

MEASURING THE EFFECT OF AMOUNT OF REQUIRED STUDENT EFFORT ON
EXAM PERFORMANCE

BY

HONG CHENG

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Adviser:

Assistant Professor Craig Zilles
Teaching Assistant Professor Neal Davis

ABSTRACT

In this thesis, I examine the relationship between the amount of required effort from students and their performance on the corresponding exams in an introductory programming class. I employed an online learning system PrairieLearn which is able to require that students complete each question correctly multiple times to receive full scores in order to quantify the amount of work students have done. Two groups of students are assigned different minimum points required in order to get a full score in a quiz. Their effort is quantified by their number of attempts and the time spent on quizzes. The findings do not show a difference between two groups. The students from both groups do not show a significant difference in their exam scores. However, the results strongly suggests that students who get higher scores in the exam spend fewer tries but a similar amount of time on quizzes to get the correct answers. The study also assign questions to two groups which experience different grading treatment on the same set of questions. Group B, which has a tougher grading treatment, is compensated with extra points towards the total quiz score. The findings show that the students in group B submit about 40% fewer incorrect answers. The study concludes that the effort from each student in online quizzes does not show a correlation with their exam performance, and they check answers more carefully in online quizzes when they perceive that there are more points at stake.

To my wife Yi.

Without her support, I would have transferred to a non-thesis track.

ACKNOWLEDGMENTS

My thanks to Professor Neal Davis, who meets with me every week, and provided countless invaluable advices from ideas to methods. I learned how to teach from him.

My thanks to Professor Craig Zilles, who took part in designing Prairie Learn, led me to it, and laid solid foundations for my research. He is also my first professor in Computer Science.

My last thanks are to all my fellow teaching assistants in CS 101. Their contributions in drafting the quizzes are extremely helpful, and I could not have provided students with all these well-designed quizzes without them.

TABLE OF CONTENTS

CHAPTER 1	INTRODUCTION	1
CHAPTER 2	RELATED WORK	3
2.1	The effect of student learning time	3
2.2	Active learning	3
2.3	Quizzes in active learning	4
CHAPTER 3	EXPERIMENT SETUP	6
3.1	Class: CS 101 Intro to Computing	6
3.2	Platform: PrairieLearn	10
3.3	Evaluation: CBTF Testing	10
CHAPTER 4	EXPERIMENT	12
4.1	First stage experiment	12
4.2	Second stage experiment	17
4.3	Third stage experiment	20
CHAPTER 5	DISCUSSION	26
5.1	Summary of the experiments	26
CHAPTER 6	CONCLUSION	28
CHAPTER 7	FUTURE WORK	29
7.1	Data collecting	29
7.2	Discourage guessing	29
7.3	PrairieLearn	30
APPENDIX A	QUIZ VIEW BY STUDENTS	31
APPENDIX B	ANALYSIS IN R	35
REFERENCES	39

CHAPTER 1: INTRODUCTION

Student learning time can be divided into two categories: in-classroom and outside-of-classroom. Although numerous studies have been done on the effect of the impact of instructional time on students' performance, fewer have studied the time spent by students outside the classroom because instructional time in the classroom is more costly to educators. It is also more difficult to track the students' time spent on learning outside classrooms without assessing them in a controlled environment.

Gromada and Shewbridge[1] point out that "student learning time is a key educational resource". In order to put the resource to its best use, the educators need seek out the most efficient way of teaching new knowledge to the students. The in-class attracts more attention since they are of higher cost and easier to measure compared to outside-classroom learning. In a study conducted by Cattaneo et al.[2], they claim that "instructional time is not only an important but, most importantly, a scarce resource in education production". They come to the conclusion that instruction time has a positive effect on student performance, but varies from student to student. However, Trout[3] finds out that "students in the one-day-a-week class performed significantly better than students in the two-days-a-week class". They come to completely different conclusions because the course material is on different subjects and none of the students have identical backgrounds. Gromada and Shewbridge[1] concluded that the performance varies from person to person, and from class to class across OECD countries. In addition, neither of the studies by Cattaneo et al.[2] and Trout[3] are able to take the students' out-classroom studying time into account.

With more advanced technology, it is now possible to consider the students' out-classroom studying time as a variable. Nowadays, students can work on homework on their computers, and they are used to it. The rise of Coursera and other online learning services attracted a number of students. New methods of teaching are also introduced, such as online quizzes and real-time feedback exercises, aiming to help students learn course materials. With these tools available and widely used, we are able to measure students' effort outside the classrooms effectively and more accurately. The online systems are able to log information from its students, including the length of time spent on viewing content, and frequency of course material utilization. The data can give us a unprecedented perspective of the students' effort outside the classroom.

Some studies have been utilizing the data gathered from online classes. For example, de Barba, Kennedy and Ainley [4] investigated the relationships between "motivation, online participation behaviors, and learning performance". They found that the impact of value

beliefs on final grade was “mediated by quiz attempts and situational interest”. They concluded that motivation was both a factor and a result in the students online participation. Giesbers, Rienties, Tempelaar, and Gijssels [5] examined the relationship between student motivation, participation, and learning performance in web-video conferences, a method sometimes used in online classes. They found that the students who have participated in more web-video conferences had significantly higher scores on the intrinsic motivation subscale. Online classes have their innate advantages over the conventional lectures when collecting data.

However, there are few studies on the students in the higher education outside of online classes. Some classes are of small size, making the sample size small and the result unconvincing. In some other classes, the students are asked to hand in hand-written assignments, making it hard to quantify their time and effort spent. The University of Illinois at Urbana-Champaign offered an introductory-level programming class, Computer Science 101 to engineering students. The class has a capacity of 720 students and has an average of about 670 students over the past four semesters. The class also has all assignments and tests computerized, making it an ideal subject to study how the time spent by students outside the classrooms interact with their performance. In this thesis, I will examine the relationship between the amount of required student effort and their performance on the corresponding exams. I will use the information of the computerized tests and assignments to evaluate their effort and the time spent.

CHAPTER 2: RELATED WORK

2.1 THE EFFECT OF STUDENT LEARNING TIME

As discussed in the introduction, it is hard to quantify the students' learning time. The three studies[1, 2, 3] first mentioned before are all focusing on the in-class time for the students. Gromada and Shewbridge [1] suggested that the in-class time needs to be scrutinized, and it is up to whether the time is used effectively. They concluded that “[w]hen allocated instruction time is used effectively, this is an important condition to improve student learning and achievement”. Although in-classroom time is easier to quantify than outside-of-classroom time, the data gathered might still be unreliable because the instruction time is not used effectively. In the conclusion, Trout [3] suggested that the students who have less in-classroom time might have made up by spending more time studying by themselves out side of the classrooms.

It might be too difficult to track the students' time-on-task no matter in-classroom or outside-of-classroom. As pointed out before, every student has their own pace [1]. The study time varies from person to person, so does their study habit. This thesis will implement an alternative way of assessing the students' effort in addition to the time-on-task. The students' number of attempts on each quiz will be taken account into evaluating how much effort they have made.

2.2 ACTIVE LEARNING

The idea of the thesis comes from “active learning”. Active learning describes the teaching methods in which “students participate in the process and students participate when they are doing something besides passively listening” [6]. “It has commonly been applied to a diverse range of learning activities, such as practical work, computer-assisted learning, role play exercises, work experience, individualized work schemes, small group discussion, collaborative problem-solving and extended project work” [7]. It was introduced in the last century and has attracted more research in the last couple decades with the rise of computers.

Before the introduction of computerized systems, active learning was to involve the students in the learning process more directly, says Bonwell in his book “Active learning: creating excitement in the classroom”[8]. Instructors design some interactive activities to help students understand the material better or just to get more of their attention. There have been numerous experiments done on active learning, and how much it is better than the

traditional ways. Some are just changing the way the assignments work. Renkl et al. used “fading examples” in assignments. Fading examples are “successive integration of problem-solving elements into example study until the learners solved problems on their own” [9]. Every time, a part of the solution is taken out so that at the end the students are asked to finish the assignment with no help provided. Other studies are more complicated and involve interactive activities in the classrooms, including “investigational tasks, small group discussion, computer-assisted learning and extended project work” [7].

With computerized systems taking a big place in education, it is more frequent that active learning techniques are used with the new technologies. In contrast to elementary math games making learning more attractive, higher education creates new ways for students to learn. For example, “creative learning spaces” (CLSs) are implemented by Lu et al. at the University of Arizona [10]. Students collaborate with each other at a round table surrounded by screens and whiteboards, and instructors teach the class through microphones and illustrations. In some other methods, students are instructed so that they teach each other and design their own questions. In a study conducted by Al-Hammoud et al. [11], it is found that over 90% of students “found the process of creating the questions for the quiz to be helpful in reviewing the material learned in the class and the process of taking the quiz to be helpful in learning the material” when engagement and collaboration in the classroom are encouraged.

The online quizzes are employed in CS 101 not only because of the experiment. The motivation of the online quizzes is to encourage the students to actively think about the concepts introduced. They will also get more practice during the process. The quiz system will integrate more active learning principles in future.

2.3 QUIZZES IN ACTIVE LEARNING

In order to get their students more involved, the researchers design many different kinds of activities to encourage the students, including quizzes [12, 13, 14]. For example, Vinney et al. [12] designed a set of quizzes for the students on the subject of introductory voice disorder concepts and concluded that “[their] mini quiz games and MQGs with traditional study, but not traditional study alone, showed better results for long-term retention than no study.” Their mini quiz games had a positive effect on the students’ learning. Nonetheless, Wang[13] also designed a quiz system called “GAM-WATA”, and pointed out that “different types of formative assessment have significant impacts on e-Learning effectiveness and that the e-Learning effectiveness of the students in the GAM-WATA group appears to be better”. Both of the two experiments which are based on their respectively new quiz systems appear

to have a positive effect on the students' performance. It is reasonable to suppose that extra work assigned to the students helps their performance. In reality, there is no doubt that quizzes of any kind helps students. Blter, Enstrm, and Klingenberg[15] studied the students' performance when they gave them quizzes with only correct or incorrect feedback. They found out that "short quizzes using generic questions with limited correct/incorrect feedback on each question, have positive effects when administered early in courses". The findings show that even quizzes with minimal feedback are useful. Therefore, it is reasonable to expect the CS 101 students will benefit from the newly-employed quiz system.

CHAPTER 3: EXPERIMENT SETUP

3.1 CLASS: CS 101 INTRO TO COMPUTING

At the University of Illinois, Computer Science 101 is an introductory-level programming class for engineering students offered every semester. It has a capacity of 720 students and an average of 680 students over the last four semesters. It is a prerequisite for many engineering and science classes and a requirement for most engineering and science degrees.

3.1.1 Student backgrounds

Since it is an introductory-level class early in long prerequisite chains, the students are mostly freshmen and sophomores. The distributions vary between spring and fall semesters. In Spring 2018, the distributions of students by year is shown in Figure 3.1.

