USING UNIFIED MODELING LANGUAGE ACTIVITY DIAGRAMS TO IMPROVE PERFORMANCE IN SOLVING PROBLEMS OF RATIOS AND PROPORTIONS

A Thesis

by

AMANDA MICHELLE SALINAS

BS, Texas A&M University-Corpus Christi, 2013

Submitted in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in

MATHEMATICS

Texas A&M University-Corpus Christi
Corpus Christi, Texas

May 2022

USING UNIFIED MODELING LANGUAGE ACTIVITY DIAGRAMS TO IMPROVE PERFORMANCE IN SOLVING PROBLEMS OF RATIOS AND PROPORTIONS

A Thesis

by

AMANDA MICHELLE SALINAS

This thesis meets the standards for scope and quality of
Texas A&M University-Corpus Christi and is hereby approved.

George Tintera, PhD
Chair

Ping-Jung Tintera, PhD
Co-Chair

Valentina Postelnicu, PhD
Committee Member

May 2022

ABSTRACT

This thesis reports on a study aimed to determine if high school students can improve their performance on solving ratio and proportion problems with one variable by applying concepts of unified modeling language activity diagrams. This activity was chosen to allow for the development of a possible means of improving mathematical performance without having to rely on the use of technology nor requiring students to participate in a computer science class, which are not always offered in public schools. A quasi-experimental control group pre-test-post-test design was used for this study. Each test consisted of 5 ratio or proportion word problems and 5 ratio or proportion equations with a single variable. The 24-student treatment group, which consisted of freshmen and sophomore engineering students, received instruction on unified modeling language and created several activity diagrams, including an activity diagram on how to solve a ratio or proportion problem with one variable. The duration of the treatment period was two weeks, totaling 500 minutes of in-class instruction. The 41-student control group, which consisted of freshmen and sophomore math students, received their regular mathematics curriculum as outlined by the Texas Education Agency. Data analysis showed that the treatment group had higher proficiency in the pre and post-test and there was statistical evidence to show that unified modeling language activity diagrams contributed to the improvement of mathematical performance on solving ratio and proportion problems.

DEDICATION

This thesis would not have been completed without the constant support and motivation from my husband, daughters, and parents. When I was faced with many roadblocks to my research and felt like giving up, they were there to push me and give me time required to work towards my Master of Science degree. Layla and Lilly, you can achieve anything you set your mind to! I truly hope I set a good example of what hard work and perseverance looks like. All my past, current, and future achievements are for you.

ACKNOWLEDGEMENTS

I must acknowledge my thesis committee members for your contributions to my research. Thank you Dr. George Tintera for your boundless patience with me as my research took much longer than anticipated. You were always there to guide me and ensure my tasks were completed despite my crazy schedule. To Dr. Valentina Postelnicu, thank you for sharing your wealth of knowledge with me and taking the time to serve on my committee. Your belief in my writing kept me going when I felt like giving up. Dr. Ping-Jung Tintera, thank you for your encouraging words and motivation. You provided the confidence I needed to move forward with my research after my proposal was completed.

I would also like to thank Amanda Cloud, my co-worker, the other half of my brain, my friend. I would have never attempted this program if you didn't ask. The experience was much more enjoyable having you there. There's no way I can repay you for your assistance and willingness to listen to me, but I shall try.

Lastly, I'd like to acknowledge those who volunteered to provide me peace and quiet when it was needed: my parents Rosie and Ernest, my sister Christina, my in-laws Clem, Javier, Elida and Stephanie, and the best neighbors ever, Vanessa and Joaquin. I know what you had to deal with, and you are greatly appreciated and loved.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

CHAPTER I: INTRODUCTION

Rationale

Solving mathematical equations with one variable is a skill that challenges many students after entering their freshman year in high school, particularly regarding ratios and proportions. Teachers have relied heavily on teaching algorithms and procedures using calculators to help students pass high stakes exams. These methods have apparently yielded short-term results given that students are able to pass their end of course exams, however, in the long-term, these same students struggle solving ratio and proportion problems outside of the mathematics classroom. Proportional reasoning is a skill that is needed for success in higher level math courses and must be developed beginning at the elementary level (Andini and Jupri, 2017).

If students are to be successful in a society that is pushing for participation in programs related to science, technology, engineering, and math, collectively called STEM, students should demonstrate mathematical ability. In efforts to boost students' mathematical ability, educational leaders and teachers have called for increasing participation in computer science programs. Research has shown that computer programming activities can enhance students' logical and critical thinking skills (Cheng, 2016). Participating in computer science courses and related activities causes students to engage in computational thinking; a process that can be beneficial to learning challenging math and science concepts (Garcia-Peñalvo and Mendes, 2018). Wing (2006) defined computational thinking (CT) as "solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science." Skills such as problem decomposition, abstraction, and developing step-by-step solutions to problems are part of CT.

Despite the recognition of the importance of CT and computer science education, only 45% of high schools in the United States offer computer science courses (Code.org, 2019). Also, according to Code.org's 2019 State of Computer Science Education, students who attend rural schools, and schools whose population is classified as economically disadvantaged, are less likely to be provided opportunities to engage in computer science activities. To truly bring computer science to all students and afford them the benefits of a computer science education, more research needs to be conducted on implementing strategies that result in computational thinking without utilizing technology. Such efforts should then be examined to determine if the strategies used can improve mathematical performance.

Purpose

The purpose of this study is to determine if high school students can use the conventions of unified modeling language (UML) activity diagrams to improve their mathematical performance in solving equations with one variable, particularly ratios and proportions. UML activity diagrams are used in software engineering to provide a visual model of sequences of activities or actions. The conventions of UML in the classroom do not require the use of technology, thus creating an easily accessible means to possibly improve the mathematical performance of students. The study also aimed to examine the mathematical performance of students who participate in a STEM program of study, in this case high school engineering courses, compared to the mathematical performance of students who do not participate in a STEM program of study.

Research Questions

This study will aim to answer the following research questions:

1. What is the relationship between using UML activity diagrams to sequence the steps for solving mathematical equations with one variable and students' performance on solving problems of ratios and proportions?
2. How does the mathematical proficiency of students in a high school STEM program of study, particularly engineering courses, compare to that of students not participating in a STEM program of study?

