Spring 4-26-2024

A Survey of the Murray State University CSIS Department of Student and Instructor Attitudes in Relation to Earlier Introduction of Version Control Systems

Gavin Johnson

Murray State University Honors College

HONORS THESIS

Certificate of Approval

A Survey of the Murray State University CSIS Department of Student and Instructor Attitudes in Relation to Earlier Introduction of Version Control Systems

Gavin Johnson

May 2024

Approved to fulfill the
requirements of HON 437

____

Dr. Jason Owen, Professor
Computer Science and Information Systems

Approved to fulfill the
Honors Thesis requirement
of the Murray State Honors
Diploma

____

Dr. Warren Edminster, Director
Honors College

Examination Approval Page

Author: Gavin Johnson

Project Title: A Survey of the Murray State University CSIS Department of Student and Instructor Attitudes in Relation to Earlier Introduction of Version Control Systems

Department: Computer Science and Information Systems

Date of Defense: 4/26/2024

Approval by Examining Committee:

(Dr. Jason Owen, Advisor) (04/26/2024)

(Dr. David Roach, Committee Member) (04/26/2024)

(Dr. Christopher Mecklin, Committee Member) (04/26/2024)

A Survey of the Murray State University CSIS Department of Student and Instructor Attitudes in Relation to Earlier Introduction of Version Control Systems

Submitted in partial fulfillment

of the requirements

for the Murray State University Honors Diploma

Gavin Johnson

May 2024

Abstract

Over the previous 20 years, the software development industry has overseen an evolution in application of Version Control Systems (VCS) from a Centralized Version Control System (CVCS) format to a Decentralized Version Control Format (DVCS). Examples of the former include Perforce and Subversion whilst the latter of the two include Github and BitBucket. As DVCS models allow software contributors to maintain their respective local repositories of relevant code bases, developers are able to work offline and maintain their work with relative fault tolerance. This contrasts to CVCS models, which require software contributors to be connected online to a main server. Given this expansion in capabilities of productivity under the DVCS model, the Open Source Software (OSS) development scene has transformed accordingly by moving to a DVCS system from the previous CVCS one. Previous work has shown that security breaches and bugs existing on any given application are patched more efficiently, albeit with occasional latency. Accordingly, many software development businesses in the contemporary industry require potential applicants to have knowledge of a VCS model of some sort for employment, with preference given to a DVCS model.

Despite this necessity, many universities, including Murray State University, fail to introduce their students to any VCS system until the last year of their undergraduate experience in Computer Science during their work throughout their senior capstone project. Additionally, many students do not apply a VCS model until their senior capstone project as well. Preliminary literature review has found that students who are permitted to conduct academic work on assignments within the Computer Science profession in their earlier years have been found to have greater engagement (Hsing and Gennarelli, 1,2). They also found community through the VCS model they used and consequently were more motivated to complete course assignments (Hsing and Gennarelli, 2). Additionally, DVCS models, such as github, maintain a log of commits through securing hashing algorithms, preventing students from violating submission standards by modifying logs to give inaccurate submission dates (Lawrence et. al., 2). However, there appears to be a gap in research concerning enthusiasm of Computer Science students specifically to integrate VCS models earlier in their education.

Given the aforementioned benefits of the DVCS model, the goal of this study is to determine the academic benefits of introducing Computer Science students to DVCS models earlier in their education by measuring student and instructor attitudes to doing so. The first part of this study shall consist of obtaining data of student familiarity to VCS models and professor instructors on VCS models through a survey. The second part of this study aims to capture their attitudes towards integrating VCS education earlier in their academic career. The final part of this study shall consist of a regression model to determine enthusiasm of students. This will include VCS instructions earlier in their education based on their year in college and whether or not they have used VCS for software development. Likewise, a second regression model will assess the enthusiasm of professors to incorporate VCS in their lower-level courses will be determined by the amount of years they have taught and the course level they teach.

List of Figures

Table of Contents

I. Introduction

In the current era of software development in the software industry, we may understand it to be optimal that workers be aware of the existence and application of version control systems (VCS) to utilize and to manage various iterations of their code amongst a team. In doing so, they must track their contributions without damaging or “breaking” the main project they are assigned to. Through this process, each software developer may “branch” their code. This capability allows developers to isolate their individual contributions from the collective sum of all work submitted to their current working project. As such, students within Computer Science and Computer Information Systems fields of study may be aware of these models to be prepared to participate in the contemporary digital workforce. The most common version control system students may be introduced to may include GitHub as a Decentralized Version Control System model (DVCS), or possibly BitBucket; however, other models, including Perforce, Mercurial, and Subversion, exist in the format of a Centralized Version Control System (CVCS) model. For the purposes of our study, we decided to conduct a survey amongst the Computer Science and Informations Systems (CSIS) department at Murray State University. As the state of this university's curriculum currently stands, students within our CSIS department are typically not introduced to VCS models throughout their coursework until their senior capstone project course. The intended purpose is to assist in the completion of their chosen and/or assigned project. Therefore, our survey existed in two distinct versions for both students and faculty in the CSIS department. Our survey for students in the CSIS department was composed to determine the familiarity of students on campus of VCS models, their current application of VCS models in their current coursework, their enthusiasm of the proposition for the university to introduce VCS models earlier in their education than their senior year, and their current year of study. For reference, potential options given for their year of study were given on an undergraduate scale, namely, freshman, sophomore, junior, and senior. Research questions given consisted of the following:

R1: Are you currently using a version control system such as Github, Bitbucket, Perforce, Subversion, etc. at any point in your university education?

