An Analysis of Variable Names Used in CS 1 Code Submissions

ARJUN NAIR, University of Illinois at Urbana-Champaign
GEOFFREY CHALLEN, University of Illinois at Urbana-Champaign

While many universities have integrated lessons on coding style into their introductory computer science curricula, very little is known about the external factors impacting the coding style of CS 1 students, specifically when it comes to the variable names that they use in their code. There has been some amount of prior research that explores the effectiveness of various methods of teaching coding style, specifically good variable naming practices, to introductory students; however, none of them focus on various inequities that may be occurring between different groups of students. In this paper, we propose three quantitative metrics for measuring variable name quality on the aggregate level and then use them to analyze the variable names found in code submissions submitted by CS 1 students at A Large Rural University. We show that while college major and prior computer science experience do not seem to play a role in the names that students assign their variables, gender and mode of submission have a significant correlation with student variable naming choices. We also show that most gains in student variable name quality were made during the first month of the course, with a flatlining effect occurring after roughly thirty days of growth, and perform a qualitative analysis on the most common lemmas that appear in student variable names. We encourage researchers at other universities to perform similar analyses of the code submitted by their own CS 1 students so that we can determine whether these trends are school-specific or broadly applicable to CS 1 students at all universities.

CCS Concepts: • **Information systems** → *Information extraction*; • **Social and professional topics** → **CS1**; *Gender*; *User characteristics*; • **Theory of computation** → Grammars and context-free languages.

Additional Key Words and Phrases: CS 1, access and equity, coding style, variable names, formal language analysis

ACM Reference Format:
Arjun Nair and Geoffrey Challen. 2021. An Analysis of Variable Names Used in CS 1 Code Submissions. 1, 1 (October 2021), 24 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Variable naming style has been shown to be extremely important for code comprehension [1], with full-word identifiers being the most easily comprehensible on average for professional programmers [12]. It is no surprise, then, that computer science departments have begun to emphasize to their students the importance of good variable naming practices and, more generally, comprehensible coding style, with many adopting code reviews and/or lectures that are geared towards teaching students how to design readable code [7].

Many researchers in computing education have investigated the efficacy of various techniques for teaching coding style, often with a specific focus on variable naming practices [6, 10]; however, there has been relatively little attention paid to the external and demographic factors that may affect an introductory student's coding style, such as prior

Authors' addresses: Arjun Nair, arjunvn2@illinois.edu, University of Illinois at Urbana-Champaign; Geoffrey Challen, challen@illinois.edu, University of Illinois at Urbana-Champaign.

Manuscript submitted to ACM

experience, gender, and major. Researchers outside of the computing education community have found that writing style is a strong predictor of group identity [14], including gender [8], age [15], social status [3], and even parenthood and political orientation [11]. Access and equity are longstanding issues in the field of computer science and the software industry, and thus it is in the interest of computer scientists and researchers alike to examine how such factors might be affecting coding style, specifically for introductory computer science students who may still be deciding on whether or not the field is right for them.

In this study, we explore this question by defining three metrics designed to assess the quality of a given variable name and then using them to analyze the variable names found in over one million code submissions written by CS 1 students over the course of the Fall 2019 semester. We uncover several trends relating to how external and demographic factors, specifically major, prior experience, gender, and mode of submission, correlate with student variable naming practices and show that variable name quality rapidly improves over the first month of CS 1 and then proceeds to plateau for the rest of the course, with this effect holding even more strongly for those students who start off with significantly weaker variable naming style than others. We also discuss the most frequent lemmas used by students in their variable names, zooming in specifically on those lemmas most often used by students with the weakest variable naming practices, and test the association between the three metrics developed for this study in order to confirm whether or not they align with the assumptions we make about them in the following section.

2 VARIABLE NAME METRICS

Measuring variable name quality is an inherently subjective task, one that could most accurately be done by human annotators with programming backgrounds. When dealing with millions of variables, however, such labeling is impractical, and, thus, the use of automated methods for measuring the overall comprehensibility of the variable names written by each student was a necessity for this study. In this section, we propose three metrics for measuring variable naming quality and posit that, on the aggregate level, they serve as an indicator of whether one group of students is writing more comprehensible variable names than another group of students.

As we will explain in further depth in Section 3.2, camel case was strictly enforced for all variable names. Thus, these metrics are designed under the assumption that students are using camel case variable names with a lowercase first letter (i.e. *camelCase*).

2.1 Variable Length

The *length* of a variable is defined as its total number of characters.

2.1.1 *Examples.* .

- *cat* has a variable length of three.
- *catDictionary* has a variable length of thirteen.

2.1.2 *Discussion.*

Previous work by Lawrie et al. [12] has shown that longer variable names are typically correlated with better code comprehension. However, as shown Binkley et al. [2], the inclusion of extraneous characters can have the opposite effect. Thus, length alone is not a sufficient metric for analyzing the overall quality of variable names; however, the use of longer variable names can be considered as an indicator of higher variable name quality when taken alongside the other two metrics.

2.2 Descriptivity

A variable is labeled as *descriptive* if it is a) longer than one character and b) each word in its name, as separated by camel case, can be found in the English dictionary. The dictionary used for this study was the Apache OpenOffice U.S. English Dictionary [17].

2.2.1 Examples.

- *setOfElephants* is descriptive. It is made up of the words *set*, *of*, and *elephants*, all of which can be found in the English dictionary.
- *setOfBlaaaaaaaaah* is not descriptive. The words *set* and *of* can be found in the English dictionary but *blaaaaaaaaah* cannot.
- *a* is not descriptive because it is only composed of a single character.
- *aBoomerang* is descriptive. It is made up of the words *a* and *boomerang*, which can both be found in the English dictionary and is not solely composed of one letter.

2.2.2 Discussion. The study by Lawrie et al. [12] also showed that full-word identifiers are typically the most comprehensible, followed closely by variables that use common multi-character abbreviations rather than full words. The Apache OpenOffice U.S. English Dictionary contains many abbreviations found in the English language; however, it excludes abbreviations found more commonly in programming such as *addr* for an address or *bool* for a boolean value. Thus, decriptivity may not be a perfectly accurate metric; however, as all student variable names are evaluated using the same dictionary, a group of students writing a significantly higher percentage of descriptive variable names than another group of students should still be a valid indicator that they are writing more comprehensible variable names overall when taken in the context of the other two metrics.

2.3 Oddness

For the following definition, consider a *lemma* to be the canonical or dictionary form of a set of words; for example, the lemma (or the *lemmatized form*) of the words *eat*, *eats*, *ate*, and *eating* is *eat*. A variable is labeled as *odd* if it does not contain a single word, as separated by camel case, whose lemmatized form is equivalent to that found in at least one variable name written by a different student for the same assessment. The lemmatizer used for this study was the one provided by the Stanford CoreNLP library [5].

2.3.1 Examples.

- Consider an assignment that received only two submissions, one from Student A and another from Student B. If Student A declared the variables listOfCows, elephant, and anotherElephant in their submission and Student B declared only the variable cowSet, the variables elephant and anotherElephant would be labeled as odd because they do not share lemmas with variables used outside of Student A's submission. In this example, listOfCows and cowSet share the overlapping lemma cow and are used by different students; therefore, neither of them would be labeled as odd.

