Missing Women in Tech: The Labor Market for Highly Skilled Software Engineers

Raviv Murciano-Goroff*
Boston University

July 30, 2021

Abstract

This paper examines the behavior of job seekers and recruiters in the labor market for software engineers. I obtained data from a recruiting platform where individuals can self-report their computer programming skills and recruiters can message individuals they wish to contact about job opportunities. I augment this dataset with measures of each individual's previous programming experience based on analysis of actual computer source code they wrote and shared within the open-source software community. This novel dataset reveals that candidates' self-reported technical skills are quantitatively one of the most important predictors of recruiter interest. Consistent with social psychology and behavioral economics studies, I also find female programmers with previous experience in a programming language are 11.07% less likely than their male counterparts to self-report knowledge of that programming language on their resume. Despite public pronouncements, however, recruiters do not appear more inclined toward recruiting female candidates who self-report knowing programming languages. Indeed, recruiters are predicted to be 6.47% less likely to express interest in a female candidate than a male candidate with comparable observable qualifications, even if those qualifications are very strong. Ultimately, a gender gap in the self-reporting of skills on resumes exists, but recruiters do not appear to be adjusting their response to such signals in ways hypothesized by the statistical discrimination theory that could increase the representation of women among software engineering recruits.

*Thank you to Aaron Mangum for his help with wrangling the data. I would like to thank Jonathan Levin, Timothy Bresnahan, Muriel Niederle, Richard B. Freeman, and Paul Oyer for their mentorship and advice. Thank you to the workers at the hiring and recruiting platform who facilitated this research and provided many helpful conversations and insights into the dynamics of the tech labor market. Thank you to Bledi Taska for providing and supporting my use of the Burning Glass data. Thank you to Andrew Chamberlain at Glassdoor. Thank you to Claudia Goldin, Prasanna Tambe, Adina Sterling, Julia Lane, Russell Funk, Ina Ganguli, Caroline Flammer, Sanaz Mobasseri, and the seminar participants in the IO and Labor workshops at Stanford University for their helpful comments. My research has been funded by the Ewing Marion Kauffman Foundation as well as the B.F. Haley and E.S. Shaw Fellowship for Economics through a grant to the Stanford Institute for Economic Policy Research. This research was also supported by NSF SciSIP Award 1932689 and NSF Education and Human Resources DGE Awards 1761008. The contents of this publication are solely the responsibility of myself.

1. Introduction

Despite concerted efforts by many tech companies, the engineering workforce remains highly gender imbalanced.[1] In 2015, tech giants such as Google, Facebook, and Twitter had a mere 17%, 15%, and 10% of their respective technical staff positions filled by female engineers.[2] The gender imbalance in tech is often attributed to factors on both the labor supply and the labor demand side. On the labor supply side, studies have revealed that women might be choosing different educational tracks (**?**), have different levels of self-confidence for computer related tasks (**?**), apply to different jobs (**???**), and leave technical career paths at higher rates (**??**). On the labor demand side, recruiters may be discriminating against female candidates (**???**).

Although research has made progress in understanding women's underrepresentation in the tech workforce, determining which recruiting practices are likely to increase diversity requires knowing both if there are gender differences in the way that job seekers signal their skills to potential employers as well as how recruiters respond to those signals. In particular, if gender differences exist in how job seekers represent their skills on their resumes, recruiters who do not adjust for those differences—such as those using gender-blind resume reviews—may overlook qualified female candidates (**?**).

On the labor supply side, laboratory experiments document gender differences in self-confidence and self-assessments (**???**). While these factors will likely influence job seekers' decisions, evidence of how male and female job seekers with similar experience choose to signal their skills to employers is still needed. On the labor demand side, the statistical discrimination literature hypothesizes that recruiters who are aware of gender differences in the signaling of skills would update their responses to observing such signals over time (**??**). Empirical evidence testing if actual recruiters' actions are consistent with that hypothesis is lacking.[3] Therefore, in order to more completely understand dis-

[1]The American Association of University Women estimates that 26% of these jobs are held by female engineers.

[2]The data for these statistics comes from the equal-opportunity data websites for these firms as well as news reports, such as https://www.huffingtonpost.com/2015/03/27/women-in-tech_n_6955940.html.

[3]**?** show in a behavioral experiment that study participants who were asked to role-play employers did not take into account that male candidates would over-estimate their abilities in a basic arithmetic task. The setting for this paper is different as candidates self-assess and self-report on specific technical skills, such as knowledge of particular programming languages, which they have invested in learning. Furthermore, unlike a lab experiment with role-playing recruiters, the recruiters studied in this paper are professional recruiters employed by actual tech companies.

parities in recruiting, data is needed on the abilities of job seekers, the skills they present to recruiters, and the response of recruiters to those self-reported skills.

In this paper, I investigate how job seekers for software engineering positions self-assess and self-report their skills on their resumes (henceforth "self-reported skills") as well as how tech company recruiters respond to self-reported skills when selecting candidates for further review. The unique data used in this paper comes from a large online recruiting platform. On the platform, job seekers post digital resumes with a list of skills they feel proficient in. For a subsample of those candidates, I am able to find actual previous computer code they created and uploaded online. By comparing the programming skills individuals claim knowledge of with their actual previous coding work, I quantify the extent of gender differences in the self-reporting of programming abilities.

In addition, recruiters from major tech companies subscribe to this platform in order to find and contact potential hires. In my data, I observe which candidates on the platform recruiters expressed interest in contacting. By comparing candidates with similar observable attributes, I test if recruiters respond to self-reported skills on the resumes of male and female candidates in a way consistent with adjusting for gender differences in the propensity to self-report skills.

The tech industry is a useful context for studying how job seekers choose to self-report skills, and how recruiters choose to respond to self-reported skills. The programming skills of a job seeker can be directly observed in the computer code that individual has written. In addition, recruiting is common within the labor market for software engineers, and tech company recruiters often filter candidates based on knowledge of particular programming languages (**???**).[4] Finally, with 1.3 million jobs in the United States, the labor market for software developers is economically significant.[5]

This paper presents three empirical results. First, female programmers are 11.07% less likely to self-report knowing programming languages that they have experience in than their male counterparts. This gender gap is not explained by controls for candidates' educational backgrounds, current occupations, experience programming in lan-

[4]Job advertisements as well as early stages of the hiring pipeline for information systems positions emphasize technical skills, while soft skills are important for being hired as well as career progression (**?**).

[5]Bureau of Labor Statistics predicts that the number of positions in software development will grow about four times faster than the average occupation over the next ten years (https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm).

guages, or the usage of the candidates' code by other programmers. Instead, the gender gap appears to reflect differences in self-assessments regarding technical skills.

Second, among all of the information recruiters could view about candidates, tech recruiters are particularly responsive to the technical skills that individuals self-report on their digital profile. Given that self-reported skills are not verified in any way, recruiters might have ignored this portion of the resumes and treated it as cheap talk. Instead, recruiters appear to utilize self-reported skills as indications of candidate's self-confidence in a skill as well as their interest in working with that skill. The predicted benefits of self-reporting are more limited, however, for those with higher levels of experience in a programming language.

Third, recruiters do not adjust for gender differences in the self-reporting of skills. As female candidates are less inclined towards self-reporting knowledge a programming language relative to their male counterparts with similar actual previous coding work, recruiters could show a preference for female candidates when reviewing male and female candidates with similar credentials who both self-report knowing a programming language. Empirically, however, I do not find evidence that recruiters are more inclined toward recruiting female candidates who self-report programming skill than otherwise similar male candidates.

This paper makes two contributions to the existing literature. The first is an empirical contribution to the literature on gender differences in the decisions of job seekers. These works demonstrate gender differences in many decisions that impact labor market outcomes, such as the propensity to enter into competitive environments (**???**), the propensity to apply for a job given the number of other applicants (**?**), self-assessments provided to potential employers (**?**), the likelihood of reapplying to an employer following a rejection (**?**), and the probability of negotiating with potential employers regarding wages (**?**).

This paper shows that female job seekers also self-report their technical skills on their resumes at a lower rate than their male counterparts despite similar levels of previous experience. This paper compliments previous works about gender differences in self-confidence and self-promotion. Many of those papers ask students in behavioral labs to take a brief quiz and then estimate how many questions they answered correctly (**?**) or self-evaluate their subjective performance (**?**). Job seekers on the platform studied in this paper decide whether or not to self-report specific computer programming skills that they have invested in learning, developed over time, and have significant experience using. Rather than measuring knowledge or skills using quizzes, I measure programming skills directly by examining actual computer source code written by the job seekers.

The second contribution of this paper is testing a hypothesis from the statistical discrimination literature. Audit studies reveal that recruiters respond to the same information on the resumes of male and female candidates differently (**????**). One reason that recruiters might respond to similar resumes differently is because they are using observable information, such gender and self-reported skills, to make inferences about candidates' unmeasured actual abilities (**??**). The theory of "belief flipping" hypothesizes that if a group of job applicants are at a disadvantage then candidates from that group who are hired would be regarded more highly than individuals hired from the non-disadvantaged groups (**?**). A parallel theory could argue that if female candidates have a higher bar before they self-report knowledge of technical skills, recruiters who are aware of this gender gap should favor female candidates who self-report that skill over male candidates with similar resumes. In a laboratory experiment, **?** showed that individuals do not take into account gender differences in self-assessments when 'hiring' others for performing a simple mathematical task. This paper builds on that work by examining the actions of professional recruiters who can adjust their response to self-reported skills over time based on comparing the resumes of job seekers with their subsequent performance in interviews.

2. Setting

Many tech companies find and recruit individuals based on digital resumes from platforms like LinkedIn. The platform whose data I use for my analysis in this paper facilities such recruiting for companies looking for software engineers.[6] The platform integrates data from two discrete websites. One website is for job seekers. This site enables indi-

[6]Known as a "passive recruiting" platform, employers on the platform initiate searches for qualified individuals whose resumes fit with open positions, and actively coax the candidates into interviewing for those openings. This is known as "talent mining". The candidates are "passive candidates" since they are not actively searching for a job (**?**).

viduals to create digital resumes showcasing their educational credentials, work histories, and a list of skills they feel proficient in. The list of skills on the resumes, known as "self-reported skills," could include technical skills, such as the names of programming languages or software packages, as well as non-technical skills, such as "Management" or "Public Speaking." Anyone, regardless of whether they were interested in being recruited for programming jobs or non-technical positions, could create a digital resume on the job seeker website.

The second website is intended for recruiters searching for computer programmers. The recruiter website allows employers to search through profiles of the individuals who have uploaded digital resumes on the job seeker website. The profiles of each candidate on the recruiter website display the education, work history, and the "self-reported skills" the candidate had written on their digital resume. In addition, the candidate profiles show information that the platform independently collected about the candidates. Specifically, above the list of self-reported skills, candidate profiles display the names of programming languages for which the platform declares that a candidate has "verified" coding experience in. The platform constructs this list of "Verified Languages" by finding and examining open source software, computer source code that individuals write and post online for others to use. The programming languages that a candidate uploaded open source code in are shown to recruiters as "Verified" languages. For each "verified" language the platform also indicates if candidates have a "High" or "Low" level of experience based on proprietary analysis of the candidates' open source code as well as contributions to question-and-answer websites about coding.[7] Only recruiters are able to view the verified languages section of the resume; when composing their digital resumes, candidates do not see what programming languages the platform says they are verified in. The platform also displayed an "overall score" for a candidate just below the candidate's name and geographic location. This score, also based on a proprietary algorithm, is intended to describe the overall 'value' of the candidate's skills, educational background, and work history.

Recruiters can search the candidate profiles by a limited set of criteria: the name of a school attended, the name of a previous employer, geographic location, the platform's

[7]The algorithm for computing this rating is proprietary, and thus the details of what is incorporated into this measure are unclear. This does not impact my analysis, however, as the recruiters and I are aware that both levels of verification are derived from actual code written by candidates.

overall score, and a verified language. Recruiters are not able to search by the gender of the candidate. In addition, recruiters could not search for candidates based on a self-reported skill.

Table 1 summarizes the main elements of the candidate profiles. Profiles also displayed a profile picture of the candidate.

After searching for candidates, recruiters would open a candidate's profile. At the bottom of the candidate profile, just below the self-reported skills list, a recruiter could choose to press a button to "save" the profile. After saving a candidate, the recruiter could view the candidate's contract information and send them messages soliciting job applications or requesting the candidate interview for an open position.

3. Data

This paper utilizes two main datasets. The first dataset, which I refer to as the "Profiles dataset," is a cross-section of profiles available for recruiters to view on the platform in December 2015. For each profile on the platform, I construct 259 variables that summarize the information about the candidate that recruiters would view. I refer to these as the "attributes" of candidates. Details about these attributes are provided in Appendix 8. Among the profile attributes included in the dataset are variables representing if a programming language appeared as self-reported as well as if a programming language is listed as "verified" along with the level of verified experience. In addition to the information shown to recruiters, for each profile I observe if any recruiter saved the profile between March 2014 and November 2016.

I restrict the Profiles dataset to candidates who are most likely to be interest in software development positions. These include candidates located in the United States, who list a bachelor's degree in Computer Science (CS) or related engineering fields, a previous job involving computer programming or at least one self-reported skill related to software engineering. In addition, the candidates must have at least one verified programming language listed on their profile.[8] Finally, the sample includes only profiles

[8]In the Online Appendix, I redo all the analysis using the full-sample of all profiles on the platform at the time. There were over 3 million candidates on the platform. The sample used for my analysis is around 4.56% of the candidates on the platform, but these candidates received 47.84% of the saves from recruiters.

where the first name of the candidate is strongly associated with either the male or female gender.[9]

The Profiles dataset contains 170,886 candidate profiles.[10] Of the candidates, 7.41% are female. Table 2 Columns (1) and (2) show the mean values of attributes displayed on the profiles of male and female candidates. On average, the attributes of the male and female candidates in this sample have some important similarities and differences. Male and female candidates are similarly likely to hold a bachelor's degree in Computer Science (29% and 27% respectively), to list at least one self-reported skill (42% and 41%) and to include computer programming as one of their self-reported skills (40% and 38%). In contrast, a significantly smaller percentage of the female candidates list their current occupation as involving programming (40% of female candidates versus 47% of male candidates). Like many other online communities, open source communities have been shown to not always treat the contributions of female users equitably (**??**). The female candidates represented in the OS Contributors dataset are therefore likely to be particularly committed programmers as they continue to write and upload source code despite potential discrimination on those platforms.[11] Finally, the rates at which recruiters contacted male and female candidates are different: 21% of male candidates and 15% of the female candidates were saved by at least one recruiter between 2014 and 2016.

