AN ABSTRACT OF THE THESIS OF

Christopher Kawell for the degree of Master of Science in Computer Science presented on August 26, 2020.

Title: A Study on the Effect of Non-Traditional Demographic Populations in an Undergraduate Computer Science Course.

Abstract approved: ____

Jennifer Parham-Mocello

There is a significant amount of research analyzing the effect of race, gender, and other common demographical data on student interest and performance in computer science. However, there is relatively little research concerning less common demographic populations, such as introverts, artistic students, and visual learners. This study investigates if these less traditional demographic data affect student performance or interest in computer science and what effect delaying coding in introductory courses may have on these populations. We taught three sections of an introductory computer science course (one which delayed coding by half a term) and compared student performance and interest with non-traditional demographic data. We find relatively little correlation between student grades and their demographic data, but this study supports the idea that the delayed programming affects student interest.

A Study on the Effect of Non-Traditional Demographic Populations in an Undergraduate Computer Science Course

by
Christopher Kawell

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented August 26, 2020
Commencement June 2021

Master of Science thesis of Christopher Kawell presented on August 26, 2020

APPROVED:

Major Professor, representing Computer Science

Head of the School of Electrical Engineering and Computer Science

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Christopher Kawell, Author

ACKNOWLEDGEMENTS

Thank you to Jennifer Parham-Mocello for guiding me through this process and helping me as I taught CS 160, an experience completely new to me.

Also, my sincere appreciation to Sabrina Jesmin and Pallavi Sapale, fellow graduate students, for their huge help while teaching CS 160 and collecting data. Their suggestions and brainstorming sessions helped make the course a good experience for my students.

The undergraduate Teaching Assistants were also invaluable during CS 160. Thank you to Benjamin Friedman, Julian Fortune, Kaitlin Hill, Arthur York, Cheng Xie, Thomas Tonini, and Calvin Gagliano for leading my lab sessions and providing constructive feedback throughout the term.

Thank you also to Kirsten Winters and Raffaele De Amicis for helping to guide me during my first year in graduate school.

And final thanks to my family who encouraged me throughout the process.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

LIST OF APPENDIX FIGURES

Introduction

At the start of the century, Computer Science (CS) programs in higher academia struggled to gain interest [1, 2]. The massive need for more CS students continues to this day, but the combined effort of many recruitment and retention strategies – along with a growing interest in engineering programs – has lessened the concern [3]. However, numerous researchers argue that many of these recruitment and retention strategies serve to bias the field by focusing attention on specific demographics. Notably, Cheryan et al (2009) indicate that the use of examples that are stereotypically associated with computer science (e.g., video games) can have a negative impact on female undergraduate participation [4]. These demographic data are often politicized and have recently gained widespread attention, but less noticeable demographic characteristics may be equally important to address. Character traits such as discipline interests, personality, and learning styles often shape individuals' views of themselves similarly to social traits such as gender and race. In this thesis, we refer to groups of people who are sectioned via these character traits as Non-Traditional Demographic Populations (NTDP). NTDPs can be any group of individuals defined by a trait that is not commonly highlighted or recognized. In this research, we concentrate on NTDPs defined by discipline interests, personality, and learning styles, which are all explained in-depth in later sections. In this thesis, we seek to determine if certain NTDPs prefer and perform better with certain programming languages or non-code-centric methods of approaching computer science than do members of other NTDPs.

Related Work

Even though we refer to personality as a *non-traditional* demographic, there is a small amount of previous research concerning personality traits and computer science education. In 1995, Bishop-Clark states that personality traits are related to success in computer programming [5]. As their research regarded programming specifically (not computer science in general), they suggested that the significance of personality traits should be tested against the different stages of programming, instead of in its totality. Slightly more recently in 2003, Chandler et al conducted the Myers Briggs Type Indicator on CS students across three institutions [6]. They found that the majority of the CS population in their study tested as Sensing, Judging, and Introverted. From this, the researchers concluded that their students "Mentally live in the present; Naturally use targets, dates and standard routines to manage their lives; Are motivated internally; Are cautious, quiet, diligent and conscientious." (p. 5, quoted from bullets). In 2009, Salleh et al analyzed the effect of personality combinations when applying Paired Programming to a CS course but found no statistical correlation when pairing certain personalities. However, little research has been conducted regarding personality traits in CS education in more recent years.

There is more recent research on learning styles and computer science. Alharbi et al (2011) used the Index of Learning Styles (by Felder and Silverman) to determine the learning styles of a small cohort of CS students [7]. They found that their students were mainly Sensing types instead of Intuitive and were extremely preferential toward Visual learning over Verbal. In 2009 and 2010, Lau and Yuen also analyzed the relationship of computer science and learning styles, but their research focused on secondary school children and considered gender as a factor [8, 9]. They found that CS female students tended toward Concrete-Sequential and Abstract-Random learning styles, while male students generally preferred Concrete-Random [9]. These results caused the researchers to call for more pedagogical approaches to focus on learning style.

In a similar vein, CS Unplugged methodologies have gained popularity in recent years. CS Unplugged originated out of the University of Canterbury, New Zealand and strives to teach computer science without the direct use of computers [10]. This CS teaching strategy is usually implemented in pre-college classrooms and attempts to raise interest levels for and comprehension of computer science at a young age. In 2013, Thies and Vahrenhold compared

CS Unplugged to more conventional teaching strategies for middle school students [11]. These researchers found that there was no statistically significant difference between Unplugged methods and conventional teaching strategies (i.e., textbook-based, collaborative, and media-supported). They thus concluded that Unplugged methods may be implemented in similar educational environments without negative repercussions on learning. Similarly, in 2017, Hermans and Aivaloglou compared results of elementary students who began a term with four weeks of unplugged lessons as opposed to other students who began with more conventional CS education [12]. They found that – though both groups ended the term with four weeks of online Scratch programming – there was no statistical difference in programming mastery. Hermans and Aivaloglou's research agrees with the previously mentioned results of Thies and Vahrenhold, but Hermans and Aivaloglou found that the Unplugged students attained better self-efficacy and increased confidence in using Scratch.

Other research has also analyzed self-efficacy and confidence levels among CS students. In 2014, Barker et al. addressed concerns of student confidence by reframing the climate of the CS classroom to minimize students "showing off" with perceived intellectual prowess [13]. The researchers achieved this by instituting collaborative work, integrative lecture methods, paired programming, and explicitly stating the expectations. These efforts led to a more inviting course experience that caused students to feel more stimulated by the course subject, according to the end-of-term student survey. Similarly, in 2011 Anderson et al. addressed this concern of student intimidation and how it related to CS attrition of students into their second year [14]. Like CS Unplugged methods, the researchers taught an introductory CS course via a Graphical User Interface programming environment, instead of an industry programming language. They found that this method of teaching led to a high reduction of intimidation among students who initially self-rated as average programmers among their peers. Furthermore, because a large number of females in the class self-rated into this category, the researchers also concluded that this method of instruction helped raise confidence among women learning CS. In 2004, Rich, Perry, and Guzdial created a course specifically to address this concern of confidence levels among women exposed to CS [15]. In contrast to common programming courses, the research provided an alternative path to computer science that provided greater opportunities for creativity, higher levels of cross-disciplinary relevance and peer collaboration, as well as restricted registration that avoided unwanted displays of intellectual prowess mentioned previously. The research led to positive results, in that it led to lower levels of intimidation among women than the traditional course equivalent, as well as greater enjoyment and interest.

Researchers have implemented other methods of teaching introductory CS courses to increase areas such as self-efficacy and student retention. For example, Story Programming teaches computer science via storytelling (e.g., the story of Hansel and Gretel could be used to teach search algorithms, loops, etc.). In 2017, Parham-Mocello et al conducted research on the use of the Story Programming teaching methodology in CS [16]. The researchers taught one of three sections of an introductory CS course using the traditional manner while the other two were taught using Story Programming. In 2018, the researcher's conducted a similar study, but replaced the Haskell section with a section completely devoid of programming [17]. Similar to Thies and Vahrenhold's research on CS Unplugged Methods [11], Parham-Mocello et al did not see any negative impact on student success or learning while using Story Programming.

Research Claim and Questions

We believe some non-traditional demographic populations (NTDPs) *prefer* and *perform better with* a delayed-coding method of approaching computer science, than do members of other NTDPs. This research addresses this claim by answering 4 research questions.

RQ1: How do different social group's grades compare within each method of teaching Computer Science Orientation (CS0)?

This research question compares populations within a specific teaching method to gain an understanding of how NTDPs differ in the CS0 course.

RQ2: How do NTDPs' grades vary between the different methods of teaching CS0?

By comparing student final grades, we are able to measure each NTDP's performance in each course section.

RQ3: How does each method of teaching CS0 affect student interest and motivation in computer science?

Student interest and motivation play a large role in recruitment and retention. We use a pre- and a post-questionnaire that the students in all sections took at the beginning and end of the term to determine students' initial interest and motivation in each section and how these change over time. There are 6 questions asked in the pre-survey and again in the post-survey to measure student interest and how it changed after the class.

RQ4: Is there a correlation between student interest and motivation in a section and their demographic information?

