Technical Report CS17-08

Computer Science As Seen By a Newcomer: Using Surveying & Visualization Tools to Glean Understanding

Claire Pickhardt

Submitted to the Faculty of
The Department of Computer Science

Project Director: Dr. Oliver Bonham-Carter
Second Reader: Dr. John Wenskovitch

Allegheny College
2017

I hereby recognize and pledge to fulfill my responsibilities as defined in the Honor Code, and to maintain the integrity of both myself and the college community as a whole.

Claire Pickhardt

Copyright © 2017
Claire Pickhardt
All rights reserved

CLAIRE PICKHARDT. Computer Science As Seen By a Newcomer: Using Surveying & Visualization Tools to Glean Understanding. (Under the direction of Dr. Oliver Bonham-Carter.)

ABSTRACT

We present a study which is designed to capture the general knowledge of computer science across a diverse group of respondents. Surveys are used to gather data which is then analyzed and displayed visually in order to be able to share findings with as many members of the community as possible. These results are designed to explain why computer science appears to be so under-represented in comparison to the populations of other disciplines across campus. This project culminates in an overall graph type evaluation garnered from the data collected and presents ideas about how to use what is evaluated to be the most useful or effective graph in data representation in a way to better market computer science as a subject. The data collected will also serve to help the department to improve its often misunderstood curriculum and also to help the department of Computer Science better extend itself to people and students who are not familiar with its discipline.

To my family.
My father, Kevin; my mother, Sarah; my sister, Katherine; & my cat, Koo.
Without you, I am nothing. Thank you for everything you do.

Acknowledgments

The process of completing this study has been a long one, but it was an entertaining ride throughout the entire process. Because the lack of students in computer science –especially the lack of female students– is troublesome to me, this project was quite near and dear to my heart. Starting out, I would never have been able to see myself ending up here, but with the help of my professors and the support of my friends and family, this document has come to completion. Many thanks to Dr. Oliver Bonham-Carter and Professor John Wenskovitch, both of whom helped me to reach my full potential and provided undying support and excitement for this project, which kept me going and never failed to lift my spirits. To my friends, thank you for listening to me talk excitedly about computer science and my results despite the fact that you had no idea what I was talking about for a majority of the time. To my family, both immediate and extended, who helped me to gauge what people outside of a college setting truly thought about computer science as a study. And finally, to the entirety of the Allegheny College computer science department, both students and faculty alike. Without such a strong, supportive, and entertaining group I can truly say that I would not have made it this far. Thank you for pushing me to stick with computer science and for all the laughs (and some tears) along the way. It was all worth it.

Contents

1 Introduction **1**

1.1 Motivation 2

1.2 Thesis 4

1.3 Thesis Outline 5

2 Related Work **6**

2.1 Primary Sources 6

2.2 Recent Results & Concessions 15

3 Method of Approach **17**

3.1 Threats to Validity 21

4 Experimental Results & Analysis **23**

4.1 Twitter Sentiment Analysis 24

4.2 Survey Results and Analysis 39

5 Discussion and Future Work **51**

5.1 Summary of Results 51

5.2 Future Work 53

5.3 Conclusion 54

A Python Code **56**

Bibliography **77**

Chapter 1

Introduction

As one of the smallest disciplines at Allegheny College, the computer science department is known for being somewhat isolated and rather intimidating for students and faculty who are not familiar with this discipline. Studies have shown that the number of college students choosing computer science as their major is steadily dropping, but the job market for computer scientists is steadily increasing. One hypothesis to explain this anomaly was developed by a study at Point Loma Nazarene University [1] that maintains that students do not possess a working knowledge or understanding of computer science and all the benefits of employment after schooling that a computer science degree would reward them with. Instead, this study concludes that students choose to focus on subjects that they may have studied in previous schooling, as they are already comfortable with the material. According to a 2006 study along with what has been seen by this researcher at Allegheny College, when students are properly introduced to computer science or decide to enroll in computer science courses, they are pleasantly surprised with the material, even to the point of changing their major or minor. Many students may have an aptitude for computer science but tend

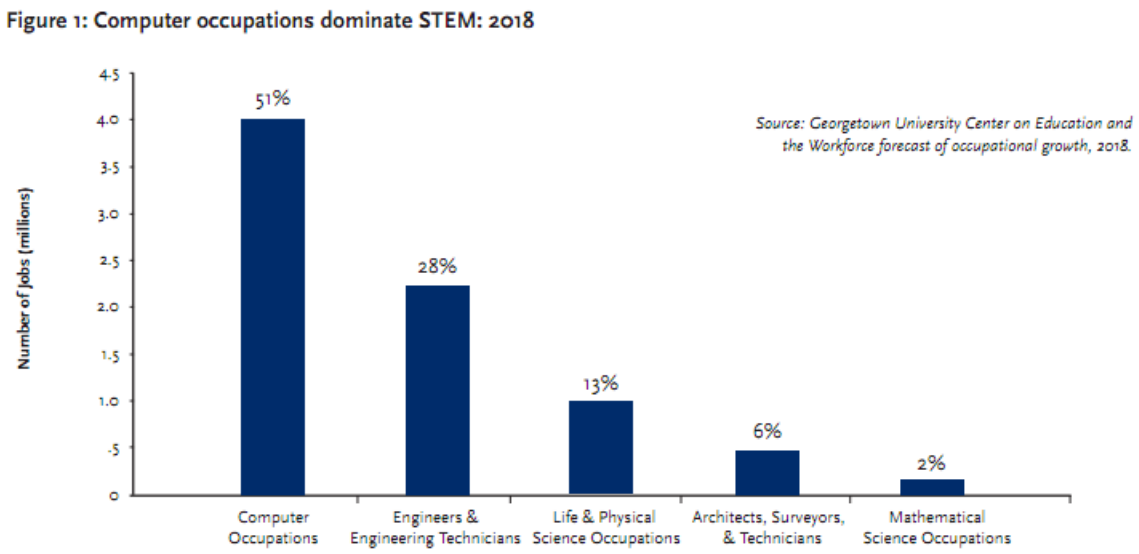


Figure 1.1: Percentage of careers in different areas of STEM

to avoid it because they are too intimidated to look into the subject itself [1]. In fact, because the number of careers in computer science is increasing, it is becoming increasingly unusual for students to avoid computer science courses which could provide their careers with so much opportunity. This can be observed in Figure 1.1, which projects the number of jobs in the industry in the year 2018 with data taken from the Georgetown Center on Education in 2012.

1.1 Motivation

One would think that these predicted career odds would cause more students to want to go into the computer science field. It can be argued that it is both equally difficult to teach and to learn computer science [20], but in a doctoral research study conducted in 2015 that focused on computer science education, it was proven that students who

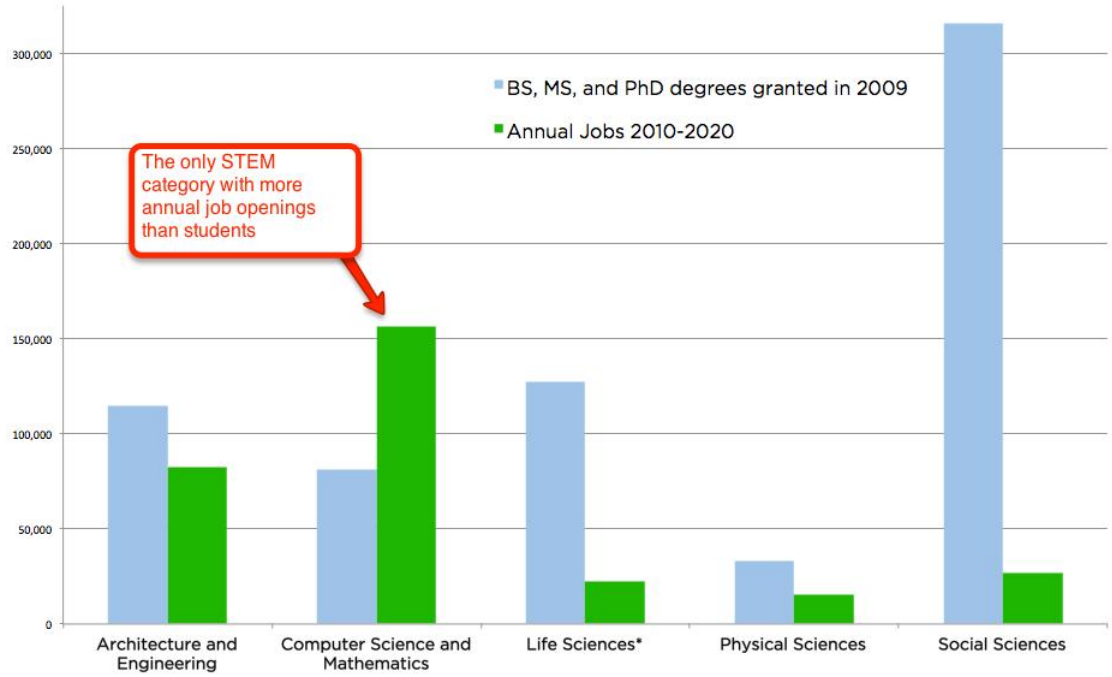


Figure 1.2: Difference in number of careers and degrees received

took even one class in the discipline no longer felt uneasy or intimidated by computer science [20]. However, one study completed by Hadi Partovi, CEO of `Code.org`, has shown that the number of computer science careers is increasing at a steady rate but the number of students receiving degrees in the field is not increasing at the same rate, leading to more open positions than students who could possibly take them [16]. A representation of this disparity can be seen in Figure 1.2. This study will seek to find why this imbalance exists, which may be partially explained by what people truly think of Computer Science as a subject.

Within the mathematical side of science, females comprise barely a quarter of the population. There exists "a marked deficit of women ... in certain disciplines ... [including] computer science" [2]. In this study, this project's survey can be expanded to be able to ask any number of questions in any field, even in regions

or schools where computer science is popular or in areas in which females are more prevalent than males. This can be directly contrasted to Allegheny College, where computer science is a very small major and the majority of the students are male [7].

Although it was originally conceptualized to help both students and faculty alike gain a better understanding of why the computer science department may have low recruitment and retention numbers, this study could also lead to interdepartmental exploration based upon how the data and visualizations have turned out. The graph evaluation is put to further use in the sense that several graphs will be utilized and analyzed for their efficiency.

1.2 Thesis

This study poses a question and seeks its answer. *Why are recruitment and retention abnormally low in the field of computer science, considering there is such a job market for the field?* The number of students entering the field of computer science is much lower than the number of jobs that need to be filled by individuals with computer science degrees. The number of students even showing any interest at all in computer science is dwindling, and the number of women graduating with computer science degrees is dismal at best. How, then, do researchers make claims as to why this is and what basis do they have to judge against? This study seeks to find why these problems exist and aims at ways to remedy these problems.

1.3 Thesis Outline

Outlined in the next chapter are multiple published papers documenting studies that are related to the work that has been done in this study. Each source is cited and more information about each source can be found in the thesis chapter at the end of this document. The survey results can be found in the chapter following, as well as documentation of the visualizations created. Upon reaching the completion of this document, the reader will find a summary of results as well as a written record of possible future works, should an individual or group choose to continue this research.

Chapter 2

Related Work

Despite the fact that computer science is a relatively new and quickly growing field, there already exist multitudes of publications in most, if not all, areas pertaining to it. With this being said, there exist several publications that describe studies that have been completed that are similar to this study, or may have some aspects similar to those in this study. Several of them have been showcased below, with a section for recent results and a concession statement about all of the help that is being done by an organization that is attempting to better educate the masses on the programming aspect of computer science.

2.1 Primary Sources

A study conducted by Brenda Wilson in 2002 [25] aimed to look at why the number of women taking computer science classes "from high school to graduate school diminishes at an alarming rate." The study was designed to see if people were not in computer science classes because they did not have, or felt that they did not have,

the aptitude to complete the study track. It also looked for any differences between males and females, and found that although there was a disparity in the number of females and males that studied computer science, there was no real difference in intellectual or academic aptitude and ability between the two genders. According to Wilson's study, the lack of females in computer science has been attributed to "lack of role models and encouragement, gender stereotyping, and lack of self esteem". The concepts of both recruitment and retention are addressed, along with previous computing experience. It was found in the study that a larger male population came into college as freshmen having more computing background than their female counterparts. It was also found that psychologically, women like to interact socially while learning and find comfort in collaboration; however, this is not normally encouraged in academic computer science, causing the females in the field to find it to be a hostile work environment.

The 2002 study mentioned above also talks about attribution. It is worth looking into both attribution as well as the impostor syndrome, experienced widely by gifted females [17]. It was discussed in Wilson's study [25] that "females tend to attribute their successes in computer science to luck and their failures to lack of ability". There is a perceived male dominance in mathematics and science, and males tend to attribute their failures to poor teaching or poor timing, while females feel failures personally. This is the *impostor syndrome* [17], which is defined within individuals as "feelings of fraudulence and a lack of confidence in their abilities" [3]. Although anyone can suffer from impostor syndrome, it is most commonly seen in women and is

defined as the lack of self-esteem with regards to academia and profession. There was even a leadership/management workshop that was restricted to only women so that they could develop themselves as leaders without the pressure to compare themselves to their respective male counterparts [24].