2.3.2 Discussion. There is no previous work to our knowledge that specifically addresses the use of unusual lemmas in variable names. In this study, we make the assumption that students who have a strong tendency to use unusual lemmas in their variable names are also more likely to be writing variable names that are generally less comprehensible. Thus, we will interpret higher percentages of odd variables within a group to be a sign that those students may potentially be

writing less comprehensible variable names overall, especially if these names are also shorter and/or less descriptive than those used by other students.

2.4 Reliability of Metrics

It is clear that there are too many subjective factors at play for these metrics to predict the quality of an individual variable with a high degree of accuracy. However, we discuss the results of this study under the assumption that students who tend to consistently use longer, more descriptive, and less odd variable names are more likely to be writing more comprehensible variable names overall. In Section 12, we conduct a short correlational analysis in order to test the validity of our assumptions; however, the metrics themselves are not the primary focus of our investigation. We encourage further research into the quantitative factors that make a variable name more comprehensible, especially those that measure the use of unusual lemmas.

3 COURSE CONTEXT

At A Large Rural University, Introduction to Computer Science, which we will refer to as CS 1 throughout the rest of the paper, is an introductory course for computer science majors and minors. It is taught in Java and covers introductory programming concepts, such as loops, classes, and inheritance, in addition to the fundamentals of Android app development.

The university offers additional courses in introductory computing that are geared towards students pursuing engineering, business, and statistics majors; as a result, CS 1 at A Large Rural University is mainly geared towards students who are planning to move on to the next courses in the CS sequence, namely Discrete Structures and Software Design Studio.

During the Fall 2019 semester of CS 1, lectures took place three times per week, and lab sections met once per week; each lecture and lab section was fifty minutes each, and attendance was enforced through a participation grade. In addition to daily small programming problems, students completed one long-term project over the course of the semester in addition to a final project at the end.

3.1 Online Submission System

Students took weekly quizzes and three midterms in a proctored computer-based testing facility through *PrairieLearn*, an online system for delivering homework assignments and tests [18]. In addition to multiple choice questions, students were given various programming problems that assessed their ability to write code in a scaffolded environment. Students had unlimited attempts to solve these problems and were scored based on the number of test cases they passed.

After each quiz/midterm testing period, several questions were released as practice problems for students to complete as optional, ungraded review exercises in an unproctored setting of their choice. Submissions for both exam and practice problems were stored in a database for research purposes.

3.2 Camel Case Variable Names

Uniform coding conventions were enforced through *checkstyle*, a system that can check for style guideline adherence in a piece of Java source code [9]. Each submission was run through this system before grading and had to be resubmitted with the required coding style if it did not pass all the style guideline checks. Camel case with a lowercase first letter was enforced through *checkstyle* for all variable names.

3.3 Start-of-the-Semester Survey

At the beginning of the course, a survey was administered through Google Forms to 899 CS 1 students. It asked them several questions, including the following:

(1) What is your major? [Free Response]
(2) What is your college? [Multiple Choice]
(3) What is your gender? [Multiple Choice]
(4) Did you take CS in high school? [Multiple Choice]
(5) What CS courses have you taken previously at the University of Illinois? [Select All That Apply]

Anonymized student ids were attached to these forms, making it possible to match a student's code submissions with their survey results.

4 EXPERIMENTAL DESIGN

We focus our study on the aforementioned set of 899 CS 1 students.

4.1 Demographic Categorization

The University of Illinois at Urbana-Champaign offers sixteen majors in Computer Science, including a "CS-Eng" major offered through the university's College of Engineering and fifteen joint interdisciplinary majors collectively referred to as "CS + X" offered through other university colleges; as a result, majors were condensed into three categories: *CS-Eng*, *CS + X*, and *Other*. Students whose indicated major (Survey Q1) contained the strings "Computer Science" or "CS", case insensitive, were labeled as *CS-Eng* if their college (Survey Q2) was the College of Engineering or *CS + X* if they were part of any other college. Students whose indicated major did not contain either these terms were grouped into the *Other* category.

Students were also grouped into four categories of prior CS experience: *None*, *HS Only*, *Uni Only*, and *HS & Uni*. Those who had taken an introductory computing course in high school (Survey Q4), including but not limited to AP Computer Science A or AP Computer Science Principles, were considered as having high school experience. Those who took at least one prior course in computing at the University of Illinois at Urbana-Champaign (Survey Q5), including but not limited to one of the terminal courses for non-majors, were considered as having university experience. Those with neither high school nor university experience were placed into the *None* category. Those with both high school and university experience were placed into the *HS & Uni* category. Finally, those with only high school or only university experience were placed into the *HS Only* or *Uni Only* categories respectively.

Students were also grouped into three categories based on their gender identification (Survey Q3): *Female*, *Male*, and *Other*. Non-binary students (those in the *Other* category) were excluded from the analysis of gender v.s. variable naming practices due to low sample size (n = 5).

4.2 Extraction and Filtration

A total of 1,117,155 submissions were submitted over the course of the Fall 2019 semester and stored in a database. Each submission record contained the code written by the student, an id representing the specific assessment that it was submitted as part of, a timestamp, and an anonymized student id that could be matched to the above demographic categories.

In order to extract variables, each submission's code was processed into a parse tree by the ANTLR Java Parser [16]. Unparseable submissions were excluded, leaving a total of 566,693 parseable submissions to be used as part of the study. For every variable declaration made in a submission, excluding those found in for control statements, the name of the variable being declared would be added to a list of variable names for that submission. Thus, in the end, each submission would have an author id, an assessment id, a list of the variable names declared in that submission, and a timestamp indicating when it was initially submitted.

Variable names that appeared in more than 90% of final submissions for a given assessment were filtered out of the variable name lists for submissions to that assignment in order to exclude those names that were either strongly hinted at in the problem description or were provided in the starter code. About 17.8% of assessments contained a variable name that had to be filtered out through this procedure.

5 Q1: STUDENT DEMOGRAPHICS

Research Question 1: How do demographic attributes, specifically major, prior computer science experience, and gender, correlate with student variable naming choices in CS 1?

5.1 Q1 Methods

In order to answer this research question, we performed three ordinary least squares (OLS) regressions, one for each variable name metric, at an alpha level of 0.05. Each regression took the following form, where *Metric* is either length, oddness, or descriptivity, *CS-Eng* is true if and only if a student's major category is *CS-Eng*, *Non-CS* is true if and only if a student's major category is *Other*, *HS* is true if and only if a student's prior experience category is either *HS Only* or *HS & Uni*, *Uni* is true if and only if a student's prior experience category is either *Uni Only* or *HS & Uni*, and *Male* is only true if and only if a student's gender is male (as mentioned earlier, due to low sample size, students outside of the *Male* and *Female* categories were excluded from this analysis).

$$Metric \sim ols(CS\text{-}Eng + Non\text{-}CS + HS + Uni + Male) \tag{1}$$

The model was fit using least squares with Python's *statsmodels* library. In the next section, we compare the resulting p-values for each variable against an alpha level of 0.05 in order to determine which demographic factors have a significant correlation with each metric. We then provide accompanying visualizations for those metrics deemed statistically significant.

5.2 Q1 Results

In Tables 1-3, presented below, we display the p-values of each correlation coefficient for each model.

