the perceived gender gap in programming skills is larger for female employers. They expected female applicants to perform 0.19 SD worse than male applicants (p <0.001). In contrast, male employers only

expected female applicants to perform 0.06 SD worse (p =0.098). This result holds when control variables are included. We see little heterogeneity by employer occupation, with both programmers and HR professionals expected female applicants to perform worse than male applicants.

We further estimate separate regressions for each combination of occupation and gender, that is, 1) female programmers, 2) male programmers, 3) female HR professionals, and 4) male HR professionals. Panel B shows that differences in beliefs about coding performance were driven by female programmers and female HR professionals. Female programmers guessed that female applicants perform on average 0.21 SD worse than male applicants (p=0.012). Similarly, female HR professionals guessed that female applicants perform 0.18 SD worse (p=0.008). In contrast, male programmers and male HR professionals show smaller and statistically insignificant gender differences in guesses about applicants' coding performance. This result holds when control variables are included in the analysis.

FINDING 3: *The perceived gender performance gap is mainly driven by female employers.*

4.3 Showing Additional Information on Profiles Reduces the Perceived Skill Gap

One possible way to close the perceived skill gap is to provide further information about the job applicant, which reduces uncertainty and reliance on beliefs that are inaccurate. Note that the applicants' personality traits significantly predict their Python score but their aptitude score does not (see Table A5).[25] There were no gender differences in personality or aptitude among our applicants (see Table A6).

To estimate the effect of providing additional information, we re-estimate Equation (2) with employers from all three treatments, include dummies for each information treatment (aptitude or personality), and interaction terms of the female profile dummy and these treatment dummies. The coefficient on these interaction terms shows the difference in the perceived skill gap between each information treatment and the baseline treatment (conditional on control variables).

[25] Our results show that the Big Five personality test is predictive of an applicant's skills in our context. However, it is not clear whether this is because certain personality traits are associated with being a good programmer, or this is a test of sophistication: applicants answering according to what traits they perceive that employers would want in them rather than answering truthfully. Regardless of the interpretation, these results suggest that personality traits may be useful information for the employers in our context.

Table 8: Differences in Beliefs after Adding Aptitude or Personality Information

	(1)	(2)
	Dep. Var: Std. Coding Score Guess	
female profile	-0.118***	-0.141***
	(0.031)	(0.034)
female profile X aptitude treatment	0.166***	0.106**
	(0.041)	(0.042)
female profile X personality treatment	0.106**	0.115**
	(0.049)	(0.047)
Aptitude treatment	-0.516***	68.410
	(0.075)	(64.284)
Personality treatment	-0.260***	35.935
	(0.078)	(94.150)
Observations	6,238	6,238
R-squared	0.034	0.131
Sample	all 10 profiles	all 10 profiles
Demographic controls	No	Yes
Aptitude controls	No	Yes
Personality controls	No	Yes
Estimated effect of female profile for		
Control	-.118	-.141
p-value baseline	[.0002]	[0]
Aptitude	.048	-.035
p-value aptitude	[.0687]	[.2743]
Personality	-.012	-.026
p-value aptitude	[.7454]	[.4896]

Note: The dependent variable in all columns is the standardized test score guess. Demographic controls are: five dummies for educational achievement, currently studying, currently working, three dummies for how applicant learnt to code (at university, online course, self-taught, other), years of programming experience, programming experience squared, experience cubed. Aptitude controls include each of the aptitude test scores (critical thinking, attention to detail, abstract reasoning, reasoning ability, and cognitive ability) and their square terms. Personality controls include the following big 5 personality scores (conscientiousness, openness, agreeableness, extraversion and emotional stability) and their square terms. Standard errors clustered at the employer level in parentheses. *** p<0.01, ** p<0.05, * p<0.1

Table 8 shows that both information treatments significantly reduce the perceived gender skill gap. In stark contrast to the baseline treatment, we observe no significant gap in either of the information treatments. In the *aptitude* treatment, the sign of the perceived skill gap even flips. Employers who were additionally shown applicants' performance on the aptitude guessed that female applicants performed 0.05 SD *better* than their male counterparts (p=0.069). The coefficient on the interaction term shows that this 0.17 SD difference in the perceived skill gap compared to the baseline experiment is statistically significant (p<0.001). Employers who were additionally shown applicants' Big 5 personality

scores guessed that female applicants performed on average 0.01 SD worse. This 0.11 SD difference in perceived skill gap compared to the baseline experiment (without controls) is statistically significant (p=0.032). We observe similar effects of the information treatments in specification in which we additionally include for basic profile controls and the information shown on the aptitude and personality tests.

Note that both aptitudes and personality traits predict employers' perceived scores, meaning that employers do use this information in their guesses (see Table A7). The provision of such information, especially aptitude, also makes the employers' guesses significantly more accurate (see Table A8).

In Table 9, we break the result down further by the gender of the employer. The first two columns report the results for female employers and the last two for male employers. We find a consistent pattern—providing additional information changes the guesses of female employers—the group who guessed lower scores for female applicants.

FINDING 4: *Providing additional information on applicants' aptitude or personality can close the perceived performance gap.*

4.4 Potential Sources of Inaccurate Gendered Beliefs about Skills

We posit three possible explanations for inaccurate gender beliefs in our pre-analysis plan. We use a series of additional tasks completed by the employers and data from the experiment to test the necessary conditions of each of the three potential sources of inaccurate beliefs.

Potential source 1: Attention Discrimination

Bartos et al. (2016) provided evidence that Czech employers were less likely to open resumes from applicants with Asian-sounding names compared to majority-sounding names and were less likely to call them back. They attributed this difference to attention discrimination. In our context, it is plausible that employers gave more attention to male applicants in their tasks. We test for attention discrimination by measuring the time employers spent on each job profile and relate it to gender differences in assessments.

Table 9: Effect of Information Treatments, separately by employer gender

	(1)	(2)	(3)	(4)
	Dep. Var: Std. Coding Score Guess			
Sample	Female employers		Male employers	
female profile	-0.187***	-0.212***	-0.067*	-0.086**
	(0.050)	(0.055)	(0.040)	(0.043)
female profile X aptitude treatment	0.238***	0.191***	0.121**	0.052
	(0.071)	(0.071)	(0.051)	(0.052)
female profile X personality treatment	0.256***	0.272***	-0.001	0.012
	(0.079)	(0.077)	(0.063)	(0.061)
Aptitude treatment	-0.756***	124.129	-0.382***	26.901
	(0.125)	(98.347)	(0.092)	(81.920)
Personality treatment	-0.484***	174.835	-0.106	-74.127
	(0.122)	(172.361)	(0.099)	(108.173)
Observations	2,422	2,422	3,746	3,746
R-squared	0.068	0.169	0.021	0.123
Sample	all 10 profiles	all 10 profiles	all 10 profiles	all 10 profiles
Demographic controls	No	Yes	No	Yes
Aptitude controls	No	Yes	No	Yes
Personality controls	No	Yes	No	Yes
Estimated effect of female profile for				
Control	-.187	-.212	-.067	-.086
p-value baseline	[.0002]	[.0002]	[.0962]	[.0454]
Aptitude	.051	-.02	.053	-.034
p-value aptitude	[.3116]	[.7249]	[.0848]	[.3889]
Personality	.068	.06	-.069	-.074
p-value personality	[.2649]	[.3156]	[.154]	[.1191]