Significance of Study

The treatment used in this study proposed to improve mathematics performance in solving ratio and proportion problems with one variable using UML activity diagrams. Understanding and creating UML activity diagrams is a simple method of engaging students in algebraic and computational thinking. Though similar to flow charts, which have been used in classrooms for decades, UML activity diagrams use particular notations and force the creator to think logically and create a step-by-step process for a given activity. This typically requires UML activity diagram creators to be specific and detailed in the process they are trying to represent. This methodology could be implemented by mathematics teachers in their classrooms without having to wait for the purchase of expensive technology or having to book the use of computer labs in school. The study will also examine the relationship of mathematic proficiency of students engaged in STEM education compared to those not participating in a STEM program.

CHAPTER II: LITERATURE REVIEW

Computer Science and Computational Thinking

The need for expertise in computer science and STEM is evident in the over 600,000 unfilled computing jobs in the United States (Code.org, 2019). Economists Fayer, Lacey, and Watson, for the U.S. Bureau of Labor Statistics (2017), projected a 12.5% employment increase in computer occupations as well as a yield of over 1 million jobs through 2024. Although between 50,000 and 65,000 students are completing a degree in computer science every year, there remains a shortage of qualified people to fill the high demand of computing related jobs (Loyalka et al., 2019, Nagar and Atkinson, 2016). Since less than half of the nation's high schools offer computer science courses, students are not exposed to computer science principles and activities that promote awareness and generate interest in pursuing degrees in computer science and related fields.

As technology continually advances and is more accessible to many students, it is necessary to broaden technological knowledge beyond simple word processing and presentation software. Skills that can be acquired in computer science courses "can be applied to many areas in STEM fields, which are necessary for maintaining scientific and innovative stature in the global market" (Montoya, 2017, p. 47). Nager and Atkinson (2016) state the computer skills and competencies are in high demand in a wide range of industries including, but not limited to, the technology sector. In 2016, United States Chief Technology Officer Megan Smith released information on President Obama's "Computer Science For All" initiative; a plan that would call for funding to provide K-12 students opportunities to learn computer science and acquire CT skills needed to be successful in a digital economy.

The push to bring computer science education to all K-12 students has called for examination to the obstacles for successful implementation of computer science programs. Wang, Hong, Ravitz, and Moghadam (2016) examined five different aspects of computer science (CS): perceptions of CS; interest and demand for CS; in school and extracurricular opportunities for CS; participation in CS; and barriers to providing or accessing CS opportunities. The researchers surveyed students, teachers, principals, parents, and superintendents across the United States. They found that perceptions of CS were influenced by knowledge of CS. A common misconception among the respondents was that computer scientists create documents and presentations or searched the internet. Wang et al. (2016) found that males were more likely to correctly identify CS as compared to female students, parents, and teachers. The majority of those surveyed also believed that CS was just as important, if not more so, than core subjects such as English, math, science, and social studies. Gender bias and stereotypes also became evident in the research. Many respondents believed that boys are more interested in CS than girls and most students and parents believed that computer scientists are white males who wear glasses. According to Wang et al. (2016), these beliefs have perpetuated the diversity issue in CS with women and minorities being underrepresented in K-12 CS programs as well as post-secondary institutions.

Another contributor to underrepresentation of minorities in CS and implementation in schools is access to technology. Compared with the reported 98% of White students having access to computers at home, only 75% of Hispanic students and 85% of Black students reported access (Wang et al., 2016). Of the principals surveyed in Wang et al's (2016) study, over 75% reported not offering CS programs in their schools which may be linked to percentages of students who reported using computers in school on a daily basis: 31% of Hispanics, 45% of

Blacks, and 42% of whites. The researchers also found that opportunities to participate in CS through dedicated classes or extracurricular activities decreased as reported household incomes decreased.

To address issues of access to technology and continue with efforts of the Computer Science for All initiative, educators have begun to utilize CS unplugged activities to introduce CS principles to students. CS unplugged activities allow students to examine fundamental computer science concepts without the need for knowledge of programming languages or use of a computer (Rodriguez, 2015). Rodriguez (2015) states that CS unplugged activities are kinesthetic and engaging, and when "successfully combined with educational aspects of computational thinking, then we will have an effective means to empower students to solve problems." According to csunplugged.org, a project by the Computer Science Education Research Group at the University of Canterbury, there are six problem solving skills addressed by CS unplugged activities: algorithmic thinking; abstraction; decomposition; generalizing and patterns; evaluation; and logic. The proposed study will create an unplugged activity that will allow for the development of a step-by-step process for solving mathematical equations, thus working on the algorithmic thinking component of CT, to determine if the exercise will improve students' ability to solve problems of ratios and proportions.

Mathematical Ability Related to Programming Ability

Wing (2008) explained that computational thinking is a type of analytical thinking which shares with mathematical thinking the methods in which a person approaches solving a problem. Mathematical problem solving requires the ability to think logically and work with algorithms applying systematic and specific approaches when solving problems; a process known as algorithmic thinking (Kaufmann and Stenseth, 2020). Attallah, Ilagure, and Chang (2018) say the first step in programming is writing a correct algorithm, thus programming requires students to be strong in algorithmic thinking and problem solving.

Sari, Sukmawati, and Zulkarnain (2018) describe a dynamic mathematical ability which includes reasoning, connection, modeling, and problem solving that can be communicated efficiently in written and verbal forms. Sari et al. (2018) explain that mathematical representation is the bridge that connects ideas and languages in computer programming. Students skilled with mathematical representation exhibit mathematical ability which allows them to be more successful in postsecondary computer education courses. Sari et al. (2018) examined mathematical ability and programming ability by comparing achievement in basic math courses and achievement in programming courses of 137 students in a Computer Science Education program. Through quantitative data analysis they were able to conclude that there is a strong relationship between mathematical ability and programming ability. The Pearson Correlation (r) coefficient was found to be 0.634, a positive direction indicating that as mathematical abilities of students increase, so do their programming abilities.

The study by Psycharis and Kallia (2017) examined how teaching programming in conjunction with mathematics affects students' problem solving and reasoning skills as well as their self-efficacy. The researchers studied 66 students in their senior year of high school in Greece. The quasi-experimental design study used the non-equivalent control group pre-test/post-test design. Both groups received the same mathematics instruction as prescribed by the syllabus, however the experimental group developed a computer program to solve a problem set. The mathematics teachers worked closely with the computer programming teacher to develop lessons for specific parts of the syllabus in the experimental group. Psycharis and Kallia (2017) administered the Cornell Reasoning Test in October as a pre-test and again in April after 90 hours of instruction as a post-test. Wilcoxon signed ranks test and the Mann-Whitney U test showed that the programming intervention significantly enhanced reasoning skills. A mathematics assessment test was given to both groups to study problem solving skills. The students were given a statistics problem in which the control group solved in the traditional way and the experimental group wrote source code to transfer to Matlab to generate a graphical solution. The Wilcoxon signed ranks test and the Mann-Whitney U test showed that the programming intervention had a statistically significant effect on problem solving for the experimental group but there was not a statistically significant difference with the control group. Psycharis and Kallia (2017) used the Motivated Strategies for Learning Questionnaire (MSLQ) to study the effects of programming on student self-efficacy in mathematics. The MSLQ was administered to students in both the control and experimental groups before the start of the course and after 90 hours of instruction. The Wilcoxon signed rank test and the Mann-Whitney U test showed that the programming course significantly enhanced students' self-efficacy in mathematics.

