Are Introductory CS Courses Effective and Encouraging?

Raefah Wahid

rw2653@barnard.edu

Abstract

Interest in the field of computer science has swelled over the years. The mix of students pursuing computer science degrees is highly diverse, not only in gender and ethnicity, but in experience as well. It is difficult to gauge the true effectiveness of an introductory computer science course when there is a mixture of experienced and inexperienced students taking the class, because the higher grades of the experienced students often skew the average of the class, presenting a misrepresentation of the abilities of students within the class. This study aims to find out how effective and encouraging an introductory CS course can be when taking into account this experience gap. Effectiveness of the course is determined by tracking performance through a pre-test and a post-test while encouragement is determined with the use of a self-efficacy survey taken alongside the tests. Statistical analyses conducted on test performance and the surveys found that there is a positive correlation between self-efficacy and performance. As the course progressed, the experience gap did not close between inexperienced and experienced students and the self-efficacy of inexperienced students did not improve by very much, suggesting the course is neither effective nor encouraging.

1 Introduction

In recent years, the field of computer science has become more popular and accessible, with more and more students enrolling in introductory classes. What often happens is that advanced and well-experienced students enroll in the same courses as inexperienced beginners, creating an experience gap. It is difficult for educators to note how well the class is doing when such a gap exists, which is why experience-based tracking in such situations has become a popular idea (Lacher et al., 2019). By tracking the experience levels of different students throughout the course, it becomes much more clear how well a particular student is doing based on their level of knowledge.

Studies have shown that students with prior experience tend to perform better in computer science courses (Karsten and Roth, 1998), which often intimidates or discourages students with less or no experience studying in the same field. This sometimes has a negative effect on the self-efficacy of inexperienced students, who are more often than not underrepresented minorities who do not have the same access to pre-college computer science programs that other students do (Rauchas et al., 2006).

While many studies have taken note of the role of prior experience in computer science classes, they are moreover focused on experienced students' progress in the class in relation to inexperienced students. Studies certainly take into account the existence of the experience gap, but there is very little discussion focused on how much the experience gap widens or lessens by the end of the course. Studies are more prone to exploring how reliable prior experience in computer science classes is as a measure of student success later on.

The purpose of this study is to gauge how effective and encouraging an introductory computer science course at Columbia University, Introduction to Python (1006), is by using performance on quizzes as a measure of effectiveness and self-efficacy as a measure of encouragement. Since 1006 is a highly diverse class, with an almost even split of experience and inexperienced students, the goal of this study is to see whether the experience gap lessens by the end of the course. Further, the study will compare demographic variables like gender, ethnicity, and

native language against performance and self-efficacy in order to find underlying factors that may be exacerbating the experience gap, such as gender bias or implicit biases in quiz questions. The study will also conduct an item analysis on the quizzes to see if there is a correlation between specific types of questions (e.g. passive and active coding) and experience.

2 Prior Research

In 2017, researchers at the International Conference on Frontiers in Education presented findings about student improvement and success in an introductory computer science coursed based on the factors of aptitude, prior experience, and comfort level (Lacher et al., 2019). Aptitude was determined by use of a computer programming aptitude test, which posed numerical and logical questions or challenges to students. Prior experience and comfort level were determined by the use of surveys. This study found that there was a statistically significant correlation between high grades and students with prior experience or an increased aptitude, and suggested that splitting a computer science classroom into two sections—one advanced and one introductory—based on experience and aptitude would benefit students and increase student retention, especially amongst minorities, who seem to lose interest in computer science or shy away from it after time (ConcannonConcannon and Barrow, 2009).

Another study on computer science education noted that students with prior experience (specifically those with exposure to AP Computer Science A in high school) received scores higher than their counterparts (Alvarado et al., 2018). However, the study noted that there was no statistically significant correlation between prior experience and retention rate; indeed, students appear to persist in computer science courses despite low scores. This particular finding seems to indicate that retention rate is not a good measure of encouragement or self-confidence. This study dealt solely with experience and neglected to include students' self-assessment of their own progress and standing in the class, which might have provided clarity on the unwavering retention rate.

