An Analysis of Introductory CoursesAffect on Student Sentiment andStereotype Toward Computer Science

Lara McConnaughey

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2019-96
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-96.html

May 24, 2019

Acknowledgement

Thank you to Professor John Denero and Chris Hunn for getting me access to the data. Thank you to Mansi Shah and Alex Stennet for tips on describing CS10 and CS61a. Thank you to Alexander Baños for proofreading. Thank you to Maxson Yang for tips on stereotypical viewpoints of computer science. Thank you to my dad and mom, Marc McConnaughey and Meilee McConnaughey, for talking through my ideas with me and giving me feedback on my writing. Thank you to Yifat Amir for giving me feedback on structure and content. Thank you to Professor David Bamman for providing invaluable feedback as the second reader for my thesis. And a big thank-you to my advisor, Professor Dan Garcia, for all the inspiration and support he has provided me through my undergraduate and graduate career.

UNIVERSITY OF CALIFORNIA, BERKELEY

MASTER PROJECT REPORT

An Analysis of Introductory Courses Affect on Student Sentiment and Stereotype Toward Computer Science

---

Author:
Lara MCCONNAUGHEY

Supervisor:
Dr. Dan GARCIA

A project submitted in fulfillment of the requirements for the degree of Master of Science

in the

Department of Electrical Engineering and Computer Sciences

May 24, 2019

Abstract

Computer science has been stigmatized as a prestigious field that requires extraordinary intelligence in order to participate. If a student doesn't perceive themselves as fitting this stigma, they are less likely to pursue the field. Quantifying perception can allow for the improvement of introductory courses and provide recommendations for curriculum. The following paper outlines methods for quantifying perception of computer science and looks at answering the following research questions:

1. How does a student's sentiment towards and stereotype of computer science change after taking a computer science class?
2. How do these changes interact with the type of class they take, their gender, and whether or not a student decides to continue learning computer science?

Contents

1 Introduction

"I think all adjectives can describe a computer scientist because every one has the potential to be a computer scientist“

— CS10 student's response when asked to describe a Computer Scientist

“smart, great at math and science, most of them graduated from or currently attends an ivy league or prestigious university, started programming as a child, passionate about their work“

— CS61A student's response when asked to describe a Computer Scientist

1.1 Motivation

Computer science has been stigmatized as a prestigious field that requires extraordinary intelligence in order to participate. If a student doesn't perceive themselves as fitting this stigma, they are less likely to pursue the field, even if they have an apparent aptitude for it [2]. Diversity in computer science has become a growing concern and is the heart of initiatives like “CS for All“, “Code.org“, and “Girls who Code“, among others. Computer science is an ever expanding field with artificial intelligence reaching into domains that affect billions. In order to build solutions for a diverse audience, a diverse set of voices are needed. Transforming the perspective of computer science from the stigma exemplified by the CS61A student's quote above to the CS10 student's view that anyone can be a computer scientist is a major step in diversifying computing.

Institutions and computer science instructors can play a role in fostering inclusive environments and curriculum. This motivates the analysis of UC Berkeley's own two introductory computer science courses and how these courses affect students' perceptions of computer science. Quantifying perception can allow for the improvement of introductory courses and provide recommendations for curriculum. The research questions for this study are as follows:

1. How does a student's sentiment towards and stereotype of computer science change after taking a computer science class?
2. How do these changes interact with the type of class they take, their gender, and whether or not a student decides to continue learning computer science?

1.2 CS10: The Beauty and Joy of Computing

The Beauty and Joy of Computing (BJC) is an introductory computer science course for students with minimal programming experience. The course aims to prepare students for future computer science courses and develop students' programming abilities for application to their field of study. The philosophy behind the course is to build a curriculum that is accessible for anyone by meeting the students where they are and leaving no students behind [9]. BJC became the first computer science course in UC Berkeley's history to have more women than men, breaking this record in 2014 with an enrollment of 106 women and 104 men [1]. Since 2014, the course has consistently had more women enrolled than men, even reaching a 65% female enrollment in Spring 2018. The course also serves as a pipeline, bringing many underrepresented students into the computer science major at UC Berkeley.

The first eight weeks of the course are taught using Snap*!*, a visual, block-based, drag and drop programming language. In the final six weeks, the curriculum shifts to using Python. The course focuses on the "Big Ideas" of computing, such as abstraction, recursion, and algorithmic complexity. Throughout the course, there are non-technical lectures that give overviews of social implication topics (Privacy, Education, Computing and the Environment, History of Computing) and computer science research areas (HCI, AI, Algorithmic Bias). These lectures are included to not only engage students of various disciplines, but also to expose them to the relevance of computer science to the their daily lives. Another way the course has students relate to computer science is through the course assignments. For the midterm and final project, the students are given the opportunity to code whatever they like. Additionally, there is one written assignment in the class called the "Explore Post" where students write a short essay on the social implications of a computational innovation of their choosing.

After taking BJC, students have a foundational computer science knowledge that they can apply to pursuing a computer science degree or to their major field of study. The course doesn't fulfill any requirements for the Computer Science major or any other major on campus, however it does fulfill the quantitative reasoning requirement for the College of L&S [1]. Additionally, as an introductory course, BJC does not prepare students to immediately pursue software engineering roles in industry.

1.3 CS61A: Structure and Interpretation of Computer Programs

The Structure and Interpretation of Computer Programs, commonly referred to by its course number CS61A, is an introductory programming and computer science course for students with various programming experience. The course follows *Composing Programs*[2] by John Denero, and focuses on abstraction techniques to manage program complexity. CS61A covers four main broad topics: functional abstraction, data abstraction, state and assignment, and meta-linguistic abstraction. The course uses Python as its main language, but also has assignments in SQL and Scheme. There are four major programming projects in the course, ranging from implementing strategies for a basic dice game, to writing an interpreter for the Scheme language in Python.

At UC Berkeley there are two pathways a student can take to earn a degree in Computer Science:

- College of Engineering (COE) Electrical Engineering and Computer Science (EECS) major: students are pre-admitted to the major before coming to UC Berkeley based on performance in High School.
- College of Letters and Science (L&S) Computer Science (CS) major: students can declare this major after meeting a 3.3 average GPA in CS61A, CS61B (Data Structure), and CS70 (Discrete Mathematics and Probability Theory).

CS61A is the first class taken when following both the COE and L&S computer science major pathways. After taking CS61A, students are prepared to take the next course in the series CS61b (Data Structures) and to pursue some entry level software engineering internships in industry.

Topic	Class	
	CS10	**CS61A**
Abstraction	✓	✓
Functions and Procedures	✓	✓
Variables and Conditionals	✓	✓
Higher Order Functions	✓	✓
Lists	✓	✓
Algorithms and Complexity	✓	✓
Recursion	✓	✓
Lambda Functions	✓	✓
Programming Paradigms	✓	✓
Social Implications	✓	✓
HCI	✓	
History of Computing	✓	
AI	✓	
Concurrency	✓	
Interfaces		✓
Object-Oriented Programming		✓
Declarative Programming		✓
Classes		✓
Streams and Linked Lists		✓
Language Abstraction		✓
Generic Operators		✓

Table 1. List of topics taught in CS10 and CS61A

1.4 Comparing CS10 and CS61A

The instruction of CS10 and CS61A are structured similarly. Both classes consist of lecture, lab, and discussion sections. Lectures are taught by the instructor of record and are webcasted, so students are not required to attend. This is particularly beneficial for CS61A because the class size typically ranges from 1,000-1,800 students and UC Berkeley's largest lecture hall has a capacity of 720. CS10's class size typically ranges from 150-350 students. Labs and discussions are taught by graduate and undergraduate teaching assistants. Lab sections consist of students reading through and working on programming exercises either alone or with a partner. Discussion sections consist of a teaching assistant lecturing in front of students, solidifying student understanding. Lab and Discussion sections usually have 15-35 students in attendance.