The second dataset utilized is referred to as the "Candidate-Language Pair" dataset. An observation in this dataset is a candidate and a language that the candidate is listed as having verified experience in. The dataset is restricted to candidate-language pairs where the language is one of the following: Java, C++, C#, Python, Ruby, PHP, and JavaScript. In addition to being some of the most widely used programming languages, the source code of these languages have distinct file extensions making it easier to search open source repositories online and detect when a candidate has uploaded code in that lan-

[9]I drop 14.85% of the observations because I am not able to code the gender of the first name. The most frequent dropped names are Terry, Christian, Chris, Pat, Krishna, Wei, and Kim. **?** showed that recruiters examine Facebook and LinkedIn photos to gauge responsibleness. This implies that recruiters do pay attention to the photos of candidates. Thus, recruiters could make inferences about the gender of the candidate from the photos of candidates shown on the platform in addition to the first names displayed.

[10]No profiles were deleted during this time, and my data agreement did not provide me access to any of the new profiles that were added to the site.

[11]In Appendix 18, I show regression results showing that for the same educational credentials and work history female candidates are less likely to be verified in a language. In other words, despite similar credentials, female candidates are less inclined towards making open source contributions on average.

guage.[12] This dataset contains 418,720 candidate-language pairs.

3.1 Measuring Programming Experience and Reputation in a Programming Language

In order to compare the self-reporting decisions of candidates with similar knowledge of a programming language, I collected additional data on candidates' coding experience.

This paper uses metrics of experience in a programming language inspired by the measures of occupational experience used in the labor economics literature. Experience in labor economics is computed as the amount of work someone has done in an occupation or the amount of time (tenure) that someone has worked in a given position (**??????**).[13] The analysis in this paper controls for experience in a similar fashion. I measure the amount of work that a candidate has done in a language by the number of lines of open source code the candidate has uploaded.[14] I approximate how long the candidate has coded in a language by the number of distinct days in which the candidate uploaded new open source code written in that language. Together, these two measures proxy for a candidate's experience with a programming language.

A challenge with these measures of experience is that they do not capture the quality of a candidate's coding work. Ideally, we would like to know if other coders consider a candidate's programming work to be high quality. Open source coders regularly evaluate each other's code. Specifically, programmers can write computer code—such as a new feature for a computer program or a fix for a software bug—and suggest that it be incorporated into an open source project run by another coder. The owner of that open source project can then choose to accept or reject the code contribution, a decision that frequently involves reviewing the quality of the contributed code (**?**).[15] Previous empirical works find evidence that the acceptance of a coder's contributions is correlated

[12]I excluded markup and style languages, such as HTML, CSS, and Markdown. Languages like Objective-C, popular for building iOS apps, has file extensions that overlap with other popular languages.

[13]Many of these papers use composite measures based on both schooling and time in the workforce.

[14]This count removes lines of code that were directly copied from the open source contributions of others ("forked" code). Some will note that terser computer code that achieves an objective might be higher quality than verbose code that does the same task. While this is often true for a specific segment of computer code, such as a single function, I count the lines of code written over the candidate's lifetime. Thus, this measure is more likely to capture the commitment and the depth of work done by a candidate in a language rather than the concision of a particular function.

[15]The open source program Git refers to this as a "pull request."

with that coder's reputation and programming expertise (????).[16] Leveraging these peer-reviews, I count the number of lines of code that a candidate has contributed to other's open source projects as an indication of the reputation of the candidate's coding work.

There are two nuances to note with this measure of programming reputation. First, there are indications that open source project owners are on average less inclined toward accepting code contributions from coders who are easily identifiable as women (?). For the analysis in this paper, this would mean that the number of lines of open source code accepted by others might somewhat understate female candidates' actual coding quality relative to male candidates.

Second, candidates are aware of if their code contributions are accepted or rejected by others. Therefore, the number of contributed and accepted lines of code is also an external validation of one's own skills.

In Appendix 10, I demonstrate that the main results of this paper are robust to using alternative metrics of coding experience and reputation. Finally, because the measures of experience are right-skewed, I use logarithmic transformations of these measures in the regression analysis.

3.2 Measuring Recruiter Interest in a Candidate

Interest in a candidate is measured by observing if a recruiter on the platform saved a candidate's contact information between March 2014 and November 2016.[17] Recruiters on the platform saved candidates in order to message them about job opportunities or invite them to interview.

The outcome of a candidate being "saved" is similar to the outcome of a "callback" in the extensive audit study literature. In those papers, researchers send fictitious resumes to employers and record if the employers attempt to contact, or callback, any of the fictional candidates (????). As those studies have noted, this outcome has drawbacks. In particular, while a 'callback' or a 'save' indicates interest in a candidate, it does not tell

[16] A possibility is that the reputation of a coder swamps the actual quality of contributed code when reviewers decide whether or not to accept a contribution. One working paper, using a select sample of 28 Java open source projects, found that the presence or lack of bugs in the code itself was not a strong predictor of whether or not a code contribution would be accepted (?).

[17] The window of time examined for the outcome has the potential to produce mismeasurement. In Appendix 13 I show that mismeasurement is unlikely to be significantly impacting the results in this paper.

us if the employer eventually extended the candidate a job offer.[18] In this paper's setting, messages between candidates and employers occur through emails or phone calls and are not recorded by the platform. Therefore, I do not have data on how many times a recruiter messaged a candidate after saving the candidate's profile, how many of those contacted candidates responded, or how many candidates were eventually hired by the firms.[19] Instead, I interpret saving a candidate as an indication that the recruiter believes the candidate is qualified for an open position and worth contacting. As **?** note, finding candidates is time-consuming for recruiters, and candidates who are not saved by a recruiter are less likely to be interviewed or extended a job offer from that firm. Finally, pre-hire decisions, such as which set of candidates a recruiter saves, can influence the decision of which candidate eventually gets hired by a firm (**???**).

In the main text of the paper, I focus on the binary outcome of whether or not a candidate was saved by any recruiter. An alternative outcome is the number of times that a candidate was saved. Given the large number of candidates and relatively small number of employers on the platform, only 20.47% of candidates were saved during this time period and a mere 3.59% were saved by more than one recruiter. Therefore, as shown in Appendix 11, the main results of the paper are qualitatively similar if a binary or count outcome is used.[20]

4. Empirical Framework

4.1 Measuring Gender Biases in Self-Reporting

The first question this paper asks is do male and female coders, with similar experience in a programming language, self-report knowing that language on their profiles at the same rate? My empirical approach draws on the literature that uses "reverse regres-

[18] To date, the only paper that I know of that connects data on the messaging of candidates on passive-recruiting websites with actual hiring outcomes comes from a Swedish Public Employment Service recruiting platform, **?**, which finds a strong correlation between being contacted on that platform and eventually being hired by some firm during the next six months.

[19] While information on which saved candidates were eventually hired is not available to me, in the Appendix 12 I show that candidates saved between January and June 2015 were 33.33% more likely to switch employers between July and December 2015. Again, I do not know if the employer that the candidate switched to is the one that previously saved the candidate.

[20] Furthermore, using a binary outcome makes the results comparable with audit studies that use a binary outcome of a callback (**????**).

sions" to study if a gender gap exists in the qualifications of male and female workers with the same wage (**?**). I estimate the following equation using the candidate-language pairs data:

$$sr_{i,l} = \alpha + \beta female_i + \gamma experience_{i,l} + \theta X_i + \epsilon_{i,l} \tag{1}$$

In this equation, $sr_{i,l}$ is an indicator for if candidate i self-reports programming language l, α is a constant, $female_i$ is an indicator for if the candidate is female, $experience_{i,l}$ represents a vector of measures of candidate i's programming experience in language l, and X_i represents controls for the candidate's educational background and work history.[21] A negative estimated β would imply that female candidates are on average less likely than male candidates to self-report a language, while a positive β would imply that female candidates are more likely to self-report knowing a language than their male counterparts.[22]

Previous literature suggests a number of reasons that might explain a potential gender gap in self-reporting programming languages. For example, on average, male and female candidates might have different educational backgrounds, occupations, or preferences over careers. After adding controls related to those hypotheses, I examine if the coefficient on the gender candidate, β, becomes smaller in magnitude. A smaller or insignificant β would imply that those additional controls explain some of the gender differences in self-reporting (**?**).[23]

[21]Throughout this paper, I primarily rely on linear probability model (LPM) specifications rather than logit or probit specifications. The motivation for using LPMs rather than logit/probits is because of the ease of interpretation of LPM. In addition, LPMs are faster to estimate when running the regressions on the full sample of millions of candidates on the site. All of the regressions were replicated in logit/probit specifications, and the results are qualitatively the same as the LPMs.

[22]It should be noted that this exercise measures if female coders self-report at rates higher or lower than male candidates given the same observable experience. This exercise does not measure whether or not male or female candidates "under-report" or "over-report" skills. In order to assess that, one would need an objective measure of what experience justifies self-reporting. As such an objective and universal bar does not exist, this exercise instead is meant to capture only if male and female coders self-report at relatively different rates. Because such a high proportion of candidates are male, however, recruiters likely form their beliefs about what level of experience justifies self-reporting based on male candidates.

[23]Note that both the educational and career choices of individuals are endogenous. Ideally, one would like to instrument for all of these in this regression analysis. Unfortunately, it is infeasible to find instruments for the plethora of observable information (e.g. school that an individual attended, their major, their employer, and their job title).

4.2 Recruiters Response to Self-Reported Skills and Languages

The second exercise in this paper tests if recruiters show more interest in candidates who self-report skills. I begin by examining which attributes on profiles are most predictive of a recruiter saving that candidate using a hedonic regression using observations from the Profiles dataset:

$$
\begin{aligned}
saved_i = &\sum_{l \in \mathcal{L}} \beta_{1,l}\mathrm{SR}_{i,l} + \beta_{2,l}\mathrm{VL}_{i,l} + \beta_{3,l}\mathrm{VLSR}_{i,l} + \beta_{4,l}\mathrm{VH}_{i,l} + \beta_{5,l}\mathrm{VHSR}_{i,l} \\
&+ \eta + \gamma SRS_i + \theta X_i + \epsilon_i
\end{aligned}
\tag{2}
$$

In this equation, $saved_i$ is an indicator for whether or not any recruiter saved profile i between 2014 and 2016. As covariates, I include indicators representing the five different ways in which programming languages appear in a profile: self-reported only (SR), verified with low (VL) or high experience (VH), or both self-reported and verified with low ($VLSR$) or high ($VHSR$) experience. I include these indicators for each of the follow languages: JavaScript, PHP, Ruby, Python, C#, C++, and Java. This set of languages is denoted by $\mathcal{L}$. I also include a vector of indicators (SRS) that represent self-reported skills, other than programming languages, shown on the profile. Finally, I include a constant (η) and a set of controls (X_i) for the candidate's attributes displayed on the profile. This vector includes the attributes documented in Appendix 8. The estimated coefficients represent the average predicted increase in the probability that a profile is saved when that profile displays a particular attribute. By comparing the magnitudes of these coefficients, I explore the relative demand associated with candidates who on their profiles advertise different sets of skills, educational credentials, and previous jobs.

The estimates of interest in this regression are the relative magnitudes of the predicted probability of being saved when a candidate is verified in a language versus when they are verified and self-reported that language ($VLSR$ versus VL and $VHSR$ versus VH). If recruiters disregard whether or not the candidate listed that language among their self-reported skills, the coefficients on $VLSR$ and VL or $VHSR$ and VH would be similar. If, however, recruiters are more likely to save candidates who also self-report then $VLSR$ and $VHSR$ would be greater than VL and VH respectively.

4.3 Different Recruiter Responses to Self-Reported Languages by Male and Female Candidates

I investigate if the predicted recruiter response to the skills listed on profiles is different for male versus female candidates. Methodologically, I draw on the literature estimating gender differences in the returns to skills by adding interactions to wage regressions (**??**). I modify Equation 2 by adding interaction terms between self-reported skills and the candidate's gender. I do this in two ways. First, I add interaction terms for the self-reported skills, including self-reported languages. This reflects one way that recruiters could treat candidates: verified skills are based on actual code written, and thus indicate similar knowledge of a language for both male and female candidates, while self-reported skills could provide different signals of programming abilities by gender.[24] Second, I add interaction terms for all the profile attributes. With this flexible formulation, I can see if there are any profile attributes where recruiters respond more to female candidates versus male candidates.

As I show in this paper, female candidates who self-report knowing a programming language have on average more experience in that language than their male counterparts. Therefore, I examine if recruiters respond more positively to self-reported skills on the profiles of female candidates as compared with those of male candidates. Operationally, I test if the coefficients on the interactions of self-reported languages and the gender of the candidate being female is positive and statistically significant. Negative or insignificant coefficients would indicate that recruiters are not adjusting their responses to self-reported programming languages in a manner consistent with the gender differences in self-reporting.

4.4 Matched Profiles Approach to Investigating Recruiter Responses to Self-Reported Languages

The above tests regarding how recruiters respond to self-reported skills on profiles could be influenced by two confounding factors. First, the estimated recruiter response to self-reported languages could be biased because of misspecification. Instead of valu-

[24] Selection into doing open source work also has the potential to introduce gender biases. That being said, calling "verified" languages "verified" implies that it reflects actual coding abilities.

ing candidate attributes in an additively separable manner, as Equation 2 suggests, recruiters may consider a complex set of interactions of a candidate's attributes when deciding whether or not to save their profile. If this is the case, the magnitude of the coefficients on self-reporting might misrepresent the importance of this attribute appearing on a candidate's profile. Second, self-reporting is endogenous, and candidates who self-promote their knowledge of programming languages are likely to list other desirable information on their profile as well. Correlations between the decision to self-report and other desirable attributes might make self-reporting predict more recruiter attention than it would on its own.

One way to placate the above concerns is to compare the level of recruiter interest in profiles that are similar in many attributes, but some self-report a particular language while others do not. This method is akin to audit studies in which researchers would construct and send similar resumes to potential employers with variation in whether the name on the resume was strongly associated with males or females (**????**). These works obtained precise estimates for the level of discrimination by employers by comparing the callback rates of male and female resumes with otherwise similar attributes.

Approximating that methodology, I constructed "profile groups" by finding profiles that are precisely the same on the attributes that recruiters could use to search for profiles. Specifically, profile groups are matched on the geographic region of the candidate, the languages that the candidate is verified in, whether or not the candidate is currently employed as a coder, which schools the candidate attended, which companies the candidate has worked for in the past, and the overall score given by the platform for the candidate. The rate at which profiles with self-reported languages are saved is then compared with profiles within the same profile group that did not self-report. For example, candidates who self-reported Java would be compared to candidates who did not self-report Java among those who graduated from Princeton University, received their BA in 2005, worked at Google, hold the title Software Engineer, reside in New York, and are verified with low experience in Java.

The advantage of this methodology is that it does not impose parametric assumptions on how recruiters combined and utilized the attributes they could observe on candidate profiles. There is, however, the possibility that self-reporting a language is correlated with some other attribute on profiles that is not matched on when constructing the profile groups (**?**). While the profiles on this platform had only a few salient sections, this exercise provides suggestive but not causal estimates of the impact of self-reporting on recruiter interest.