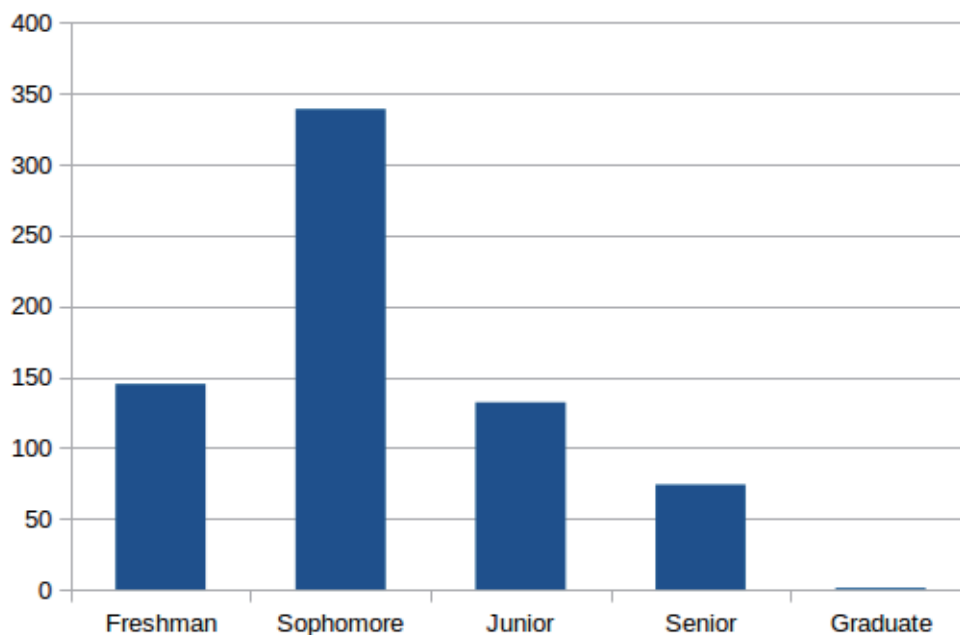


Figure 3.1: Student year distributions

A total of 691 students in CS 101 Spring 2018 come from various majors. Figure 3.2 shows the distribution of majors. The top two majors are Civil Engineering and Mechanical Engineering because CS 101 is required by these large programs. To better look at the distributions of students, we can group those majors from College of Engineering as “engineering”. Figure 3.3 illustrates the percentage of engineering students compared to students

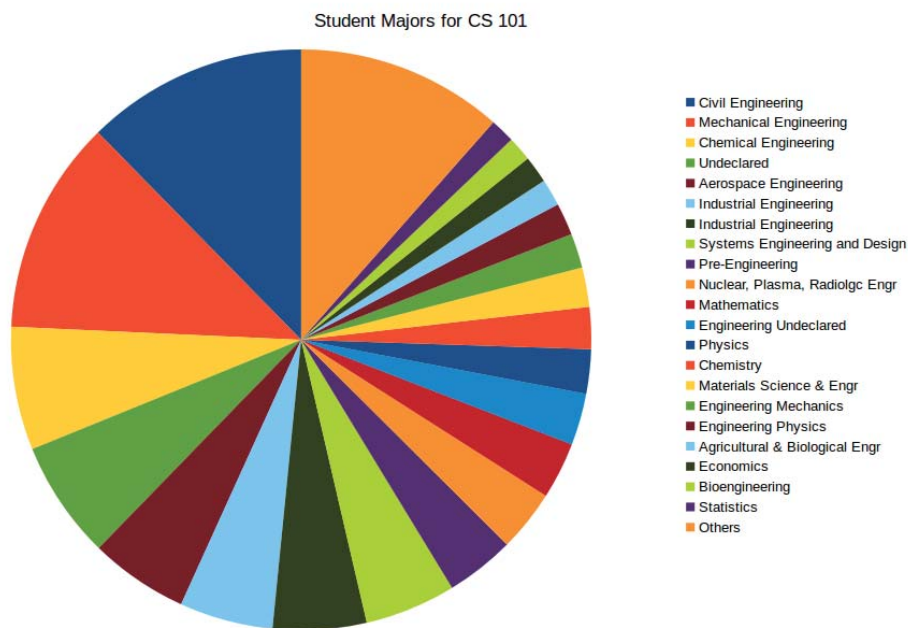


Figure 3.2: Student major distributions

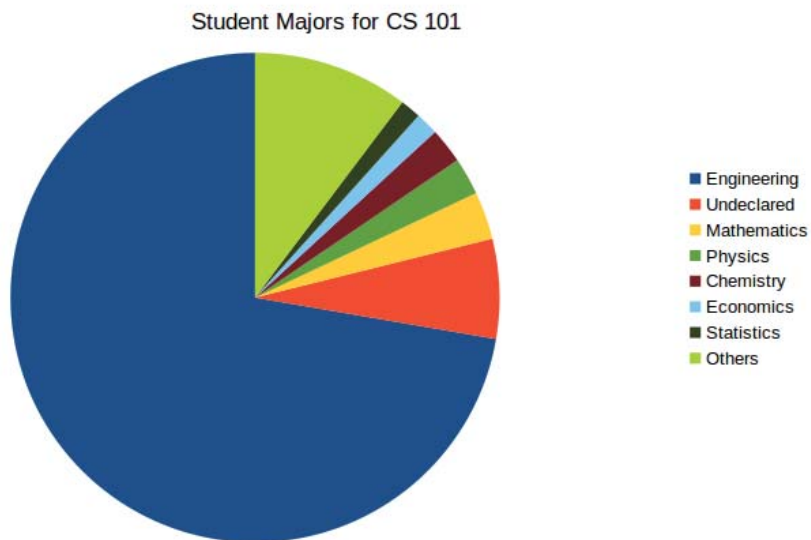


Figure 3.3: Student major distributions simplified. Physics and chemistry are usually also considered engineering majors.

in other majors. Physics and chemistry are usually considered “engineering”, but they are in the College of Liberal Arts and Science at the University of Illinois. They made up about 5% of the total class.

With 72% of the class being an engineering students, this subgroup of students differ from the overall sampling from the university. A few of the students also have experience in coding from high school or other classes.

In an entrance survey of the class, 338 of 708 students identify themselves as having programmed before, while the other 370 students have no experience in Figure 3.4. In the same survey but conducted on students in Fall 2017, about the same percentage of students report that they have no experience at all. There are more survey samples than students in the class because some students have dropped the class in the semester. However, this fact doesn’t affect the conclusion that the more than half of CS 101 students has no experience in coding.

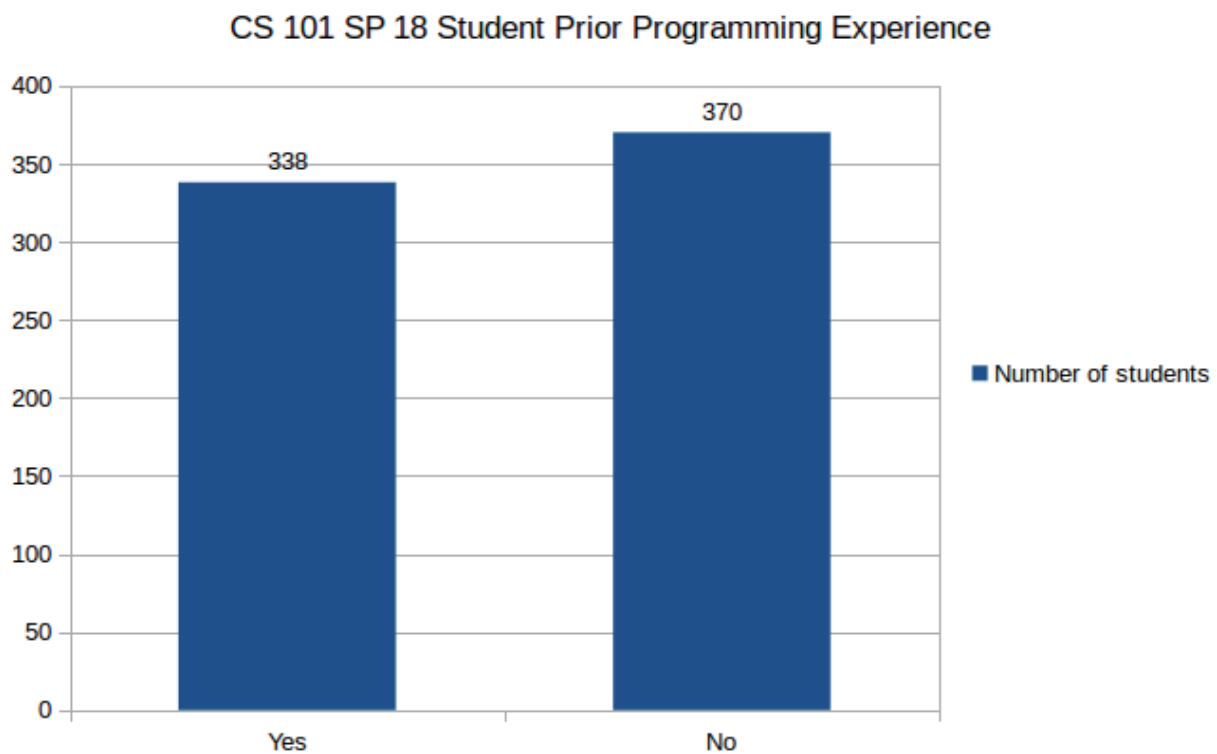


Figure 3.4: Student prior programming experience

3.1.2 Course content

The course covers spreadsheets, Python, and MATLAB from the conventional perspective of teaching programming languages. In its total 29 lectures, 4 of them are on spreadsheets, 19 on Python and 6 on MatLab.

CS 101 assumes zero programming experience from students, as the survey shows. In the first weeks of CS 101, computational thinking is introduced to the students via spreadsheet functions. Later in the semester, the first half of the Python part covers all basic programming principles: variables, functions, control flows, and data structures. The second half emphasizes on more advanced principles via libraries and functions: *numpy* for numeric computations, *matplotlib* for plotting, *numpy.random* for random distributions, and *itertools* for brute forcing as well as *scipy.optimize* for optimization. The MATLAB part strengthens students' knowledge by having them apply the intro programming concepts they learnt in Python to a new language. It also highlights the difference between two modern scripting languages.

3.1.3 Course structure

The whole class meets for a 50-minute lecture twice every week in an auditorium. Students are also required to attend a two-hour lab section of 40 students every week under the guidance of teaching assistants and course aids. There are also homework assignments due every week and online quizzes due the day after every lecture. There are six exams throughout the course. Each exam is weighted equally and is 50 minutes long.

In the lab section, students work on hands-on problems on Engineering Workstations (EWS). They write their own functions according to the given questions or fill in the blank lines within the scaffolding code blocks. Typically there is little variation between students' solutions to the lab problems, and most students finish them within the given time. In the second part of the class, they are also set up to do pair programming and encouraged to swap between the “observer” and the “driver” [16].

3.1.4 After-lecture quizzes

The quizzes are designed to help students digest materials learned from lectures. They are due the day after the lecture to ensure that students refresh their knowledge shortly after lecture. Multiple studies have been done on the effect of after-lecture quizzes. In 2014, instructors in University of California, Irvine surveyed students and “students reported that

having weekly “low stakes” quizzes and reviewing them in class helped them understand key concepts better” [17].

The quizzes in CS 101 generally recap on the previous materials and are short as well as low stakes. Each quiz consists of ten questions, mostly multiple choice.

3.2 PLATFORM: PRAIRIELEARN

PrairieLearn is an online learning system with “adaptive scoring” and “randomized problem variants” [18]. It provides questions through a web interface and grades the submission from students almost instantly. For quizzes, students need to answer each question correctly a fixed number of times to get full points. The course administrators can easily configure the assignments by adjusting the types and difficulty of the questions.

3.2.1 Adaptive scoring

In PrairieLearn, students are awarded twice the points when they make a streak in answering questions. If a student starts a new question and answers it correctly twice in a row, they will be awarded one point for the first correct submission, and two points for the second. If a student answers it correctly three times in a row, they will get a total of six (1+2+3) points. Incorrect answers give zero points and end the streak but give no punishments on the score.

3.2.2 Randomized variants of problems

It would be meaningless to make students repeat one question multiple times if that one question doesn’t change. We have to generate randomized variants of problems every time so that students won’t see the same problem the second time.

We implement randomized variants mostly by using *random* package from Python. For questions with numbers, random numbers are generated for input and output is calculated in the back end. For questions without numbers, several variants are preset and randomly retrieved using *random.choice*.

3.3 EVALUATION: CBTF TESTING

Exams in CS 101, like a number of other large classes on campus, run in Computer-Based Testing Facilities (CBTF). [19]

3.3.1 Introduction

A professor can register their class with CBTF for computerized testing. He can set a range of days, usually four or five, for one exam. The student will be able to schedule the exam at their preferred time in advance. The proctors in the exam room will check students' identifications and assign them to random workstations. Each workstation is specially configured so that no unauthorized applications may be used and no external websites can be accessed. The system keeps track of the remaining time of the exam. The proctors usually have no knowledge about the materials the students are being tested on, given that multiple exams are running concurrently in the same room.

3.3.2 Testing with Relate

Although in CBTF, the exams in CS 101 are not conducted with PrairieLearn, but a similar online learning system called Relate. In contrast to PL, Relate is not able to parameterize problems and offers a less accessible interface to browse through questions. Despite of it, the exams are conducted in Relate because students are more familiar with the coding interface in Relate, which is identical to that in homework.

Students also get a chance to participate in a non-mandatory "Exam 0" which is essentially a syllabus quiz. Exam 0 has the same set up as the following exams and should get students familiar with the testing environment.