This research question probes deeper into the questionnaire data mentioned in RQ3 to determine how interest and motivation in each section differs for members of different NTDPs. To answer this research question, we analyze student preference within each section and specific demographic information. Similar preferences within these groups would suggest that NTDPs influence student preference.

Research Methods

This research was conducted with undergraduate students taking the Computer Science Orientation course at Oregon State University (OSU). The following outline is from previous research using the same course [16].

> "At Oregon State University, students in the College of Engineering are required to take an orientation course to fulfill a degree requirement, and Computer Science Orientation (CS 0) is offered once a year to fulfill that requirement for students interested in majoring in CS. This course is primarily taken by incoming first-year students who are declared Computer Science majors, but students with prior computer science experience or outside the major may take the course as well. In the past, Python was used as the coding language with students beginning to write small programs as early as week two in a ten-week term. The lectures are focused on teaching basic Python including variables, control flow (both conditional statements and looping), functions and lists with exposure to how to design solutions to computer science problems and the idea of testing. In-class exercises are completed in groups and used to stimulate learning in a computer-free environment."

CS0 is a course primarily taken by first-year students majoring in computer science, though not strictly limited to majors, and is divided among three course sections. In Fall of 2019, when data was collected for this research, these sections contained roughly between 65 and 130 students each. Each section is taught by an instructor who leads the lectures and supervises 4-7 undergraduate teaching assistants, who lead the course labs and help with general course needs (grading, office hours, etc.). Each section covers the same course learning outcomes using different approaches, and often, multiple sections are led by the same instructor.

CS0 Structure

At OSU, CS0 has already been used as a testbed for experimental methods for CS education. In Fall 2017, Parham-Mocello et al [16] conducted research on the use of the Story

Programming teaching methodology in CS (described on page 4 of this thesis). The researchers taught one of the three sections of CS0 in the traditional manner, while the other two were taught using Story Programming. This is an approach that uses stories to explain computational concepts and delays coding for the first 5 weeks of the 10-week term. One Story Programming section used Python to introduce programming, while the other section used a functional language called Haskell. In the Fall of 2018, the researchers had 2 traditional sections while the other 2 were replaced with sections that did not involve programing [17]. The course catalog was also updated to express these differences, and advisors were encouraged to inform students of these variations.

For this research, in the Fall of 2019, the CS0 sections were divided based on student experience with programming. The intermediate and experienced sections were identical in lecture and language using Python, but they differed in their lab, where the intermediate section used Cozmo robots to motivate and teach programming. The intermediate and experienced sections were taught by the same instructor, and the beginner section was taught by a different instructor using a different pedagogical approach and Haskell as the programming language. In the beginner's section, programming was delayed until the second half of the term, and Story Programming was central to the first half. All sections' lectures were held nearly at the same time on the same days to discourage students from choosing a section based on schedule or time preference. Throughout the rest of this thesis, we will refer to the beginner section as the delayed coding section, as this was its main distinguishing characteristic.

OSU advisors were asked to encourage students to choose a section based on previous experience, and the OSU course catalogue reflected this as well. These precautions were generally successful. Though, near the beginning of the term, a few students chose to change sections due to experience level (some because they had more experience for their current section while others because they had less). At the beginning of the term, students were asked if they wanted to participate in the research study, and we only include results from students agreeing to participate.

Participants

The total number of students who consented and took part in the study was 256. To factor in students who chose to change sections near the beginning of the term, we used the courses' final grades to determine student final placements. For this reason, we only include students who completed one of the course sections. These numbers in each course section were 87, 113, and 56 for the beginner, intermediate, and experienced sections, respectively (Table 1). There was also a small honors section from which we attained 16 consenting students, but this population is not considered in this study, as it does not help to answer our research questions nor is a large enough population to reach confident conclusions. The total number of students who took CS0 in Fall 2019 was 320, but this study uses the data of 256 students after consent, removal of the honors section, and students dropped the course. To better determine NTDPs, all students were required to take a learning styles questionnaire and a personal identity questionnaire as part of their assignment (both explained in detail later), but only the answers of the consenting students are included in this study.

Table 1.1: Final Placement of Consenting Students

Beginner	Intermediate	Experienced
87	113	56

Non-Traditional Demographic Populations

To group students by non-traditional demographics, all students were required to take a personal identity questionnaire and a learning styles questionnaire. The personal identity questionnaire was composed of a series of statements for which the students rated how much they agreed or disagreed (1 = completely disagree, 4 = completely agree). Similar to Goodwin and Lee's research on undergraduate engineering identity [18], most of the statements were worded to capture personal belief, not hard fact or the perceived view of others. For example, the statement concerning the students' identity as a scientist was phrased "I think of myself as a science person" instead of "*I am* a science person" or "*people view me as* a science person." There were 4 statements that followed this template: "I think of myself as a [science person] / [math person] / [engineer] / [artistic person]." Two other statements ("I am good with computers" and "I am an introverted person") were phrased more naturally. It is important to understand that the demographic populations measured by this questionnaire are not objective characteristics of the CS students requiring formal definitions of concepts such as "artistic" or "introverted." Instead, this questionnaire seeks to capture students' personal view or opinion of themselves, suggesting that such personal views may affect a student's performance and interest in CS. For example, this research does not seek to understand how an artistically-oriented brain approaches CS, but instead seeks to understand how a student's opinion of themselves as "artistic" affects their performance and interest in CS.

The questionnaire was answered with pen and paper, thus some answers were not expressed as 1, 2, 3, and 4. In the process of cleaning the data, we treated an answer of 0 the same as 1 (completely disagree) and all 2.5s were discarded, as they expressed complete neutrality between "agree" and "disagree" and consisted of only 11 total responses across all 6 questions. 1s, 2s, and 1.5s were treated as disagreements with the questionnaire statement and 3s, 4s, and 3.5s were treated as agreements with the statement.

Table 1.2: Personal Identity Questionnaire with Possible Responses

Statement/Question	**Possible Response**
I think of myself as a science person	1 – 4
I think of myself as a math person	1 – 4
I think of myself as an engineer	1 – 4
I am good with computers	1 – 4
I think of myself as an artistic person	1 – 4
I am an introverted person	1 – 4

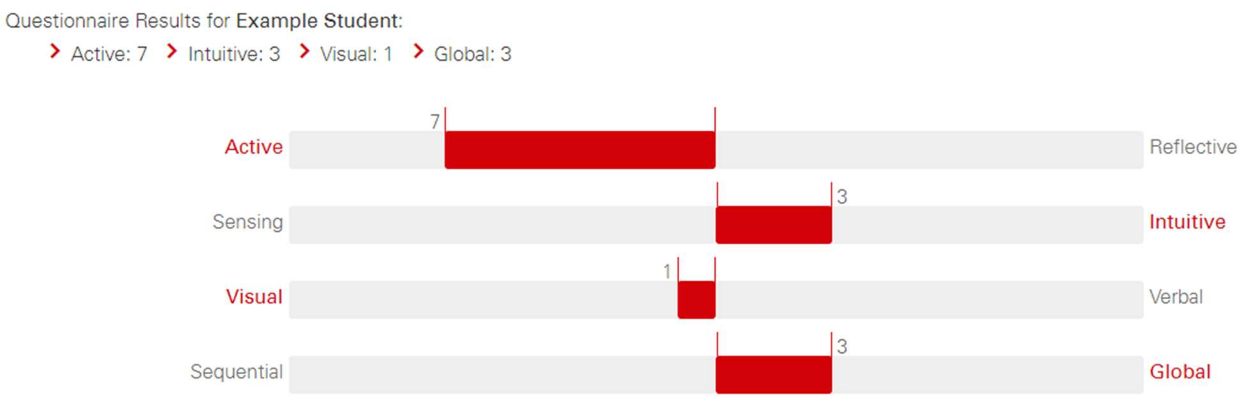
The learning styles questionnaire used in this study was created by Felder and Silverman and is commonly used to assess individual's preferences along 4 dimensions of learning styles [19]. The four dimensions are provided below, as stated in Felder and Spurlin's article on the questionnaire [20]:

- "*sensing* (concrete thinker, practical, oriented toward facts and procedures) or *intuitive* (abstract thinker, innovative, oriented toward theories and underlying meanings).

- *visual* (prefer visual representations of presented material, such as pictures, diagrams and flow charts) or *verbal* (prefer written and spoken explanations).
- *active* (learn by trying things out, enjoy working in groups) or *reflective* (learn by thinking things through, prefer working alone or with a single familiar partner).
- *sequential* (linear thinking process, learn in small incremental steps) or *global* (holistic thinking process, learn in large leaps)."

After answering the series of questions, the students were provided a rating for each dimension which expressed where the student was on that spectrum. The rating was an odd number between 1 and 11. For example, the student in figure 1 received a score of 7 on the Active side of the Active/Reflective spectrum, which means that the student tends to be more of an active learner than a reflective learner, as can be seen in figure 1. This questionnaire was administered to the students only once late in the term since the course was not expected to have serious impact on student learning styles.

Figure 1: Example Result of the Learning Styles Questionnaire.



