A Statistical Study of Female Students in a Software Engineering Class: Preparedness, Performance, and Contribution

1st Jialin Cui
Department of Computer Science
North Carolina State University
Raleigh, USA
jcui9@ncsu.edu

2nd Runqiu Zhang
Department of Statistics
University of Virginia
Charlottesville, USA
rz2cv@virginia.edu

3rd Qinjin Jia
Department of Computer Science
North Carolina State University
Raleigh, USA
qjia3@ncsu.edu

4th Fangtong Zhou
Department of Computer Science
North Carolina State University
Raleigh, USA
fzhou@ncsu.edu

5th Ruochi Li
Department of Computer Science
North Carolina State University
Raleigh, USA
rli14@ncsu.edu

6th Ed Gehringer
Department of Computer Science
North Carolina State University
Raleigh, USA
efg@ncsu.edu

***Abstract*—This is a research-to-practice full paper. Several research studies indicate that women who have opted into a computing career path must regularly contend with negative stereotypes about their technical abilities. These stereotypes are often cited as contributing factors to the underrepresentation of women in computing. To counter these stereotypes and enhance female participation in computer science, numerous interventions have been designed. However, most existing research tends to rely on anecdotal evidence and questionnaires to study these stereotypes. In contrast, our study collected data from over 900 students over a span of eight years and adopted a comprehensive quantitative approach to examine these stereotypes about female students. We utilized pre-class GitHub contribution metrics to evaluate students' programming experience and an array of in-class grading items to measure students' performance. Additionally, we mined the project repositories' git logs to gain insights into students' contributions to team projects. Our investigation began by probing whether there was a notable difference in the technical backgrounds or preparedness between female and male students. The results indicated that males tended to be better prepared. Next, we explored potential disparities in class performance between the two genders. Our findings revealed that males and females each excelled in different areas. We were also interested in discerning if female and male students contributed equally to team projects; our analysis affirmed that the contributions were comparable between the two groups. If allowed to choose their teammates, we examined whether they showed a preference for single-gender teams or mixed-gender teams. Our conclusions indicated no marked preference. This paper aims to augment the body of research on computing education by assisting educators in gaining a better understanding of female students in the class. Moreover, it tests the stereotypes by comparing them with empirical results.**

***Index Terms*—Software Engineering Education, Gender Study, Statistical Study, Quantitative Study, GitHub**

I. Introduction

The field of computer science, both within industry and academic settings, is characterized by a persistent gender imbalance, as documented in recent statistics by the National Science Foundation (NSF) [1], [2]. Despite increasing demand for skilled professionals and a growing emphasis on diversity and inclusion, women remain significantly underrepresented. This underrepresentation is further compounded by enduring negative stereotypes regarding women's technical abilities [3], which are often supported by the misguided belief that women are not as inherently suited for the field as their male counterparts [4].

In an effort to combat these stereotypes and create a more inclusive environment, a variety of interventions have been implemented. These range from public awareness campaigns that highlight the achievements of women in technology, to substantial revisions in educational curriculums designed to be more welcoming to students of all genders. Nonetheless, the effectiveness of these interventions is frequently questioned. Many existing studies rely heavily on anecdotal evidence, self-reported surveys, and relatively small sample sizes, which can introduce biases and fail to reflect broader trends.

To provide a more robust analysis, we conducted a comprehensive statistical study that encompassed a dataset of 982 students (772 males and 210 females) collected over an eight-year period. This study scrutinizes various facets of the students' experiences within the realm of computing education. We used fourteen distinct grading metrics to evaluate performance in a software engineering class. Prior programming experience was quantified using pre-class contributions to GitHub, a method previously validated by multiple studies [5]–[11] as a reliable indicator of programming proficiency.

Furthermore, we meticulously analyzed the contributions of each student within team projects, with a particular focus on dissecting the contributions by gender. Our research aims to provide an exhaustive examination of the disparities between female and male students in their software engineering education, potentially offering insights into the effectiveness of current educational strategies and interventions.

Our study rigorously investigates four central questions that address the core issues stemming from the prevailing stereotypes about gender roles in computer science:

RQ1. Do female students' technical backgrounds significantly differ from those of their male peers?

RQ2. Are there marked disparities in class performance between the genders?

RQ3. In team projects, do contributions from female and male students equate?

RQ4. If allowed to choose teammates, do students tend to form single-gender teams or mixed-gender teams?

Through these inquiries, our study aims to provide educators and policymakers with robust empirical insights. By presenting new evidence on these pertinent issues, we hope to foster a more informed, data-driven approach to tackling the significant questions surrounding gender dynamics in computer science education. The outcomes could potentially influence curriculum designs, teaching methods, and intervention strategies aimed at enhancing diversity and inclusivity within the field.

II. Related Work

The study of gender disparities in computer science education has become increasingly relevant due to ongoing concerns about equity and diversity within the field. Numerous research efforts have examined disparities in preparedness before entering classes, performance during the coursework, and contributions within team projects. Recognizing and addressing these disparities is critical not only for educational equity but also because diversity in the workplace is widely recognized to foster innovation through a variety of experiences and perspectives [12]–[15]. This section offers a detailed overview of significant studies that have illuminated these dimensions.