Variable	p	coef
CS-Eng	0.159	0.1952
Non-CS	0.211	0.1310
HS	0.279	0.0951
Uni	0.258	-0.1455
Male	0.012	-0.2157

Table 1. p-values and coefficients for each demographic attribute in the OLS regression for average variable length. Note that coefficients are measured in number of characters.

Variable	p	coef
CS-Eng	0.634	0.54
Non-CS	0.593	-0.45
HS	0.243	-0.83
Uni	0.440	-0.81
Male	0.005	-1.95

Table 2. p-values and coefficients for each demographic attribute in the OLS regression for percent descriptivity. Note that coefficients are measured in percentage points.

Variable	p	coef
CS-Eng	0.311	-1.15
Non-CS	0.971	0.03
HS	0.190	-0.94
Uni	0.065	1.94
Male	0.002	2.13

Table 3. p-values and coefficients for each demographic attribute in the OLS regression for percent oddness. Note that coefficients are measured in percentage points.

It can be observed above that major, as defined by the three categories *CS-Eng*, *CS + X*, and *Other*, had no significant correlation with the length, descriptivity, or oddness of the variable names that students declared in their code. Similarly, taking CS courses in high school and/or university was also shown to have no significant correlation with the length, descriptivity, or oddness of the variable names that students declared.

Gender, however was shown to have a significant correlation with all three metrics. Namely, female students were more likely than male students to write longer ($p = 0.012$), more descriptive ($p = 0.005$), and less odd variable names ($p = 0.002$). Figure 1 shows the distribution of average variable length for female and male students; it can be observed from the histogram that female students are more likely than male students to be in the category of students that declare variables of five characters or longer on average. Figure 2 shows the distribution of the percentage of descriptive variables for female and male students; as shown by the histogram the majority of students in both the female and male student groups had 90% or more of their variables classified as descriptive, but female students had an even larger

majority, proportionate to their size, that fell within the 90% or more category. Figure 3 shows the distribution of the percentage of odd variables for female and male students; it can be observed that the majority of female students had less than 10% of their variables labeled as odd, whereas for male students, only about 45% of them fell within the 10% or less category.

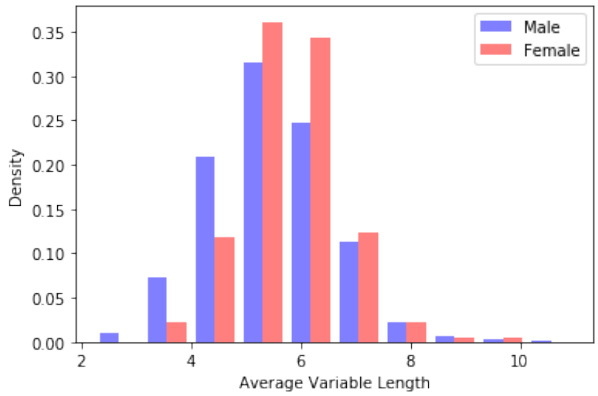

Fig. 1. Distribution of the average length of variables declared by female and male students for all assignments.

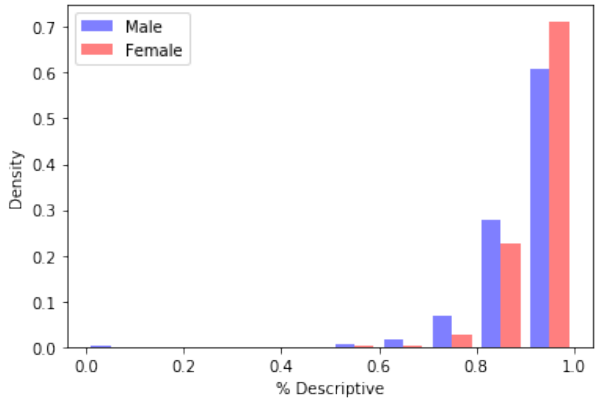

Fig. 2. Distribution of the percentage of descriptive variables declared by female and male students for all assignments.

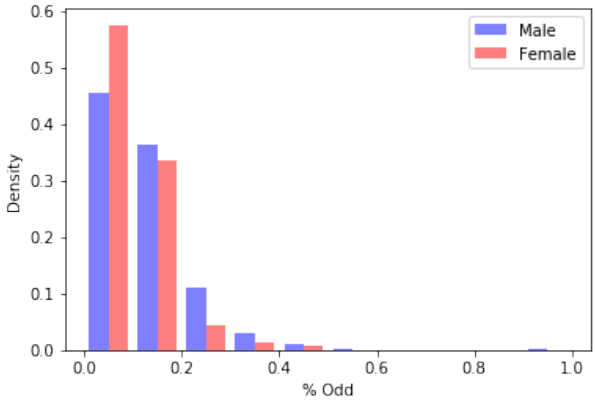

Fig. 3. Distribution of the percentage of odd variables declared by female and male students for all assignments.

6 Q2: MODE OF SUBMISSION

Research Question 2: Is there a significant difference between the variable names that CS 1 students declare under timed, proctored exam conditions in a testing facility as opposed to those that they declare under untimed, unproctored practice conditions in a setting of their choice?

6.1 Q2 Methods

As with Q1, we performed three OLS regressions for this analysis, one for each variable name metric, at an alpha level of 0.05. Each of these regressions took the following form, where *Metric* is either length, oddness, or descriptivity and *Exam* is true if and only if the variable name in question was submitted as part of a timed, proctored exam.

$$Metric \sim ols(Exam) \tag{2}$$

The model was fit using least squares with Python's *statsmodels* library. In the next section, we compare the resulting p-values for each variable against an alpha level of 0.05 in order to determine whether any significant difference exists between the length, descriptivity, and/or oddness of variable names declared during timed, proctored exams and those of variable names declared as part of untimed, unproctored practice submissions. We provide accompanying visualizations for those metrics deemed statistically significant at an alpha level of 0.05.

6.2 Q2 Results

Table 4, presented below, shows the p-value of the *Exam* correlation coefficient for each variable name metric.

Metric	p	coef
Length	0.356	0.0098
Descriptivity	0.000	3.96
Oddness	0.000	-5.75

Table 4. p-values and coefficients for the Exam parameter of each OLS regression. Coefficient for length is measured in number of characters, and coefficients for descriptivity and oddness are measured in percentage points.

It can be observed that students wrote significantly more descriptive ($p \approx 0.000$) and less odd variable names ($p \approx 0.000$) when under timed, proctored exam conditions as opposed to untimed, unproctored practice conditions. There was no significant correlation found, however, between variable length and mode of submission ($p = 0.356$). Figures 4 and 5 show the distributions of the percentages of descriptive and odd variables respectively for both exam and practice submissions.

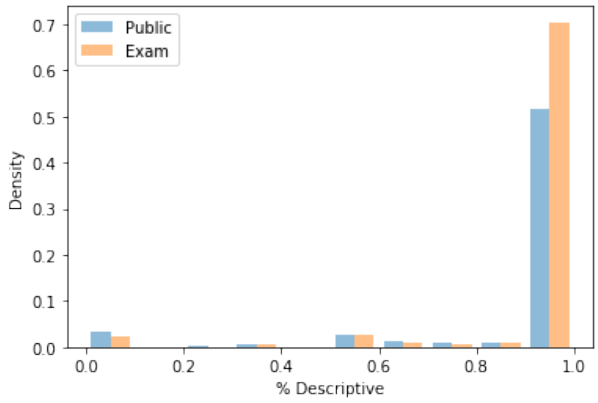

Fig. 4. Distribution of the percentage of descriptive variables declared by each student when completing publicly available review exercises v.s. timed, proctored exams.