Note: The dependent variable in all columns is the standardized test score guess. Columns 1 and 2 show regressions with only female employers and Columns 3 and 4 include regressions with only male employers. Demographic controls are: five dummies for educational achievement, currently studying, currently working, three dummies for how applicant learnt to code (at university, online course, self-taught, other), years of programming experience, programming experience squared, experience cubed. Aptitude controls include each of the aptitude test scores (critical thinking, attention to detail, abstract reasoning, reasoning ability, and cognitive ability) and their square terms. Personality controls include the following big 5 personality scores (conscientiousness, openness, agreeableness, extraversion and emotional stability) and their square terms. Standard errors clustered at the employer level in parentheses. *** p<0.01, ** p<0.05, * p<0.1

We do not find that employers' attention differs by the gender of the applicant. Employers spent on average 24.9 seconds on a female profile and 25.1 seconds on a male profile.[26] The difference of 0.2 seconds is not statistically significant (p=0.798). We also fail to see significant differences in time spent on female compared to male profiles if we limit our analysis to the first profile employers saw (61.9 seconds for female profiles and 60.0 seconds for male profiles, p=0.624), or if we do our analysis separately for female and male employers (25.0 vs 25.2 for female employers, p=0.745; 25.0 vs 25.0 for male employers, p=0.937). These results suggest that attention discrimination does not explain the gender differences in predicted test scores in our setting.

Potential Source 2: Representative Heuristic

The representative heuristic (Kahneman and Tversky, 1973; Tversky and Kahneman, 1983) provides a rationale for inaccurate beliefs and has been applied in many environments, including stock prices (Barberis et al., 1998), insurance purchase (Dumm et al., 2020), and medical decisions (Graber et al., 2001). Bordalo et al. (2016) present a model of stereotypes that formalizes the predictions of the heuristic. They propose that people's judgment about how different two groups are along a certain characteristic (e.g. age) is driven by the part of the distribution where these two groups differ the most. For example, when asked to guess how old people are in Florida compared to the rest of the US, people's judgments would be influenced by the relatively larger share of old people in Florida (at other parts of the age distributions, Florida and the US are more similar). The difference in this "representative type" then leads people to exaggerate the true age difference. In other words, because the "stereotypical" Floridian is old, people overestimate the number of old people and the average age in Florida. Bordalo et al. (2019) apply the model to study beliefs about gender. They found evidence consistent with the model among laboratory subjects, where stereotypes caused subjects to exaggerate their beliefs about gender differences in a range of laboratory tasks.

In our context, this model of representativeness heuristics predicts that employers' beliefs about gender differences in applicants' skills would be exaggerated if there are gender differences between the skills distributions of the two genders. However, there are no statistically significant gender differences along any part of the skills distributions in our

[26] For this analysis, we remove outlier guesses for which employers spent unusual long periods of time (18 profiles on which employers spent more than 5 minutes). Findings are not subject to outlier guesses (48.2 seconds for female profiles vs. 27.5 seconds for male profiles, p=0.347).

sample of tech professionals who applied for programming jobs (see Section 4.1). As there are no obvious representative types, the model cannot explain our employers' beliefs in our context.[27]

Potential source 3: Selection Neglect

Selection neglect describes a tendency to draw false inferences from one sample to a non-random sub-sample, and it has been suggested as a key driver for market entry failures (Camerer and Lovallo, 1999), and inferior financial investments (Koehler and Mercer, 2009; Jehiel, 2018) as well as educational investments (Streufert, 2000).[28] We posit that selection neglect might also influence beliefs in the hiring and assessment context, especially in a male-dominant industry. In our setting, it is plausible that employers have accurate beliefs about gender differences in programming skills in the population, but ignore that job applicants represent a selected sample of the population. One could expect that men are on average better at programming than women in the population (or men are more likely to be able to program) simply because there are more men than women trained as programmers in the population. However, this expectation is not sensible when zooming in on the sample of men and women who choose to become professional programmers. Conditional on being a professional programmer, gender differences in programming skills should be smaller or non-existent compared to the general population. Failing to account for this indicates selection neglect.

To investigate selection neglect, we asked employers for their beliefs about gender differences in programming skills in the population and among applicants separately (in random order to avoid ordering effects). In particular, we asked "Among all people living in the United States (regardless of their profession), do you think women or men are, on average, better at programming? Please answer on a scale that ranges from "women are much better" "men are much better". While the second question states, "Among people who

[27] It is also possible that our employers had the general population instead of professional programmers in mind when forming stereotyped beliefs. Following Bordalo et al. (2016), employers should overestimate the proportion of professional programmers in the general population being male because men are representative of professional programmers. Assuming that professional programmers are among the most skilled programmers in the general population, this may translate to employers overestimating the likelihood of men being good programmers. However, in an incentivized task, we find that our employers do not overestimate the share of male programmers in the population. They believe that 73.5% of professional programmers are male whereas 78.9% are male according US Census data from 2018. This suggests that stereotyped beliefs based on the general population cannot readily explain the perceived gender differences in programming skills.

[28] There is also lab experimental literature on selection neglect providing causal evidence for its role (Lopez-Perez et al, 2022)

applied for this job, do you think women or men are on average better at programming? Please answer on a scale that ranges from "women are much better" to "men are much better". We would see strong evidence for selection neglect if answers to those two questions are statistically the same.

Applicants could answer both questions on a 101-point scale. For ease of interpretation, we rescale both answers to range from -50 points to 50 points, with 0 points meaning women and men are equally good at programming and positive values indicating beliefs that men are better at programming. Our results show that employers believed that men were 13.5 points better at coding than women in the population ($p<0.001$). They also believed that male applicants 11.6 points were better at coding than female applicants in our sample ($p<0.001$). The difference between population and applicant sample of 1.9 points is statistically significant ($p = 0.003$), suggesting that employers took selection into account to some extent. However, since the belief about gender differences among the applicant sample remains significant, employers were likely not adjusting for selection sufficiently.

We see an interesting pattern when we look at the gender of the employer. Both male and female employers believed that women are better than men at programming in the population. However, we see an interesting gender difference when comparing the answers about the population and the applicant sample. Male employers believed that the male-female skill gap was 2.6 points lower in our applicant sample compared to the population sample, suggesting that they took some selection into account ($p< 0.001$). In contrast, female employers believed that the male-female skill gap was only 0.4 points lower for our applicant sample compared to the population sample, and this difference is not significantly different from zero ($p= 0.716$). Female employers did not significantly distinguish between the population and our applicant sample. This apparent selection neglect might explain why female employers were more likely to believe that female applicants were worse at programming (Finding 3).

FINDING 5: *We find evidence consistent with selection neglect that could explain why female employers perceive female applicants as less skilled.*

5 Discussion

With perfect information about the skills of applicants, employers would not need to rely on gender as a signal. However, employers do not have perfect information. This is why it is important to improve our understanding of whether employers do rely on gender as a signal, how accurate their beliefs about gender are, and show the sources of such beliefs. These questions motivate our study.