Other studies have analyzed students' self-efficacy in engineering courses and the effect it has on retention rate. One study found that there was no significant difference in self-efficacy across genders or ethnicities, but there was across age (i.e. how long a student had been enrolled in the college), which could be understood as experience, although whether or not this constitutes as experience in a college setting or in computer science alone is unclear (Concannon and Barrow, 2009). This study admits that the population was largely homogeneous, with male and female students having similar skill levels and abilities (measured by grades from high school and scores from college entrance exams); so, the absence of difference across self-efficacy might be explained by the fact that there was no difference in experience across genders. Another study with a more diverse sample of students found that there was a statistically significant correlation between gender and self-efficacy in a computer science setting, with female students' self-efficacy being much lower than their male counterparts (Shull and Weiner, 2002). The same study hypothesized that this might be due to the fact that women have fewer models and mentors in STEM fields.

Very few, if any, studies in this particular field have conducted an item analysis on the sorts of questions asked of computer science students. The general understanding is that experienced students out-perform inexperienced students on average, but there hasn't been any sort of comparison between the two groups on specific types of computer science questions or topics (e.g. tracing, function calls, object oriented programming, recursion, etc.). It is worth it to ask if there are specific concepts that widen the experience gap than others, and if so, then those are the concepts that inexperienced students need further help with in order to reduce the gap.

3 Research Questions and Hypotheses

Based on previous research, the following hypotheses were developed:

(1) By the end of the course, the experience gap will not have closed (i.e. inexperienced students will continue to perform at a worse rate than inexperienced students).
(2) By the end of the course, inexperienced students' self-efficacy will not have improved.
(3) Women and ethnic minorities' self-efficacy will worsen as the course progresses.

The goal of this study is to not only mark and keep track of the experience gap, but to also

explore the many facets of this gap through an item analysis. Is there a clear delineation between concepts that experienced students struggle with versus those inexperienced students struggle with? Are there topics that inexperienced students are able to master quickly and, in terms of performance, are on par with experienced students?

Although prior research does not discuss these questions, the following hypotheses was developed based on the assumption that experienced students have had more practice coding than inexperienced ones:

(4) Experienced students will perform better than inexperienced students on active coding questions.
(5) Inexperienced and experienced students will perform equally on passive coding questions.

Many studies in computer science education also fail to take into account implicit biases. Since introductory computer science classes tend to be highly diverse, with students whose native language is often not English, it would be interesting to observe whether or not non-native English speakers' performance differs from that of native English speakers'.

(6) Non-native English speakers will perform worse on more verbose or concept-heavy quiz items.

4 Method

This study began in Spring 2019 and is currently ongoing. It is being conducted on the course Introduction to Python (1006) at Columbia University.

1006 classes occur twice a week, with one day devoted to lecture and the other devoted to a workshop session. During the lecture, the course instructor will go over new topics and examples that clarify concepts. During the workshop sessions, students will collaborate with one another on practice problems and improve their understanding of key computer science topics.

4.1 Data Collection

At the beginning of the semester, all students in the course are given a consent form and a brief overview of the study. Those who choose to participate fill out the consent form and provide demographic information about themselves. After these forms are collected, each student is issued a unique participant ID number in order to ensure anonymity, and the demographic data is entered onto a spreadsheet.

After roughly one-third of the semester has gone by and the students have learned basic computer science concepts, they will be given a survey and a pre-test (Quiz 1). The survey asks students to rate their abilities, skills, motivation, and understanding on a scale from 'Strongly Agree' to 'Strongly Disagree.' Students are also asked questions about their comfort or confidence in their own skills, which is rated on a scale of 'Usually' to 'Never.' The surveys, in essence, are an assessment of students' self-efficacy.

Quiz 1 tests simple concepts: accessing elements from a data structure, tracing through loops, tracing through function calls, and active coding with conditionals. The quizzes are graded by TAs for the course.

After about two-thirds of the course has gone by, a second survey and a post-test (Quiz 2) are handed out to students. While the second survey is identical to the first one, the quiz is very different. Quiz 2 tests students on more advanced topics: object instances, tracing through loops, active coding with data structures, and active coding with recursion. This quizzes are again graded by TAs for the course and inputted alongside the participant ID numbers.

4.2 Data Cleaning

While quiz grades were already entered onto a spreadsheet by TAs, the surveys were done on paper by the students and had to be manually entered. In the beginning of the course, around seventy students had consented to the study. But as the course progressed, students either dropped out or didn't complete the survey given to them, bringing the sample size lower.

As survey data was manually entered, many students were found to have missed a statement on the survey, forgotten to turn over the page and answer additional questions on the back, or forgone the survey entirely. As a result, there were many students with missing survey data. In order to complete the study, these students' data were taken out of the sample size, bringing the total count of

participants down from around seventy to forty-eight.