A. Preparedness in Computer Science: Gender Perspectives

The issue of gender-based differences in preparedness at the outset of computer science education is well-documented, with several studies illustrating a clear divide. Duran et al. [16] provide a comprehensive analysis of the preparedness levels of 4728 students in both MOOCs and traditional courses, discovering a notable disparity in prior experience between men (64.3%) and women (35.7%). Their findings suggest systemic differences in the exposure to and engagement with computer science before formal education begins, highlighting the need for targeted pre-university outreach and support programs.

Wilcox et al. [17] further explore this theme by assessing the prior exposure of 153 students to programming. Their initial surveys reveal stark differences: 13% of female students reported no prior programming experience, compared to just 1% of their male counterparts, and only 38% of female students were familiar with Java programming, versus 62% of males. These statistics underscore the importance of introductory programming courses that accommodate varying levels of prior knowledge, ensuring that all students have the opportunity to succeed.

Babes-Vroman et al. [18] extend these findings in their survey of 1843 students from a leading public R1 research institution. Their research corroborates earlier results, showing that male students are more likely to have engaged in self-taught programming activities, which could give them an advantage in early computer science courses.

B. Performance Differences by Gender in CS Courses

The performance of students in computer science courses, as stratified by gender, is a central focus of academic investigation. Ioannis et al. [19] studied 89 students and found statistically significant gender-based differences in average scores across several course modules, including core programming and advanced software topics, with males generally outperforming females. These findings raise questions about the inclusivity of teaching methods and the potential need for pedagogical strategies that address diverse learning styles.

In contrast, Wagner [20] conducted a comprehensive review of student performance data from 129 UK universities over a decade and noted that male students tended to secure a larger share of first-class degrees. However, female students had higher average UCAS points upon entry, suggesting that the academic potential measured at admission did not translate into similar outcomes, indicating possible systemic issues within the educational environment.

Byrne et al. [21], however, offer a different perspective by analyzing the performance of 110 students and finding negligible differences in average scores between genders. While females had a slightly superior mean score (43.9% vs. 39.7% for males), the difference was not statistically significant. This suggests that under certain conditions, such as when teaching is adjusted to be more inclusive, gender may not significantly impact performance.

C. Gendered Differences in Team Contributions

Within team projects, gender dynamics can influence both the process and outcomes of collaborative work. Meadows et al. [22] found that men were more likely to dominate technical discussions and presentations, displaying 20% more technical slides than women, who presented more non-technical content. This division of labor not only reflects but also reinforces stereotypes about gender roles in technology.

A follow-up study by Meadows et al. [23] showed that female students frequently took on more administrative and report-writing tasks, suggesting that even within ostensibly equitable groups, traditional gender roles often prevail. This

indicates a need for interventions that promote equitable participation in all aspects of project work.

Strehl et al. [24] explored how these biases extend to task allocation, finding that even when individuals possess identical skills, gender biases in task delegation persist. This challenges the assumption that technical prowess alone determines task assignments and underscores the influence of deeply ingrained societal stereotypes.

The body of literature on gender disparities in computer science education provides a robust foundation for understanding the challenges faced by female students. Each study not only adds to our understanding of these issues but also highlights the complexity of effectively addressing them within educational settings. Our research builds on this groundwork, aiming to further dissect and understand these dynamics in computer science education.

III. Background

This section provides a detailed overview of the course structure and the educational environment in which our research was conducted, shedding light on the specific aspects that are directly relevant to our study of gender disparities in a master's level software engineering course.

A. Class Structure

The course at the heart of our research is a master's level software engineering course focused on object-oriented design using Ruby on Rails, a popular server-side web application framework. Over the past fifteen years, this course has been consistently overseen by the same instructor, ensuring a stable and controlled educational environment for our study.

Students in this course are introduced to the Ruby programming language and the Ruby on Rails framework. The curriculum is carefully structured to cover a broad range of foundational and advanced topics. Initially, students engage with the basics of object-oriented design, including creating use cases, refactoring, Test-Driven Design (TDD), Behavior-Driven Design (BDD), as well as Agile and Scrum methodologies. These topics are essential for understanding modern software development practices and provide a practical framework within which students can develop their coding skills.

As the course progresses, more complex topics are introduced. These include the SOLID design principles, first proposed by Robert C. Martin in his influential work on Agile software development [25], and various design patterns, such as those cataloged by the Gang of Four (GoF) [26]. These principles and patterns are crucial for writing clean, maintainable code, and understanding them is essential for any software engineer.

The assessment strategy for the course is comprehensive and designed to evaluate a wide range of skills. It includes fourteen grading items that assess both individual and collaborative skills: two mid-term tests, one final exam, one GitHub tutorial assignment, four code-based tasks, two design documents, three peer-reviewed assignments, and evaluations of team members. This variety ensures that students are not only tested on their theoretical knowledge but also on their practical and interpersonal skills.

The projects integrated into the course are particularly significant. The first project helps students tackle basic coding challenges similar to those they might encounter in technical interviews, thereby enhancing their practical coding skills in Ruby. In the subsequent projects, students work in teams to develop comprehensive web applications using Ruby on Rails. This collaborative aspect of the course is vital, as it mirrors real-world software development environments.

