5-2020

Mutually exclusive: a survey of ethical decision making in technology

Connor McPherson
University of Tennessee at Chattanooga, fkn734@mocs.utc.edu

Mutually Exclusive: A Survey of Ethical Decision Making in Technology

Connor McPherson

Departmental Honors Thesis
The University of Tennessee at
Chattanooga Computer Science

Examination Date: 3/17/20

Dr. Claire L. McCullough
Professor of Computer Science and Engineering
Thesis Director

Dr. Joseph D. Dumas II
UC Foundation Professor in the Department of Computer Science and Engineering
Department Examiner

Dr. Sumith Gunasekera
Associate Professor in the Department of Mathematics
Statistics Department Examiner

Contents

Abstract

People make hundreds of decisions every day. Developers, security consultants, operations engineers, designers, and engineers all make small decisions that affect the final product. The values people choose to promote and ignore appear in the constraints and biases of the products they craft. This paper discusses the process of developing, distributing, and analyzing a values survey to computer professionals and students in East Tennessee. I use advanced calculations of significance and beta for chi-squared tests to determine significance and discuss the ethical conclusions from the survey's data.

Introduction

Ethics in programming is a hot topic in data privacy [18] and machine learning [19], but programmers' values also play a significant role in product development and maintenance. When a developer, security consultant, or operations specialist makes any decision about a product, that decision comes from what that programmer values and, given a buildup of similar choices, can affect the project. One example is Huff and Cooper's empirical study of sex-bias in design. The researchers asked a group of designers to propose designs for educational software for students. In two test groups, the researchers explicitly mentioned that the students were male or female, while they told a third test group to propose designs for "students." Huff and Cooper found that the designs proposed by subjects in the gender-unspecified group were empirically similar to the designs proposed for boys and were different from the designs proposed for girls, even among designers who were female [15]. Huff and Cooper showed how preexisting biases and values implicitly influenced the design and gave rise to bias in the software.

Cory Knobel, CEO of RAW Consulting, and Geoffrey Bowker, Director of the Values in Design Laboratory at Bren, warned that "conversations and analyses of the values found in technologies are generally engaged after design and launch, and most users are faced with a daunting set of decisions already made on their behalf" [17].

Friedman and Nissenbaum, in their seminal essay, "Bias in Computer Systems", outlined three kinds of bias in software: Preexisting, technical and emergent [13].

- Preexisting bias is societal, systematic bias held implicitly by consumers of a society that disseminates those biases. This includes gender bias, as in Cooper and Huff's research, and racial bias [13].
- Technical bias is exclusion by the constraints of software or hardware, and the design choices made as a result. As Friedman and Nissenbaum explain, "A technical constraint imposed by the size of [an airport monitor] screen forces a piecemeal presentation of flight options and, thus, makes the algorithm chosen to rank flight options critically important. Whatever ranking algorithm is used… the system will exhibit technical bias" [13].
- Emergent bias is the most difficult to spot during design, as it develops after development with changes to the software's environment after launch, creating scenarios or use cases that designers never had to consider during development [13]. This bias can unveil the

> values inherent in the designers. Programmers building software for coworkers will see emergent problems if that software is distributed to the public.

Friedman and Nissenbaum explain,

> Envision a hypothetical system designed for a group of airlines all of whom serve national routes. Consider what might occur if that system was extended to include international airlines. A flight-ranking algorithm that favors [flying with the same company for every flight segment] when applied in the *original* context with national airlines leads to no systematic unfairness. However, in the new context with international airlines, the automated system would place these airlines at a disadvantage and, thus, comprise a case of emergent bias [13].

Don Gotterbarn, current chair of the Association for Computing Machinery (ACM) Committee on Professional Ethics, said that,

> The changes in technology and the kinds and number of impacted stakeholders changed the fundamental nature of society. The development of the cell phone has changed people's access to information and to a wide variety of entertainment… Computers impact all areas of our lives and many life preserving functions are relegated to a piece of computer guided machinery [10].

If programming as a discipline is to continue having as profound an impact, the decisions programmers think about should not end at the design and maintenance of a project. As Don Gotterbarn continued, "It is not sufficient to limit any computer discipline to addressing purely technical issues. As a profession, we must not retreat behind the obscurity and complexity of computing artifacts. We must acknowledge and embrace our role in shaping society and take responsibility for our part in those changes" [10].

It is because of these concerns that the ACM and other professional societies develop and publish codes of ethics for their members. In the field of computer science, agencies such as the ACM and the British Computing Society release codes of ethics for their members. They promote integrity, professionalism, leadership, and public good. The ACM describes the purpose of these codes as to "serve as a basis for ethical decision-making." Recent research shows that these codes don't affect programming habits [4]; however, the codes are a comprehensive view of the values programmers parse through in the decisions they make. This research uses the professional codes of ethics as a lens for the values and internal biases of programmers and how those biases work their way into the products they create, support, and maintain.

Research Question

This research aims to answer one question:

- Is the way programmers respond to ethical scenarios dependent on their age, years of experience, or role as a student?

The goal of this research is to understand the individual biases that influence programmers' choices during ethical dilemmas that plague modern programming, as well as to gain an understanding of how to correct those biases.

Related Work

To test how programmers respond to ethical situations, this research uses a scenario-based model based on past surveys. These two surveys use scenarios in their tests and use statistical regressions to interpret their data:

In 1996, Dr. Susan Harrington at Georgia State studied codes of ethics and the influence on the denial of responsibility, with the conclusion that codes of ethics do have some small effect, especially for people who deny their responsibility to be ethical. While this helps build the case for the use of codes of ethics, it shows that codes are not strong enough to enforce ethical responsibility. Harrington concludes that "at minimum, managers must use a multifaceted approach to deterring computer abuse and not depend upon the simple solution of codes of ethics. The use of tactics, such as codes of ethics, for purposes of general deterrence should not be overstated but should not be discarded" [14].

In 2018, Andrew McNamara, Justin Smith, and Emerson Murphy-Hill surveyed software developers to determine whether the codes of ethics affect professional decision-making. The research concluded that they do not. McNamara's research shows that computer scientists are not significantly affected by codes of ethics; however, independent from the ACM code, the survey did not compare how respondents select responses according to their own biases [4].

Like these two studies, this research includes several ethical scenarios. Unlike these two studies, this research is entirely observational. It does not use a control group with a controlled stimulus. The goal of this research is to observe the ethical climate of programmers through the lens of the codes of ethics. Many of the scenarios used in this survey (questions 0, 3, 4, 7, 9 in Appendix), are adapted from the questions made by McNamara, Smith, and Murphy-Hill, making this research a continuation of their work. The questions in this survey place pairs of ethical values in a mutually exclusive scenario. This ensures that the respondents' results describe how they would react in an everyday decisions that force them to choose between two values.

For example, one question says:

Question 0: The last customer meeting for your project was a disaster. Communication has been limited for the last month and the customer is expecting a full report from today's meeting. As you leave your office for the meeting, you overhear the administrative assistant saying,

"If Joe calls in, please see that he calls home. His spouse says there is a mini-crisis."

You are to meet with Joe at the customer's office, and the two of you are to lead the meeting. Joe's participation is critical. Joe is quite nervous and often gives a bad impression if distracted. What do you do?

- Relay the information to Joe before the meeting
- Not relay the information to Joe before the meeting (see Appendix)

This question is one of the scenarios adapted from McNamara, Murphy-Hill, and Smith's survey, except that this version adds the variable of the unhappy customer and how critical this meeting is for keeping them informed. This scenario is a choice between helping your coworker and your responsibility to your client, two cornerstone values for many codes of ethics [1, 3, 7, 24].

Methodology

Tested Values

The survey was built using Google Forms, and covered six categories common to most ethical codes:

- **Transparency**, the principle of being open and honest to all stakeholders about everything that goes on before and during software production [1,7,24].
- **Respect for Privacy**, the principle of respecting other people's data, sensitive or otherwise [1,3,7,24].
- **Respect for Intellectual Property**, the principle of honoring other people's work, property, and ideas [1,3,7,24].
- **Helping colleagues**, the principle of helping one's fellow workers, and teaching them what they need to know to succeed [1,3,7,24].
- **Quality assurance**, the principle of refusing to release software that falls short of what has been promised in terms of security, usability, and completeness [1,3,7,24].
- **Self-improvement**, the principle of continual learning in computing, ethics, and the skills of communication [1,7,24].

These values were chosen based on their regularity through the above codes and their applicability in scenarios that force respondents to choose one over the other. Other values in the codes included competence, quality of life (of all people), social good, and security, which are often dependent on many of the above values. Decisions that promote privacy usually support security [23]. A programmer who values self-improvement will, by extension, become more competent. To keep the survey simple, it only tests independent values from the codes of ethics.

Survey Format

Each scenario in the survey is a multiple-choice question, including two responses that favor one value more than the other and sometimes two other responses that respect both equally. Respondents picked responses to each scenario based on how they would act in that situation. Along with these scenarios, respondents supplied their age, years of experience, student status (whether or not they were a student), and how highly they thought they held each value.

Beginning a survey, a respondent would agree to the following consent form:

> You are being invited to participate in a research study about the ethical beliefs of computing professionals. This study is being conducted by Connor McPherson (<my-email>) and is advised by Dr. Claire McCullough (<Dr-Claire's-email>) at the University of Tennessee at Chattanooga. This research has been approved by the UTC Institutional

Review Board.

This survey will take about 20 minutes to complete.

This survey is anonymous. Do not indicate your name on the survey. No one will be able to identify you or your answers, and no one will know whether or not you participated in the study. Your participation in this study is voluntary. By clicking “I agree” you are verifying that you 18 years of age or older and are voluntarily agreeing to participate. You are free to stop answering questions at any time or to decline to answer any particular question you do not wish to answer for any reason.

If you have any questions about your rights as a subject/participant in this research, or if you feel you have been placed at risk, you can contact Dr. Amy Doolittle, the Chair of the Institutional Review Board at 423-425-5563. Additional contact information is available at www.utc.edu/irb.

Thank you very much for your time and support.

The survey asked for age, years of experience, and whether the respondent was a student, but avoided recognizable data, such as physical characteristics or gender for security to avoid identification of respondents.