The gender gap in science is so important to some, in fact, that there is a book published in 2013, called *Women, Science, and Technology: A Reader in Feminist Science Studies* [26]. The book opens with an overview fitting of this study, and it is as follows:

> The scholarship represented here begins with familiar feminist themes related to social biases that discourage the participation and advancement of women in science, technology, engineering, and mathematics (STEM) [26]

It goes on to say that women are continually marginalized within areas of technological advancement and rest of the book continues in a similar fashion. A 2014 article written for NPR, titled "When Women Stopped Coding", speaks to the fact that women are not portrayed as scientific in media. A graph of the data collected by NPR pertaining to the number of women in differing science fields can be seen in Figure 2.1. The article can be found at `http://www.npr.org/sections/money/2014/10/21/357629765/when-women-stopped-coding` and goes on further to state that the majority of tech savvy individuals portrayed by Hollywood are seemingly nerdy teenage boys who use their knowledge of technology to win over girls who they find to be attractive. Because of this, it was believed that computers were only meant

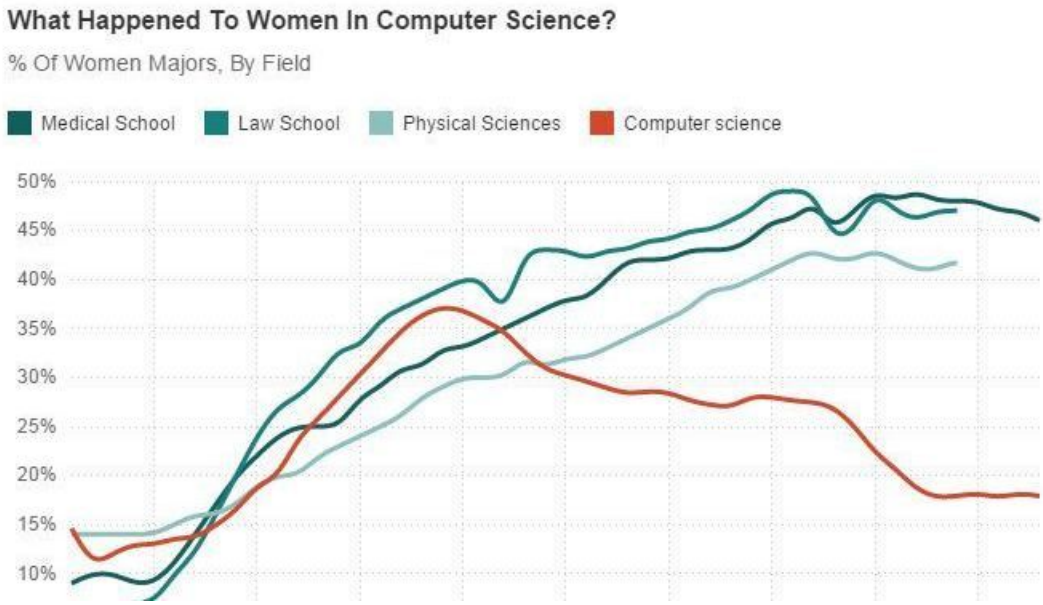


Figure 2.1: Women in science; computer science decreases heavily as time increases

to be studied by males and this belief has been perpetuated through the years. The way in which women are both perceived and portrayed in the media may also play a hand in the low recruitment and retention rates of women in computer science. In the 2016 book *Rise of the Rocket Girls: The Women Who Propelled Us, from Missiles to the Moon to Mars* by Natalia Holt, the stories of women who worked in NASA in the 1940s are brought to life. Most of these women held graduate degrees in math and science, and acted as "human computers, and later, programmers" for NASA's Jet Propulsion Laboratory [12]. Despite their successes and the book's data analysis as well as its publication, women are still running from computer science as a study.

Jane Margolis has published several books and papers about why women are underrepresented in the field of computer science. Her book "Unlocking The Clubhouse: The Carnegie Mellon Experience" details her work with students in the computer science department at Carnegie Mellon University [14]. Margolis found that when per-

sonal computers became popular household items, parents were more likely to buy their sons computers rather than their daughters, even if their daughters had shown interest in computers. Because the fact that girls were not receiving computers was not public knowledge at the time of the study, professors at Carnegie Mellon believed that if students were taking an interest in studying computer science, that meant that those students had an adequate background in the field. However, as Margolis noted, this was not the case and this caused girls wanting to study computer science to be intimidated and feel isolated from the rest of their male peers, who were more technologically savvy, and the girls felt as though they could not succeed [14]. Many things contribute to the lack of students in computer science, and one may be the type of visualizations used to portray data regarding computer science itself.

Many methods exist for generating visualizations. In fact, there even exist some software and systems that provide a graphical user interface (GUI) that allows users to input their data and receive a visualization automatically [22]. A biological and graphical study was conducted in 2016, and although it does not relate to the discipline of computer science, the aim of the study was to look at different visualizations of data and determine the most efficient one, as this proposed study would aim to do [4]. One of the visualizations generated in this 2016 study can be seen in Figure 2.2. There is an entire book, *Data Visualization With Python & JavaScript*, published in 2016 to help beginners in visualization generation create accurate and sophisticated representations of their data [5].

According to the *Journal for Research in Mathematics Education*, *document lit-*

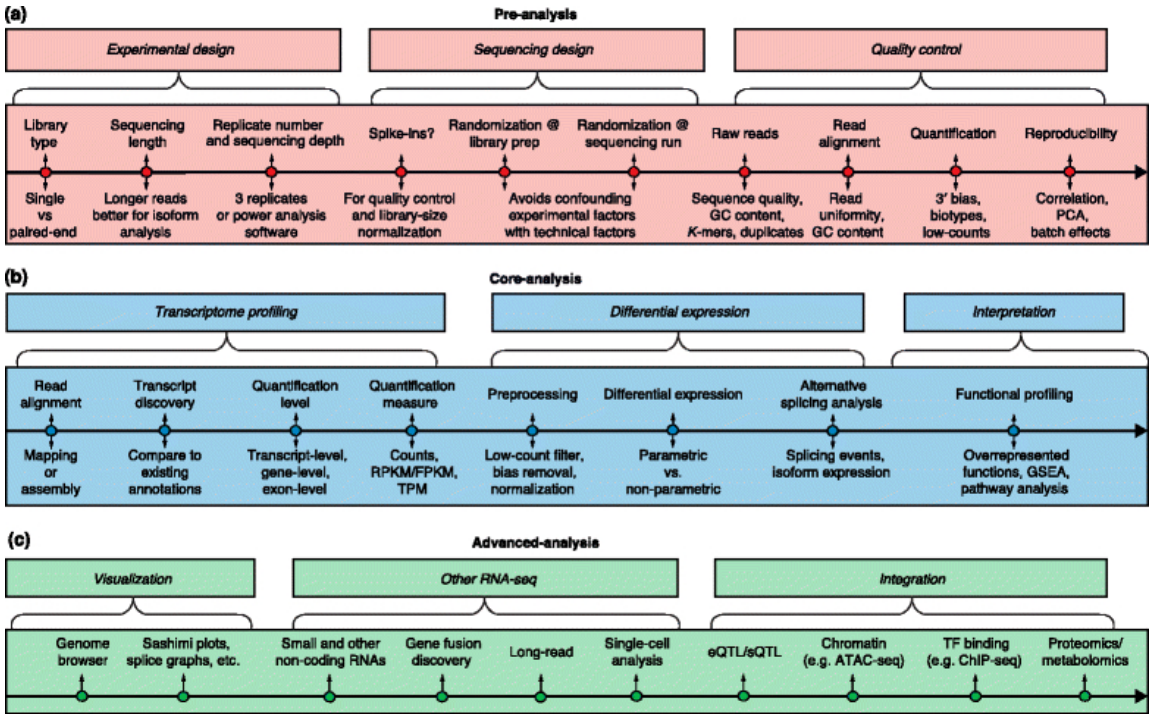


Figure 2.2: Image used to represent RNA sequencing data from Conesa's study

eracy is necessary in the reading of graphs. Document literacy is defined as "the knowledge and skills required to locate and use information contained in various formats, including [...] tables, and graphics" [8]. This has everything to do with how the information in the graph is understood by the viewer. Graphs and overlays are supposed to make information easier to read, but sometimes they work against comprehension. This project creates several graphs of the same data and presents them to Allegheny College and analyzes which graphs are most effective in transmitting information.

Although graphs and visualizations are often used in educational materials such as software and textbooks, not every graph is useful in helping students learn. This is proven by a study on comprehension completed by Priti Shah and James Hoeffner in

2002 [21]. The reading of graphs depends entirely on the viewers ability to comprehend visuals and map those visual features to meanings depending on their personal experiences. This is especially good for this study because students at Allegheny College have similar knowledge backgrounds because of their admission into the school. The interesting portion of this study will lie in the ability or inability to produce graphs effectively and/or answer questions about the subject of computer science based upon different major and minor track choices.

There exist several different design principles that should be taken into account when graphs are being created. The format should be chosen depending on what needs to be communicated. The next design principle will be utilized in this study, and that is to use different visual representations to display the same data. Working memory of the viewer should be minimized as much as possible. Color should be chosen carefully. Lastly, graphs and texts within those graphs should be kept relatively consistent to provide cohesiveness [21]. Spatial cognition in graph recognition and interpretation is vital. In a 2006 graph comprehension study conducted by Susan Trickett and J. Gregory Trafton [23], a group of meteorologists received multiple weather charts and were asked which one they would use to best convey the information presented to the general public. In some of the graphs, data was purposely omitted so that the data had to be inferred, simulating graphs that do not contain enough information and proving that each meteorologist had to use their knowledge of weather to make claims.

In a study by Kirsten Risden [19], three different web interface designs were com-

pared competitively against each other to test user comprehension, and also to figure out if one is better than the other in terms of usability. Two regular two-dimensional browsers were contrasted by a newer three-dimensional browser interface to test the usefulness of the novel three-dimensional browser. The study used an algorithm to create a graph layout and, in some cases, introduced deliberate distortions to gauge whether users could detect them or not. The main evaluation in this study was to have them complete tasks having to do with directory management, because that is what the "target population of users cite as their primary task". Additionally, the directory management mimics adding content to a larger website. This study used "empirical evaluation across the three use interfaces" [19].

Overall, there have been multitudes of studies on graph comprehension and visualization understanding, but this study would create visualizations specific to Allegheny and its student/faculty population, which would then be able to be evaluated and expanded upon in years to come, should any future researchers decide to recreate this study. This could be shared with Allegheny faculty to increase their knowledge on what students find to be useful and easy to read so that professors could use those graphs to help their students to learn better; it would also aid in the computer science department's recruitment efforts. This may also be useful in the business world when presentations are being made to esteemed colleagues who need to better understand how the business is working or what needs to be done. They need to be presented with data visualizations that are easy to understand and work with.

Data visualizations are vital to this study, but simply cannot be generated without

any dataset. On the other hand, the dataset is important to this study but cannot be easily read, which is why these must both included in this project. Many of the techniques discussed above are utilized, such as visualization generation through Python. The results that are collected from this project could greatly help the computer science department at Allegheny College as well as countless others. It serves to identify and address the reputation of the department. Overall, it is extremely useful to glean this data so that the researcher can fully understand why computer science study retention is so low and will work to identify ways in which this can be fixed.

Although data will be collected and analyzed, it is also extremely helpful to have a general consensus on how the world views computer science and aspects of the computer science field. As a way of generating something to compare against, Twitter sentiment analysis is utilized. "Sentiment analysis is a growing area of Natural Language Processing with research ranging from document level classification [...] to learning the polarity of words and phrases" [9]. In fact, in a study conducted by Efthymios Kouloumpis in 2011, there were several different datasets used and analyzed for their general sentiment and these datasets were categorized within the regulatory structure established by Twitter, as outlined later in this paper, specifically in Section 4.

2.2 Recent Results & Concessions

Although it would seem from the primary sources above that computer science is completely foreign to most people who are not involved in the field, there exists a group that is trying to change this. The Hour of Code project works to make introductory programming tutorials available and entertaining to the masses, and hopes to change the minds of so many who are intimidated by computer science. The project hopes to start children learning how to code at a very young age so that when they grow up, they will be familiar with how to code and will be able to enter the workforce and go into industry. The organization is developing multiple coding tutorials that take the form of games and play on aspects of popular culture, making it more relatable for the average person. An image of a tutorial from this organization may be found in Figure 2.3. A study was completed to test the effectiveness of the Hour of Code on general individuals, regardless of their background in computer science. The study tested retention of information learned during the tutorials, and the end result was that despite the effort that goes in to making the tutorials both entertaining and engaging, the tutorials do not actually help people retain the information that they learn when they are learning how to code [18].