Addressing Research Questions

To answer *RQ1* and *RQ2*, we collected the final grades of the consenting students. As each section was graded via the 5-point grading scale most common in American higher education, we were able to compare the student grades from one section with another. By grouping students by NTDPs, we are able to compare the performance of different demographic populations across sections. The sections did not have identical assignments, so we are unable to compare grades for assignments across different sections. For this reason, we chose to only analyze final grades.

Using the data collected from the personal identity questionnaire and the learning styles questionnaire, we were able to study 20 different non-traditional demographic populations, each of the populations with an opposite population: introvert vs extrovert, scientific person vs non-scientific person, artistic vs non-artistic, engineer vs non-engineer, math person vs non-math person, comfortable with computers vs not comfortable, active vs reflective, sensing vs intuitive, visual vs verbal, and sequential vs global. In this research, we compare populations by measuring them against their contrasting NTDP, and throughout this research study, we will refer to students who *self-identified* as a member of a population (e.g., “I think of myself as an artistic person”) as a member of that population (e.g., an artistic person) for simplicity’s sake.

Results and Discussion

Table 2.1 shows the percentage of all students in the study that identified in each of the demographic populations. In 9 of the 10 populations studied, the students are fairly well balanced between each NTDP and its contrasting NTDP (neither representing more than two-thirds of the total population). However, students are overwhelmingly more Visual than Verbal learners (83% Visual) and this holds true across all three sections. Tables 2.2, 2.3, and 2.4 show the percentages for each of these sections, highlighting with bold all contrasting NTDPs which are vary in size by more than two-thirds. Note that far more students identify as science people and engineers than otherwise in both the Intermediate and Experienced sections. More detailed information on the subject can be found in Appendix A.

Table 2.1: NTDP Percentages of Total Students in the Study

NTDP	Percent of Students	Contrasting NTDP	Percent of Students
Active	47%	Reflective	53%
Sensing	62%	Intuitive	38%
Visual	**87%**	**Verbal**	**13%**
Sequential	63%	Global	37%
Science Person	66%	Non-Science Person	34%
Math Person	61%	Non-Math Person	39%
Engineer	65%	Non-Engineer	35%
Comfortable with Computers	41%	Not Comfortable with Computers	59%
Artistic Person	47%	Non-Artistic Person	53%
Introvert	45%	Extrovert	55%

Table 2.2: NTDP Percentages in the Delayed-Programming Section

NTDP	Percent of Students	Contrasting NTDP	Percent of Students
Active	44%	Reflective	56%
Sensing	65%	Intuitive	35%
Visual	**89%**	**Verbal**	**11%**
Sequential	59%	Global	41%
Science Person	47%	Non-Science Person	53%
Math Person	64%	Non-Math Person	36%
Engineer	47%	Non-Engineer	53%
Comfortable with Computers	56%	Not Comfortable with Computers	44%
Artistic Person	39%	Non-Artistic Person	61%
Introvert	33%	Extrovert	67%

Table 2.3: NTDP Percentages in the Intermediate Section

NTDP	Percent of Students	Contrasting NTDP	Percent of Students
Active	52%	Reflective	48%
Sensing	61%	Intuitive	39%
Visual	**85%**	**Verbal**	**15%**
Sequential	**71%**	**Global**	**29%**
Science Person	**74%**	**Non-Science Person**	**26%**
Math Person	60%	Non-Math Person	40%
Engineer	**73%**	**Non-Engineer**	**27%**
Comfortable with Computers	36%	Not Comfortable with Computers	64%
Artistic Person	50%	Non-Artistic Person	50%
Introvert	47%	Extrovert	53%

Table 2.4: NTDP Percentages in the Experienced Section

NTDP	Percent of Students	Contrasting NTDP	Percent of Students
Active	33%	Reflective	67%
Sensing	57%	Intuitive	43%
Visual	**86%**	**Verbal**	**14%**
Sequential	55%	Global	45%
Science Person	**75%**	**Non-Science Person**	**25%**
Math Person	57%	Non-Math Person	43%
Engineer	**70%**	**Non-Engineer**	**30%**
Comfortable with Computers	35%	Not Comfortable with Computers	65%
Artistic Person	51%	Non-Artistic Person	49%
Introvert	59%	Extrovert	41%

RQ1: How do different social group's grades compare within each method of teaching CS0?

This research question compares populations within a specific teaching method to gain an understanding of how NTDPs differ in the CS0 course. We find little correlation between NTDPs and student grades in each section, except among artistic vs non-artistic students in the delayed coding section and among students who feel comfortable vs not comfortable with computers in the intermediate section.

Because the data are not normally distributed, we use the non-parametric Wilcoxon Signed Rank test to compare the students' final grades of each population against its opposite (e.g., math person vs non-math person). The alternative hypothesis for RQ1 is that the NTDPs are different populations based on final grades (i.e., student final grades vary based on student NTDP). Because we compare each population to its opposite and compare populations within each of the 3 sections, this results in 30 Wilcoxon Tests to answer RQ1.

Out of 30 tests, grades from only two NTDPs statistically differed with p-values < 0.05: artistic vs non-artistic students in the delayed coding section (p-value=0.014) and students who feel comfortable vs not comfortable with computers in the intermediate section (p-value=0.002; see figures 2.1 and 2.2). The non-artistic students in the delayed-coding section have slightly higher grades than the students who think of themselves as artistic; whereas, students who thought of themselves as more comfortable with computers did slightly worse in the intermediate section.

This largely contradicts the idea that these non-traditional demographic data affect a student's course performance. The two statistically significant correlations do little to dispel this conclusion, as both appear as isolated results, neither consistent across all sections nor consistent within a section. However, it is worth noting that students in the intermediate section who self-identified as being good with computers scored lower than those who claimed not to be good with computers. This may be an example of overconfidence, where students' past experience in the material has given them a false sense of preparation. Similar trends have been seen in other college subjects such as economics [21], but more research is needed to determine if such a trend exists among computer science students.

Figure 2.1: Artistic vs. Non-Artistic final grades in the Delayed-Programming section.

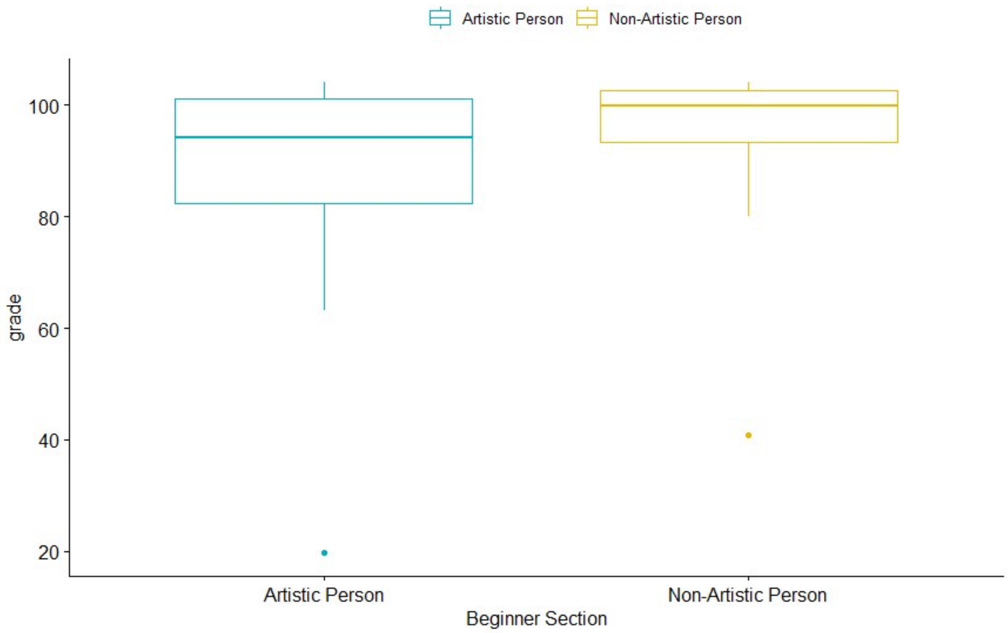
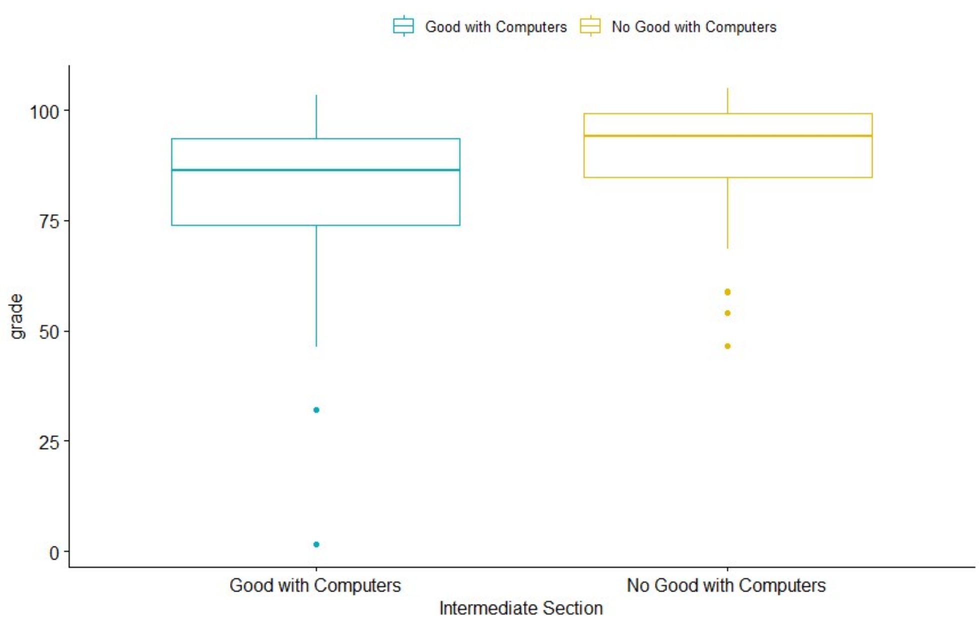


Figure 2.2: Good with Computers vs. Not Good final grades in the Intermediate section.