After each respondent completed the survey, he/she received a personalized report of their survey results through a self-hosted Heroku web app. The app was built in NodeJS and used a modified version of respondents’ data stored in a google sheet. The scrubbed google sheet held each respondents’ “rank” for each ethical value. When a respondent submitted their response, a script on a Google Web App would trigger that went through each question and added one to the rank for values that a response favors and subtracted one from values that are less favored. This was done quickly enough so that a respondent could get his/her results immediately through a URL to the Heroku app. The app delivered the data in a clean, readable format (see Figure 1). Each user was given a different URL with an encrypted extension (e.g., mysterious-cliffs-30411.herokuapp.com/results/U2FsdGVkX19kT+d2UCrBq7U8efXx520B) to ensure that each respondent’s results were private.

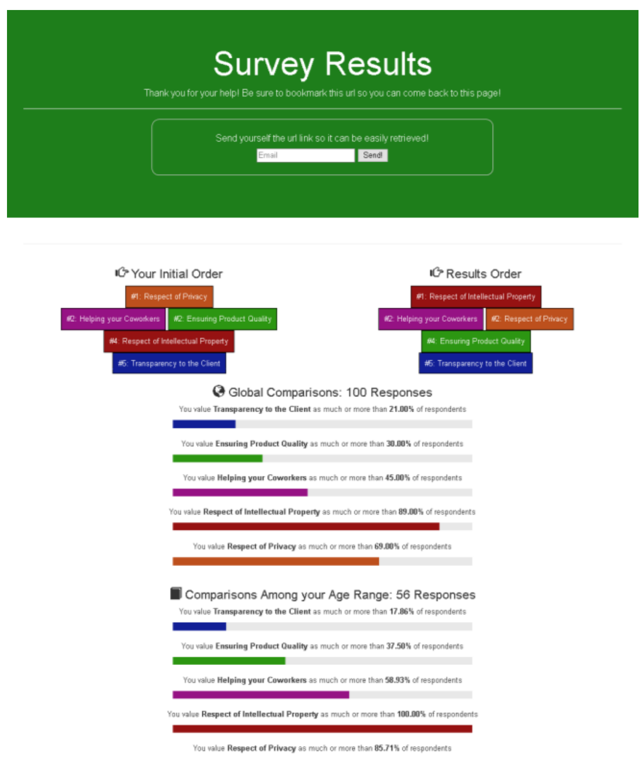

Figure 1: Survey Results Screen

Data Collection

This survey was exclusively advertised to programmers in the East Tennessee area to get a geographically consistent sample. The sample came from the southeast region of the US, specifically from the Chattanooga region and surrounding businesses. This research was limited geographically so that future studies can use it in meta-analysis with other surveys. Meta-analysis is “the method for combining the results from different studies on the same outcome of interest” [16]. If the survey is distributed to other areas of the world, the data can be combined to gain a larger view of programmers’ beliefs and values.

Distributing this survey in southeast Tennessee is unique among the United States due to the region’s strong startup support network. Significant contributors to this network are Launch Tennessee [11] and entrepreneur centers such as the Company Lab in Chattanooga [2] (where my survey collection is centered). This survey was distributed through these communities, making this survey not just a discussion on the values of programmers, but of programmers in the startup culture of Greater Chattanooga.

The survey was piloted with help from the Carbon Five community, a software contracting company that hosts a local hack night every two weeks, and posted through the forums of the ChaDev programmer community. The survey was also advertised with help from leaders in the ChaTech Council, a sponsor of events on new topics in computing. Lastly, the survey was emailed to members of the University of Tennessee at Chattanooga’s faculty and industrial advisory board to spread the survey to a broader audience. This helped gain responses from students at UTC and professionals at the Chattanooga branch of CGI, a global IT consulting company.

Results

Data collection officially closed February 13th, 2020, with a total of 90 responses from local professionals in computing.

The respondent count, distributed by age:
18-30 years of age: 57
31-40 years of age: 21
41-50 years of age: 3
51+ years of age: 9

The low number of older respondents meant the 41-50 and 51+ age groups had to be combined to create a large enough group. For all tables in the Appendix, these groups are merged under the 41+ age group.

By years of experience:
0-5 years of experience: 38
6-10 years of experience: 15
11-15 years of experience: 14
16-20 years of experience: 5
21-25 years of experience: 5

26+ years of experience: 13

Similar to the case of the age groups, the experience groups had too few responses per group, so the 6-10 and 11-15 were groups combined, and the 16 and up groups were merged into the 16+ group:

0-5 years of experience: 38
6-15 years of experience: 29
16+ years of experience: 23

By Student Status:
Not a student: 48
Is a student: 42

Total responses: 90

This data can be represented quantitatively with stacked columns for each question (see Figure 2).

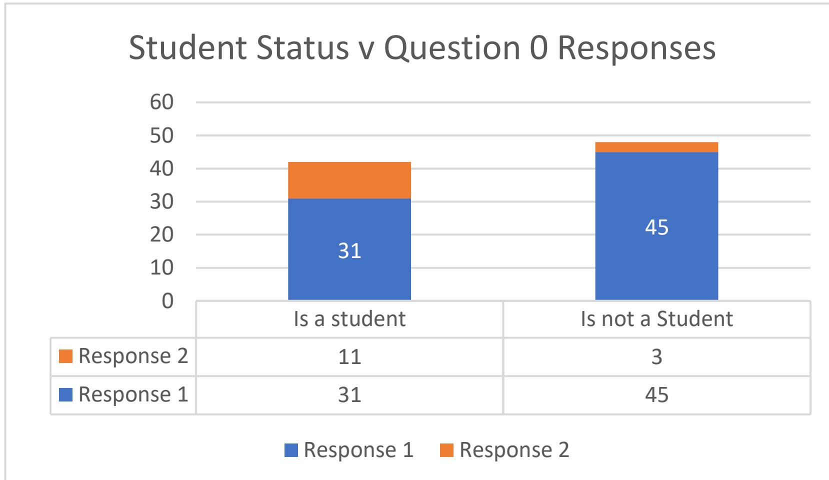

Figure 2: Stacked counts of the responses to Question 0

Choice of Regression

To get significant results from the data, a regression method had to be selected to test the relation between how programmers responded and their demographics (years of age, years of experience, and status as a student). The available regressions for this question depend on the types of data represented in the independent and dependent variables (see Figure 3).

This research studies individual responses to questions and their dependence on categories such as "0-5 years of experience" and "6-10 years of experience." As shown in Figure 3, to test whether there is a correlation between age categories and question responses (two categorical groups), the chi-squared test is the most appropriate.

		Dependent Variable	
		Categorical	**Continuous**
Independent Variable	**Categorical**	Chi-Squared	ANOVA
	Continuous	Logistic Regression	Linear Regression

Figure 3: Choice of Statistical Procedure by data type

Using the Chi-Squared Test

The chi-squared test for independence is a test that determines whether two categorical factors are related based on a "contingency table" of the counts of each category [20]. The test works by stating a "Null Hypothesis." The hypothesis assumes the categories have no relation to each other with the hope that the observed data will contradict this assumption. If it does, we can say that the categories are related.

The chi-squared test begins by comparing the variation of counts between several categories based on the assumed distribution of values if the categories weren't related [20]. To do this, the data is compiled into a contingency table:

Student Status v Question 0	Is a Student	Is not a Student	***Row Sum***
Chose Response 1	31	45	76
Chose Response 2	11	3	14
Column Sum	42	48	

The variance of the table, called the chi-squared value, can be calculated with the equation:

$$\chi^2 = \sum \frac{\left(O_{i,j} - E_{i,j}\right)^2}{E_{i,j}}$$

such that $O_{i,j}$ is the sample value from the table above, and $E_{i,j}$ is the expected value if the variables in the table are not related [20]. Each expected value is calculated from the row sum and column sum for each element.

$$E_{i,j} = \frac{rowsum_i * colsum_j}{N}$$

So for the above table, the χ^2 value is:

$$\chi^2 = \frac{(31 - 35.47)^2}{35.47} + \frac{(45 - 40.53)^2}{40.53} + \frac{(11 - 6.53)^2}{6.53} + \frac{(3 - 7.47)^2}{7.47} = 6.79$$

This calculation can be done in Python using the scipy package:

```
import numpy as np
from scipy.stats import chi2_contingency
obs = np.array([[31, 45], [11, 3]])
chiVal, _, _, _= chi2_contingency(obs, False)
print(chiVal)
>>> 6.780511546723954
```

This chi-squared value represents the variance of our table from the assumed expected values. For this value to represent a significant dependence (which would contradict the null hypothesis), the chi-squared value must be greater than a “critical point,” calculated from the chi-squared distribution based on the degrees of freedom (df) and significance level, alpha (α) (see Figure 4). The degrees of freedom for a chi-squared table is calculated with the formula [20]:

(Number of Rows-1) x (Number of Columns-1)

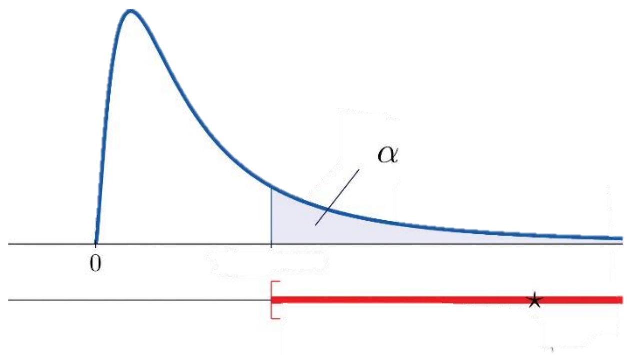

Figure 4: Chi-squared graph with the critical region highlighted

For a 2-by-2 table (as in the case above), the degrees of freedom would be (2-1)*(2-1) = 1. The chi-squared test also uses a selected alpha, which represents the chance of getting a false result. A significant result should have a significantly low alpha, such as 0.05, to lower the chance of a faulty result to 5%. Using this alpha, the critical value can be calculated in Python with the chi2.isf command:

```
import numpy as np
from scipy.stats import chi2_contingency
from scipy.stats import chi2

for table in getTables():
    chiVal, _, df, _ = chi2_contingency(table)
    if(chiVal > chi2.isf(0.05, df)):
        print(table, "is significant")
>>> [[45 31]
     [ 3 11]] is significant
```

The getTables() method does have to consider a few caveats, however. The chi-squared test assumes that the expected value ($E_{i,j}$) of each cell is greater than 5 for at least 80% of the cells and that all cells are greater than 1 [20]. This can usually be ensured by having more samples than 5 times the number of cells in any table. For my survey, the cell count never exceeds 10, which would make a sample size of 90 acceptable. Unfortunately, for questions 1, 3, 7, and 9, the results are skewed towards one or two responses. Experts advise combining the rows to remove the rows with too few counts [20], but while this validates the use of the chi-squared test, it limits the conclusions we can make on the relations after testing.