We rigorously study the beliefs employers hold about job applicants for a tech job in an industry where women are underrepresented. We determine how accurate employers' beliefs are by comprehensively assessing applicants' skills and asking employers to guess applicants' skills. While we do not find significant gender differences in actual skills, we see gender differences in perceived skills: tech and HR professionals who were involved in hiring programmers believed that female programmers were worse than their male counterparts and they believed that other employers shared their beliefs. These inaccurate beliefs are stronger for female employers, which squares well with selection neglect. Female employers appeared to not consider that women applying for a programming position are different from women in the general population.

Beliefs of this kind can perpetuate labor market discrimination and harm female programmers. One potential solution to this problem is to provide additional information about job applicants to reduce employers' reliance on gender as a signal of applicant skills. Our field experiment shows that this solution can work. Employers who were shown profiles that included information on applicants' aptitude or personality, besides standard resume information, did not show evidence of inaccurate beliefs. We also find that female employers, whose beliefs were more inaccurate than the male counterparts, corrected their beliefs to a larger extent than male employers in the presence of additional information.

These results suggest that employers face an information problem and use applicants' gender to address it. However, their inference is inaccurate because they hold wrong beliefs. Therefore, in male-dominant sectors where employers may have pessimistic beliefs about female applicants, employers should consider additional information to reduce the scope for statistical discrimination. Such information could come from psychometric and additional aptitude testing, which is frequently used to assess applicants. We provide evidence that these tests are not only indicative of performance but also help to correct employers' biased beliefs about women and remove the possibility of statistical discrimination against women in the hiring of programmers.

Our paper raises important questions for future research. First, while employers' belief about applicants' abilities would likely have an important influence on shortlisting and hiring decisions, it would be useful for future research to study how inaccurate beliefs translate to broader employment outcomes. Second, which information should employers collect to reduce inaccurate statistical discrimination? In most real-world contexts, employers can only obtain proxies for skills, such as personality and aptitude. In our case, we found that while only personality is predictive of actual skills, both sets of information mitigated biases against female. Perhaps what really matters is whether people believe a piece of information is predictive of actual skills, regardless of its actual predictiveness. Third, is it important that there are no gender differences in the additional information provided, as is the case with aptitude and personality scores in our experiment? Answers to these questions could provide employers with more generalized recommendation and mitigate (inaccurate) statistical discrimination without harming efficiency.

Our results on employers' beliefs about the gender skill gap have broader policy implications. Ip et al. (2020) show that support for affirmative action policies such as gender quotas crucially depends on beliefs about whether there is a skill difference between male and female workers. Affirmative action plays a big role in the male-dominant tech industry. Our employers' beliefs about gender differences in programming skills suggest that support for affirmative action for programmers may be limited. Thus, correcting inaccurate beliefs on gender differences is not only important in hiring decisions but also important for the success of broader policies that help correct gender imbalances in male-dominated industries.

References

Aigner, D. J., & Cain, G. G. (1977). Statistical theories of discrimination in labor markets. *ILR Review*, *30*(2), 175-187.

Arrow, K. J. (1973). The Theory of Discrimination, Discrimination in Labor Markets, Ashenfelter, O. and A. Rees eds., 3-33.

Ballantyne, I., & Povah, N. (2017). *Assessment and development centres*. Routledge.

Barberis, N., Shleifer, A., & Vishny, R. (1998). A model of investor sentiment. *Journal of Financial Economics*, *49*(3), 307-343.

Bartoš, V., Bauer, M., Chytilová, J., & Matějka, F. (2016). Attention discrimination: Theory and field experiments with monitoring information acquisition. *American Economic Review*, *106*(6), 1437-75.

Becker, Gary S. (1957). The Economics of Discrimination. Chicago: The University of Chicago Press.

Bertrand, M., & Duflo, E. (2017). Field experiments on discrimination. *Handbook of economic field experiments*, *1*, 309-393.

Besley, T., Folke, O., Persson, T., & Rickne, J. (2017). Gender quotas and the crisis of the mediocre man: Theory and evidence from Sweden. *American Economic Review*, *107*(8), 2204-42.

Bohren, J. A., Imas, A., & Rosenberg, M. (2019). The dynamics of discrimination: Theory and evidence. *American Economic Review*, *109*(10), 3395-3436.

Bohren, J. A., Haggag, K., Imas, A., & Pope, D. G. (2020). *Inaccurate statistical discrimination: An identification problem* (No. w25935). National Bureau of Economic Research.

Bohren, J. A., Hull, P., & Imas, A. (2022). *Systemic discrimination: Theory and measurement* (No. w29820). National Bureau of Economic Research.

Bordalo, P., Coffman, K., Gennaioli, N., & Shleifer, A. (2016). Stereotypes. *The Quarterly Journal of Economics*, *131*(4), 1753-1794.

Bordalo, P., Coffman, K., Gennaioli, N., & Shleifer, A. (2019). Beliefs about gender. *American Economic Review*, *109*(3), 739-73.

Camerer, C., & Lovallo, D. (1999). Overconfidence and excess entry: An experimental approach. *American Economic Review*, *89*(1), 306-318.

Chan, A. (2022). Discrimination and Quality Signals: A Field Experiment with Healthcare Shoppers. *Mimeo.*

Coffman, K. B., Exley, C. L., & Niederle, M. (2021). The role of beliefs in driving gender discrimination. *Management Science*, *67*(6), 3551-3569

Correll, S. J., Benard, S., & Paik, I. (2007). Getting a job: Is there a motherhood penalty?. *American Journal of Sociology*, *112*(5), 1297-1338.

Crawley, B., Pinder, R., & Herriot, P. (1990). Assessment centre dimensions, personality and aptitudes. *Journal of Occupational Psychology*, *63*(3), 211-216.

Danz, David, Lise Vesterlund, and Alistair J. Wilson (2022). "Belief elicitation and behavioral incentive compatibility." *American Economic Review*.

Dee, T. S. (2005). A teacher like me: Does race, ethnicity, or gender matter?. *American Economic Review*, *95*(2), 158-165.

Delfino, A. (2021). Breaking gender barriers: Experimental evidence on men in pink-collar jobs. *IZA Working paper*, DP No. 14083.

Deloitte (2021) Women in the Board Room. 7th Edition. Available online: https://www2.deloitte.com/content/dam/Deloitte/global/Documents/gx-women-in-the-boardroom-seventh-edition.pdf

Dumm, R. E., Eckles, D. L., Nyce, C., & Volkman-Wise, J. (2020). The representative heuristic and catastrophe-related risk behaviors. *Journal of Risk and Uncertainty*, *60*(2), 157-185.

Ehrlinger, J., Plant, E. A., Hartwig, M. K., Vossen, J. J., Columb, C. J., & Brewer, L. E. (2018). Do gender differences in perceived prototypical computer scientists and engineers contribute to gender gaps in computer science and engineering?. *Sex Roles*, *78*(1), 40-51.

Fairlie, R. W., Hoffmann, F., & Oreopoulos, P. (2014). A community college instructor like me: Race and ethnicity interactions in the classroom. *American Economic Review*, *104*(8), 2567-91.

Fershtman, C., & Gneezy, U. (2001). Discrimination in a segmented society: An experimental approach. *The Quarterly Journal of Economics*, *116*(1), 351-377.