3.3.3 Academic Integrity in CBTF

When a test is distributed across several days, there is always suspicion of widespread cheating. For instance, a group of students take the exams earlier so that they can give out the questions to others who haven't taken the exam. In the traditional tests, students would have to get instructor's approval to take the exam at a different time, no matter earlier or later. However, the scheduling of CBTF testing is very flexible. In most scenarios, students will be able to choose a time slot from a four-day period.

Although students tend to select a later time slot [20], work by Chen et al. [21] suggests that cheating is not wide spread in the CBTF exams. The variation of students' performance over time in an exam period is explainable [22] but won't be discussed here.

In addition, the screens in CBTF are polarized, making it hard to spot one's neighbor's answers. The seat assignment algorithm also assigns students taking tests of different classes to be neighbors to further reduce possibility of cheating.

CHAPTER 4: EXPERIMENT

4.1 FIRST STAGE EXPERIMENT

The experiment aims to find the correlation between the students' performance and their effort spent in the class. The performance is evaluated simply based on their exam scores. The effort is evaluated by the number of attempts they have made in quizzes.

4.1.1 Method

In the first stage experiment, I randomly assigned students into two groups by their last digit of university identification number (UIN). In the period of seven lectures spanning four weeks starting from the first lecture on Python, after each lecture, both groups are assigned the same quiz problems on programming principles mentioned in section 3.1.2.

Students in group A are required to get a minimum of **two** points for each question, and students in group B are required to get **three**. In section 3.2.1, it explains how the points are calculated. Group B has an overall slightly more workload than group A, but the students in both groups will get full points with two consecutive attempts.

The quizzes rolled out to students over the time period from the fifth lecture to the eleventh lecture. As mentioned in section 3.1.2, this is where Python is first introduced in class. This particular part of class is chosen because these quizzes will help students better in the beginning by reinforcing basic concepts in programming. Every quiz is due at the next day after the corresponding lecture. This is to ensure that all students still have fresh memories about the content.

The exam takes place after the eleventh lecture in the CBTF. Section 3.3 shares some details about how exams are handled in CS 101.

After collecting the data for quizzes and the exam, I sought to find correlations between them using data analysis and significance tests. To be more specific, I examined the quiz average duration vs exam score and the quiz average attempts vs exam score.

4.1.2 Results

After omitting all students that have dropped the class or failed to take the exam, I gather the exam scores. I also calculate the average number of attempts for quiz problems from each student, omitting those that completed less than five quizzes and those that have completed

the wrong version of the quiz. After cleaning, group A has 198 samples where group B has 193.

Table 4.1: Exam 2 scores of group A and group B. A t -test gives no significance between two groups.

Exam 2 Scores	Group A	Group B
1st Quantile	85	85
Median	100	95
Mean	87.12	86.27
3rd Quantile	100	100
Standard Deviation	21.08	22.51

If we take a closer look at the exam scores in Table 4.1, a t -test on exam scores of two groups give us a p-value of 0.3009. Therefore, we can not reject the null hypothesis, and there is no significant difference between two groups in exam scores. The result here is not enough to support that the grouping has any effect on their exam scores.

Students in group A has an average score of 87.55 on the exam while group B has 85.49. Although it is insignificant after being tested with a t -test, it is still interesting seeing a group that are required to do less work got slightly better scores in the exam. It could be a statistical error.

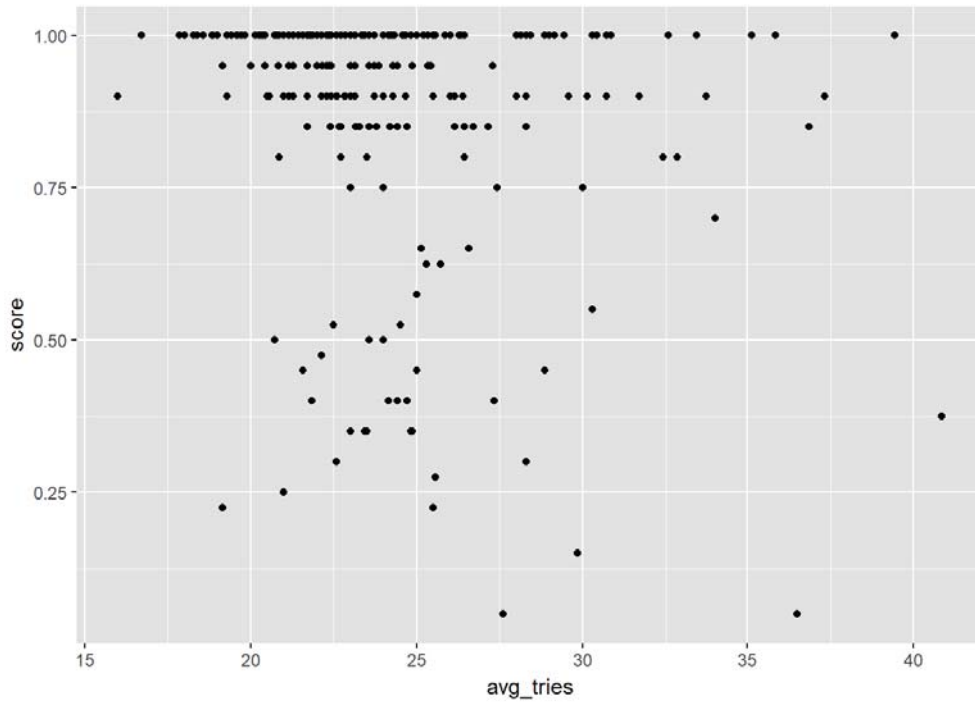


Figure 4.1: Average number of attempts on quizzes vs exam scores in group A

Figure 4.1 shows a scatter plot relating the average number of tries to the exam scores. There is no strong correlation visually. Taking account of the exam scores and the average attempts of quizzes separately, the exam has a median of 95, so it is no surprise from the dense cluster of students getting 100% on the exams. With ten questions each quiz, it is also no surprise from the dense cluster of students in the range of first quarter 21 to third quarter 25. The same analysis is done on group B in Figure 4.2, there is no significant correlation observed either. The same dense cluster in the exam score is also observed. Since each student is required three points for each question, which is slightly more than group A, it is also reasonable that the dense cluster expanded a little, and is in the range from 21 to 27.

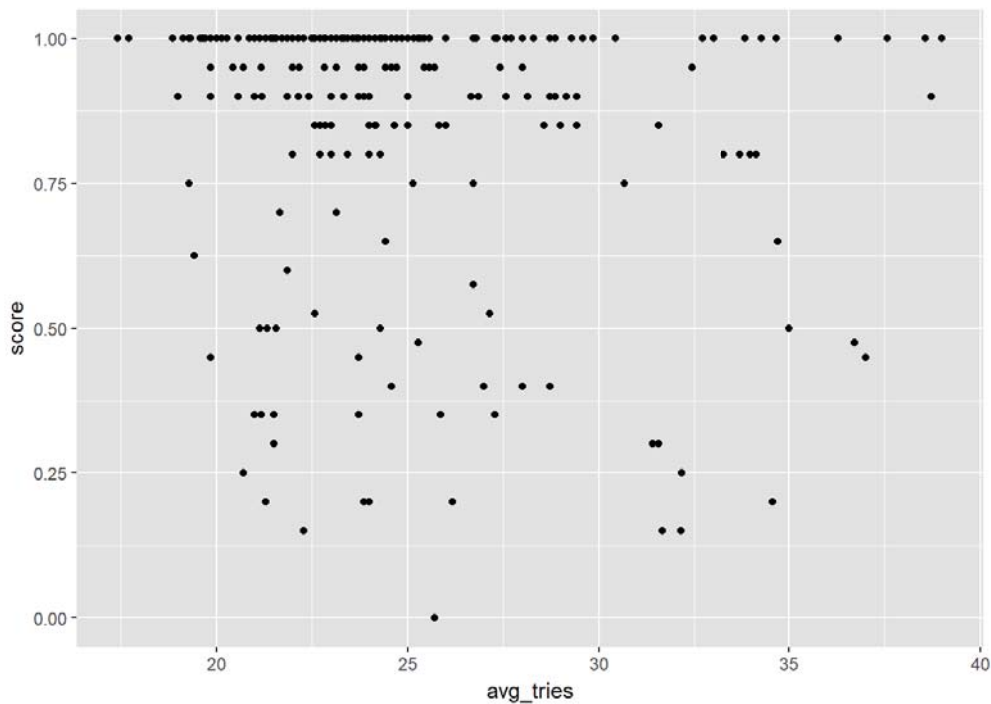


Figure 4.2: Average number of attempts on quizzes vs exam scores in group B

In general, group A does less work than group B does in quizzes. We can combine the two groups in one figure and see if there is any correlation between them. In Figure 4.3, a slight declining line is fitted for both groups. In both groups, students who have fewer attempts on quizzes get better scores. But it is important to point out that the fitted line only decline about 10% over the entire range of x-axis. It is clear to see that two groups have no distinguishable differences at all on the fitted line given that they are twined to each other.

In order to fully examine if any of the quiz has an effect on students' exam scores, I created a linear model with the numbers of quiz attempts and the exam scores to see if there is a

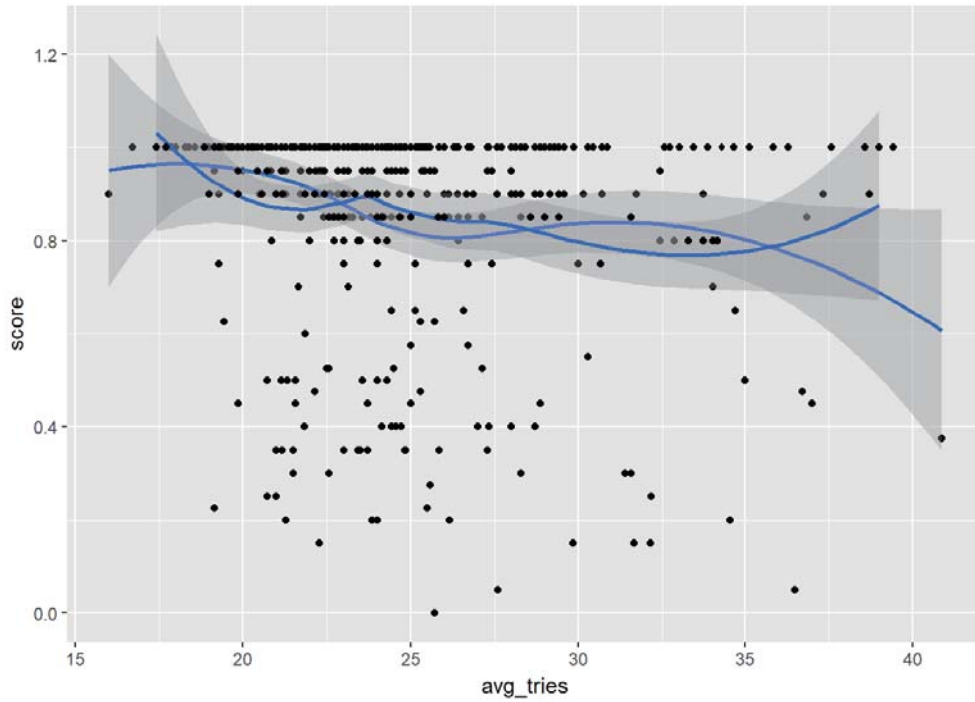


Figure 4.3: Average number of attempts on quizzes vs exam scores with fitting

way to predict the exam scores. For group A, the outcome is shown in Table 4.2. It is clear that even the lowest one of them has a p-value of 0.212. There is little or no value in looking at whether there are positive or negative correlations on each individual quiz due to the dominant p-value on the intercept. The same outcome is also retrieved from group B in Table 4.3. The two groups are almost identical.

The same analysis is done on the minutes they spent on quizzes and their exam scores. In Figure 4.4, it shows no correlation at all with a flat line for both group A and group B on the same level. It appears that group A and group B are spending about the same time on quizzes, even with the difference of the required points. The linear modeling proves the finding more concretely. In Table 4.4 and Table 4.5, the time has an even smaller estimate and t value than the attempts after normalization.

To summarize the results of the first experiment:

- There is no difference in exam scores between two groups.
- There is a slightly negative correlation between number of attempts on quizzes and exam score. Students have fewer attempts perform better.
- There is no correlation between time spent on quizzes and exam scores.