Later in the course, students engage in projects involving real Open Source Software (OSS) hosted on GitHub. These projects, which are part of an NSF-backed initiative, require students to apply their coding and design skills to contribute to ongoing software projects. Tasks in these projects include refactoring existing code, creating tests, troubleshooting, and adding new functionalities. Each team is required to submit a pull request to the primary repository, These projects provide students with real-world experience in software development and open-source contribution.

B. Data Collection

This subsection outlines the comprehensive methods employed to gather and analyze data, ensuring a robust approach to understanding gender disparities in computer science education through various metrics and sources.

1) Gender Information and Class Grades: With approval from the Institutional Review Board (IRB), we meticulously collected gender information directly from the University's registration office. This approach guarantees complete accuracy in gender identification, avoiding the common inaccuracies encountered in studies that attempt to infer gender based on names. Additionally, we compiled detailed performance data through fourteen grading items, as described in Section III-A. This data spans from 2016 to 2023, allowing us to analyze trends and draw insights over a significant period, thus providing a deep understanding of academic performance across different cohorts within the master's level software engineering course.

2) Technical Background: We conducted a thorough aggregation of all project submissions presented as pull requests from 2011 to 2023. By linking GitHub accounts associated with these pull requests to individual students using class rosters, we ensured a high degree of data integrity. Although not all student GitHub accounts could be retrieved, our methodology achieved a retrieval rate exceeding 90% for each semester post-Fall 2016, as illustrated in Figure 1. This high retrieval rate supports robust analysis and interpretation of the data.

Employing the GitHub API, we developed custom tools to fetch publicly available GitHub contribution statistics, capturing a comprehensive snapshot of each student's programming involvement up to July 2023. This method provided us with a plethora of metrics, some of which required additional calculations on our part to derive meaningful insights. For example, while GitHub directly reports the volume of code

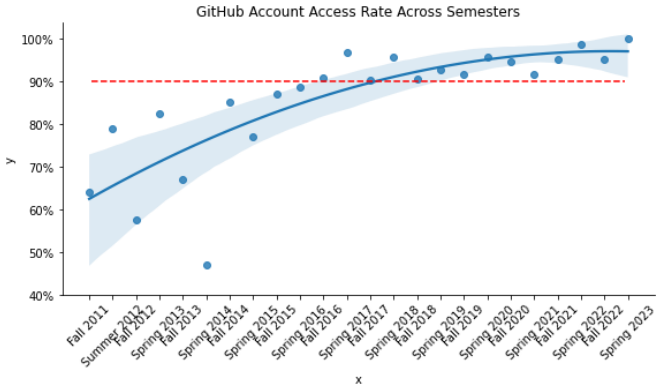

Fig. 1: GitHub Account Access Rate Across Semesters from 2011 to 2023

written in specific languages for individual repositories, we computed the cumulative code size for each language a student used across all their repositories. This involves summing up the total amount of code written in each language, such as Python, JavaScript, or Java, to provide a comprehensive view of a student's coding activity by language. We distinguished between Type I repositories, which a user personally owns, and Type II repositories, to which a user contributed but did not own. Our analysis focused on Type I repositories, as they more accurately reflect a user's personal expertise and coding history. Moreover, we computed the code volume within a user's Type I repositories in each of the top 10 languages on GitHub [27]. These languages include Python, JavaScript, Java, TypeScript, Shell, C++, Ruby, PHP, C#, and C.

TABLE I: GitHub Metrics Collected from GitHub

Metric	Explanation
Days	Number of days the user has experience on GitHub before the class.
PC	The number of private contributions made by this user.
commits	The number of public commits made by this user.
comments	The number of comments made by this user in commits, issues, and pull request discussions.
PR	The number of pull requests made by this user.
PR review	The number of pull requests reviewed by this user.
issues	The number of issues created by this user.
repos	The total number of repositories created by this user or the user collaborated in.
code size	total size (storage size in Byte) of type I repositories owned by this user.
languages	The number of distinct programming languages used in the user's type I repositories.
pop size	The amount (storage size in Byte) of code written in popular GitHub languages within the user's type I repositories.
TC	The total contribution (TC) is calculated by summing the user's private contributions, commits, comments, PRs, PR reviews, repos and issues.

For each metric in Table I, its creation date was retrieved through the GitHub API. We then mapped these metrics to the academic calendar from 2016-2023 to align each student's GitHub activity with the corresponding semester start date, ensuring an accurate representation of their prior programming experience. This meticulous approach to data collection supports a nuanced analysis of the technical background of students, providing vital insights into how prior experience correlates with academic performance and engagement in the course.

3) Contribution to Teams: Our analysis of team contributions focuses on data from the second project during the period from Fall 2021 to Spring 2023, as described in Section III-A. This project was selected for detailed analysis for two primary reasons: 1) the consistency of the project theme, which involves building a web application for different customer classes, ensured uniformity across semesters, thus facilitating meaningful inter-semester comparisons; 2) the OSS projects undertaken in the third and fourth projects saw each team working on different aspects of the code, which presented substantial challenges for a quantitative comparison of contributions due to their disparate nature. Consequently, we excluded these projects from our analysis. The decision to focus on the last four semesters was driven by the availability of detailed and reliable data, which was not as available in earlier semesters.