Fiedler, K. (2000). Beware of samples! A cognitive-ecological sampling approach to judgment biases. *Psychological Review*, *107*(4), 659.

Forbes, C. E., & Schmader, T. (2010). Retraining attitudes and stereotypes to affect motivation and cognitive capacity under stereotype threat. *Journal of Personality and Social Psychology*, *99*(5), 740.

Gertsberg, M., Mollerstrom, J., & Pagel, M. (2022). *NBER working paper nr. 28465.*

Graber, M. A., Bergus, G., Dawson, J. D., Wood, G. B., Levy, B. T., & Levin, I. (2000). Effect of a patient's psychiatric history on physicians' estimation of probability of disease. *Journal of General Internal Medicine*, *15*(3), 204-206.

Hardy, Jay H., I.,II, Gibson, C., Sloan, M., & Carr, A. (2017). Are applicants more likely to quit longer assessments? examining the effect of assessment length on applicant attrition behavior. *Journal of Applied Psychology, 102*(7), 1148-1158. doi:https://doi.org/10.1037/apl0000213

Hartwell, Christopher & Orr, Tyler & Edwards, John. (2020). Reducing Online Application Redundancy: Effects on Applicant Attrition and Quality. International Journal of Selection and Assessment. 10.1111/ijsa.12282.

Holm, H. J. (2000). Gender-based focal points. *Games and Economic Behavior*, *32*(2), 292-314.

Hoogendoorn, S., Oosterbeek, H., & Van Praag, M. (2013). The impact of gender diversity on the performance of business teams: Evidence from a field experiment. *Management Science*, *59*(7), 1514-1528.

Hoogendoorn, S., & Van Praag, C. (2012). Ethnic diversity and team performance: a randomized field experiment. In *Academy of Management Proceedings* (Vol. 1, p. 13736). Briarcliff Manor, NY 10510: Academy of Management.

Hossain, Tanjim, and Ryo Okui. (2013). "The binarized scoring rule." *Review of Economic Studies* 80 (3), 984-1001.

Ip, E., Leibbrandt, A., & Vecci, J. (2020). How do gender quotas affect workplace relationships? Complementary evidence from a representative survey and labor market experiments. *Management Science*, *66*(2), 805-822.

Jehiel, P. (2018). Investment strategy and selection bias: An equilibrium perspective on overoptimism. *American Economic Review*, *108*(6), 1582-97.

Kahneman, D., & Tversky, A. (1973). On the psychology of prediction. *Psychological Review*, *80*(4), 237.

Kawakami, K., Dovidio, J. F., & Van Kamp, S. (2007). The impact of counterstereotypic training and related correction processes on the application of stereotypes. *Group Processes & Intergroup Relations*, *10*(2), 139-156.

Kline, P. M., Rose, E. K., & Walters, C. R. (2021). Systemic discrimination among large US employers (No. w29053). *National Bureau of Economic Research.*

Koehler, J. J., & Mercer, M. (2009). Selection neglect in mutual fund advertisements. *Management Science*, *55*(7), 1107-1121.

Kübler, D., Schmid, J., & Stüber, R. (2018). Gender discrimination in hiring across occupations: a nationally-representative vignette study. *Labour Economics*, *55*, 215-229.

Lane, T. (2016). Discrimination in the laboratory: A meta-analysis of economics experiments. *European Economic Review*, *90*, 375-402.

Lepage, Louis Pierre, Endogenous Learning, Persistent Employer Biases, and Discrimination (2021). *Available at SSRN 3640663.*

List, J. A. (2004). The nature and extent of discrimination in the marketplace: Evidence from the field. *The Quarterly Journal of Economics*, *119*(1), 49-89.

López-Pérez, R., Pintér, Á., & Sánchez-Mangas, R. (2022). Some conditions (not) affecting selection neglect: Evidence from the lab. *Journal of Economic Behavior & Organization*, *195*, 140-157.

Lowe, M. (2021). Types of contact: A field experiment on collaborative and adversarial caste integration. *American Economic Review*, *111*(6), 1807-44.

Mengel, F., & Campos Mercade, P. (2021). Irrational Statistical Discrimination. *Available at SSRN 3843579*.

National Science Foundation (NSF). (2019). *Women, Minorities, and Persons with Disabilities in Science and Engineering*. Available at https://ncses.nsf.gov/pubs/nsf19304/

Neumark, D., (2018). Experimental research on labor market discrimination. *Journal of Economic Literature*, 56(3), pp.799-866.

Neumark, D., Bank, R. J., & Van Nort, K. D. (1996). Sex discrimination in restaurant hiring: An audit study. *The Quarterly Journal of Economics*, *111*(3), 915-941.

Olivetti, C., Patacchini, E., & Zenou, Y. (2020). Mothers, peers, and gender-role identity. *Journal of the European Economic Association*, *18*(1), 266-301.

Paluck, E. L., Green, S. A., & Green, D. P. (2019). The contact hypothesis re-evaluated. *Behavioural Public Policy*, *3*(2), 129-158.

Petit, Pascale. 2007. "The Effects of Age and Family Constraints on Gender Hiring Discrimination: A Field Experiment in the French Financial Sector." *Labour Economics* 14 (3): 371–91.

Pew Research Center (2021, April 1). *STEM Jobs See Uneven Progress in Increasing Gender, Racial and Ethnic Diversity.*

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

Rao, Gautam. (2019). "Familiarity Does Not Breed Contempt: Generosity, Discrimination, and Diversity in Delhi Schools." *American Economic Review*, 109 (3): 774-809.

Sarsons, H. (2017). Recognition for group work: Gender differences in academia. *American Economic Review*, *107*(5), 141-45.

Scacco, A., & Warren, S. S. (2018). Can social contact reduce prejudice and discrimination? Evidence from a field experiment in Nigeria. *American Political Science Review*, *112*(3), 654-677.

Sheng, E. (2018, Oct 22). Silicon Valley's dirty secret: Using a shadow workforce of contract employees to drive profits. *CNBC.* https://www.cnbc.com/2018/10/22/silicon-valley-using-contract-employees-to-drive-profits.html

Slonim, R., & Guillen, P. (2010). Gender selection discrimination: Evidence from a trust game. *Journal of Economic Behavior & Organization*, *76*(2), 385-405.

Spychalski, A. C., Quiñones, M. A., Gaugler, B. B., & Pohley, K. (1997). A survey of assessment center practices in organizations in the United States. *Personnel Psychology*, *50*(1), 71-90.

Streufert, P. (2000). The effect of underclass social isolation on schooling choice. *Journal of Public Economic Theory*, *2*(4), 461-482.

Tversky, A., & Kahneman, D. (1983). Extensional versus intuitive reasoning: The conjunction fallacy in probability judgment. *Psychological Review*, *90*(4), 293.