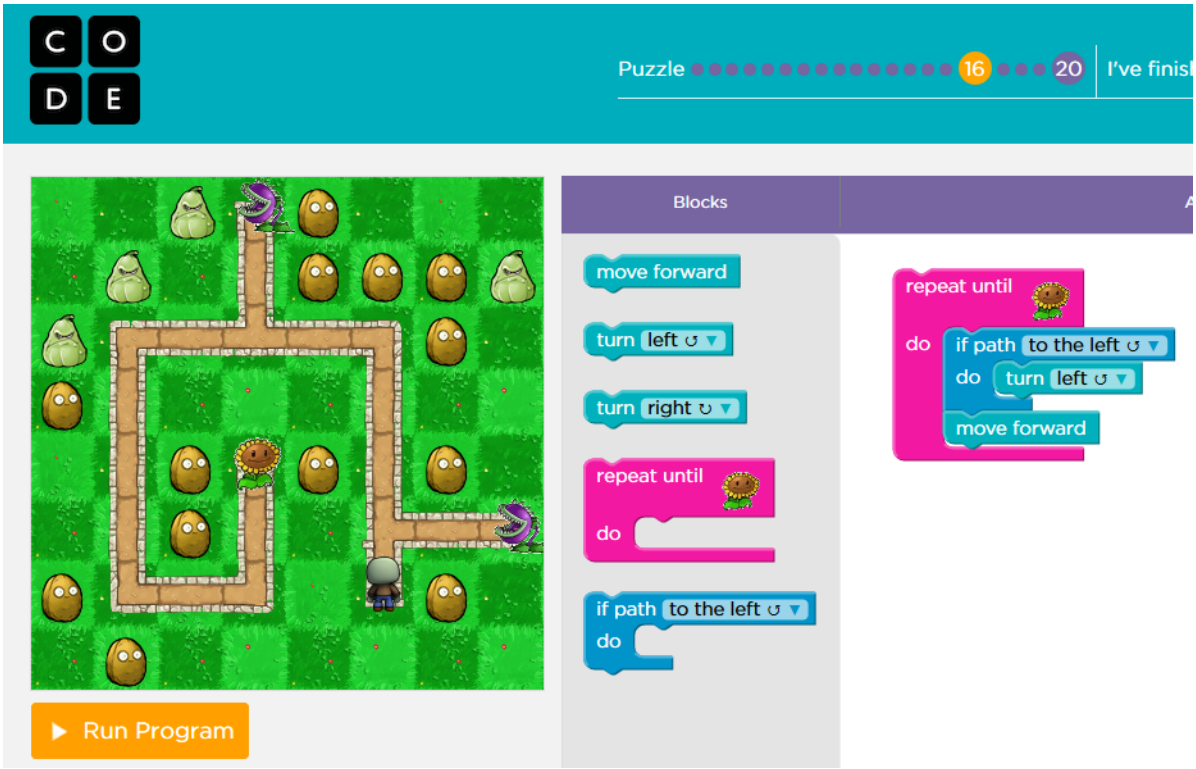


Figure 2.3: The Hour of Code organization works to give the general population a user interface to help them to learn programming

Chapter 3

Method of Approach

The way in which answers to the questions outlined in the introductory section of this paper are answered is through data collection and analysis, and then extrapolation from this data analysis. Each of the datasets collected will work to answer the questions posed in different ways.

The first results analyzed will be that of the Twitter Sentiment Analysis. Tweets are scraped from the internet and analyzed using a pre-existing sentiment analyzing tool. This tool is discussed in detail in Section 4.1 of this paper. The tweets that are gathered change every day and are gathered by the user when the user types in a query or phrase into a search bar. There may be several words in the phrase or just a single one, but every change to the objects queried in the search bar changes the results of the sentiment analysis. These results will help this researcher to develop a better understanding of the general consensus of computer science and its offshoots, and give a basis of comparison for the next dataset analyzed within the project.

Secondly, a survey was sent out with several questions pertaining to computer science as a topic of study. The survey was conducted through the Google Forms tool,

which allows for easy creation, manipulation, and sharing [13]. The Google Forms tool also displays a summary of the results to each question in different visualizations, depending on the type of input the question calls for. The questions and their answering methods are as follows:

1. What is your gender? (Male/Female/Other)
2. Are you a college graduate? (Yes/No)
3. What is/was your major? (Input)
4. Have you ever taken a programming lesson before? (Yes/No/Maybe)
5. When you hear the words 'Computer Science', how do you feel? (Checkboxes)
    - Happy
    - Scared
    - Intrigued
    - Unintelligent
    - Sad
    - Excited
    - Intelligent
    - Indifferent
    - Confused
    - Wonderful

6. What types of careers do you feel come from a knowledge of computer science? (Input)

7. Do you think computer science as a study is saturated with students? (Yes/No/Not Sure)

8. Have you ever participated or would you ever consider participating in exercises to promote learning code (i.e. hour of code)? (Yes/No/Maybe)

9. Would you ever want to go back to school to study computer science? (Yes/No/-Maybe/Not Applicable)

10. What do you think employees do in computer science/programming positions? (Checkboxes)

    - Sit at a desk and code all day
    - Test the work of others
    - Make art
    - Work in collaborative groups
    - Play video games all day
    - Travel for work

This survey and its results from a sample of people surveyed will help to answer several questions, one of which is *Why are, or are, people generally intimidated by computer science as a study?* To refrain from looking at just numbers and data,

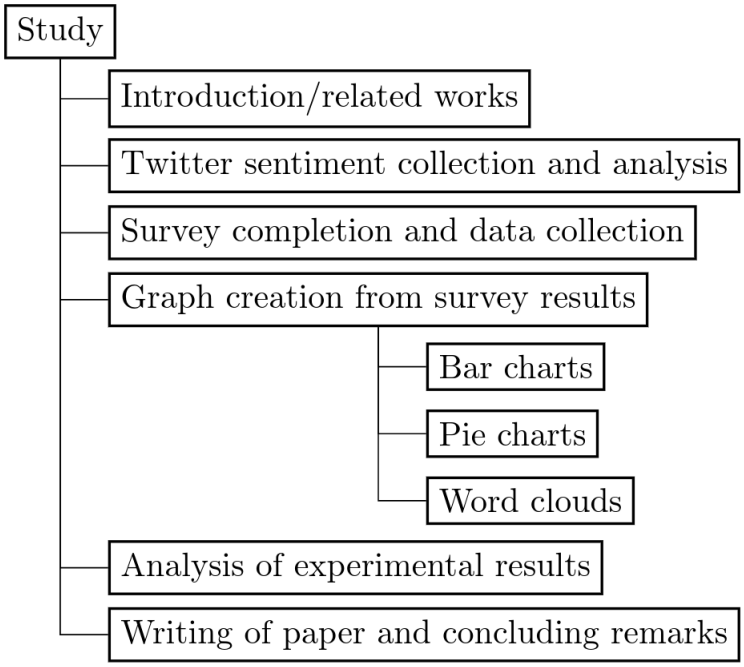


Figure 3.1: Workflow

visualizations were created from this data. The analysis can be found in Section 4 of this paper. All of the data from this survey was collected in a CSV file, and there were over fifty result sets collected from those who took the survey. Because there exists a file with the results saved and stored in it, it makes it easier to go back and revisit the data in order to create visual representations from it. In Figure 3.1 you can see an outline of the workflow completed in this research project.

For each question, two graphs have been created showcasing the results and are placed side by side, showing that there are multiple ways to show the same result. For the most part, the graphs consist of pie charts and bar graphs, all created in Python. However, there were several word clouds generated online with the results from a handful of the questions. The website `https://tagul.com/create` was used

for these creations. This method, of using a preexisting website to create word clouds online, was ultimately more useful than creating them with any language because using the online tool made it possible to create them to be much more cohesive with the other graphs created. This decision, to make the visualizations all work cohesively together, ultimately will make it easier to compare the graphs against each other to figure out which graph is more useful than another. The graphical methods chosen to represent the data collected were chosen because they are simple and show the data quite differently. There are infinitely many ways to visually represent data, so the limit for this project was drawn at two visualizations for each question out of three graphical types being used.

3.1 Threats to Validity

The threats to the validity of this project consist of any outside sources that cannot be controlled by this researcher that may cause the results of this study to not be as strong. One of the potential threats lies in the fact that the participants in the survey are taking this survey on the internet and are able to answer questions however they like, honest or not (however, this researcher hopes that they were indeed being honest). Allegheny College is also a small liberal arts college, and not a technical college, so in this there may potentially be the reason why there is a small department of computer science present here. There also may have been unequal representation of participants who took the survey, however because it was an anonymous survey,

this could not be controlled.

Chapter 4

Experimental Results & Analysis

In this chapter, the reader will find a detailed description and analysis of the results from the aforementioned Twitter sentiment experiment as well as the survey. The Twitter sentiment analysis results are taken from seven consecutive days of monitoring so as to get a strong sense as to how the general population continuously feels. The visualizations created from the survey results are laid out in figures and documented, with the analysis of these questions residing in the captions of the figures.

In order to establish the validity of the results collected from the sample pool and analyzed, it is pertinent that the feelings of the general population are collected and analyzed to be used as a baseline. In order to find this general consensus on how a large group feels about computer science in reference to certain sectors of the field, sentiment analysis of tweets from the popular social media site Twitter was completed over a period of several days. Twitter, since its creation in March of 2006, has risen to sit among the ranks of other social media powerhouses such as Facebook, Tumblr, and Instagram [10]. Despite the fact that it limits users' posts to a certain number of characters, it has become a platform for others to share their thoughts and is thriving

as this report is written. Because of its popularity and its structure –it forces users to post multiple times in order to get a certain point across, thus users post several times per day– it is a media platform that changes constantly and is worth analyzing.

4.1 Twitter Sentiment Analysis

Tweets are small blurbs that are written by individuals to be shared with their online followers, and must contain 140 characters or less [10]. There is also the concept of the *hashtag* phrase in which the user types a pound sign and the follows up with a phrase that can be connected with other tweets that also contain the same hashtagged phrase. A hashtag may be searched, and every tweet containing that hashtag will appear and be both clickable and visible to the user. If a phrase contained in the hashtag contains a space character, it is no longer in the hashtag. A hashtagged phrase ends when a space is entered, so when two words exist within a hashtag, they must be joined without a space character. These hashtags trend and the top trending hashtags can be seen from each user's account [10].

There are several tools that are available online for the average person to utilize to obtain tweet sentiment. This researcher used a tool that was developed by North Carolina State University and comes accompanied by several visualization tools that show the user their results in several different ways. It can be found and accessed at `https://www.csc2.ncsu.edu/faculty/healey/tweet_viz/tweet_app/`. When utilizing this tool, the user is able to input certain phrases and a large mass of tweets

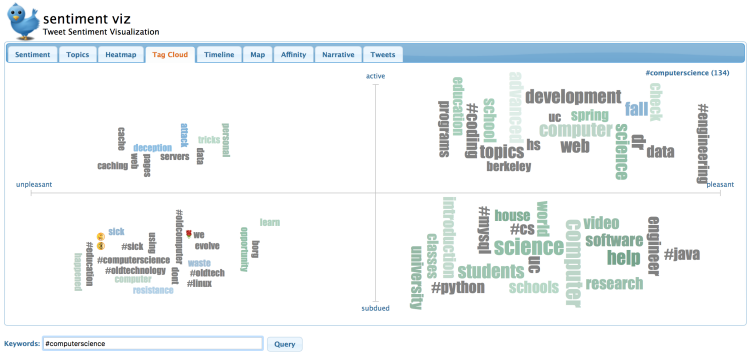
are gathered that contained those phrases and the sentiment of all of these tweets is shown in different visual representations, including but not limited to heatmaps, timelines, and word/tag clouds. When showing the output in the "Sentiment" category, there exists a large circle that is outlined by emotive adjectives. The results are plotted in this circle, with positive results on the right side of the middle line, and negative results resting on the left side of the middle line. The words on the right side include, but are not limited to: pleasant, contented, serene, elated, excited; whereas the words on the left include, but are not limited to: unpleasant, stressed, sad, nervous, tense. This middle line, dividing the positive and negative results, is also present in the "Tag Cloud" section. For the tag clouds produced, it is noted on the tool's website that "Common words from the emotional regions Upset, Happy, Relaxed, and Unhappy are shown. Words that are more frequent are larger." The tool was used to search the phrases "#ComputerScience", "#ComputerScience programming", and "#ComputerScience women" repeatedly over a period of several days in March 2017. The tweets change daily, so it was pertinent that several days' worth of content be recorded in order for a true general consensus to be recorded as baseline sentiment. Sentiment and tag clouds were recorded each day, and screen captures of each result from every day can be seen in the collection of figures below. There was a conscious effort made when choosing what phrases to query. "#ComputerScience" is a hashtagged phrase that forces the two words to be searched together, rather than searching for tweets that mention them both separately. This made certain that the tweets gathered as results were relevant to the aim of the project, and did not have

to do with *just computers* or *just science*, but rather computer science as a whole subject.

One thing that can be noted that happened consistently, and is to be expected, is the narrowing of result numbers as the queried phrases became more specific. When the blanket phrase "#ComputerScience" is queried, more results show up than when a word is added afterwards. Consistently, with the phrase "#ComputerScience" is searched, the number of related and sentiment-checked tweets sits well above one hundred. As "programming" is added after it, the number dwindles to around thirty results and when "women" as added after, the number hovers somewhere between ten and twenty results. In Figure 4.1 through Figure 4.21, one can see the sentiment expressed as well as tag clouds with the corresponding queries from each day over the seven day period.