The learning outcomes for CS10 and CS61A are very different. CS10 is a survey course where students get a birds-eye view of the field of computer science and

[1] https://ls.berkeley.edu/quantitative-reasoning
[2] http://composingprograms.com/

develop foundational computational thinking skills. CS61A is a technical course that teaches students to manage program complexity using abstraction techniques. Students can easily pick up new languages after taking the course because of their deep understanding of the concepts involved in programming. A list comparing the topics taught in both courses can be found above in Table 1.

1.5 Related Works

The stereotypical image of computer science has been identified as a barrier to diversifying the field [21]. To understand why this is the case, it is important to define the stereotype and current perception of students. Surveys of young undergraduate students asking them to describe computer science found that the descriptors generalized into these main categories: technology-oriented, singularly focused on computers, lacking interpersonal skills, masculine, physically characterized by wearing glasses, being pale, skinny and unattractive [6]. Additionally, a multi-year research project by Google and Gallup found similar results when studying the perception of computer science of 7th to 12th grade students, parents of 7th to 12th graders, and 1st to 12th grade teachers and school faculty. This study found computer science is perceived as a male dominated field with more than 60% of students, parents, and teachers responding that boys are more interested in computer science than girls and more than a third of each group saying that boys are more likely to be successful in computer science. Moreover, more than half of each group thinks a person needs to be very smart and good at STEM to learn computer science.[10]

It's worth noting that in both studies, there was not a significant difference in the way that men and women described computer science. This may be due to the fact that the computer science stereotype is perpetrated through popular culture and media, which both groups have equal exposure to. Research has shown that a student's sense of belonging in a field influences their interest, persistence and performance [5] [20]. Although the gender disparity in STEM fields is a widely highlighted issue, unbalanced interest from men and women exist in many other fields [7]. English as a major in universities is female dominated, which is one reason why men are less likely to align their interests with the major [3]. This is the same with computer science. The computer science stereotype is incongruent with the normative female gender role and carries a message of who does and doesn't belong in the field [8] [16]. A study of students at two large universities found five factors that influence student's decision to major in Computer Science: ability (expectation of success), fit (identity alignment), enjoyment, utility (value to themselves and society), and opportunity cost [14]. Stereotype can affect a students' view of how well they will succeed, their fit with computer science and the utility of computer science.

In media, one is most likely to see "Whites" and "people wearing glasses" followed by "Asians" portrayed as computer scientist [10]. However, studies have shown that changing perception of computer science is possible and can impact the intent of students to take computer science courses. In a study conducted at Stanford and University of Washington, students' stereotypical perceptions of computer science were altered by reading print news articles. The articles were titled "Study finds computer science no longer dominated by 'geeks'" and "Study finds computer science continues to be dominated by 'geeks'." Students read one of the two articles and then their interest in majoring computer science was measured. The study found that women, but not men, were more likely to consider majoring in computer science after reading the article claiming computer scientists no longer fit the geeky stereotype [6].

Another study found that classroom environments also affect interest in enrolling in computer science. Female students were less likely to show interest in enrolling in a computer science course after seeing a classroom that stereotypically reflects computer science. In the study, stereotype was exhibited by electronics, video games, science fiction books and posters. Male and Female students were equally interested in computing when the classroom only included neutral and non-stereotypical objects [15].

One intervention recommends teaching students that the computer science stereotypes are constructed rather than required attributes for participation in the field. In a study interviewing students about computer science stereotypes, some students were enrolled in a seminar that deconstructs the computer science stereotype by highlighting the creative side of the field, visiting local companies, and attending research talks. One student who took this seminar was asked if the seminar influenced her decision in majoring in computer science. She responded, "I think that has confirmed it (majoring in computer science) a lot because I have seen people like myself in that class who are interested in, but not completely sure they want to do this. And again, they are women" [13].

In Semmen et al's study titled *"Who Are You? We Really Wanna Know... Especially If You Think You're Like a Computer Scientist"*, perception of computer science by girls (age 15-17) before and after an eight week computer science program was measured [19]. The girls were asked to describe themselves and computer science in surveys administered in the beginning and end of the program. The researchers measured the sentiment and the stereotype of the responses and found, that after exposure to computer science, the girls described computer science more positively and less stereotypically. Additionally, their descriptions of themselves and computer science overlapped more [19].

The research described in the following sections takes inspiration from this study's recommendation to apply these measures to other programs and hypothesized that stereotype and sentiment measurements will differ between required and non-required classes. The methods presented are also similar to the methods of measuring sentiment and stereotype to analyze student perception in Semmen et al.'s study.

2 Data

2.1 Overview

The data for this study comes from surveys administered at the beginning and end of CS10 and CS61A during the Fall 2018 semester. The start-of-semester survey was given during the first week of instruction. The end-of-semester survey was given during the last week of instruction, two weeks before the final exams for each course. In Fall 2018 CS10 was taught by Professor Dan Garcia and CS61A was taught by Professor John Denero. Both Professors are distinguished teachers at UC Berkeley and have won multiple departmental teaching awards. 237 students completed CS10 in Fall 2018. There were 213 responses to the start-of-semester survey and 211 responses to the end-of-semester survey. 1767 students completed CS61A in Fall 2018. Across both the start-of-semester and end-of-semester surveys, there were 1724 respondents. To elicit perception of computer science, the following prompt was included in all of the surveys:

List all the adjectives or phrases you can think of to describe a computer scientist, such as "athletic," "creative," or "likes math".

This prompt was identical in the start-of-semester and end-of-semester surveys. However, the content of the surveys differed from the start-of-semester to the end-of-semester and from CS10 to CS61A. The start-of-semester surveys included questions that are geared towards allowing the instructor and teaching assistants to get an overview of the demographics for the students in the class. The end-of-semester surveys were used by the courses to gather feedback about the students' experiences during the semester. Responding to the above prompt was optional for both students in CS10 and CS61A.

2.2 Class Demographics

The following section visualizes the demographics of student survey responses. Survey questions were not mandatory, so responses to some demographic questions were left empty. The graphs below exclude empty responses as a response category. Additionally, students who did not answer the computer science descriptor prompt in the start-of-semester and end-of-semester survey were excluded from this study, thus are also excluded in the demographics summarized below.

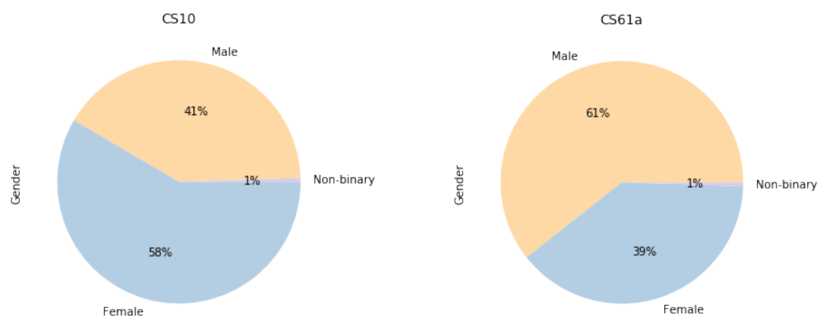


FIGURE 2.1: Students by gender in CS10 and CS61A

Students were ask to identify their gender. In CS10 (N=177) there were 104 female students, 74 male students, and 1 non-binary student. In CS61A (N=574) there were 223 female, 348 male , and 3 non-Binary students.