For questions 3, 7, and 9, combining two rows is sufficient, but question 1 is skewed too far towards one response so that the chi-squared test cannot work, and may not even be needed to see a trend in user responses.

Student v Question 1	Not a Student	Is a Student
Response 1	0	0
Response 2	6	5
Response 3	1	1
Response 4	41	36

Only by combining response 1, 2, and 3 can the chi-squared test be applied. While combining two responses may still allow for significant results, setting one response against every other can only tell us how one response is favored. For this reason, getTables() does not return the table for question 1; however, it is easily seen that all respondents are likely to choose response 4. Question 1 is a scenario that forces respondents to choose between helping coworkers and ensuring that a released product is usable:

Question #1: Helping Coworkers v Quality Assurance: The deadline for your team's project is tomorrow. The development team finished the product and handed it off to the operations team a month ago, but the product isn't working on any of the computers other than the developers', and two major bugs have sprung up in the last week. Communication between teams has devolved into making demands neither side can fulfill. What do you do?

- Spend all night fixing the product.
- Extend the deadline and sit down with the other team to discuss what went wrong and how to do better in the future.
- Ship it. Cut communication with the other team to release patches quickly over the next month.
- Ship the broken product before sitting down with the other team to release patches slowly over the next year.

The overwhelming bias of respondents towards choosing response 4 shows that respondents are more likely to value helping their coworkers than to value ensuring the quality of product that is about to be released. This is true irrespective of whether a respondents is older, younger, a student or otherwise.

This research used a survey with 10 questions, each of which can be compared against respondent age, years of experience, and student status. If we don't count question 1, this results in 27 contingency tables. The above code ran through every contingency table (by calling getTables()) and found only one significant result. With 95% certainty (1-alpha), we can say that whether a respondent is a student affects how they answer question zero (see Appendix). Unfortunately, with an alpha of 0.05, that is the only result.

Calculating Power

For any statistical test, there is a set of false results that are considered true and true results that are considered false. These are called Type I (α) and Type II (β) errors.

For the above example, there is a set of data that

	Should be rejected	Shouldn't be rejected
Test rejects the null hypothesis	True Positives	α (false negative rate)
Test fails to reject the null hypothesis	β (false positives rate)	True Negatives

Type I errors (α) represent seeing correlations where one doesn't exist. For the previous alpha of 0.05, there is only a 5% chance that our singular result is wrong. The sacrifice for this accuracy is that it increases β, the Type II error rate, resulting in missed significant results. This is especially true for the chi-squared test, which is a low-power test [9]. For this data, with only one significant result, a better alpha has to be calculated to increase power while keeping alpha acceptably low.

Beta is a function of the degrees of freedom, alpha, and chi-squared variable calculated above, and can be calculated by the Python equation:

```
def calcBeta(alpha, df, chi2Var):
    beta = ncx2.cdf(chi2.isf(alpha, df), df, chi2Var)
    return beta
```

Beta can then be used to calculate the true positive rate, which is the "power" of a function. Power is equal to 1-β. A low beta results in a high power, which is good. To counteract the chi-squared test's naturally low power, this research uses the youden index of a ROC Curve.

A ROC curve (Receiver Operating Characteristic) is a visual representation of the tradeoff between the true positive rate (power) and false negative rate (alpha) [12]. Figure 5 shows an example of a ROC curve (orange). The diagonal (blue) shows where power (true positive rate) and alpha (false negative rate) are equal. A point on the ROC is better when power is greater and alpha is smaller. This scoring of a point is its "youden" (J) [22].

$$J = \text{power} - \text{alpha}$$

The point where the youden is largest is the "optimal cut-point." The alpha value of this optimal cut point is the "calculated alpha," which is a superior value for alpha than the nominal alpha, 0.05.

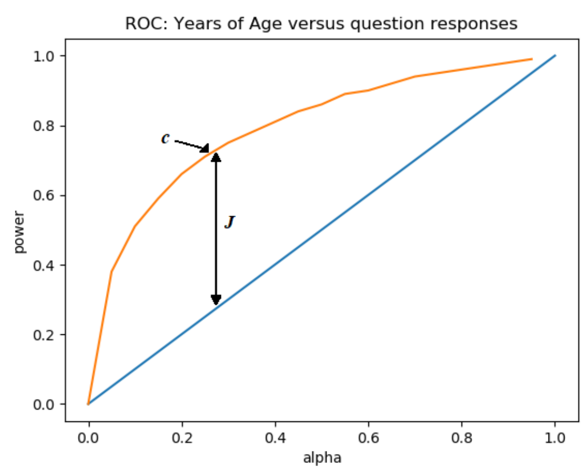

Figure 5: ROC curve depicting the youden (J) and optimal cut-point (c)

Finding the optimal cut point is easy to do in Python:

```python
def findOptimalCutpoint(df, chiVal):
    c = 0
    youden = 0
    # Iterate over every alpha and find the one with the highest jouden
    for alpha in np.arange(0, 1, 0.0125):
        power = calcPower(alpha, df, chiVal)
        if round(power, 3) - round(alpha, 3) > youden:
            youden = round(power, 3) - round(alpha, 3)
            c = round(alpha, 3)
    powerAtC = calcPower(c, df, chiVal)
    return c, powerAtC
```

Because power is a function of the chi-squared value of each contingency table, every contingency table has a different optimal cut-point and resulting alpha. Using Python's statistical packages, this program calculated each table's alpha before testing each chi-squared value for significance:

```python
    import chiQuestionVStudent
    results = chiQuestionVStudent.getResults()
    tables = chiQuestionVStudent.getTables()

    for i in questionUtil.ACCEPTED_VALUES:
        c, _, power = findOptimalCutpoint(results[i])
        chiVal, _, df, _ = chi2_contingency(tables[i])
        if(chiVal > chi2.isf(c, df)):
            print("Student v " + str(i), "is significant: alpha-
"+str(c) + " chiVal-"+str(chiVal) + " power-"+str(power) + " df-"+str(df))
```

This program ran over every contingency table again, and found the following results:

- The response to question 0 is related to whether the respondent is a student (alpha=0.15 power=0.809)
- The response to question 2 is related to whether the respondent is a student (alpha=0.262 power=0.535)
- The response to question 3 is related to whether the respondent is a student (alpha=0.162 power=0.807)
- The response to question 5 is related to whether the respondent is a student (alpha=0.162 power=0.806)
- The response to question 7 is related to whether the respondent is a student (alpha=0.25 power=0.651)
- The response to question 8 is related to whether the respondent is a student (alpha=0.175 power=0.779)
- The response to question 9 is related to whether the respondent is a student (alpha=0.15 power=0.801)

- The response to question 0 is related to the respondent's age (alpha=0.225 power=0.683)
- The response to question 3 is related to the respondent's age (alpha=0.15 power=0.817)
- The response to question 8 is related to the respondent's age (alpha=0.15 power=0.806)

Interestingly, the test failed to find any relation between question responses and the respondents' years of experience.

Keep in mind; this does not prove that there is no relation between programmer values and years of experience. The chi-squared test works by rejecting the null hypothesis that there is no relation between two factors [17]. We cannot prove that there is no relation when we began the test with that assumption.

Finding Trends

While these results are significant, the chi-squared test cannot tell us how they are significant, or what these results signify. The results from the chi-squared test have to be studied to find the trends between the independent variable (age and student status) and the dependent variable (question responses). For this, the significant contingency tables have to change into proportions (percentages). Doing this removes the number of samples from the data, effectively forgetting vital information, and so is a topic of controversy among statisticians [6], but proportions are still used in many tests of linear relation, specifically the Cochran-Armitage test [5], and so can still be considered useful.

For probability, a single count shouldn't be divided by the total N, but by the number of respondents in that age category, so we can find P(A | age) rather than just P(A ∩ B). This allows us to compare question responses according to the independent variable. This is called a Conditional Distribution [8].

The counts for our original example table were:

Student Status v Question 0	Is a Student	Is not a Student	***Row Sum***
Chose Response 1	31	45	76
Chose Response 2	11	3	14
Column Sum	42	48	

When each count is divided by the column sum, this results in:

Student Status v Question 0	Is a Student	Is not a Student	***Row Sum***
Chose Response 1	73.8%	93.75%	84.44%
Chose Response 2	26.2%	6.25%	15.56%
Column Sum	100%	100%	100%

This table shows that respondents who are not students are more likely to choose response 1 than respondents who are students. Question 0 is a question about valuing your coworkers versus keeping good relations with your client:

#0: Helping Coworkers v Transparency to the Client: The last customer meeting for your project was a disaster. Communication has been limited for the last month and the customer is expecting a full report from today's meeting. As you leave your office for the meeting, you overhear the administrative assistant saying,

"If Joe calls in, please see that he calls home. His spouse says there is a mini-crisis."

You are to meet with Joe at the customer's office, and the two of you are to lead the meeting. Joe's participation is critical. Joe is quite nervous and often gives a bad impression if distracted. What do you do?