United Nations Women (2021) Women in politics: New data shows growth but also setbacks. Available Online: https://www.unwomen.org/en/news/stories/2021/3/press-release-women-in-politics-new-data-shows-growth-but-also-setbacks

Upwork (2020) *Freelance Forward 2020.* Available Online: https://www.slideshare.net/upwork/freelance-forward-2020

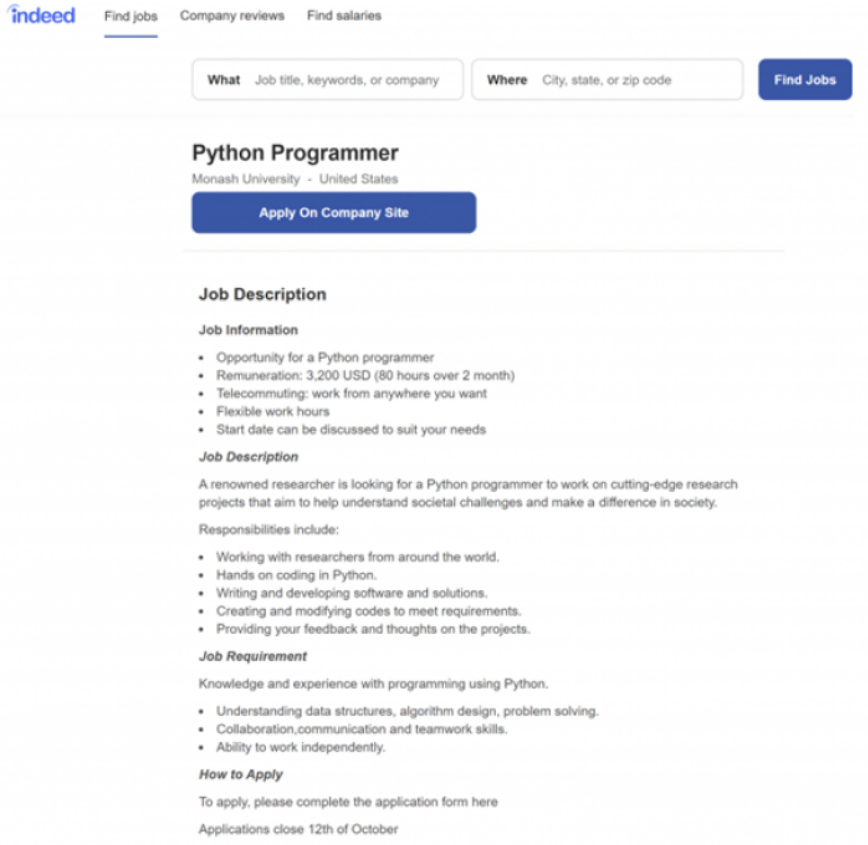
Upwork (2021a) *Freelance Forward 2021.* Available Online: https://www.upwork.com/research/freelance-forward-2021

Upwork (2021b) *Future Workforce Report 2021.* Available Online: https://www.upwork.com/research/future-workforce-report

Appendix

FIGURES

Figure A1: The Job Advertisement on a common job board



Note: This is the job advertisement posted on Indeed.com

Figure A2: Example of a profile from the aptitude treatment

Profile

Here is the profile of a real applicant:

Name	Lisa Z.
Highest Education level	Post Graduate
Currently Studying	No
Currently Working	Not employed
Years of Programming Experience	8
Learned Programming from	University
Aptitude Results:	
Cognitive ability	5
Abstract reasoning	4
Critical thinking	3
Reasoning ability	7
Attention to detail	5

Note: Aptitude results are scored 0/10 where a higher number indicates a higher aptitude.

Figure A3: Example of a profile from the personality treatment

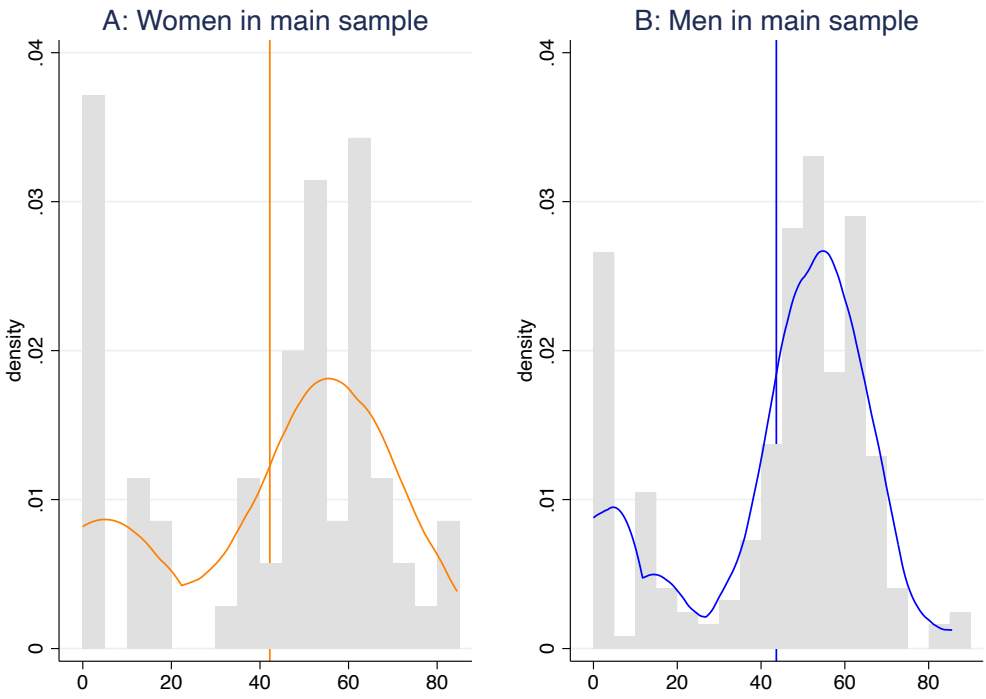
Profile

Here is the profile of a real applicant:

Name	Oliver L.
Highest Education level	Some College
Currently Studying	Yes
Currently Working	Not employed
Years of Programming Experience	5
Learned Programming from	University
Personality Results:	
Conscientiousness	8
Extraversion	6
Agreeableness	7
Openness	6
Emotional Stability	5

Note: Personality results are scored 1/9 where a higher number indicates an applicant is more conscientious, extraverted, agreeable, open or emotionally stable.

Figure A4: Distribution of applicants' Python scores by gender



Note: Figure A shows the distribution of women's Python scores, Figure B shows the distribution of men's Python scores. In both figures, grey bars show a histogram, the line going from left to right shows the density, and the vertical lines show the sample means.