4.3 Statistical Analysis

After all data is inputted, a self-efficacy score was computed. After quantifying the scale from one to five, positive statements (e.g. “I have a thorough understanding of how programming works”) were added while negative statements (e.g. “I get frustrated when programming”) were subtracted. As there were more positive statements than negative ones, the end result was that the lower the self-efficacy score was, the more confident a student felt in their abilities and skills. Conversely, the less confident a student felt, the higher their self-efficacy score was.

After these self-efficacy scores were computed and entered into a spreadsheet, it was time to begin computing correlations. The primary goal was to see how effective and encouraging 1006 was, which meant looking at performance and self-efficacy. Pearson’s correlation coefficient was computed between performance and self-efficacy at the beginning of the course (with Quiz 1 and Survey 1) and at the end of the course (with Quiz 2 and Survey 2) in order to see how related the two measures are.

The independent variables experience, gender, ethnicity, and native language were split into different categories and an analysis of variance were used to establish the relationship between these variables and the measures of performance and self-efficacy. All statistics were run on Python and graphed on matplotlib in order to better visualize the general trend of performance and self-efficacy across all dependent variables.

An item analysis was conducted on the questions from Quiz 1 and Quiz 2 (with the exception of question 3 from Quiz 1, which was excluded because all students in the reduced sample size received full points for their answers). An analysis of variance was conducted between each question and experience level in order to gauge if there might be a correlation between experience and certain coding concepts. An analysis of variance was also conducted between each question and the native language of students, in order to establish if there were any implicit biases within the quiz questions.

5 Results

When conducting an analysis on Survey 1 (self-efficacy) against Quiz 1 (performance), the Pearson’s r-value was found to be -0.424, with a p-value of approximately 0.003.

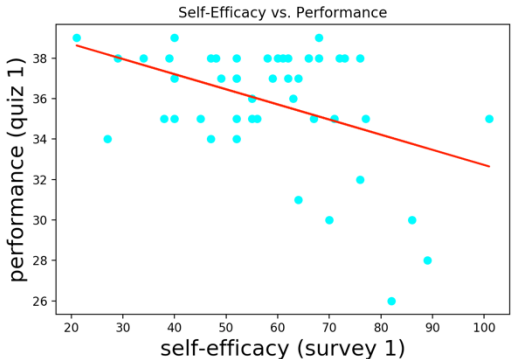

Figure 1 : Relationship between self-efficacy and performance at the beginning of the course.

Figure 1 displays a significant correlation between performance and self-efficacy. As self-efficacy increases (i.e. as the student feels worse), the performance decreases. Conversely, as self-efficacy decreases (i.e. as the student feels better), performance increases.

The same analysis was done for Survey 2 against Quiz 2 in order to see if there was any change in the relationship between self-efficacy and performance by the end of the course. Here, the Pearson’s r-value was found to be -0.512, with a p-value of approximately 0.0002.

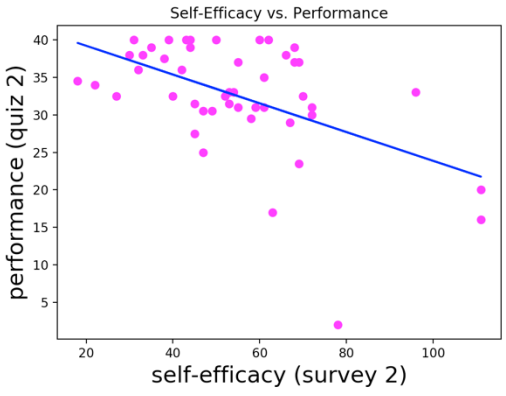

Figure 2 : Relationship between self-efficacy and performance at the end of the course.

From Figure 1 to Figure 2, there is a general upward and leftward trend that the data points exhibit, indicating that while there are some students with high self-efficacy and low

performance by the end of the course, many do improve in performance and self-efficacy.

Experience in the class was split across three levels: low, medium, and high. Low experience corresponded to students who entered 1006 with no prior programming experience whatsoever. Medium experience corresponded to students who entered 1006 with some prior programming experience (e.g. they had attended one coding class and knew one to two programming languages). High experience corresponded to students who entered 1006 with advanced knowledge of programming (e.g. they had attended more than one coding class prior to the course and knew more than two programming languages). Analysis of variance was conducted between these levels of experience and performance and self-efficacy in order to determine if there was any significant relationship between experience and the latter two variables.