Table 4.2: Group A linear modeling results of number of quiz attempts vs exam scores using *caret train* with *lm*

	Estimate	Std. Error	<i>t</i> value	$\Pr(> t)$
(Intercept)	1.0757574	0.0803181	13.394	$< 2e - 16$
Quiz4	-0.0003788	0.0023886	-0.159	0.874
Quiz5	0.0013650	0.0022322	0.612	0.541
Quiz6	-0.0021683	0.0021750	-0.997	0.320
Quiz7	-0.0001365	0.0019857	-0.069	0.945
Quiz8	-0.0011876	0.0015223	-0.780	0.436
Quiz9	-0.0038655	0.0030869	-1.252	0.212
Quiz10	-0.0024992	0.0022443	-1.114	0.267

Table 4.3: Group B linear modeling results of number of quiz attempts vs exam scores using *caret train* with *lm*

	Estimate	Std. Error	<i>t</i> value	$\Pr(> t)$
(Intercept)	0.875490	0.013088	66.894	$< 2e - 16$
Quiz4	-0.002362	0.014897	-0.159	0.874
Quiz5	0.010410	0.017023	0.612	0.541
Quiz6	-0.016888	0.016941	-0.997	0.320
Quiz7	-0.001044	0.015184	-0.069	0.945
Quiz8	-0.012068	0.015469	-0.780	0.436
Quiz9	-0.017074	0.013635	-1.252	0.212
Quiz10	-0.015284	0.013725	-1.114	0.267

- There is no difference in correlations of either average attempts or average time between two groups.

From the large p-value on the intercept in Table 4.2, Table 4.3, and the score distribution on Exam 2 in Figure 4.3, it looks that the exam is too easy or at least not distinguishable enough for students to show how they have mastered the materials. The median is also too high that half of the class gets more than 95% in Exam 2. All these impact the data significantly.

It might also be the case that repetitive work is not helping their learning, since group B has a lower average Exam 2 score than group A. However, it does not have enough support and more experiments are needed to gather proof.

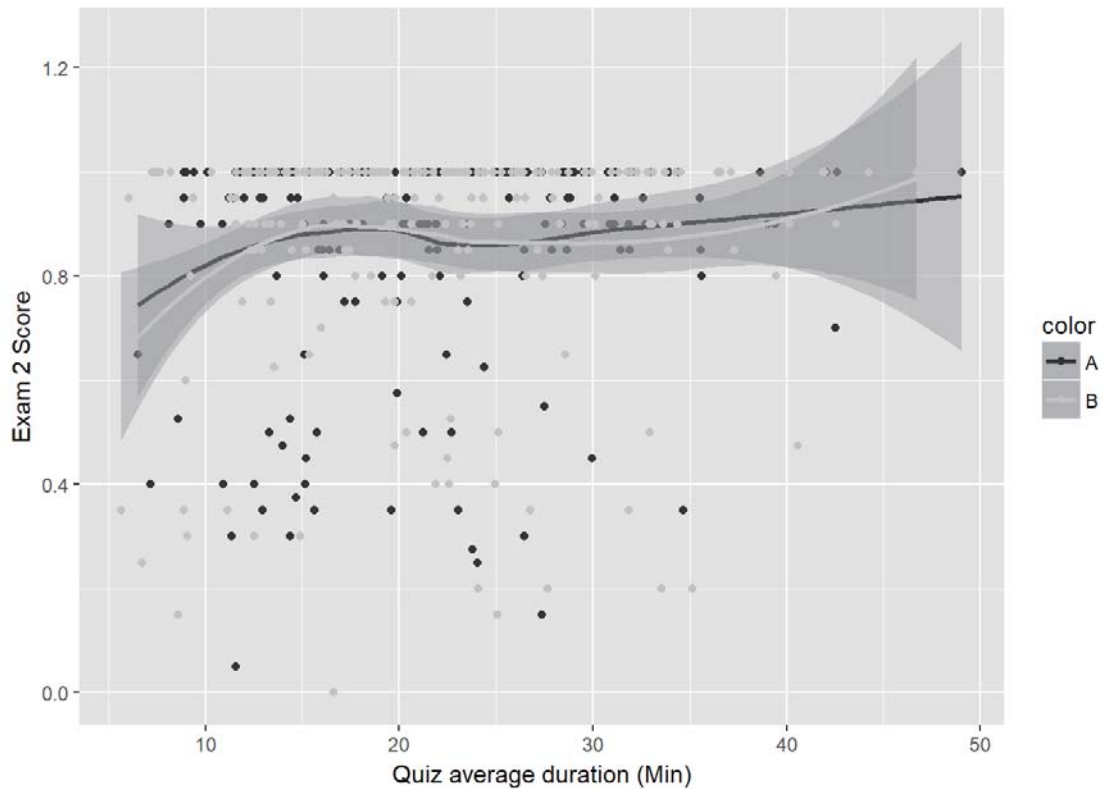


Figure 4.4: Average minutes spent on quizzes vs exam scores with fitting

4.2 SECOND STAGE EXPERIMENT

The “low stakes” quizzes and the easy exam are not distinguishable enough. After retrieving the result for the first stage, I designed the second stage experiment with a more challenging exam, Exam 3, and higher stake quizzes.

4.2.1 A more challenging exam

As observed in the previous table, both groups have a 0% p-value of the intercept, meaning that the exam scores for everyone is too high so that there is no distinguishing can be drawn here. Due to scheduling with CBTF this semester, there are only two weeks apart to the next exam. By convention, it has always been a more challenging exam since we have covered more on the materials, including lists and dictionaries. This exam might serve better as an evaluation of how students have learnt the materials from the class.

A comparison between the two exams is drawn in Table 4.6. Exam 3 is a lot more difficult than the previous one, having a relatively low median of 71%. A curve is provided for Exam 3, but grades before curving are used in the analysis.

Table 4.4: Group A linear modeling results of average minutes on quizzes vs exam scores using *caret train* with *lm*

	Estimate	Std. Error	<i>t</i> value	Pr(> <i>t</i>)
(Intercept)	8.535e-01	3.485e-02	24.489	2e-16
Quiz4	-1.499e-03	9.661e-04	-1.551	0.1224
Quiz5	1.511e-04	6.408e-04	0.236	0.8139
Quiz6	5.952e-04	5.997e-04	0.992	0.3222
Quiz7	1.024e-03	8.088e-04	1.266	0.2070
Quiz8	-3.048e-04	7.847e-04	-0.388	0.6981
Quiz9	3.525e-05	8.079e-04	0.044	0.9652
Quiz10	1.380e-03	1.090e-03	1.266	0.2069

Table 4.5: Group B linear modeling results of average minutes on quizzes vs exam scores using *caret train* with *lm*

	Estimate	Std. Error	<i>t</i> value	Pr(> <i>t</i>)
(Intercept)	0.867619	0.015375	56.431	2e-16
Quiz4	0.007337	0.016720	0.439	0.6613
Quiz5	0.003413	0.021008	0.162	0.8711
Quiz6	0.027194	0.020485	1.328	0.1859
Quiz7	-0.036601	0.019713	-1.857	0.0649
Quiz8	-0.026553	0.019502	-1.362	0.1749
Quiz9	0.030178	0.020528	1.470	0.1431
Quiz10	-0.005328	0.020580	-0.259	0.7960

4.2.2 Method

I disregarded the previous exam score as an evaluation of students and used the new exam instead. I also added two quizzes in the time period between two exams. In short, I used the same approach as above, but added more gathered data. The same questions from the first experiment are asked here as well.

4.2.3 Results

There is no difference observed between group A and group B Exam 3 scores in Table 4.7. A t-test of exam scores between two groups gives a p-value of 0.49, where group A has an average score of 70.64, and group B 69.25.

There is some correlation to observe in Figure 4.5. There is apparently a declining in the exam scores as the average number of quiz attempts increases on the left side of the graph. The correlation is not obvious any more after average of 23 attempts possibly due to decreasing density. Also, there is a subgroup of the top 20% students that are doing very

Table 4.6: Exam Scores Distribution

	Exam 2	Exam 3
1st Quantile	85	55
Median	95	71
Mean	85	70.56
3rd Quantile	100	92
Max	100	100
Standard Deviation	21.77	24.54

Table 4.7: Exam 3 scores of group A and group B. A t -test gives no significance between two groups.

Exam 2 Scores	Group A	Group B
1st Quantile	58.25	53
Median	73.5	70
Mean	71.4	69.69
3rd Quantile	92	92
Standard Deviation	24.38	24.74

well in the quizzes and the exams as shown in the left declining part.

Within group A, several numbers of quiz attempts show some significance on a flat slope as shown in Table 4.8. Within group B in Table 4.9, no significance is shown except on the intercept. It could be due to the requirement that made group B have a higher variation in the attempt numbers, which is 4.12 compared to A's 3.83.

Table 4.8: Group A linear modeling results of number of quiz attempts vs Exam 3 scores using *caret train* with *lm*

	Estimate	Std. Error	t value	$\Pr(> t)$
(Intercept)	4.296913	0.465683	9.227	$< 2e - 16$
Quiz4	-0.014953	0.013054	-1.145	0.25347
Quiz5	0.035770	0.012077	2.962	0.00345
Quiz6	-0.003299	0.011729	-0.281	0.77880
Quiz7	-0.008834	0.011295	-0.782	0.43513
Quiz8	-0.001033	0.008724	-0.118	0.90586
Quiz9	-0.024903	0.015865	-1.570	0.11818
Quiz10	-0.029330	0.013973	-2.099	0.03714
Quiz11	-0.011734	0.014165	-0.828	0.40851
Quiz12	-0.016782	0.015555	-1.079	0.28201

Given that there are still no difference between two groups, it is safe to say that the extra work does not contribute to make their scores better. Individual quiz attempt has a slope

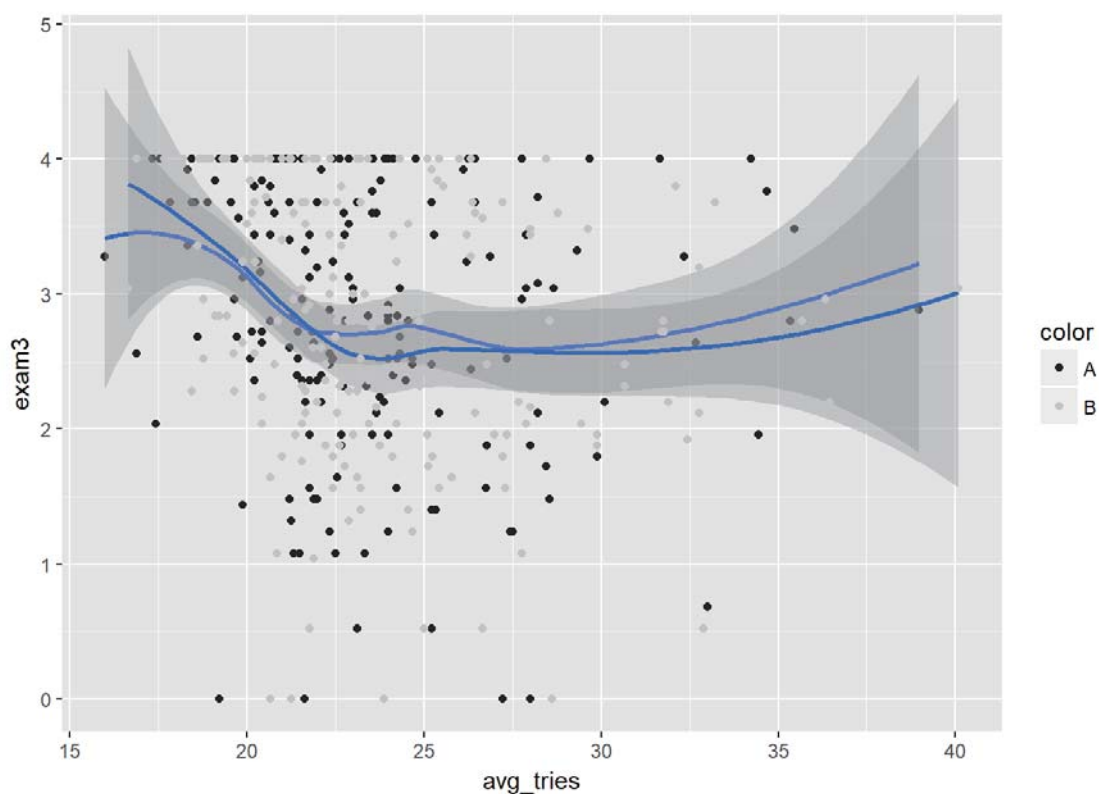


Figure 4.5: Average number of attempts on quizzes vs Exam 3 scores. Exam scores are out of 4. Average quiz attempts are from 16 to 39.

too low to be considered effective as well, so it is also difficult to predict which quiz helps students most. Collectively, as observed in Figure 4.5, there is a strong negative correlation between average number of attempts on quizzes with the exam score in class.