RQ2: How do NTDPs' grades vary between the different methods of teaching CS0?

By comparing student final grades, we are able to measure each NTDP's performance in each course section. Unlike for RQ1 where we compared NTDPs to their opposites, to answer this question, we compare the NTDPs in each section against the identical NTDPs in the other sections. We find correlations between final grades and the course section for 9 of the demographic populations: Extroverts, introverts, non-artistic people, engineers, math people, people comfortable with computers, science people, non-science people, and visual learners (see figure 2).

Similar to data in RQ1, the data for this question are not normally distributed. Since there are more than two independent populations in RQ2, we use the non-parametric Kruskal-Wallis test to compare each NTDP's final grades against each of the three course sections. The alternative hypothesis for RQ2 is that each NTDPs *in each section* are different populations based on final grades (i.e., a population's final grades vary based on course section). For each test that returned a p-value < 0.05, we conduct a Bonferroni t-test to determine which sections are significantly different. Because we compare each population in each section to the identical population in each of the other sections, there are a total of 20 Kruskal-Wallis tests to answer RQ2.

After conducting Bonferroni t-tests, we find that there is a significant difference between NTDP final grades for the following sections, measuring for an adjusted p-value < 0.05:

- Introverts in the delayed coding and experienced sections (p-value=0.017).
- Extroverts in the delayed coding and intermediate sections (p-value=0.004).
- Non-artistic people in the delayed coding and intermediate sections (p-value≈0).
- Non-artistic people in the delayed coding and experienced sections (p-value=0.01).
- Engineers in the delayed coding and intermediate sections (p-value=0.04).
- Math people in the delayed coding and intermediate sections (p-value≈0).
- People comfortable with computers in the delayed coding and intermediate sections (p-value≈0).

- People comfortable with computers in the delayed coding and experienced sections (p-value=0.03).
- Science people in the delayed coding and intermediate sections (p-value=0.048).
- Non-science people in the delayed coding and intermediate sections (p-value≈0).
- Visual people in the delayed coding and intermediate sections (p-value=0.02).
- Visual people in the delayed coding and experienced sections (p-value=0.033).

In each case, there is a significant difference between the delayed coding section and one of the other sections. This result may be influenced by the fact that the delayed coding section was taught by a different instructor than the other two sections. However, less than half of the tests return a significant difference, which suggests a more complex explanation than an overall difference in grading styles. The most notable results for RQ2 are the 3 NDTPs which have statistically significant differences when comparing the delayed coding section to each of the other sections (non-artistic people, people good with computers, and visual learners). These results suggest that students who self-identified as non-artistic or good with computers performed better in the delayed coding section than both other sections. The reason for this is unclear, but it may be another example of overconfidence interfering with performance in the "more advanced" sections, similar to the results for RQ1. The results also suggest that visual learners performed better in the delayed coding section than in the other sections. This makes sense, as the delayed coding section implemented stories, videos, and images in place of text-based programming for the first half of the term; whereas, the two other sections use text-based programming throughout the entire term.

Figure 2.3: Extroverts' final grades per section.

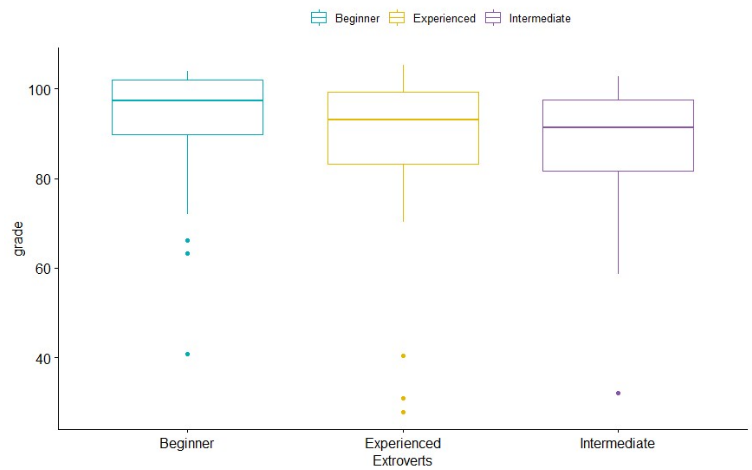


Figure 2.4: Introverts' final grades per section.

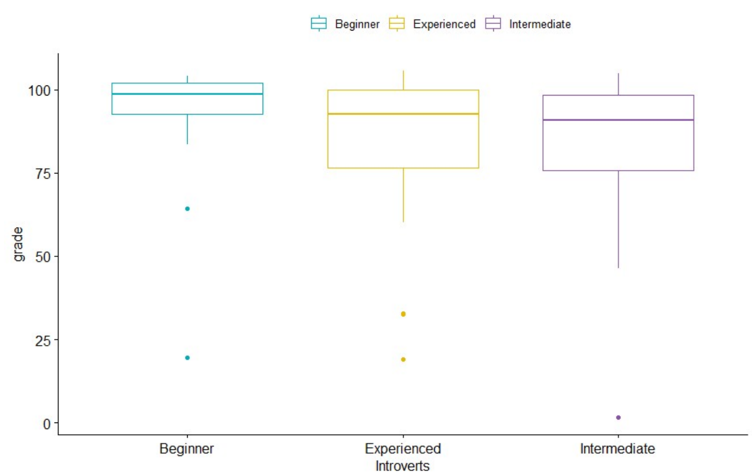


Figure 2.5: Non-Artistic People's final grades per section.

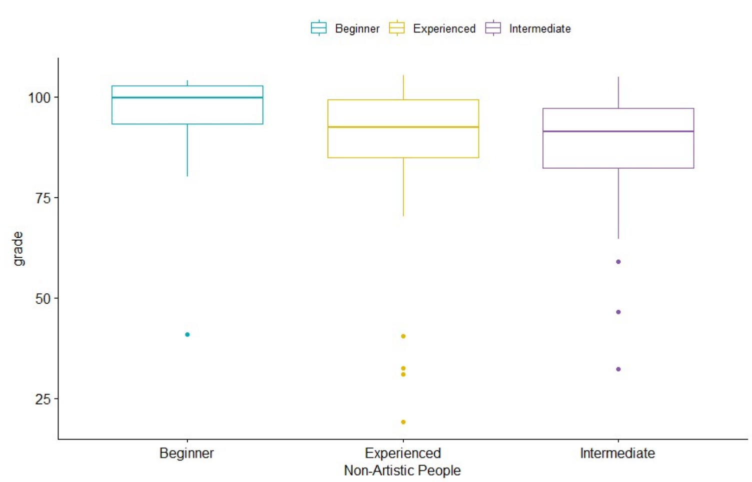


Figure 2.6: Engineers' final grades per section.

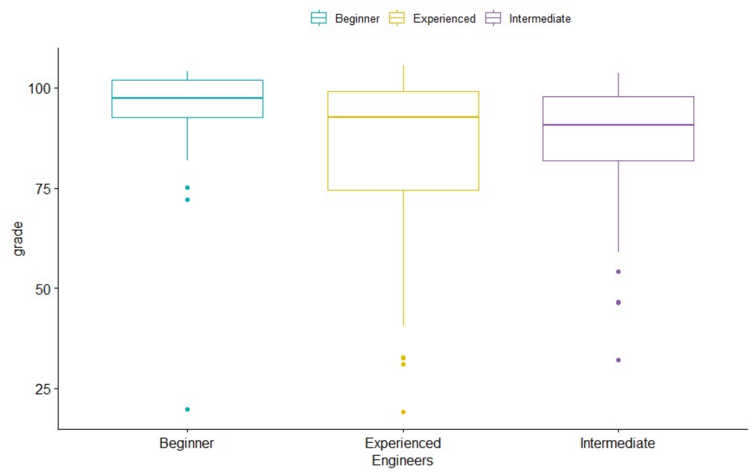


Figure 2.7: Math People's final grades per section.