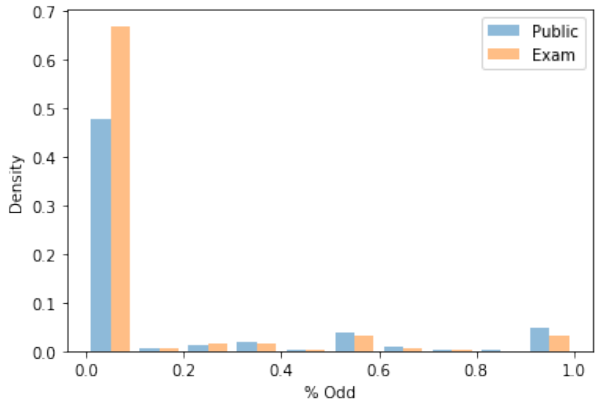

Fig. 5. Distribution of the percentage of odd variables declared by each student when completing publicly available review exercises v.s. timed, proctored exams.

7 Q2.1: DEMOGRAPHIC DISTRIBUTION OF EFFECTS

We conducted a followup analysis, Q2.1, in order to potentially identify the cause of the trends observed in our investigation of Q2.

Research Question 2.1: Is there a particular group of students that is more likely to write more descriptive and/or less odd variable names under timed, proctored exam conditions as opposed to untimed, unproctored exam conditions?

7.1 Q2.1 Methods

In order to answer this research question, we first had to create two (potentially overlapping) groups of students: those who wrote more descriptive variable names under exam conditions and those who wrote less odd variable names under exam conditions. In order to do this, we performed two OLS regressions for each student, one which measured whether the student was writing significantly more descriptive variable names under exam conditions and another which measured whether they were writing significantly less odd variable names under exam conditions. Each regression took the following form, where *Metric* is either descriptivity or oddness and *Exam* is true if and only if the variable name in question was submitted as part of a timed, proctored exam.

$$Metric \sim ols(Exam) \tag{3}$$

Each model was fit using least squares with Python's *statsmodels* library. For the rest of section Q2.1, consider *Group A* and *Group B* to be defined as such:

- *Group A* was composed of the students whose OLS *Exam* parameter for descriptivity had a positive correlation coefficient and a p-value less than 0.05.
- *Group B* was composed of the students whose OLS *Exam* parameter for oddness had a negative correlation coefficient and a p-value of less than 0.05.

We then performed two OLS regressions over the entire group of students in this study. The regressions took the following forms, where *A* is true if and only if the student is part of *Group A*, *B* is true if and only if the student is part of *Group B*, *CS-Eng* is true if and only if a student's major category is *CS-Eng*, *Non-CS* is true if and only if a student's major category is *Other*, *HS* is true if and only if a student's prior experience category is either *HS Only* or *HS & Uni*, *Uni* is true if and only if a student's prior experience category is either *Uni Only* or *HS & Uni*, and *Male* is only true if and only if a student's gender is male

$$A \sim ols(CS\text{-}Eng + Non\text{-}CS + HS + Uni + Male) \tag{4}$$

$$B \sim ols(CS\text{-}Eng + Non\text{-}CS + HS + Uni + Male) \tag{5}$$

For either OLS, a parameter having a p-value of less than 0.05 indicates that there is a significant difference between the proportions of that respective demographic in Groups A and/or B as opposed to the wider population of students who were part of this study. In the next section, we present a chart of correlation coefficients and p-values for these two OLS regressions and use it to determine which of these demographics, if any, were significantly more likely to be part of Groups A and/or B.

7.2 Q2.1 Results

Tables 5 and 6 show the correlation coefficients and p-values for the two OLS regressions conducted for Groups A and B respectively.

Metric	p	coef
CS-Eng	0.992	0.06
Non-CS	0.019	-11.20
HS	0.014	9.84
Uni	0.443	4.46
Male	0.262	4.37

Table 5. p-values and coefficients for each parameter of the Group A OLS Regression. Coefficients are measured in percentage points (percent chance of being in the group of students who use significantly more descriptive variables under exam conditions).

Metric	p	coef
CS-Eng	0.542	3.89
Non-CS	0.536	-2.99
HS	0.707	-1.53
Uni	0.801	1.49
Male	0.725	1.40

Table 6. p-values and coefficients for each parameter of the Group B OLS Regression. Coefficients are measured in percentage points (percent chance of being in the group of students who use significantly less odd variables under exam conditions).

First of all, it can be observed that there was no significant difference found in the distribution of major (p = 0.542 for *CS-Eng* and p = 0.536 for *Non-Major*), prior CS experience (p = 0.707 for *HS* and p = 0.801 for *Uni*), or gender (p = 0.352 for *Male*) between Group B (students who wrote less odd variable names under exam conditions) and the general pool of students. Likewise, there was no significant difference in gender distribution between Group A (students who wrote more descriptive variable under exam conditions) and the general pool of students (p = 0.262); however, students who had high school CS experience (p = 0.014) and those majoring in CS (p = 0.019) were significantly more likely to be part of Group A. University experience (p = 0.443), however, did not differ significantly between Group A and the general pool of students, and there was no significant difference between *CS-Eng* and *CS + X* (p = 0.992) when it came to their representation in Group A v.s. the general pool of students.

8 Q3: COURSE PROGRESS

Research Question: How do the length, descriptivity, and oddness of the variables that CS 1 students declare change as they progress through the course?

8.1 Q3 Methods

In order to answer this research question, we performed three ordinary least squares (OLS) regressions, one for each variable name metric, at an alpha level of 0.05. Each regression took the following form, where *Metric* is either length, oddness, or descriptivity and *Time* is the number of days elapsed (floating-point) between the time of submission and the start of the course (defined as 12:00 A.M. UTC on August 26th).

$$Metric \sim ols(Time) \tag{6}$$

The model was fit using least squares with Python's *statsmodels* library. In the next section, we compare the resulting p-values for the *Time* parameter on each model against an alpha level of 0.05 in order to determine whether course progress has an impact on student variable naming practices. We provide accompanying visualizations for those metrics deemed statistically significant; note that, for the sake of easier visualization, the figures plot student variable naming practices based on assessment averages (timestamp is based on average submission timestamp) rather than each individual submission, so as to reduce the number of points from hundreds of thousands down to less than one hundred.

8.2 Q3 Results

Table 7 shows the correlation and p-values for the *Time* parameter of each metric's OLS regression.

Metric	p	coef
Length	0.000	0.0060
Descriptivity	0.000	0.05
Oddness	0.000	-0.04

Table 7. p-values and coefficients for the Time parameter of each metric's OLS regression. Coefficient for length is measured in number of characters per day, and coefficients for descriptivity and oddness are measured in percentage points per day.

It can be observed that, as the course progressed, student variable names became significantly longer ($p \approx 0.000$), more descriptive ($p \approx 0.000$), and less odd ($p \approx 0.000$). Figures 6, 7, and 8 show that most of these gains occurred during the first thirty days of the course and then plateaued afterwards.