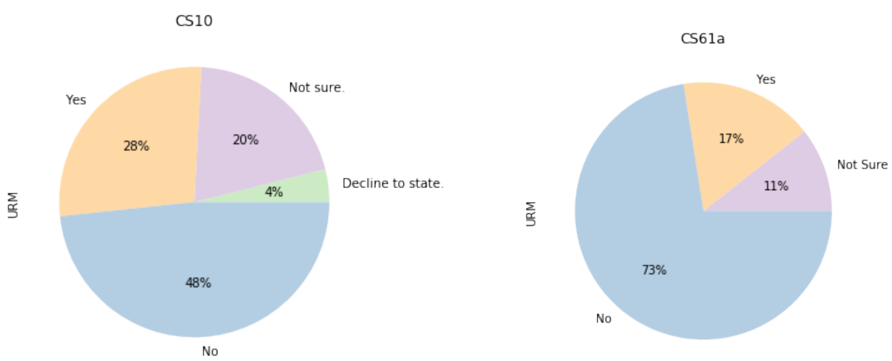


FIGURE 2.2: CS10 and CS61A responses to: Do you consider yourself to be a member of an underrepresented ethnic or racial minority within computer science courses?

Students were ask to identify if they consider themselves an Underrepresented Minority (URM) in computer science. In CS10 (N=178) the responses were as follows: 49 Yes, 86 No, 36 Not Sure, and 7 Declined to State. In CS61A (N=178) the responses were as follows: 102 Yes, 438 No, and 64 Not Sure.

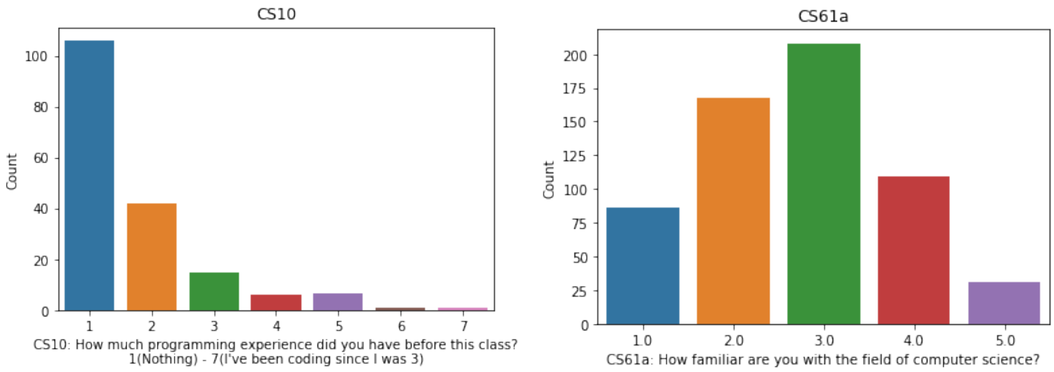


FIGURE 2.3: CS10 (N=178) and CS61A (N=602) survey responses to computer science experience and familiarity

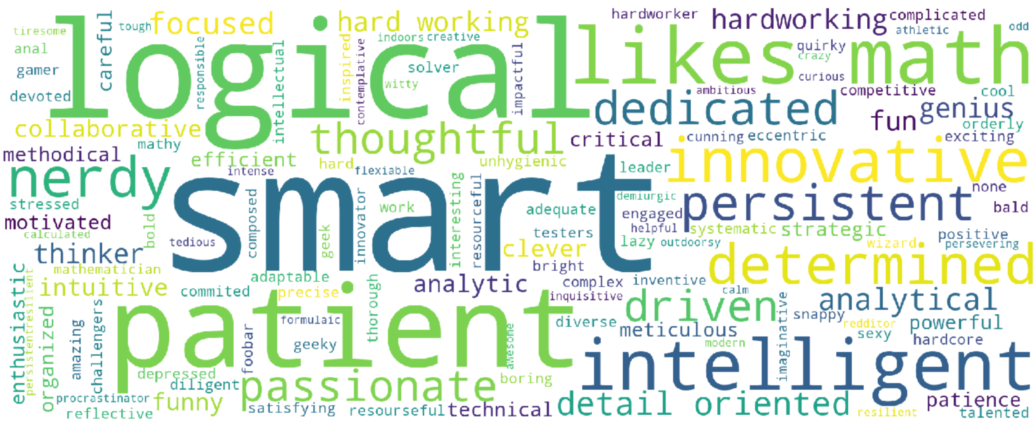
Students were asked about their familiarity and experience with computer science. As shown in Figure 2.3, CS10 students did not have much experience with computer science prior to taking the class. 60% of the 178 students responded that they had no prior experience with computer science before taking CS10. CS61A students were fairly familiar with computer science before taking the course with 57.5% of students reporting their familiarity as $\geq$3 on a scale of 1-5.

2.3 Computer Science Descriptors

The words and phrases used in response to the computer science perception prompt were parsed by common delimiters such as commas, semi-colons, newlines, quotation marks and back slashes. Punctuation was removed, all words were made lowercase, and stripped of leading and trailing spaces. Spelling of common words were also regularized. For example, some students wrote “problemsolver" as one word; this was changed to be two words, “problem solver." Words and phrases were not regularized by their meaning. For example, ”logical", ”logical thinker", ”logical thinking", and ”logically oriented" were all responses given by students that all have similar meaning, but were left as is. Table 2.1 shows the number of unique words and phrases used by students.

Class	Total Words	Start Words	End Words
CS10	424	318	171
CS61A	1427	850	880

TABLE 2.1: Number of unique words and phrases used to describe computer science in the start-of-semester and end-of-semester.



In answering the start-of-semester prompt, CS10 students ($\bar{x} = 4.3$, $\sigma = 2.3$) used, on average, the same number of words as CS61A students ($\bar{x} = 4.6$, $\sigma = 1$). However, the number of words used in the end-of-semester prompt by CS10 students ($\bar{x} = 2.56$, $\sigma = 1.3$) was on average less than CS61A students ($\bar{x} = 4.7$, $\sigma = 0.9$).

Most students answered in phrases and words, however, some students answered the prompt with sentences. There were instances where students wrote sentences reflecting on the prompt. For example, a student wrote in the start-of-semester survey, "geek, white male, sits alone and codes all day, not athletic, not social (these are stereotypes of a computer scientists and or not what i believe a computer scientists is)." The student recognizes that there are stereotypes associated with computer science, but explains that the words listed doesn't reflect their perception. However, in their end of semester survey the same student listed the following adjectives: "impact, influence, systematic, diverse, hegemonic", which seems like their perception of computer science solidified after taking the course because they were able to list specific adjectives.

CS10 start-of-semester		**CS10 end-of-semester**		**CS61A start-of-semester**		**CS61A end-of-semester**	
creative	10% (76)	creative	20.3% (87)	likes math	5.8% (166)	smart	7.9% (176)
smart	5.6% (42)	smart	8.1% (35)	smart	5.6% (158)	creative	7.4% (165)
logical	3.8% (28)	logical	4.2% (18)	logical	4.6% (130)	logical	6.4% (144)
problem-solver	3.5% (26)	patient	3.7% (16)	creative	4.6% (130)	likes math	3.8% (85)
intelligent	3.2% (24)	likes math	3.2% (14)	intelligent	3% (86)	intelligent	3.3% (78)
innovative	3.1% (23)	intelligent	2.1% (9)	hardworking	2.4% (67)	nerdy	3.4% (76)

TABLE 2.2: Percentages and counts of the top 6 words used to describe computer science in the start and end semester surveys of CS10 (Start N=735, End N=428) and CS61A (Start N=2826, End N=2218). Percentages based on number of times word is used over total number of words used.

3 Analysis

Inspired by previous work [19], two facets of perception of computer science were examined. The first looks at measuring if a student's attitude towards computer science is positive, negative or neutral. This kind of measurement is commonly known as sentiment analysis. The second looks at measuring how closely a student's viewpoint of computer science aligns with the over-generalized image of computer scientists. There is no standard practice or name for performing this analysis, but it will be referred to here as stereotype analysis. The next two sections describe the algorithms implemented for sentiment and stereotype analysis. The following sections analyze the sentiment and stereotype as it interacts with various demographic and environmental factors such as the gender of the student, the gender of the student's teaching assistant (TA), and whether the student moves on to take another computer science class.