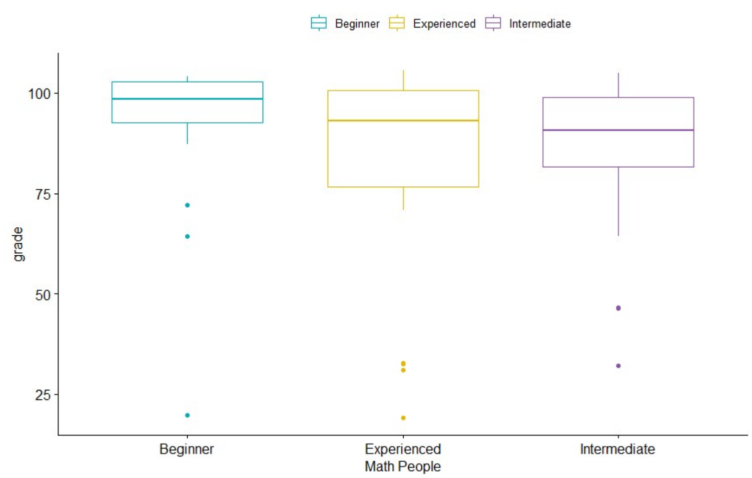


Figure 2.8: People Good with Computers' final grades per section.

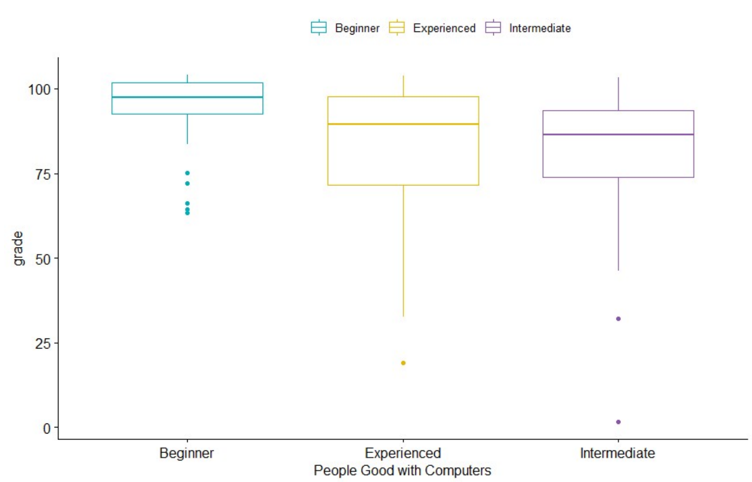


Figure 2.9: Science People's final grades per section.

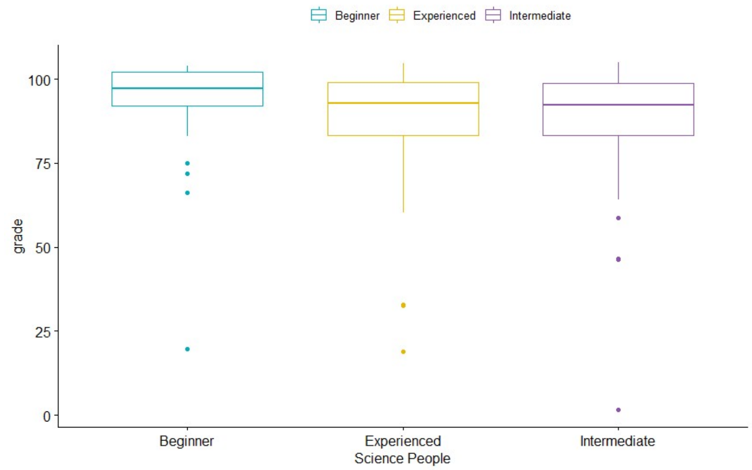


Figure 2.10: Non-Science People's final grades per section.

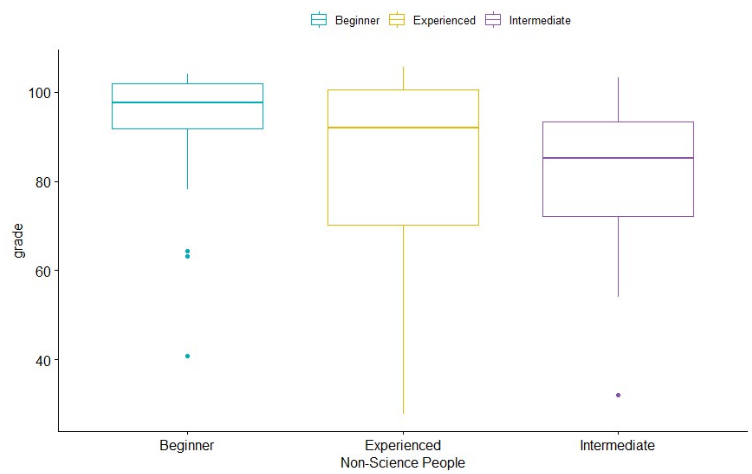
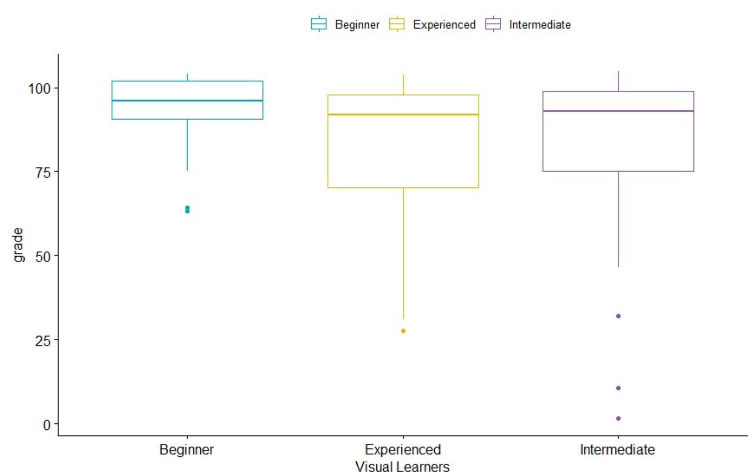


Figure 2.11: Visual Learners' final grades per section.



RQ3: How does each method of teaching CS0 affect student interest in computer science?

Student interest and motivation play a large role in recruitment and retention. We use a pre- and a post-questionnaire that the students in all sections took at the beginning and end of the term to determine students' initial interest and motivation in each section and how these change over time. Note that RQ3 does not concern NTDPs, but simply measures student interest between course sections. As stated above in the Research Questions section, we measure student interest using 6 questions asked first in the pre-survey and again in the post-survey. Students rate their interest level in the following questions:

- "This class."
- "Learning more about computer science."
- "Learning more about programming/coding."
- "Majoring in computer science."
- "Taking more computer science classes."
- "Using computation in my job after college."

Each question has the following options: Extremely Interested, Somewhat Interested, and Not Interested at All. To evaluate how student interests change over time, we coded the responses from 3 to 1 respectively and subtracted each student's pre-survey responses from their post-survey responses. A positive value implies a positive change in interest, while a negative response indicates a negative change (0 means no change in interest). We find a correlation between students' change in interest and the course section in only 2 of the 6 areas of interest measured: "This class" and "Taking more computer science classes."

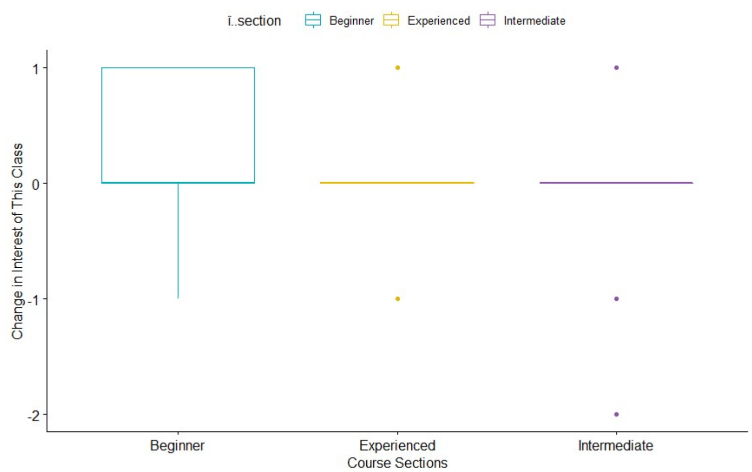
We use a Kruskal-Wallis test to compare each section's change in student interest level. The alternative hypothesis for RQ3 is that the students in each course section are different populations based on student change in interest (i.e., a population's change in interest vary based on course section). For each test that returns a p-value < 0.05, we use a Bonferroni t-test to determine which sections significantly differed. Because we compare the population in each section to the populations in each of the other sections for each of the interest level questions, there are 6 Kruskal-Wallis tests to answer RQ3.

After conducting Bonferroni t-tests, we find that there is a significant difference between the change in interest for the following sections, measuring for an adjusted p-value < 0.05:

- The delayed coding and intermediate sections for interest level in this class (p-value≈0).
- The delayed coding and experienced sections for interest level in this class (p-value=0.017).
- The delayed coding and intermediate sections for interest level in taking more computer science classes (p-value=0.006).

As each of these significant differences include the delayed coding section, RQ3 supports the idea that the delayed programming in the delayed coding section affects student interest especially in the current course but also in the students' desire to take more computer science courses in the future.

Figure 2.12: Change in Interest of This Class per section.