In Figure 4.6, the relation between the average minutes spent on quizzes and the Exam 3 scores is shown. Compared to Figure 4.4, it is more distributed across the exam scores, but it is still a flat line. The same result is also retrieved from the linear modeling in Table 4.10 and Table 4.11.

4.3 THIRD STAGE EXPERIMENT

4.3.1 Exam-format quiz

The numbers of quiz attempts are having a higher variation because there are no punishments in making the wrong choice. Although it was designed to be low stakes at first, it might help us to see what raising the stakes can yield.

Table 4.9: Group B linear modeling results of number of quiz attempts vs Exam 3 scores using *caret train* with *lm*

	Estimate	Std. Error	t value	$\Pr(> t)$
(Intercept)	2.787565	0.071315	39.088	$< 2e - 16$
Quiz4	-0.129398	0.089273	-1.449	0.149
Quiz5	0.008758	0.096862	0.090	0.928
Quiz6	0.038395	0.094091	0.408	0.684
Quiz7	0.009840	0.100686	0.098	0.922
Quiz8	-0.101673	0.090212	-1.127	0.261
Quiz9	-0.112165	0.076617	-1.464	0.145
Quiz10	-0.031264	0.074240	-0.421	0.674
Quiz11	0.019408	0.086188	0.225	0.822
Quiz12	0.018545	0.084487	0.219	0.827

Table 4.10: Group A linear modeling results of minutes in quizzes vs Exam 3 scores using *caret train* with *lm*

	Estimate	Std. Error	t value	$\Pr(> t)$
(Intercept)	2.8039745	0.1632916	17.172	$< 2e - 16$
Quiz4	-0.0053623	0.0045267	-1.185	0.2376
Quiz5	-0.0024855	0.0030024	-0.828	0.4087
Quiz6	0.0037140	0.0028100	1.322	0.1878
Quiz7	-0.0073496	0.0037895	-1.939	0.0539
Quiz8	-0.0033405	0.0036767	-0.909	0.3647
Quiz9	0.0023315	0.0037854	0.616	0.5387
Quiz10	0.0043185	0.0051064	0.846	0.3987
Quiz11	0.0057094	0.0041109	1.389	0.1664
Quiz12	0.0028701	0.0046531	0.617	0.5381

In the exam format, each problem in a quiz is assigned a list of points. In contrast to the format in section 3.2.1, students will be deducted points if they made a mistake. Each question has a maximum of 10 possible points, and every quiz has ten problems, making the maximum score possible to be 100.

For students in group A, every time they submit an incorrect answer, the system will notify them and ask for another input. It also reduces the maximum possible point by one. For example, if one student makes two mistakes before they get the correct answer, they only gets 8 out of 10 possible points.

For students in group B, an incorrect submission reduces the maximum possible point by two. So that if one student makes two mistakes before they get the correct answer, they only gets 6 out of 10 possible points.

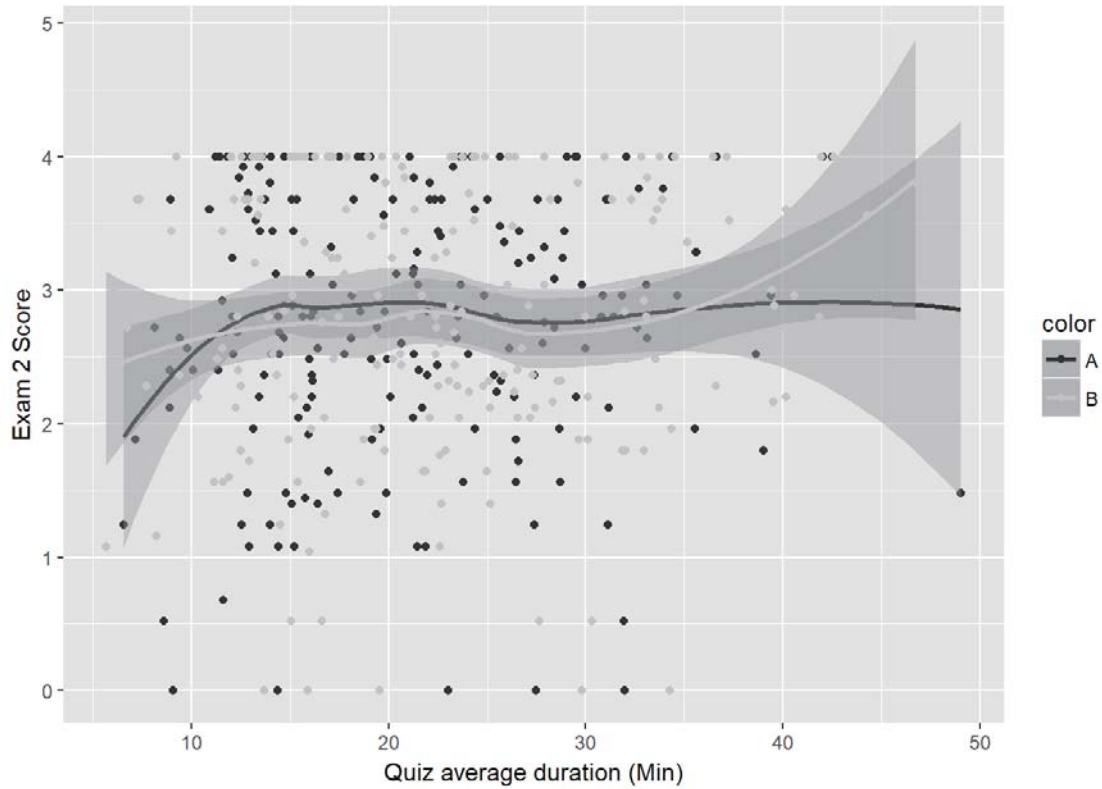


Figure 4.6: Average minutes spent on quizzes vs Exam 3 scores. Exam scores are out of 4. Average quiz attempts are from 4 to 50.

To offset the difference between two groups, students in group A are informed that they will get 10 extra points counting towards their final grade for each quiz, and students in group B get 15. In fact, these quizzes are graded by completion after the experiment ends.

4.3.2 Method

Since the exam-format quizzes no longer require multiple attempts on random variants (section 3.2.2) of the same questions, we can not compare the exam scores with the number of attempts. Instead, we draw the correlation between the raw quiz scores and the exam score. The quiz score actually reflects how much effort student spend since they will most likely keep trying until they get the right answer and stop there.

The overall trend should be of no surprise that those who have a higher quiz score will get a higher exam score. It might be interesting to see what differences two groups make, and how “higher” the stakes will affect their performance. In the experiment setup, group B has a higher stake than group A. It will also be interesting to examine how students respond to higher stakes if we only look at the quiz scores themselves.

Table 4.11: Group B linear modeling results of minutes in quizzes vs Exam 3 scores using *caret train* with *lm*

	Estimate	Std. Error	<i>t</i> value	Pr(> <i>t</i>)
(Intercept)	2.8633539	0.1403844	20.397	$< 2e - 16$
Quiz4	-0.0060049	0.0038540	-1.558	0.1208
Quiz5	0.0008991	0.0030015	0.300	0.7648
Quiz6	0.0041486	0.0025999	1.596	0.1122
Quiz7	-0.0054455	0.0033319	-1.634	0.1038
Quiz8	-0.0068691	0.0027794	-2.471	0.0143
Quiz9	0.0043874	0.0024251	1.809	0.0720
Quiz10	-0.0046013	0.0034834	-1.321	0.1881
Quiz11	0.0028777	0.0034768	0.828	0.4089
Quiz12	0.0085640	0.0044153	1.940	0.0539

4.3.3 Results

Table 4.12: Quiz 13 - 16 average score of group A and group B. A *t*-test gives 0.000228 between two groups.

Average Score	Group A	Group B
1st Quantile	91.75	93.25
Median	95	95.25
Mean	92.52	94.7
3rd Quantile	97	97.25

Table 4.13: Quiz 13 - 16 number of mistakes for group A and group B.

Average Score	Group A	Group B
1st Quantile	3	2.75
Median	5	4.75
Mean	7.48	5.3
3rd Quantile	8.25	6.75

Group B has a better average than group A in the average score of Quiz 13 - 16. A *t*-test on the number of average scores of quizzes gives 0.000228 with group A having average score of 92.52 and group B having average score of 94.70. The null hypothesis is rejected in this case. Students in group B are doing better probably because of the higher stakes. To be more explicit, students in group A makes 7.48 wrong attempts on average while students in group B makes only 5.29. That means group A students make 41% more mistakes than group B. Similarly, the students in group B spend more time on quizzes as well. Group A

has an average of 20 minutes over four quizzes, while group B has 22.37. The p-value is 0.045 after a t -test.

The one that has a deeper “V” shape is of group A in Figure 4.7. If we focus more on the right half of the graph, we can see that A has a steeper slope between 85 to 100. However, the quiz score is not reflective of how students’ performance due to difference in calculating scores between two groups in section 4.3.1. It is better to look at how many problems they have done wrong.

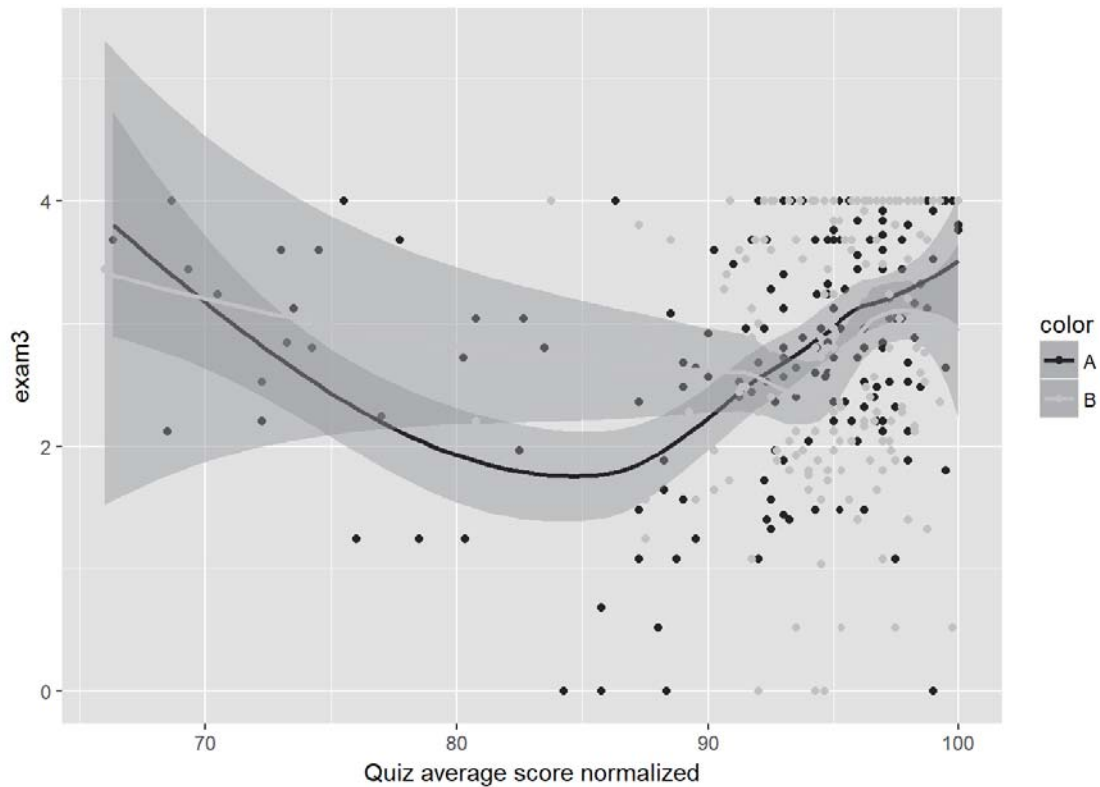


Figure 4.7: Quiz 13-16 scores vs Exam 3 scores with fitting

In Figure 4.8, the quiz scores are calculated using the same metric between both groups. It might be more straightforward if we view it as the average incorrect attempts vs Exam 3 scores in Figure 4.9. There are fewer mistakes made in group B as in the left part of the graph, the grey curve is below the black curve. This is a stand alone finding for average number of incorrect attempts.