TABLES

Table A1: Summary statistics all eligible applicants

	(1) N	(2) mean	(3) sd	(4) min	(5) max
Female	2,132	0.165	0.371	0	1
Currently studying	2,132	0.299	0.458	0	1
Currently employed	2,126	0.575	0.494	0	1
Education					
High school without graduation	2,132	0.0211	0.144	0	1
High school graduate	2,132	0.0492	0.216	0	1
Some college	2,132	0.203	0.402	0	1
2-year college	2,132	0.0675	0.251	0	1
4-year college	2,132	0.456	0.498	0	1
Postgrad	2,132	0.203	0.402	0	1
Years coding exper.	2,131	5.626	6.028	0	45
Learned coding					
in university	2,132	0.788	0.409	0	1
in online course	2,132	0.289	0.454	0	1
self-taught	2,132	0.595	0.491	0	1
other way	2,132	0.212	0.408	0	1

Table A2: Gender differences in applicants' Python test scores including applicants from the incentive treatment

	(1)	(2)	(3)	(4)
	Dep. Variable: Std. Coding Skill			
female	-0.135	-0.181	0.092	0.037
	(0.107)	(0.111)	(0.080)	(0.083)
Constant	0.048	-0.455	0.308***	0.377
	(0.044)	(0.398)	(0.034)	(0.312)
Observations	622	621	534	533
R-squared	0.003	0.040	0.003	0.030
Controls	No	Yes	No	Yes
Include Zeros	Yes	Yes	No	No

Note: This table replicates the regressions shown in Table 5 with additionally including data from applicants from the incentive treatment and applicants from all four treatments who we did not classify as having attempted the Python test. Additional control variables in Columns (2) and (4) are four indicators of applicants' level of education (high school graduate, some college, graduated 2-year college, graduated 4-year college, postgrad; base group: less than high school), one dummy variable indicating if the applicant is currently studying, one dummy variable indicating if the applicant is currently employed, applicants' stated coding experience in years, coding experience-squared. Heteroskedasticity robust standard errors in parenthesis, as well as four dummy variables indicating how the applicant learned coding (in university, in an online course, self-taught, in another way). We standardize the Python test score for both samples by subtracting the mean and dividing by the standard deviation of Python score in our main sample. *** $p<0.01$, ** $p<0.05$, * $p<0.1$

Table A3: Gender gaps in sub-scores

	(1) Std. Test cases	(2) Std. Efficiency	(3) Std. Complexity	(4) Std. Convention	(5) Std. Errors
Panel A: without controls					
female	-0.139	0.165	-0.062	-0.081	-0.070
	(0.142)	(0.138)	(0.137)	(0.139)	(0.139)
Constant	0.031	-0.036	0.014	0.018	0.015
	(0.062)	(0.063)	(0.063)	(0.063)	(0.063)
Observations	318	318	318	318	318
R-squared	0.003	0.005	0.001	0.001	0.001
Controls	No	No	No	No	No
Panel B: with controls					
female	-0.209	0.153	-0.175	-0.117	-0.101
	(0.155)	(0.141)	(0.145)	(0.158)	(0.155)
Constant	0.038	-0.027	0.478	-0.053	0.017
	(0.515)	(0.290)	(0.599)	(0.531)	(0.483)
Observations	318	318	318	318	318
R-squared	0.041	0.060	0.044	0.025	0.041
Controls	Yes	Yes	Yes	Yes	Yes

Note: Additional control variables in Panel B are four indicators of applicants' level of education (high school graduate, some college, graduated 2-year college, graduated 4-year college, postgrad; base group: less than high school), one dummy variable indicating if the applicant is currently studying, one dummy variable indicating if the applicant is currently employed, applicants' stated coding experience in years, coding experience-squared. Heteroskedasticity robust standard errors in parenthesis, as well as four dummy variables indicating how the applicant learned coding (in university, in an online course, self-taught, in another way). All dependent variables are standardized to have a mean of zero and standard deviation of one for the main sample. *** $p<0.01$, ** $p<0.05$, * $p<0.1$

Table A4: Second-order beliefs about gender differences in coding skill

	(1)	(2)	(3)	(4)	(5)
	Std. guess	Std. guess	Std. guess	Std. guess	Std. guess
female profile	-0.097***	-0.127***	-0.183*	-0.344***	-0.132
	(0.032)	(0.042)	(0.106)	(0.125)	(0.122)
Constant	0.048	-0.661**	0.079	-0.341	0.188
	(0.049)	(0.257)	(0.075)	(0.813)	(0.182)
Observations	2,400	2,400	240	240	240
R-squared	0.002	0.084	0.012	0.057	0.005
Controls	No	Yes	No	Yes	No
Sample	All profiles	All profiles	1st profile	1st profile	Bonus profile

Note: Controls are: five dummies for educational achievement, currently studying, currently working, three dummies for how applicant learnt coding (at university, online course, self-taught, other), years of programming experience, and programming experience squared. All dependent variables are standardized by subtracting the mean and dividing by the standard deviation of all guesses made in the control treatment. Columns 1-2 report standard errors clustered at the employer level. Standard errors reported in all other columns are heteroskedasticity robust. *** p<0.01, ** p<0.05, * p<0.1

Table A5: Predictors of actual Python score

(1)	(2)	(3)	(4)	(5)
		Dep Var: Actual Python Score		
Predictor	Coef.	se	p-val	N
Currently studying	-0.026	0.129	0.842	318
Currently employed	-0.019	0.112	0.866	318
Years coding exper.	0.004	0.009	0.698	318
Learned coding				
in university	0.200	0.166	0.228	318
in online course	-0.136	0.133	0.305	318
self-taught	-0.130	0.112	0.246	318
other	0.034	0.128	0.790	318
Educational attainment				
(Base: Some high school without graduation)				
High school graduate	-0.442	0.534	0.408	318
Some college	-0.208	0.484	0.668	318
2-year college	-0.190	0.567	0.738	318
4-year college	-0.177	0.467	0.704	318
Postgrad	0.010	0.475	0.983	318
Aptitude Test (0-10)				
Cognitive Abilities	0.064	0.045	0.161	156
Abstract Reasoning	0.048	0.027	0.071	156
Critical Thinking	0.014	0.033	0.679	156
Reasoning Ability	0.048	0.030	0.110	156
Attention to Detail	0.031	0.037	0.404	156
Aptitude test (combined)	0.062	0.046	0.183	156
Personality Test (1-9)				
Openness	0.185	0.050	0.000	162
Conscientiousness	0.133	0.046	0.005	162
Extraversion	0.144	0.047	0.003	162
Agreeableness	0.158	0.047	0.001	162
Emotional Stability	0.119	0.046	0.010	162

Note: This table shows coefficients, standard errors, and p-values, and number of observations from regressions of applicants' actual Python test scores (Columns 2-5). For educational attainment, these come from two single regressions with one dependent variable (either actual score or guess of actual score) and five educational attainment dummies. For the remaining predictors shown in Column (1) the coefficients, standard errors, p-values, and number of observations come from bivariate regressions with the predictor in Column (1) as the only independent variable. Each of the five different Aptitude test sub scores can range from 0 to 10, with 10 being the best performance. Aptitude Test (combined) is the unweighted average of all five sub scores. Each of the five different personality scores can range from 1 to 9, with 9 indicating applicants' personality is most open/conscientious/extraverted/agreeable or emotionally stable. P-values in Column (4) are based on heteroskedasticity robust standard errors.