3.1 Sentiment

Sentiment analysis is a deeply studied problem in the Machine Learning sub-field of Natural Language Processing. It is a text categorization task, which aims to classify an author's attitude about a topic as having positive (1), negative (-1), or neutral (0) sentiment. Sentiment analysis has been applied to a variety of domains such as Twitter Tweets, Amazon Product Reviews, and Movie reviews. In turn, the the sentiment of these domains can be used to inform fields like politics and marketing. In this study, sentiment analysis is applied to analyze the student's sentiment towards computer science. Often times, while performing sentiment analysis, the object on which the author is expressing their viewpoint must be identified, however, that is not necessary in this study because the words and phrases being classified were prompted descriptors of computer science.

3.1.1 Algorithm

To perform sentiment analysis, the pre-trained VADER (Valence Aware Dictionary for sEntiment Reasoning) [12] was used. VADER is a parsimonious rule-based model that combines lexical features with rules for syntactic and grammatical conventions to predict sentiment. The model is trained on twitter data and performs well in micro-blog contexts. This matches the data in this study because the students were asked to provide short answers to the prompt. When bench-marked against other state of the art sentiment analyzers, VADER performs just as well or better [12].

Taken from Semmens's "Who Are You?" paper [19], the sentiment for a students response was computed as follows:

$$S(W) = \frac{\sum_{(p \in W)} \delta(p)}{|W|}$$

$S(W)$ is the average sentiment computed for the segmented words and phrases in a student's response W. $\delta(p)$ is the sentiment predicted by VADER for a given word or phrase p. $|W|$ is the number of words and phrases in the student's response. The VADER model outputs a compound score for the text being predicted on that is the normalized valence composite score. The compound score takes the valence scores of each word in the text, adjusts it with the learned rules, and normalizes the score between -1 (negative sentiment) and +1 (positive sentiment) [12]. This compound score was used for $\delta(p)$ and is averaged to get the sentiment for a student's response. The sentiment averages computed with VADER yielded expected scores for the responses given by students as shown in Table 3.1 below. For example, descriptions with words like "geeky" and "sad" receive a negative sentiment score, while descriptions with words like "creative" and innovative receive a positive sentiment score.

Words & Phrases	S(W)
small, geeky, no social life, sad, lonely	-0.2574
nerd, lame, not cool, boring, one dimensional	-0.25536
nerdy, unattractive, rigid, boring, obese	-0.18764
glasses, dog person, white, male, asian	0
creative, innovative, ambitious	0.4525
loves logic, loves math, loves animals, loves people, loves nature	0.571900

TABLE 3.1: Example sentiment scores for words and phrases from student responses.

3.1.2 Discussion

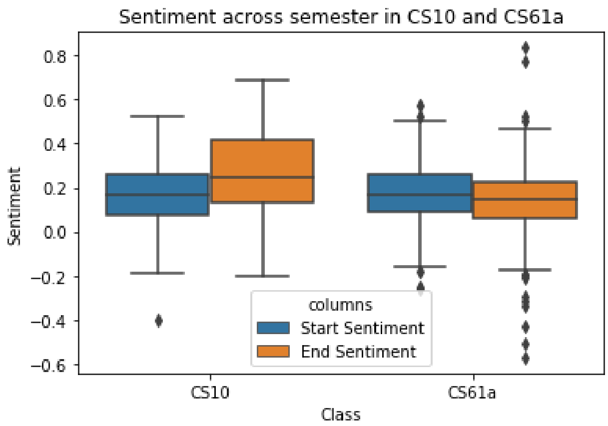


FIGURE 3.1: Average sentiment calculations across start-of-semester and end-of-semester for CS10 and CS61A.

Figure 3.1 shows the change in sentiment between the start of the semester and the end of the semester for students taking CS10 and CS61A. A t-test on the sentiment in the beginning of the semester versus the end of the semester reveals significant differences in student sentiment. In CS10, the descriptions of computer science became significantly more positive from start ($\bar{x} = .168$, $\sigma = 0.139$) to end ($\bar{x} = .246$, $\sigma = 0.164$) after the class (p < .0001). In CS61A, the descriptions of computer science became significantly less positive from start ($\bar{x} = .171$, $\sigma = 0.127$) to end ($\bar{x} = .139$, $\sigma = 0.141$) after the class (p < .0001). From this we can conclude that exposure to computer science material changes students' sentiment of computer science. However, sentiment can either change positively or negatively, as shown by CS10 students' sentiment increasing and CS61A students' sentiment decreasing.

As previously described, CS10 teaches about non-technical aspects of computer science while CS61A is a rigorous technical introduction to the field. The dissimilarity in course content could account for the different directional change in sentiment. Experience with other students and teaching assistants in the class could also be a factor of sentiment change. Poor experiences can result in decreasing sentiment, while positive experiences can result in an increase. To comprehensively test this, future work could interview students about their experiences in each course and correlate it with their calculated sentiment. Table 3.2 gives a comprehensive overview of the sentiment calculations. General sentiment is a good indicator of perception of computer science, but doesn't give a full picture of how the computer science stereotype may be a barrier to entry for groups that don't fit it. For example, Table 3.1 shows the response, "glasses, dog person, white, male, asian" is given a sentiment of 0. However, this description is highly stereotypical of the computer science field.

	count	mean	std	25%	50%	75%	min	max
CS10								
Start-of-Semester	170	0.168	0.139	0.072	0.170	0.257	-0.401	0.524
End-of-Semester	170	0.246	0.164	0.134	0.249	0.416	-0.202	0.690
CS61A								
Start-of-Semester	605	0.171	0.127	0.088	0.170	0.257	-0.257	0.571
End-of-Semester	605	0.139	0.141	0.060	0.150	0.224	-0.571	0.836

TABLE 3.2: Summary of sentiment calculations for CS10 and CS61A.

3.2 Stereotype

Previous research has shown that the stereotypical male representation of computer scientists portrayed in media deters women from the field of computer science. [6] Quantifying how strong someone's view of computer science aligns with this stereotype can help in the analysis of how to break these barriers. Unlike sentiment analysis, stereotype analysis is not a studied problem. Semmens's method, outlined in his paper, is inspired by sentiment analysis and uses a similar formula to calculate an average stereotype rating. The ratings will also be categorical: stereotypical (1), non-stereotypical (-1), and neutral (0).

3.2.1 Algorithm

The top 100 most used terms across CS10 and CS61A responses were selected and hand-classified, by myself and a CS10 teaching assistant, as one of the categories previously described [1]. Words were classified based on the two annotators own contemporary view of stereotype, which was informed by the categories that were found to generalize computer science descriptions by undergraduates [6]. These top 100 words accounted for 70.1% of words and phrases used by CS10 students and 64.8% of words and phrases used by CS61A students.

Stereotypical	Non-Stereotypical	Neutral
smart	creative	curious
logical	patient	focused
likes math	passionate	dedicated
intelligent	collaborative	organized
nerdy	interesting	practical

TABLE 3.3: Most common stereotypical, non-stereotypical, and neutral labeled words used by students.

[1] A list of the annotated words are included in the Appendix.

For words that were out of the annotated stereotype vocabulary, categorization was propagated through word similarity using word embeddings. Word embeddings are vectorized representations of words and phrases in a multi-dimensional space that represents semantics. Models are trained to generate word vectors so that words that have similar meanings and distributions have high similarity.[17] Google's pretrained word2vec [2] model along with Genism[3], a topic modeling library, were used to vectorize words and compute their similarity.