The trend in the quiz scores from 90 to 95 is predicted. The quiz average score is proportional to the exam score. In the range of quiz scores from 95 to 100, there are fewer samples to observe. Therefore, the deviation is not significant enough to observe.

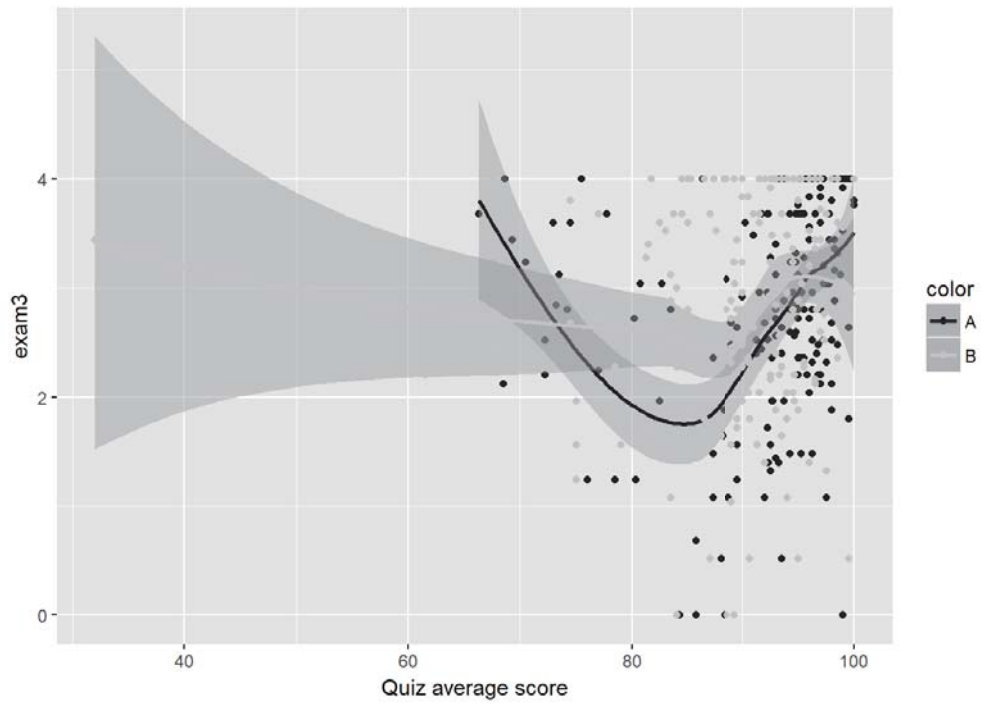


Figure 4.8: Quiz 13-16 scores normalized vs Exam 3 scores with fitting

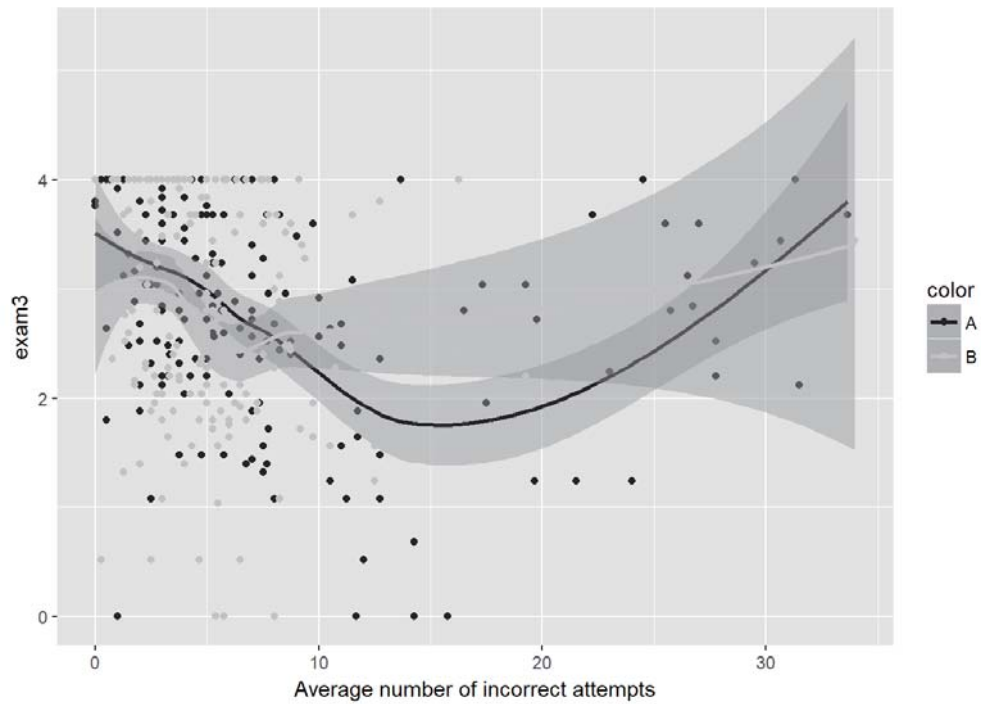


Figure 4.9: Quiz 13-16 average incorrect attempts vs Exam 3 scores with fitting

CHAPTER 5: DISCUSSION

5.1 SUMMARY OF THE EXPERIMENTS

The first stage (section 4.1.2) doesn't show a difference between two groups or a correlation within each group. To be more specific, setting different points required to achieve by students has no effect on their exam scores. The number of attempts students make in any quiz has no effect on their exam scores.

From the results of the first experiment, for students' sake, the course administrators should just lower the amount of points required because it doesn't affect their learning that how much work they have done. It also suggests that if the quizzes to be more informative in the feedback, these assignments can help students better.

The second stage (section 4.2.3) shows that the changing metric of students' performance distinguishes the difference between student groups. There are about 20% students who finish quiz using fewer attempts and get a high score on the exam. There is a significant negative correlation between students' average attempts on quizzes and their exam score in the dense area of averaging 18 to 23 tries. Each individual quiz's number of attempts has no effect on the exam score in both two groups, given that the intercept still has the highest p-value.

From the result of the second experiment, it merely proves that the students who perform better on the exam spend less attempts on quizzes. It doesn't necessarily prove this system to be effective, but might support to its legitimacy. The fact that this group of students spend less attempts but still get better scores supports us to keep doing this in some ways. It will reduce the "chore" for those who already know the drill, but gives more exercise to those who need them.

The third stage (section 4.3.3) also shows that having a higher stake quiz makes students more cautious about their submission, resulting in a significant 41% difference between the average numbers of times they make a wrong choice.

We learnt from the third experiment that the students perform better when the stakes are higher. However, it is impossible to determine the reasons of it directly. Given that it is an unproctored quiz, many factors can come in play, and it cannot be directly reflective of their knowledge on the material. However, since the quiz actually allows them to look up content from the lecture notes or online, I am optimistic to believe that they spend more effort in working on quizzes.

Both the first and the second stage experiments show that the time each student spent

on quizzes has no correlation with the exam scores at all. The amount of time spent in each quiz is not giving any useful information possibly because everyone has their own pace, and a lot of students incline to take constant breaks after starting the quiz or multi-tasking. Without enforcing a short time period (under 20 minutes), there is no reliable information we can retrieve on one's time spent effectively. Cattaneo et al[2] and Trout [3] have similar results stating that the individual study time is highly heterogeneous.

Individual question or quiz's effect on students' scores is not considered because it is not meaningful when analyzing the behavior of students. Each question's difficulty and discrimination are calculated and will be used in the following semesters. It will help construct a better question bank over the years.

CHAPTER 6: CONCLUSION

It is hard to objectively evaluate one student's knowledge in one class. There are multiple fields that an introductory-level programming class cover, from declaring variables to recursion. However, there is no clear boundary between these fields, making it difficult to evaluate the students' performance in different fields. The best we can do is to set milestones and goals, and link the questions to them. The quizzes designed for the purpose of this experiment has been well-documented and used for later semesters.

From the experiment results, increasing points required to achieve by students in quizzes has no effect on their exam scores. There is no point in setting the minimum point required for students to do at a higher number. In fact, if I were to do the experiment again, I will try see the difference between setting the point required to 1 and 2 instead of 2 and 3. In future quizzes for CS 101, I would highly recommend to lower the minimum point required.

The findings do not show a difference between two groups. The students from both groups do not show a significant difference in their exam scores. However, the results strongly suggests that students who get higher scores in the exam spend fewer tries but similar amount of time on quizzes to get the correct answer. The study concludes that the effort from each student in online quizzes does not show a correlation with their exam performance.

Similar to previous findings[2, 3, 23], time is not an important factor for individuals' performance. To be more specific, the students that spent more time on quizzes do not necessarily perform better or worse in the quizzes or exams. It is highly volatile and person-to-person. Everyone has their own pace of studying, and different speed of learning knowledge.

The most important finding from the experiments is that that the group with higher stakes quizzes performance significantly better in those open quizzes. Although each quiz is only worth 0.5% towards the total grade in CS 101, students still spend a good amount of effort on them. This strongly supports to make quizzes higher stakes in future semesters so that students will be dealing with quizzes more cautiously and hopefully learn more from them. It is not enough to argue that those having higher stake quizzes have a deeper understanding of the material, but it is certain that they spend more effort to make sure their answers are correct.

CHAPTER 7: FUTURE WORK

7.1 DATA COLLECTING

It might be effective to tune the parameters of the experiment more drastically. As mentioned in section 3.2.1, group A has a minimum requirement of 2 points and group B has 3 points. The point difference in the experiment might be too small. It will be helpful to see whether the result to change if we have 2 points and 4 points instead.

It might also be effective to have the students in future semesters take part in the same experiment. However, if the experiment isn't changed, there will be little chance that a different result is retrieved.

7.2 DISCOURAGE GUESSING

In tests, instructors commonly penalize wrong answers to discourage random guessing [24, 25]. However, it is only feasible in the summative assignments. In the PrairieLearn homework format, the students will not receive any kind of penalty. In computerized assignments, since the students can get the feedback instantly, previous studies have identified a new behavior named “rapid guessing”. “Rapid guessing” refers to the situation that some students keep submitting answers until they hit the right one.

Does randomization of question discourage guessing? In an experiment set up like section 3.2.2 where one student needs to get the right answer multiple choices and the values are constantly changing. From a student's perspective, it might be easier to figure out how to solve one question than random guessing. Or maybe some students still think it will be easier just to keep guessing quickly until they get the right answer.

Wise[26] studied the students' behavior in a computerized test, and filtered out the submissions that took only a few seconds. It might be helpful to eliminate random guessing by setting a time limit between submissions.

In a current setup like PrairieLearn, does the randomization of the questions discourage guessing? We might be able to know it if we split two students into two groups. Group A has no random variants of questions while group B has all questions randomized and choices scrambled in order. No correct answer will be given for any question. No penalty will be given for incorrect submissions. Every time the students in group A make a submission, they will be notified if the answer is correct or incorrect. Then they will see the exact same question again to make a second attempt. The students in group B, however, when they

make an incorrect submission, they will see a slightly different question with values changed. We might be able to observe a difference between number of attempts between two student groups.

7.3 PRAIRIELEARN

PL is still an ongoing project with much to work on. A script for translating RELATE YAML files to PL questions is developed for the experiment. PL can use a better question management system, a student management view, and more types of questions supported. To learn more about PL and contribute, visit <https://github.com/PrairieLearn>.

APPENDIX A: QUIZ VIEW BY STUDENTS

The appendix includes examples of quizzes that students interact with. It also includes questions of different types both in Prairie Learn and Relate mentioned in section 3.2 and 3.3.