For this exercise, only profile groups with candidates who self-reported a language as well as candidates who did not self-report a language are used in the identification of the estimated coefficients. Because singleton profile groups are dropped, the profiles used for this exercise, which I refer to as the Matched Profile dataset, are a subset of the Profiles dataset. Summary statistics for the matched profiles are shown in Table 2 Columns (3) and (4). As shown, this sub-sample is slightly younger, less experienced, and less skilled than the full sample.

Using the profile groups, I estimate the following equation on the Matched Profiles data:

$$saved_i = \mu_{g(i)} + \sum_{l \in \mathcal{L}} \beta_l sr_{i,l} + \epsilon_i \tag{3}$$

This equation predicts whether or not a profile was saved by a recruiter between 2014 and 2016, $saved_i$, by a fixed effect for the profile group, $\mu_{(g(i))}$, and an indicator for if the profile lists language l among the self-reported skills section, $sr_{i,l}$. The coefficient on $sr_{i,l}$ represents the average difference in the probability that a profile with language l in the self-reported skills list is saved by recruiters relative to those that do not. A positive β_l would indicate that recruiters are responding to self-reported languages. For testing if recruiters adjust to gender differences in the supply side, I also include interaction terms between the candidate being female and the self-reporting of programming languages.

5. Results

5.1 Gender Differences in the Propensity to Self-Report Programming Language Skills

I begin by examining if male and female candidates self-report programming languages on their resumes when controlling for the amount of code previously written in that language. In Figure 1, the 418,720 candidate-languages pairs from the OS Contributors dataset are grouped into quartiles by language according to the total number of lines of open source code the candidate uploaded over his or her lifetime. The vertical axis shows the probabilities that male and female candidates within each quartile self-report their verified programming languages as self-reported skills. The figure reveals that within every quartile female coders have a lower propensity than their male counterparts to self-report knowledge of programming languages on their resumes.

Table 3 quantifies the gender difference in the propensity to self-report programming languages. This table displays the coefficients estimated from Equation 1 using the candidate-language pairs dataset. Column (1) shows that on average female coders are 3.1 probability points less likely to self-report knowledge of a programming language they have uploaded open source code in. Relative to the average rate that all candidates self-report programming languages, this implies a 11.07% gender gap in self-reporting. Column (2) adds the main measures of previous experience in a language as controls. The coefficients on all of these measures are positive and significant, demonstrating that candidates with more experience in a language are also more likely to self-report knowledge of that language. With these controls, the estimated gender gap shrinks to 2.2 probability points, or a 7.86% difference, but remains significant. This shows that controlling for measures of previous experience can only explain a portion of the observed gender gap in self-reporting programming language skills.

Column (3) investigates if differences in educational backgrounds and occupations can explain the remaining gender gap in self-reporting. That column adds controls for the educational and occupational history of the candidate, including indicators for the candidate's college major, year that the candidate graduated, the universities attended, current and past employers, and current occupational title. The estimated coefficients on having a bachelor's degree in Computer Science and currently being employed as a coder are both positive and significant. One might have thought that candidates with degrees in Computer Science would not feel the need self-report knowledge of individual programming languages. Instead, self-reporting programming languages is positively correlated with that educational credential and professional coding experience.

In addition, if female candidates are more likely to know multiple languages, they may be more selective in which languages they list on their resumes. Therefore, in Column (3), I also include indicators for all the other languages that the candidate is verified in. The estimated coefficient on the candidate being female is -0.021 and significant, indicating that knowledge of other languages does not explain gender differences in the propensity to self-report.

Programming languages come in many varieties, are used for different types of work, and are often learned in different contexts. In Columns (4)-(8) of Table 3, I explore the heterogeneity in the gender gap in self-reporting across different programming languages. Each column displays the results of estimating Equation 1 for candidates with previous experience in a specific language. For all seven of the languages considered in this paper— four shown in this table and three shown in Appendix 16—female candidates are less likely to self-report knowledge of that language.

The languages typically associated with web development show large gaps in self-reporting. The popular web development language JavaScript, shown in Column (4), has an estimated coefficient on the gender of the candidate of -0.022, which relative to the mean rate of self-reporting this language translates to female coders being 8.06% less likely to self-report knowing this language than their male counterparts. The coefficients for Ruby and PHP similarly show female coders self-reporting at rates 23.5% and 12.8% lower respectively.

Languages that are frequently taught in introductory programming classes show smaller gaps in self-reporting. Columns (6) and (8) show the estimates for the programming languages Python and Java, two of the commonly used languages for introductory classes on programming and popular languages for industry jobs (**??**). For Python, the estimated coefficient on the candidate being female is -0.018, which describes a 6.2% gender gap. For Java, the estimated coefficient of -0.016 represents a 4.57% gender gap. In contrast to self-taught languages, where candidates decision to self-report is entirely based on self-assessment of their proficiency, candidates who received formal training in a programming language can consider their performance in the class when deciding whether to self-report the language or not. In addition, given the popularity of these languages, candidates might be less selective about listing knowledge of these languages.

Table 4 Column (1) pools the languages and adds interaction terms between the gender of the candidate and the measures of experience. If these interaction terms are positive, it would imply that female coders with high levels of experience close the gender gap in self-reporting. The interactions with the amount of code uploaded and the number of days with uploads are insignificant. The interaction with the number of lines of open source accepted by others is -0.03 and significant at the 10% level. Taken together, these estimates show that the gender gap in self-reporting language is present through much of the experience distribution. This is reinforced in Appendix 14, where I estimate the gender gap in self-reporting by quartile.

A number of possible motivations could influence the self-reporting of programming language skills. In particular, if female candidates who do not self-report programming skills are simply less interested in either receiving messages from recruiters or pursuing careers in programming, the observed gender gap might reflect efficient signaling.

One possibility is that female coders are already employed at desirable tech companies, and thus have less interest in messages about jobs at other employers. As shown in Table 2, female coders in this sample are somewhat more likely to be working at a top tier tech company already. Columns (3)-(11) of Table 3, however, include controls for the candidate's current employer as well as how long the candidate has been at their current employer. Therefore, differences in current employers cannot explain the observed gender gap in self-reporting.

Another possibility is that female candidates may be messaged more than male candidates, and in order to avoid garnering more attention from recruiters, female candidates might choose not to self-report all of the skills they posses on their resumes. Only 15% of the female candidates in my sample were saved one or more times and less than 1% were saved by recruiters from more than three different companies. Therefore, female candidates are likely not receiving large volumes of messages because of this particular platform. In addition, I estimate the gender difference in self-reporting after controlling for the number of times that a candidate was previously saved in Table 4 Column (2). The estimated gender gap remains approximately 2 probability points. While being saved does not correlate perfectly with the number of messages that a candidate receives from recruiters on this platform, it does not appear that even the sought-after female candidates received excessive recruiter attention on this platform. While the possibility remains that female coders are more heavily recruited through other channels, and thus are less inclined to self-report skills on this particular platform, it seems somewhat unlikely that candidates would choose to post a digital resume to be seen by recruiters but specifically leave off technical skills they possess.

Female coders may have on average different preferences over jobs involving programming. Studies have found that women in technical roles are less likely to remain on technical career paths (**??**). This could be because of differences in preferences over careers, differences in self-assessments that constrain career-relevant decisions, or because of unpleasant or discriminatory work environments (**???**). If female engineers are less likely to remain in programming related jobs, they may decide to forgo self-reporting their technical skills and instead emphasize other skills aligned with jobs that do not involve programming. In Column (3) of Table 4, I show the results of estimating Equation 1 using only candidates who continue to upload open source code during the year after the profile data was collected (2016-2017). The estimated coefficient on the candidate being female is -0.012 and significant. Thus, even candidates who continued to code show a small gender gap in self-reporting. Since the samples are different, this coefficient is not directly comparable with the previous results. That being said, it seems likely that some, but not all, of the gender gap in self-reporting is related to differences in preferences over coding in the future.

Another possibility is that female candidates anticipate lower returns to possessing technical skills, and thus choose not to highlight technical skills. This idea has been noted as a possible motivation differences in human capital investment choices (**?**). While the data available to me does not allow me to test this theory, it seems unlikely to be able to explain the gender gap in self-reporting in this context. The female candidates in the data have already invested in learning programming languages. Even if the returns to self-reporting are lower for female candidates, the costs of self-reporting—particularly high experience candidates—are very low. Differences in returns to knowing programming languages are more likely to drive the costly decision of whether or not to learn a programming language, and less likely to significantly influence the decision of whether or not to self-report a known programming language.

The above results regarding differences in the self-reporting of programming languages are particular to technical skills. As shown in Table 2, male and female candidates have a qualitatively indistinguishable likelihood of self-report skills on their resumes: 43% of males and 42% of female candidates list at least one self-reported skill.

In Column (4) of Table 4, I estimate Equation 1 using only candidate-language pairs in which each candidate lists at least one self-reported skill on his or her profile. The estimated coefficient on female is -0.015 and significant implying a 2.63% lower rate of female candidates self-reporting programming languages than their male counterparts. The candidates in this sub-sample feel confident enough to declare themselves proficient in some skills, however a small gender gap in self-reporting competence in a programming language remains even within this population. In Appendix 9, I show that female candidates are more likely to self-report non-technical skills on their profiles, such as Customer Service and Public Speaking, than their male counterparts with similar levels of previous experience in programming languages. These gender differences in self-reporting skills could be because female coders have more skills—both technical and non-technical—than male coders or because male coders shy away from self-reporting their non-technical skills. The result that female coders self-report non-technical skills but are more conservative about self-reporting technical skills is consistent with prior studies that show that the gender gap in self-confidence is most pronounced in work that is considered male-typed and is not found for non-gendered and female-typed work (**??**). Additional work, such as a behavioral experiment or finding observational data demonstrating non-technical skills, will be needed to conclusively say which of these factors explains the gender differences in non-technical skills.

The findings in this section reveal a gender gap in self-reporting programming languages even after controlling for experience in those languages. The gap in self-reporting is present for all languages, although somewhat smaller for languages that are frequently learned in introductory computer programming courses. The gender gap in self-reporting programming languages does not appear to be entirely explained by differences in education, occupation, or even interest in computer programming. Finally, gender gap in self-reporting of programming languages appears to be specific to technical skills. Taken together, these results indicate that self-reported programming languages on male and female resumes conveyed different information about the actual experience of those candidates in that language.

5.2 Recruiters Value Specific Technical Skills

The results presented in the previous section show that female candidates self-report knowledge of programming languages at lower rates than their male counterparts. Given the plethora of information recruiters can observe about a candidate on their profile, including educational credentials, work histories, and verified skills, do recruiters pay attention to self-reported skills when deciding which candidates they are interested in? In this section, I investigate if self-reported skills predict recruiter interest.

Table 5 shows the results of estimating Equation 2. In Column (1), the regression is estimated with a dependent variable of whether or not a profile is saved by any recruiter. The coefficients on the self-reporting of technical skills are similar in magnitude to the education and work history information typically found prominently on paper resumes. For example, self-reporting knowledge of the JavaScript runtime enviornment NodeJS is associated with 0.08 higher probability of being saved, while the database program MongoDB is associated with a 0.05 higher probability of being saved. In comparison, candidates who hold a bachelor's degree in Computer Science (CS) are associated with a 0.04 higher probability of being saved by a recruiter. The similar magnitudes of the coefficients on self-reporting specific technical skills with the coefficient on having a credential indicating general technical knowledge reveals that self-reported skills are important predictors of recruiter interest.

The increase in the predicted probability of a candidate being saved from self-reporting a programming language is also large. The indicator for candidates who are verified with low experience in JavaScript is 0.02, which means that that those candidates are predicted to be saved at a 2 probability point higher rate than a candidate with no verified skills. Candidates who are verified with low experience and also self-report knowing the JavaScript language are associated with a 5 probability point higher rate of being saved. Together, these numbers imply that self-reporting the language in addition to being verified is associated with a 3 probability point higher rate of being saved. For those with verified high experience in this language, the comparison shows that candidates who also self-report the language are predicted to be saved at a rate 8 probability points higher than if they did not self-report. The coefficients on languages other than JavaScript show a similar pattern.

The predicted increases in recruiter attention from self-reporting most programming languages are larger for less experienced candidates. Comparing the ratio of the coefficients $VHSR$ and VH with the ratio of the coefficients $VLSR$ and VL for the same language reveals the relative predicted gains from self-reported across experience levels. For the languages JavaScript, Ruby, C#, and Java, the ratios of $VLSR$ and VL are significantly larger than the ratio of $VHSR$ and VH. For example, for JavaScript, candidates with verified low experience are three times as likely to get saved if they also self-report knowing the language. Candidates who are verified high are only two times as likely to be saved if they also self-report.[25] For the other languages, the ratios are not significantly different from one another. These results reveal that for many of the languages the predicted responsiveness of recruiters to self-reported skills is larger for those with less observable evidence of experience in a programming language. As a candidate's observable experience with a programming language increases, the added information that recruiters get from seeing that a candidate self-reported language in addition to being verified gets smaller.[26]

The importance of skills can be quantified by examining the variation in which candidates were saved that can be explained by just the skills on a profile. When Equation 2 is estimated using only the verified languages and self-reported skills present on candidate profiles, the adjusted R^2 is 0.207. When estimating Equation 2 with all profile attributes, the adjusted R^2 is 0.266. The comparison of these two measures implies that almost three-quarters of the variation in which candidates were saved by a recruiter that can be explained through a linear model of the profile attributes could be captured by just using the verified languages and self-reported skills.

These results provide evidence that recruiters paid considerable attention to the languages and self-reported skills on candidate profiles. Even though recruiters could see if a candidate has educational credentials that would imply programming abilities, such as a degree in Computer Science, as well as verified programming experience, self-reported technical skills predict significantly more recruiter interest. A number of aspects of this labor market encourages recruiters to search for candidates based on technical skills. First, the labor market for software engineers exhibits very high churn. Can-

[25] This is significantly different at the 1% level.

[26] This finding has similarities with the result from the statistical discrimination literature that as workers' productivity becomes more observable, employers put more weight on hard-to-observe correlates of productivity instead of the easy to observe ones (**??**).

didates in the Profiles dataset report switching employers on average every 2.4 years. From an employer's perspective, if their employees will only be at their firm for a limited period of time then investing in training them in particular technologies is relatively costly. Instead, employers are likely to search for candidates who already possess experience in the technology stack that their workers will need to use on the job. Second, the labor market is relatively tight. Tech companies have frequently bemoaned the "skills gap" in which they are unable to find adequate numbers of job seekers qualified in the particular technical skills they desire. If finding candidates with proficiency in particular skills is the primary constraint for the employers, recruiters are likely to start their searches filtering candidates on the basis of the skills listed on the candidate profiles. Third, tech companies have complained that many traditional Computer Science curricula do not teach the practical skills required for building production-ready, large-scale computer programs. Indeed, employers may only get benefits only from task-relevant experience, such as working with a particular programming language, rather than general computer skills (**?**).Therefore, employers are less inclined to limit their searches based on pedigree. Finally, recruiters are incentivized to message candidates who are likely to respond to respond to their messages and apply for job opportunities at their employer. Coders, who often have strong preferences for working with specific technologies (**?**), express their interest in working with particular technologies through the self-reported skills they list. Therefore, recruiters are likely to pay attention to self-reported skills even when they can observe other indications of the candidates' coding abilities. Finally, some recruiters believe that recruiting based on candidates' skills is more "objective" than recruiting on the basis of the schools candidates attended.