#ComputerScience

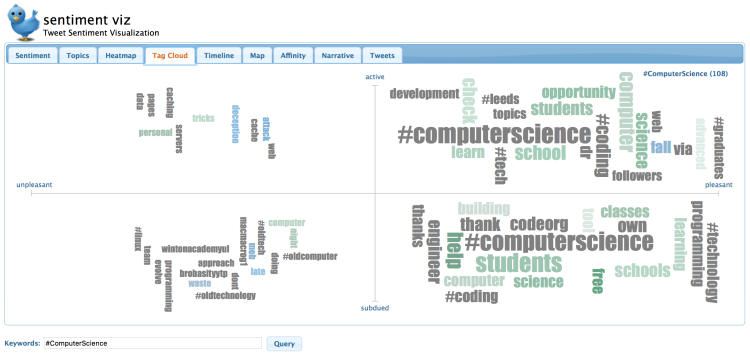
(a) Sentiment



(b) Tag cloud

Figure 4.1: Day 1

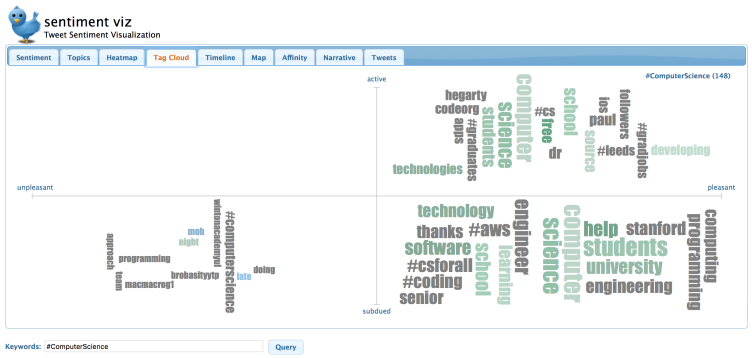
(a) Sentiment



(b) Tag cloud

Figure 4.2: Day 2

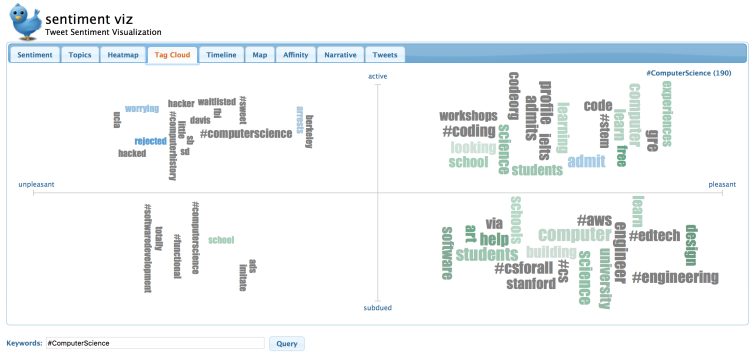
(a) Sentiment



(b) Tag cloud

Figure 4.3: Day 3

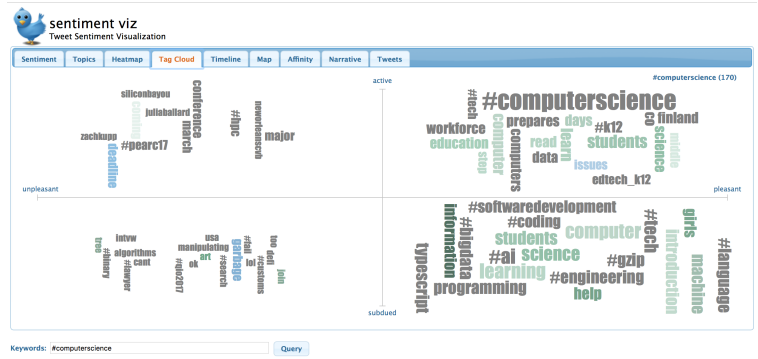
(a) Sentiment



(b) Tag cloud

Figure 4.4: Day 4

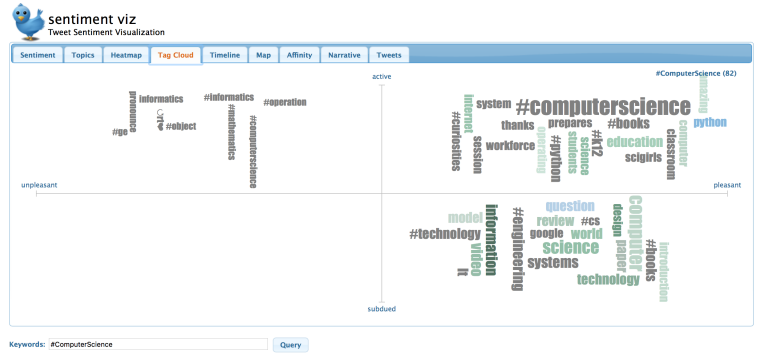
(a) Sentiment



(b) Tag cloud

Figure 4.5: Day 5

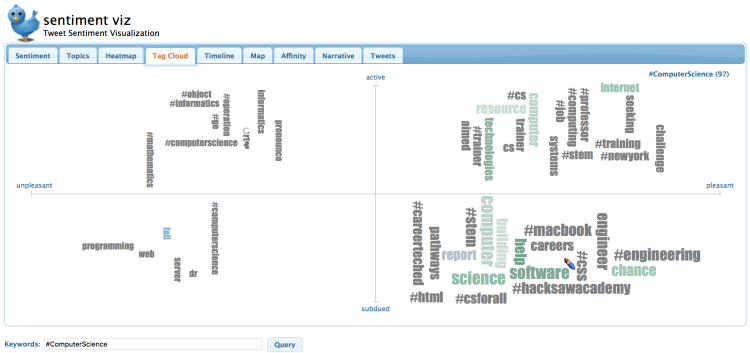
(a) Sentiment



(b) Tag cloud

Figure 4.6: Day 6

(a) Sentiment

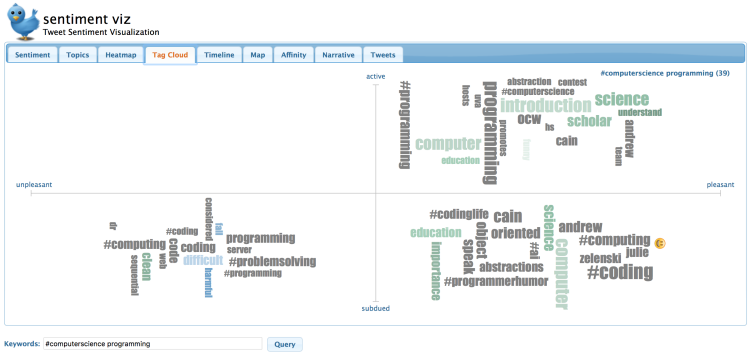


(b) Tag cloud

Figure 4.7: Day 7

#ComputerScience programming

(a) Sentiment



(b) Tag cloud

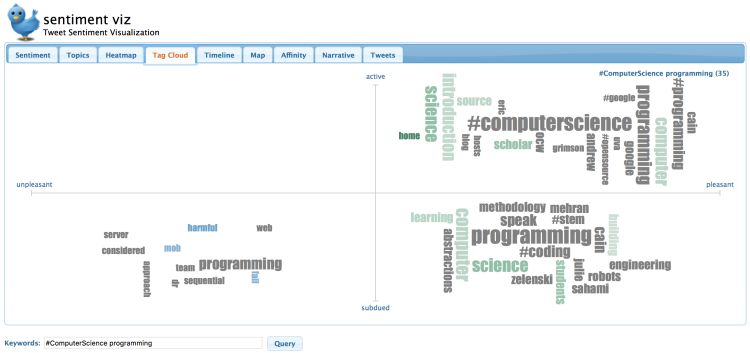
Figure 4.8: Day 1

(a) Sentiment

(b) Tag cloud

Figure 4.9: Day 2

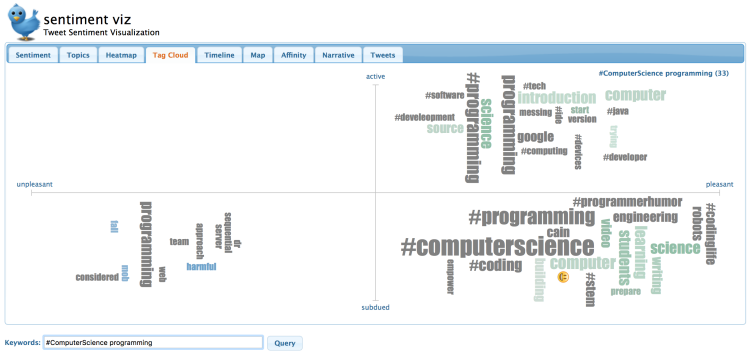
(a) Sentiment



(b) Tag cloud

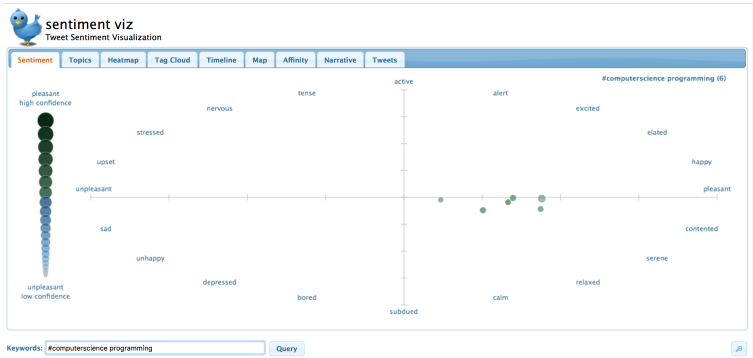
Figure 4.10: Day 3

(a) Sentiment

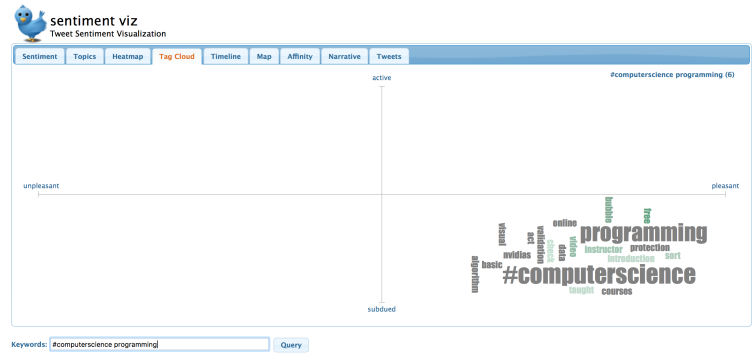


(b) Tag cloud

Figure 4.11: Day 4



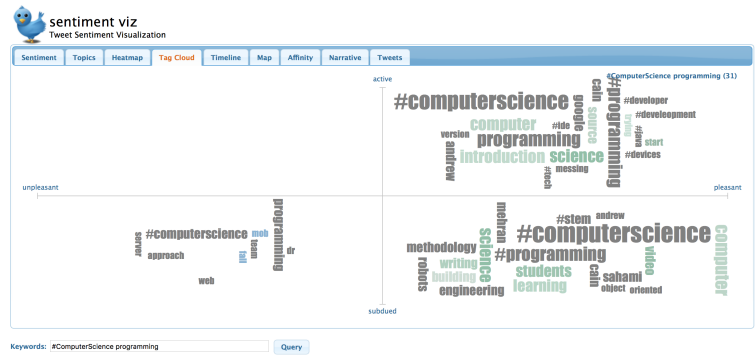
(a) Sentiment



(b) Tag cloud

Figure 4.12: Day 5

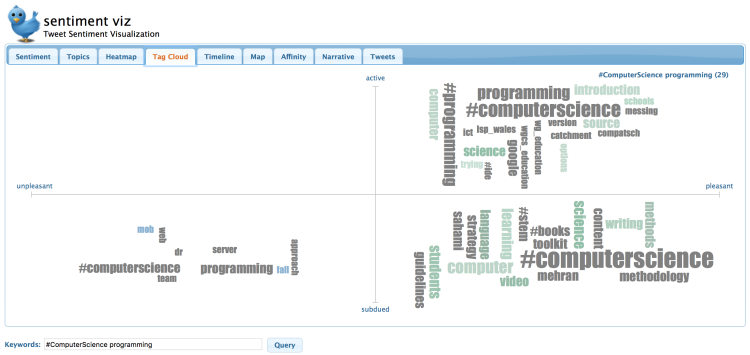
(a) Sentiment



(b) Tag cloud

Figure 4.13: Day 6

(a) Sentiment

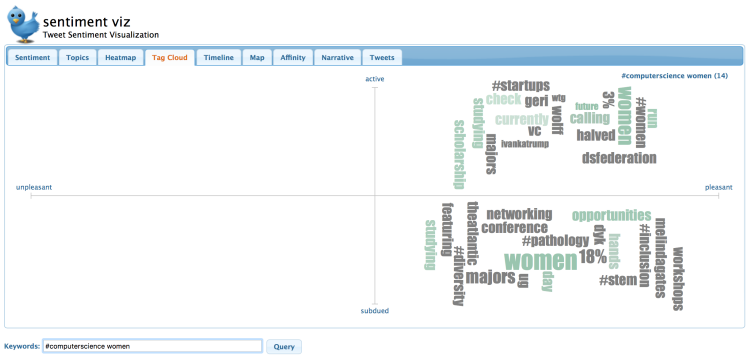


(b) Tag cloud

Figure 4.14: Day 7

#ComputerScience women

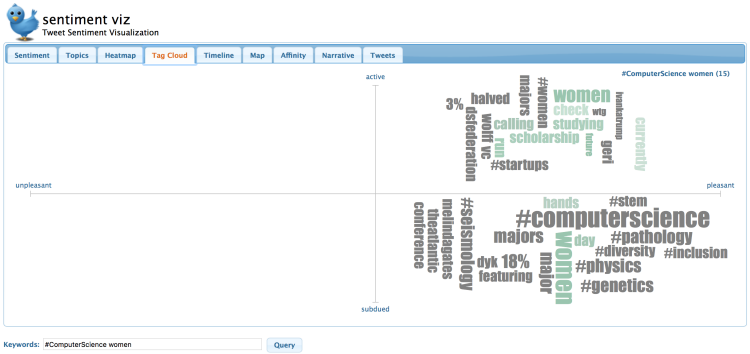
(a) Sentiment



(b) Tag cloud

Figure 4.15: Day 1

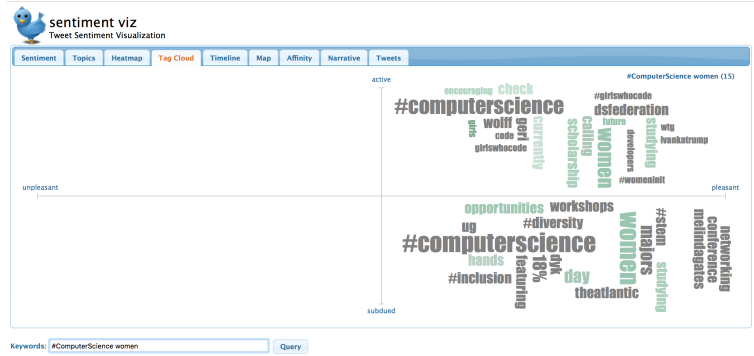
(a) Sentiment



(b) Tag cloud

Figure 4.16: Day 2

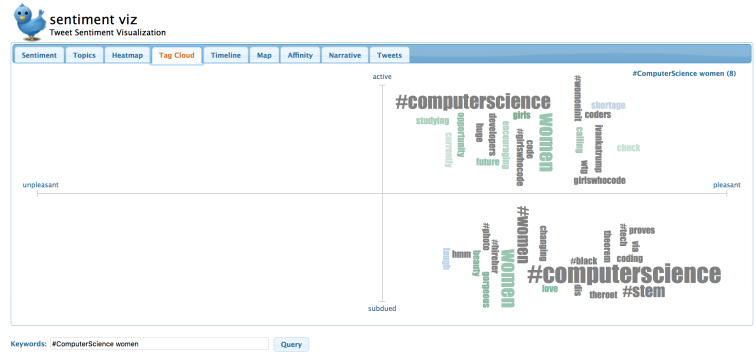
(a) Sentiment



(b) Tag cloud

Figure 4.17: Day 3

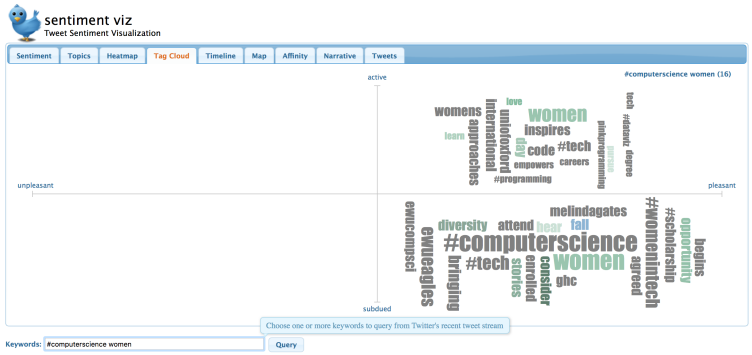
(a) Sentiment



(b) Tag cloud

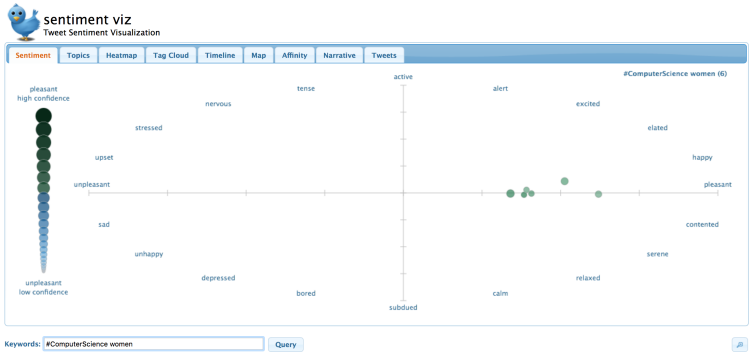
Figure 4.18: Day 4

(a) Sentiment



(b) Tag cloud

Figure 4.19: Day 5

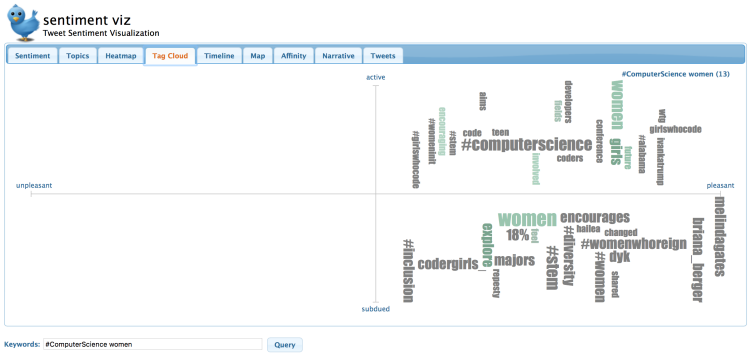


(a) Sentiment

(b) Tag cloud

Figure 4.20: Day 6

(a) Sentiment



(b) Tag cloud

Figure 4.21: Day 7

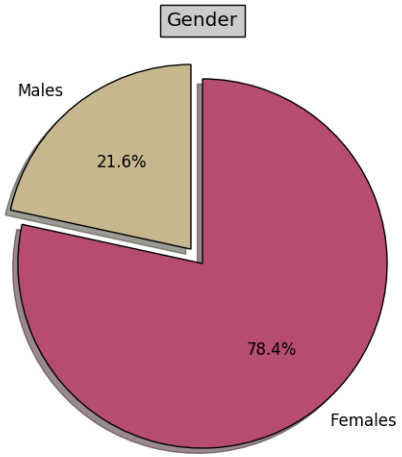
As one can see, the results do not change drastically over several days. The sentiment portion of the figures shown stays primarily to the right of the middle line in every test completed, showing that the general sentiment towards computer science is positive, at least on Twitter. It should also be noted that the number of dots in the sentiment portions, representing tweet results, dwindles drastically between the "#ComputerScience" results and the "#ComputerScience women". It should also be noted, that in today's age and in general social media tactics, people are not as apt to share their negative experiences over their positive ones for fear of being looked upon differently or becoming an annoyance to the individuals who "follow" them. There was also a study conducted in 2015 that looked into the sentiment of several languages, and revealed that there is a "universal positivity bias" in the words that we use [6], which is directly correlated with the results received in this project's Twitter sentiment analysis experiment. The positivity of these sentiments provides a good baseline to compare against when looking at the results and analysis of the survey written for this project.

4.2 Survey Results and Analysis

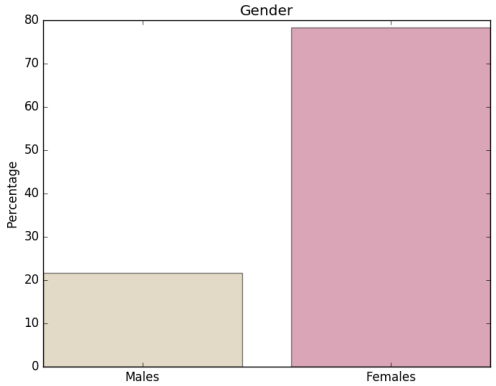
The questions on the Google Forms survey can be found in the Method of Approach section, but will also be reiterated in this section as well. This survey gleaned fifty-one data sets as results from the fifty-one individuals that completed it. One of the appeals of the Google Forms tool is the fact that it is easily sharable between