Response 1: Relay the information to Joe before the meeting

Response 2: Not relay the information to Joe before the meeting

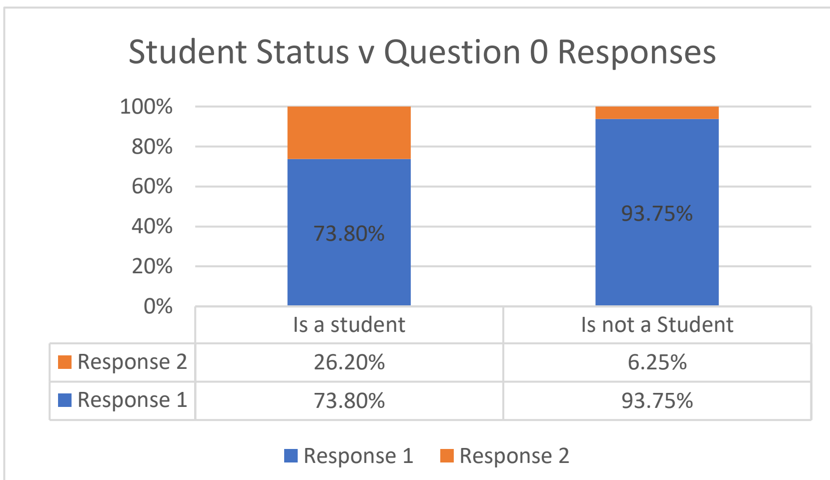

Figure 6: Stacked proportions of the Student Status v Question 0 contingency table

Respondents who are no longer students are far more likely to value their coworkers over their clients. Figure 5 visualizes this relationship. The relationships for all other tables show similar leanings (see Appendix), which are compiled into these results:

Student v 0: Non-students value coworkers over clear communication with the client

Student v 2: Non-students are more likely to honor their NDAs, even if it means missing a project milestone

Student v 3: Non-students are more likely to contact customers about issues during project specification, while students are more likely to build the project even with the flawed requirements

Student v 5: Non-students are more likely to opt to release a product immediately without data-collection software even if it means project bugs go undiscovered

Student v 7: Non-students are more likely to sell data to third parties, while students are more likely to add an opt-out setting for customers
Student v 8: Non-students are more likely to value the licenses of privacy-invasive libraries and use them as-is, while non-students are more likely to attempt to find a different library
Student v 9: Non-students are more likely to tell employers, rather than customers, about valuable information about the risks a product may have, while students are more likely to tell customers
Age v 0: Users older than 18-30 years old are more likely to value coworkers over communication with the client
Age v 3: The older a user is, the more likely he/she is to contact customers about issues during project specification
Age v 8: The older a user is, the more likely he/she is to value the licenses of privacy-invasive libraries and use them as-is, while 18-30-year-olds are more likely to attempt to find a different library

The last significant trend is from question 0, which was not calculated due to its uniform skew towards one response:

Student v 1: Both students and non-students value helping coworkers over ensuring that a shipped product is usable.

The conditional distribution table for Question 1 illustrates this:

Student v Question 1	Not a Student	Is a Student
Response 1	0.00%	0.00%
Response 2	12.50%	11.90%
Response 3	2.08%	2.38%
Response 4	85.41%	85.71%

Question 1 is a question comparing helping coworkers (with whom communications have broken down for) and the quality of a product that is shipping tomorrow. Most respondents would ship the product even if it doesn't work and would instead take time to repair the relationship with their coworkers.

Conclusions

From these results, we can infer about the values of programmers in southeast Tennessee:

- Most programmers seem to value their coworkers more than the good of a singular project.
- Older programmers care even more about coworkers than younger respondents do.
- Respondents out of college are more likely to value releasing a project quickly, even at the expense of quality or privacy.

- Respondents out of college are more likely to value intellectual property, while students are more likely to respect privacy.
- Older programmers and non-students value clients more than younger students.

Most of these results match common understandings of programming (such as more experienced programmers being more beholden to their bosses), but it is significant that all programmers, especially older ones, value their coworkers, or at least treat it as the ethical thing to do. This shows that, contrary to the stereotype, programmers are not predisposed to live in solitude or to be antisocial. Programmers feel a responsibility to their colleagues.

Potential Improvements for Future Surveys

While this survey found many significant results, thanks to the binary choices in the scenarios, there were several questions with a prominent "right" answer, allowing respondents to potentially "game" the survey, giving answers that are not honest but are more ethical. Questions with this problem are:

#1: Helping Coworkers v Quality Assurance: The deadline for your team's project is tomorrow. The development team finished the product and handed it off to the operations team a month ago, but the product isn't working on any of the computers other than the developers', and two major bugs have sprung up in the last week. Communication between teams has devolved into making demands neither side can fulfill. What do you do?

- Spend all night fixing the product.
- Extend the deadline and sit down with the other team to discuss what went wrong and how to do better in the future.
- Ship it. Cut communication with the other team to release patches quickly over the next month.
- Ship the broken product before sitting down with the other team to release patches slowly over the next year.

For this question, it is ethically superior to select option 2: Extend the deadline and sit down with the other team to discuss what went wrong. This not only values releasing a quality product but focuses on mending the relationship with the other team. While most respondents chose response 4, making this question skewed more towards helping coworkers, even at the expense of product quality, 24% of people chose option 2, making it the second-highest option, but whether respondents chose it honestly, or just because it is the "correct answer" is unclear.

#3: Quality Assurance v Transparency to the Client: In going over a software specification that your company has just been hired to create, your team discovers a large flaw in the requirements that could potentially hurt the customer's productivity when the product is finished. Your company has spent the last year trying to negotiate this lucrative contract and your managers do not want to tell the customers about the issue because it might extend the negotiations even further. What do you do?

- Complete the project following the original, but flawed requirements
- Update the requirements to fix the issue without the customer's feedback or knowledge
- Tell the customer about the issue after the contract has been signed
- Tell the customer about the issue immediately so the requirements can be updated

Question 3 is supposed to set quality assurance and transparency to clients apart, but instead is a dilemma between transparency to the client and following orders. To tell the client about the problem (valuing transparency) would cause the final product to be better (valuing quality), making this not a binary at all, but more of an ethical dilemma. The results from it are still valuable, but it cannot be said that respondents value transparency over quality.

#4: Helping Coworkers v Privacy: The company you work for is struggling to enter a lucrative market dominated by two of the Big Eight tech companies. When trying to figure out how to import data from one of those competitors' websites, you discover a severe 0-day vulnerability which would allow an exploiter to easily access all of the competitor's customer data. What do you do?

- Download all the user data that company has and use it to make your product competitive
- Do nothing about the vulnerability
- Report the information to the competitor through their dedicated means of bug reporting
- Download the company's data before anonymously reporting the issue
- Tell the company that the bug exists, and offer to be hired on as consultants to tell them where it is

Question 4 had no significant results, but if there were a significant result, it would be about the method of reporting an issue, between whether a respondent would respond anonymously, for free, or as a consultant. It would not yield conclusions about a respondents' likelihood to help coworkers. The question of downloading data is an ethical dilemma, but not a binary, and does not fit the criteria of this survey. Out of all the questions, this one would have to be modified most to qualify as a dilemma.

#6: Helping Coworkers v Intellectual Property: Your team uses proprietary third-party software to support your current project. Without it, the product cannot be worked on or improved. Due to an unforeseen emergency, the deadline for your project is pushed back one week, but in that time the license for the software expires. The shortest option to rent the license is for one year. Your boss doesn't want to buy the license for one year when you only need it for one week. Before your next team meeting, you learn one of the developers you are directly responsible for was able to bypass the license-check by hacking the login page and has already made good progress. What do you do?

- Report the employee to your boss
- Make your employee report the bug and ask your boss to renew the license
- Pirate the software for just one week

- Make your employee report the bug but use the software while the bug is being fixed

Question 6 suffers from many of the same problems as question 3: The two values it tries to set against each other are too easily valued together. This scenario takes holding a coworker to an ethical standard as a form of help; however, valuing that also values intellectual property, making it the clear right answer. As a result, all respondents chose answer two, "Make your employee report the bug and ask your boss to renew the license." Because of this, the question had no significance in the survey.

In these cases, respondents could respond dishonestly to be more ethically correct. Of course, if the survey didn't include these options, it could have alienated respondents who would respond in those ways. For future surveys, these questions should be reworked to either discount those options or make them less palatable, so only people who would make those choices would select them in the survey.

Other questions should also be reworded to represent all values equally. For instance, question 0 has the options:

- Relay the information to Joe before the meeting
- Don't relay the information to Joe before the meeting

This frames the question as wholly a coworker welfare question when it is a question matching coworker welfare and transparency to the customer. A better set of responses would be:

- Relay the information to Joe before the meeting
- Lead the client meeting with Joe before telling him

Problems with this work

While 90 responses is statistically significant for most chi-squared tables, the sample size is too small for some purposes. First, among of the student population, 39 out of the 43 were from the 18-30 age range. Therefore, it is more significant that age affects questions 0, 3, and 8, as that includes the result about students, who make up 39 of the 57 younger respondents.

Second, the sample size could not ensure that questions 3, 7, and 9 had expected results higher than 5.0 for each cell, which is the required minimum for the chi-squared test [20]. While combining rows is a simple way to correct small sample sizes, it limits the results that can be drawn from the survey. Future versions of the survey should consider rewriting or removing responses that had too low a row-sum.

Future Work

For future works in this field, besides revising some survey questions, data should be collected in a different geographical area, preferably in a place with both a thriving startup community and large corporations. This would allow a comparison of programmers from kinds of businesses, as well as geographies. This survey was exclusively distributed in the Chattanooga area to its thriving startup district. Later works can compare these results to others, and compile the data using meta-analysis to gain a fuller view of programmers' values.

Future iterations of this survey could also include new questions to test some of the unconsidered values from the codes of ethics: Competence, Quality of Life, Social Good, Self-Improvement, and Security. These questions will be more difficult to create but could yield a better understanding of the nuances between similar values, such as privacy and security.

The survey distribution's results screen was a great way to incentivize responses but would need to be upgraded to a paid platform, rather than a free Heroku dyno, to upscale to a larger sample size. Also, the ranking system would be more reliable if ranks were only increased based on favorable responses rather than being decreased by unfavorable responses, which may have overcompensated for the differences in values involved in a respondents' choice.

Lastly, the survey also included a 6 by 6 matrix that allowed respondents to rate each value. Some respondents interpreted this as an exclusive list that required each value to have a separate number, while others treated it as valuing based on a Likert scale. This confusion invalidated the question. Future surveys will have to choose one and phrase the question and label the values to signify how respondents should answer the question. This will allow more advanced analysis using ANOVA regressions (see Figure 3 above) and an analysis of not just the values programmers hold, but how well they think they hold them.

References

[1] "AAPC Code of Ethics." Code of Ethics, AAPC, 2016, www.aapc.com/aboutus/code-of-ethics.aspx.

[2] "About Us – The Company Lab (CO.LAB)." *The Company Lab*, colab.co/about-us/.

[3] "ACM Code of Ethics and Professional Conduct." *ACM Code of Ethics and Professional Conduct*, Association for Computing Machinery, 22 June 2018, www.acm.org/code-of-ethics.