Quiz04: Quiz 04 - A (UIN ends in 0 ~ 4)			
Total points: 0/13		0%	Available credit: 100% ?
Answer questions in any order. Questions can be repeated and not all questions must be completed. Answering a question correctly will increase its score by its current value, and will also increase its value. Answering a question incorrectly will reset its value. Total points is the sum of all question points.			
First quiz of CS 101 on Prairie Learn!			
Question	Value	History	Awarded points
Quiz problems			
Quiz04.1. pedantic	1		0/1
Quiz04.2. print	1		0/1
Quiz04.3. Evaluate Expression	1		0/2
Quiz04.4. Evaluate Expression	1		0/2
Quiz04.5. Evaluate Expression	1		0/2
Quiz04.6. Evaluate Expression	1		0/2
Quiz04.7. Evaluate Expression	1		0/2
Quiz04.8. Distinguish parts of code	1		0/1

Figure A.1: An overview of the content before starting each quiz in PL.

Quiz05.4. Basic Conditional Logic

What should go in the blank to cause it to print "Print this" once?

```
x = 3
if x > 2:
    x = 1
_____
print("Print this")
```

☐ (a) elif x < 2:

☐ (b) if x > 0:

☐ (c) if x == 0:

☐ (d) else:

Grade

Save

Quiz 05

Total points: 0/13

Score: 0%

Question

Value: 1

History:

Awarded points: 0/2

Report an issue with this question

Previous question

Next question

Figure A.2: An example of the questions provided to students. It is randomized (section 3.2.2) and adaptive scoring (section 3.2.1).

Quiz15: Quiz 15 - A (UIN ends in 0 ~ 4)			
Total points: 0/100		0%	Assessment is open and you can answer questions
This is not proctored. Do your best and your final score will have 10% extra credit. (If you get 90% overall, you will be curved to 100%).			
Question	Best submission ?	Available points ?	Awarded points ?
Quiz problems			
Question 1	unanswered	10, 9, 8, 7, 6, 5, 4, 3, 2, 1 ?	0 /10
Question 2	unanswered	10, 9, 8, 7, 6, 5, 4, 3, 2, 1 ?	0 /10
Question 3	unanswered	10, 9, 8, 7, 6, 5, 4, 3, 2, 1 ?	0 /10
Question 4	unanswered	10, 9, 8, 7, 6, 5, 4, 3, 2, 1 ?	0 /10
Question 5	unanswered	10, 9, 8, 7, 6, 5, 4, 3, 2, 1 ?	0 /10
Question 6	unanswered	10, 9, 8, 7, 6, 5, 4, 3, 2, 1 ?	0 /10
Question 7	unanswered	10, 9, 8, 7, 6, 5, 4, 3, 2, 1 ?	0 /10
Question 8	unanswered	10, 9, 8, 7, 6, 5, 4, 3, 2, 1 ?	0 /10
Question 9	unanswered	10, 9, 8, 7, 6, 5, 4, 3, 2, 1 ?	0 /10
Question 10	unanswered	10, 9, 8, 7, 6, 5, 4, 3, 2, 1 ?	0 /10
Submit your answer to each question with the Grade button on the question page. Look at Best submission to confirm that each question has been graded. When you are done, please logout and close your browser; there is no need to do anything else.			

Figure A.3: An overview of the exam-format quiz introduced in the second stage of experiment (section 4.3.1). Students can directly see available points and the declining available points.

Quiz15: Quiz 15 - B (UIN ends in 5 ~ 9)			
Total points: 0/100		0%	Assessment is open and you can answer questions
This is not proctored. Do your best and your final score will have 15% extra credit. (If you get 90% overall, you will be curved to 100%).			
Question	Best submission ?	Available points ?	Awarded points ?
Quiz problems			
Question 1	unanswered	10, 8, 6, 4, 2, 1 ?	0 /10
Question 2	unanswered	10, 8, 6, 4, 2, 1 ?	0 /10
Question 3	unanswered	10, 8, 6, 4, 2, 1 ?	0 /10
Question 4	unanswered	10, 8, 6, 4, 2, 1 ?	0 /10
Question 5	unanswered	10, 8, 6, 4, 2, 1 ?	0 /10
Question 6	unanswered	10, 8, 6, 4, 2, 1 ?	0 /10
Question 7	unanswered	10, 8, 6, 4, 2, 1 ?	0 /10
Question 8	unanswered	10, 8, 6, 4, 2, 1 ?	0 /10
Question 9	unanswered	10, 8, 6, 4, 2, 1 ?	0 /10
Question 10	unanswered	10, 8, 6, 4, 2, 1 ?	0 /10
Submit your answer to each question with the Grade button on the question page. Look at Best submission to confirm that each question has been graded. When you are done, please logout and close your browser; there is no need to do anything else.			

Figure A.4: An overview of the exam-format quiz of group B. The declining is steeper, but more extra credit is offered.

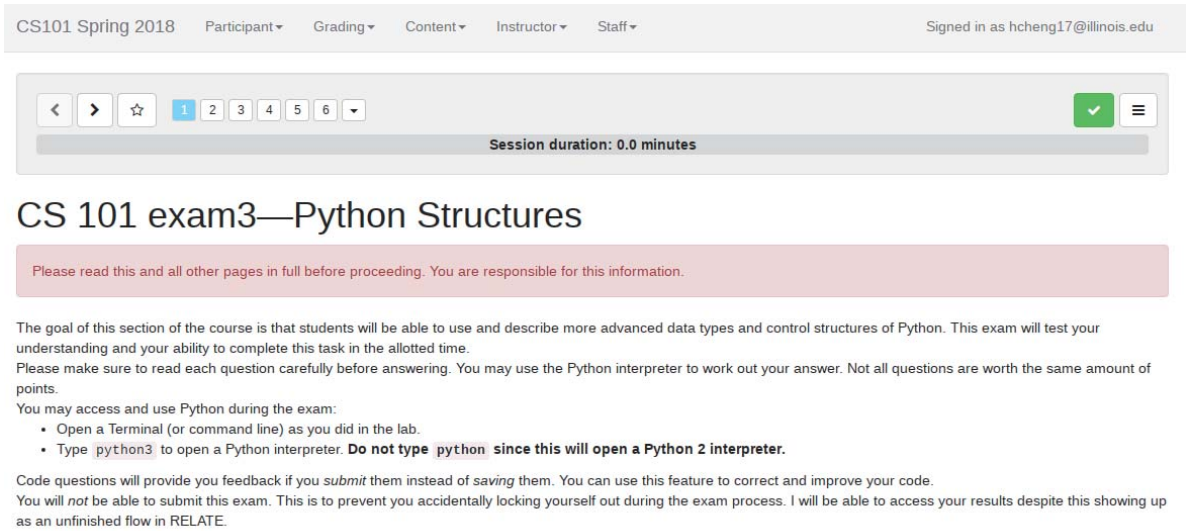


Figure A.5: An example of exam starting page in Relate.

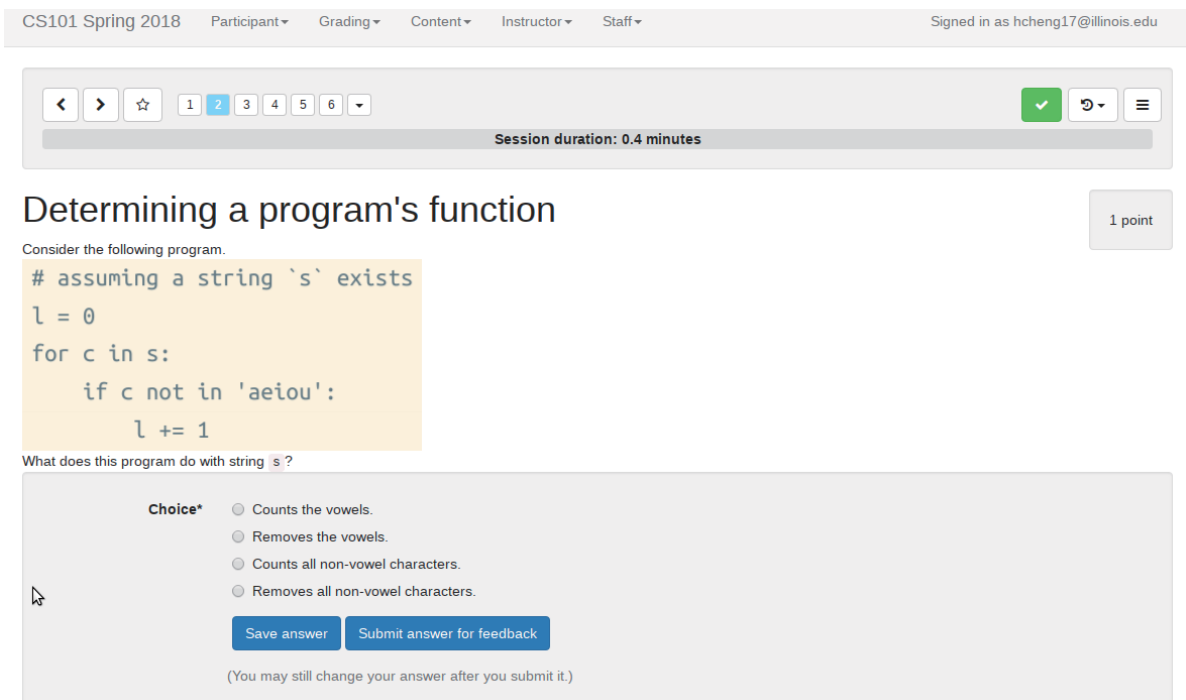
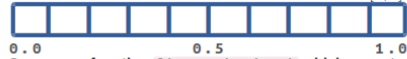


Figure A.6: An example of multiple choice questions in Relate exams. Students won't be able to see if they have the right answer after submission.

Generate equally spaced coordinates

3 points

We wish to generate n equally-spaced x coordinates in $[a, b]$. In this figure, these correspond to the vertical dividing lines between segments.



Compose a function `linspace(a, b, n)` which accepts a left-hand bound `a`, a right-hand bound `b` (both `float`s), and a number of intervals `n` (an `int`), and returns a `list` containing the coordinates dividing the space equally. `n` should be optional with a default value of `11`. You may assume that $b > a$.

For instance, `linspace(0, 1)` should return the list:

```
[ 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0 ]
```

while `linspace(0, 10, n=10)` should return the list:

```
[ 0.0, 1.11111111, 2.22222222, 3.33333333, 4.44444444, 5.55555556, 6.66666667, 7.77777778, 8.88888889, 10.0 ]
```

Note that both the left- and right-hand bounds are *included*. You may use a `for` loop or a `while` loop to implement the solution to this question. You may also assume that $a < b$, so don't worry about testing or swapping those values. Finally, please note that small numerical errors are permissible, e.g., `1.00000004` or similar.

A formula for the i th element is

$$x_i = a + i \frac{(b-a)}{n-1}.$$

Importing `linspace` from a library will not work. Work with lists here, not NumPy arrays.

Answer*

1

Press F9 to toggle full-screen mode. Set editor mode in [user profile](#).

Save answer

Submit answer for feedback

(You may still change your answer after you submit it.)

Figure A.7: An example of more sophisticated coding questions in Relate exams. Students are able to submit multiple times and see their scores directly for each try. Only the last try is counted.

APPENDIX B: ANALYSIS IN R

B.1 DATA CLEANING IN EXPERIMENTS

Raw data are gathered from the backend of PL and Relate. Question data related to quiz instances are imported in csv files from PL. Exam scores are downloaded from Relate in the backend.

As shown in Listing B.1, students are divided into two groups by their last digit in their university identification number (UIN). Each group is assigned a different version of quizzes. However, there was no constraint from them selecting the other version of quizzes. There were a handful of reports from students that they “accidentally did the wrong version”. In that case, their submission was excluded in the study.

Listing B.1: Grouping students by the last digit of UIN

```
lastCharacterLt5 <- function(x, n = 1){  
  as.numeric(substr(x, nchar(x)-n+1, nchar(x))) < 5  
}  
  
dice = function(student_info) {  
  data.frame(NetId = student_info$Net.ID,  
             UID = student_info$Email.Address,  
             UIN = student_info$UIN)  
}  
  
student_info_A = dice(student_info[lastCharacterLt5(student_info$UIN),])  
student_info_B = dice(student_info[!lastCharacterLt5(student_info$UIN),])
```

Since we are taking average of tries, it is safe against some students missing quizzes. We still take the average number. In some rare scenarios, students start the quiz but decide to do it later after browsing through it, and they forget to do the quiz. This will severely impact the average number of attempts since the average is 16 over 12 samples, and it will decrease almost 10% of the average number of attempts. If that is the case, a minimum attempt for at least half of the questions is required. The Listing B.2 shows an example of the implementation.