Findings such as these indicate that mathematical ability is important to success in programming and that programming can be used to improve mathematical ability. Further research should be conducted to support these findings and determine if even brief exposure and introduction to basic programming principles, such as UML activity diagrams, can have an impact on mathematical ability.

UML Activity Diagrams

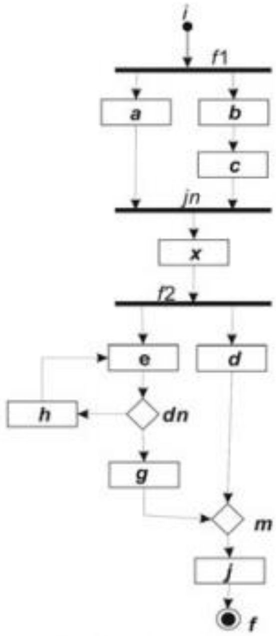
Unified Modeling Language (UML) is a modeling language that uses formal notation to integrate processes to produce, organize, and document artefacts created for the development of software or business processes (Tagliati and Caloro, 2008). Sabitzer, Demarle-Meusel, and

Jarnig (2018) define UML activity diagrams as flowcharts which are used to sequence events from a starting point to a defined end or solution. After examining modeling processes, Sabitzer et al. (2018) determined that the likes of UML activity diagrams could be used to introduce computational thinking as modeling requires abstraction, reduction, problem decomposition, simplification, generalization, and classification. They also determined that modeling is a good tool for story telling which draws parallels to algorithmic thinking, a skill utilized in computational thinking. Sabitzer et al. (2018) aimed to use modeling to determine which techniques were useful and practical for teachers and students in foreign language classes as well as examine the potential for modeling techniques to improve general learning competencies of abstraction, problem solving, and text comprehension. Their research project is still ongoing, but they published data showing that teachers and students found modeling useful and practical despite having difficulties with abstraction and generalization.

According to Chen, Jiang, Hong, and Lin (2018) UML activity diagrams can be used to model sequences and activities of business organizations and even organize the details of how an operation or function works. A UML activity diagram as depicted by Chen et al. (2018) is shown in Figure 1. Activity diagrams indicate the start of a process with a solid circle. Arrows indicate the flow to the next event, whereas rectangles indicate the action and diamonds indicate a question or a decision. Some processes will form loops before moving on to the next step in a process. The end of the sequence, or solution, is indicated by a solid circle within another circle. Chen et al. (2018) provided a graphic (See Figure 2) indicating translations rules; how different parts within the UML activity diagram are interpreted. A simpler activity diagram indicating the activities performed by an ATM once a user inserts their card is shown in Figure 3.

Figure 1

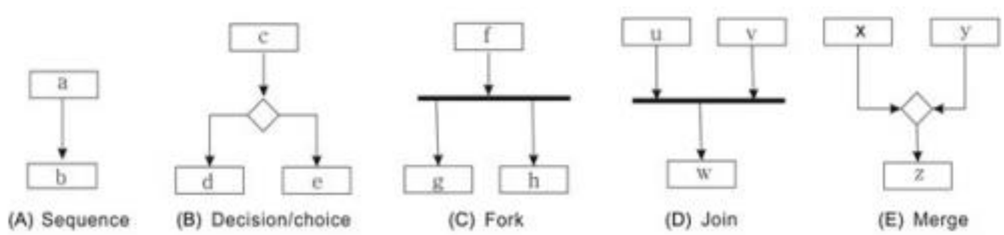
Example Activity Diagram



Activity Diagram depicted in Chen et al. (2018)

Figure 2

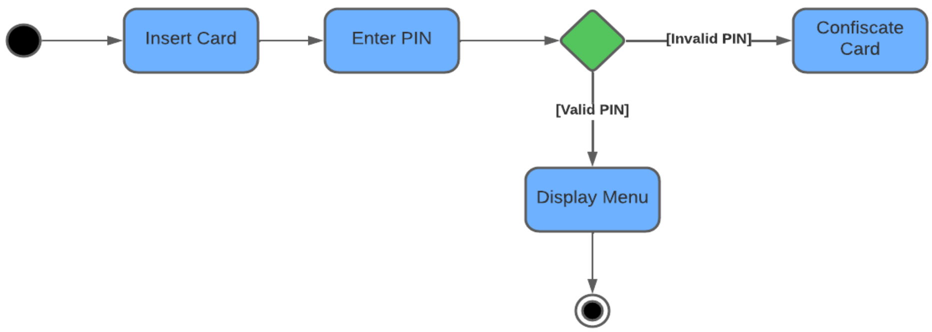
UML Activity Diagram Translation Rules



The basic constituent parts of UML activity diagrams.

Figure 3

A simple activity diagram



This activity diagram depicts the series of events by an ATM once a card is inserted.

There are limited resources currently available addressing the use of UML activity diagrams in high school classrooms to improve solving of mathematical equations, thus necessitating further research. As UML activity diagrams have shown promise for teaching students in foreign language classes and introducing computational thinking skills, research should be conducted to determine if UML activity diagrams can have an impact on mathematical performance.

CHAPTER III: METHODS

Research Design

A quasi-experimental control group pretest-posttest design was used for this study. Students registered in the Principles of Applied Engineering and Engineering Design and Presentation course served as the experimental group and students registered in Pre-Algebra and Algebra 1 served as the control group. All participants were given a pre-test to establish a baseline measure of performance in solving ratio and proportion problems. A post-test was administered after the treatment period to compare the performance of students in each group to their pre-test results. The experimental group had received a two-week treatment in which they were taught the conventions of UML activity diagrams and then were tasked to create activity diagrams to sequence the events for a task of their choosing and for solving a ratio or proportion equation with one variable.