[4] Andrew, Justin Smith, and Emerson Murphy-Hill. "Does ACM's code of ethics change ethical decision making in software development?" *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 2018.

[5] Armitage, Peter. "Tests for linear trends in proportions and frequencies." *Biometrics* 11.3 (1955): 375-386.

[6] Arnold, Barry C., et al. *Conditional specification of statistical models*. Springer Science & Business Media, 1999.

[7] “BCS Code of Conduct for BCS.” British Computer Society, 3 June, 2015, www.bcs.org/membership/become-a-member/bcs-code-of-conduct/.

[8] Bergsma, Wicher, and Tamás Rudas. "Conditional and marginal association in contingency tables." (2002).

[9] Cohen, Jacob. *Statistical power analysis for the behavioral sciences*. Routledge, 2013.

[10] Don Gotterbarn. 2016. THINKING PROFESSIONALLY Codes of ethics---: the conscience of a profession: connecting technology and society. *ACM Inroads* 7.4: 33-35.

[11] “Entrepreneurship Resources - Tennessee Startups - Who Is Launch Tennessee.” *Launch Tennessee*, launchtn.org/about/.

[12] Fawcett, Tom. "An introduction to ROC analysis." *Pattern recognition letters* 27.8 (2006): 861-874.

[13] Friedman, Batya, and Helen Nissenbaum. "Bias in computer systems." *ACM Transactions on Information Systems (TOIS)* 14.3 (1996): 330-347.

[14] Harrington, Susan J. "The effect of codes of ethics and personal denial of responsibility on computer abuse judgments and intentions." *MIS quarterly* (1996): 257-278.

[15] Huff, Charles, and Joel Cooper. "Sex Bias in Educational Software: The Effect of Designers' Stereotypes on the Software They Design 1." *Journal of Applied Social Psychology* 17.6 (1987): 519-532.

[16] Kelley, George A., and Kristi S. Kelley. "Statistical models for meta-analysis: A brief tutorial." *World journal of methodology* 2.4 (2012): 27.

[17] Knobel, Cory, and Geoffrey C. Bowker. "Values in design." *Communications of the ACM* 54.7 (2011): 26-28.

[18] Lutz, Christoph, and Aurelia Tamò. "RoboCode-Ethicists: Privacy-friendly robots, an ethical responsibility of engineers?." *Proceedings of the ACM Web Science Conference*. 2015.

[19] Malhotra, Charru, Vinod Kotwal, and Surabhi Dalal. "Ethical Framework for Machine Learning." *2018 ITU Kaleidoscope: Machine Learning for a 5G Future (ITU K)*. IEEE, 2018.

[20] McHugh, Mary L. "The chi-square test of independence." *Biochemia medica: Biochemia medica* 23.2 (2013): 143-149.

[21] Moore, David S. "Tests of chi-squared type." *Goodness-of-fit-techniques*. Routledge, 2017. 63-96.

[22] Schisterman, Enrique F., et al. "Optimal cut-point and its corresponding Youden Index to discriminate individuals using pooled blood samples." *Epidemiology* (2005): 73-81.

[23] Schneier, Bruce. *Schneier on security*. John Wiley & Sons, 2009.

[24] "Software Engineering Code of Ethics." Software Engineering Code of Ethics, IEEE, 1999, www.computer.org/web/education/code-of-ethics.

Acknowledgements

The author would like to acknowledge the help of Anna Sherman from ChaDev and Carla Askonas from ChaTech for their help in promoting the survey, and the Carbon Five community for helping pilot the survey during early stages of development. The survey would not have done as well without their support. The author also wants to thanks Dr. Sumith Gunasekera for his mentorship and advice during the data compilation portions of this research.

Appendix

Survey Questions

Respondent Information
What is your age?

- 18-30
- 31-40
- 41-50
- 51 or older

How many years of experience in computing have you had?

- 0-5
- 6-10
- 11-15
- 16-20
- 21-25
- 26 or more
- I'm a student

How important do you think each of these issues are to your career as a programmer (chosen on a scale of 6 between "Most Valued" and "Least Valued")?
Honesty to the Client
Quality Assurance
Respect of Privacy
Respect of Intellectual Property
Helping Colleagues

Questions:

#0: Helping Coworkers v Transparency to the Client: The last customer meeting for your project was a disaster. Communication has been limited for the last month and the customer is expecting a full report from today's meeting.
As you leave your office for the meeting, you overhear the administrative assistant saying,
"If Joe calls in, please see that he calls home. His spouse says there is a mini-crisis."
You are to meet with Joe at the customer's office, and the two of you are to lead the meeting. Joe's participation is critical. Joe is quite nervous and often gives a bad impression if distracted. What do you do?

- Relay the information to Joe before the meeting
- Not relay the information to Joe before the meeting

#1: Helping Coworkers v Quality Assurance: The deadline for your team's project is tomorrow. The development team finished the product and handed it off to the operations team a month ago, but the product isn't working on any of the computers other than the developers', and two major bugs have sprung up in the last week.

Communication between teams has devolved into making demands neither side can fulfill. What do you do?

- Spend all night fixing the product.
- Extend the deadline and sit down with the other team to discuss what went wrong and how to do better in the future.
- Ship it. Cut communication with the other team to release patches quickly over the next month.
- Ship the broken product before sitting down with the other team to release patches slowly over the next year.

#2: Quality Assurance v Respect of Intellectual Property: You and your friend work at two competing companies. Two days before the release deadline for a particularly time-consuming issue, you and your friend are talking over lunch. Suddenly, your phone rings. A personal emergency has come up. You absolutely won't be able to fix the issue before the deadline. Your friend offers to finish the code for you. When you two first met, you helped him a lot with his code and he wants to repay the favor. You were required to sign a Non-Disclosure-Agreement for this job, and you know your company isn't willing to hire consultants at this time (especially ones from their top competitor). What do you do?

- Finish the project in time with your friend's help
- Honor your contract and politely refuse

#3: Quality Assurance v Transparency to the Client: In going over a software specification that your company has just been hired to create, your team discovers a large flaw in the requirements that could potentially hurt the customer's productivity when the product is finished. Your company has spent the last year trying to negotiate this lucrative contract and your managers do not want to tell the customers about the issue because it might extend the negotiations even further. What do you do?

- Complete the project following the original, but flawed requirements
- Update the requirements to fix the issue without the customer's feedback or knowledge
- Tell the customer about the issue after the contract has been signed
- Tell the customer about the issue immediately so the requirements can be updated

#4: Helping Coworkers v Privacy: The company you work for is struggling to enter a lucrative market dominated by two of the Big Eight tech companies. When trying to figure out how to import data from one of those competitors' websites, you discover a severe 0-day vulnerability which would allow an exploiter to easily access all of the competitor's customer data. What do you do?

- Download all the user data that company has and use it to make your product competitive
- Do nothing about the vulnerability
- Report the information to the competitor through their dedicated means of bug reporting
- Download the company's data before anonymously reporting the issue

- Tell the company that the bug exists, and offer to be hired on as consultants to tell them where it is

#5: Quality Assurance v Privacy: You and your coworkers have been working for the last year on an update to an already existing accessibility app to make texting on smartphones easier. The software is used in a wide variety of applications, and you believe there may be issues that haven't been found. The release deadline is approaching, and one coworker suggests configuring the initial release to send an error report of everything being done by the user whenever a system breakdown occurs. This data collection would keep track of all recent events, running apps and current texting channels. Data collection for the sake of improving the software is allowed in the company's privacy policy. What do you do?

- Begin development of the data collection software
- Request to push back the deadline and build a small group of users to test the software with
- Release the software without collecting data and wait for users to report errors
- Develop the data collection software to get information on customers for future use and begin work on the next update without checking for errors in the last update

#6: Helping Coworkers v Intellectual Property: Your team uses proprietary third-party software to support your current project. Without it, the product cannot be worked on or improved. Due to an unforeseen emergency, the deadline for your project is pushed back one week, but in that time the license for the software expires. The shortest option to rent the license is for one year. Your boss doesn't want to buy the license for one year when you only need it for one week. Before your next team meeting, you learn one of the developers you are directly responsible for was able to bypass the license-check by hacking the login page and has already made good progress. What do you do?

- Report the employee to your boss
- Make your employee report the bug and ask your boss to renew the license
- Pirate the software for just one week
- Make your employee report the bug but use the software while the bug is being fixed

#7: Transparency to the Client v Privacy: Your company has been collecting anonymous usage statistics for their products for many years, but has recently been struggling to acquire new users, causing the company to consider scaling down operations. Seeing your company struggle and knowing the value of its customer data, an advertising company approaches you to use your company's user data to improve their ad recommendations. Your privacy policy does not explicitly mention selling user data to third party vendors. Turning down this offer may result in employees being fired. You are in charge of this decision; what do you do?

- Sign a contract with the advertising company without telling your users
- Sign the contract and add an opt-out setting for users to stop having their usage data collected
- Decline the offer with the advertising company

#8: Intellectual Property v Privacy: The team you lead is working on a smartphone app for finding local restaurants. For the past two months, the development team has been looking for the right library for querying Google Maps around the user's location, and you have recently found a library with all the functions the project needs. The library is open-sourced under a limited license that allows companies to use it commercially as long as they don't modify the library. After going over the source code, you find that the library tracks and saves unnecessary data, including users' name, phone number, birthday and common times the user is online, and you can't find where any of this data is used. Your coworkers are alarmed when you show them and one of them recommends that the library be edited to remove the features that save this data, but doing so would breach the library's license. You've tried getting into contact with the library's maintainer, with no response. The team has spent too much time searching for a library already. What do you do?

- Modify the library to remove the unnecessary data collection
- Use the library as is
- Don't use the library and hope another suitable library is found soon
- Extend the library's data collection to build a more personalized experience for the user

#9: Transparency to the Client v Intellectual Property: The company is currently being sued by a customer who is claiming that he was injured by one of the company's products. When your development duties take you to a part of your company's open sourced code that has not been looked at in years, you find a corner case that might support the customer's personal injury claim. There is a large sum of money at stake and the company is currently in good shape to win the case. What do you do?