5.3 Gender Differences in Recruiters' Response to Profile Attributes

Recruiters on this platform saw only coarse indications of candidates' experience with programming languages: if a candidate was verified with high or low experience in a language and whether or not candidates self-reported knowing a language. Using more nuanced measures of experience, the analysis in the previous section showed that female candidates who self-report a language have on average more programming experience than male candidates with similar profiles. Therefore, we might expect that a recruiter looking at two equally qualified male and female candidates, who both self-report knowing a programming language, would be more inclined towards the female candidate. In this section, I assess if recruiters actions are consistent with favoring female candidates who self-report knowledge of a programming language relative to male candidates with similar profiles.

5.3.1 Overall Recruiter Response

Table 5 shows the results of estimating the hedonic regression, Equation 2. Column (2) displays the coefficients when self-reported skills are interacted with the gender of the candidate. This model represents a situation in which recruiters believe that the information they gain from verified languages is similar for male and female candidates, but the response to self-reported skills should be adjusted because of the gender gap in self-reporting. The estimated coefficients on most of the attributes of candidates remain unchanged from the specification that did not include interaction terms (shown in Column (1)). The interactions between self-reported skills and the gender of the candidate are also mostly noisy and not significantly different from zero. While these regression results indicate that recruiters do not adjust their response to self-reported skills to favor female candidates, there is still the possibility that recruiters could be leaning towards female candidates based on other profile attributes.

In Column (3) of Table 5, I show the results of adding interactions between the gender of the candidate and both the self-reported and verified programming languages to Equation 2. The coefficients on these interactions are insignificant for almost all languages. This is shown graphically in Figure 2. The plots in that figure display the average predicted probability that a candidate is saved given the representation of a language on the candidate's profile. The predicted rates of being saved are similar both genders for the majority of languages and configurations. Therefore, recruiters do not appear to be adjusting their responses to self-reported languages to favor female candidates. While not statistically significant, for the languages Ruby, C#, and Java, the rate that verified-high and self-reporting female candidates will be saved is predicted to be slightly higher than for male candidates. Given that female candidates who are verified with low experience who also self-report are predicted to be saved at lower rates than male candidates, this does not support the theory that recruiters are adjusting to the gender gap in self-reporting languages. Instead, the available evidence is consistent with recruiters showing particular interest in female candidates with high experience in these languages. Finally, I compare the predicted probability that profiles in my dataset would be saved if they were all male versus if they were all female. This comparison reveals that the average predicted probability of being saved for female candidates when using this flexible specification is 6.47% lower than for male candidates.[27] This shows that not only are recruiters not favoring female candidates who self-report, but overall they do not appear to be saving female candidates at higher rates.

The above results provide evidence that recruiters are not systematically adjusting their screening process to favor female candidates who self-report technical skills over male candidates with equivalent displayed information.

5.3.2 Experienced Recruiter Response

Why might recruiters not use gender to adjust their screening process and find the most experienced coders? One possible reason is that even recruiters who are striving to make their workforce more gender balanced may not be aware of the gender gap in the self-reporting of technical skills. Experienced recruiters, who would have had more opportunities to compare candidates' self-reported information with subsequent in-person interviews, could be more likely to update response to self-reported skills (**???**). Therefore, I test if more experienced recruiters are more inclined toward female candidates who self-report knowledge of technical skills.

In Column (4) of Table 5, I estimate the hedonic regression using the candidates from the Profiles dataset with the dependent variable equal to one when a recruiter with at least 12 months on this recruiting platform saved the candidate. Among the interaction terms between the gender of the candidate and self-reported skills, only the coefficients on self-reporting C++ with low or no verified experience are positive; the majority of the interaction terms of simply not significantly different from zero and noisy. These results imply that recruiters who have more experience using this recruiting platform do not show a consistent preference for female coders who self-report knowledge of a programming language relative to similar male coders.

[27] This is significant at the 1% level with an F-stat of 12.56.

5.3.3 Entry Level Candidates

When considering an experienced candidate, recruiters have more signals of the candidate's abilities and interests, such as the candidate's work history. Audit study evidence suggests that gender differences in employers' response to similar resumes of experienced candidates tend to be small or insignificant (?). In contrast, when considering entry level candidates, recruiters typically have less information to draw upon and make more assumptions based on observables, including gender (?). Thus, recruiters reviewing entry level candidates may rely more heavily on candidate's self-report when deciding whom to save. Since the self-reported skills will be potentially more salient for the recent college graduates (?), I test if recruiters lean towards female candidates when faced with similarly qualified candidates who both self-report knowing a programming language among candidates who graduated in the past five years.

In Column (5) of Table 5, I estimate Equation 2 using only observations of candidates who graduated within the past five years. As with the regressions using all observations, the coefficients on the interaction of the candidate being female and the self-reporting of skills and languages are largely insignificant. The only significant coefficient, the interaction of self-reporting JavaScript and being verified with high experience, is negative. Again, despite the potentially more salient aspect of self-reported skills for these candidates, the regression results reiterate that recruiters do not appear to be systematically leaning towards female candidates who self-report knowing technical skills.

5.3.4 Robustness of Findings using Matched Profiles

One concern with the hedonic regression analysis above is that recruiters might actually be responding to some other attribute of candidate profiles that is merely correlated with the self-reporting of programming languages. Ideally, we would like to see how recruiters respond when reviewing two resumes with the exact same attributes except one self-reports a language while the other does not. While I am unable to conduct such an audit study experiment on this platform, I leverage a subset of the Profiles data by finding groups of very similar profiles with variation in whether or not the candidate self-reported programming languages. By comparing the rate at which profiles that differ only in their self-reporting of programming languages get saved, I isolate the influence of self-reporting on recruiter responses.[28]

The results of this exercise are in displayed in Column (1) of Table 6. Specifically, the table shows the estimated coefficients of Equation 3, a linear probability model with fixed effects for matched profile groups. For all but one language, self-reporting that language predicts a higher level of recruiter interest. For the language JavaScript, candidates who self-report that language were 5.7 probability points more likely to have been saved than those that did not self-report a language. Relative to the mean rate of being saved, candidates who self-reported this language were saved at a rate 29% higher than those who did not self-report a language.

Column (2) of Table 6 reports the estimated coefficients of Equation 3 along with interaction terms on between the self-reporting of programming languages and the gender of the candidate. The interaction terms are negative and significant for the languages JavaScript and Python. The interaction terms on PHP, Ruby, C#, and C++ are not significantly different from zero. Only the interaction with Java is positive and significant at the 5% level. These estimates reveal that for the majority of programming language skills, recruiters do not on average appear to be more inclined toward female candidates who self-report knowledge of programming languages over male candidates.

These results are very similar to the results found using the hedonic regression.

5.3.5 Attributes Beyond Self-Reported Languages

Ultimately, recruiters are not finding and recruiting all of the qualified female candidates on this platform. One possible reason is that recruiters may be concerned with factors other than a candidate's coding prowess. For example, recruiters may be concerned about possible gender differences in the attrition or probability that a candidate will accept a recruiting invitation. In Figure 3, I plot the probability that a recruiter saved a candidate versus the number of lines of code that a candidate contributed to the open source projects of others, a measure that is correlated with the candidate's reputation in the open source community. While recruiters are more likely to save candidates with strong reputations, female coders are less likely to be saved than male coders. This indicates that recruiters are likely taking into account factors other than the coding experience or coding reputations of candidates.

[28] In Appendix 17, I also perform this exercise using propensity score matching.

6. Conclusion

The tech workforce remains highly gender imbalanced despite considerable efforts by companies to increase diversity and inclusion. In the labor market for software engineers, many companies regularly contact and recruit candidates regarding job openings. Much of this recruiting takes place using online recruiting platforms where individuals can self-report their skills and recruiters can message qualified individuals. I analyzed the behavior of individuals and recruiters on one such recruiting platform. I focused on quantitatively one of the most important predictors for which candidates are recruited: the self-reporting of technical skills.

My analysis revealed a gender difference in the propensity to self-report known programming languages on the labor supply side that recruiters on the labor demand side do not appear to adjust for in their decisions. In particular, female coders who contribute to open source software projects in a programming language are on average 11.07% less likely to self-report knowing that same language. While one might have expected that recruiters would therefore be more likely to save female candidates who self-report knowing a language than their male counterparts with similar profiles—anticipating that on average the female candidates would be more experienced in that language than the male candidates—this is not the case that recruiters do this form of "belief-flipping." Instead, female candidates are predicted to receive similar benefits from the self-reporting programming languages and on average receive 6.46% less recruiter attention overall after controlling for the candidate attributes shown to recruiters.

Depending on the motivation for the supply-side difference in the propensity to self-report, these findings could have very different implications. If female candidates do not self-report because of a "confidence gap," the lack of adjustment by recruiters for the difference in self-reporting behavior means that employers are likely contacting some male candidates based on self-reported coding skills while overlooking some female candidates with similar actual previous coding experience. If, however, the gender difference in propensity to self-report is primarily due to average differences in preferences for occupations involving coding, the behaviors of both the labor demand and supply sides may be efficient. While my data does not allow me to decompose the precise motivation for the difference in the propensity to self-report, the results of this study provide empirical evidence of gender differences in an important part of the recruiting and hiring pipeline.

These results suggest that employers need to think carefully about the implementation of gender-blind resume reviews as well as algorithmic and machine learning based recruiting systems.[29] While gender-blind resume review can eliminate labor demand biases and potential discrimination, careful consideration must be taken for gender differences in labor supply side behavior (**?**).

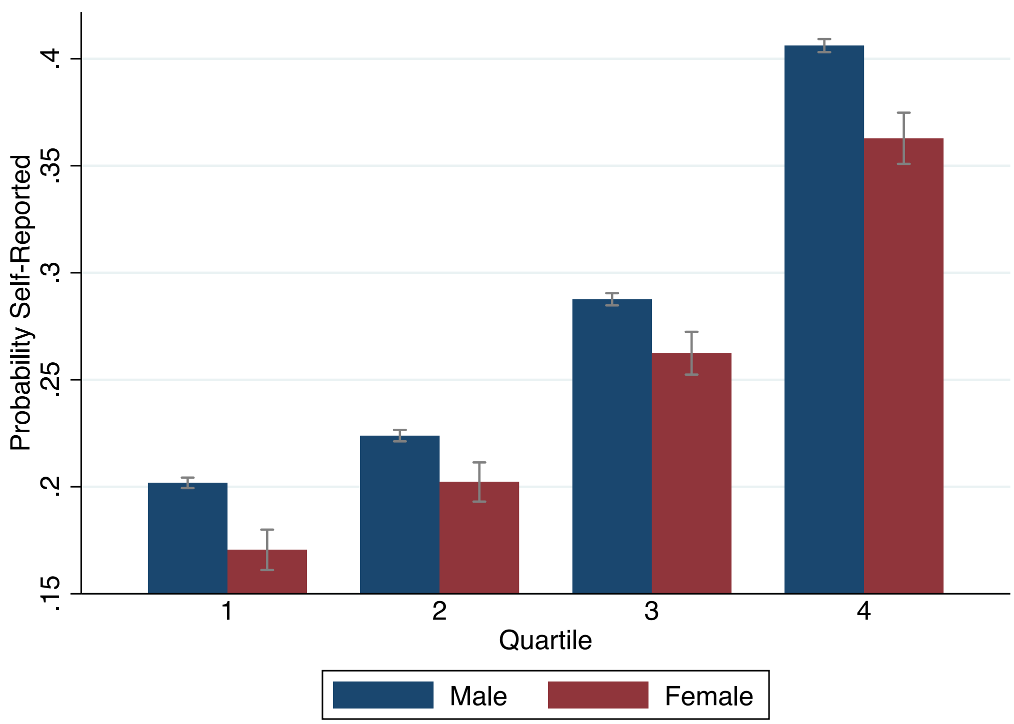
The finding that female candidates are less inclined towards self-reporting knowledge of technical skills on their resumes may have parallels in other industries. If, as suggested by others, the self-reporting of skills is influenced by cultural constructs about gender and occupations, there may be industries in which the gender gap in self-reporting is smaller, larger, or even reversed (**?**). In order to examine self-reporting in other occupations and industries, researchers should look for observational data documenting actual abilities in skills that are comparable across candidates and with self-reported skills.

Finally, while this paper has illuminated factors in the recruiting of candidates that could lead to gender imbalances in the hiring pipeline, in order to change the gender balance in the tech workforce, systematic investigations of interviews and job offer decisions will be needed.

[29]The potential issues that arise can be seen in Amazon's experience building an algorithmic recruiting pipeline (https://slate.com/business/2018/10/amazon-artificial-intelligence-hiring-discrimination-women.html).