There was no significant relationship found between experience and performance at the beginning of the course, but there was a correlation towards the end. It is also worth noting that there is a general upward trend in performance across all levels of experience.

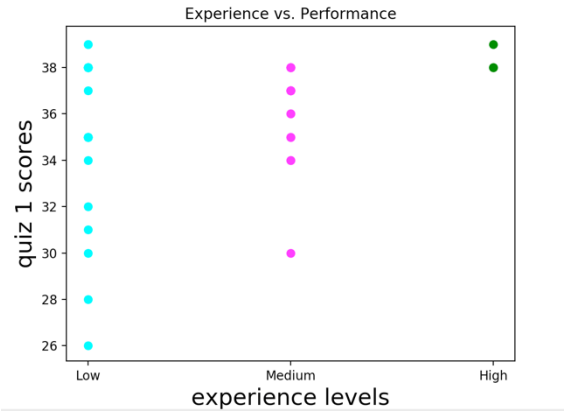

Figure 3 : Relationship between experience and performance at the beginning of the course. ANOVA = 2.114, p-value = 0.132.

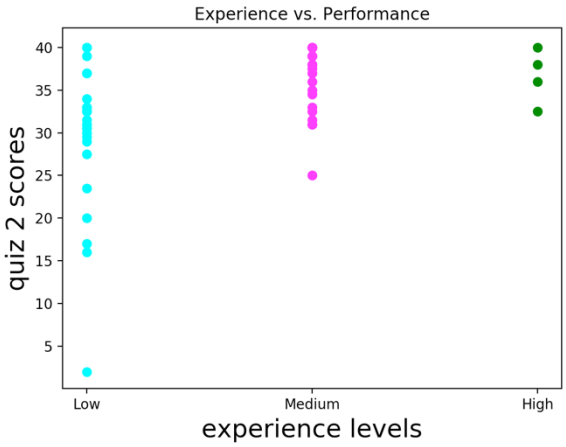

Figure 4 : Relationship between experience and performance at the end of the course. ANOVA = 5.152, p-value = 0.009.

While there was only a significant correlation between experience and performance by the end of the course, there was a significant correlation between experience and self-efficacy throughout the course. There is a slight downward trend across all levels of experience, indicating some improvement in self-efficacy by the end of the course.

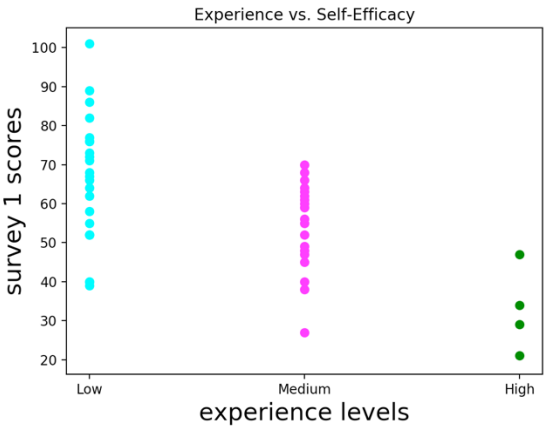

Figure 4 : Relationship between experience and self-efficacy at the beginning of the course. ANOVA = 10.603, p-value = 0.0001.

Figure 6 : Relationship between experience and self-efficacy by the end of the course. ANOVA = 4.383, p-value = 0.0183.

Similar to the relationship between experience and performance, there was no significant correlation between gender and performance at the beginning of the course but there was one at the end.

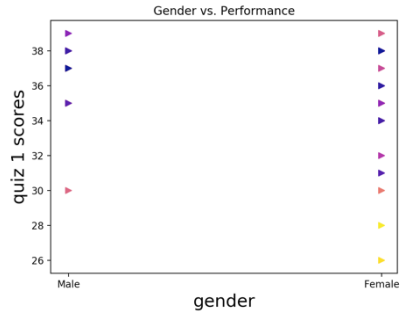

Figure 7 : Relationship between gender and performance at the beginning of the course. ANOVA = 2.319, p-value = 0.134.