Participants

High school students attending a small Title I public high school in South Texas during the 2021-2022 school year who were registered for the Principles of Applied Engineering or Engineering Design and Presentation course, and students in Pre-Algebra and Algebra 1 were participants in the study. The high school in this study serves approximately 574 students in grades 9-12, of which 88% are classified as economically disadvantaged. The demographic make-up consists of Hispanic (89%), African American (8%), Caucasian (2%), and Native American (1%) students. Eleven percent of the student population is served by Special Education and 4% are English Language Learners. The engineering courses are elective classes, and the mathematics courses are core classes. All students are enrolled in a math class, but not all students are enrolled in an engineering class. Students in these courses were primarily freshmen and sophomores ranging in age from 14 to 16 years old; the sophomore group having been enrolled and completing the Principles of Applied Engineering course the previous school year, 2020-2021. There were 164 students eligible to participate in the study, however only 65 students returned signed consent forms allowing for analysis of their data.

Instruments

Pre-test/Post-test

The pre-test and post-test were developed based on literature review and TEA curriculum standards to ensure the development of a valid assessment instrument as well as to facilitate the use of the data for this study. Each question contained a mathematical equation or a word problem requiring the construction of an equation with one variable related to the concepts of ratio and proportion problems. An example equation is shown below:

$$\text{Solve for } x: \ \frac{5}{16} = \frac{40}{x}$$

An example word problem is shown below:

> Ben's car averages 36 miles per gallon of fuel. How many miles, on average, can Ben's car travel with 14 gallons of fuel? (This uses a unit ratio (miles per gallon). A verbal problem parallel to the equation above might say: Ben's car used 3 gallons of gas when driving 50 miles. How many gallons will he need to drive 400 miles?

See Appendix **A** for the pre-test instrument and Appendix **B** for post-test instrument used for this study. The post-test was not identical to the pre-test to ensure that students simply did not remember correct answers and were able to demonstrate mastery of concepts. Both the pre-test and post-test had five different word problems and five equations with one variable. The questions were marked correct if the value for the variable was successfully calculated. There was not partial credit given for setting up the equation correctly. The pre-test was administered to all math classes during the third week of September 2021. The post-test was administered to all math classes during the second week of March 2022.

UML Activity Diagrams Treatment

The experimental group received a two-week treatment working with UML activity diagrams from January 19, 2022, thru February 2, 2022. The treatment occurred in the engineering classes for 50 minutes a day during the 10 school day window. During this period, the control group continued to receive their normally prescribed curriculum in their mathematics courses. The author of the study provided instruction on the conventions of UML activity diagrams and their application to software development. Students practiced creating simple UML activity diagrams for activities of their choice, or activities that related to engineering projects they had previously worked on in class. The treatment culminated in the students creating an activity diagram that sequenced the actions taken to solve a mathematical equation of ratios and proportions that contained one variable. The students had a reference sheet of symbols they were allowed to use when constructing their UML activity diagrams. Figure 4 depicts the reference provided to the students during the two-week treatment. The students were allowed to use the reference sheet when creating their UML activity diagrams. When students completed their activity diagrams for solving a ratio or proportion problem with one variable, it was found that students only used five symbols: initial node, activity, control flow, branch, and complete activity flow. See Appendix **C** for a typical representation of student created UML activity diagrams for solving a ratio or proportion equation with one variable. Given the nature of the research and an IRB exemption, images of student work are not authorized to display in this thesis.

Figure 4

UML Student Reference Sheet

Basic Notation of the Activity Diagram

Initial Node	A black circle is the standard notation for an initial state before an activity takes place. It can either stand alone or you can use a note to further elucidate the starting point.
Activity	The activity symbols are the basic building blocks of an activity diagram and usually have a short description of the activity they represent.
Control Flow	Arrows represent the direction flow of the flow chart. The arrow points in the direction of progressing activities.
Branch	A marker shaped like a diamond is the standard symbol for a decision. There are always at least two paths coming out of a decision and the condition text lets you know which options are mutually exclusive.
Fork	A fork splits one activity flow into two concurrent activities
Join	A join combines two concurrent activities back into a flow where only one activity is happening at a time.
	The final flow marker shows the ending point for a process in a flow. The difference between a final flow node and the end state node is that the latter represents the end of all flows in an activity.
Complete Activity Flow	The black circle that looks like a selected radio button is the UML symbol for the end state of an activity. As shown in two examples above, notes can also be used to explain an end state.
Notes	The shape used for notes.

Methods of Analysis

The research questions were analyzed using data collected from the instruments administered to the experimental and control groups. Microsoft Excel was used to organize data collected and JMP software was used for statistical analysis. Prior to analysis of data, consent forms were collected and reviewed, and a spreadsheet of student names and ID numbers were sent via email with password encryption to the principal investigator (PI). The mathematics department chair organized all scores from the pre-test and post-test into a single spreadsheet and sent it via email with password encryption to the PI as well. The researching author submitted rosters of the students who were part of the treatment group. The PI then deidentified the data and removed data from students who did not submit a signed consent form. Once the PI completed the deidentified spreadsheet, it was sent to the researching author via email using password encryption so that data analysis could begin.

Research Question 1

To determine if UML activity diagrams can be used to improve students' performance in solving problems of ratios and proportions, the results of the pre-test and post-test were compared with both groups. Table 1 below was used to compare the data from the control and experimental groups. Data was also organized by proficiency percentage, scores that were 70% or greater, for both groups as shown in Table 2. Analysis of Variance (ANOVA)-Post analysis was conducted using JMP software with $\alpha = 0.05$ as the criterion for statistical significance. This method used linear regression to compare the treatment effects with the following model:

$$Y_i^{[p]} = \beta_0^{[p]} + \beta_1^{[p]} X_i + \varepsilon_i^{[p]}.$$

Group assignments were designated by indicator, with $X_i = 1$ for the treatment group and $X_i = 0$ for the control group.

Table 1
Organization of Pre/Post-Test Data

	Pre-test	**Post-test**	**Improvement**
Control Group			
Treatment Group			

Research Question 2

To determine if proficiency of mathematical skills differs between the STEM program engineering students and non-engineering students, the percentage of students scoring 70% or higher on the pre and post-test as well as the data from Table 1 was compared. Table 2 below shows how the proficiency data was organized. ANOVA-Change statistical analysis was conducted for both sets of data using JMP software with $\alpha = 0.05$ as the criterion for statistical significance. This method also used linear regression to compare the treatment effects with the following model:

$$Y_i^{[c]} = \beta_0^{[c1]} + \beta_1^{[c1]} X_i + \varepsilon_i^{[c1]}.$$

Group assignments were designated by indicator, with $X_i = 1$ for the treatment group and $X_i = 0$ for the control group.