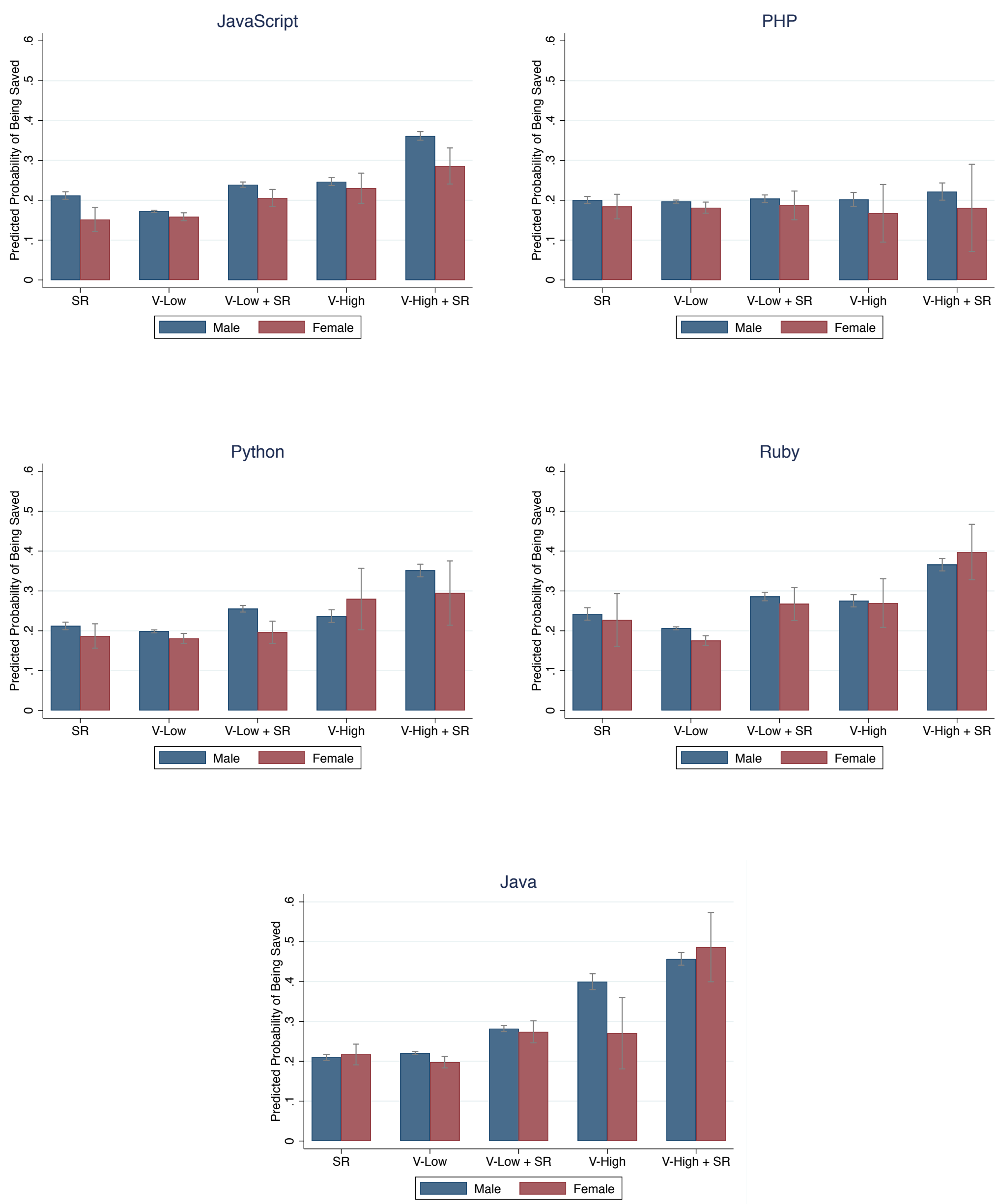
7. Tables and Figures

Figure 1: Probability a Programming Language is Self-Reported on Resume



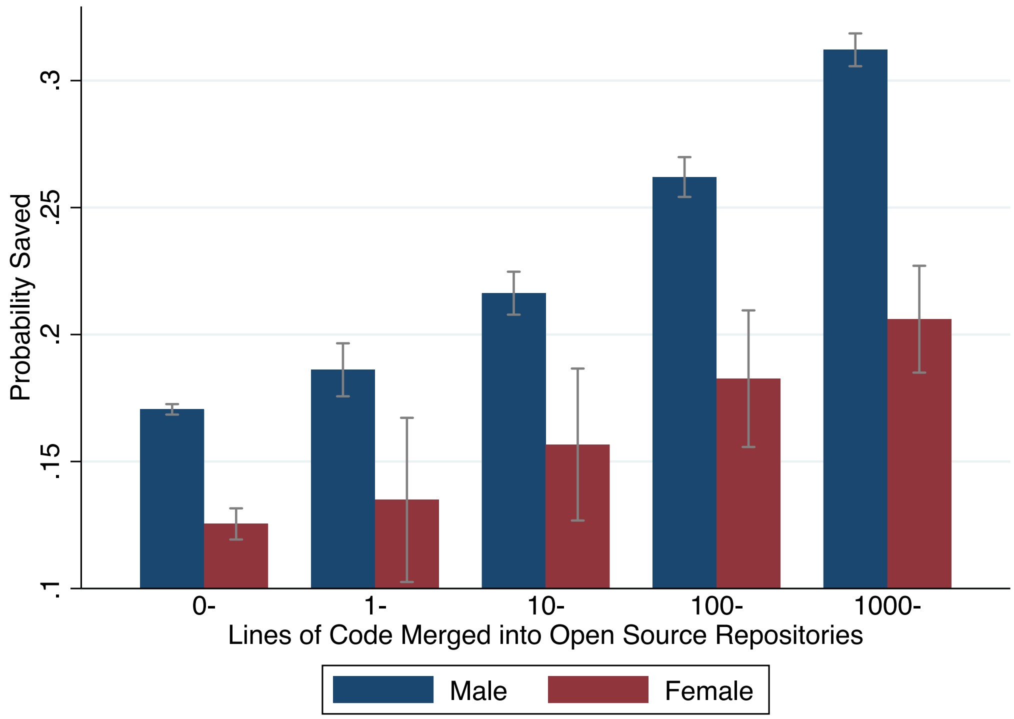
Candidate-language pairs are grouped into quartiles according to the total lines of code in programming language they have uploaded to open source over their lifetime. The quartiles are defined by programming language. The mean probability that male and female coders within each of these quartiles list the programming language among the candidates' self-reported skills are plotted. The 95% confidence interval on the means are also plotted.

Figure 2: Average Predicted Probability Candidates are Saved Conditional on the Appearance of Programming Languages on Profiles



Note. The above plots show the average predicted probability that a candidate is saved based on estimates of Equation 2. The equation includes interactions between the gender of the candidate and the way languages appeared on the candidate profiles.

Figure 3: Probability a Profile is Saved versus Lines of Open Source Code Accepted by Other Coders



Candidates from the Profiles dataset are grouped according to the total number of lines code that the candidate contributed to and had accepted into others' open source projects. The vertical axis shows the probability that a candidate was saved.

Table 1: Components of Candidate Profiles (recruiter's view)

Component	Example	Origin	Searchable by Recruiters
Name	Jane Doe	Written by job seeker	Searchable
Geographic Location	San Francisco, CA	Written by job seeker	Searchable
Overall Score	5	Added by platform	Searchable
Educational Credentials	BA in Computer Science, University of Texas at Austin 2005	Written by job seeker	Searchable
Employment History	Google, Software Engineer, 2005-2008	Written by job seeker	Searchable
Verified Programming Languages	Python (High)	Added by platform	Searchable
Self-Reported Skills	Python, JavaScript, Public Speaking, Writing	Written by job seeker	Not searchable

Table 2: Mean Value of Attributes on Profiles

	All		Matched Profiles	
	Males	Females	Males	Females
Saved by at least one recruiter	0.21	0.15	0.16	0.12
Number of times saved by recruiters	0.39	0.23	0.30	0.18
Bachelors Year	2005.93	2007.74	2007.96	2008.79
Bachelors in CS	0.29	0.27	0.14	0.15
Masters Degree	0.16	0.23	0.11	0.18
Ph.D. Degree	0.04	0.05	0.03	0.04
Currently Coder	0.47	0.40	0.38	0.33
Employed at Top 10 Tech Company	0.02	0.03	0.01	0.01
Number of SR Skills	11.26	10.18	9.59	8.04
Number of SR Languages	1.08	0.90	0.87	0.70
Number of Verified Languages	2.47	2.14	2.13	1.91
Lists Bachelors Degree	0.50	0.56	0.24	0.33
Lists SR Skill	0.43	0.42	0.37	0.36
SR Programming	0.40	0.38	0.34	0.33
SR JavaScript	0.25	0.22	0.21	0.18
SR PHP	0.13	0.10	0.12	0.08
SR Ruby	0.09	0.07	0.08	0.06
SR Python	0.16	0.15	0.13	0.11
SR C#	0.10	0.06	0.07	0.04
SR C++	0.12	0.10	0.09	0.08
SR Java	0.21	0.19	0.17	0.15
N	158,230	12,656	78,570	6,820

The mean values of attributes on the profiles of male and female candidates in the Profiles dataset are shown in the first two columns. The third and fourth column shows the mean attributes for male and female candidates in the Match Profiles dataset. The above variables are described in detail in Appendix 8

Table 3: Predicted Probability a Programmer Lists a Language as a Self-Reported Skill

	Self-Reported							
	(1) All	(2) All	(3) All	(4) JavaScript	(5) Ruby	(6) Python	(7) C++	(8) Java
Female	-0.031***	-0.022***	-0.021***	-0.022***	-0.036***	-0.018***	-0.030***	-0.016**
	(0.004)	(0.004)	(0.004)	(0.005)	(0.006)	(0.007)	(0.009)	(0.007)
Ln(Code)		0.009***	0.008***	0.008***	0.015***	0.011***	-0.002***	0.005***
		(0.000)	(0.000)	(0.000)	(0.001)	(0.001)	(0.001)	(0.001)
Ln(Days with Uploads)		0.034***	0.032***	0.033***	0.036***	0.031***	0.030***	0.031***
		(0.000)	(0.000)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)
Ln(Accepted OSS Code)		0.002***	0.001**	-0.001	0.007***	-0.001	-0.002**	0.003***
		(0.000)	(0.000)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)
Bachelors in CS			0.048***	0.044***	0.020***	0.042***	0.077***	0.095***
			(0.003)	(0.004)	(0.004)	(0.005)	(0.006)	(0.006)
Current Coder			0.147***	0.186***	0.119***	0.134***	0.101***	0.152***
			(0.002)	(0.003)	(0.003)	(0.004)	(0.004)	(0.004)
Constant	0.280***	0.178***	-0.051**	-0.044	-0.064	-0.025	-0.010	0.016
	(0.001)	(0.001)	(0.022)	(0.033)	(0.040)	(0.046)	(0.047)	(0.058)
Geo. FE	No	No	Yes	Yes	Yes	Yes	Yes	Yes
BA Year FE	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Schools FE	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Employers FE	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Other Verified Langs. FE	No	No	Yes	No	No	No	No	No
N	418,720	418,720	418,720	115,328	61,516	64,090	38,153	59,973
N Candidates	170,886	170,886	170,886	115,328	61,516	64,090	38,153	59,973
Dep. Mean	0.28	0.28	0.28	0.31	0.20	0.29	0.23	0.35
Adj. R^2	0.00	0.05	0.12	0.14	0.14	0.13	0.09	0.12

Standard errors clustered at the profile level in parentheses

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

The above table shows the estimates of Equation 1 using the candidate-language pair observations. Each column adds additional controls. Female is an indicator for the candidate being female. Code is the number of lines of code that the candidate has written and uploaded in a programming language. Days with Uploads are the number of days in which the candidate uploaded open source code. Accepted OSS Code are the number of lines of code that the candidate has uploaded and had accepted into others' open source projects. Bachelors in CS is an indicator for a bachelor's degree in Computer Science being present on the resume. Current coder is that the listed current occupation title is associated with being a programmer. Geographic Fixed Effects are indicators for the state and region listed on the resume. Employers Fixed Effects are indicators for the names of all current and previous employers. Other Verified Languages are indicators for all the languages that an individual has verified on their resume.

Table 4: Auxiliary Results Related to Self-Reported Programming Language Skills

	Self-Reported			
	(1) All	(2) All	(3) Coding 2016-17	(4) 1+ Skills
Female	-0.025***	-0.019***	-0.012**	-0.015***
	(0.005)	(0.004)	(0.006)	(0.005)
Ln(Code)	0.008***	0.008***	0.011***	0.017***
	(0.000)	(0.000)	(0.000)	(0.000)
Ln(Days with Uploads)	0.032***	0.031***	0.028***	0.032***
	(0.000)	(0.000)	(0.001)	(0.001)
Ln(Accepted OSS Code)	0.001**	0.000	0.005***	0.002***
	(0.000)	(0.000)	(0.000)	(0.001)
Ln(Code) x Female	0.001			
	(0.001)			
Ln(Days with Uploads) x Female	-0.003			
	(0.002)			
Ln(Accepted OSS Code) x Female	-0.003*			
	(0.002)			
Bachelors in CS	0.057***	0.053***	0.058***	0.081***
	(0.003)	(0.003)	(0.004)	(0.003)
Current Coder	0.149***	0.141***	0.148***	0.087***
	(0.002)	(0.002)	(0.003)	(0.003)
Saves, Before Dec. 2015		0.021***		
		(0.002)		
Constant	-0.030	-0.026	-0.052	0.264***
	(0.022)	(0.022)	(0.039)	(0.045)
Geo. FE	Yes	Yes	Yes	Yes
BA Year FE	Yes	Yes	Yes	Yes
Schools FE	Yes	Yes	Yes	Yes
Employers FE	Yes	Yes	Yes	Yes
N	418,720	418,720	195,285	205,289
N Candidates	170,886	170,886	74,657	73,078
Dependent Mean	0.28	0.28	0.27	0.57
Adj. R^2	0.11	0.12	0.11	0.08

Standard errors clustered at the profile level in parentheses

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

The above table shows the estimates of Equation 1. The first column includes both measures of previous

Table 5: Linear Probability Model Predicting if a Profile is Saved

	Saved			Saved, Experienced Recruiters	Sav
	(1)	(2)	(3)	(4)	
Female	-0.01***	-0.01***	-0.00	-0.00	
	(0.00)	(0.00)	(0.01)	(0.00)	
Bachelors in CS	0.04***	0.04***	0.04***	0.01***	
	(0.00)	(0.00)	(0.00)	(0.00)	
SR NodeJS	0.08***	0.08***	0.08***	0.10***	
	(0.03)	(0.03)	(0.03)	(0.03)	
Female x SR NodeJS		0.01	0.01	0.03	
		(0.15)	(0.15)	(0.15)	
SR MongoDB	0.05***	0.05***	0.05***	0.03***	
	(0.01)	(0.01)	(0.01)	(0.01)	
Female x SR MongoDB		0.00	0.00	-0.01	
		(0.03)	(0.03)	(0.02)	
SR Customer Service	-0.01	-0.01	-0.01	-0.00	
	(0.01)	(0.01)	(0.01)	(0.01)	
Female x SR Customer Service		0.01	0.00	-0.02	
		(0.02)	(0.02)	(0.01)	
JavaScript SR, No V	0.02***	0.02***	0.02***	0.01	
	(0.01)	(0.01)	(0.01)	(0.00)	
Female x JavaScript SR, No V		-0.06***	-0.06***	-0.02	
		(0.02)	(0.02)	(0.02)	
JavaScript V-Lo, No SR	0.02***	0.02***	0.02***	0.01***	
	(0.00)	(0.00)	(0.00)	(0.00)	
Female x JavaScript V-Lo, No SR			-0.00	0.01	
			(0.01)	(0.00)	
JavaScript SR and V-Lo	0.05***	0.05***	0.05***	0.02***	
	(0.00)	(0.00)	(0.00)	(0.00)	
Female x JavaScript SR and V-Lo		-0.01	-0.02	-0.01	
		(0.01)	(0.01)	(0.01)	
JavaScript V-Hi, No SR	0.08***	0.08***	0.08***	0.07***	
	(0.01)	(0.01)	(0.01)	(0.00)	
Female x JavaScript V-Hi, No SR			-0.00	0.00	
			(0.02)	(0.02)	
JavaScript SR and V-Hi	0.16***	0.16***	0.16***	0.14***	
	(0.01)	(0.01)	(0.01)	(0.01)	
Female x JavaScript SR and V-Hi		-0.06**	-0.06**	-0.05**	
		(0.02)	(0.03)	(0.02)	
Controls	Yes	Yes	Yes	Yes	

Table 6: Matched Profile Regressions

	Saved	
	(1)	(2)
Female	-0.013***	-0.008*
	(0.004)	(0.004)
SR JavaScript	0.057***	0.060***
	(0.006)	(0.006)
Female × SR JavaScript		-0.026
		(0.019)
SR PHP	0.011*	0.011*
	(0.006)	(0.006)
Female × SR PHP		-0.005
		(0.023)
SR Ruby	0.076***	0.076***
	(0.007)	(0.007)
Female × SR Ruby		-0.004
		(0.028)
SR Python	0.047***	0.049***
	(0.006)	(0.006)
Female × SR Python		-0.033
		(0.022)
SR C#	0.023***	0.025***
	(0.007)	(0.007)
Female × SR C#		-0.026
		(0.030)
SR C++	-0.008	-0.009
	(0.007)	(0.007)
Female × SR C++		0.014
		(0.025)
SR Java	0.037***	0.035***
	(0.006)	(0.006)
Female × SR Java		0.026
		(0.021)
Group Fixed Effect	Yes	Yes
N	85,390	85,390
N Groups	19,549	19,549
Dep. Mean	0.16	0.16
R^2	0.026	0.026

The above table shows the coefficients from estimating Equation 3 using the Match Profiles dataset. The dependent variable is an indicator for if the candidate was saved by a recruiter between March 2014 and November 2016. A fixed effect for each profile group is included in the regression. Standard errors are

8. Data and Description of Variables

The data for this paper comes from a hiring and recruiting platform used by both job seekers and employers. The platform is effectively composed of two different sites. The job seeker site enables those interested in jobs to create a digital resume. While some of the individuals who created a digital resume were actively seeking new jobs, many created a profile simply to be visible to potential employers. The job seeker site is general and many workers who were not software engineers or programmers also created a profile. The recruiter site provides a means for employers to find individuals to contact about potential jobs. This site is particularly geared towards finding engineers and technical workers. A range of employers subscribed to the recruiter site including both large and medium sized firms. The majority of subscribers serve as recruiters for a particular employer.