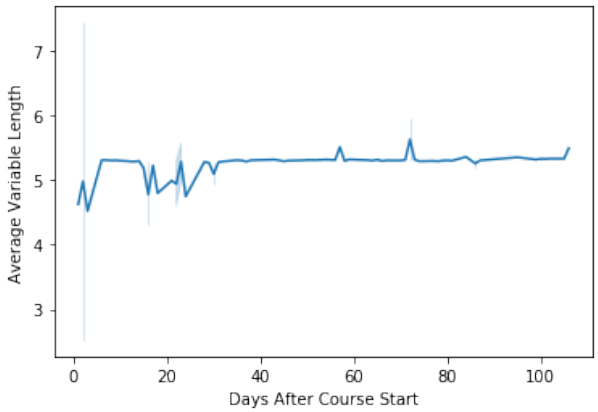

Fig. 6. Average length of variable names for each assessment as plotted by the average submission timestamp for that assessment.

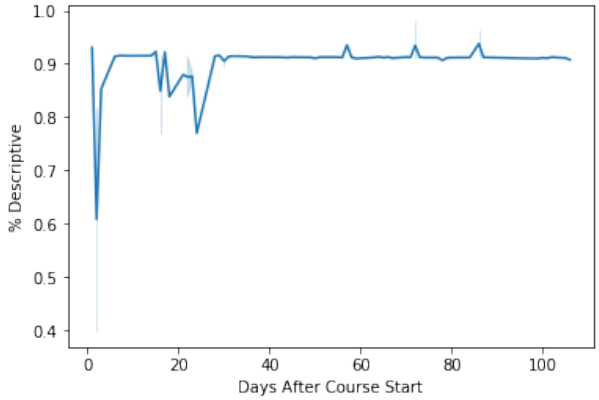

Fig. 7. Percent descriptivity of variable names for each assessment as plotted by the average submission timestamp for that assessment.

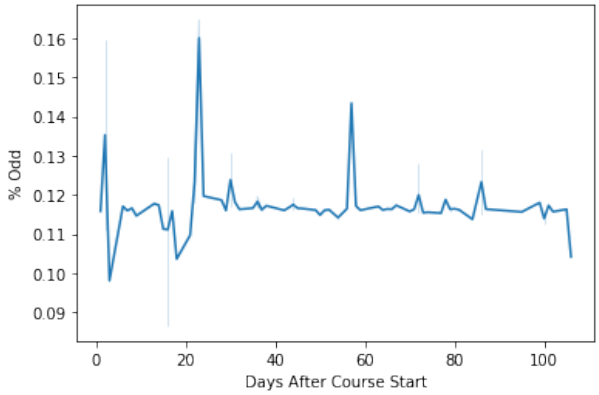

Fig. 8. Percent oddness of variable names for each assessment as plotted by the average submission timestamp for that assessment.

9 Q3.1: COURSE PROGRESS OF STUDENTS WITH WEAKER VARIABLE NAMING STYLE

We conducted a followup analysis, Q3.1, in order to focus specifically on the progress of students who started off with weaker variable naming style.

Research Question: For students who start off with variable names that are significantly shorter, less descriptive, and/or more odd than those of other students, how does course progress affect their progress along these metrics?

9.1 Q3.1 Methods

In order to answer this research question, we first had to identify students with weak variable naming practices. We defined three (potentially overlapping) groups of students:

- *Group A*: Students whose average variable length over the first fourteen days of the course was significantly lower than that of the general population of students (at an alpha level of 0.05).

- *Group B*: Students whose percent descriptivity (proportion of variables labeled descriptive) over the first fourteen days of the course was significantly lower than that of the general population of students (at an alpha level of 0.05).
- *Group C*: Students whose percent oddness (proportion of variables labeled odd) over the first fourteen days of the course was significantly higher than that of the general population of students (at an alpha level of 0.05).

In order to determine the students in each group, we performed three OLS regressions for each student, one for each variable name metric, over only those submissions from the first fourteen days of the course. Each regression took the following form, where *Metric* is either length, descriptivity, or oddness and *Student* is true if and only if the variable name in question was submitted as part of that specific student's submission.

$$Metric \sim ols(Student) \tag{7}$$

Each model was fit using least squares with Python's *statsmodels* library. Groups were formed based on the following conditions.

- *Group A* was composed of the students whose OLS *Student* parameter for length had a negative correlation coefficient and a p-value less than 0.05.
- *Group B* was composed of the students whose OLS *Student* parameter for descriptivity had a negative correlation coefficient and a p-value less than 0.05.
- *Group C* was composed of the students whose OLS *Student* parameter for oddness had a positive correlation coefficient and a p-value of less than 0.05.

For each group of students, we performed an ordinary least squares regression on variables from their submissions where *Metric* is equal to length, descriptivity, or oddness for Groups A, B, and C respectively and *Time* is the number of milliseconds elapsed between the time of submission and the start of the course (defined as 12:00 A.M. UTC on August 26th).

$$Metric \sim ols(Time) \tag{8}$$

The three models were fit using least squares with Python's *statsmodels* library. In the next section, we compare the resulting p-values for the *Time* parameter on each model against an alpha level of 0.05 in order to determine whether course progress has an impact on the variable naming practices of students who start off with variables of low length, low descriptivity, or high oddness. We provide accompanying visualizations for those metrics deemed statistically significant. As with Q3, for the sake of easier visualization, the graphs plot the averages for each assessment rather than each submission individually, so as to reduce the overall number of points.

9.2 Q3.1 Results

Table 8 shows the correlation and p-values for the *Time* parameter of each group's OLS regression.

Group	Metric	p	coef
Group A	Length	0.000	0.0031
Group B	Descriptivity	0.000	0.08
Group C	Oddness	0.000	-0.07

Table 8. p-values, coefficients, and corresponding metrics for each group's OLS regression. Coefficient for length is measured in number of characters per day, and coefficients for descriptivity and oddness are measured in percentage points per day.

It can be observed that as the course progresses, students in Group A are significantly more likely to write longer variables ($p \approx 0.000$) , those in Group B are more likely to write more descriptive variables ($p \approx 0.000$), and those in Group C are more likely to write less odd variables over time ($p \approx 0.000$). Figures 9, 10, and 11 show that over the first thirty days of the course, students starting off with weaker variable naming practices show a great deal of variation from assessment to assessment, moreso than other students but end up improving in the end just as the students in the general population did. It should be noted, when comparing these figures with those in the previous investigation, Q3, that the students in Groups A, B, and C do not fully catch up to the general population in terms of length (4.4 v.s. 5.5), descriptivity (83% v.s. 90%), and oddness (21% v.s. 12%) respectively.

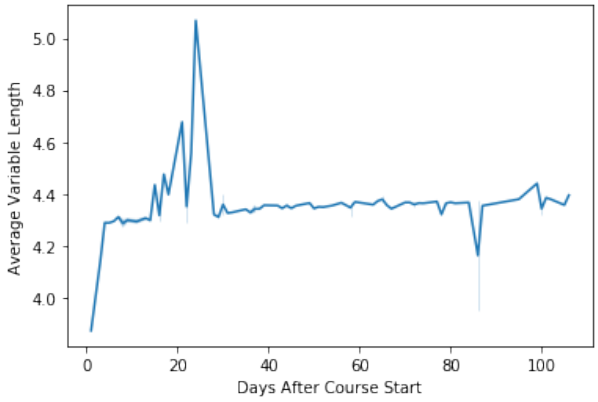

Fig. 9. Average length of Group A variable names for each assessment as plotted by the average submission timestamp for that assessment.