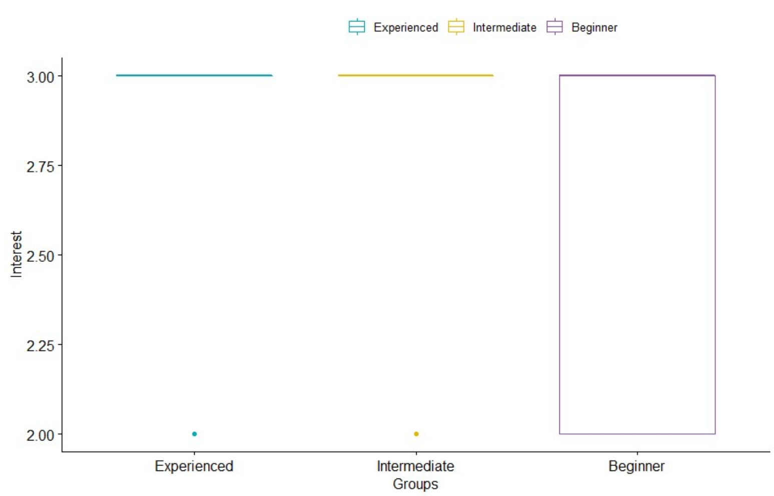
RQ4: Is there a correlation between student interest and their demographic information?

This research question probes deeper into the questionnaire data mentioned in RQ3 to determine how interest in each section differs for members of different NTDPs. To answer this research question, first we analyze each demographic population's final interest level for each interest area in the post-survey, and then, we compare the data between each course section. We find nearly no correlation between an NTDP's final interest level and course section. We analyze final interest level here instead of change in interest because final interest captures how NTDP interest levels differ. In RQ3, we were interested only in how each section affected interest levels, thus we measured change in interest. However, here we analyze each demographics interest after taking a course section, whether or not they changed significantly over the term.

We use a Kruskal-Wallis test to compare each NTDP's interest level between course sections. The alternative hypothesis is that the NTDP's final interest levels are different populations based on course (i.e., a population's final interest vary based on course section). For each test that returned a p-value < 0.05, we conduct a Bonferroni t-test to determine which sections were significantly different. Because we compare each of the 20 population's responses to each of the 6 interest level questions, there are a total of 120 Kruskal-Wallis tests to answer RQ4. Out of the 120, only 1 test returned a p-value < 0.05, for which the Bonferroni t-test shows a significant difference between the delayed coding and intermediate sections for students who prefer the sensing learning style and interest in "Learning more about programming/coding", p-value = 0.042 (see figure 4).

As 119 out of the 120 tests failed to reject the null hypothesis, the implication is that the NTDPs analyzed in this research do not differ in interest between course sections. However, the researchers notice that the significant majority of students responded with an interest level of 3 across all sections and all interest level questions (20 responses with interest level 1, 452 for interest level 2, and 3114 for interest level 3). This means that the data are heavily saturated with high interest levels. Providing students with more than 3 options for future studies may resolve this.

Figure 2.13: Final Interest in Learning more about programming/coding per section.



After comparing each NTDPs' interest level between course sections, we compare each NTDP's interest level to its opposite within each section (e.g., the interest level for introverts vs. extroverts in the delayed coding section). We find some correlation between engineers and non-engineers in all three sections.

We use a Wilcoxon Signed Rank test to compare each contrasting NTDP's interest level in each course section. The alternative hypothesis is that the section's final interest levels are different populations based on NTDP (i.e., a section's final interest vary based on NTDP). Because our grouping variable has only 2 options (the NTDP and its opposite), there is no need to run Bonferroni t-tests. Because we compared the 10 populations to their opposites for each of the 6 interest level questions in each of the 3 course sections, there are a total of 180 Wilcoxon Signed Rank tests. Out of the 180, 26 tests have a p-value < 0.05 (rejecting the null hypothesis). We provide the complete list of 26 tests in table 2.

Though 26 tests out of 180 may seem few, it demonstrates a far greater likelihood for variation between contrasting NTDPs within a section than NTDPs compared across all sections (tested on the previous page). Most notably, there is a statistically significant difference between engineers and non-engineers for many of the questions across all 3 course sections. This suggests that students who self-identified as engineers had a different interest level than student who did not consider themselves to be engineers.

Table 2.5: All contrasting NTDPs per interest area in each section that returned significant differences.

Comparison	Interest level in…	Course Section
Artistic vs. Non-artistic	Taking more computer science classes	Delayed Coding
People comfortable vs. Not comfortable with computers	Using computation in my job after college	Delayed Coding
People comfortable vs. Not comfortable with computers	This class	Intermediate
Engineers vs. Non-engineers	Learn more about computer science	Delayed Coding
Engineers vs. Non-engineers	Majoring in computer science	Delayed Coding
Engineers vs. Non-engineers	Using computation in my job after college	Delayed Coding
Engineers vs. Non-engineers	Learning more about programming/coding	Experienced
Engineers vs. Non-engineers	Majoring in computer science	Experienced
Engineers vs. Non-engineers	Taking more computer science classes	Experienced
Engineers vs. Non-engineers	Using computation in my job after college	Experienced
Engineers vs. Non-engineers	Learning more about programming/coding	Intermediate
Engineers vs. Non-engineers	Majoring in computer science	Intermediate
Engineers vs. Non-engineers	Taking more computer science classes	Intermediate
Engineers vs. Non-engineers	Using computation in my job after college	Intermediate
Introverts vs. Extroverts	Using computation in my job after college	Delayed Coding
Math vs. Non-math	Using computation in my job after college	Delayed Coding
Science vs. Non-science	Learn more about computer science	Delayed Coding
Science vs. Non-science	Using computation in my job after college	Delayed Coding
Science vs. Non-science	Learning more about programming/coding	Intermediate
Science vs. Non-science	Learn more about computer science	Intermediate
Science vs. Non-science	Majoring in computer science	Intermediate
Science vs. Non-science	Taking more computer science classes	Intermediate
Visual vs. Verbal Learner	Learning more about programming/coding	Delayed Coding
Visual vs. Verbal Learner	Learn more about computer science	Delayed Coding
Visual vs. Verbal Learner	Majoring in computer science	Delayed Coding
Visual vs. Verbal Learner	Using computation in my job after college	Delayed Coding

Conclusion

In this thesis, we investigate if certain non-traditional demographic populations prefer or perform better with certain programming languages or non-code-centric methods of approaching computer science than do members of other NTDPs. While the results are largely inconclusive concerning student performance, this study provides important insights into the relationship of NTDPs and computer science students, particularly concerning student interest.

The study finds little difference in final grades between contrasting NTDPs within each section. Though comparisons between sections suggest better performance for many NTDPs in the delayed coding section, this may largely be due to different teaching and grading styles. However, the research suggests that demographic populations that self-identify more closely with the sciences sometimes experienced lower final grades. This may be due to overconfidence, but more research is needed to determine the cause. Furthermore, visual learners performed better in the delayed coding section than either of the other sections, which suggests that delaying programming in the course helps this set of learners. This is significant, as the vast majority of students in the study show a preference for the visual learning style, but more research is needed to determine the extent of this effect.

Concerning student interest, this study supports the idea that the delayed programming affects student interest, as we see a greater change in students' interest in the course and in the students' desire to take more computer science courses in the future. However, we see almost no difference in student interest between course sections among NTDPs, though the data are heavily saturated with high interest levels. Collecting data with a greater variety in responses may have more varying results. This study also finds a significant difference in student interest between some contrasting NTDPs within each section. Most notably, there is a statistically significant difference between engineers and non-engineers across all 3 course sections. This suggests that students who self-identify as engineers have a different interest level than students who do not consider themselves to be engineers. As this course is an orientation course, many students may not have established an identity as an engineer, and this appears to affect their level of interest for the field of computer science. This computer science program is part of the School of Electrical Engineering and Computer Science in the College of Engineering, so this result may not be surprising. However, it underlines the necessity to create curriculum that addresses student identity and encourages students to view themselves as a part of their school and college, not just as a computer scientist.

References

[1] C. Kelleher and R. Pausch, “Using storytelling to motivate programming,” Commun. ACM, vol. 50, no. 7, p. 58, Jul. 2007, doi: 10.1145/1272516.1272540.

[2] J Vegso. “Drop in CS bachelor’s degree production,” Computing Research News, vol 18, no. 2, p. 5, Mar. 2006.

[3] 2016. The Condition Of STEM 2016. [pdf] Available at: <https://www.act.org/content/dam/act/unsecured/documents/STEM2016_52_National.pdf> [Accessed 13 August 2020].

[4] S. Cheryan, V. C. Plaut, P. G. Davies, and C. M. Steele, “Ambient belonging: How stereotypical cues impact gender participation in computer science.,” Journal of Personality and Social Psychology, vol. 97, no. 6, pp. 1045–1060, 2009, doi: 10.1037/a0016239.

[5] C. Bishop-Clark, “Cognitive style, personality, and computer programming,” Computers in Human Behavior, vol. 11, no. 2, pp. 241–260, 1995, doi: 10.1016/0747-5632(94)00034-F.

[6] J Chandler, J Carter, and I Benest, “Extrovert or Introvert? The Real Personalities of Computing Students,” in the 4th Annual LTSN-ICS Conference, NUI Galway. 2003.