In the following, I describe each variable on each profile record in the dataset:
Number of Times Saved. The number of times any recruiter who subscribed to the platform pressed a button on the profile indicating that they wished to contact this candidate between March 2014 and November 2016.

Saved. An indicator for if any recruiter pressed a button on the candidate's profile indicating that they wished to contact this candidate between March 2014 and November 2016.

Lists Bachelor's Degree. An indicator for if the candidate described completing a Bachelor's degree and information about that degree appeared on their profile.

Bachelor's Year. The year in which the bachelor's degree listed on the profile was completed. If multiple bachelor's degree are listed then year of the first degree is used.

Bachelor's in CS. Whether or not any of the completed bachelor's degrees listed on the profile have majors in Computer Science, Information Science, Information Systems, Information Technology, Artificial Intelligence, Data Processing, Databases, or System Administration. This list of majors comes from the Department of Education's IPEDS database as being part of Computer and Information Sciences.

Bachelor's School. The school attended for which the first listed bachelor's degree was completed. Only schools within the top 25 colleges for Computer Science according to U.S. News & World Report are included. Many schools have a variety of names and acronyms. Therefore, I limit the number of schools that are distinctly identified. The schools are Carnegie Mellon University, Massachusetts Institute of Technology, Stanford University, University of California-Berkeley, University of Illinois-Urbana-Champaign, Cornell University, University of Washington, Princeton University, Georgia Institute of Technology, University of Texas-Austin, California Institute of Technology, University of Wisconsin-Madison, University of California-Los Angeles, University of Michigan-Ann Arbor, Columbia University, University of California-San Diego, University of Maryland-College Park, Harvard University, University of Pennsylvania, Brown University, Purdue

University-West Lafayette, Rice University, University of Southern California, Yale University, and Duke University.

Rank of BA School in CS. The U.S. New & World Report Ranking of the college or university in which the candidate received their bachelor's degree.

Schools Attended. Schools attended for any degree. I include only schools within the top 25 colleges for Computer Science according to U.S. News & World Report. The schools are Carnegie Mellon University, Massachusetts Institute of Technology, Stanford University, University of California-Berkeley, University of Illinois-Urbana-Champaign, Cornell University, University of Washington, Princeton University, Georgia Institute of Technology, University of Texas-Austin, California Institute of Technology, University of Wisconsin-Madison, University of California-Los Angeles, University of Michigan-Ann Arbor, Columbia University, University of California-San Diego, University of Maryland-College Park, Harvard University, University of Pennsylvania, Brown University, Purdue University-West Lafayette, Rice University, University of Southern California, Yale University, and Duke University.

Masters Degree. Indicator for if a master's degree appears on the profile.

Masters Degree Year. Year of first master's Degree.

Ph.D. Degree. Indicator for if a Ph.D. degree appears on the profile.

Ph.D. Year. Year of first Ph.D. degree.

Currently Coder. Indicator for if the profile lists a current job with a title associated with a job involving programming. These include any job title with the words, "software", "sde", "coder", "programmer", "developer", "engineer", or "hacker."

Past Employers. Employers listed in the employment history of the candidate. Only the top 25 employers for tech workers according to Glassdoor's survey in 2015 are included as separate indicators in the regressions.

Months of Work Experience. Number of months since graduated from college.

Months in Current Role. Number of months with current job title.

Average Months at Employer. Average number of months with past employers.

Average Months in Job Title. Average number of months in each job title.

Internship. Employers listed in the employment history of the candidate where the job title included the phrase "intern." Only the top 25 employers for tech workers according to Glassdoor's survey in 2015 are included as separate indicators in the regressions.

Internship at Current Employer. Indicator if the candidate did an internship at their current employer.

Geographic location. Indicators for the state in which candidate currently resides.

"Overall Candidate Scores". The platform estimates two scores for each candidate's predicted relative level of technical skill and potential as an employee. These two scores are between one and five and displayed prominently on their profile. They used a proprietary method for constructing this score that incorporated analysis of the candidate's work history, education, and open source contributions.

Lists SR Skills. An indicator for if the candidate listed at least one self-reported skill on their profile.

SR Programming. An indicator for if the candidate listed "Programming" as a self-reported skill.

SR Software Dev./Engineering. An indicator for if the candidate listed either "Software Development" or "Software Engineering" as a self-reported skill.

SR Programming. An indicator for if the candidate listed "Web Applications" as a self-reported skill.

SR Git/SVN. An indicator for if the candidate listed either "Git" or "SVN" as a self-reported skill.

SR REST. An indicator for if the candidate listed "REST" as a self-reported skill.

SR Web Dev. An indicator for if the candidate listed "Web Development" as a self-reported skill.

SR Agile. An indicator for if the candidate listed either "Agile Methodologies" or "Agile Practices" as a self-reported skill.

SR Project Management. An indicator for if the candidate listed "Project Management" as a self-reported skill.

SR Program Management. An indicator for if the candidate listed "Program Management" as a self-reported skill.

SR Management. An indicator for if the candidate listed "Management" as a self-reported skill.

SR Leadership. An indicator for if the candidate listed "Leadership" as a self-reported skill.

SR Customer Service. An indicator for if the candidate listed "Customer Service" as a self-reported skill.

SR Social Media. An indicator for if the candidate listed "Social Media" as a self-reported skill.

SR Public Speaking. An indicator for if the candidate listed "Public Speaking" as a self-reported skill.

SR Team Building. An indicator for if the candidate listed "Team Building" as a self-reported skill.

SR JavaScript. An indicator for if the candidate listed either "JavaScript" or various JavaScript libraries as a self-reported skill.

Programming Language in Work Descriptions. An indicator for if the candidate wrote the name of a programming language in a description of their employment history. Note that candidate's descriptions of their previous jobs were not shown to recruiters. Only job titles, employer names, and employment dates were shown to recruiters.

Photo of Candidate. If the candidate uploaded a photo with their digital resume.

Verification Level in Programming Languages. The level of verification ascribed by the platform in each programming language.

Note that many of these fields are missing because a candidate did not fill them in on their digital resume. Missing values are treated as a distinct value since recruiters would see a blank field when information was missing.

In the following table, I show additional summary statistics for the profile attributes.

Table 7: Means of Profiles Attributes

	All		Matched Profiles	
	Males	Females	Males	Females
Saved	0.21	0.15	0.16	0.12
Saves	0.39	0.23	0.30	0.18
Lives in SF Bay Area	0.16	0.22	0.20	0.25
Bachelors Year	2005.93	2007.74	2007.96	2008.79
Bachelors in CS	0.29	0.27	0.14	0.15
Bachelor's Ranked Top 10	0.06	0.09	0.01	0.03
Masters Degree	0.16	0.23	0.11	0.18
Ph.D. Degree	0.04	0.05	0.03	0.04
Currently Coder	0.47	0.40	0.38	0.33
Employed at Top 10 Tech Company	0.02	0.03	0.01	0.01
Number of SR Skills	11.26	10.18	9.59	8.04
Number of SR Languages	1.08	0.90	0.87	0.70
Number of Verified Languages	2.47	2.14	2.13	1.91
Lists Bachelors Degree	0.50	0.56	0.24	0.33
Lists SR Skill	0.43	0.42	0.37	0.36
Coder 0-5 Years After BA	0.31	0.29	0.16	0.18
Coder 5-10 Years After BA	0.16	0.13	0.07	0.08
Coder 10-15 Years After BA	0.07	0.05	0.02	0.02
Employers 0-5 Years	1.10	1.35	0.57	0.85
Employers 0-10 Years	1.48	1.72	0.72	1.05
Years in Current Job	1.10	0.73	0.88	0.59
Verified JavaScript	0.67	0.68	0.65	0.67
Verified PHP	0.29	0.24	0.25	0.20
Verified Ruby	0.36	0.34	0.34	0.33
Verified Python	0.38	0.33	0.31	0.27
Verified C#	0.19	0.09	0.13	0.07
Verified C++	0.23	0.15	0.16	0.10
Verified Java	0.35	0.31	0.30	0.28
SR Programming	0.40	0.38	0.34	0.33
SR JavaScript	0.25	0.22	0.21	0.18
SR PHP	0.13	0.10	0.12	0.08
SR Ruby	0.09	0.07	0.08	0.06
SR Python	0.16	0.15	0.13	0.11
SR C#	0.10	0.06	0.07	0.04
SR C++	0.12	0.10	0.09	0.08
SR Java	0.21	0.19	0.17	0.15
GitHub Followers	0.80	0.36	0.69	0.34
Months Since Last GitHub Action	9.92	14.92	10.83	16.15
N	158,230	12,656	78,570	6,820

The mean values of attributes on the profiles of male and female candidates in the Profiles dataset are shown in the first two columns. The third and fourth column shows the mean attributes for male and female candidates in the Match Profiles dataset.

The following tables show tabulations of how one example language, JavaScript, appeared on the profiles of candidates:

Table 8: Appearance of JavaScript on Profiles of Male Candidates

	Verified - High	Verified - Low	Not Verified
Self-Reported	0.07	0.14	0.04
Not Self-Reported	0.05	0.41	0.28

Each cell shows the fraction of all male profiles that have the programming language JavaScript listed in either the “Self-Reported Skills” list, the “Verified Languages” list, or both lists. The last line of the table shows the fraction of individuals who are “Verified” at a level of experience who also self-report the language. The left cell shows the fraction of “Verified - High” experience individuals who also self-report knowing JavaScript, while the right cell shows the fraction of “Verified - Low” experience candidates who self-report JavaScript.

Table 9: Appearance of JavaScript on Profiles of Female Candidates

	Verified - High	Verified - Low	Not Verified
Self-Reported	0.04	0.14	0.04
Not Self-Reported	0.04	0.46	0.28

Each cell shows the fraction of all female profiles that have the programming language JavaScript listed in either the “Self-Reported Skills” list, the “Verified Languages” list, or both lists. The last line of the table shows the fraction of individuals who are “Verified” at a level of experience who also self-report the language. The left cell shows the fraction of “Verified - High” experience individuals who also self-report knowing JavaScript, while the right cell shows the fraction of “Verified - Low” experience candidates who self-report JavaScript.

Below, I show summary statistics regarding the measures of experience from the candidate-language pairs dataset.

Table 10: Mean Value of Candidate Language Pairs

	All	
	Males	Females
Code	13,941	12,267
Years of Coding	1.2	.65
Days with Uploads	122	51
OSS Watchers	3.1	1.6
StackOverflow Answers	1.9	.35
Accepted OSS Code	2,065	1,673
Number of OSS Repos. Contributed to	.33	.18
Months Since First OSS Code in Lang	9.7	5.3
N	391,582	27,138

The mean values of the measures of experience and reputation are shown for candidate-language pair observations in the above table.

9. Self-Reporting Non-Technical Skills

The below table shows the results of estimating Equation 1 using observations from the Profiles dataset. The dependent variable in these regressions is a non-technical skill. I control for the languages that the candidate is verified in as well as the candidate's occupation and educational background.

The positive coefficients on the indicator for the candidate being female reveals that on average female candidates are more likely to self-report many of these non-technical skills. For example, female candidates are significantly more likely to self-report "Project Management," "Customer Service," "Public Speaking," and "Social Media" as skills than male counterparts.

Table 11: Predicting if a Programmer Lists a Non-Technical Skill

	Self-Reported				
	(1) Project Management	(2) Customer Service	(3) Leadership	(4) Management	(Public
Female	0.004**	0.010***	0.002	-0.002*	0.0
	(0.002)	(0.001)	(0.001)	(0.001)	(0.
BA in CS	-0.015***	-0.013***	-0.013***	-0.013***	-0.0
	(0.001)	(0.001)	(0.001)	(0.001)	(0.
Current Coder	-0.002**	-0.000	-0.006***	-0.008***	-0.0
	(0.001)	(0.001)	(0.001)	(0.001)	(0.
Constant	-0.003	0.002	-0.004	-0.008***	0.
	(0.009)	(0.006)	(0.007)	(0.001)	(0.
Geo. FE	Yes	Yes	Yes	Yes	Y
BA Year FE	Yes	Yes	Yes	Yes	Y
Schools FE	Yes	Yes	Yes	Yes	Y
Employers FE	Yes	Yes	Yes	Yes	Y
Verified Langs. FE	Yes	Yes	Yes	Yes	Y
N	170,886	170,886	170,886	170,886	170
Dependent Mean	0.03	0.01	0.02	0.02	0
R2	0.01	0.01	0.01	0.01	0

Standard errors in parentheses

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

The above table shows the estimates of Equation 1 using the candidate-language pair observations. The dependent variable is an indicator for if the candidate self-reports a non-technical skill. Each column has a different non-technical skill as the dependent variable. Female is an indicator for the candidate being female. Code is the number of lines of code that the candidate has written and uploaded in a programming language. Days with Uploads are the number of days in which the candidate uploaded open source code. Accepted OSS Code are the number of lines of code that the candidate has uploaded and had accepted into others' open source projects. Bachelors in CS is an indicator for a bachelor's degree in Computer Science being present on the resume. Current coder is that the listed current occupation title is associated with being a programmer. Geographic Fixed Effects are indicators for the state and region listed on the resume. Employers Fixed Effects are indicators for the names of all current and previous employers.