Table 2
Organization of Proficiency Data

	Pre-test	**Post-test**
Control Group% > 70%		
Treatment Group% > 70%		

CHAPTER IV: DATA ANALYSIS

Upon receipt of the deidentified spreadsheet containing the pre-test and post-test scores for the control and treatment group, the averages of each were calculated using Microsoft Excel software. There were 65 student scores to analyze, 41 of which were in the control group, and 24 in the treatment group.

Research Question 1

Research question one asked: “What is the relationship between using UML activity diagrams to sequence the steps for solving mathematical equations with one variable and students’ performance on solving problems of ratios and proportions?” Table 3 below shows the results of the initial data analysis.

Table 3
Average scores and improvement for pre/post-test

	Pre-test	**Post-test**	**Improvement**
Control Group	19.76%	68.54%	61%
Treatment Group	51.67%	81.25%	61%

Table 3 shows that the experimental group had higher percentages in both the pre-test and post-test instruments. However, the improvement percentage is the same in both groups. JMP software was used to conduct ANOVA-Post analysis, the results of which are shown in Figure 5.

Figure 5

ANOVA-Post for Pre/Post-Test

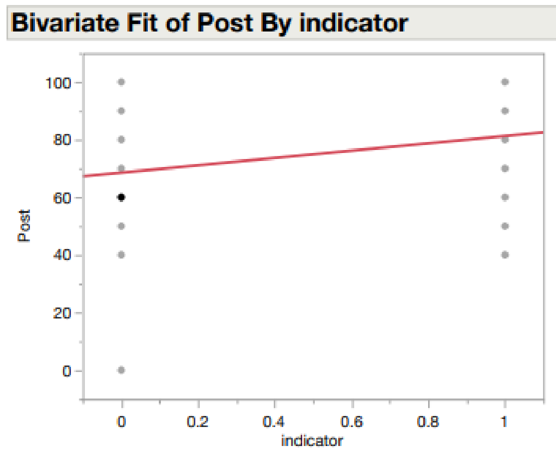
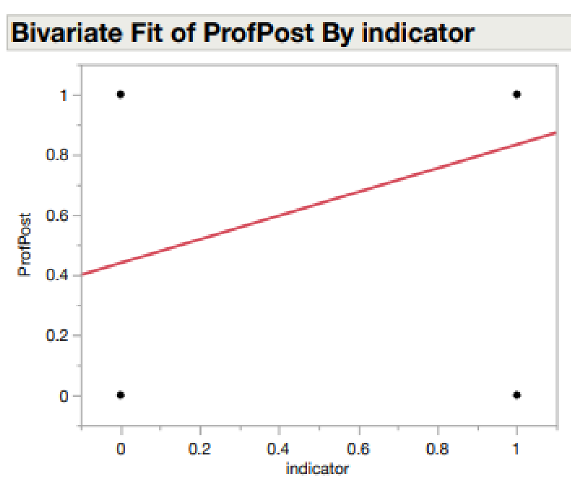


Figure 5 shows the resulting graph of ANOVA-Post analysis. The control group was represented by the indicator 0 and the treatment group by indicator 1. The response variable (Post) is the post-test score The output table from JMP produced $p = 0.013$ with $\alpha = 0.05$ as the criterion for significance. See Appendix **D** for the full ANOVA-Post output from JMP.

The proficiency percentages were also compared using ANOVA-Post with the JMP software. The results are shown in Figure 6 below.

Figure 6

ANOVA-Post for Proficiency



The control group was represented by the indicator 0 and the treatment group by indicator 1. The response variable is the score of 1 for proficiency or 0 for lack of it on the post-test. The output table from JMP produced $p = 0.002$ with $\alpha = 0.05$ as the criterion for significance. See Appendix **E** for the full ANOVA-Post proficiency output from JMP.

Research Question 2

Research question two asked: "How does the mathematical proficiency of students in a high school STEM program of study, particularly engineering courses, compare to that of students not participating in a STEM program of study?" Table 4 below shows the data used to attempt to answer the research question.

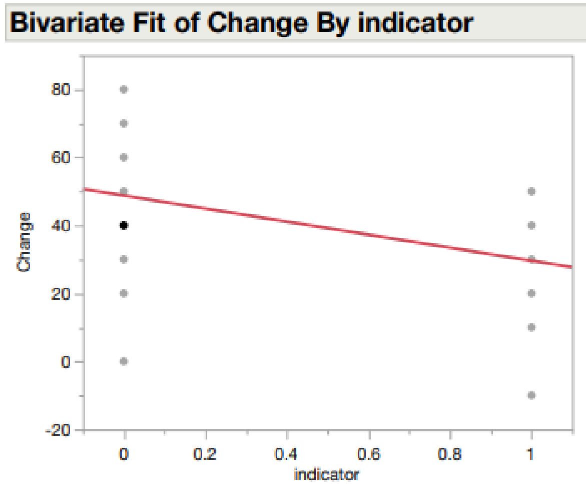
Table 4
Average proficiency scores pre/post-test

	Pre-test	**Post-test**
Control Group% > 70%	2%	44%
Treatment Group% > 70%	25%	83%

Table 4 shows that the treatment group consisting of the STEM program engineering students had higher proficiency percentages in both the pre-test and post-test instruments of the study. Proficiency scores are the percentage of students who scored 70% or higher in the pre and post-test instruments. Figure 6 shows the resulting graph of ANOVA-Change for the data set shown in Table 4.

Figure 7

ANOVA-Change for Pre/Post-Test



Using the data from Table 3, ANOVA-Change statistical analysis in JMP was used to produce Figure 7. The indicators used were 0 for the control group and 1 for the treatment group. The response variable is the change from the pre-test to post-test score. The output table from JMP produced $p < 0.001$ with $\alpha = 0.05$ as the criterion for significance. See Appendix **F** for the full JMP statistical analysis results.

Figure 8

ANOVA-Change for Proficiency

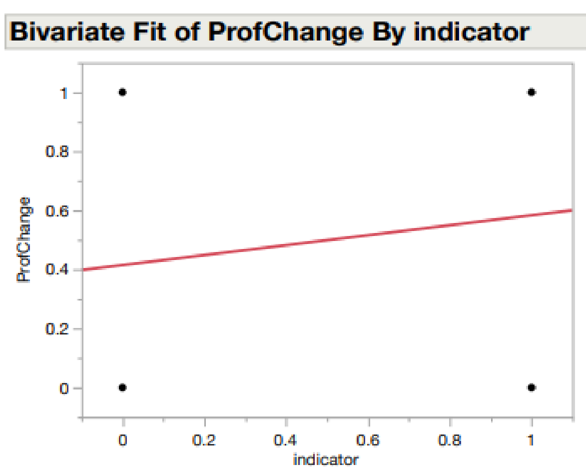


Figure 8 above shows the result of ANOVA-Proficiency Change by indicator using the data set from Table 4. The indicators used were 0 for the control group and 1 for the treatment group. The response variable is the change in proficiency from the pre-test to the post-test. The output table from JMP produced $p = 0.195$ with $\alpha = 0.05$ as the criterion for significance. See Appendix **G** for the full JMP ANOVA-Change proficiency statistical analysis results.