Table A6: Gender Differences in Aptitude Scores and Personality Traits

	(1) Cognitive Abilities	(2) Abstract Reasoning	(3) Critical Thinking	(4) Attention to Detail	(5) Attention to Detail
Panel A: Aptitude scores					
female	0.192	1.851	-3.540	-1.244	-1.244
	(3.233)	(4.864)	(3.800)	(3.338)	(3.338)
Constant	51.779***	63.443***	25.598***	50.656***	50.656***
	(1.532)	(2.433)	(2.102)	(1.868)	(1.868)
Observations	156	156	156	156	156
R-squared	0.000	0.001	0.004	0.001	0.001
Controls	No	No	No	No	No
	Openness	Conscientiousness	Extraversion	Agreeableness	Emotional Stability
Panel B: Personality scores					
female	0.222	0.492*	0.365	0.333	0.341
	(0.265)	(0.277)	(0.262)	(0.281)	(0.269)
Constant	5.556***	6.119***	6.024***	6.167***	6.048***
	(0.157)	(0.176)	(0.170)	(0.173)	(0.182)
Observations	162	162	162	162	162
R-squared	0.003	0.012	0.007	0.006	0.006
Controls	No	No	No	No	No

Note: This table shows the coefficients from bivariate regressions of an aptitude sub-score (in Panel A) or a personality score (in Panel B) on a female applicant dummy variable. Each aptitude score can range from 0 to 10, with 10 being the best performance. Each of the five different personality scores can range from 1 to 9, with 9 indicating applicants' personality is most open/conscientious/extraverted/agreeable or emotionally stable. The regressions are based on applicants from our main sample. Heteroskedasticity robust standard errors are in parenthesis. *** p<0.01, ** p<0.05, * p<0.1

Table A7: Predictors of Python Score Guess

	(6)	(7)	(8)	(9)
	Dep Var: Std. Python Score Guess			
Predictor	Coef.	se	p-val	N
Currently studying	-0.118	0.031	0.000	6,238
Currently employed	-0.021	0.010	0.039	6,238
Years coding exper.	0.026	0.003	0.000	6,238
Learned coding				
in university	0.260	0.036	0.000	6,238
in online course	0.026	0.030	0.389	6,238
self-taught	0.021	0.026	0.419	6,238
other	0.061	0.033	0.066	6,238
Educational attainment (Base: Some high school without graduation)				
High school graduate	-0.033	0.141	0.814	6,238
Some college	0.063	0.125	0.616	6,238
2-year college	0.305	0.135	0.024	6,238
4-year college	0.433	0.121	0.000	6,238
Postgrad	0.451	0.124	0.000	6,238
Aptitude Test (0-10)				
Cognitive Abilities	0.117	0.013	0.000	3,415
Abstract Reasoning	0.060	0.008	0.000	3,415
Critical Thinking	0.080	0.009	0.000	3,415
Reasoning Ability	0.066	0.008	0.000	3,415
Attention to Detail	0.077	0.011	0.000	3,415
Aptitude test (combined)	0.013	0.001	0.000	3,415
Personality Test (1-9)				
Openness	0.037	0.016	0.021	2,823
Conscientiousness	0.034	0.013	0.012	2,823
Extraversion	0.040	0.014	0.005	2,823
Agreeableness	0.028	0.014	0.042	2,823
Emotional Stability	0.039	0.014	0.005	2,823

Note: This table shows coefficients, standard errors, and p-values, and number of observations from regressions of employers' guesses of applicants' Python test scores. For educational attainment, these come from two single regressions with one dependent variable (either actual score or guess of actual score) and five educational attainment dummies. For the remaining predictors shown in Column (1) the coefficients, standard errors, p-values, and number of observations come from bivariate regressions with the predictor in Column (1) as the only independent variable. Each of the five different Aptitude test sub scores can range from 0 to 10, with 10 being the best performance. Aptitude Test (combined) is the unweighted average of all five sub scores. Each of the five different personality scores can range from 1 to 9, with 9 indicating applicants' personality is most open/conscientious/extraverted/agreeable or emotionally stable. P-values in Column (4) are based on standard errors clustered at the employer level.

Table A8: The Effect of Aptitude and Personality Treatment on Prediction Error

	(1) Prediction Error	(2) Prediction Error
Aptitude treatment	-7.168***	-7.318***
	(0.813)	(0.832)
Personality treatment	-2.135**	-1.579
	(1.040)	(1.041)
Constant	33.804***	27.310***
	(0.611)	(2.574)
Observations	6,238	6,238
R-squared	0.019	0.035
Controls	No	Yes

Note: The dependent variable in both regressions is the difference absolute value of the difference between applicants' actual Python score and an employer's guess of this score. Neither of those values are standardized before calculating our "prediction error" dependent variable. The independent variables shown in the table are dummy variable for the employer being in the aptitude treatment. Additionally, the controls in column 2 consist of the following control variables: five dummies for educational achievement, currently studying, currently working, three dummies for how applicant learnt coding (at university, online course, self-taught, other), years of programming experience, and programming experience squared. Heteroskedasticity robust standard errors are in parenthesis. *** p<0.01, ** p<0.05, * p<0.1

Appendix A1 – Sample Restrictions

A1.1 Applicant Sample

In total we had 2,132 eligible applicants. We classify applicants as eligible if they lived in the US, were either male or female, provided an email address, and were not excluded for other reasons (e.g. applied for job 1 as well as job 2). "Eligible" in this context refers to the inclusion in our analysis. Some of "non-eligible" applicants (e.g. non-binary applicants) were still considered for the job.

We invited 1,608 eligible applicants (821 for job 1 and 787 for job 2) to take the Python test. These invited applicants consist of all female applicants and a random sample of all male applicants. Of all invited applicants, 310 were female and 1,298 were male. Here are the numbers of invited applicants per treatment:

- Treatment #1 (no-incentive, aptitude): 409
- Treatment #2 (no-incentive, personality): 407
- Treatment #3 (incentive, aptitude): 396
- Treatment #4 (incentive, personality): 396

In our analysis, we focus on applicants in treatments 1 and 2. Applicants in treatments 3 and 4 were offered a fee for completing the Python test and a reward for performing well on it. This treatment was part of a different research project. We exclude those from our analysis as their application procedure deviated from the norm.

Applicants assigned to treatments 1 and 3 were sent a Mettl test link containing the Python test and an aptitude test. Applicants assigned to treatments 2 and 4 were sent a Mettl test link containing the Python test and a personality test.

Of the 816 eligible and invited applicants from treatments 1 and 2, 331 attempted the Python test. We classify applicants as attempting the test if they have attempted the long and the short programming question. In our data, we recognize those as not having "NA" for the testcases evaluation. Here are the number of applicants who attempted the test per treatment:

- Treatment #1: 166
- Treatment #2: 165

Of the 331 applicants in treatments 1 and 2, we used 318 applicant profiles in the stage 2 experiment. We did not use the remaining 13 applicants for various reasons. For example, we did not include some applicants in the stage 2 experiment because they used different email addresses when filling in the applicant survey and when entering their details in the Mettl online form and we only recognized this after running the stage 2 experiment.

These 318 applicants form our main sample for estimating the gender gap in programming skills. In robustness checks, we additionally exclude applicants who scored zero on the Python test.

A1.2 Employer sample

Our sample of employers consists of 748 individuals who have read through the general descriptive part of the stage 2 experiment. Out of those, 641 employers started the guessing task and 625 finish it. Table A10 illustrates the sample by employer treatment.

Table A10: Initial Assignment and Drop-out by Treatment

Treatment	Initial assignment	Drop-out	Attrition rate
Baseline	251	11	4.4%
Aptitude	256	35	13.7%
Personality	241	77	32.0%

Attrition and the effect of the aptitude and personality treatments on the perceived skill gap

We observe some differences in attrition rates between treatments. One possible reason for the difference in attrition rates could be that participants receive additional information on aptitude or personality as well as associated control questions in the information treatments. This may lead to more impatient participants dropping out. There is

some evidence supporting this interpretation: among those who completed the experiments, participants in the information treatments passed slightly more comprehension questions in their first attempt compared to the baseline treatment (7.93 in aptitude, 8.23 in personality, and 7.72 in baseline; see Table 4).