Using the GitHub API, we meticulously extracted detailed contribution data for each team member within the project teams. This included the number of commits made, lines of code added, and files changed in the team project repositories. For the purpose of our analysis, we chose to focus on the number of commits and lines of code added as primary metrics. We reasoned that these metrics more directly reflect the substantive contributions to the development process, whereas the number of deletions and file changes were considered less indicative of overall team contribution and thus omitted from our primary analysis.

Additionally, we gathered comprehensive data on team assembly, specifically examining the gender composition of three-member teams throughout the entire duration of the course from 2011 to 2023. This dataset helps us explore potential biases towards forming single-gender or mixed-gender teams, a key aspect in understanding the dynamics of team collaboration in educational settings.

4) Final Dataset: In sum, our final dataset comprises three distinct sub-datasets, each pivotal for addressing specific research questions due to the inherent limitations and strengths in the data-collecting process:

Pre-class GitHub Contributions and In-class Grades: We collected a comprehensive set of data for 982 students (772 males and 210 females), which includes detailed GitHub contribution metrics prior to their enrollment in the course and their academic performance across multiple grading items. This dataset is instrumental in addressing RQ1 (technical background differences) and RQ2 (performance disparities based on gender).

Commit and Code Addition Records: For a subset of 237 students (grouped into 79 trios) across four semesters, we maintain detailed records of commit numbers and code additions. This dataset facilitates our examination of RQ3, which

investigates whether the contributions within team projects are equitably distributed among genders.

Gender Composition of Teams: Finally, we have accumulated data on the gender composition of 306 teams (comprising 918 individuals) from 2011 to 2023. This extensive dataset allows us to explore RQ4, focusing on whether there is a preference for single-gender or mixed-gender teams within the course's collaborative projects.

These datasets, with their respective focuses, provide a robust foundation for a comprehensive analysis of gender dynamics in computer science education, offering insights into the impacts of gender on various aspects of academic and collaborative performance within the course.

IV. Methodology

This section elaborates on the statistical methodologies and data processing techniques employed to analyze the complex dataset we compiled, aiming to uncover gender disparities in software engineering education and teamwork dynamics.

A. Processing Raw Data

To standardize and compare performance metrics across different academic terms and assignment types, we applied a z-score transformation to each grading item referenced in Section III-A. This normalization process involves subtracting the mean and dividing by the standard deviation of each grade type within the corresponding semester. This allows us to mitigate the effects of varying grading scales and instructional rigor over time, facilitating a fair comparison across semesters and student cohorts.

Furthermore, to encapsulate a broader perspective of student performance, we aggregated these z-scores into average scores for each category of assignments: exams, projects, design documents, and peer reviews. This aggregation resulted in eighteen distinct z-scores per student (fourteen from individual assignments and four from averaged categories), which were then utilized in subsequent statistical analyses.

B. Statistical Tests for Hypothesis Testing

To investigate the potential differences in technical backgrounds, class performance, and team contributions between female and male students, we employ the t-test because of its efficacy in comparing means between two groups. The choice of Student's t-test or Welch's t-test depends critically on the equality of variances among the groups, which we first assess using Levene's test.

1) Levene's Test for Equality of Variances: Levene's test is pivotal in our analysis as it determines whether the assumption of equal variances is valid. This test is particularly robust against deviations from a normal distribution, making it ideal for real-world educational data, which often deviates from perfect normality. By examining the equality of variances across our gender-segmented groups, we can ascertain whether to proceed with a Student's or modified t-test approach. A significant result from Levene's test, indicating the presence of unequal variances, directs us to use a more suitable test for such conditions.

2) Welch's t-Test: When Levene's test indicates unequal variances among our sample groups, we opt for Welch's t-test. This version of the t-test does not assume homogeneity of variances and is designed to accommodate data with unequal variances and unequal sample sizes. Welch's test adjusts the degrees of freedom based on the sample sizes and variance estimates, which often results in a more reliable calculation of the p-value when comparing means under heteroscedastic conditions.

3) Student's t-Test: If the variances are found to be equal according to Levene's test, we employ the Student's t-test. This traditional parametric test assumes that the two groups are sampled from populations with equal variances and is used to examine if there is a statistically significant difference between the group means. This test is straightforward and powerful under conditions of homoscedasticity.

C. Hypothesis Testing across Different Areas

We specifically tested the following hypothesis across different areas.

1) Technical Backgrounds: Null Hypothesis (H0): There is no significant difference between the pre-class GitHub metrics for male and female students; the group means are equal. Alternative Hypothesis (H1): There is a significant difference between the pre-class GitHub metrics of male and female students; the group means are not equal. This hypothesis tests whether gender differences exist in the technical preparedness of students as measured by their activities on GitHub before class.

2) Class Performance: Null Hypothesis (H0): There is no significant difference in class performance scores between male and female students; the means are equal. Alternative Hypothesis (H1): There is a significant difference in class performance scores between male and female students; the means are not equal. This analysis will help determine if there is a systemic performance gap between genders within the academic setting.