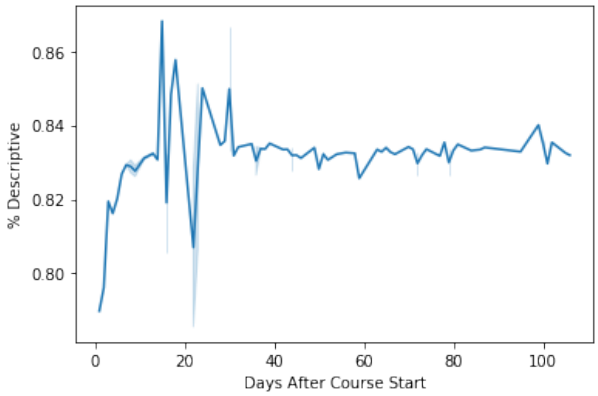

Fig. 10. Percent descriptivity of Group B variable names for each assessment as plotted by the average submission timestamp for that assessment.

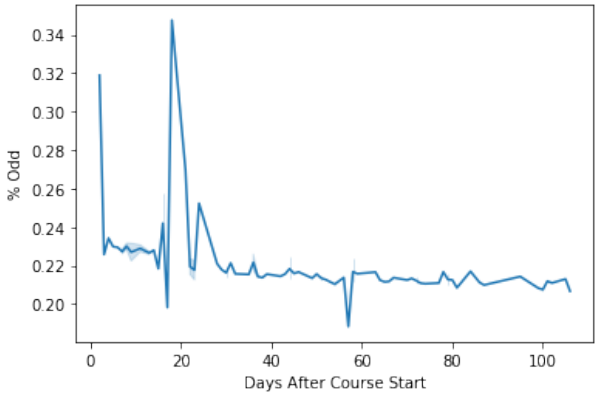

Fig. 11. Percent oddness of Group C variable names for each assessment as plotted by the average submission timestamp for that assessment.

10 Q4: MOST COMMON LEMMAS

Research Question: What are the most common lemmas used by CS 1 students in their variable names?

10.1 Q4 Methods

In order to answer this research question, we extracted a new list of variable names, one that only included variables declared in final submissions (i.e. each student's last submission for each assessment), so as to not give excess weight to students who submit a high number of submissions for each assessment or to assessments which had more repeat submissions than others. As detailed in Section 4.2, variable names that appeared in more than 90% of final submissions for a given assessment had already been filtered out of these variable lists, eliminating most of the names that were either given or hinted at in the problem description.

We then lemmatized each variable name and created a dictionary of counts for each lemma. We divided those counts by the total number of variables in order to get the frequency of each lemma. In the next section, we present the fifteen lemmas with the highest frequencies.

10.2 Q4 Results

Table 9 shows the fifteen most common lemmas used by CS 1 students, arranged from most common to least common, and their corresponding counts and frequencies. Note that, for the purpose of calculating frequencies, the total number of variables in the list was 44,559.

	count	name	index	array	temp	sum	new	a	current	to
Count	2613	2375	1739	1583	1500	1278	1241	1039	1016	996
Frequency	0.0586	0.0533	0.0390	0.0355	0.0336	0.0287	0.0279	0.0233	0.0228	0.0224

	pivot	type	return	value	counter
Count	940	851	797	792	692
Frequency	0.0211	0.0191	0.0179	0.0178	0.0155

Table 9. The counts and frequencies of the 15 most common lemmas used by CS 1 students.

It can be observed from the table above that fourteen of the fifteen most common lemmas are relevant in a programming context; *pivot*, for example, likely refers to the pivots used in sorting, while *current* and *temp* are fairly common intermediate values used for iterating and swapping. The lemma *a* is, arguably, unusual as a lemma in programming; it is possible that many students are repeatedly using *a* as a variable name because it is the first letter in the alphabet. Other than this, however, there are no lemmas that stand out as particularly inappropriate in a programming context, except for possibly *return*, *value*, and *name*. These lemmas, while somewhat indicative of the function of the variable, may be suboptimal placeholders for even more comprehensible variable names (for example, a pythagorean theorem function could have its return value be named *hypotenuse* instead of *return* or *returnValue*).

11 Q4.1: MOST COMMON LEMMAS FOR STUDENTS WITH WEAKER VARIABLE NAMING STYLE

Research Question: What are the most common lemmas for students who use variable names that are significantly shorter, less descriptive, or more odd than other students?

11.1 Q4.1 Methods

In order to answer this research question, we had to once again collect three (potentially overlapping) groups of students with weak variable naming practices just as we did in Q3.1. This time, however, we collected the groups of students who had weak variable naming practices over the entire duration of the course, not just the beginning.

In order to determine the students in each group, we performed three OLS regressions for each student, one for each variable name metric. Each regression took the following form, where *Metric* is either length, descriptivity, or oddness and *Student* is true if and only if the variable name in question was submitted as part of that specific student's submission.

$$Metric \sim ols(Student) \tag{9}$$

Each model was fit using least squares with Python's *statsmodels* library. Groups were formed based on the following conditions.

- *Group A* was composed of the students whose OLS *Student* parameter for length had a negative correlation coefficient and a p-value less than 0.05.
- *Group B* was composed of the students whose OLS *Student* parameter for descriptivity had a negative correlation coefficient and a p-value less than 0.05.
- *Group C* was composed of the students whose OLS *Student* parameter for oddness had a positive correlation coefficient and a p-value of less than 0.05.

For each group, we followed the process outlined in Section 10.1 to extract their fifteen most common lemmas and their corresponding counts and frequencies.

11.2 Q4.1 Results

Tables 10, 11, and 12 show the fifteen most common lemmas among the groups of students that write significantly shorter, less descriptive, or more odd variable names respectively.

	array	name	count	new	index	original	a	to	age	return
Count	26360	23132	18108	17265	16846	12102	11836	10381	10000	9418
Frequency	0.0808	0.0709	0.0555	0.0529	0.0516	0.0371	0.0363	0.0318	0.0306	0.0288

	temp	b	sum	return	c
Count	8508	7024	6294	5374	4894
Frequency	0.0261	0.0215	0.0193	0.0165	0.0155

Table 10. The counts and frequencies of the 15 most common lemmas used by students who started off with significantly lower average variable length than the general population.

	array	name	new	count	index	original	to	return	age	temp
Count	21375	17118	14822	14351	14296	10016	9224	8314	7394	5872
Frequency	0.0848	0.0679	0.0588	0.0569	0.0567	0.0397	0.0366	0.0330	0.0293	0.0233

	a	value	sum	height	pivot
Count	4974	3977	3887	3628	3581
Frequency	0.0197	0.0158	0.0154	0.0144	0.0142

Table 11. The counts and frequencies of the 15 most common lemmas used by students who started off with significantly lower percent descriptivity than the general population.

	array	name	count	new	index	original	to	return	age	temp
Count	21035	16730	14007	13286	12711	9676	8985	8347	7111	5465
Frequency	0.0855	0.0680	0.0569	0.05450	0.0516	0.0393	0.0365	0.0339	0.0289	0.0222

	a	value	sum	height	b
Count	5437	4162	3754	3246	3122
Frequency	0.0221	0.0169	0.0153	0.0132	0.0127

Table 12. The counts and frequencies of the 15 most common lemmas used by students who started off with significantly higher percent oddness than the general population.