users. The distribution of this survey takes full advantage of that. After this survey was created, it was sent to several members of the Allegheny College community from all different areas of study, as well as some individuals outside of the Allegheny community. The survey was sent out in an email that asked each participant to share the survey with several of their friends or colleagues, and the survey was spread in this way. There were fifty-one participants. Each question is stated, and two graphs follow that show the results of that question, respectively, in two different ways. Underneath this presentation follows a description of the results as well as a comparison of the graphs and the verdict on their effectiveness in showing those results. As the reader will learn, there is no one right way to represent data, and it truly matters what data is being shown. For all of the questions, there was no one best way to represent the data, but the graphical analysis and the time and thought that was put in to figuring out the most effective way to show that data should be noted and carried out by any researcher presenting their data.

What is your gender?



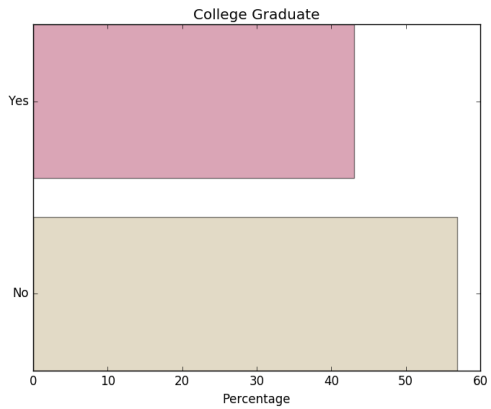
(a) Pie Chart



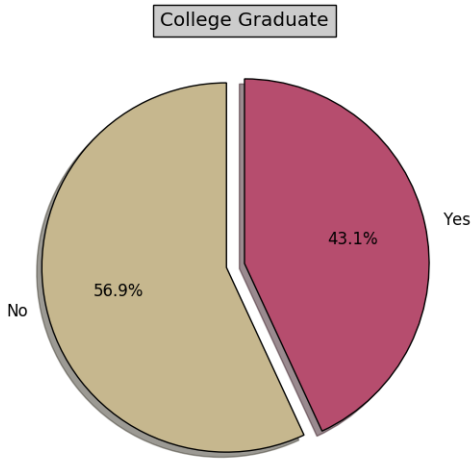
(b) Bar chart

Figure 4.22: The results above show the number of people in each gender that took the survey. This question was important to ask, as one of the major concerns facing the computer science industry today is the lack of females graduating with degrees in computer science. Although the option for "other" gender with an input section was provided, it was not utilized by any of the participants. In this instance, it is more useful to display the results in a pie chart. This is because the pie chart shows the percentage of each gender as part of a whole, with the whole representing the number of participants. The bar chart representing the results holds lesser value in this instance because although the results are represented well, they are represented in a way that compares them to each other. The number of participants that were female looms as a bar over the number of participants that were male, and the total overall number cannot be seen without the user having to calculate it themselves. However, in the pie chart seen here, the percentages of each gender are clearly displayed and they can be noted and compared against each other, as part of the whole participant population, causing the pie chart to be the better representation in regards to this particular question.

Are you a college graduate?



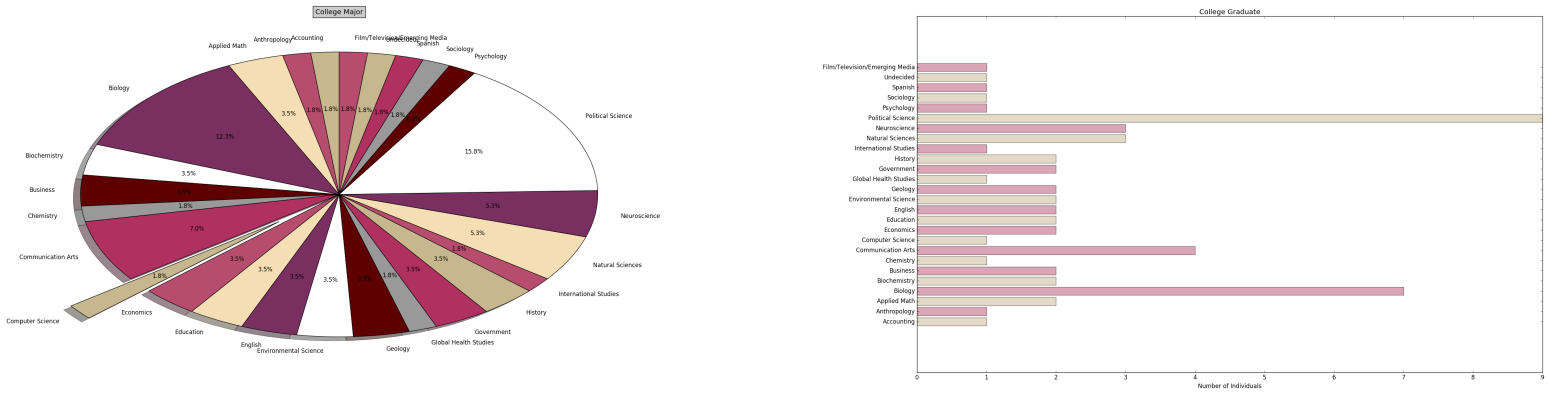
(a) Bar Chart



(b) Pie Chart

Figure 4.23: This question is pertinent to the results of the study as well. It is necessary to know if the participants being polled have a degree, so that the average age of the participants can be taken into account. The participants who have a degree may have a better working knowledge of the careers associated with computer science, as they are most likely in or have most likely taken part in industry. The portion of the participants who possess a degree may answer certain questions in the survey differently than those who do not possess a degree. As in the last question, a pie chart better represents this data, as it shows the two result pools as two part of a greater whole. It compares the percentage of a certain answer against all of the answers, which better represents the results. Comparing directly to the pie chart, the bar chart falls flat and once again, the two answers in bar form look as though they are competing. They sit there, on the bar chart, not giving the reader the full picture.

What is/was your major?