[7] A Alharbi, D Paul, F Henskens, and M Hannaford, “An Investigation into the Learning Styles and Self-Regulated Learning Strategies for Computer Science Students,” Dec, 2011.

[8] W. W. F. Lau and A. H. K. Yuen, “Exploring the effects of gender and learning styles on computer programming performance: implications for programming pedagogy,” British Journal of Educational Technology, vol. 40, no. 4, pp. 696–712, Jul. 2009, doi: 10.1111/j.1467-8535.2008.00847.x.

[9] W. W. F. Lau and A. H. K. Yuen, “Gender differences in learning styles: Nurturing a gender and style sensitive computer science classroom,” AJET, vol. 26, no. 7, Dec. 2010, doi: 10.14742/ajet.1036.

[10] CSUnplugged.org. Computer Science Unplugged. http://csunplugged:org/ [Accessed 27 May 2020], 2014.

[11] R. Thies and J. Vahrenhold, “On plugging ‘unplugged’ into CS classes,” in Proceeding of the 44th ACM technical symposium on Computer science education - SIGCSE ’13, Denver, Colorado, USA, 2013, p. 365, doi: 10.1145/2445196.2445303.

[12] F. Hermans and E. Aivaloglou, “To Scratch or not to Scratch?: A controlled experiment comparing plugged first and unplugged first programming lessons,” in Proceedings of the 12th Workshop on Primary and Secondary Computing Education, Nijmegen Netherlands, Nov. 2017, pp. 49–56, doi: 10.1145/3137065.3137072.

[13] L. J. Barker, M. O’Neill, and N. Kazim, “Framing classroom climate for student learning and retention in computer science,” in Proceedings of the 45th ACM technical symposium on Computer science education - SIGCSE ’14, Atlanta, Georgia, USA, 2014, pp. 319–324, doi: 10.1145/2538862.2538959.

[14] M. Anderson, A. McKenzie, B. Wellman, M. Brown, and S. Vrbsky, “Affecting attitudes in first-year computer science using syntaxfree robotics programming,” ACM Inroads, vol. 2, no. 3, pp. 51–57, Aug. 2011, doi: 10.1145/2003616.2003635.

[15] L. Rich, H. Perry, and M. Guzdial, “A CS1 course designed to address interests of women,” SIGCSE Bull., vol. 36, no. 1, p. 190, Mar. 2004, doi: 10.1145/1028174.971370.

[16] Parham-Mocello, J., Ernst, S., Erwig, M., Shellhammer, L., and Dominguez, E. (2019). Story Programming: Explaining Computer Science Before Coding. Special Interest Group in Computer Science Education (SIGCSE 2019).

[17] Jennifer Parham-Mocello and Martin Erwig. 2020. Does Story Programming Prepare for Coding? In the 51st ACM Technical Symposium on Computer Science Education (SIGCSE’20). ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3328778.3366861.

[18] Godwin, A., & Lee, W. C. (2017, June), A Cross-sectional Study of Engineering Identity During Undergraduate Education Paper presented at 2017 ASEE Annual Conference & Exposition, Columbus, Ohio. https://peer.asee.org/27460

[19] Felder, R., & Soloman, B. (1997). Index of Learning Styles Questionnaire. Retrieved August 13, 2020, from https://www.webtools.ncsu.edu/learningstyles/index.php

[20] R Felder and J Spurlin, “Applications, Reliability and Validity of the Index of Learning Styles” in the Int. J. Engng Ed. Vol. 21, No. 1, pp. 103-112, 2005.

[21] Grimes, P. (2002). The Overconfident Principles of Economics Student: An Examination of a Metacognitive Skill. The Journal of Economic Education, 33(1), 15-30. Retrieved July 29, 2020, from www.jstor.org/stable/1183081

Appendix A: Demographic Data

Below is more detailed information on the spread of demographic data collected in this study.

Personal Identity Questionnaire Results

Figure 3.1: Total responses to the Personal Identity Questionnaire.

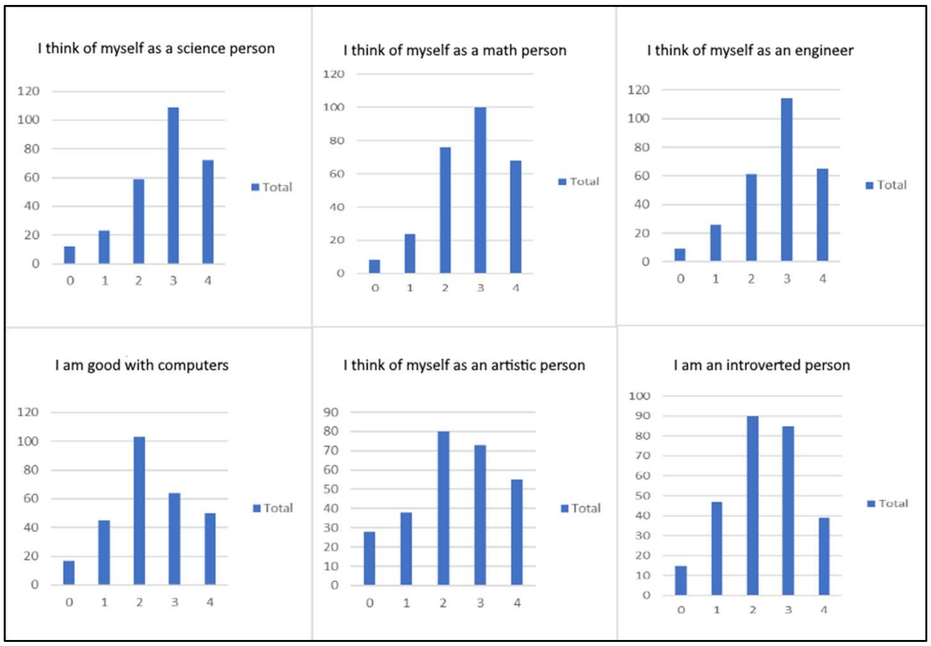


Figure 3.2: Delayed-Coding section responses to the Personal Identity Questionnaire.

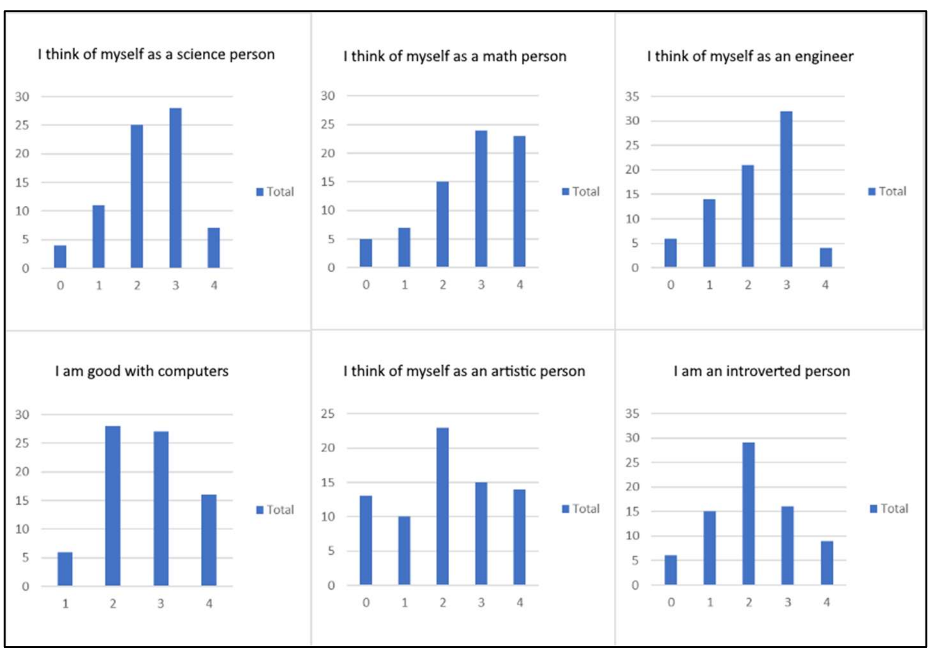


Figure 3.3: Intermediate section responses to the Personal Identity Questionnaire.