3) Team Contributions: Null Hypothesis (H0): The average contributions to team projects by male and female students are equal. Alternative Hypothesis (H1): The average contributions to team projects by male and female students are not equal. This addresses the question of whether there is a gender disparity in how students contribute to team-based projects, which is critical for understanding collaborative dynamics in educational environments.

Each hypothesis is designed to uncover specific aspects of gender disparities in computer science education, leveraging robust statistical techniques to ensure the validity and reliability of our findings. These analyses not only help us understand existing disparities but also guide the development of targeted interventions to promote equity in educational outcomes.

D. Complex Team Dynamics and Gender Composition Analysis

Given the self-selected nature of teams in the class, understanding the underlying patterns in team composition required

a nuanced statistical approach:

1) Monte Carlo Simulation: To effectively model the expected frequencies of team compositions, we employed Monte Carlo simulation. This technique is particularly useful in scenarios where direct calculation methods become infeasible due to the complexities introduced by changing sample sizes after each team formation—a typical characteristic of hypergeometric distributions. We simulated the random formation of 306 three-member teams from our pool of 918 students across 100,000 iterations to estimate the baseline distribution of team gender compositions.

2) Chi-Square Goodness-of-Fit Test: Utilizing the expected frequencies derived from our Monte Carlo simulations, we then conducted a chi-square goodness-of-fit test. This test compares the observed frequencies of actual team formations to the simulated expected frequencies, assessing whether deviations from these expected patterns can be attributed to random chance or reflect a systematic bias toward certain team compositions based on gender.

Our null hypothesis posits that team formations are random with respect to gender, with any observed deviations being purely coincidental. Conversely, our alternative hypothesis suggests that gender plays a significant role in how teams are formed, indicating a preference for either single-gender or mixed-gender teams.

V. Results

Table II presents the results of Welch's t-tests and Student's t-tests for each type of pre-class GitHub contribution, comparing the means for male and female students. Days, commits, repos, languages, and TC metrics do not have equal variances, so we applied Welch's t-test. For the other metrics, we used the traditional Student's t-test. Cells with p-values below 0.05 are highlighted in yellow, indicating a statistically significant difference between the genders for that metric. The following are metrics that have statistically significant differences between male and female students.

Days Active: Males were active on GitHub significantly longer than females before the course, with males averaging 760.81 days compared to 560.34 for females. This suggests that male students may have earlier exposure to GitHub, which could influence their familiarity and comfort with this essential tool in software development.

Commits: Male students also committed to GitHub repositories more than twice as much as female students (47.03 vs. 22.46). This higher frequency of commits implies more consistent coding practices among male students, possibly reflecting greater confidence or interest in experimenting with and finalizing coding projects.

Repositories Created: The number of repositories created by males was also higher (4.82 vs. 2.93). Creating repositories can be an indicator of initiating new projects or coursework, suggesting that male students are more likely to experiment with new ideas or engage in side projects.

Programming Languages Used: Male students used more programming languages than females (5.24 vs. 3.58), indicating a broader exploration of different technologies and programming environments, which can be crucial for diverse problem-solving capabilities.

TABLE II: t-test Results of Mean GitHub Metrics: Female vs Male

Welchs t-test	t stat	p-value	Female	Male
Days	3.66	0.00	560.34	760.81
commits	3.55	0.00	22.46	47.03
repos	3.53	0.00	2.93	4.82
languages	2.73	0.01	3.58	5.24
TC	4.34	0.00	34.05	96.03
Student's t-test	**t stat**	**p-value**	**Female**	**Male**
PC	1.69	0.09	2.56	35.31
comments	0.62	0.54	2.57	3.24
PR	0.47	0.64	2.21	2.54
PR review	1.31	0.19	0.29	0.61
issues	1.37	0.17	1.03	2.49
code size	0.59	0.56	4467042	5992084
pop size	0.60	0.55	2700916	4079220

Total Contributions: Total contributions, encompassing commits, pull requests, and other GitHub activities, were significantly higher for males (96.03 vs. 34.05). This comprehensive metric underscores a higher level of overall engagement with GitHub among male students, which might translate to a more robust portfolio of work before class.

The following are metrics that do not have statistically significant differences between male and female students.

Private Contributions: Despite a higher average of private contributions from males (35.31 vs. 2.56), this difference was not statistically significant, suggesting that when contributions are not public, the engagement level between genders is more comparable.

Comments, Pull Requests, and Reviews: For comments, pull requests, and pull request reviews, no significant differences were found. This similarity in engagement levels in collaborative and communicative aspects of GitHub projects suggests that gender does not significantly influence how students interact with others in public repositories.

Issues Raised, Code Size, and Popular Code Size: These metrics also showed no significant gender differences, indicating that the scale of work and types of issues engaged with are similar across genders.

A similar interpretation can be made for table III. This table compares the difference in means of the eighteen grading metrics mentioned in section II. Design doc grade, design doc review grade, and review AVG grade metrics do not have equal variances, so we applied Welch's t-test. For the other metrics, we used the traditional Student's t-test. Cells with p-values below 0.05 are highlighted in yellow, signifying a statistically significant difference between male and female students for that particular grading metric. Cells shaded in pink indicate that females have a higher average z-score for that grading metric, while cells in blue denote a higher average z-score for males. The following are metrics that have statistically significant differences between male and female students.