Average stereotype was computed in the same manner as average sentiment:

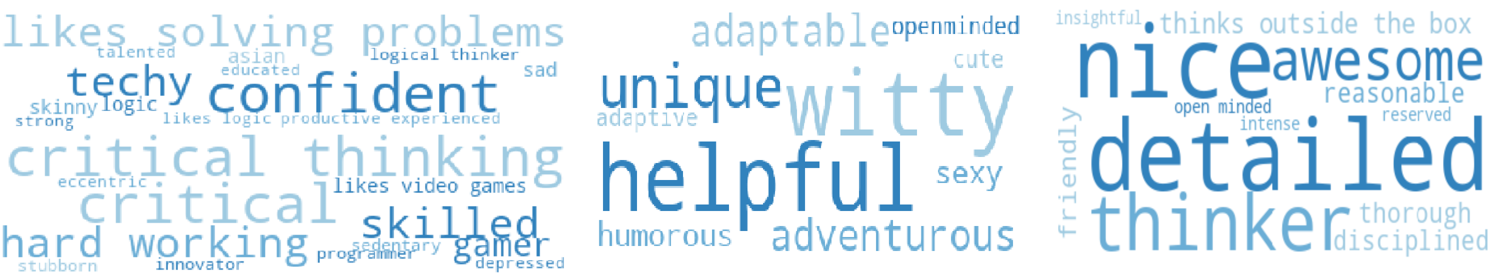
$$T(W) = \frac{\sum_{(p \in W)} \gamma(p)}{|W|}$$

$T(W)$ is the average stereotype computed for the words and phrases in a student's response W. $\gamma(p)$ is the stereotype classification predicted using the following algorithm:

1. If the word is in the stereotype vocabulary return it's categorization.
2. If the word is not in the stereotype vocabulary do the following:
   (a) Tokenize the target phrase.
   (b) Remove stop words from target phrase using NLTK's[4] stop word corpus.
   (c) Find the word in the vocabulary that has the highest similarity with the target phrase:

   $$L(P) = \underset{v \in V}{\text{argmax}} \sum_{(w \in P)} \phi(w, v)$$

   L(P) returns the word in V (our vocabulary) that has the highest total cosine similarity (denoted as $\phi(w, v)$) to each word in P (the target phrase).
   (d) Label the target word or phrase with the stereotype label of the $L(P)$



(A) Stereotypical (B) Non-stereotypical (C) Neutral

FIGURE 3.2: The next 50 most frequent words classified using the stereotype classification described.[5]

[2] https://code.google.com/archive/p/word2vec/
[3] https://radimrehurek.com/gensim/
[4] https://www.nltk.org/book/ch02.html

Figure 3.2 shows the next fifty most frequently used words by students segmented by the stereotype predicted by the model. These words were also hand-annotated. There was a 90% agreement between the model and the annotator. The model generalizes well in capturing words that have similar stems or words in common. For example, the word “logical" is in the annotated vocabulary as stereotypical (1) and the model also predicts "logic" and "logical thinker" as stereotypical.

Words & Phrases	T(W)
creative, flexible, collaborative, careful, passionate	-1
devoted, fun, passionate, fashionable, not afraid of failure	-0.6
athletic, tall, short, man, woman	-0.2
smart, introverted, driven, creative, focused	0.2
enjoys math, quite smart, enjoys coding	1
nerd, geek, introvert, glasses, gamer	1

TABLE 3.4: Example stereotype scores for words and phrases from student responses.

Table 3.4 shows some examples of stereotype scores. The last example in the table is correctly classified as stereotypical even though there were out-of-vocabulary words such as “gamer" and “glasses." The model’s accuracy can be improved if there are more words annotated for the vocabulary. Table 3.5 gives a comprehensive overview of the stereotype calculations.

3.2.2 Discussion

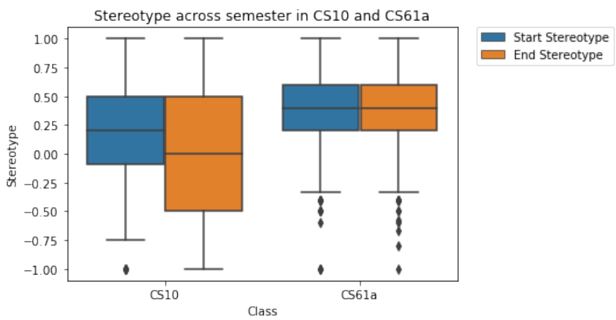


FIGURE 3.3: Average stereotype calculations across start-of-semester and end-of-semester for CS10 and CS61A.

A paired t-test [6] on the stereotype at the beginning of the semester versus the end of the semester reveals significant differences in stereotypical viewpoints for CS10 students, but not for CS61A students. Figure 3.3 shows the change in stereotype score between the start of the semester and the end of the semester for students taking CS10 and CS61A. In CS10, the descriptions of computer science became significantly less stereotypical from start ($\bar{x} = .182$, $\sigma = 0.49$) to end

[6]https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.ttest_rel.html

($\bar{x} = .025$, $\sigma = 0.673$) after taking the class ($p < .01$). In CS61A, the descriptions of computer science, with regard to stereotype, stayed relatively the same ($p = .402$) from start ($\bar{x} = .419$, $\sigma = 0.386$) to end ($\bar{x} = .403$, $\sigma = 0.398$).

From this, we can conclude that the two courses change students' views of computer science differently. Table 3.5 gives a comprehensive overview of the stereotype calculations. CS10 students seem to start with a less stereotypical view of the field than CS61A students. This may be due to the demographics that make up each course.

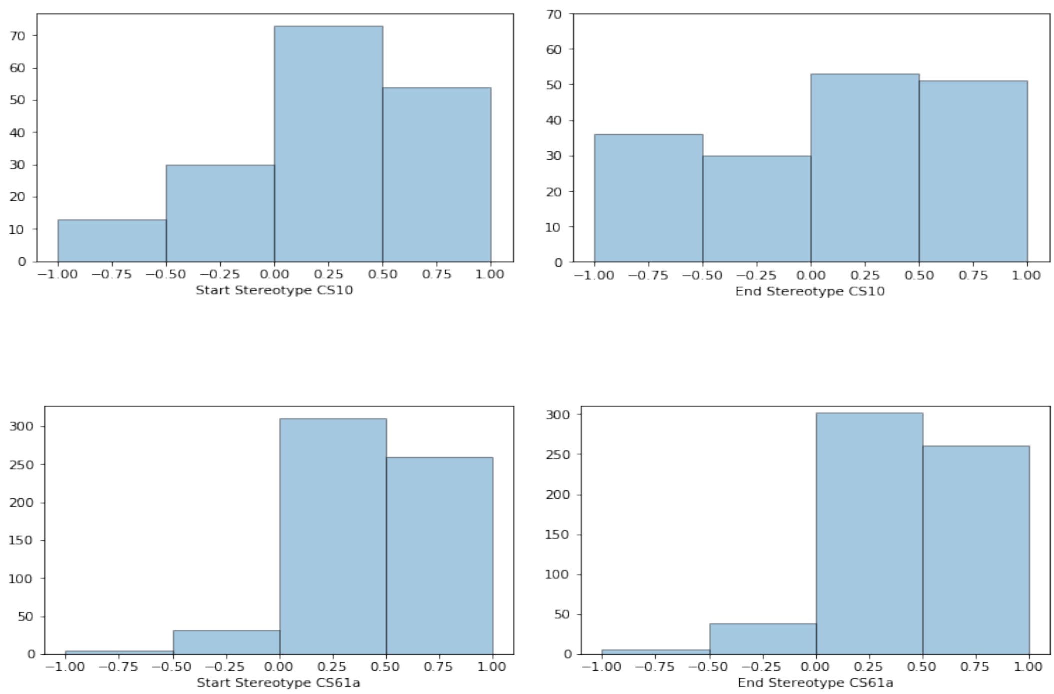


FIGURE 3.4: Histogram of start and end stereotype scores for CS10 and CS61A (Stereotypical=1, Non-Stereotypical=-1)

Although at the end of the semester CS10 students described computer science with fewer stereotypical words, the number of students who described computer science stereotypically remained relatively the same as shown in Figure 3.4. The words that had the most drop in usage were intelligent (-15), innovative (-14), and logical (-10). The words that had the most increase in usage were creative (+11), thoughtful (+4), and thinker (+3). However, these phrases alone don't account for the drop in stereotype score in Figure 3.4. This can be taken to mean that the CS10 course curriculum causes a general shift towards a non-stereotypical view, but there may be other factors in the student's experience that influence their description. Demographic differences and external factors will be explored in the next section.