Listing B.2: Set minimum number of attempts to be considered in averaging

```
library(dplyr)
```



```

avgIfValid = function(tries) {
  mean(tries[tries > 5])
}

getValues = function(sumOfAttempts) {
  data.frame(Quiz4 = sumOfAttempts$Quiz4,
             Quiz5 = sumOfAttempts$Quiz5,
             ...,
             Quiz16 = sumOfAttempts$Quiz16)
}

sumOfAttemptsValuesA = getValues(sumOfAttemptsA)
sumOfAttemptsValuesB = getValues(sumOfAttemptsB)
sumOfAttemptsA = sumOfAttemptsA
  %>% mutate(avg_tries =apply(sumOfAttemptsValuesA,1,avgIfValid))
sumOfAttemptsB = sumOfAttemptsB
  %>% mutate(avg_tries =apply(sumOfAttemptsValuesB,1,avgIfValid))

```

B.2 LINE FITTING IN PLOTTING

A simple ggplot in Listing B.3 is used in line fitting and plotting.

Listing B.3: Fitting and plotting average number of attempts in quizzes vs exam scores

```

library(ggplot2)
ggplot() +
  geom_point(aes(avg_tries, exam3, color = color), final_A) +
  geom_smooth(aes(avg_tries, exam3, color = color), final_A) +
  geom_smooth(aes(avg_tries, exam3, color = color), final_B) +
  geom_point(aes(avg_tries, exam3, color = color), final_B) +
  labs(x = "Average number of incorrect attempts")+
  scale_colour_grey()

```

Individual plotting in Listing B.4 is also implemented with ggplot.

Listing B.4: Plotting average number of attempts in quizzes vs exam scores within each group

```

ggplot(final_A) +

```

```

    geom_point(aes(avg_tries, exam3)) +
    labs(x = "Quiz_average_score")

ggplot(final_B) +
  geom_point(aes(avg_tries, exam3)) +
  labs(x = "Quiz_average_score")

```

B.3 LINEAR MODELING OVER OTHER MODELS

No other modeling would make sense other than linear modeling when we simply need to find if a correlation exists. Listing B.5 shows how it is done in R with *caret* library.

Listing B.5: *caret* training number of attempts in quizzes vs exam scores within each group with *lm*

```

library(caret)
final_A_values = cbind(final_A)
final_A_values$UID = NULL
final_A_values$avg_tries = NULL
final_A_values$NetId = NULL
final_A_values$UIN = NULL
final_A_values$Last.Name = NULL
final_A_values$First.Name = NULL
final_A_values$exam3_raw = NULL
final_A_values$score = NULL
final_A_values$color = NULL
final_A_values$Quiz13 = final_A_values$Quiz14 = final_A_values$Quiz15 = final_A_values$

cv_5 = trainControl(method = "cv", number = 5)
sim_glm_cv = train(
  exam3 ~ .,
  data = final_A_values,
  trControl = cv_5,
  method = "lm")
summary(sim_glm_cv)

```

B.4 NORMALIZATION OF QUIZ SCORES

Section 4.3.3 mentions the difference between two grading methods in group A and group B. The following script B.6 shows how originally it was used at the beginning of the third stage. Since the students in group B lose two points for every incorrect submission when the students in group A only lose one point, I have to divide the points the students in group B lose by two to get their number of incorrect submissions. I subtract that from the total available points for the normalized point. Listing B.7 shows the code implementation. Later, it occurs to me that it is more straightforward to calculate the number of incorrect submissions directly. Therefore, I subtract the normalized points from all total available points again in Listing B.8.

Listing B.6: Original implementation of taking average scores

```
# Calculate average score
avgIfValid = function(scores) {
  mean(scores[scores > 5])
}
```

Listing B.7: Normalized way of taking average scores

```
# Normalized score
avgIfValid = function(scores) {
  mean(scores[scores > 5])
}

avgIfValidB = function(scores) {
  100 - (100 - mean(scores[scores > 5])) / 2
}
```

Listing B.8: Directly calculating number of incorrect attempts

```
# Number of mistakes made
avgIfValid = function(scores) {
  100 - mean(scores[scores > 5])
}

avgIfValidB = function(scores) {
  (100 - mean(scores[scores > 5])) / 2
}
```

REFERENCES

- [1] A. Gromada and C. Shewbridge, “Student learning time,” no. 127, 2016. [Online]. Available: <https://www.oecd-ilibrary.org/content/paper/5jm409kqqkjh-en>
- [2] M. A. Cattaneo, C. Oggenfuss, and S. C. Wolter, “The more, the better? the impact of instructional time on student performance.” *Education Economics*, vol. 25, no. 5, pp. 433 – 445, 2017. [Online]. Available: <http://search.ebscohost.com/login.aspx?direct=true&db=asn&AN=123929642>
- [3] B. Trout, “The effect of class session length on student performance, homework, and instructor evaluations in an introductory accounting course.” *Journal of Education for Business*, vol. 93, no. 1, pp. 16 – 22, 2018. [Online]. Available: <http://search.ebscohost.com/login.aspx?direct=true&db=asn&AN=127481076>
- [4] P. Barba, G. Kennedy, and M. Ainley, “The role of students & motivation and participation in predicting performance in a mooc,” *Journal of Computer Assisted Learning*, vol. 32, no. 3, pp. 218–231, 8 2017. [Online]. Available: <http://doi.org/10.1111/jcal.12130>
- [5] B. Giesbers, B. Rienties, D. Tempelaar, and W. Gijsselaers, “Investigating the relations between motivation, tool use, participation, and performance in an e-learning course using web-videoconferencing.” *Computers in Human Behavior*, vol. 29, no. 1, pp. 285 – 292, 2013. [Online]. Available: <http://search.ebscohost.com.proxy2.library.illinois.edu/login.aspx?direct=true&db=asn&AN=83162059>
- [6] D. Weltman, *A comparison of traditional and active learning methods: an empirical investigation utilizing a linear mixed model*.
- [7] C. Kyriacou, “Active learning in secondary school mathematics,” *British Educational Research Journal*, vol. 18, no. 3, pp. 309–318, 1992. [Online]. Available: <http://www.jstor.org/stable/1500835>
- [8] C. C. Bonwell and J. A. Eison, *Active learning: creating excitement in the classroom*. Jossey-Bass, 2005.
- [9] A. Renkl, R. K. Atkinson, U. H. Maier, and R. Staley, “From example study to problem solving: Smooth transitions help learning.” *Journal of Experimental Education*, vol. 70, no. 4, p. 293, 2002. [Online]. Available: <http://search.ebscohost.com.proxy2.library.illinois.edu/login.aspx?direct=true&db=asn&AN=6909242>
- [10] L. Sixing, L. Lazos, and R. Lysecky, “Feal: Fine-grained evaluation of active learning in collaborative learning spaces.” *Proceedings of the ASEE Annual Conference & Exposition*, pp. 12 101 – 12 116, 2017. [Online]. Available: <http://search.ebscohost.com.proxy2.library.illinois.edu/login.aspx?direct=true&db=asn&AN=125730541>

- [11] R. Al-Hammoud, A. Khan, O. Egbue, and S. Phillips, “An innovative teaching method to increase engagement in the classroom: A case study in science and engineering.” *Proceedings of the ASEE Annual Conference & Exposition*, pp. 1539 – 1553, 2017. [Online]. Available: <http://search.ebscohost.com.proxy2.library.illinois.edu/login.aspx?direct=true&db=asn&AN=125729721>
- [12] L. A. Vinney, L. Howles, G. Levenson, and N. P. Connorb, “Augmenting college students’ study of speech-language pathology using computer-based mini quiz games.” *American Journal of Speech-Language Pathology*, vol. 25, no. 3, pp. 416 – 425, 2016. [Online]. Available: <http://search.ebscohost.com.proxy2.library.illinois.edu/login.aspx?direct=true&db=asn&AN=117675154>
- [13] T.-H. Wang, “Web-based quiz-game-like formative assessment: Development and evaluation.” *Computers & Education*, vol. 51, no. 3, pp. 1247 – 1263, 2008. [Online]. Available: <http://search.ebscohost.com.proxy2.library.illinois.edu/login.aspx?direct=true&db=asn&AN=32731937>
- [14] K. Anthis and L. Adams, “Scaffolding: Relationships among online quiz parameters and classroom exam scores.” *Teaching of Psychology*, vol. 39, no. 4, pp. 284 – 287, 2012. [Online]. Available: <http://search.ebscohost.com.proxy2.library.illinois.edu/login.aspx?direct=true&db=asn&AN=86729491>
- [15] O. Blter, E. Enstrm, and B. Klingenberg, “The effect of short formative diagnostic web quizzes with minimal feedback.” *Computers & Education*, vol. 60, no. 1, pp. 234 – 242, 2013. [Online]. Available: <http://search.ebscohost.com.proxy2.library.illinois.edu/login.aspx?direct=true&db=asn&AN=83160809>
- [16] L. Williams, “Integrating pair programming into a software development process,” *Proceedings 14th Conference on Software Engineering Education and Training. In search of a software engineering profession (Cat. No.PR01059)*.
- [17] D. J. Dimas, F. Jabbari, and J. Billimek, “Using recorded lectures and low stakes online quizzes to improve learning efficiency in undergraduate engineering courses.” *Proceedings of the ASEE Annual Conference & Exposition*, pp. 1 – 15, 2014. [Online]. Available: <http://search.ebscohost.com.proxy2.library.illinois.edu/login.aspx?direct=true&db=asn&AN=115956122>
- [18] M. West, G. Herman, and C. Zilles, “Prairielearn: Mastery-based online problem solving with adaptive scoring and recommendations driven by machine learning,” *2015 ASEE Annual Conference and Exposition Proceedings*.
- [19] C. Zilles, R. Deloatch, J. Bailey, B. Khattar, W. Fagen, C. Heeren, D. Mussulman, and M. West, “Computerized testing: A vision and initial experiences,” *2015 ASEE Annual Conference and Exposition Proceedings*.
- [20] C. Zilles, M. West, and D. Mussulman, “Student behavior in selecting an exam time in a computer-based testing facility,” *2016 ASEE Annual Conference & Exposition Proceedings*.

- [21] B. Chen, M. West, and C. Zilles, “Do performance trends suggest wide-spread collaborative cheating on asynchronous exams?” *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale - L@S 17*, 2017.
- [22] J. Bailey, C. Zilles, and M. West, “Measuring revealed student scheduling preferences using constrained discrete choice models: American society for engineering education,” *Measuring Revealed Student Scheduling Preferences using Constrained Discrete Choice Models: American Society for Engineering Education*. [Online]. Available: <https://www.asee.org/public/conferences/78/papers/19940/view>
- [23] M. Romero and E. Barber, “Quality of e-learners time and learning performance beyond quantitative time-on-task,” *The International Review of Research in Open and Distributed Learning*, vol. 12, no. 5, p. 125, 2011.
- [24] M. L. Campbell, “Multiple-choice exams and guessing: Results from a one-year study of general chemistry tests designed to discourage guessing.” *Journal of Chemical Education*, vol. 92, no. 7, pp. 1194 – 1200, 2015. [Online]. Available: <http://search.ebscohost.com.proxy2.library.illinois.edu/login.aspx?direct=true&db=asn&AN=108643720>
- [25] M. P. Espinosa and J. Gardeazabal, “Optimal correction for guessing in multiple-choice tests.” *Journal of Mathematical Psychology*, vol. 54, no. 5, pp. 415 – 425, 2010. [Online]. Available: <http://search.ebscohost.com.proxy2.library.illinois.edu/login.aspx?direct=true&db=asn&AN=53381415>
- [26] S. L. Wise, “An investigation of the differential effort received by items on a low-stakes computer-based test.” *Applied Measurement in Education*, vol. 19, no. 2, pp. 95 – 114, 2006. [Online]. Available: <http://search.ebscohost.com.proxy2.library.illinois.edu/login.aspx?direct=true&db=asn&AN=20549696>