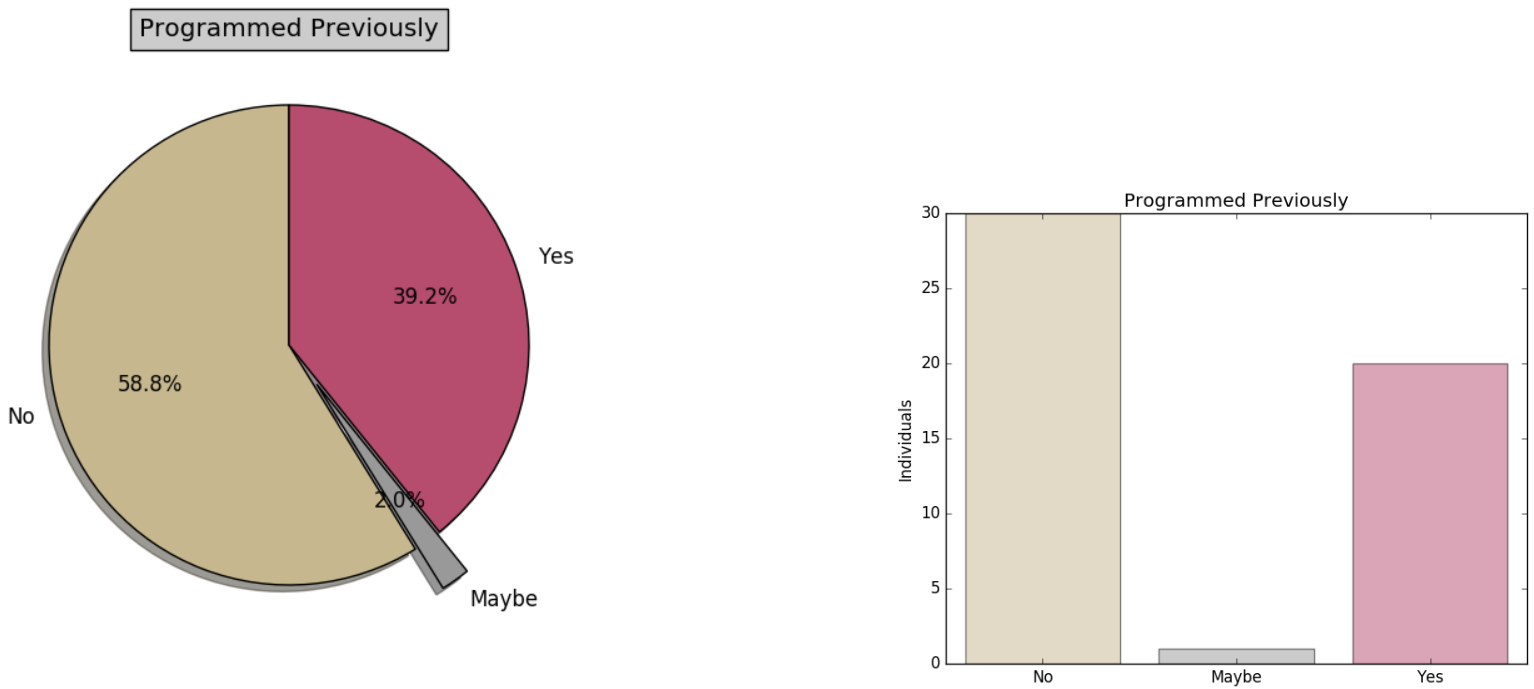


(a) Pie Chart

(b) Bar chart

Figure 4.24: A person's area of study says a lot about them personally, but even more so, it speaks to whether or not they have potentially taken computer classes or would have a working knowledge of technology. A person with a background in English studies is less likely to have a knowledge of computer science than a person who has a background in applied mathematics. This researcher took the liberty to explode the slice in the pie chart that represents the participant who is a computer science major so as to highlight the fact that the vast majority of individuals that completed this survey did not come from a background in computer science. In this instance, it is the bar chart that represents data in a more efficient way, over the pie chart. There are far too many results, or areas of major study, to be represented by the likes of this pie chart. Although the colors are cohesive and the percentages are aptly represented, the bar chart represents the number of individuals within each major area in a more understandable and readable manner.

Have you ever taken a programming lesson before?



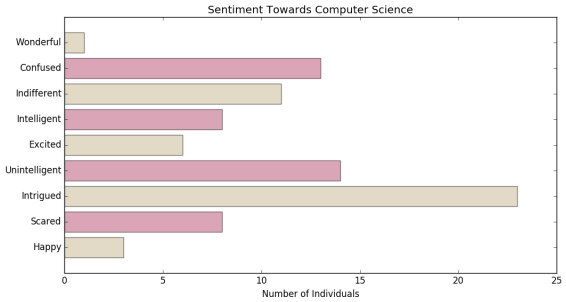
(a) Pie Chart

(b) Bar chart

Figure 4.25: From this question and its results, the readers learns that a majority of the participants in the study have never interacted with programming before. Almost two-thirds of the people who took this survey have never coded anything, and that almost two-thirds is easily visible in the pie chart. In the bar chart, it does not look like there is as large a disparity between the people who have taken a programming lesson before and those who have not. Seeing how the results stack up against each other in terms of all the data collected is useful in comparing them against each other, so the pie chart is the better option here.

When you hear the words "computer science", how do you feel?

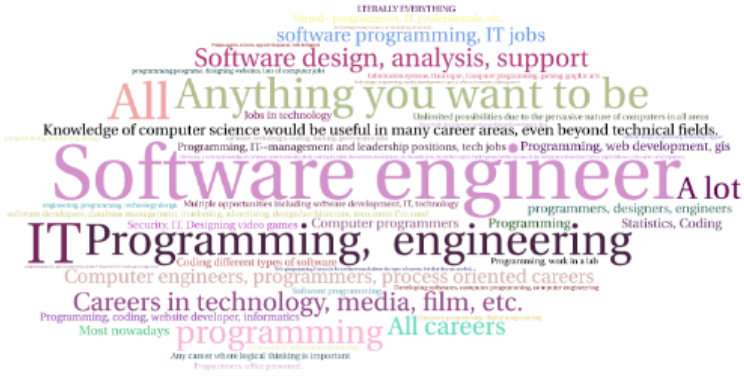
(a) Word Cloud



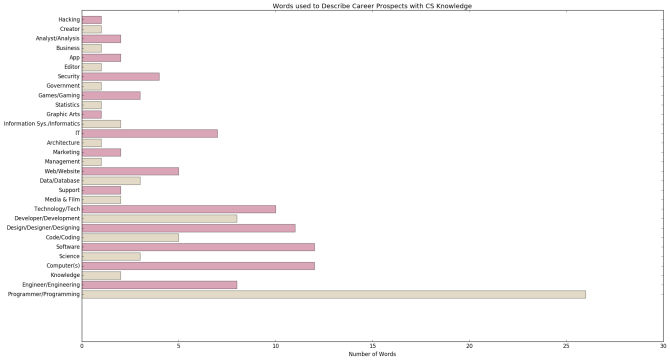
(b) Bar chart

Figure 4.26: This question was asked to try to gather what the participant population truly felt about computer science as a study. The question was multiple choice, with predetermined words as the answers. The answers were in checkbox form, which, in the Google Forms service, allows the user to choose more than a singular answer. This is why, though there were only fifty-one people surveyed, there are many more results accounted for than just fifty-one. In the word cloud visualization, the word "intrigued" is largest because that was the word to gain the most results. However, the next two words that gained the second- and third-most results are "confused" and "unintelligent". The fact that these negative words were high scoring is problematic, and this will be revisited later in this section. Both of these visualizations are incredibly useful in displaying the results of this question. The gauge for which is more useful depends solely on the time that the reader has to look at them or the specificity that the user desires. If the user only has a quick second to glance at both of these visualizations, the word cloud will alert them to all they need to know for a quick overview of the results. However, one thing that the word cloud does not do is show the actual number of results collected for each word. This is where the bar graph is useful, for the bar graph shows the exact number of responses for each word and shows all of the results in a neat and organized manner.

What types of careers do you feel come from a knowledge of computer science?



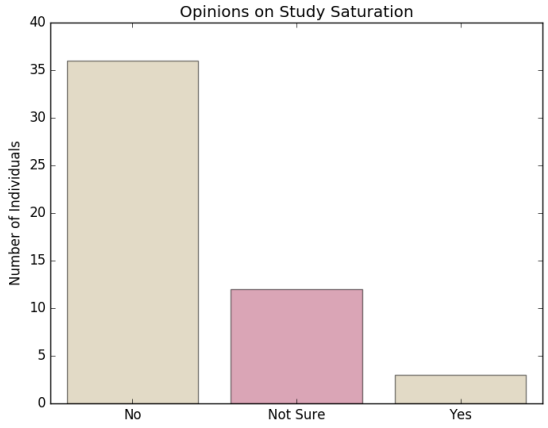
(a) Word cloud



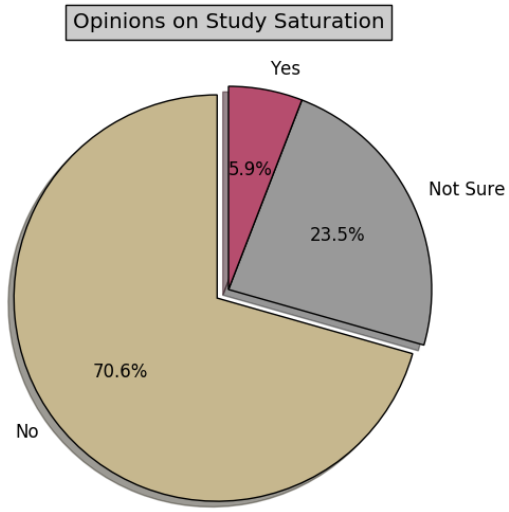
(b) Bar chart

Figure 4.27: The way in which participants were able to answer this question was through text-based input on their part. They were able to respond in words and phrases of their own choosing. Some answers were lengthy, and some only consisted of a word or two. Either way, the visualizations seen above are quite different in nature. The word cloud was created by taking the exact wording from the CSV file containing the results, whereas the bar chart was created in a much more meticulous manner. All of the filler words or words that did not apply to the question's answer were discarded, leaving only the counts of the words that did apply to the question's answer. Those numbers are represented as the bars in the bar chart. In this instance, although the word cloud is aesthetically appealing, the data can be better read from the bar chart. This is because the words become difficult to read in the word cloud, but the results are much more pared down before they are placed into the bar chart. All of the data in the bar chart is of the utmost importance because of this.

Do you think that computer science as a study is saturated with students?



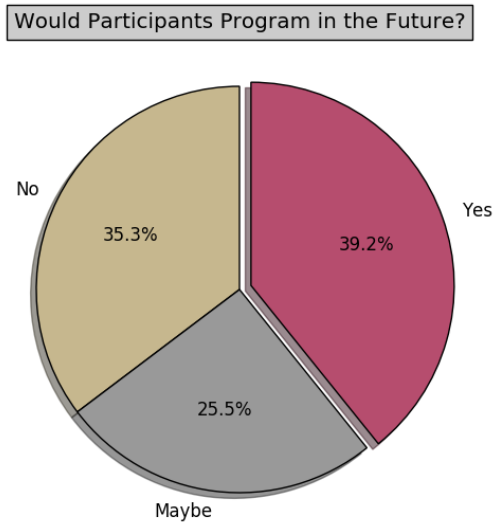
(a) Bar chart



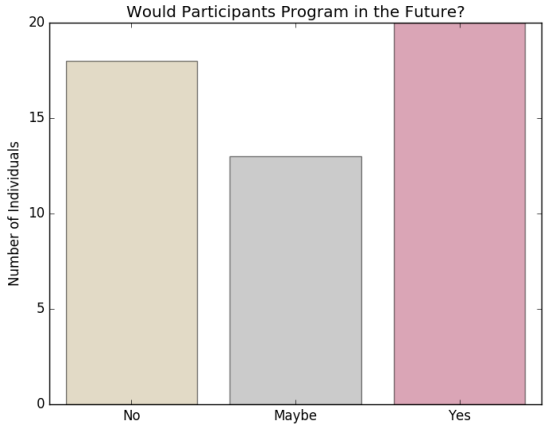
(b) Pie chart

Figure 4.28: One of the major problems in computer science, especially at Allegheny College, is the number of students majoring in this area. Despite career prospects being high for students graduating with degrees in computer science, the number of students remains low. This question is designed specifically to see if the general population is knowledgeable about the student deficit in computer science as a subject study. As it turns out, the people who took this survey do know that computer science as a study is not saturated at all. Only three people out of the fifty-one participants believe that there are enough students studying computer science. This being said, despite people knowing that there is a lack of students interested in computer science, there is no influx into the department. The bar graph shows how little people believe that the area of study is saturated quite well. Although the pie chart shows percentages of the total results well, that is not necessary for these results to be understood well. The bar chart is more useful than the pie chart for displaying the results from this question.

Have you ever participated or would you ever consider participating in exercises to promote learning code (i.e. hour of code)?



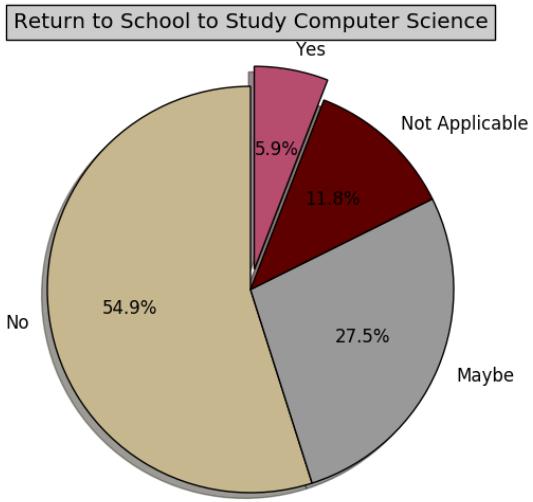
(a) Pie Chart



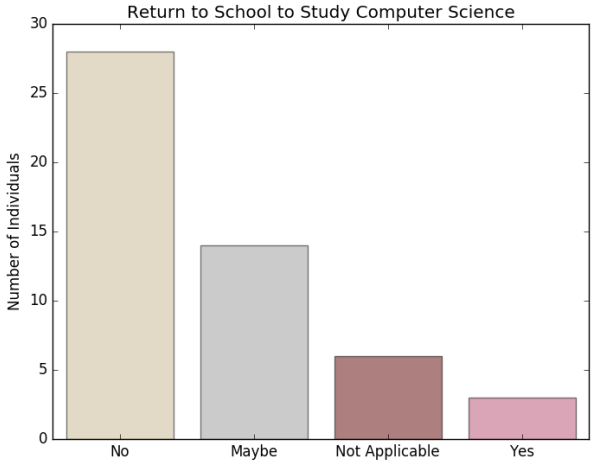
(b) Bar chart

Figure 4.29: The fact that participants may or may not have programmed before, as asked previously, does little to gauge the interest levels of people who may want to learn. By asking people if they have participated or if they would like to participate, this question seeks to know if people look upon learning how to code favorably. If the reader takes "Maybe" as positive, then well over half of the participants would be interested in learning how to code or participating in a coding exercise. In the bar chart, the difference between the bars looks small, because the number of "Yes" responses is very close to the number of "No" responses. However, when they are juxtaposed together, along with the "Maybe" answers in the pie chart, it becomes obvious that the "No" slice is dwarfed by the "Yes" and "Maybe" slices combined. The pie chart shows the results considerably better than the bar chart in this instance.

Would you ever want to go back to school to study computer science?