	count	mean	std	25%	50%	75%	min	max
CS10								
Start-of-Semester	170	0.182	0.492	-0.093	0.200	0.500	-1	1
End-of-Semester	170	0.025	0.673	-0.500	0.000	0.500	-1	1
CS61A								
Start-of-Semester	605	0.419	0.386	0.200	0.400	0.75	-1	1
End-of-Semester	605	0.404	0.398	0.2	0.4	0.6	-1	1

TABLE 3.5: Stereotype Calculations

3.3 Gender and CS Perception

It's been shown that girls, more so than boys, are less likely to enroll in a computer science course if the environment of the course reflects the computer science stereotype [15]. This could be one reason that CS10 has a much higher female enrollment than CS61A. CS10's curriculum includes topics that show the breathe of application of computer science. This section will explore the differences in sentiment and stereotype scores across gender. Although one student in CS10 and three students in CS61A identified as having a non-binary gender identity, they will be excluded from analysis because the sample size is not large enough to draw any accurate conclusions.

Gender in relation to sentiment and stereotype change was analyzed using one-way ANOVA to determine statistical significance, followed by a post-hoc Tukey HSD test [7]. Change was calculated by taking a student's start-of-semester score and subtracting it from their end-of-semester score. The null hypothesis is that the change in stereotype and sentiment does not differ between males and females. The significance threshold is set to $\alpha = 0.1$ throughout each measurement.

- **CS10 Sentiment**: No significant effect of gender on change in sentiment $F(1, 169) = 0.78$, $p > 0.1$. The mean difference from females to males is 0.02.
- **CS61A Sentiment**: No significant effect of gender on change in sentiment $F(1, 571) = 1.90$, $p > 0.1$. The mean difference from females to males is -0.02.
- **CS10 Stereotype**: There is a significant effect of gender on change in stereotype $F(1, 169) = 3.91$, $p < 0.05$. Females finish the course with a less stereotypical view than males, with a mean difference between the groups of -0.22.

[7] The statsmodels library was used for statistical analysis [18].

- **CS61A Stereotype**: There is a significant effect of gender on change in stereotype $F(1, 571) = 4.45$, $p < 0.05$. Females finish the course with a more stereotypical view than males, with a mean difference between the groups of 0.07.

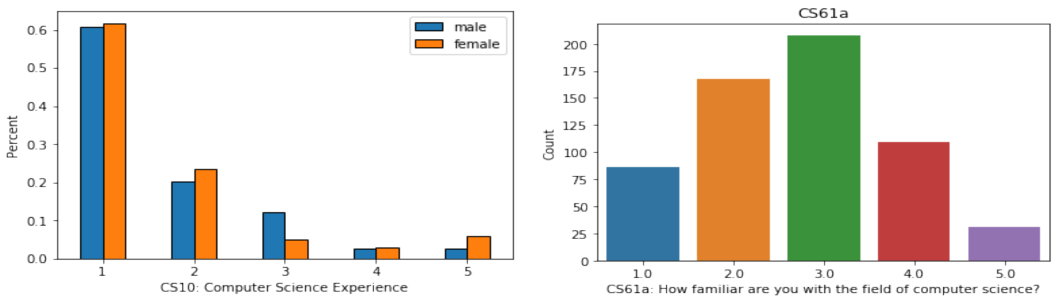


FIGURE 3.5: Students' self-reported computer science experience before taking CS10 or CS61A

From these measurements it can be concluded that change in sentiment doesn't differ by gender, but stereotype does. In CS10, females on average start with a more stereotypical viewpoint of computer science than males, although the distribution of computer science experience between males and females in CS10 is similar, as shown in Figure 1.5. Females, in CS10, on average end with a less stereotypical viewpoint than males. In CS61A, females on average had similar start stereotype scores as males, but females' end stereotype scores were higher on average than males. This means that females in CS61A described computer science as more stereotypical than males. This finding may shed light on perception and retention issues. The CS61A curriculum is more technical, where as the CS10 curriculum focuses more on creative-problem solving and opportunities in the field. It seems females are more affected by their learning environment perpetrating computer science stereotypes than males. Section 3.5 will go into analysis on the relation between sentiment and stereotype and retention of females versus males.

3.4 Gender of TA

Computer Science is a male-dominant field and most college students correctly stereotype that the majority of computer science majors are male. This leads to fewer females entering the computer science major for the same social reasons that English remains a female dominated major [3]. Altering stereotypical viewpoints of computer science can allow for this barrier of inclusion to be broken. Female instructors

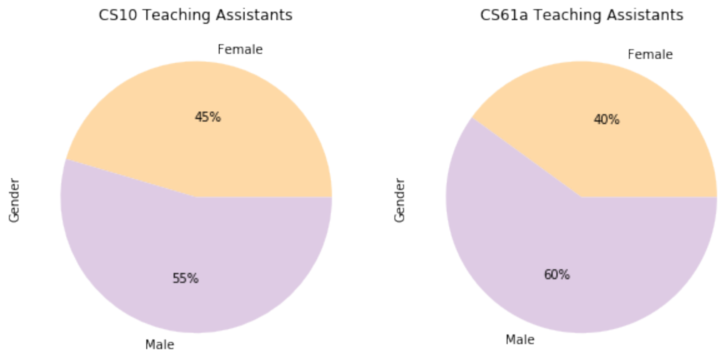


FIGURE 3.6: CS10 Teaching Assistants (N=11) and CS61A Teaching Assistants (N=53) by gender

and teaching assistants break the stereotype that computer science is a completely masculine field, however, previous studies have shown that deviation from stereotypes mainly affects female viewpoints [6]. In this section we will look at the effect that teaching assistant gender has on changing stereotype view.

The end-of-semester survey for CS10 asked students which teaching assistant they interacted with the most. Using a one-way ANOVA to measure significance of gender of TA in CS10 effects on male and female students' ending stereotype produced the following values:

- **CS10 Female Student Stereotype**: Significant effect of gender on ending stereotype for females $F(1, 100) = 3.54$, $p < 0.1$. The mean difference of having a female vs. male teaching assistant is 0.43.
- **CS10 Male Student Stereotype**: Significant effect of gender on ending stereotype for males $F(1, 69) = 3.41$, $p > 0.1$. The mean difference of having a female vs. male teaching assistant is 0.26.

These results indicate that for both males and females having a female teaching assistant will impact their view of computer science, changing it to be less stereotypical. Both instructors of CS10 and CS61A were male, however, future work should analyze how instructor gender impacts change in stereotype.

3.5 URM and CS Perception

This section will explore the relationship between change of sentiment and stereotype scores and a whether or not a student identifies themselves as a underrepresented minority (URM) in computer science. As shown in Table 1.6, 15% of the CS10 and 10% of CS61A students responded that they were unsure if they are an URM in computer science. These students will be omitted from the analysis. Like the previous section URM status in relation to sentiment and stereotype change was analyzed using one-way ANOVA to determine statistical significance, followed by a post-hoc Tukey HSD test.

URM	CS10	CS61A
Yes	47	102
No	82	438
Not Sure	24	64

TABLE 3.6: Shows count of student responses to the question: *Do you consider yourself an underrepresented minority in Computer Science?*

- **CS10 URM Sentiment**: No significant effect of URM on change in sentiment $F(1, 129) = 0.125$, $p > 0.1$. The mean difference of non-URM and URM is 0.012.
- **CS61A URM Sentiment**: No significant effect of URM on change in sentiment $F(1, 540) = 1.8$, $p > 0.1$. The mean difference of non-URM and URM is -0.02.
- **CS10 URM Stereotype**: No significant effect of URM on change in stereotype $F(1, 129) = 0.39$, $p > 0.1$. The mean difference of non-URM and URM is 0.069.
- **CS61A URM Stereotype**: No significant effect of URM on change in stereotype $F(1, 540) = 1.4$, $p > 0.1$. The mean difference of non-URM and URM is -0.05.