One caveat with the above table is that, unlike in the analysis of technical skills, I do not have measures of experience in these non-technical skills. Therefore, I am unable to say if there is a gap in self-reporting relative to the actual abilities of the male and female candidates in these skills.

10. Alternative Measures of Coding Skills

In the following table, I show the results of estimating Equation 1 using seven different measures of a candidate's previous coding experience in a programming language. The specific measures used are as follows:

Log(Code): Number of lines of code in a language uploaded to open source repositories. Lines that are copied or "forked" from other open source repositories are removed.

Ln(Years of Coding): Count of the number of days in which a candidate uploaded open source code to open source repositories. Converted into years for readability.

Ln(OSS Watchers): Count of the number of "watchers" on GitHub of the open source code repositories created by a candidate in a programming language.

Ln(StackOverflow Answers): Number of StackOverFlow Answers tagged with a programming language contributed by a candidate.

Ln(Code Merged in OSS): Number of lines of code uploaded to other's open source repositories on GitHub that were approved and "merged" into a repository.

Ln(Number of OSS Repos. Contributed to): The number of other's open source repositories that accepted and "merged" code contributions from the candidate.

Ln(Months Since First OSS Code in Lang): The number of months since the first time that the candidate upload code to a GitHub repository in that language.

While each of these measures captures a slightly different aspect of the coding experience of a candidate, the measures can broadly be categorized as relating to the quantity of coding work done, the amount of time spent working in that language, and the quality or reputation of the work in that language. The measures representing quantity of coding are Log(Code). The measures representing time are Ln(Years of Coding) and Ln(Months Since First OSS Code in Lang). The measures that capture reputation are coding work that required others review and approval of the code Ln(OSS Watchers), Ln(Code Merged in OSS), and Ln(Number of OSS Repos. Contributed to).

An alternative way of measuring an individual's reputation in the open source world are the number of "watchers" of that candidate's open source projects and the number of "followers" of the candidate. Watchers are other coders who subscribe to updates about the project, and followers are those who subscribe to updates about any code contributions made by an individual to any open source projects. The theory behind using the number of watchers and followers is that they are correlated with the number of users of the code and predict the number of contributors to open source projects in the future (Crowston, Hosion, and Annabi, 2006; Sheoran, Blincoe, Kalliamvakou, Damian, and Ell, 2014). Many empirical works, therefore, have used followers and watchers to proxy for reputations of individuals and projects (Sen, Singh, and Borle, 2012; Cai and Zhu, 2016; Joy, Thangavelu, and Jyotishi, 2018). In the Appendix of this paper, I replicate my results using the number of watchers and the number of followers as proxies for reputation. All of the results show robustness to using these different measures of experience and reputation alone or in combination.

Across all of these measures of previous experience in a language, I find that female candidates are on average less likely to self-report a programming language. These results are displayed below:

Table 12: Self-Reporting Regressions Using Alternative Measures of Experience

	Self-Reported							
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Female	-0.031***	-0.035***	-0.017***	-0.029***	-0.018***	-0.029***	-0.027***	-0.026*
	(0.004)	(0.004)	(0.004)	(0.004)	(0.004)	(0.004)	(0.004)	(0.004
Ln(Code)		0.022***						
		(0.000)						
Ln(Years of Coding)			0.219***					
			(0.002)					
Ln(OSS Watchers)				0.049***				
				(0.002)				
Ln(StackOverflow Answers)					0.078***			
					(0.001)			
Ln(Code Merged in OSS)						0.012***		
						(0.000)		
Ln(Number of OSS Repos. Contributed to)							0.075***	
							(0.002)	
Ln(Months Since First OSS Code in Lang)								0.041**
								(0.002
Constant	0.280***	0.166***	0.250***	0.275***	0.261***	0.272***	0.270***	0.175**
	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.006
N	418,720	418,720	418,720	418,720	418,720	418,720	418,720	418,72
N Candidates	170,886	170,886	170,886	170,886	170,886	170,886	170,886	170,88
Dependent Mean	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
Adj. R^2	0.00	0.03	0.03	0.00	0.01	0.00	0.00	0.00

Standard errors in parentheses

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

The above table shows the estimates of Equation 1 using the candidate-language pair observations. The dependent variable is an indicator for if the candidate self-reports a technical skill. Each column controls for experience in that programming language using different measures of previous coding experience in that language.

11. Number of Times Saved As Outcome

In the main text of the paper, I use whether or not a candidate was saved as an indicator of recruiters interest in a candidate. Those linear probability models make the results directly comparable with the results of previous audit studies (**?**).

An alternative outcome variable would the number of times that a candidate was saved. In the following table, I show that the fundamental results of the paper—that recruiters show more interest in candidates who self-report programming skills and that recruiters do not show more interest in the female self-reporters—are robust to using this alternative outcome.

Table 13: Predicting Number of Times Saved using Matched Profiles

	Saves	
	(1)	(2)
Female		-0.010
		(0.006)
SR JavaScript	0.133***	0.146***
	(0.015)	(0.016)
Female × SR JavaScript		-0.144***
		(0.034)
SR PHP	-0.030*	-0.036*
	(0.018)	(0.019)
Female × SR PHP		0.033
		(0.039)
SR Ruby	0.240***	0.245***
	(0.025)	(0.026)
Female × SR Ruby		-0.117*
		(0.063)
SR Python	0.065***	0.072***
	(0.017)	(0.018)
Female × SR Python		-0.089**
		(0.037)
SR C#	0.002	0.000
	(0.019)	(0.019)
Female × SR C#		-0.018
		(0.045)
SR C++	-0.050**	-0.053**
	(0.021)	(0.022)
Female × SR C++		0.036
		(0.042)
SR Java	0.141***	0.139***
	(0.014)	(0.014)
Female × SR Java		0.018
		(0.037)
Group Fixed Effect	Yes	Yes
N	170,886	170,886
N Groups	59,576	59,576
[illegible]	[illegible]	[illegible]

12. Labor Market Correlates of a Candidate Being Saved

Recruiters save candidate that they are interested in contacting about job opportunities or in order to invite a candidate to interview for an open position. Due to data restrictions, much of the subsequent communications between candidates and potential employers is not available for analysis.

I show that a candidate being saved is correlated with a candidate subsequently changing jobs. In order to do this, I regress if a candidate switched jobs between June 2015 and December 2015 on an indicator for if a recruiter saved the candidate between January 2015 and June 2015 as well as a set of controls for the attributes of the candidate. To avoid having the estimates influenced by internships and very short job changes, this regression is estimated using only candidates from the Profile dataset that have graduated from college. The estimates are displayed below in Table 6.

Table 14: Labor Market Correlates of a Candidate Being Saved

	Moved Jobs, 06/2015 - 12/2015		
	(1)	(2)	(3)
Saved, 03/2014-06/2015	0.004***	0.002**	0.003**
	(0.001)	(0.001)	(0.001)
Education Controls	Yes	Yes	
Work Controls	Yes	Yes	
Verified and SR Skills Controls	No	Yes	
Profile Group FE			Yes
N	149,219	149,219	149,219
Dep. Mean	0.006	0.006	0.006
Adj. R^2	0.00	0.02	0.00

Standard errors in parentheses

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

The above table shows a subset of the coefficients from an OLS regression on the Profile dataset. The dependent variable is whether or not the candidate changed employers between June 2015 and December 2015. The main independent variable is whether or not the candidate was saved between March 2014 and June 2015. Controls are also included for the educational background, work history, and verified and self-reported skills of the candidate. In column (3), the regression is conducted using the Match Profile dataset and includes a fixed effect for the match profile group. Robust standard errors are used.

In the first column, I estimate the regression with only controls for the educational credentials and work history of the candidate. The coefficient on the candidate having been saved is 0.004 and significant. Candidates who are saved between January 2015 and June 2015 are 0.4 probability points more likely to switch jobs in the subsequent six months than those who were not saved. Once additional controls are added for the self-reported and verified skills of the candidate, the estimated coefficient shrinks to 0.2 probability points. Relative to the baseline rate of switching jobs of 0.006, this implies that those who were saved were 33.33% more likely to switch jobs in the subsequent six months.

This evidence is correlational and should not be taken as causal. In particular, reverse causation is possible: recruiters might contact candidates who are more likely to switch jobs based on elements that are not observable in my data. In the third column of Table 6, I match candidate profiles on the basis of the information that recruiters could search for candidates on this platform. I then compare candidates who were saved versus those that were not. Again, the estimated coefficient on a candidate being saved implies a close to 50% higher rate of changing jobs. This is suggestive evidence that saving a candidate may predict subsequent law market moves.

13. Timing of Self-Reporting and Potential for Mismeasurement

There are two possible ways that mismeasurement could occur because of timing and the way in which the data for this paper was collected. First, a candidate could have added a self-reported skill to their profile after the profiles data was collected but before a recruiter saved the candidate's profile. This is because the data on which candidates self-report was collected in December 2015, but the data on which candidates are saved extended until November 2016. In this case, the profile will appear in my data as a saved profile that did not self-report when in fact it should be counted as a profile that self-reported. The estimated rate that profiles that did not self-report were saved will be higher. Given that on average the candidates who self-reported got saved at a higher rate, this will shrink the differences between the save rate for these two sets of candidates.

Second, a candidate could have been saved prior to that candidate self-reporting a programming language on their profile. This is because the data on which candidates were saved starts in March 2014 but the data on which candidates self-report comes from December 2015. In this case, the profile will appear in my data as a saved profile that did self-report when in fact it should be counted as a profile that did not self-report. The estimated rate that profiles that did self-report were saved will be higher, making the difference between the save rate of self-reporters and non-self-reporters larger.

By re-running the recruiter regression with an outcome of whether or not a recruiter saved a candidate between December 2015 and January 2016, I can eliminate the possibility of mismeasurement due to the second scenario described above. In the following table, I show the results of running this regression using the matched profiles dataset. These estimates show positive coefficients on self-reporting, which reiterates that self-reporting does predict more attention from recruiters. The possibility of mismeasurement because of the first scenario described above remains. That form of mismeasurement, however, would tend to attenuate the correlations. Because the estimated coefficients have remained positive and mostly significantly, such mismeasurement seems unlikely to be making a qualitative difference.

While the above robustness checks above show that it is unlikely that mismeasurement is significantly changing results, both forms of mismeasurement occur when candidates delay adding self-reported skills to their profiles. I investigate if there are gender differences in the rate of adding self-reported skills on profiles on this platform.

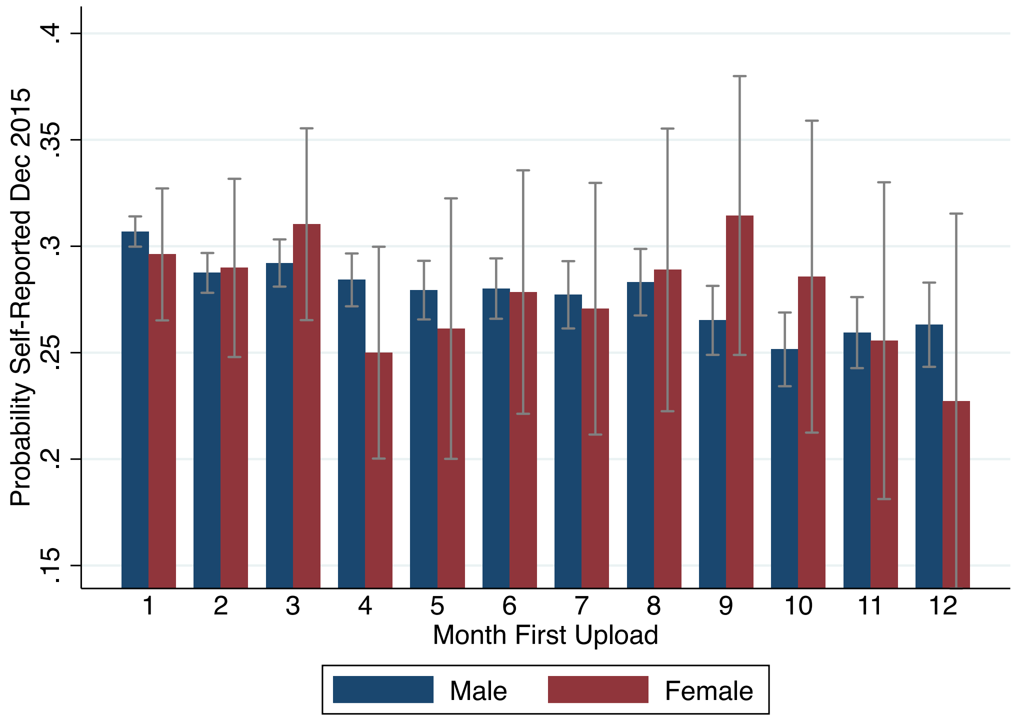
The profiles data provided to me is a cross-section of the profiles, therefore it is impossible for me to see when a candidate posted a new skill. Instead, I look at candidates whose first open source contribution on GitHub in that language occurred between January 2015 and December 2015. I then look by gender at the rates of self-reporting that language in December 2015. While we will expect that female candidates may have a different propensity to self-report in levels, I compare the rate of self-reporting by those who first uploaded code closer January 2015 versus those who uploaded code to December 2015. The figure below shows the results. The noisy estimates do not reveal systematic differences by gender in whether or not a candidate has self-reported based on how many months before they first uploaded code in that language.

Table 15: Testing for Mismeasurement Due to the Timing of Data Collection

	Saved - 12/15 - 11/16		Saved - 12/15 - 1/16	
	(1)	(2)	(3)	(4)
Female		0.002		-0.001
		(0.002)		(0.001)
SR JavaScript	0.027***	0.029***	0.018***	0.020***
	(0.004)	(0.004)	(0.003)	(0.003)
Female × SR JavaScript		-0.020		-0.026***
		(0.013)		(0.008)
SR PHP	0.008*	0.008*	0.002	0.001
	(0.004)	(0.005)	(0.003)	(0.003)
Female × SR PHP		-0.006		0.014
		(0.016)		(0.011)
SR Ruby	0.015***	0.015***	0.019***	0.018***
	(0.005)	(0.006)	(0.004)	(0.004)
Female × SR Ruby		0.006		0.008
		(0.019)		(0.014)
SR Python	0.020***	0.020***	0.012***	0.012***
	(0.004)	(0.005)	(0.003)	(0.003)
Female × SR Python		0.001		0.006
		(0.016)		(0.010)
SR C#	0.026***	0.027***	-0.000	-0.002
	(0.006)	(0.006)	(0.004)	(0.004)
Female × SR C#		-0.015		0.027*
		(0.021)		(0.016)
SR C++	-0.007	-0.009*	0.005	0.004
	(0.005)	(0.005)	(0.004)	(0.004)
Female × SR C++		0.025		0.012
		(0.018)		(0.012)
SR Java	-0.001	-0.001	-0.002	-0.002
	(0.004)	(0.004)	(0.003)	(0.003)
Female × SR Java		0.003		-0.002
		(0.013)		(0.008)
Group Fixed Effect	Yes	Yes	Yes	Yes
N	78,731	78,731	78,731	78,731
N Groups	17,679	17,679	17,679	17,679
Dep. Mean	0.05	0.05	0.02	0.02
R^2	0.009	0.009	0.008	0.008

The above table shows the coefficients from estimating Equation 3 using the Match Profiles dataset. The dependent variable in the first two columns is an indicator for the candidate being saved between December 2015 and November 2016. The dependent variable in the

Figure 4: Probability A Candidate is Saved Given the Number of Self-Reported Skills



Candidates are grouped by the number of self-reported skills on their profile in bins of 5. The probability that a candidate is saved is then computed for candidates within each bin. The probability is shown on the vertical axis, and the horizontal axis shows the number of self-reported skills. Along with the probability of being saved, a 95% confidence interval on the probability is displayed. All profiles are used in the above figure.