CHAPTER V: DISCUSSION AND CONCLUSIONS

Research Question 1

The results of this study show that the treatment group performed better in the pre and post-test instruments. Though the improvement from pre-test to post-test was the same for both the treatment and control group, the treatment group started with a higher pre-test percentage score, thus they had a smaller margin for improvement. The ANOVA-Post analyses by indicator produced statistically significant values of $p = 0.013$ and $p = 0.002$ suggesting that the UML activity diagram treatment did have a positive effect on the mathematical performance of problems on ratios and proportions. These findings are consistent with Sari et al. (2018) in which they describe a strong relationship with mathematical ability and programming ability. Even though students did not write an actual computer program to solve problems of ratios and proportions, the UML activity diagrams forced them to think in a way that is structured like a computer by describing the step-by-step methodology for solving a ratio or proportion problem with one variable. This computer science related activity did in fact produce an improvement from pre-test to post-test score, which was likely attributed to the students' engagement in computational thinking. As described by Garcia-Peñalvo and Mendes (2018), computational thinking processes are beneficial to learning mathematical concepts.

Research Question 2

The treatment group demonstrated higher proficiency percentages compared to the control group but ANOVA-Change comparing proficiency values produced a statistically insignificant value of $p = 0.195$. The ANOVA-Change comparing all pre-test and post-test data produced a statistically significant value of $p < 0.001$, however the regression line was negative. This is likely due to the fact that the treatment group did start at higher pre-test scores. These findings showed that the STEM students performed better than non-STEM students, but it is not likely attributed to being in the STEM program.

Limitations of Study

This study was not able to produce an accurate picture of all eligible participants due to the lack of signed consent forms returned. The data presented in this study only analyzed 40% of eligible student data. Another factor that may have impacted results was the instructional time that elapsed between the pre-test and post-test. This study also occurred during the COVID-19 pandemic. Though the high school in this study had all students attending, many students were absent or receiving additional help to recover from their learning loss attributed to virtual learning the year prior. Also, due to curriculum constraints within the math courses, there were very few dates allowed to administer each instrument. This could be a factor as to why the control group had similar improvement from pre to post-test when compared with the treatment group. If the treatment could have been administered earlier in the year, it would have been expected that the improvement would have been greater than that of the control group. The reasoning for this hypothesis was that the control group would have had less time and practice with mathematical concepts in their classroom.

Further Research

UML activity diagrams seem promising in helping students improve their mathematical performance in solving ratio and proportion problems, but refined methods are required. A study using randomized assignment that does not allow so much time to elapse between the treatment period and post-test may be best to determine outcomes of this treatment. It may also be valuable to conduct student surveys to get their feedback on the effectiveness of using UML activity diagrams to improve mathematical performance.

If UML activity diagrams do in fact help students to improve performance in solving ratio and proportion problems, it should be experimented with other math concepts and possibly be introduced in primary or middle school. Research could be taken even further with testing for computational and algorithmic thinking skills before and after the UML activity diagram treatment.

Another aspect of this study that could be explored further is the mathematical performance of students before and after completing a STEM program of study in high school. How does the performance of those students compare to students who complete a different program of study in high school?

This study aimed to find a possible means of improving mathematical performance by utilizing a method that is little to no-cost, does not require technology, and does not require much effort for math teachers to learn. Furthermore, the UML activity diagram treatment introduces computer science concepts to students who may not otherwise experience them. This method may pique the interests of students who had no idea about software development if a teacher takes the time to explain how UML activity diagrams are typically used. Nevertheless, students will develop a step-by-step method for solving a problem with UML activity diagrams; a task that will likely help a student to improve their future performance when solving mathematical problems.

REFERENCES

Andini, W., & Jupri, A. (2017). Student obstacles in ratio and proportion learning. *Journal of Physics: Conference Series*, 812, 12048.

Attallah, B., Ilagure, Z., & Chang, Y. (2018). The impact of competencies in mathematics and beyond on learning computer programming in higher education. *2018 Fifth HCT Information Technology Trends (ITT).* pp. 77-81.

Chen, H., Jiang, J. M., Hong, Z., & Lin, L. (2018). Decomposition of UML activity diagrams. *Software: practice and experience*, *48*(1), 105-122.

Cheng, H. "Teaching math with computer programming can help narrow achievement gap," *EdSource*, April 26, 2016, http://edsource.org/2016/teaching-math-with-computer-programming-can-help-narrow-achievement-gap/563371

Code.org (2019). *Why Computer Science?* https://code.org/promote

CS Unplugged. (2019, March 10). *Computational Thinking and CS Unplugged: What is Computational Thinking?* https://csunplugged.org/en/computational-thinking/

Fayer S, Lacey A, Watson A. (2017). *STEM Occupations: Past, Present, and Future.* Bureau of Labor Statistics, Washington, DC. Retrieved from https://www.bls.gov/spotlight/2017/science-technology-engineering-and-mathematics-stem-occupations-past-present-and-future/pdf/science-technology-engineering-and-mathematics-stem-occupations-past-present-and-future.pdf

García-Peñalvo, F. J., & Mendes, A. J. (2018). Exploring the computational thinking effects in pre-university education. *Computers in Human Behavior*, 80, 407-411.

Kaufmann, O.T., & Stenseth, B. (2020). Programming in Mathematics Education. *International Journal of Mathematical Education in Science and Technology.*

pp. 1-20.

Loyalka, P., Liu, O., Li, G., Chirikov, I., Kardanova, E., Gu, L., . . . Tognatta, N. (2019). Computer science skills across China, India, Russia, and the United States. *Proceedings of the National Academy of Sciences of the United States of America*, 116(14), 6732-6736.

Montoya, A. (2017). Computer science for all: Opportunities through a diverse teaching workforce. *Harvard Journal of Hispanic Policy,* 29, 47-62.