- Sell the information to the customer
- Tell your employer but don't reveal the information to the customer
- Reveal the information to the customer without telling your employer
- Tell your employer before revealing the information in court

Raw Data

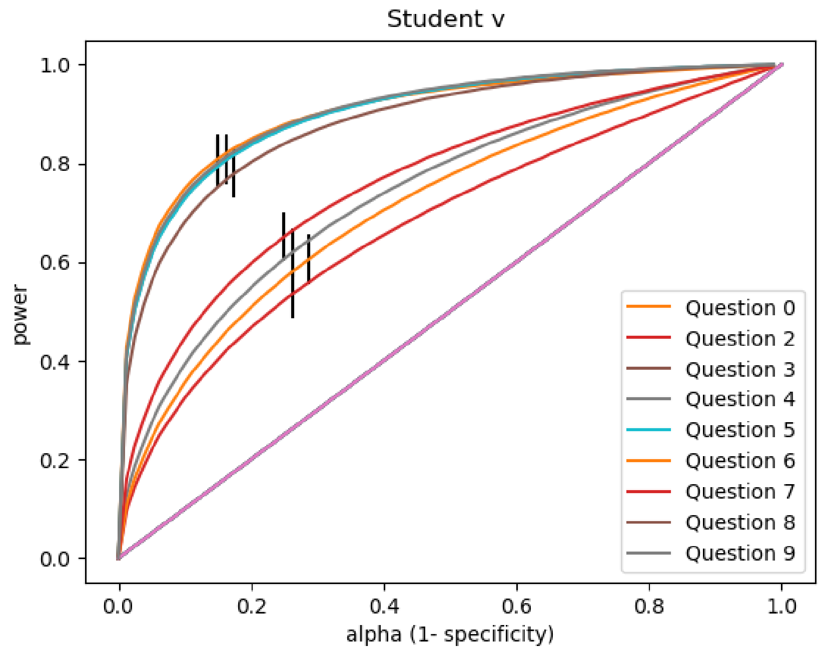

Figure 7 – Roc Curves for Question responses based on whether the respondent is a student (optimal cut points marked)

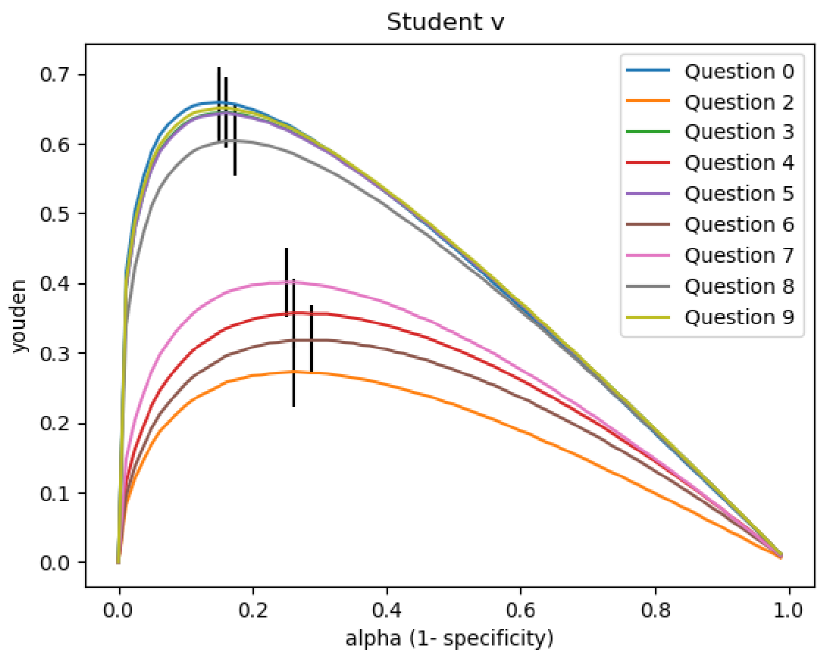

Figure 8 – Curve of Youden according to alpha for question responses based on whether the respondent is a student (optimal cut points marked)

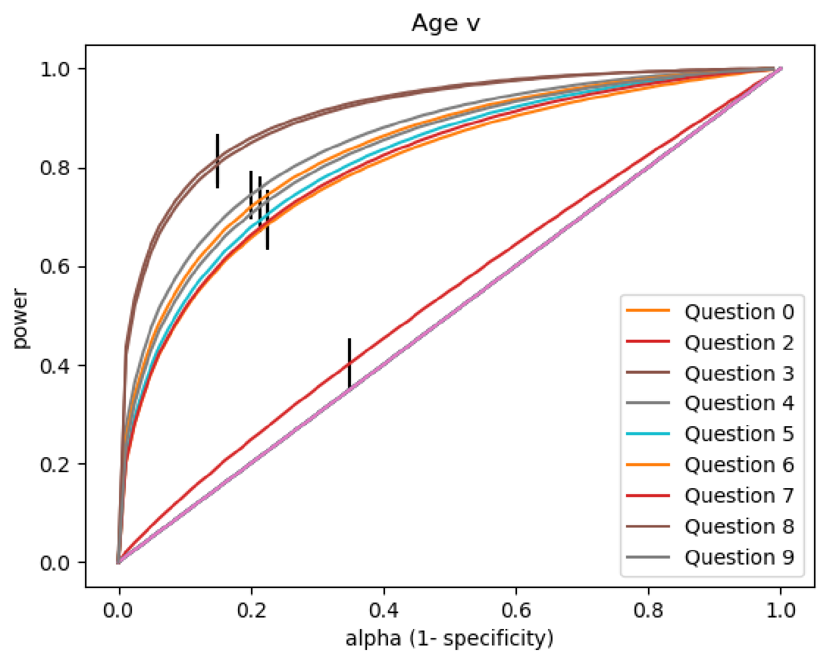

Figure 9 – Roc Curves for Question responses based on respondents' age (optimal cut points marked)

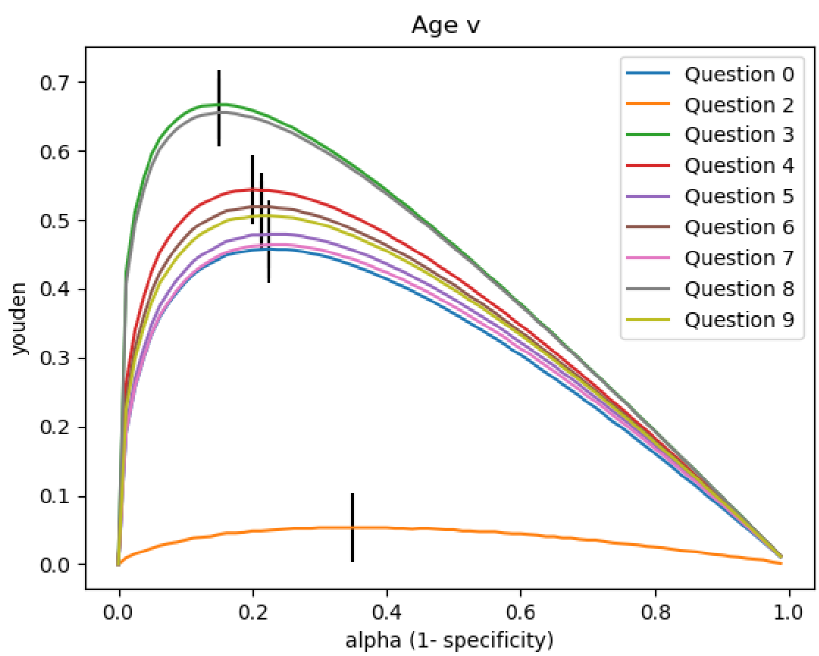

Figure 10 – Curve of Youden according to alpha for question responses based on respondents' age (optimal cut points marked)

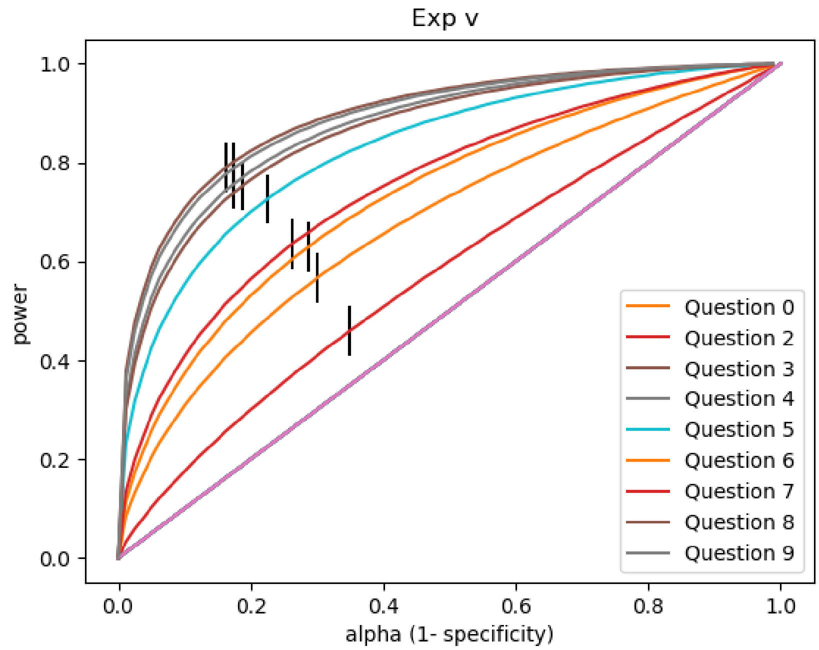

Figure 11 – Roc Curves for Question responses based on respondents' years of experience (optimal cut points marked)