Overall, there is no significant relationship between whether or not a student identifies as an underrepresented minority and their change in sentiment or stereotype. Figure 1.6 shows the start-of-semester and end-of-semester stereotype scores segmented by class and URM status. The end-of-semester stereotype distributions for both student that do and don't identify as an URM are similar. The previous section showed a significant relationship between change in stereotype and gender, so it is unexpected that URM status differs and doesn't have a significant relationship with change in stereotype perception. The absence of relationship found in this study may reflect the true absence of relationship amongst this general population

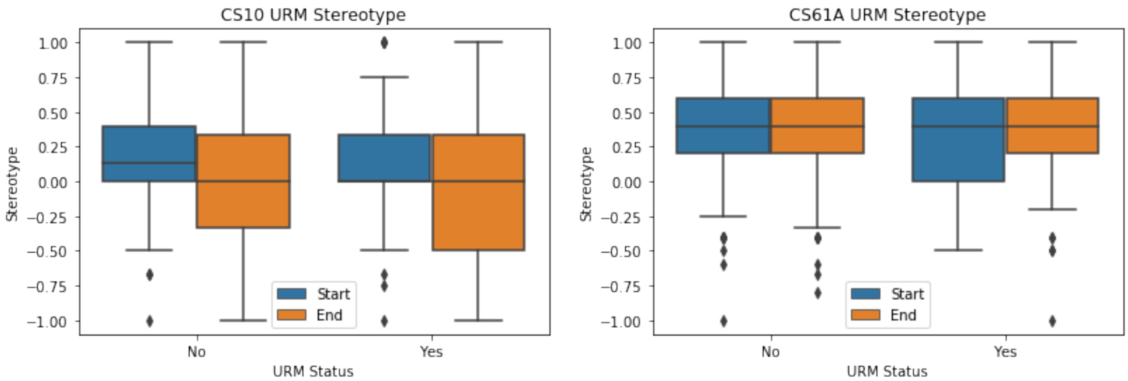


FIGURE 3.7: Stereotype scores of CS10 and CS61A students split by whether they identify as a URM.

or it may be an issue with students' self-identification of URM status in response to the survey question. As shown in table 1.6, many students in both CS10 and CS61a were not sure if they were an URM in computer science. Instead of having students self-identify as an URM in computer science their URM status should be determined based on a regularized standard. To prove there is a true absence of URM status effect on stereotype, this study should be repeated in a different semester or at at different institution with a standardized URM classification. All of these groups are not represented in media as prototypical computer scientists, so future work and better methods of measuring computer science perception will need to be developed to formative feedback on diversifying computer science.

3.6 Retention

The terms CS1 and CS2 are used quite frequently among computer science institutions to describe their introductory courses. In consequence, the terms have different denotations. For the purposes of this study we will use the common connotation that CS1 is an introductory programming course and CS2 is a basic data structures course [11]. This means that CS61A is a CS1 course and CS10 is a CS0 course.

The ACM's 2018 report on "Retention in Computer Science Undergraduate Programs in the U.S." defines retention as a student taking CS1[8] and then subsequently enrolling in CS2[9] [4]. The report finds that retention is a multi-dimensional issue, but

[8] CS1 refers to an introductory programming course
[9] CS2 refers to a basic data structures course

lists interventions that institutions can take to retain students. Amongst these interventions is providing students with a well-rounded education of computer science. This is something that the CS10 curriculum is centered around, which is why CS10 provides a pipeline for underrepresented students to venture into the field. Retention of CS10 students can be defined by whether or not a student enrolled in CS61A. Retention of CS61A students can be defined by whether or not a student enrolled in CS61B UC Berkeley's CS2 course. Since, the computer science stereotype can be a barrier of entry into the discipline, it can be hypothesized that students that come out of the course with a less stereotypical view and a more positive sentiment about computer science are more likely to be retained. The following section will see if this is true for females coming out of CS10 and CS61A.

Logistic regression [10] was used to analyze the effects of stereotype and sentiment on retention. The target variable is a binary indicator of whether or not a student was retained. End-of-semester sentiment and stereotype were used as predictor variables. Figure 3.7 shows the the fitted coefficients for logistic regression models trained on CS10 (N=100) and CS61A (N=223) female sentiment and stereotype.

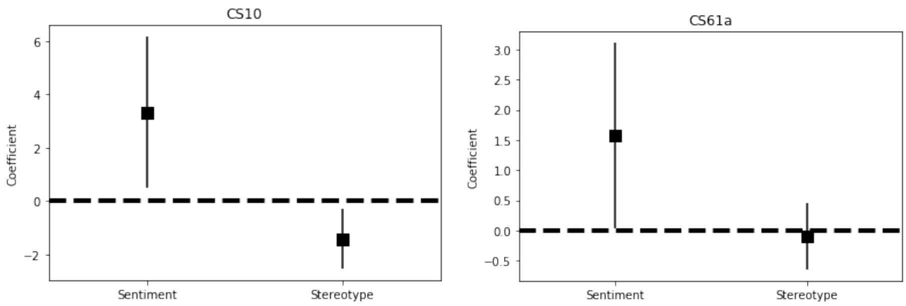


FIGURE 3.8: Dot-and-whisker plot of predictor coefficients with error bars showing confidence intervals.

For female students in CS10 and CS61A, the sentiment has a significant positive relationship with retention ($p < 0.05$), meaning the more positive the end-of-semester sentiment the more likely the student is to be retained. Stereotype has a slightly negative correlation with retention, meaning a less stereotypical viewpoint makes it more likely that a student will be retained. However, the relationship between stereotype and retention was not significant ($p > 0.05$). Some of the responses from female students that were retained in computer science are shown in Table 3.6. In

[10] The statsmodel logistic regression model was used for this analysis [18]

conclusion, female students who have a positive sentiment are more likely to be retained in computer science.

Start-of-Semester	End-of-Semester
likes programming, nerdy, confident, likes business	diverse, creative, collaborative, smart
like math, marketable, abstract, black screen, computer, typing	aspiring, diverse, concise, make magic with my knowledge, powerful
meticulous, problem solver, asian, perseverance, intelligent	designer, engineer, artist, photographer, lazy
high salary	logical thinking, creative, smart, independent, strong communication skill

TABLE 3.7: Responses from female students that moved on from CS10 to CS61A or CS61A to CS61B

4 Conclusion

4.1 Future Work

This study outlines a way to measure computer science perception, but through the analysis the limitation of this studies procedures were revealed. The following section outlines some of the future work that come from the limitations and findings of this study.

The stereotype annotations were highly subjective based on the two annotators perception of stereotype. To curate less subjective annotations methods like crowd sourcing could be used. For example, the words could be put on an online platform like mechanical turk and each words annotation could be the category with the majority vote amongst annotators. Additionally, like mentioned in previous work [19], perception of stereotype may change, so it is necessary to continue to update the stereotype annotations overtime. Tracking perception change overtime can also lead to related research on the factors that influence the change. This study was only carried out at UC Berkeley. To see how different course curriculum changes computer science perception the same measurements outlined in this paper should be applied to other introductory courses at various institutions.

As previously mentioned, retention is a multi-dimensional issue. In this study it was found that if a student has positive sentiment towards computer science, they are more likely to enroll in another computer science course. However, this does not take into account other factors, such as whether or not they intended to major in computer science going into the introductory course. Conducting student interviews from sample of students in CS10 and CS61A who were and weren't retained will give better cues for the factors that increase sentiment and in return cause students to continue pursuing computer science.