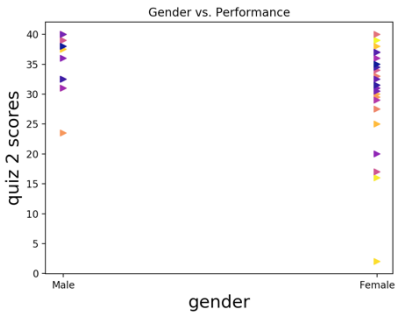

Figure 8 : Relationship between gender and performance at the end of the course. ANOVA = 7.627, p-value = 0.008

There was also a significant relationship between gender and self-efficacy throughout the course. Despite female students' performance being nearly on par with that of male students by the end of the course, female students' self-efficacy did not change very much.

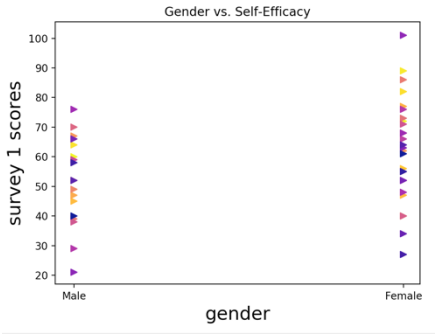

Figure 9 : Relationship between gender and self-efficacy at the beginning of the course. ANOVA = 4.544, p-value = 0.039.

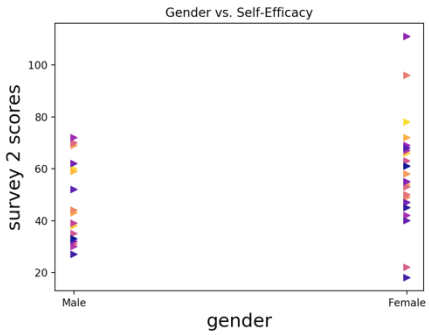

Figure 10 : Relationship between gender and self-efficacy at the end of the course. ANOVA = 5.016, p-value = 0.0301.

As seen in the below Figures 11-14, there were no significant relationships between ethnicity and performance or ethnicity and self-efficacy throughout the class.

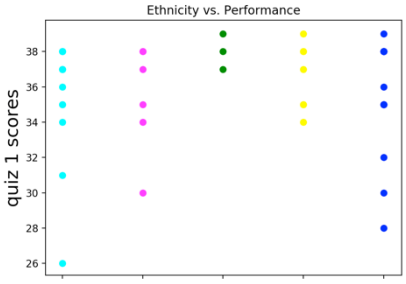

Figure 11 : Relationship between ethnicity and performance at the beginning of the course. ANOVA = 0.604, p-value = 0.662.

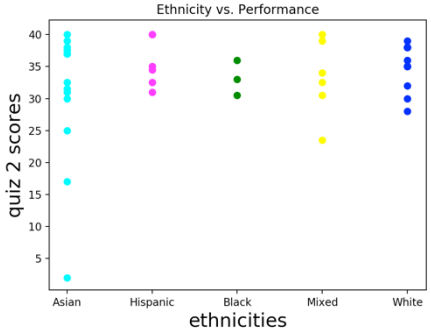

Figure 12 : Relationship between ethnicity and performance at the end of the course. ANOVA = 1.028, p-value = 0.404.

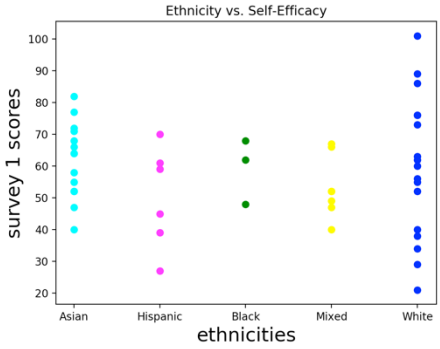

Figure 13 : Relationship between ethnicity and self-efficacy at the beginning of the course. ANOVA = 0.694, p-value = 0.601.

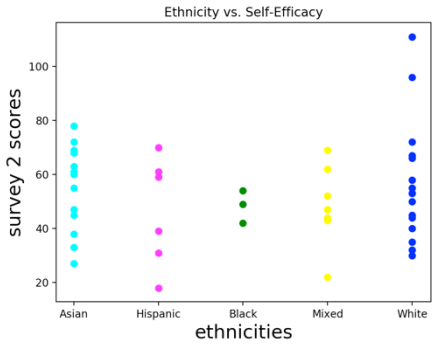

Figure 14 : Relationship between ethnicity and self-efficacy. ANOVA = 0.810, p-value = 0.526.