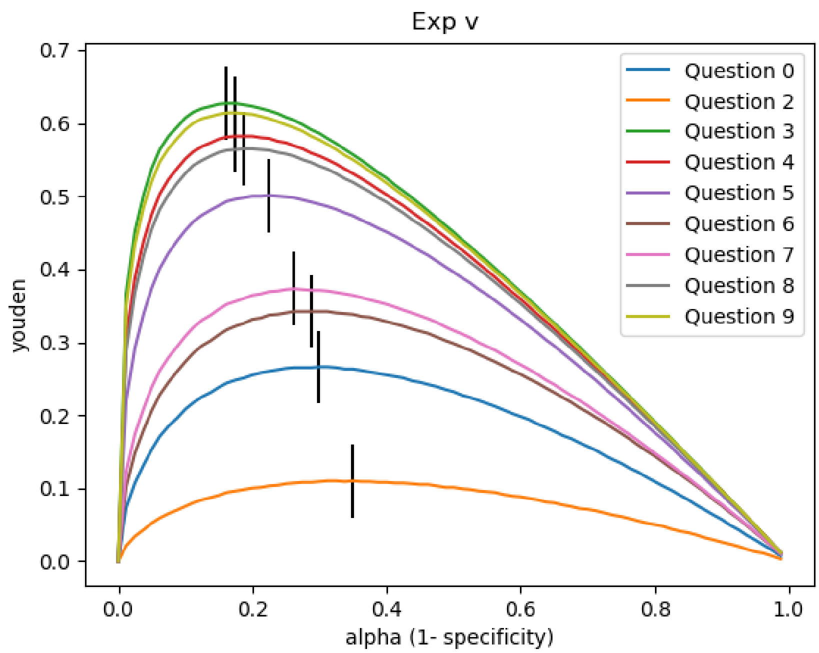

Figure 12 – Curve of Youden according to alpha for question responses based on respondents' years of experience (optimal cut points marked)

Tables for Significant Results:

Student v Question 0	Not a Student	Is a Student
Response 1	45	31
Response 2	3	11

Student v Question 0 Expected Values	Not a Student	Is a Student
Response 1	40.53	35.47
Response 2	7.47	6.53

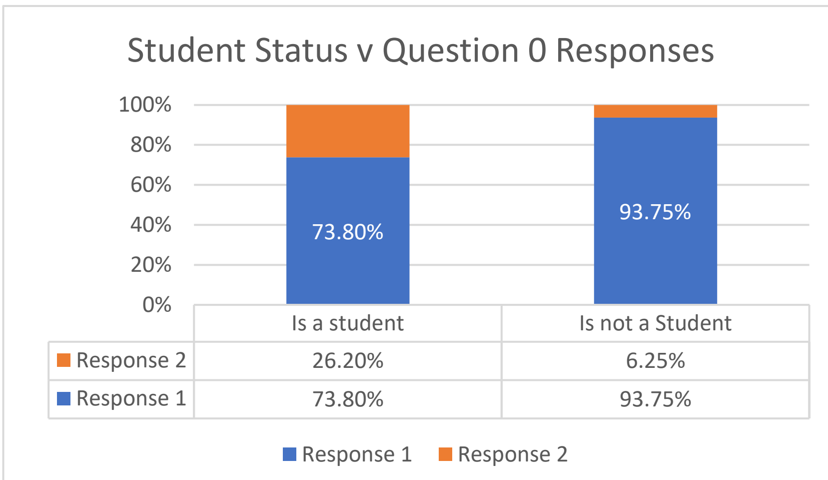

Figure 13: Stacked proportions of the Student Status v Question 0 contingency table

Student v Question 1	Not a Student	Is a Student
Response 1	0	0
Response 2	6	5
Response 3	1	1
Response 4	41	36

Student v Question 2	Not a Student	Is a Student
Response 1	4	8
Response 2	44	34

Student v Question 2 Expected Values	Not a Student	Is a Student
Response 1	6.40	5.60
Response 2	41.60	36.40

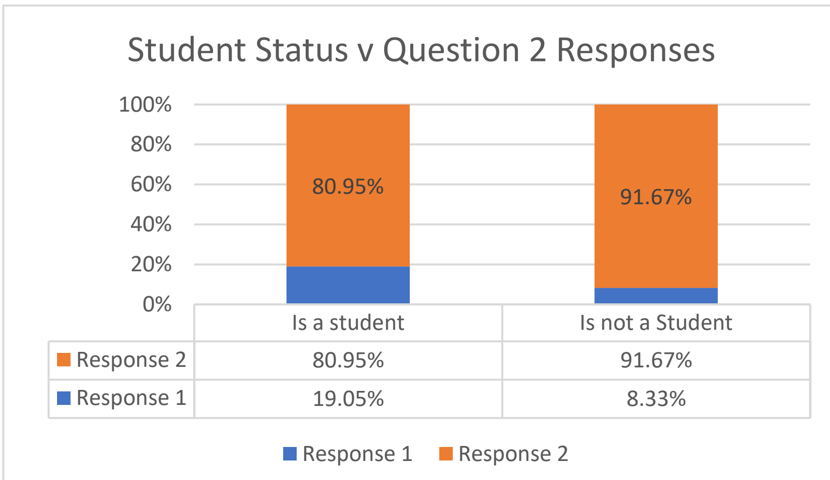

Figure 14: Stacked proportions of the Student Status v Question 2 contingency table

Student v Question 3	Not a Student	Is a Student
Response 1	2	9
Response 2	4	5
Response 3	3	1
Response 4	39	27

Student v Question 3 Expected Values	Not a Student	Is a Student
Response 1	5.87	5.13
Response 2	4.8	4.20
Response 3	2.13	1.87
Response 4	35.20	30.80

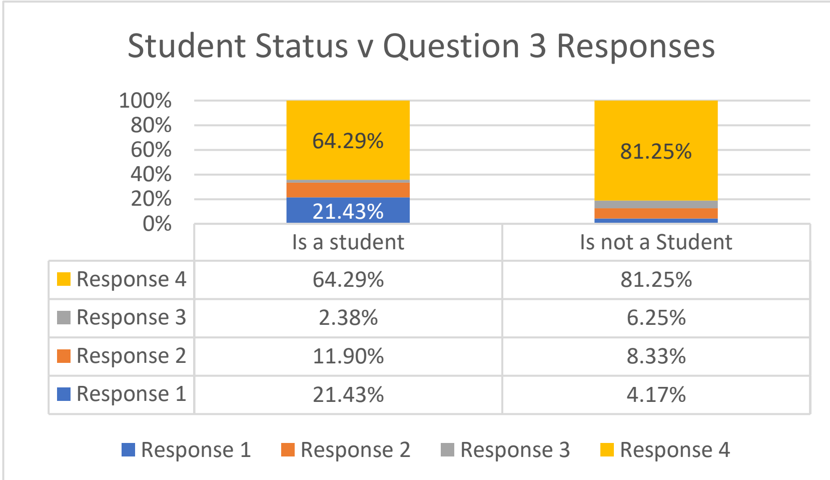

Figure 15: Stacked proportions of the Student Status v Question 3 contingency table

Student v Question 5	Not a Student	Is a Student
Response 1	9	8
Response 2	25	29
Response 3	10	1
Response 4	4	4

Student v Question 5 Expected Values	Not a Student	Is a Student
Response 1	9.07	7.93
Response 2	28.80	25.20
Response 3	5.87	5.14
Response 4	3.27	3.73

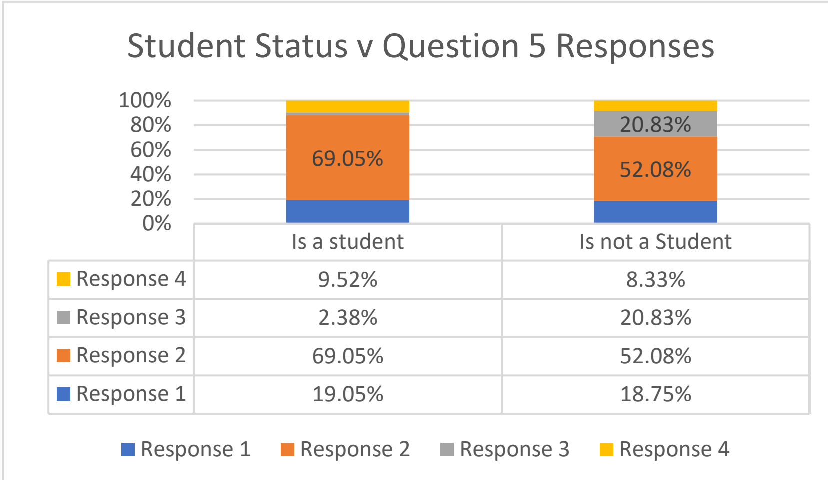

Figure 16: Stacked proportions of the Student Status v Question 5 contingency table

Student v Question 7	Not a Student	Is a Student
Response 1	19	10
Response 2	27	31
Response 3	2	1

Student v Question 7 Expected Values	Not a Student	Is a Student
Response 1	15.47	13.53
Response 2	30.93	27.07
Response 3	1.60	1.40

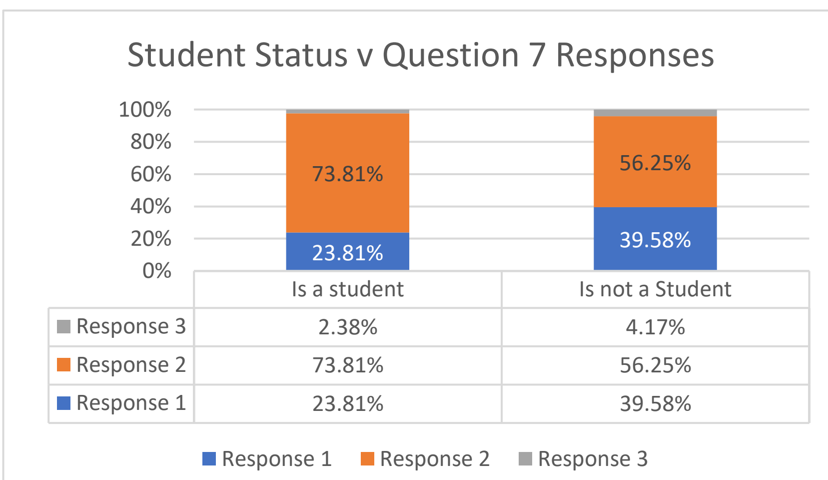

Figure 17: Stacked proportions of the Student Status v Question 7 contingency table

Student v Question 8	Not a Student	Is a Student
Response 1	7	6
Response 2	17	7
Response 3	6	13
Response 4	18	16

Student v Question 8 Expected Values	Not a Student	Is a Student
Response 1	6.93	6.07
Response 2	12.80	11.20
Response 3	10.13	8.87
Response 4	18.13	15.87