It can be observed from the tables above that the groups do not differ very much from each other in terms of the fifteen most common lemmas used and their frequencies; in fact, with the exception of lemmas *b* and *c*, every lemma that can be found in the fifteen most common lemmas of Group A can also be found in those of Groups B and C. In addition to the lemma *a*, which appears in the list for all groups as it did for the general population, two additional single-letter lemmas can be found: *b* and *c*, with *b* appearing in Groups A and C and *c* appearing only in Group A. This suggests that many of the students with weaker variable naming practices might be more likely than other students to simply use the letters of the alphabet, in lexographical order, as variable names in place of more meaningful identifiers that could have been used to improve the comprehensibility of the code.

12 Q5: CORRELATION BETWEEN VARIABLE NAME METRICS

If variable length, descriptivity, and oddness are reliable indicators of comprehensibility on the aggregate level, then it would stand to reason that there would be a high correlation between them when measured as averages for each student. In this section, we test this hypothesis and show that there is, in fact, a correlation between all of these metrics on the aggregate level.

Research Question: Is there a correlation between the length, descriptivity, and oddness of the variables written by CS 1 students?

12.1 Q5 Methods

For each student, we calculated the average length of their variables and the percentages which were descriptive and odd. We then performed the following three OLS regressions, where *AvgLength* is the average length of all variables declared by a student, *PercentDescriptive* is the percentage of all variables they declared that are descriptive, and *PercentOdd* is the percentage of their variables that are labeled as odd.

$$AvgLength \sim ols(PercentOdd) \tag{10}$$

$$PercentDescriptive \sim ols(AvgLength) \tag{11}$$

$$PercentOdd \sim ols(PercentDescriptive) \tag{12}$$

For any of the three above OLS regressions, having a p-value of less than 0.05 indicates that there is a significant correlation between the two metrics compared in the regression. In the next section, we present the correlation

coefficients and p-values of each regression and show that all three of these metrics are correlated in such a manner that could be predicted based on the assumptions outlined in Section 2.4.

12.2 Q5 Results

Table 13 shows the correlation coefficients and p-values for the three OLS regressions conducted as part of this investigation.

Metric 1	Metric 2	p	coef
Length	Oddness	0.000	-0.004645
Length	Descriptivity	0.004	0.00525
Oddness	Descriptivity	0.000	-0.4297

Table 13. p-values and coefficients for each OLS regression, with Metric 2 being the predictor variable. Coefficients are measured in the units of Metric 1 divided by the units of Metric 2, where length is measured in number of characters and descriptivity/oddness are measured in percentage points.

It can be observed from the table above that there is a significant positive correlation between average length and percent descriptivity ($p = 0.004$) and a significant negative correlation between both of those metrics and oddness ($p \approx 0.000$ for both). This observation is consistent with the premise that higher average length, higher percent descriptivity, and lower percent oddness are signs of greater variable name comprehensibility on the aggregate level, as these three metrics correlate together in the same fashion.

13 DISCUSSION

Our results show that major and prior CS experience, whether in high school, university, or both, do not seem to have a significant correlation with variable naming practices among the general pool of students, at least for those students taking this specific course. This is a surprising result, as both commitment to the field (reflected by major) and prior experience might be expected to affect the way in which students write their code. Prior computing experience is often cited as one of the biggest challenges facing underrepresented groups in technology [4], and thus it is refreshing to see that CS 1, at least at the University of Illinois at Urbana-Champaign, appears to be fairly accessible and equitable when it comes to teaching good variable naming practices to students of all experience levels and majors.

Our analysis of gender, however, does reveal a significant difference in the ways in which female and male students write variable names on average, with women tending to write longer, more descriptive, and less odd variable names than men. Previous work has shown differences in the types of vocabulary used by female and male university students when writing essays [8]; as such, one possible explanation for the disparity between female and male students is that the same "gender-linked" language differences that impact essay writing, especially those that pertain to vocabulary choice, may also influence the variable naming decisions of students. Ishikawa [8] showed that female students tend to use more intensifiers and modifiers than male students in their essay writing; if such differences were to extend to variable naming choices as well, this could, at the least, explain the tendency of female students to write longer variable names, as including more modifiers in the variable name will increase its total number of characters. This may also explain the lower percentage of odd variables seen in submissions made by female students, as having more words (and hence more lemmas) in each variable name increases the chance that they will share at least one lemma in common with another student's variable name in the same assessment. We put forward two possible explanations

that may account for the difference in the percentage of descriptive variables declared by female and male students. One possibility is that male students, for whatever reason, are more likely to purposely use non-descriptive words as variable names when writing code, potentially so that they can work through the exam or exercise quicker. Another possibility, however, is that male students have a higher incidence of spelling errors in their variable names, which will often cause an otherwise comprehensible variable name to be labeled as non-descriptive.

Surprisingly, students tended to write names that were significantly more descriptive and less odd when placed under timed, proctored exam conditions as opposed to untimed, unproctored practice conditions in a setting of their choice. Previous work by psychologists has shown that the mere presence of other individuals can serve as a means of social facilitation that improves a person's performance on various tasks, especially those that they have practiced beforehand [13]; thus, it is possible that the presence of the proctor as well as the other test-takers in the room caused students to become more proficient at writing comprehensible variable names. It is also possible, however, that this effect was caused not by social facilitation but rather by the time limit imposed on students while they took the exam or even the testing facility in which they sat as they wrote their code. Further research would need to be conducted in order to separate the effect of the time limit and the facility itself from the presence of the proctor and the other test-takers.

As students progressed through the course, the variable names they wrote became significantly longer, more descriptive, and less odd; however, as seen in Figures 6, 7, and 8, most of these gains were made during the first thirty days of the course. Those who started off with weaker variable naming practices improved rapidly over the course of the first thirty days and plateaued afterwards just like the students in the general population; however, even after improving, they were not able to catch up entirely to students who started off with better variable naming practices.

The most common lemmas used by students tended to correspond with fairly common variable names often used by programmers, such as *current*, *temp*, and *pivot*. Certain lemmas, however, suggested that students were using somewhat vague identifiers to describe their variables; examples include *return*, *value*, and *array*, all of which could be replaced by a more specific and meaningful name. The most common lemmas used by students with weaker variable naming practices were fairly similar to those used by students in the general population, although they included a higher incidence of single-letter lemmas, specifically *a*, *b*, and *c*. Given that these are the first three letters in the alphabet, they were likely used in a context-free manner as the first names that these students could think of rather than as meaningful descriptions for the values being stored within them.

Our final investigation revealed that the average length of a student's variable name has a positive correlation with the percentage of their variables labeled as descriptive and a negative correlation with the percentage of their variables labeled as odd. This is consistent with our assumption that higher length, higher descriptivity, and low oddness can be taken together as indicators of overall comprehensibility; however, it is, of course, not enough on its own to actually prove this assumption to be true.