While attrition does not matter for estimating the average perceived skill gap among employers who completed the experiment *within* each treatment, it could lead to differences *between* treatments, which could affect our interpretation of Finding 4 on the effect of additional information in reducing perceived skill gap.

However, the difference in attrition rates is very unlikely to affect the interpretation and validity of Finding 4. We outline four reasons below:

First, it is unclear why the kind of employers who drop out of the experiment would also be the ones who believe that female applicants are particularly worse programmers compared to male applicants. Even if they did, we would expect largest reduction in the perceived skill gap for the personality in which 32.0% of employers have dropped out compared to the aptitude treatment which only has a drop-out rate of 13.7%. Yet, the results show the opposite pattern. While employers in the personality treatment perceived female applicants to perform 0.012 SD worse than male applicants, the perceived skill gap is flipped in favor of men in the aptitude treatment where employers perceived male applicants to be 0.048 SD better in programming.

Second, we show in Table 4 that the average characteristics of employers is very similar across the three treatments. The only statistically significant difference between employers in those treatments are the differences in number of comprehension questions passed which, as mentioned above, can serve as proxy for patience when filling out the questionnaire.

Third, our results do not change if we reweight our estimates giving more weights to people who we were less likely to observe in the sample as predicted by their comprehension questions. We do this reweighting in three steps. Firstly, we estimate the probability of dropping out based on the number of comprehension questions passed using a logistic regression. Secondly, we generate the predicted probability of dropping for each employer.

Lastly, we re-estimate Table 8 using the inverse of those predicted probabilities as weights. Using this approach, our point estimates hardly change. Without controls, the perceived skill gap in the baseline treatment is 0.120 SD (compared to 0.118 SD without weights); the perceived skill gap is 0.046 SD in the aptitude treatment (vs 0.048 SD) and -0.007 SD in the personality treatment (vs. -0.012 SD). Adding control variables does not change the picture. The perceived skill gap conditional on applicant controls is now -0.141 SD (compared to -0.141), aptitude treatment -0.035 SD (vs. -0.035 SD), personality treatment -0.23 SD (vs -0.026 SD).

Finally, we can show with back-of-the envelope calculations that the perceived skill gap among the dropped-out employers would have to be implausibly large to account for all the differences in the treatments. We can see from Table A10 that if 24 fewer people had dropped out from the aptitude treatment, the attrition rates in this treatment would have been 4.3%, almost identical to the attrition rate of 4.4% in the baseline treatment. We can therefore compute how large the average perceived skill gap for these 24 employers would have to be to make the average of the aptitude treatment as large as the average of the baseline treatment. The answer is -1.65 SD.[29] This number is too large to be plausible. It is 14 times as large as the average perceived skill gap in the baseline treatment. Using the same approach, we can see that if 67 more employers had dropped out of the personality treatment, the drop-out rate would have been 4.1%. To equalize the perceived skill gap between the personality and the baseline treatment, these 67 employers must have had an average perceived skill gap of -0.377 SD. While smaller, this number is again implausible large. It is 3.2 times as large as the average perceived skill gap in the baseline treatment.

Taken together, our finding on the differences in the average perceived skill gap between treatments is highly unlikely to be affected by attrition.

[29] If (221 (actual employers in aptitude treatment) * 0.048 SD (average perceived skill gap of actual employers in aptitude treatment) + 24 ("missing employers" from aptitude treatment)* X)/245 (number of actual + missing employers) = -0.118 SD, then X = -1.65 SD.

Appendix A2 – Details about the Python Test

Before we advertised for the job, we developed a comprehensive assessment of programming skills. Our skill measure is the score on a Python programming skills test. This test was administered by Mettl, a company that specializes in pre-hiring screening and job applicant skills assessment.[30] The test consists of two hands-on programming tasks in which applicants are asked to write two programs in an online coding simulator. Mettle offers long and short hands-on Python programming questions. The difference between long and short questions is in the complexity of the task and how much time the applicant has for completing it.

To select the programming test questions from the Mettle test database, in June 2019 we surveyed professional Python programmers located in the US. We aimed to collect 15 valid observations. To achieve this goal, we surveyed 26 programmers. We dropped 11 observations because they were either not residing in the US, failed at least one attention check, had technical difficulties with filling in the survey, or did not understand all coding metrics.

In the survey, the programmers read the descriptions of three commonly used long programming tasks and four commonly used short programming tasks with examples. After reading those, the programmers ranked the long and short questions in terms of their usefulness for *"knowing a job applicant test score is for predicting their Python coding skill"*. We then selected the long programming question with the lowest average rank and the short programming question with the lowest average rank for the test.

With this procedure, we selected the following two questions. For the first question (long question), the applicant has to write a program that adds up the largest row sum and the largest column sum from any N-rows*M-columns array of numbers. For the second question (short question), the applicant has to write a program that can determine whether characters in the first string can be rearranged to form characters in the second string.

[30]Mettl is one of the largest and fastest growing online talent measurement solution providers globally. The company has been at the forefront of online assessment technology since its inception in 2010. It is assisting over 1,500 global companies, 24 Sector Skill Councils and 15 educational institutes across 80+ countries.

Programming skills consist of many dimensions. To create metrics to evaluate the applicants' test performance, we held conversations with several industry and academic experts. The following five metrics were identified as the most appropriate from these conversations:

1. Test cases score

Input 10 examples and check if the program returns the correct answers. The more correct answers, the higher the score. This measures if the applicant's program does what it is supposed to do.

2. Efficiency score

This measures how fast the applicant's program runs compared to a benchmark. A good programmer should be able to write efficient programs.

3. Complexity score

A program (Pylint) is used to analyse the applicant's code and calculate the McCabe's Cyclomatic complexity score. Less complex code is easier to test and maintain.

4. Coding convention score

A program (Pylint) is used to analyse whether the applicant's code follows coding conventions as outlined in the style guide for Python code PEP 8. Codes that follow a convention are easier to understand by others such that other programmers can repair, maintain or build on them.

5. Frequency of error score

The number of programming errors in the applicant's code is measured. The more errors there are in the program, the more likely something will go wrong with the program.

We determined the weights of these measures with the help of the responses of the above-mentioned programmers' survey. In this survey, we described each of the five measures and asked if the respondent understood it. We asked each respondent to "*decide how much each score should be weighted so that the final score for the applicant is the best measure of their Python coding skill".* By taking the average answers of the 15 respondents, we determined the overall test weights as shown in Table A11.

Table A11: Weights of Python Coding Skill Measure

Score	Weight
Test cases	28.53%
Efficiency	21.40%
Complexity	21.07%
Coding convention	11.53%
Frequency of errors	17.47%

The resulting formula to calculate the Python coding skill measure (i.e. the final Python test score) is:

*Python skill measure = test cases score*0.2853 + efficiency score *0.2140 + complexity score*0.2107 + coding convention score*0.1153 + frequency of errors score*0.1747*