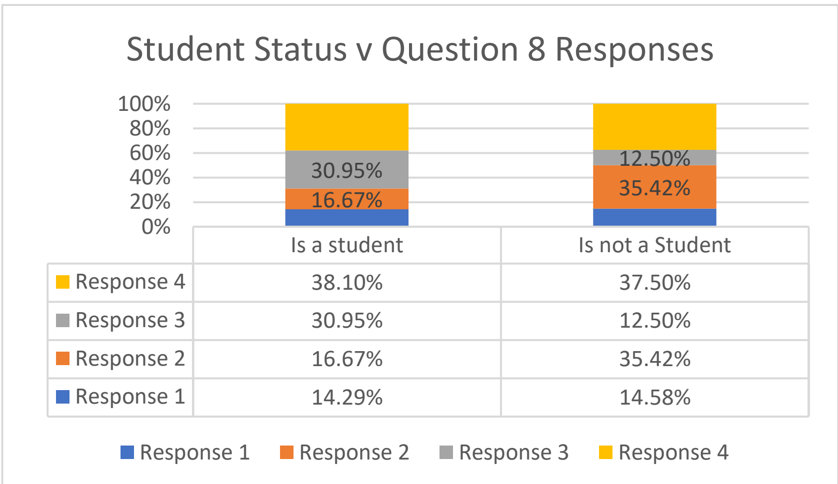

Figure 18: Stacked proportions of the Student Status v Question 8 contingency table

Student v Question 9	Not a Student	Is a Student
Response 1	2	2
Response 2	26	12
Response 3	1	0
Response 4	19	28

Student v Question 9 Expected Values	Not a Student	Is a Student
Response 1	2.13	1.87
Response 2	20.27	17.73
Response 3	0.53	0.47
Response 4	25.07	21.93

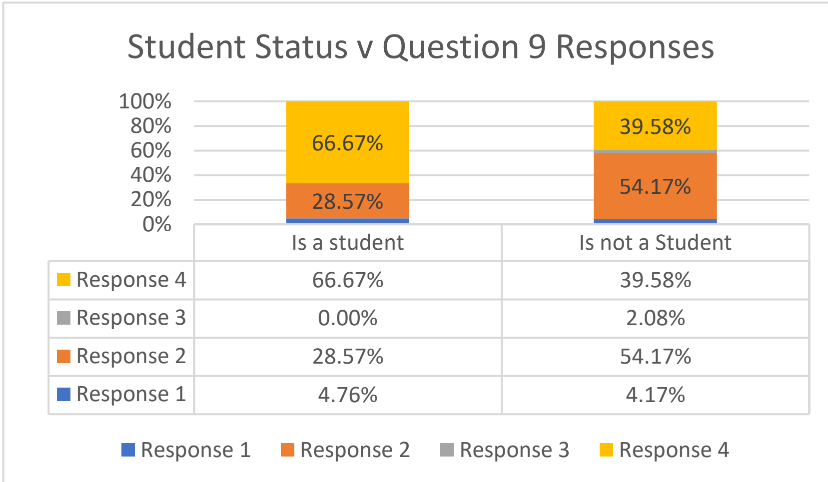

Figure 19: Stacked proportions of the Student Status v Question 9 contingency table

Age v Question 0	18-30	31-40	41+
Response 1	45	20	11
Response 2	12	1	1

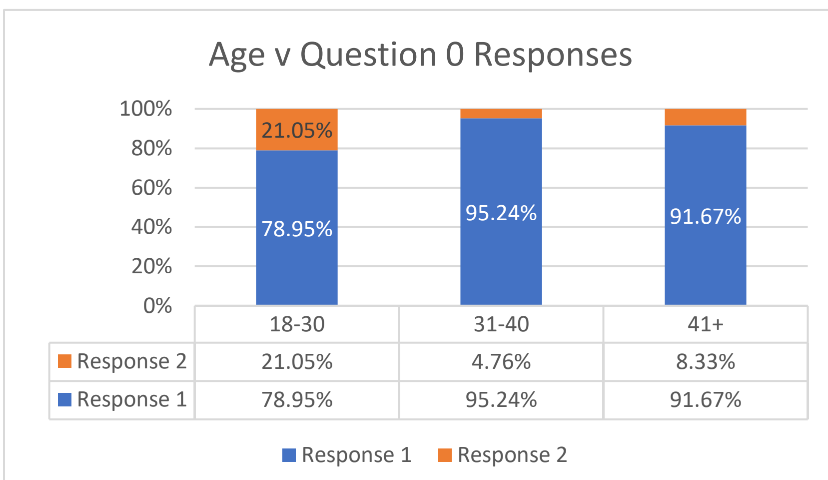

Figure 20: Stacked proportions of the Age v Question 0 contingency table

Age v Question 3	18-30	31-40	41+
Response 1	11	0	0
Response 2	4	4	1
Response 3	3	1	0
Response 4	39	16	11

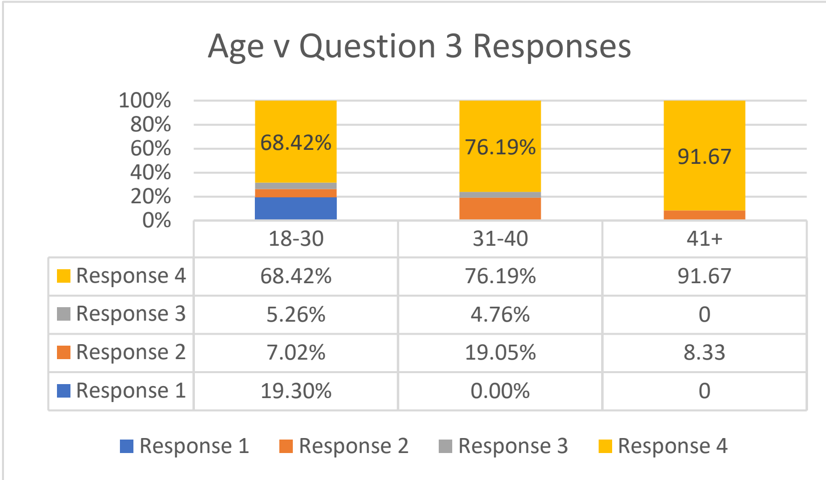

Figure 21: Stacked proportions of the Age v Question 3 contingency table

Age v Question 8	18-30	31-40	41+
Response 1	9	4	0
Response 2	11	7	6
Response 3	16	2	1
Response 4	21	8	5

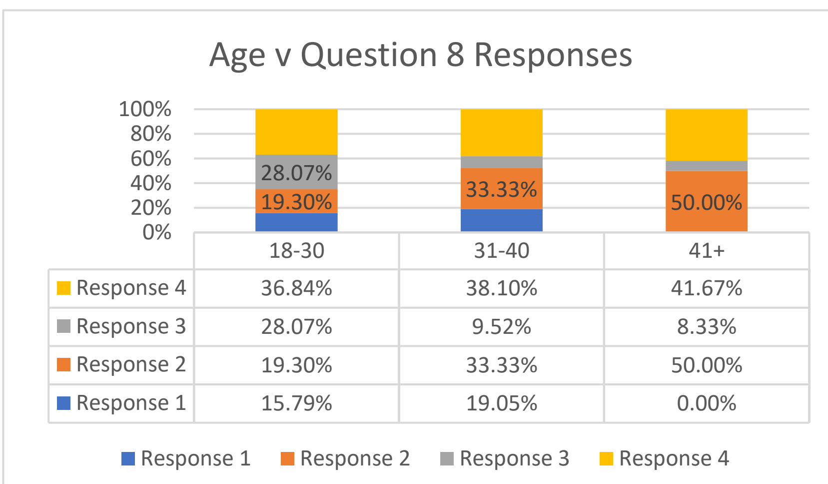

Figure 22: Stacked proportions of the Age v Question 8 contingency table

Tables for Non-Significant Results:

Student v Question 4	Not a Student	Is a Student
Response 1	2	1
Response 2	1	2
Response 3	1	2
Response 4	35	24
Response 5	9	13

Student v Question 6	Not a Student	Is a Student
Response 1	6	4
Response 2	1	2
Response 3	34	25
Response 4	7	11

Age v Question 1	18-30	31-40	41+
Response 1	0	0	0
Response 2	8	3	0
Response 3	2	0	0
Response 4	47	18	12

Age v Question 2	18-30	31-40	41+
Response 1	8	3	1
Response 2	49	18	11

Age v Question 4	18-30	31-40	41+
Response 1	2	1	0
Response 2	3	0	0
Response 3	3	0	0
Response 4	32	17	10
Response 5	17	3	2

Age v Question 5	18-30	31-40	41+
Response 1	11	2	4
Response 2	36	12	6
Response 3	5	4	2
Response 4	5	3	0

Age v Question 6	18-30	31-40	41+
Response 1	8	1	1
Response 2	2	0	1
Response 3	34	15	10
Response 4	13	5	0

Age v Question 7	18-30	31-40	41+
Response 1	14	9	6
Response 2	41	11	6
Response 3	2	1	0

Age v Question 9	18-30	31-40	41+
Response 1	3	1	0
Response 2	19	13	6
Response 3	1	0	0
Response 4	34	7	6

Experience v Question 0	18-30	31-40	41+
Response 1	30	25	21
Response 2	8	4	2

Experience v Question 1	18-30	31-40	41+
Response 1	0	0	0
Response 2	4	6	1
Response 3	0	2	0
Response 4	34	21	22

Experience v Question 2	18-30	31-40	41+
Response 1	4	5	3
Response 2	34	24	20

Experience v Question 3	18-30	31-40	41+
Response 1	7	3	1
Response 2	4	1	4
Response 3	0	3	1
Response 4	27	22	17

Experience v Question 4	18-30	31-40	41+
Response 1	2	0	1
Response 2	2	1	0
Response 3	2	1	0
Response 4	19	22	18
Response 5	13	5	4

Experience v Question 5	18-30	31-40	41+
Response 1	7	4	6
Response 2	24	18	12
Response 3	2	6	3
Response 4	5	1	2

Experience v Question 6	18-30	31-40	41+
Response 1	5	3	2
Response 2	1	2	0
Response 3	23	18	18
Response 4	9	6	3

Experience v Question 7	18-30	31-40	41+
Response 1	9	12	8
Response 2	27	17	14
Response 3	2	0	1

Experience v Question 8	18-30	31-40	41+
Response 1	7	4	2
Response 2	7	6	11
Response 3	9	7	3
Response 4	15	12	7

Experience v Question 9	18-30	31-40	41+
Response 1	3	0	1
Response 2	11	15	12
Response 3	0	1	0
Response 4	24	13	10