13.1 Avenues for Future Research

We propose two types of followup studies that could help the computing education community better understand the variable naming practices of CS 1 students as well as the external and demographic factors that influence them:

13.1.1 Reproducing Results at Other Universities. It is unclear whether or not our findings actually extend to students in CS 1 courses offered outside of the University of Illinois at Urbana-Champaign. We propose that researchers at other universities conduct similar studies on students in their own introductory computing courses (preferably those

designed for majors), over the course of multiple semesters if possible, and compare their results with ours in order to see whether the trends that we have uncovered hold universally.

13.1.2 Verifying Quantitative Measures of Aggregate Variable Name Quality. This study relies on the assumption that the metrics length, descriptivity, and oddness give us meaningful information about the quality of a variable name when considered on the aggregate level. While the correlational analysis we conducted in Q5 showed that length, descriptivity, and oddness are, in fact, correlated in the manner that we would expect under this assumption, this finding alone is not enough to prove that these metrics are a good indicator of overall comprehensibility. We propose that researchers conduct followup studies that aim to determine the quantitative factors that can predict the quality of a programmer's variable names on the aggregate level, especially those measures that relate to the presence of unusual lemmas.

14 CONCLUSION

This study shows that gender and testing environment, but not prior experience or major, serve as valid predictors for student variable name quality in CS 1. Specifically, we found that female students typically write consistently longer, more descriptive, and less odd variable names than men and that students from all groups are more likely to write longer and more descriptive variable names when writing code in a timed, proctored testing environment as opposed to a unproctored review environment of their choice. We found that students made gains in terms of variable length and descriptivity during the first month of the course, but they then proceeded to plateau along these metrics afterwards; additionally, the students who had variables with the significantly shorter length, lower descriptivity, and/or higher oddness than other students in the beginning of the course showed huge improvement along these metrics as the course progressed. The list of most common lemmas, both for students with weak variable naming style and for those in the general population, mostly corresponded with relevant concepts in programming taught in the course; however, especially for students with poor variable naming style, the use of single-letter placeholders and vague filler words such as *array*, *return* and *value* were fairly common. We tested the association between the three metrics proposed at the beginning of the study and found that the correlations between them were significant and predictable based on our assumptions about those metrics and how they relate to comprehensibility. For CS 1 instructors who would like to introduce their students to the principles of good coding style, specifically as it pertains to variable naming, it is especially important to note the issue of single-letter or context-free placeholders and encourage students early on to abandon these practices before they become increasingly habitual. For courses with online code submission systems, it may be possible to conduct automated variable name quality analysis of the code students submit to any given assessment; if implemented, such a system could help instructors capture the overall trends in their students' variable naming style over time. Further studies will need to be conducted at other universities in order to confirm whether or not the trends that we have uncovered hold universally or are just applicable to this specific course; additional work must also be done to confirm the validity of the three metrics used in this study to approximate comprehensibility.

ACKNOWLEDGMENTS

We would like to thank the members of the CS 1 course staff who provided feedback at various stages of the project.

REFERENCES

[1] E. Avidan and D. G. Feitelson. 2017. Effects of Variable Names on Comprehension: An Empirical Study. In *2017 IEEE/ACM 25th International Conference on Program Comprehension (ICPC)*. 55–65. https://doi.org/10.1109/ICPC.2017.27

[2] Dave Binkley, Dawn Lawrie, Steve Maex, and Christopher Morrell. 2009. Identifier length and limited programmer memory. *Science of Computer Programming* 74, 7 (2009), 430 – 445. https://doi.org/10.1016/j.scico.2009.02.006

[3] Philip Bramsen, Martha Escobar-Molano, Ami Patel, and Rafael Alonso. 2011. Extracting Social Power Relationships from Natural Language. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, 773–782. https://www.aclweb.org/anthology/P11-1078

[4] Gallup Inc. Google Inc. 2016. Diversity Gaps in Computer Science: Exploring the Underrepresentation of Girls, Blacks and Hispanics. (2016). http://goo.gl/PG34aH

[5] Stanford NLP Group. 2020. Lemmatization. https://stanfordnlp.github.io/CoreNLP/lemma.html.

[6] Christopher Hundhausen, Anukrati Agrawal, Dana Fairbrother, and Michael Trevisan. 2009. Integrating Pedagogical Code Reviews into a CS 1 Course: An Empirical Study. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education* (Chattanooga, TN, USA) *(SIGCSE '09)*. Association for Computing Machinery, New York, NY, USA, 291–295. https://doi.org/10.1145/1508865.1508972

[7] Theresia Devi Indriasari, Andrew Luxton-Reilly, and Paul Denny. 2020. A Review of Peer Code Review in Higher Education. *ACM Trans. Comput. Educ.* 20, 3, Article 22 (Sept. 2020), 25 pages. https://doi.org/10.1145/3403935

[8] Yuka Ishikawa. 2015. Gender Differences in Vocabulary Use in Essay Writing by University Students. *Procedia - Social and Behavioral Sciences* 192 (2015), 593 – 600. https://doi.org/10.1016/j.sbspro.2015.06.078 The Proceedings of 2nd Global Conference on Conference on Linguistics and Foreign Language Teaching.

[9] Roman Ivanov. 2020. Checkstyle. https://checkstyle.sourceforge.io/.

[10] Iftikhar Ahmed Khan, Mehreen Iftikhar, Syed Sajid Hussain, Attiqa Rehman, Nosheen Gul, Waqas Jadoon, and Babar Nazir. 2020. Redesign and validation of a computer programming course using Inductive Teaching Method. *PloS one* 15, 6 (06 2020), e0233716–e0233716. https://doi.org/10.1371/journal.pone.0233716

[11] Miriam Koschate, Elahe Naserian, Luke Dickens, Avelie Stuart, Alessandra Russo, and Mark Levine. 2021. ASIA: Automated Social Identity Assessment using linguistic style. *Behavior Research Methods* (2021). https://doi.org/10.3758/s13428-020-01511-3

[12] Dawn Lawrie, Christopher Morrell, Henry Feild, and David Binkley. 2007. Effective identifier names for comprehension and memory. *Innovations in Systems and Software Engineering* 3, 4 (2007), 303–318. https://doi.org/10.1007/s11334-007-0031-2

[13] Hazel Markus. 1978. The effect of mere presence on social facilitation: An unobtrusive test. *Journal of Experimental Social Psychology* 14, 4 (1978), 389 – 397. https://doi.org/10.1016/0022-1031(78)90034-3

[14] Dong Nguyen, A Seza Doğruöz, Carolyn P Rosé, and Franciska de Jong. 2016. Computational sociolinguistics: A survey. *Computational linguistics* 42, 3 (2016), 537–593.

[15] Dong Nguyen, R. Gravel, D. Trieschnigg, and T. Meder. 2013. "How Old Do You Think I Am?" A Study of Language and Age in Twitter. In *ICWSM*.

[16] Terence Parr. 2017. ANTLR Java Parser. https://github.com/antlr/antlr4/blob/master/runtime/Java/src/org/antlr/v4/runtime/Parser.java.

[17] Marco A.G. Pinto. 2021. English Dictionaries for Apache OpenOffice. https://extensions.openoffice.org/en/project/english-dictionaries-apache-openoffice.

[18] M. West, Geoffrey L. Herman, and Craig B. Zilles. 2015. PrairieLearn: Mastery-based Online Problem Solving with Adaptive Scoring and Recommendations Driven by Machine Learning.