(a) Pie Chart

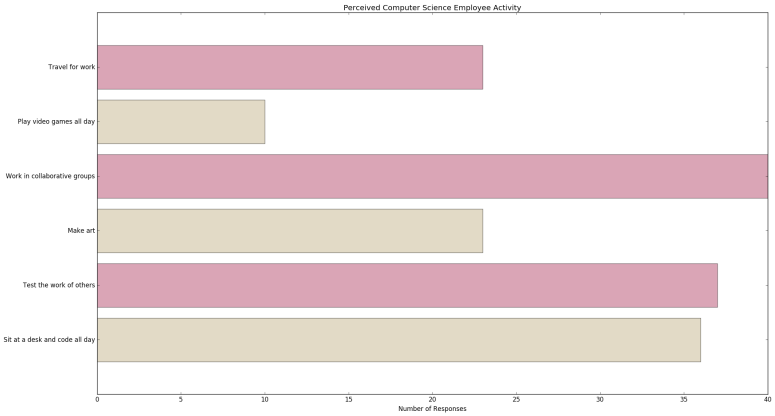


(b) Bar chart

Figure 4.30: The participants of this study were asked, should they return to school, if they would study computer science. The majority said that no, they would not be interested in studying computer science. Only a very small fraction of the participants expressed interest in wanting to study computer science. In today's economic climate and with a job shortage in America, this does not make sense. The pie chart shows an exploded "Yes" to highlight its small percentage. However, this is already obvious to any reader when looking at the bar chart. For this reason, the bar chart better represents the sad data that shows that people are uninterested in learning such an important study.

What do you think employees do in computer science/programming positions?

(a) Word cloud



(b) Bar chart

Figure 4.31: With all the results that have been displayed up to this point, it is no question that there are some participants who know the computer science field well and others who are grasping at straws. When asked to choose from a list what computer scientists do for a living, day in and day out, the most chosen option was "Work in collaborative groups". This is shown in the two visualizations above. As pleasing as it is to look at the word cloud, it does not keep the phrases together, but rather counts the uses of each individual word. The bar chart keeps the phrases intact and shows the results in a specific yet cohesive and easy-to-read way. The bar chart takes precedence in usefulness over the word cloud to accurately represent the results from this final question.

Chapter 5

Discussion and Future Work

In the following chapter, the results from the previous chapter are discussed and summarized. Following this summarization, there is a section that includes any new ways to recreate this project, should any individual or department choose to. Ultimately, there is a conclusion that includes potential ways answer the questions and solve the problems presented in the introductory section of this paper.

5.1 Summary of Results

The Twitter sentiment analysis results all err on the side of positivity. With this being said, the results did contain some outliers. There were the occasional negative results, but especially with regards to the "#ComputerScience" sections, the results were many and all clustered to the right of the middle line and were green in color, indicating positivity. These results do little to answer the question as to why there are less students in computer science departments as compared to other areas of study.

However, the results from the survey are much more interesting and get into

more detail, so that the data can be analyzed and the answer to the aforementioned question can be extrapolated and compared to the results of the twitter sentiment analysis.

Ultimately, there are not enough people that are willing to branch out into a type of study that they do not know. Because computer science is a new field, lower education (grade schools and high schools) has only just started teaching it to students within the past five or so years, and at some schools, computer science is not taught at all. Computer science remains thinly veiled behind the term "technology" when it comes to educational classes outside of colleges and universities. This does little to help the population that has already completed those years of schooling, which is where the problem lies. Almost half of the participants in the study say that they have already graduated from college, meaning that their lower education has been complete for at least five years. Because people coming into colleges and universities already have a background in other areas of study, they are drawn to those areas of study and do not branch out to areas of study that may be foreign to them, which is exactly what computer science is. Emotive words that most of the participants in the study associate with computer science are "Intrigued" but also "Unintelligent" and "Confused". People are aware, though, that careers in computer science include programming most of the time, and that a lot of the population that studies computer science go into IT when they enter industry. Each of the questions and their analyses are discussed in great detail in the previous chapter. It is worth noting that the majority of the people who took this survey would not be interested

in studying computer science, as their aversion to the subject is apparent despite the fact that they seem to think positively of the career prospects that a knowledge of computer science brings.

5.2 Future Work

If this study were to be repeated, the questions in the survey may be re-written to adjust correctly to the time in which the study is being conducted. The graphs may be changed as well, so as to give the researcher more freedom or even to analyze different types of visualizations other than word clouds, pie charts, and bar charts. Hopefully, the problem with computer science recruitment and retention will be solved so that this study need not be repeated.

In the future, this researcher aims to ease others' qualms with computer science. If every individual who is already involved in computer science speaks out about the positive aspects of it, others' may be more apt to want to try to study it in the future. As time continues and computers and technology continue to grow into and permeate millions upon millions of peoples' lives, it can be hoped that people will want to make a difference in the technology that they do know and that this will lead them to study computer science.

Informational sessions at colleges about the uses of computer science and benefits that come from having a knowledge of it should happen often. If one student comes to an information session and learns just one thing that he finds interesting, it will

reach the ears of others or may motivate him to try his hand at coding and/or programming. The benefits of full transparency in reference to the types of materials that are used and learned about by computer science students will only serve to benefit small departments and campuses alike.

5.3 Conclusion

Clearly, computer science as a whole seems impenetrable to a majority of the participants. Technological fields are male-dominated, but computer science will continue to be male-dominated unless some changes take place. These changes may be things such as increasing group study time, so that females feel as though they can have a voice and can learn alongside others. Another potential change to the way that computer science is taught is to advertise it as relevant to try to make individuals see that so many aspects of their daily lives exists because of some facet of computer science. Computer science should be taught to younger students who are within lower education, so that the students coming into higher education will not be wary of computer science, as has been seen at many colleges, including Allegheny College. Technology is ever changing, which makes it difficult to keep up in the field of computer science. Because of this, the individuals who decide to take on studying computer science need to be able to adapt and teach themselves new topics as they are developed and discovered. Overall, there exists fear within the participants in regard to computer science, but there also exists intrigue. It is up to computer scientists, now, to find

a way to play on that intrigue and to convince people to step out of their comfort zones.