14. Self-Reporting Across Levels of Experience

In the following table, I estimate Equation 1. Instead of a single indicator for the gender of the candidate, I include indicators for the quartile of the candidate based on lines of code in a language that they have uploaded to open source websites.

The first column includes just the indicators for the quartile and the interaction with the gender of the candidate. The significant negative values for each of the quartiles shows that the gender gap is present for all levels of experience. This finding parallels the gender gap seen in Figure 1.

The later columns include increasing numbers of controls.

Table 16: Predicted Probability a Programmer Self-Reports by Quartile of Experience

	Self-Reported			
	(1)	(2)	(3)	(4)
Code Quartile=2	0.022***	0.019***	0.014***	0.014***
	(0.002)	(0.002)	(0.002)	(0.002)
Code Quartile=3	0.086***	0.083***	0.072***	0.072***
	(0.002)	(0.002)	(0.002)	(0.002)
Code Quartile=4	0.204***	0.195***	0.183***	0.183***
	(0.002)	(0.002)	(0.002)	(0.002)
Code Quartile=1 x Female	-0.031***	-0.021***	-0.022***	-0.022***
	(0.006)	(0.006)	(0.005)	(0.005)
Code Quartile=2 x Female	-0.022***	-0.015***	-0.021***	-0.021***
	(0.005)	(0.005)	(0.005)	(0.005)
Code Quartile=3 x Female	-0.025***	-0.020***	-0.026***	-0.026***
	(0.006)	(0.006)	(0.006)	(0.006)
Code Quartile=4 x Female	-0.043***	-0.037***	-0.046***	-0.046***
	(0.008)	(0.008)	(0.007)	(0.007)
Bachelors in CS			0.049***	0.049***
			(0.003)	(0.003)
Current Coder			0.150***	0.150***
			(0.002)	(0.002)
Constant	0.202***	0.135***	-0.042*	-0.042*
	(0.002)	(0.002)	(0.023)	(0.023)
Geo. FE	No	No	Yes	Yes
BA Year FE	No	No	Yes	Yes
Schools FE	No	No	Yes	Yes
Employers FE	No	No	Yes	Yes
Other Verified Langs. FE	No	Yes	Yes	Yes
N	418,720	418,720	418,720	418,720
N Candidates	170,886	170,886	170,886	170,886
Dependent Mean	0.28	0.28	0.28	0.28
Adj. R^2	0.03	0.04	0.10	0.10

Standard errors in parentheses

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

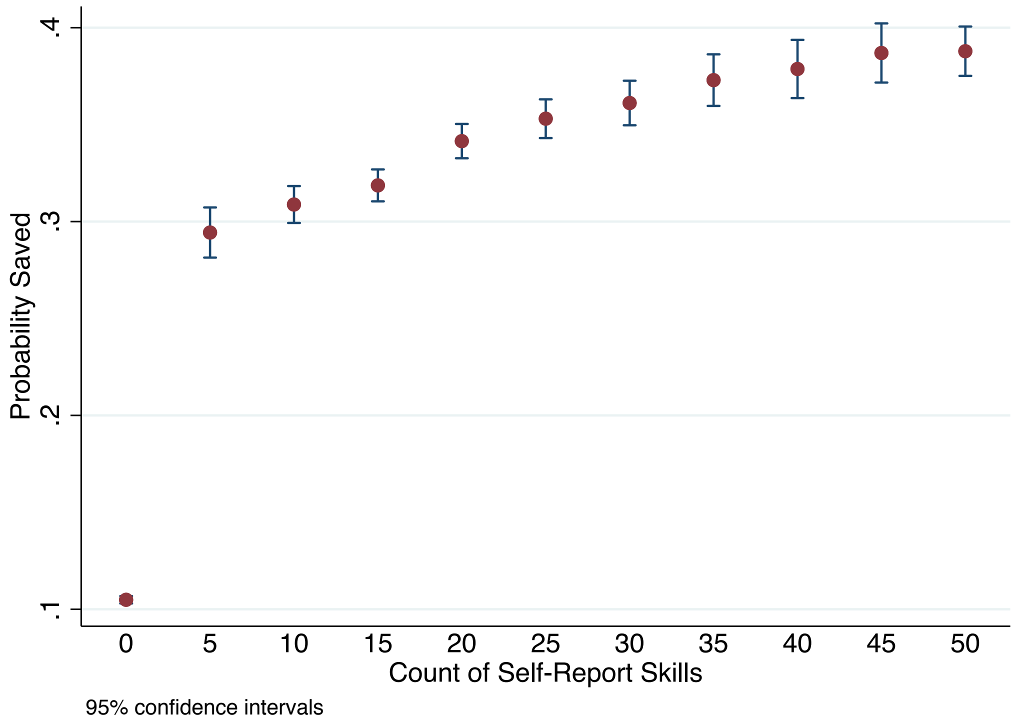
The above table shows the estimates of Equation 1 using the candidate-language pair observations. Female is an indicator for the candidate being female. Candidate-language pairs are binned by language into quartiles based on the number of lines of code that the candidate has uploaded over their lifetime to open source websites in that language. Bachelors in CS is an indicator for a bachelor's degree in Computer Science being present on the resume. Current coder is that the listed current occupation title is

15. Non-Linear Predicted Returns for Self-Reporting Skills

There are potential reasons why a candidate would find it costly to self-report a large number of skills. First, there is the cost of dealing with messages from recruiters. Self-reporting skills that are relevant to jobs that one is not truly interested in could translate into receiving recruiting messages form employers with non-relevant job opportunities. Second, self-reporting skills could mean being tested on those skills during interviews. If someone is not particularly confident in their proficiency in that skill, they may feel embarrassed or miss an opportunity for a job if they do not perform well during an interview. Third, there is a social cost of self-reporting skills. The job seeker site for this recruiting platform is publicly available for others to see. If individuals feel like they will be judged about their abilities if they self-report skills for which they are marginally proficient, this could be considered a cost.

Another potential cost is that the recruiters would not trust candidates would list many self-reported skills, and thus, the returns to listing additional skills would be diminishing. I find evidence of this by examining the observed probability that candidates are saved given the number of skills on their resumes. In the figure below, I show that the predicted returns to self-reported skills are diminishing.

Figure 5: Probability A Candidate is Saved Given the Number of Self-Reported Skills



Candidates are grouped by the number of self-reported skills on their profile in bins of 5. The probability that a candidate is saved is then computed for candidates within each bin. The probability is shown on the vertical axis, and the horizontal axis shows the number of self-reported skills. Along with the probability of being saved, a 95% confidence interval on the probability is displayed. All profiles are used in the above figure.

16. Self-Reporting of Other Languages

The below table shows the results of estimating Equation 1 using candidate-language pairs for all the main programming languages considered in this paper.

Table 17: Predicted Probability of Self-Reporting Languages

	Self-Reported						
	(1) JavaScript	(2) PHP	(3) Ruby	(4) Python	(5) C#	(6) C++	(7) Java
Female	-0.022***	-0.025***	-0.018***	-0.036***	-0.055***	-0.030***	-0.016**
	(0.005)	(0.007)	(0.007)	(0.006)	(0.012)	(0.009)	(0.007)
Ln(Code)	0.008***	0.007***	0.011***	0.015***	0.006***	-0.002***	0.005***
	(0.000)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)
Ln(Days with Uploads)	0.033***	0.029***	0.031***	0.036***	0.027***	0.030***	0.031***
	(0.001)	(0.001)	(0.001)	(0.001)	(0.002)	(0.001)	(0.001)
Ln(Accepted OSS Code)	-0.001	-0.001	-0.001	0.007***	0.003*	-0.002**	0.003***
	(0.001)	(0.001)	(0.001)	(0.001)	(0.002)	(0.001)	(0.001)
Bachelors in CS	0.044***	0.024***	0.042***	0.020***	0.068***	0.077***	0.095***
	(0.004)	(0.006)	(0.005)	(0.004)	(0.008)	(0.006)	(0.006)
Current Coder	0.186***	0.153***	0.134***	0.119***	0.142***	0.101***	0.152***
	(0.003)	(0.004)	(0.004)	(0.003)	(0.005)	(0.004)	(0.004)
Constant	-0.044	-0.086**	-0.025	-0.064	0.079	-0.010	0.016
	(0.033)	(0.034)	(0.046)	(0.040)	(0.061)	(0.047)	(0.058)
Geo. FE	Yes	Yes	Yes	Yes	Yes	Yes	Yes
BA Year FE	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Schools FE	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Employers FE	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Other Verified Langs. FE	No	No	No	No	No	No	No
N	115,328	49,104	64,090	61,516	30,556	38,153	59,973
N Candidates	115,328	49,104	64,090	61,516	30,556	38,153	59,973
Dep. Mean	0.31	0.25	0.29	0.20	0.27	0.23	0.35
Adj. R^2	0.14	0.09	0.13	0.14	0.10	0.09	0.12

Standard errors in parentheses

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

The above table shows the estimates of Equation 1 using the candidate-language pair observations. Female is an indicator for the candidate being female. Code is the number of lines of code that the candidate has written and uploaded in a programming language. Days with Uploads are the number of days in which the candidate uploaded open source code. Accepted OSS Code are the number of lines of code that the candidate has uploaded and had accepted into others' open source projects. BA in CS is an indicator for a bachelor's degree in Computer Science being present on the resume. Current coder is that the listed current occupation title is associated with being a programmer. Geographic Fixed Effects are indicators for the state and region listed on the resume. Employers Fixed Effects are indicators for the names of all current and previous employers. Other Verified Languages are indicators for all the languages that an individual has verified on their resume.

17. Propensity Score Matching

Examining if recruiters show more interest in candidates who self-report knowledge of languages by matching profiles on attributes is a compelling exercise. That methodology approximates the compelling audit study methodology (**???**). A drawback, however, to using matching profiles is that it requires matched profile groups to have enough variation within the groups in order to estimate the coefficients. If profile groups do not have many candidates then they cannot be used for estimating the coefficients.

An exercise that uses more of the data but has a similar motivation is propensity score matching. In this exercise, I match profiles by predicting the probability that a candidate self-reports each language based on the attributes of the profiles. I then use that predicted value as a propensity score. While there are still candidates who are dropped due to lack of variation, the number is somewhat smaller.

Table 18: Propensity Score Matching Average Treatment Estimates

	Saved						
	JavaScript	PHP	Ruby	Python	C#	C++	Java
ATE							
SR JavaScript	0.116***						
	(0.003)						
SR PHP		0.077***					
		(0.003)					
SR Ruby			0.147***				
			(0.005)				
SR Python				0.116***			
				(0.004)			
SR C#					0.097***		
					(0.004)		
SR C++						0.076***	
						(0.004)	
SR Java							0.124***
							(0.003)
N	157,647	150,722	140,222	137,728	134,844	132,729	132,658
Dep. Mean	0.21	0.21	0.22	0.22	0.22	0.22	0.22

The above table shows the estimated average treatment effect from self-reporting a language based on propensity score matching. Each column uses a propensity score based on predicting the self-reporting of that language.

The positive estimated treatment effects imply that recruiters are more interested in candidates who self-report knowing a language than those who do not self-report.

18. Selection of Sample

The sample of candidates used for the main analysis in this paper comes from the set of candidates who are verified in at least one of the following languages: JavaScript, PHP, Ruby, Python, C#, C++, Java. In order to understand the selection into this sample, I run a regression on all of the candidates on the platform predicting which candidates will be verified in a programming language. The estimated coefficients are shown below:

Table 19: Predicting Verification in a Language

	Verified						
	(1) JavaScript	(2) PHP	(3) Python	(4) Ruby	(5) C#	(6) C++	(7) Java
Female	-0.017***	-0.008***	-0.010***	-0.009***	-0.005***	-0.006***	-0.009***
	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)
BA in CS	0.024***	0.010***	0.017***	0.012***	0.011***	0.012***	0.026***
	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)
Current Coder	0.038***	0.019***	0.019***	0.021***	0.011***	0.012***	0.019***
	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)
Constant	0.016***	0.009***	0.010***	0.008***	0.006***	0.006***	0.006***
	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)
Geo. FE	Yes	Yes	Yes	Yes	Yes	Yes	Yes
BA Year FE	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Schools FE	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Employers FE	Yes	Yes	Yes	Yes	Yes	Yes	Yes
N	3,744,305	3,744,305	3,744,305	3,744,305	3,744,305	3,744,305	3,744,305
Dep. Mean	0.02	0.01	0.01	0.01	0.01	0.01	0.01
Adj. R^2	0.04	0.02	0.03	0.02	0.01	0.02	0.03

Standard errors in parentheses

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

The above table shows the estimated coefficients from regressions predicting if a candidate is verified in each language. Each column predicts a different programming language.

The coefficients are positive for a candidate having a Bachelors in Computer Science and currently being employed as a coder. The coefficient on the candidate being female is negative and significant.

This tells us that even after controlling for educational background and employment history factors, female coders are less inclined to participate in open source coding. Thus, the female coders in the sample used for the main analysis in this paper may be particularly committed coders. This is consistent with the results of **?** who show that on average the open source code contributions of female coders are more frequently accepted by other coders (although only when it is not obvious that the contribution was made by a female coder).

19. Self-Reporting Regressions Using Semi-Parametric Controls

Another way to examine the self-reporting behavior of job seekers is to control non-parametrically for the observable attributes of the candidate. In the below regression, I take the candidate-language pairs where the candidate is also in the Matched Profile dataset. I then run a linear probability model with the dependent variable of whether or not the candidate self-reported a language and independent variables of the candidate being female as well as a fixed effect for the profile group of the candidate.

Table 20: Self-Reporting Languages Using Matched Profile Sample

	(1) Self-Reported
Female	-0.026***
	(0.005)
Constant	0.246***
	(0.001)
Profile Group FE	Yes
N	180,732
N Candidates	85,390
Dependent Mean	0.24
R2	0.19

Standard errors in parentheses
* $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$

The above table shows the estimated coefficient from a regression using the candidate-language pairs where the candidate is in the Match Profile sample. The dependent variable is an indicator for if the candidate self-reports a language. The independent variables are a fixed effect for the profile group and an indicator for if the candidate is female.

The results confirm that female candidates are on average less likely to self-report knowing a programming language that they have verified experience coding in.