Design Documentation and Review: Female students scored

TABLE III: t-test Results of Mean Grades: Female vs Male

Welchs t-test	t stat	p-value	Female	Male
Design doc	-2.76	0.01	0.15	-0.04
Design doc review	-2.24	0.03	0.13	-0.04
Review AVG	-1.23	0.22	0.08	-0.02
Student's t-test	t stat	p-value	Female	Male
Github Project	0.17	0.86	-0.018	0.003
Program 1	0.95	0.34	-0.064	0.016
Program 2	0.11	0.91	-0.003	0.006
OSS project	0.85	0.39	-0.048	0.018
Final project	0.42	0.68	-0.026	0.006
Program 2 review	-0.4	0.69	0.027	-0.003
OSS review	-1.14	0.25	0.084	-0.016
OSS writeup	0.72	0.47	-0.039	0.017
Exam 1	2.66	0.01	-0.159	0.047
Exam 2	1.78	0.07	-0.103	0.034
Final exam	3.72	0.00	-0.213	0.067
Project AVG	0.71	0.48	-0.040	0.016
Doc AVG	-0.92	0.36	0.060	-0.011
Exam AVG	3.42	0.00	-0.198	0.062
Teammate review	2.06	0.04	-0.123	0.036

higher in design documentation, with a mean z-score of 0.15 compared to -0.04 for males. This significant difference suggests that female students may excel in tasks requiring detailed documentation and structured thinking, skills crucial for successful software engineering projects.

Exams: Male students performed better in high-stakes testing environments, as indicated by significant differences in major exams. For the final exam, the mean z-score for males was 0.067, while for females, it was -0.213. This disparity could reflect different stress responses, study habits, or possibly variations in how exam content aligns with each gender's learning strengths.

Teammate Review: Another significant finding is in the context of peer evaluations, where male students received higher evaluations from their peers compared to female students. This could reflect social dynamics within teams or differences in perceived contributions to group work.

For many metrics such as GitHub project contributions, routine programming assignments, and other review scores, no statistically significant differences were observed. This indicates that in regular coursework and collaborative activities, male and female students perform comparably, suggesting that course design might effectively equalize opportunities for engagement and success.

Table IV shows the results of the Student's t-tests comparing the mean contributions in team projects between female and male students, encompassing metrics such as team commits, additions, and their corresponding percentages. The lack of significant differences across all metrics indicates that there is no substantial gender disparity in contributions to team projects.

Finally, Table V displays the observed and expected frequencies of the four possible team compositions, each consisting of three members, drawn from a pool of 200 females and 718 males. When forming teams with both female and male students, we can identify four combinations: 3M represents three males, 3F represents three females, 2M1F represents two males and one female, 2F1M represents two females and one male. Notably, our observed frequencies closely align with the expected frequencies. The Chi-Square Goodness-of-Fit test yielded a p-value of 0.38 based on our observed and expected frequencies.

TABLE IV: t-test Results of Mean Contributions: Female vs Male

Student's t-test	t stat	p-value	Female	Male
team commit	0.11	0.91	24.94	25.18
addition	0.69	0.49	1995	2243
commit%	1.31	0.19	31%	34%
addition%	1.15	0.25	30%	34%

TABLE V: Observed vs Expected Team Frequency

	Observed	Expected	Female	Male
3M	151	136	200	718
3F	4	4		
2M1F	114	127		
2F1M	37	39		

VI. Discussion

RQ1. Do female students' technical backgrounds significantly differ from those of their male peers?

From the perspective of pre-class GitHub contributions, the answer to this question is affirmative. We observed distinct variations in pre-class GitHub contributions between male and female students. Several categories showcase substantial differences in the means of pre-class GitHub contributions between the two groups. For instance, distinctions are evident in the number of pre-class commits, repositories owned, languages employed, and days of experience on GitHub. Male students, on average, engaged with GitHub earlier than their female counterparts. Before the class began, male students boasted an average of 760.81 days of GitHub experience, while females averaged 560.34 days. The mean commit contribution and the number of repositories for male students are roughly double those of their female peers (47.03 vs 22.46 for commits and 4.82 vs 2.93 for repositories). Additionally, male students tend to utilize an average of 5.24 programming languages, whereas female students typically use 3.58 languages.

These disparities hint at potential differences in technical backgrounds or previous experiences between male and female students prior to entering the class. Our findings align with several other studies that suggest male students often possess a more robust computer science background than female students [16]–[18]. While these studies drew their conclusions from undergraduate student populations, our research indicates that the technical background gap between male and female students persists even in our graduate institution. However, it's crucial to highlight that solely relying on GitHub contributions doesn't provide a comprehensive view of a student's entire technical background. So, although these differences are apparent, a deeper investigation is necessary

to arrive at a definitive conclusion about the overall technical backgrounds of students.

RQ2. Are there marked disparities in class performance between the genders?