Appendix A

Python Code

```
# Code taken and manipulated from MatPlotLib
# https://pythonspot.com/en/matplotlib-bar-chart/
# What types of careers do you feel come from a knowledge of
   computer science?

import matplotlib.pyplot as plt; plt.rcdefaults()
import numpy as np
import matplotlib.pyplot as plt

objects = (
    'Programmer/Programming',
    'Engineer/Engineering',
    'Knowledge',
    'Computer(s)',
    'Science',
    'Software',
    'Code/Coding',
    'Design/Designer/Designing',
```

```
    'Developer/Development',
    'Technology/Tech',
    'Media & Film',
    'Support',
    'Data/Database',
    'Web/Website',
    'Management',
    'Marketing',
    'Architecture',
    'IT',
    'Information Sys./Informatics',
    'Graphic Arts',
    'Statistics',
    'Games/Gaming',
    'Government',
    'Security',
    'Editor',
    'App',
    'Business',
    'Analyst/Analysis',
    'Creator',
    'Hacking'
    )
y_pos = np.arange(len(objects))
performance =
   [26,8,2,12,3,12,5,11,8,10,2,2,3,5,1,2,1,7,2,1,1,3,1,4,1,2,1,2,1,1]
```

```
plt.barh(y_pos, performance, align='center', alpha=0.5, color= ["#
    c6b78e" ,"#b64d6e"])
plt.yticks(y_pos, objects)
plt.xlabel('Number of Words')
plt.title('Words used to Describe Career Prospects with CS Knowledge
    ')

plt.show()
```

```
# Code taken and manipulated from MatPlotLib
# https://pythonspot.com/en/matplotlib-bar-chart/
# What do you think employees do in computer science/programming
    positions? (Pick a few)

import matplotlib.pyplot as plt; plt.rcdefaults()
import numpy as np
import matplotlib.pyplot as plt

objects = (
    'Sit at a desk and code all day',
    'Test the work of others',
    'Make art',
    'Work in collaborative groups',
    'Play video games all day',
    'Travel for work'
```

```
    )
y_pos = np.arange(len(objects))
performance = [36,37,23,40,10,23]

plt.barh(y_pos, performance, align='center', alpha=0.5, color= ["#
   c6b78e" ,"#b64d6e"])
plt.yticks(y_pos, objects)
plt.xlabel('Number of Responses')
plt.title('Perceived Computer Science Employee Activity')

plt.show()
```

```
# Code taken and manipulated from MatPlotLib
# https://pythonspot.com/en/matplotlib-bar-chart/
# What is your gender?

import matplotlib.pyplot as plt; plt.rcdefaults()
import numpy as np
import matplotlib.pyplot as plt

objects = ('Males', 'Females')
y_pos = np.arange(len(objects))
performance = [21.6,78.4]

plt.bar(y_pos, performance, align='center', alpha=0.5, color=["#
   c6b78e" ,"#b64d6e"])
plt.xticks(y_pos, objects)
```

```
plt.ylabel('Percentage')

plt.title('Gender')


plt.show()
```

```
# Code taken and manipulated from MatPlotLib

# http://matplotlib.org/1.2.1/examples/pylab_examples/pie_demo.html

# What is your gender?


from pylab import *

from matplotlib import cm


# make a square figure and axes

figure(1, figsize=(6,6))

ax = axes([0.1, 0.1, 0.8, 0.8])


colors = ["#c6b78e" ,"#b64d6e"]




# The slices will be ordered and plotted counter-clockwise.

labels = 'Males', 'Females'

fracs = [21.6,78.4]

explode=(0.1, 0)


pie(fracs, explode=explode, labels=labels,

                autopct='%1.1f%%', colors=colors, shadow=True,

                   startangle=90)
```

```
                # The default startangle is 0, which would start
                # the Frogs slice on the x-axis.  With startangle
                   =90,
                # everything is rotated counter-clockwise by 90
                   degrees,
                # so the plotting starts on the positive y-axis.

title('Gender', bbox={'facecolor':'0.8', 'pad':5})

show()
```

```
# Code taken and manipulated from MatPlotLib
# https://pythonspot.com/en/matplotlib-bar-chart/
# Are you a college graduate?

import matplotlib.pyplot as plt; plt.rcdefaults()
import numpy as np
import matplotlib.pyplot as plt

objects = ('No', 'Yes')
y_pos = np.arange(len(objects))
performance = [56.9,43.1]

plt.barh(y_pos, performance, align='center', alpha=0.5, color= ["#
   c6b78e" ,"#b64d6e"])
plt.yticks(y_pos, objects)
plt.xlabel('Percentage')
```

```
plt.title('College Graduate')


plt.show()
```

```
# Code taken and manipulated from MatPlotLib
# http://matplotlib.org/1.2.1/examples/pylab_examples/pie_demo.html
# Are you a college graduate?


from pylab import *
from matplotlib import cm


# make a square figure and axes
figure(1, figsize=(6,6))
ax = axes([0.1, 0.1, 0.8, 0.8])


colors = ["#c6b78e" ,"#b64d6e"]




# The slices will be ordered and plotted counter-clockwise.
labels = 'No', 'Yes'
fracs = [56.9,43.1]
explode=(0.1, 0)


pie(fracs, explode=explode, labels=labels,
                autopct='%1.1f%%', colors=colors, shadow=True,
                   startangle=90)
                # The default startangle is 0, which would start
```

```
                # the Frogs slice on the x-axis.  With startangle
                   =90,
                # everything is rotated counter-clockwise by 90
                   degrees,
                # so the plotting starts on the positive y-axis.

title('College Graduate', bbox={'facecolor':'0.8', 'pad':5})

show()
```

```
# Code taken and manipulated from MatPlotLib
# https://pythonspot.com/en/matplotlib-bar-chart/
# Have you ever participated or would you ever consider
   participating in exercises to promote learning code (i.e. hour of
    code)?

import matplotlib.pyplot as plt; plt.rcdefaults()
import numpy as np
import matplotlib.pyplot as plt

objects = ('No','Maybe','Yes')
y_pos = np.arange(len(objects))
performance = [18,13,20]

plt.bar(y_pos, performance, align='center', alpha=0.5, color=["#
   c6b78e","#999999","#b64d6e"])
plt.xticks(y_pos, objects)
```

```
plt.ylabel('Number of Individuals')
plt.title('Would Participants Program in the Future?')


plt.show()
```

```
# Code taken and manipulated from MatPlotLib
# http://matplotlib.org/1.2.1/examples/pylab_examples/pie_demo.html
# Have you ever participated or would you ever consider
   participating in exercises to promote learning code (i.e. hour of
    code)?


from pylab import *
from matplotlib import cm


# make a square figure and axes
figure(1, figsize=(6,6))
ax = axes([0.1, 0.1, 0.8, 0.8])


colors = ["#c6b78e","#999999","#b64d6e"]




# The slices will be ordered and plotted counter-clockwise.
labels = 'No','Maybe','Yes'
fracs = [18,13,20]
explode=(0, 0,0.06)


pie(fracs, explode=explode, labels=labels,
```

```
                autopct='%1.1f%%', colors=colors, shadow=True,
                    startangle=90)
                # The default startangle is 0, which would start
                # the Frogs slice on the x-axis.  With startangle
                    =90,
                # everything is rotated counter-clockwise by 90
                    degrees,
                # so the plotting starts on the positive y-axis.

title('Would Participants Program in the Future?', bbox={'facecolor'
    :'0.8', 'pad':5})

show()
```

```
# Code taken and manipulated from MatPlotLib
# https://pythonspot.com/en/matplotlib-bar-chart/
# What is/was your major?

import matplotlib.pyplot as plt; plt.rcdefaults()
import numpy as np
import matplotlib.pyplot as plt

objects = (
    'Accounting',
    'Anthropology',
    'Applied Math',
    'Biology',
```

```
    'Biochemistry',
    'Business',
    'Chemistry',
    'Communication Arts',
    'Computer Science',
    'Economics',
    'Education',
    'English',
    'Environmental Science',
    'Geology',
    'Global Health Studies',
    'Government',
    'History',
    'International Studies',
    'Natural Sciences',
    'Neuroscience',
    'Political Science',
    'Psychology',
    'Sociology',
    'Spanish',
    'Undecided',
    'Film/Television/Emerging Media'
    )
y_pos = np.arange(len(objects))
performance = [1,1,2,7,2,2,1,4,1,2,2,2,2,2,1,2,2,1,3,3,9,1,1,1,1,1]
```

```
plt.barh(y_pos, performance, align='center', alpha=0.5, color= ["#
    c6b78e" ,"#b64d6e"])
plt.yticks(y_pos, objects)
plt.xlabel('Number of Individuals')
plt.title('College Graduate')


plt.show()
```

```
# Code taken and manipulated from MatPlotLib
# http://matplotlib.org/1.2.1/examples/pylab_examples/pie_demo.html
# What is/was your major?


from pylab import *
from matplotlib import cm


# make a square figure and axes
figure(1, figsize=(6,6))
ax = axes([0.1, 0.1, 0.8, 0.8])


colors = ["#c6b78e" ,"#b64d6e","#f5deb3", "#793061","#FFFFFF","#5
    f0000","#999999","#b03060"]



# The slices will be ordered and plotted counter-clockwise.
labels = (
    'Accounting',
    'Anthropology',
```

```
    'Applied Math',
    'Biology',
    'Biochemistry',
    'Business',
    'Chemistry',
    'Communication Arts',
    'Computer Science',
    'Economics',
    'Education',
    'English',
    'Environmental Science',
    'Geology',
    'Global Health Studies',
    'Government',
    'History',
    'International Studies',
    'Natural Sciences',
    'Neuroscience',
    'Political Science',
    'Psychology',
    'Sociology',
    'Spanish',
    'Undecided',
    'Film/Television/Emerging Media'
    )
fracs = [1,1,2,7,2,2,1,4,1,2,2,2,2,2,1,2,2,1,3,3,9,1,1,1,1,1]
```

```
explode=(0,0,0,0,0,0,0,0,0.3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)


pie(fracs, explode=explode, labels=labels,
                autopct='%1.1f%%', colors=colors, shadow=True,
                   startangle=90)
                # The default startangle is 0, which would start
                # the Frogs slice on the x-axis.  With startangle
                   =90,
                # everything is rotated counter-clockwise by 90
                   degrees,
                # so the plotting starts on the positive y-axis.


title('College Major', bbox={'facecolor':'0.8', 'pad':5})


show()
```

```
# Code taken and manipulated from MatPlotLib
# https://pythonspot.com/en/matplotlib-bar-chart/
# Have you ever taken a programming lesson before?


import matplotlib.pyplot as plt; plt.rcdefaults()
import numpy as np
import matplotlib.pyplot as plt


objects = ('No','Maybe','Yes')
y_pos = np.arange(len(objects))
performance = [30,1,20]
```

```
plt.bar(y_pos, performance, align='center', alpha=0.5, color=["#
    c6b78e","#999999","#b64d6e"])
plt.xticks(y_pos, objects)
plt.ylabel('Individuals')
plt.title('Programmed Previously')

plt.show()
```

```
# Code taken and manipulated from MatPlotLib
# http://matplotlib.org/1.2.1/examples/pylab_examples/pie_demo.html
# Have you ever taken a programming lesson before?

from pylab import *
from matplotlib import cm

# make a square figure and axes
figure(1, figsize=(6,6))
ax = axes([0.1, 0.1, 0.8, 0.8])

colors = ["#c6b78e","#999999" ,"#b64d6e"]



# The slices will be ordered and plotted counter-clockwise.
labels = ('No','Maybe','Yes')
fracs = [30,1,20]
explode=(0, 0.2, 0)
```

```
pie(fracs, explode=explode, labels=labels,
                autopct='%1.1f%%', colors=colors, shadow=True,
                   startangle=90)
                # The default startangle is 0, which would start
                # the Frogs slice on the x-axis.  With startangle
                   =90,
                # everything is rotated counter-clockwise by 90
                   degrees,
                # so the plotting starts on the positive y-axis.

title('Programmed Previously', bbox={'facecolor':'0.8', 'pad':5})

show()
```

```
# Code taken and manipulated from MatPlotLib
# https://pythonspot.com/en/matplotlib-bar-chart/
# Would you ever want to go back to school to study computer science
   ?

import matplotlib.pyplot as plt; plt.rcdefaults()
import numpy as np
import matplotlib.pyplot as plt

objects = ('No','Maybe','Not Applicable','Yes')
y_pos = np.arange(len(objects))
performance = [28,14,6,3]
```

```
plt.bar(y_pos, performance, align='center', alpha=0.5, color=["#
    c6b78e","#999999","#5f0000","#b64d6e"])
plt.xticks(y_pos, objects)
plt.ylabel('Number of Individuals')
plt.title('Return to School to Study Computer Science')

plt.show()
```

```
# Code taken and manipulated from MatPlotLib
# http://matplotlib.org/1.2.1/examples/pylab_examples/pie_demo.html
# Would you ever want to go back to school to study computer science
    ?

from pylab import *
from matplotlib import cm

# make a square figure and axes
figure(1, figsize=(6,6))
ax = axes([0.1, 0.1, 0.8, 0.8])

colors = ["#c6b78e","#999999","#5f0000","#b64d6e"]



# The slices will be ordered and plotted counter-clockwise.
labels = 'No','Maybe','Not Applicable','Yes'
fracs = [28,14,6,3]
```

```
explode=(0, 0,0,0.1)


pie(fracs, explode=explode, labels=labels,
                autopct='%1.1f%%', colors=colors, shadow=True,
                   startangle=90)
                # The default startangle is 0, which would start
                # the Frogs slice on the x-axis.  With startangle
                   =90,
                # everything is rotated counter-clockwise by 90
                   degrees,
                # so the plotting starts on the positive y-axis.


title('Return to School to Study Computer Science', bbox={'facecolor
   ':'0.8', 'pad':5})


show()
```

```
# Code taken and manipulated from MatPlotLib
# https://pythonspot.com/en/matplotlib-bar-chart/
# When you hear the words "computer science", how do you feel?


import matplotlib.pyplot as plt; plt.rcdefaults()
import numpy as np
import matplotlib.pyplot as plt


objects = (
    'Happy',
```

```
    'Scared',
    'Intrigued',
    'Unintelligent',
    'Excited',
    'Intelligent',
    'Indifferent',
    'Confused',
    'Wonderful'
    )
y_pos = np.arange(len(objects))
performance = [3,8,23,14,6,8,11,13,1]

plt.barh(y_pos, performance, align='center', alpha=0.5, color= ["#
   c6b78e" ,"#b64d6e"])
plt.yticks(y_pos, objects)
plt.xlabel('Number of Individuals')
plt.title('Sentiment Towards Computer Science')

plt.show()
```

```
# Code taken and manipulated from MatPlotLib
# https://pythonspot.com/en/matplotlib-bar-chart/
# Do you think that computer science as a study is saturated with
   students?

import matplotlib.pyplot as plt; plt.rcdefaults()
import numpy as np
```

```
import matplotlib.pyplot as plt


objects = 'No','Not Sure','Yes'
y_pos = np.arange(len(objects))
performance = [36,12,3]


plt.bar(y_pos, performance, align='center', alpha=0.5, color=["#
    c6b78e" ,"#b64d6e"])
plt.xticks(y_pos, objects)
plt.ylabel('Number of Individuals')
plt.title('Opinions on Study Saturation')


plt.show()
```

```
# Code taken and manipulated from MatPlotLib
# http://matplotlib.org/1.2.1/examples/pylab_examples/pie_demo.html
# Do you think that computer science as a study is saturated with
    students?


from pylab import *
from matplotlib import cm


# make a square figure and axes
figure(1, figsize=(6,6))
ax = axes([0.1, 0.1, 0.8, 0.8])


colors = ["#c6b78e","#999999","#b64d6e"]
```

```
# The slices will be ordered and plotted counter-clockwise.
labels = 'No','Not Sure','Yes'
fracs = [36,12,3]
explode=(0.07, 0,0)

pie(fracs, explode=explode, labels=labels,
                autopct='%1.1f%%', colors=colors, shadow=True,
                   startangle=90)
                # The default startangle is 0, which would start
                # the Frogs slice on the x-axis.  With startangle
                   =90,
                # everything is rotated counter-clockwise by 90
                   degrees,
                # so the plotting starts on the positive y-axis.

title('Opinions on Study Saturation', bbox={'facecolor':'0.8', 'pad'
   :5})

show()
```

Bibliography

[1] Lori Carter. Why students with an apparent aptitude for computer science don't choose to major in computer science. *ACM SIGCSE Bulletin*, 38(1):27–31, 2006.

[2] Stephen J Ceci, Donna K Ginther, Shulamit Kahn, and Wendy M Williams. Women in science: The path to progress. *Scientific American Mind*, 26(1):62–69, 2015.

[3] Amanda Chapman. Using the assessment process to overcome imposter syndrome in mature students. *Journal of Further and Higher Education*, pages 1–8, 2015.

[4] Ana Conesa, Pedro Madrigal, Sonia Tarazona, David Gomez-Cabrero, Alejandra Cervera, Andrew McPherson, Michał Wojciech Szcześniak, Daniel J Gaffney, Laura L Elo, Xuegong Zhang, et al. A survey of best practices for rna-seq data analysis. *Genome biology*, 17(1):1, 2016.

[5] Kyran Dale. *Data Visualization with Python and JavaScript: Scrape, Clean, Explore & Transform Your Data*. " O'Reilly Media, Inc.", 2016.

[6] Peter Sheridan Dodds, Eric M Clark, Suma Desu, Morgan R Frank, Andrew J Reagan, Jake Ryland Williams, Lewis Mitchell, Kameron Decker Harris, Isabel M Kloumann, James P Bagrow, et al. Human language reveals a universal positivity bias. *Proceedings of the National Academy of Sciences*, 112(8):2389–2394, 2015.

[7] Daniel Fried and Stephen G Kobourov. Maps of computer science. In *Visualization Symposium (PacificVis), 2014 IEEE Pacific*, pages 113–120. IEEE, 2014.

[8] Susan N Friel, Frances R Curcio, and George W Bright. Making sense of graphs: Critical factors influencing comprehension and instructional implications. *Journal for Research in mathematics Education*, pages 124–158, 2001.

[9] Efthymios Kouloumpis, Theresa Wilson, and Johanna D Moore. Twitter sentiment analysis: The good the bad and the omg! *Icwsm*, 11(538-541):164, 2011.

[10] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM, 2010.

[11] Claus Lewerentz and Frank Simon. Metrics-based 3d visualization of large object-oriented programs. In *Visualizing Software for Understanding and Analysis, 2002. Proceedings. First International Workshop on*, pages 70–77. IEEE, 2002.

[12] Jennifer Light. Space science: The women who launched nasa. *Nature*, 532(7597):34–35, 2016.

[13] Beth Daniel Lindsay. Using google forms to track library space usage. *Journal of Access Services*, 13(3):159–165, 2016.

[14] Jane Margolis and Allan Fisher. *Unlocking the clubhouse: Women in computing.* MIT press, 2003.

[15] C MELODY CARSWELL, Sylvia Frankenberger, and Donald Bernhard. Graphing in depth: perspectives on the use of three-dimensional graphs to represent lower-dimensional data. *Behaviour & information technology*, 10(6):459–474, 1991.

[16] Hadi Partovi. What percent of stem should be computer science? - linkedin pulse, Jun 2014.

[17] Sally M Reis. We can't change what we don't recognize: Understanding the special needs of gifted females. *Gifted Child Quarterly*, 31(2):83–89, 1987.

[18] Alexander Repenning and Ashok Basawapatna. Drops and kinks: Modeling the retention of flow for hour of code style tutorials. In *The 11th Workshop in Primary and Secondary Computing Education (WiPSCE 2016), Münster, Germany*, 2016.

[19] Kirsten Risden, Mary P Czerwinski, Tamara Munzner, and Daniel B Cook. An initial examination of ease of use for 2d and 3d information visualizations of web content. *International Journal of Human-Computer Studies*, 53(5):695–714, 2000.

[20] Anthony Robins. The ongoing challenges of computer science education research. *Computer Science Education*, 25(2):115–119, 2015.

[21] Priti Shah and James Hoeffner. Review of graph comprehension research: Implications for instruction. *Educational Psychology Review*, 14(1):47–69, 2002.

[22] Balaji T Thattai, Karthikeyan Shanmugam, Chi Yao, and Yee Lang Lau. Systems and methods for generating data visualization applications, February 24 2016. US Patent App. 15/052,449.

[23] Susan Bell Trickett and J Gregory Trafton. Toward a comprehensive model of graph comprehension: Making the case for spatial cognition. In *Diagrammatic representation and inference*, pages 286–300. Springer, 2006.

[24] Susan Vinnicombe and Val Singh. Women-only management training: An essential part of women's leadership development. *Journal of Change Management*, 3(4):294–306, 2002.

[25] Brenda Cantwell Wilson. A study of factors promoting success in computer science including gender differences. *Computer Science Education*, 12(1-2):141–164, 2002.

[26] Mary Wyer, Mary Barbercheck, Donna Cookmeyer, Hatice Ozturk, and Marta Wayne. *Women, science, and technology: A reader in feminist science studies*. Routledge, 2013.

[27] Jeff Zacks, Ellen Levy, Barbara Tversky, and Diane J Schiano. Reading bar graphs: Effects of extraneous depth cues and graphical context. *Journal of experimental psychology: Applied*, 4(2):119, 1998.