Similarly, as seen below in Figures 15-18, there were no significant relationships between native language and performance or native language and self-efficacy throughout the course.

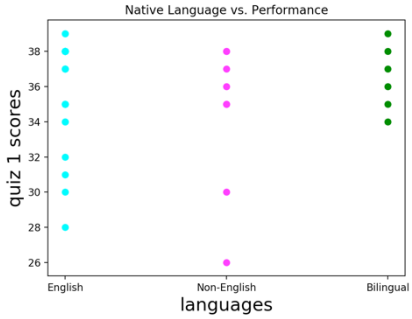

Figure 15 : Relationship between native language and performance at the beginning of the course. ANOVA = 1.340, p-value = 0.272.

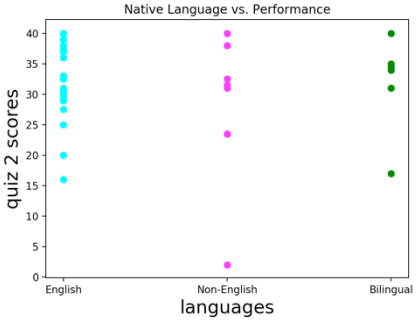

Figure 16 : Relationship between native language and performance at the end of the course. ANOVA = 0.905, p-value = 0.412.

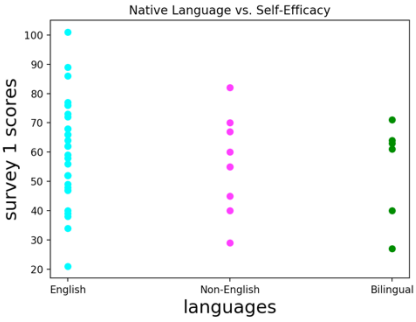

Figure 17 : Relationship between native language and self-efficacy. ANOVA = 0.323, p-value = 0.723.

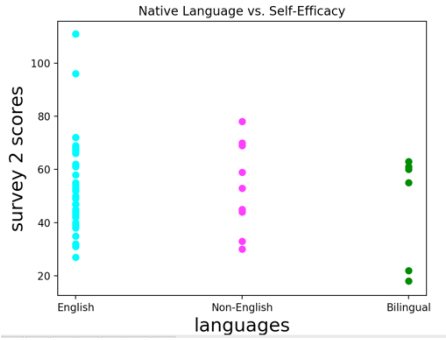

Figure 18 : Relationship between native language and self-efficacy at the end of the course. ANOVA = 0.636, p-value = 0.536.

An item analysis was conducted between each quiz question and experience, but there was no statistically significant correlation found. A second item analysis was conducted between each quiz question and native language, and it was only with the second question on Quiz 1 that any statistically significant correlation was found (ANOVA = 4.139, p-value = 0.023).

Question 2 (10 pts)

Show the output of the following program.

```
a = [['C', 'S', 'P', 'F', 'C', 'X', 'R', 'R', 'F'],\
     ['F', 'Z', 'O', 'Z', 'V', 'F', 'P', 'F', 'P'],\
     ['X', 'I', 'N', 'D', 'C', 'O', 'G', 'Q', 'A'],\
     ['Q', 'R', 'D', 'L', 'X', 'C', 'N', 'I', 'N'],\
     ['Z', 'B', 'R', 'E', 'T', 'U', 'K', '!', '!']]

result = []
k = 0
for j in (0,1,2):
    for i in (1,2,6):
        c = a[j+k][i]
        result.append(c)
    k += 1

print(result)
```

```
['S','P','R','I','N','G','B','R','K']
```

Figure 19 : Question 2 from Quiz 1.

Non-native English speakers may have scored lower on this particular question because native English speakers may have deduced that the answer would be 'SPRINGBRK' after only a couple of iterations of the for loops. Non-native English speakers who did not intuitively realize the answer may have traced through the loops until the very end, accidentally committing some arithmetic error since they were tracing through more iterations than others.

6 Conclusion

There is a strong correlation between self-efficacy and performance throughout the course; as self-efficacy worsens, so does performance and vice versa.