Based on Table III, distinct grading metrics display significant differences between female and male students. Among these significant disparities, females excelled in areas like the "Design doc" and "Design doc review," whereas males outperformed in the first and final exams. Furthermore, male students had a higher exam average z-score (0.062 vs. -0.198).

A notable observation was the significant difference in teammate review scores between the genders. Male students, on average, earned a z-score of 0.036, while their female counterparts received a z-score of -0.123. This indicates that, on average, female students receive more negative ratings from their peers, given their negative z-score. Intriguingly, male and female students were found to make equal contributions to their team projects (see below), which hints at potential bias against female students in the perception of their peers.

Outside of these specific grading metrics, most other items showed no significant variations between the genders. Both male and female students performed equally well across all projects. An interesting observation worth highlighting is the female students' consistently higher average z-scores in documentation writing and review tasks. In contrast, male students earned higher average z-scores in most other grading categories, although many of these differences weren't statistically significant.

All in all, we do observe a statistically significant difference between male and female exam performance. Although male and female student contributions to team projects are equal, they tend to receive different teammate ratings.

RQ3. In team projects, do contributions from female and male students equate?

From Table IV, examining team project contributions such as commits, additions, and their corresponding percentages, we find no statistically significant difference between female and male students' contributions. This indicates that both genders are similarly active and contribute at comparable levels to their team projects. While male students display a slightly higher mean value, the difference isn't statistically significant. Such findings are encouraging, suggesting that in collaborative environments like team projects, both genders contribute equally. Even though earlier analysis suggested that female students might possess a weaker technical background and receive lower peer ratings, there is no discernible difference in their contributions to team projects when compared to their male counterparts.

RQ4. If allowed to choose teammates, do students tend to form single-gender teams or mixed-gender teams?

The earlier discussion regarding observed and expected frequencies for team compositions indicates that observed frequencies closely align with expected ones. A p-value of 0.38 suggests that we lack sufficient evidence to reject the null hypothesis. This implies that when given the freedom to select teammates, students do not exhibit a strong preference for single-gender teams over mixed-gender teams, or vice versa. Team composition seems organic, likely influenced by various factors besides gender. Our findings affirm that female computer science students are not marginalized in team project scenarios.

In conclusion, while certain differences in technical backgrounds and class performance exist between male and female students, they contribute equally in team projects. When forming teams, students don't exhibit a strong preference based on gender. Our findings do not support negative stereotypes about females' technical abilities; given the same time frame (i.e., the team project duration), both females and males contributed comparably to the project. This observation counters any assumption of a technical ability gap between the genders. Instead, there may be a preparedness gap, which may have its roots in cultural factors or individual interests rather than innate ability. These insights are vital for educators to understand and better cater to the needs of their diverse student population. However, one thing worth noting is that although female students are not marginalized in team formation, they do receive lower teammate ratings in team projects. This is an interesting phenomenon that warrants further analysis.

VII. Threats to Validity

A primary concern regarding the validity of our study pertains to our data source. Since we gathered data exclusively from our class, our conclusions might not be universally applicable. Specifically, the background of postgraduate students could differ from that of undergraduates. Moreover, our class, akin to many graduate courses, was predominantly composed of international students, with a significant majority coming from India. Thus, the differences we observed could be attributed to cultural variations. We recognize that not every student might opt for GitHub as their preferred platform for version control and collaboration. Hence, a proficient student might favor other platforms over GitHub, leading to an incomplete representation of their actual experience in our dataset. Nonetheless, given its current prominence, GitHub remains a preeminent choice.

VIII. Conclusions

Our investigation into the distinctions between male and female computer science students in terms of technical background, class performance, and team project contributions has yielded enlightening insights. While male students generally showcased a stronger presence in pre-class GitHub contributions, potentially indicating a more robust technical foundation, the actual class performance and team contributions highlighted a more nuanced picture. Both genders made comparable contributions in team projects, and class performances showed areas of strength for both groups.

References

[1] N. S. Foundation, "Bachelor's, master's, and doctor's degrees conferred by postsecondary institutions, by sex of student and field of study: 2019-20," National Science Foundation, Tech. Rep., 2020. [Online]. Available: https://nces.ed.gov/programs/digest/d21/tables/dt21_318.30.asp

[2] ——, "Employment status of scientists and engineers, by age, sex, ethnicity, race, and disability status: 2021," National Science Foundation, Tech. Rep., 2021. [Online]. Available: https://ncses.nsf.gov/pubs/nsf23315/data-tables

[3] D. I. Miller, A. H. Eagly, and M. C. Linn, "Women's representation in science predicts national gender-science stereotypes: Evidence from 66 nations." *Journal of Educational Psychology*, vol. 107, no. 3, p. 631, 2015.

[4] S. Cheryan, V. C. Plaut, C. Handron, and L. Hudson, "The stereotypical computer scientist: Gendered media representations as a barrier to inclusion for women," *Sex roles*, vol. 69, pp. 58–71, 2013.

[5] J. Cui, R. Zhang, R. Li, Y. Song, F. Zhou, and E. Gehringer, "Correlating students' class performance based on github metrics: A statistical study," in *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*, ser. ITiCSE 2023. New York, NY, USA: Association for Computing Machinery, 2023, p. 526–532.