Nager, A., & Atkinson, R. (2016). The case for improving U.S. computer science education. *National Consortium of Secondary STEM Schools*, 21 (1), 18-19.

Psycharis, S., & Kallia, M. (2017). The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy and problem solving. *Instructional Science*, *45*(5), 583-602.

Rodriguez, B. R. (2015). *Assessing computational thinking in computer science unplugged activities*. Retrieved from ProQuest Dissertations Publishing. (UMI 1606865)

Sabitzer, B., Demarle-Meusel, H., & Jarnig, M. (2018, April). Computational thinking through modeling in language lessons. In *2018 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1913-1919). IEEE.

Sari, D. P., Sukmawati, R. A., & Zulkarnain, I. (2018, December). The Relationship Between Mathematical Ability and Programming Ability of Computer Science Education Students. In *1st International Conference on Creativity, Innovation and Technology in Education (IC-CITE 2018)*. Atlantis Press.

Smith, M. (2016, January 30). *Computer Science for All.* The National Archives and Records

Administration. https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all

Tagliati, L. V., & Caloro, C. (2008). UML and Object Oriented Drama. *Journal of Object Technology*, *7*(1), 85-101.

Wang, J., Hong, H., Ravitz, J., & Hejazi Moghadam, S. (2016, February). Landscape of K-12 computer science education in the US: Perceptions, access, and barriers. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (pp. 645-650).

Wing, J.M. (2006). Computational Thinking. *Commun ACM*, 49, 33-35.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366, 3717-3725.

APPENDIX A

PRE-TEST INSTRUMENT

1. The ratio of the heights, in cm, of two dogs is 4: 9. What is the height, in cm, of the smaller dog if the larger dog is 36 cm tall?

2. Four out of every six dental patients request fluoride treatment after their dental cleaning treatments. If 120 patients have dental cleanings this week, how many will choose to have fluoride treatments as well?

3. The Lakers have won 27 games and lost 17 games. At this rate, how many games should they win, in an 82 game season?

4. 2 ½ : 3 ½ = n : 32

5. 0.41.5 = 12n

6. 1240 = n25

7. 1 : 2 = n : 9

8. A Toyota Prius drove 94.5 km on 20 L of gasoline. How far should it be able to go on a full, 50 L, tank?

9. If two gallons of paint covers 825 sq. ft., how much paint is needed to cover 2640 sq. ft.?

10. 8 : 19 = 14 : n

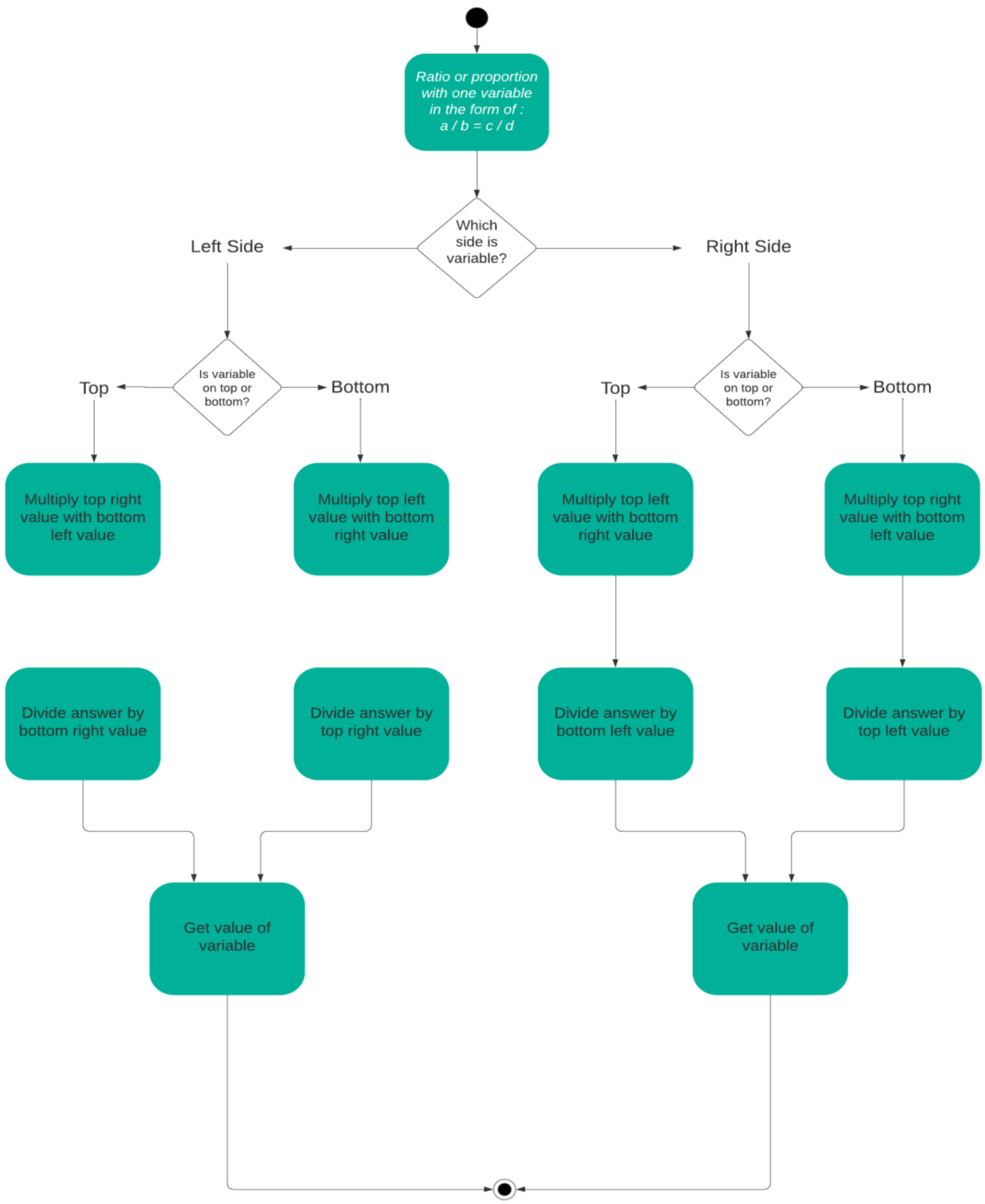
APPENDIX B

POST-TEST INSTRUMENT

1. 24 : 14= n : 13
2. 10n =21.7
3. Louie bought 12 kilograms of shrimp for $252 for a party he is having. He decided to buy 5 more kilograms at the last minute. How much will it cost?
4. Johnny can type a 600 word essay in just 8 minutes. How long will it take to type a 2,100 word essay?
5. It takes the Appealing Fruit company about 16 hours to peel, box, and ship 500 cartons of apples. How many cartons can they box and ship in 2 days?
6. k18= 174
7. In a high school, the ratio of freshmen to seniors is 4:5. If there are 150 seniors, how many freshmen are there?
8. 36x = 87
9. The ratio of an object's weight on Earth to its weight on Neptune is 5:7. How much would a person who weighs 150 pounds on Earth weigh on Neptune?
10. 2.73 =3.6r