Lastly, the same study should be conducted, but instead with the curriculum CS61A changed to reflect CS10's. CS10's curriculum includes guest lectures and computer science exploration tasks that allow students to explore the diversity of the field. If CS61A includes some of these guest lectures or tasks, students could potentially come out of the class with a more positive and less stereotypical view of computer science.

4.2 Closing Summary

The goal of this project was to understand how the perception of computer science is changed by taking a computer science class. Computer Science is a highly stigmatized field and students who don't think they don't fit the stereotypical mold of a computer science often exclude themselves from the field. Quantifying students' perception aids in better understanding perception as it interacts with a student's gender, a student's URM status and weather or not a student decides to continue learning computer science. Additionally, comparing perception of students taking different introductory curriculum can allow for the improvement of introductory courses. This study analyzed CS10, an introductory course for students with no computer science experience, and CS61A, an introductory course for student with varying computer science experience. Natural language processing was used to analyze a student's sentiment towards and stereotypical perception of computer science at the beginning and end of the semester.

This project found that taking CS10 has a significant effect on positively changing the sentiment towards and decreasing the stereotypical view of computer science. In comparison, after taking CS61A students' sentiment towards computer science slightly decreases and their stereotypical view stays relatively the same. Females in CS10 tended to view computer science more stereotypically before taking CS10, but after taking CS10 they viewed computer science less stereotypically than males. In CS61A, males and females started with similar stereotypical perceptions of computer science, but after the course females' views of computer science became more stereotypical than males. For both female students in CS10 and CS61A, a positive sentiment in computer science correlates with whether or not the student moves on to take another computer science course. These findings tell us that computer science courses play a role in how student's perceive computer science. CS10 as a survey course breaks students' stereotypes and increases their sentiment of computer science, which in turn provides a pipeline for diversifying the field. Creating inclusive environments that show computer science as a discipline with applications to many fields is an important step in breaking barriers that prevent diversity in computer science.

5 Appendix

creative	-1
patient	-1
passionate	-1
thoughtful	-1
fun	-1
collaborative	-1
interesting	-1
openminded	-1
funny	-1
motivated	-1
patience	-1
diverse	-1
enthusiastic	-1
athletic	-1
helpful	-1
unique	-1
artistic	-1
imaginative	-1
engaged	-1
visionary	-1
ambitious	-1
flexible	-1
adventurous	-1
wellrounded	-1
cool	-1

smart	1
logical	1
intelligent	1
innovative	1
likes math	1
problem solver	1
nerdy	1
analytical	1
good at math	1
efficient	1
clever	1
genius	1
tech savvy	1
quirky	1
inventive	1
analytic	1
critical thinker	1
likes puzzles	1
technical	1
problem solving	1
complex	1
geeky	1
antisocial	1
math oriented	1
nerd	1
math genius	1
likes maths	1
studious	1
good with numbers	1
snappy	1
geeks	1
successful	1
skilled	1
productive	1
fast at typing	1
logic	1
hardcore	1
quick	1
quick thinker	1
problem solvers	1
likes computers	1
calculated	1
computational	1
calculative	1
techy	1
knowledgeable	1

hardworking	0
persistent	0
determined	0
focused	0
dedicated	0
detail oriented	0
driven	0
diligent	0
resourceful	0
curious	0
thinker	0
careful	0
detailed	0
methodical	0
strategic	0
resilient	0
organized	0
thinks outside the box	0
rational	0
practical	0
meticulous	0
reflective	0
observant	0
inquisitive	0
intuitive	0
complicated	0
trial error	0
insightful	0
thorough	0

TABLE 5.1: 100 most frequent used words by students in CS10 and CS61A annotated with stereotype score.

Acknowledgements

Thank you to Professor John Denero and Chris Hunn for getting me access to the data. Thank you to Mansi Shah and Alex Stennet for tips on describing CS10 and CS61a. Thank you to Alexander Baños for proofreading. Thank you to Maxson Yang for tips on stereotypical viewpoints of computer science. Thank you to my dad and mom, Marc McConnaughey and Meilee McConnaughey, for talking through my ideas with me and giving me feedback on my writing. Thank you to Yifat Amir for giving me feedback on structure and content. Thank you to Professor David Bamman for mentoring me in the field of Natural Language Processing and for providing invaluable feedback as the second reader for my thesis. And a big thank-you to my advisor, Professor Dan Garcia, for all the inspiration and support he has provided me through my undergraduate and graduate career.

Bibliography

[1] Kristen V. Brown. “Tech shift: More women in computer science classes”. In: *SFGate* (2014). URL: http://www.sfgate.com/education/article/Revamped-computer-science-classesattracting-more-5243026.php.

[2] Lori Carter. “Why students with an apparent aptitude for computer science don’t choose to major in computer science”. In: *SIGCSE*. 2006.

[3] Sapna Cheryan and Victoria C. Plaut. “Explaining Underrepresentation: A Theory of Precluded Interest”. In: *Sex roles*. 2010.

[4] Sapna Cheryan and Victoria C. Plaut. “Retention in Computer Science Undergraduate Programs in the U.S.” In: *Sex roles*. 2010.

[5] Sapna Cheryan et al. “Ambient belonging: how stereotypical cues impact gender participation in computer science.” In: *Journal of personality and social psychology* 97 6 (2009), pp. 1045–60.

[6] Sapna Cheryan et al. “The Stereotypical Computer Scientist: Gendered Media Representations as a Barrier to Inclusion for Women”. In: 2013.

[7] Sapna Cheryan et al. “Why are some STEM fields more gender balanced than others?” In: *Psychological bulletin* 143 1 (2017), pp. 1–35.

[8] Allan Fisher and Jane Margolis. “Unlocking the clubhouse: women in computing”. In: *SIGCSE*. 2002.

[9] Daniel D. Garcia, Brian Harvey, and Tiffany Barnes. “The beauty and joy of computing”. In: *Inroads* 6 (2015), pp. 71–79.

[10] Google. “Images of Computer Science: Perceptions Among Students, Parents and Educators in the U.S.” In: (2015).

[11] Matthew Hertz. “What do "CS1" and "CS2" mean?: investigating differences in the early courses”. In: *SIGCSE*. 2010.

[12] Clayton J. Hutto and Eric Gilbert. “VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text”. In: *ICWSM*. 2014.

[13] Colleen M. Lewis, Ruth E. Anderson, and Ken Yasuhara. “"I Don’t Code All Day": Fitting in Computer Science When the Stereotypes Don’t Fit”. In: *ICER*. 2016.

[14] Colleen M. Lewis, Ken Yasuhara, and Ruth E. Anderson. “Deciding to major in computer science: a grounded theory of students’ self-assessment of ability”. In: *ICER*. 2011.

[15] Allison Master, Sapna Cheryan, and Andrew N. Meltzoff. “Whether She Belongs : Stereotypes Undermine Girls ’ Interest and Sense of Belonging in Computer Science”. In: 2015.

[16] Aidan McGowan, Philip Hanna, and Neil Anderson. “Computing gender wars — A new hope”. In: *2017 IEEE Frontiers in Education Conference (FIE)* (2017), pp. 1–8.

[17] Tomas Mikolov et al. “Distributed Representations of Words and Phrases and their Compositionality”. In: *NIPS*. 2013.

[18] Skipper Seabold and Josef Perktold. “Statsmodels: Econometric and Statistical Modeling with Python”. In: 2010.

[19] Rob Semmens, Chris Piech, and Michèlle Friend. “Who Are You? We Really Wanna Know... Especially If You Think You’re Like a Computer Scientist”. In: *GenderIT*. 2015.

[20] Claude M. Steele. “A threat in the air. How stereotypes shape intellectual identity and performance.” In: *The American psychologist* 52 6 (1997), pp. 613–29.

[21] Jennifer Wang, Sepehr Hejazi Moghadam, and Juliet Tiffany-Morales. “Social Perceptions in Computer Science and Implications for Diverse Students”. In: *ICER*. 2017.