[6] J. Cui, R. Li, K. Lou, C. Liu, Y. Xiao, Q. Jia, E. Gehringer, and R. Zhang, "Can pre-class github contributions predict success by student teams?" in *Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Software Engineering Education and Training*, ser. ICSE-SEET '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 40–49. [Online]. Available: https://doi.org/10.1145/3510456.3514144

[7] J. Cui, R. Zhang, R. Li, F. Zhou, Y. Song, and E. Gehringer, "How pre-class programming experience influences students' contribution to their team project: A statistical study," in *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, ser. SIGCSE 2024. New York, NY, USA: Association for Computing Machinery, 2024, p. 255–261. [Online]. Available: https://doi.org/10.1145/3626252.3630870

[8] J. Cui, F. Zhou, R. Zhang, R. Li, C. Liu, and E. Gehringer, "Predicting students' software engineering class performance with machine learning and pre-class github metrics," in *2023 IEEE Frontiers in Education Conference (FIE)*, 2023, pp. 1–9.

[9] J. Cui, R. Zhang, R. Li, F. Zhou, Y. Song, and E. Gehringer, "A comparative analysis of github contributions before and after an oss based software engineering class," in *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*, ser. ITiCSE 2024. New York, NY, USA: Association for Computing Machinery, 2024, p. 576–582. [Online]. Available: https://doi.org/10.1145/3649217.3653535

[10] J. Cui, F. Zhou, C. Liu, Q. Jia, Y. Song, and E. Gehringer, "Utilizing the constrained k-means algorithm and pre-class github contribution statistics for forming student teams," in *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*, ser. ITiCSE 2024. New York, NY, USA: Association for Computing Machinery, 2024, p. 569–575. [Online]. Available: https://doi.org/10.1145/3649217.3653634

[11] J. Cui, R. Zhang, F. Zhou, R. Li, Y. Song, and E. Gehringer, "How much effort do you need to expend on a technical interview? a study of leetcode problem solving statistics," in *2024 IEEE 36th International Conference on Software Engineering Education and Training (CSEE&T)*, 2024.

[12] I. J. Hoever, D. Van Knippenberg, W. P. Van Ginkel, and H. G. Barkema, "Fostering team creativity: perspective taking as key to unlocking diversity's potential." *Journal of applied psychology*, vol. 97, no. 5, p. 982, 2012.

[13] A. W. Woolley, C. F. Chabris, A. Pentland, N. Hashmi, and T. W. Malone, "Evidence for a collective intelligence factor in the performance of human groups," *science*, vol. 330, no. 6004, pp. 686–688, 2010.

[14] A. K.-y. Leung, W. W. Maddux, A. D. Galinsky, and C.-y. Chiu, "Multicultural experience enhances creativity: the when and how." *American psychologist*, vol. 63, no. 3, p. 169, 2008.

[15] R. Xi and M. P. Singh, "The blame game: Understanding blame assignment in social media," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 2, pp. 2267–2276, 2024.

[16] R. Duran, L. Haaranen, and A. Hellas, "Gender differences in introductory programming: comparing moocs and local courses," in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 2020, pp. 692–698.

[17] C. Wilcox and A. Lionelle, "Quantifying the benefits of prior programming experience in an introductory computer science course," in *Proceedings of the 49th acm technical symposium on computer science education*, 2018, pp. 80–85.

[18] M. Babes-Vroman, I. Juniewicz, B. Lucarelli, N. Fox, T. Nguyen, A. Tjang, G. Haldeman, A. Mehta, and R. Chokshi, "Exploring gender diversity in cs at a large public r1 research university," ser. SIGCSE '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 51–56. [Online]. Available: https://doi.org/10.1145/3017680.3017773

[19] B. Ioannis and K. Maria, "Gender and student course preferences and course performance in computer science departments: A case study," *Education and Information Technologies*, vol. 24, no. 2, pp. 1269–1291, 2019.

[20] I. Wagner, "Gender and performance in computer science," *ACM Transactions on Computing Education (TOCE)*, vol. 16, no. 3, pp. 1–16, 2016.

[21] P. Byrne and G. Lyons, "The effect of student attributes on success in programming," in *Proceedings of the 6th annual conference on Innovation and technology in computer science education*, 2001, pp. 49–52.

[22] L. A. Meadows and D. Sekaquaptewa, "The influence of gender stereotypes on role adoption in student teams," in *2013 ASEE Annual Conference & Exposition*, 2013, pp. 23–1217.

[23] ——, "The effect of skewed gender composition on student participation in undergraduate engineering project teams," in *2011 ASEE Annual Conference & Exposition*, 2011, pp. 22–1449.

[24] E. A. Strehl and R. Fowler, "Experimental evidence regarding gendered task allocation on teams," in *2019 ASEE Annual Conference & Exposition*, 2019.

[25] R. C. Martin, *Agile software development: principles, patterns, and practices*. Prentice Hall PTR, 2003.

[26] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. USA: Addison-Wesley Longman Publishing Co., Inc., 1995.

[27] GitHub, "The top programming languages," 2022. [Online]. Available: https://octoverse.github.com/2022/top-programming-languages