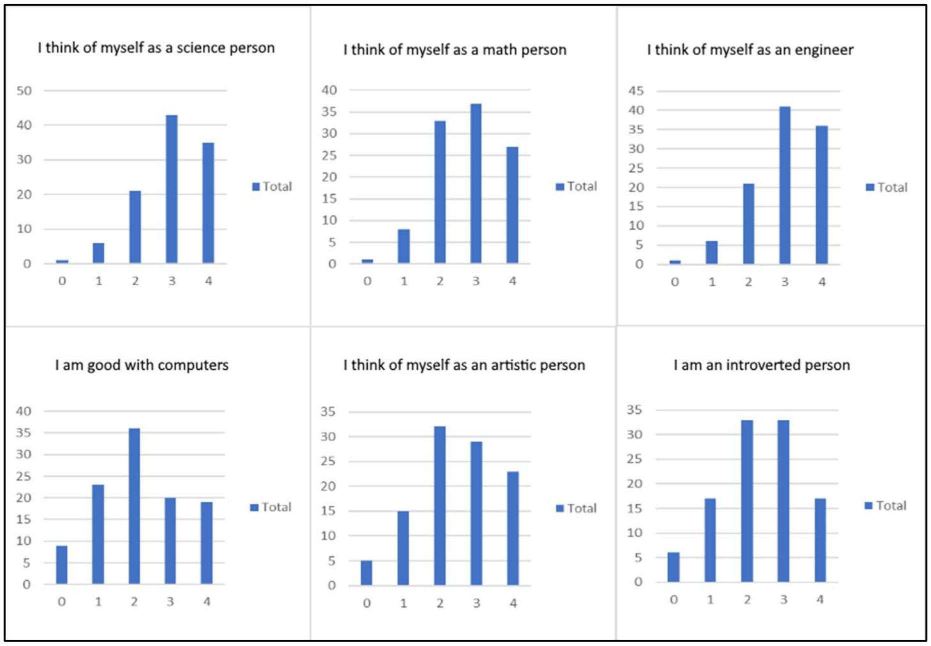
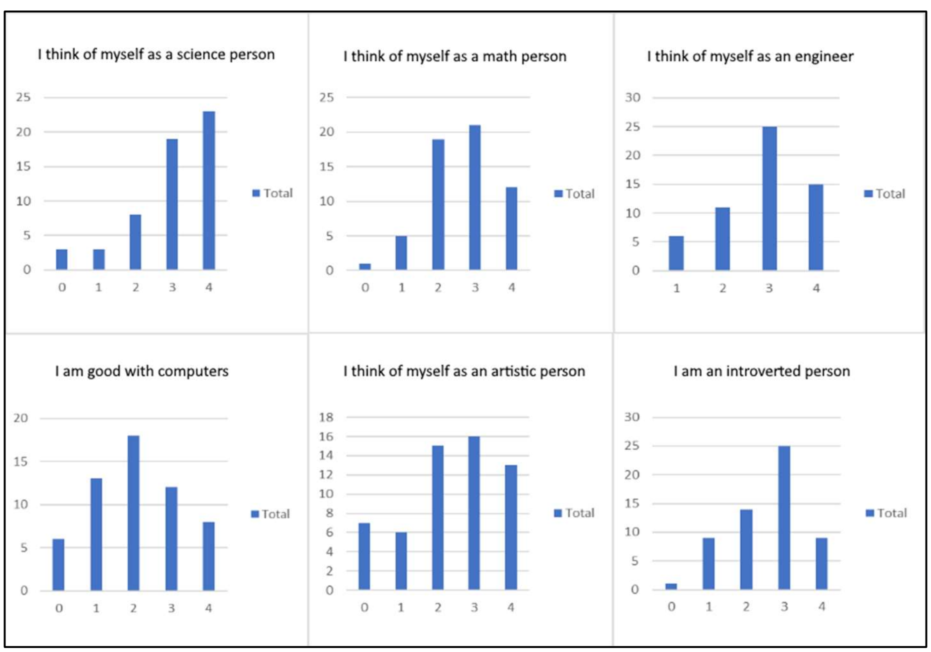


Figure 3.4: Experienced section responses to the Personal Identity Questionnaire.



Learning Styles Questionnaire Results

Figure 3.5: Total responses to the Learning Styles Questionnaire.

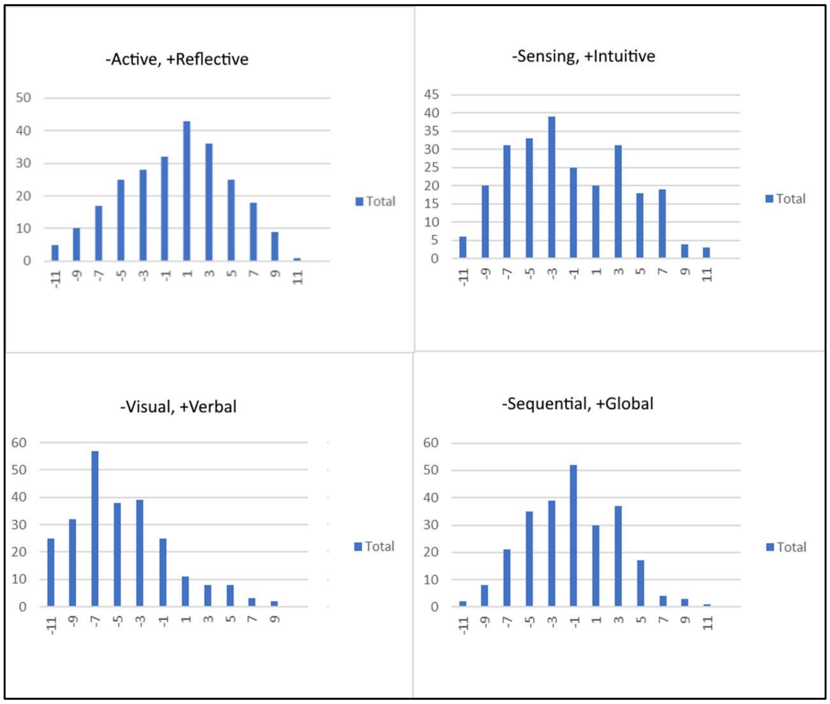


Figure 3.6: Delayed-Coding section responses to the Learning Styles Questionnaire.

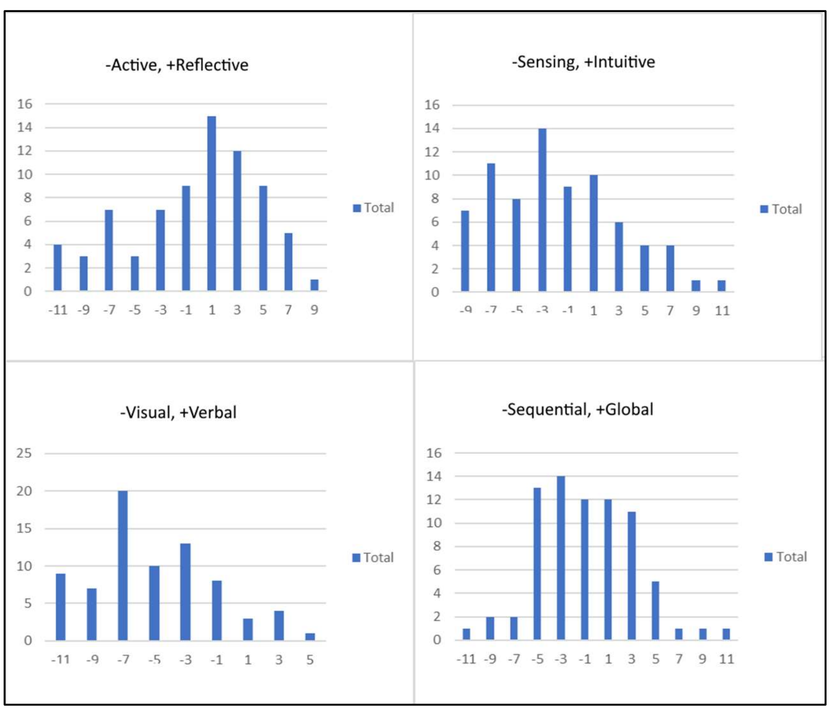


Figure 3.7: Intermediate section responses to the Learning Styles Questionnaire.

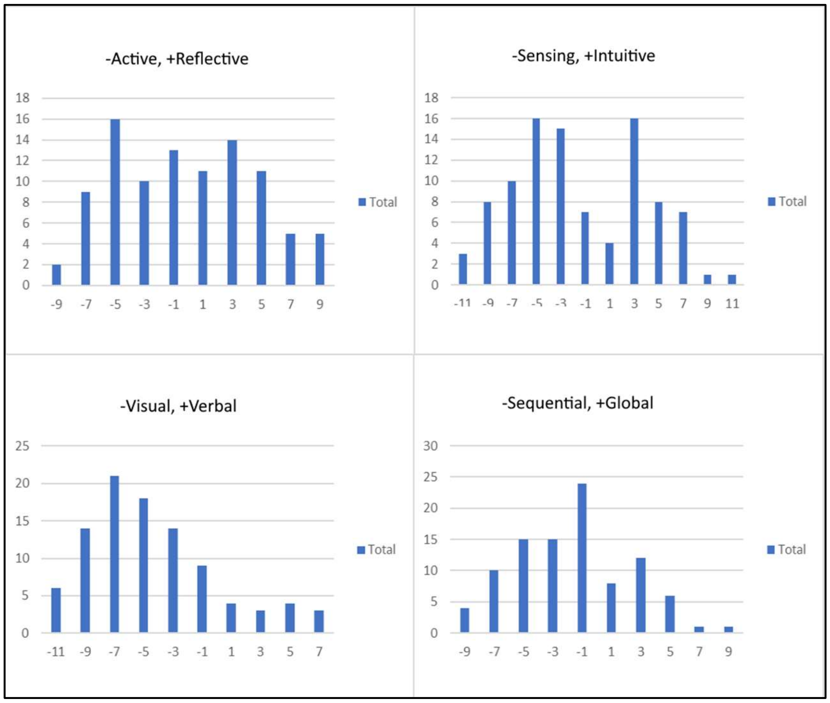


Figure 3.8: Experienced section responses to the Learning Styles Questionnaire.