APPENDIX C

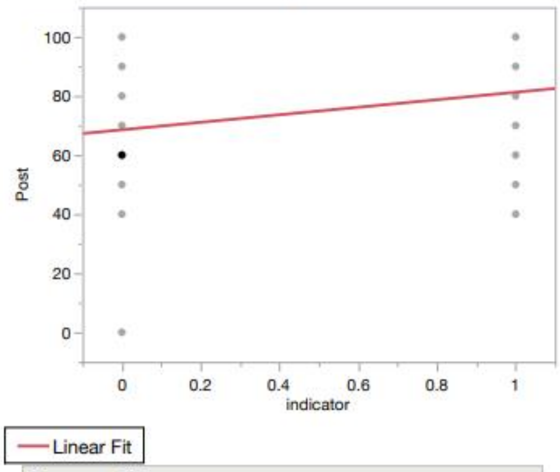
EXAMPLE OF TYPICAL STUDENT CREATED UML ACTIVITY DIAGRAM



APPENDIX D

ANOVA-POST JMP OUTPUT

Bivariate Fit of Post By indicator



Linear Fit

Post = 68.536585 + 12.713415*indicator

Summary of Fit

RSquare	0.094031
RSquare Adj	0.079651
Root Mean Square Error	19.34429
Mean of Response	73.23077
Observations (or Sum Wgts)	65

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	1	2446.843	2446.84	6.5388
Error	63	23574.695	374.20	Prob > F
C. Total	64	26021.538		0.0130*

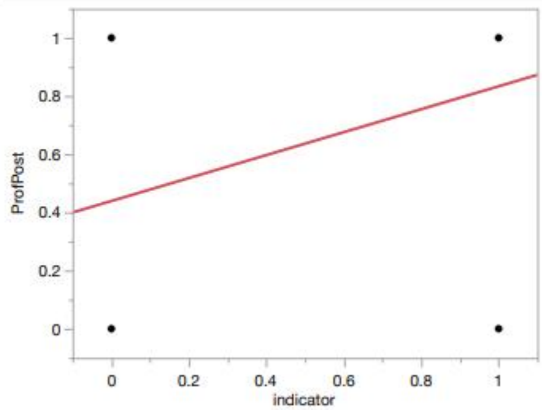
Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	68.536585	3.02107	22.69	<.0001*
indicator	12.713415	4.97178	2.56	0.0130*

APPENDIX E

ANOVA-PROFICIENCY POST JMP OUTPUT

Bivariate Fit of ProfPost By indicator



Linear Fit

Linear Fit

ProfPost = 0.4390244 + 0.3943089*indicator

Summary of Fit

RSquare	0.149115
RSquare Adj	0.135609
Root Mean Square Error	0.461724
Mean of Response	0.584615
Observations (or Sum Wgts)	65

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	1	2.353721	2.35372	11.0405
Error	63	13.430894	0.21319	Prob > F
C. Total	64	15.784615		0.0015*

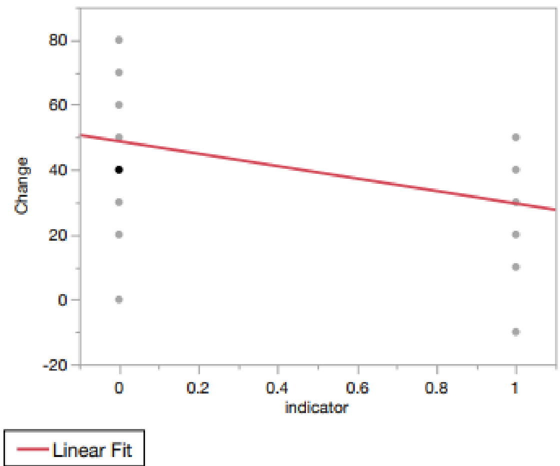
Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	0.4390244	0.072109	6.09	<.0001*
indicator	0.3943089	0.11867	3.32	0.0015*

APPENDIX F

ANOVA-CHANGE JMP OUTPUT

Bivariate Fit of Change By indicator



Linear Fit

Change = 48.780488 - 19.197154*indicator

Summary of Fit

RSquare	0.254587
RSquare Adj	0.242755
Root Mean Square Error	16.10228
Mean of Response	41.69231
Observations (or Sum Wgts)	65

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	1	5578.988	5578.99	21.5169
Error	63	16334.858	259.28	**Prob > F**
C. Total	64	21913.846		<.0001*

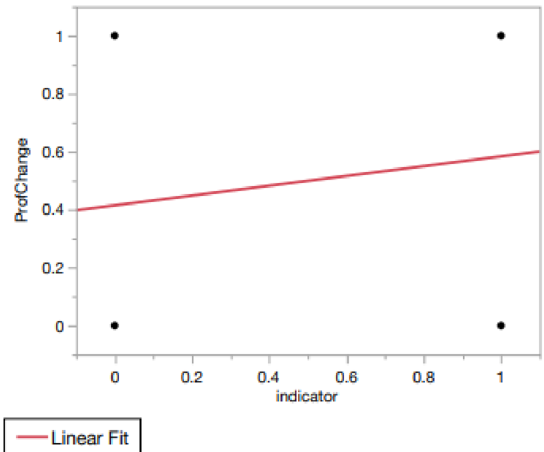
Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	48.780488	2.514754	19.40	<.0001*
indicator	-19.19715	4.138534	-4.64	<.0001*

APPENDIX G

ANOVA-PROFICIENCY CHANGE JMP OUTPUT

Bivariate Fit of ProfChange By indicator



Linear Fit

ProfChange = 0.4146341 + 0.1686992*indicator

Summary of Fit

RSquare	0.026569
RSquare Adj	0.011118
Root Mean Square Error	0.500548
Mean of Response	0.476923
Observations (or Sum Wgts)	65

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	1	0.430832	0.430832	1.7196
Error	63	15.784553	0.250548	Prob > F
C. Total	64	16.215385		0.1945

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	0.4146341	0.078172	5.30	<.0001*
indicator	0.1686992	0.128649	1.31	0.1945