The effect of experience on performance in the beginning of the semester was somewhat unexpected. While there is no effect of experience on performance at the beginning of the course, there is some effect towards the end; namely, inexperienced students do not perform as well as more experienced ones. A possible explanation for this could be that items tested on Quiz 1 were very basic, and inexperienced students' knowledge of these concepts were on par with that of experienced students. However, as the course progressed and the topics became more advanced, inexperienced students lost traction and experienced students were able to stand out. This is a particularly troubling piece of information, because it shows that inexperienced and experienced students begin on a somewhat level playing field when the concepts are basic, but as the course progresses and the concepts become more difficult, the experience gap grows wider.

There was an effect of experience on self-efficacy throughout the course. While it was hypothesized that inexperienced students' self-efficacy would not improve by the end of the course, there was a general downward trend in self-efficacy by the second survey, indicating that some inexperienced students did feel more confident.

There was an effect of gender on experience at the end of the course but not at the beginning; this could be related to the reason why there was an effect of experience on performance at the end of the course but not the beginning. There was an effect of gender on self-efficacy throughout the course. Female students were less confident about themselves despite the fact their performance was on par with male students' performance for the most part. This could be an indication of lack of female support or gender bias within the class setting.

It was hypothesized that ethnic minorities' self-efficacy would worsen, but there was in fact no correlation between ethnicity and self-efficacy or ethnicity and performance through the course. Likewise, there was no correlation between native language and performance in general or native language and self-efficacy, meaning the hypothesis

that non-native English speakers would perform worse on more verbose questions was incorrect.

The item analysis also showed that there were no statistically significant correlations between experience and quiz items.

In general, it is apparent that Introduction to Python is effective in teaching students basic concepts of computer science and programming in the beginning of the semester but when more advanced topics are broached towards the latter part of the semester, this effectiveness is lost. Furthermore, although students' self-efficacy improves slightly by the end of the course, it is still low throughout the course, suggesting that students need more resources or ways to help reinforce confidence in themselves and their skills. While Introduction to Python certainly has the potential to be effective and encouraging, it still misses the mark.

7 Future Work

This study is currently ongoing. During the Fall 2019 semester, more data is being collected about prior experience and its effects on students in 1006.

The next step in this study would be to find an effective way to close the experience gap and better motivate inexperienced students. A proposed way to do this would be to split 1006 into two sections: one for advanced students and one for beginner students. These two sections would take place at the same time, with advanced students simply reviewing familiar concepts and working on programming projects for the majority of the semester and beginner students learning computer science concepts at an appropriate and steady pace while attending workshop sessions to further their own coding skills. This separation should ease any discomfort or discouragement inexperienced students would otherwise feel when in the presence of more advanced or high-performing students. Self-efficacy and performance will be monitored in this class to see if there is any improvement in closing the experience gap using this method.

References

Christine Alvarado, Gustavo Umbelino, and Mia Minnes. 2018. The Persistent Effect of Pre-College Computing Experience on College CS Course Grades. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, pp. 876-881.

Lisa L. Lacher, Albert Jiang, Yu Zhang, and Mark C. Lewis. 2019. Aptitude and Previous Experience in CS1 Classes. In *International Conference Frontiers in Education: CS and CE | FECS'17*, pp. 87-95.

Sarah Rauchas, Ian Sanders, and Benjamin Kumwenda. 2006. The Effect of Prior Programming Experience in a Scheme-based Breadth-First Curriculum at Wits. Abstract.

Thayer W. McGahee and Julia Ball. 2009. How to Read and Really Use an Item Analysis. *Nurse Educator*, 34(4):166-171.

Jerard Kehoe. 1995. Basic Item Analysis for Multiple-Choice Tests. *Practical Assessment, Research & Evaluation: A Peer-Reviewed Electronic Journal*, 4(10).

James P. Concannon and Lloyd H Barrow. 2009. A Cross-Sectional Study of Engineering Students' Self-Efficacy by Gender, Ethnicity, Year, and Transfer Status. *Journal of Science Education and Technology*, 18(2):163-172.

Peter J. Shull and Michael Weiner. 2002. Thinking Inside the Box: Self-Efficacy of Women in Engineering. *International Journal of Engineering Education*, 18(4):438-446.

Rex Karsten and Roberta M. Roth. 1998. The Relationship of Computer Experience and Computer Self-Efficacy to Performance in Introductory Computer Literacy Courses. *Journal of Research on Computing in Education*, 31(1):14-24.

Contribution

While this project was done in coordination with Dr. Daniel Bauer's team at the Columbia University Computer Science Department, all prior research, methodology, survey and quiz data entry, and statistical analyses were completed by Raefah Wahid.