R2: Are you currently using a version control system such as Github, Bitbucket, Perforce, Subversion, etc. in your current university education?

R3: Answer this question if you have answered Yes to either of the previous two questions. If you have answered No to both of the previous two questions, please leave this question blank. On a scale of 1 to 5, with 1 being not interested at all and 5 being enthusiastic, how enthusiastic would you be in introducing a version control system to CSIS students earlier than their senior year?

R4: What is your current year at Murray State University?

Likewise, professors in the CSIS department of Murray State University were given a similar survey, in which they were inquired of their familiarity of VCS models, their current application of VCS models in their instructional capabilities, their enthusiasm for introducing students in their courses to VCS models earlier than their senior year, and the amount of years they have currently taught thus far as a professor at Murray State University. Research questions included in their survey consisted of the following:

R1: Are you currently using a version control system such as Github, Bitbucket, Perforce, Subversion, etc. at any point in your current profession as an academic instructor at Murray State University?

R2: Have you ever used a version control system such as Github, Bitbucket, Perforce, Subversion, etc. at any point in your current profession as an academic instructor at Murray State University?

R3: Answer this question if you have answered Yes to either of the previous two questions. If you have answered No to both of the previous two questions, please leave this question blank. On a scale of 1 to 5, with 1 not being interested at all and 5 being enthusiastic, how enthusiastic would you be in introducing a version control system to your CSIS students in CSIS courses earlier than their senior year?

R4: How many years have you currently been teaching at Murray State University?

The responses to these questions were collected in two distinct google spreadsheets and quantitatively analyzed for statistical significance for potential conclusions of possible improvement of course curricula insofar as earlier introduction of VCS models to CSIS students than their senior year was concerned.

II. Literature Review

2.1. Application of VCS in Software Industry (Github and Bitbucket)

Version control systems have been utilized throughout the software development industry, with Git and Github recording approximately 6.8% of respondents to a poll conducted in Eclipse foundation having reported as using Git to manage their projects, 12.6% reported having using Centralized Version Systems (CVS), 3% reported having used Mercurial, and 58.3% reported having used Subversion (Lawrence et. al, 2013). Git and Github in particular were reported in August of 2013 having over 3 million contributors to its overall systems, with 7 million repositories created (Lawrence et. al., 2013). As of January 25, 2023, Github is reported to have over 100 million users contributing to its software base (Dohmke, 2023). BitBucket, another DVCS model for companies, has reported 10 million users existing on the cloud for their services and over 28 million repositories having been created through their VCS model (Yap, 2019). By these numbers, we may understand DCVS models to be on the rise in comparison to CVCS models in the contemporary digital age insofar as the software development industry is concerned.

2.2. Evolution from CVCS to DVCS

Over the previous 20 years, both companies and independent software developers have seen a transition from CVCS models to DVCS models in their software development workflow. Examples of CVCS models include Perforce and Subversion while examples of DVCS models include Github, Mercurial, and Bitbucket. We here discuss main differences between DVCS and CVCS models.

In the case of CVCS models, software developers contributing to a collective software application, otherwise known as "contributors", or "committers", are given authorization to contribute to the mainline of the software application in question (Alwis & Sillito, 2009). The mainline is the main collection of code to which all contributors submit their individual code contributions to, and may download, or “pull”, the collective software contributions therefrom. This collection of code is otherwise known as a “repository”. Under this model, each contributor must both be connected online to the main server, and must be given administrative access to contribute to the mainline. Latency in communication with the code mainline may become apparent if software developers lose their internet connection or cannot otherwise contribute to the mainline. Software exists on a central mainline server. If the mainline server crashes, all work made on the software project in question shall be lost as well. Additionally, if contributors submit code that may violate standards of encapsulation in software construction, the main software application may break.

In the case of DVCS models, software developers contributing to a collective software application may create their own distinct versions of the code with contributions or modifications to the mainline, though these contributions and/or modifications exist outside the mainline in separate versions, otherwise known as “branches”. Contributors may create their own copies of the mainline software application, which may be downloaded and pulled from, otherwise known as “forks”. Should contributors complete a feature on their respective branch, they may include their contributions to the mainline after first determining their software contributions do not interfere with the reliability of software code in the mainline. This process of submitting a branch into the mainline branch for software contribution is otherwise known as “merging” a branch. Benefits of the DVCS model include remote access to independent branches and/or forks of the mainline repository (Alwis & Sillito, 2009). In other words, unlike the CVCS model, software contributors do not have to be connected to the main server nor connected online to commit code to their respective branch and/or fork. Given this capacity for DVCS models to allow for the formation of remote servers and remote software reduplication, this allows for maintaining access to software application code in the event the mainline server may crash and work made on said software application may be lost.

2.3. Impact of DVCS on OSS Environment

Given the fact that many DVCS models, such as Github and Bitbucket, currently exist to assist in the development of Open Source Software, we must further understand the implications of the Open Source Software (OSS) scene on the contemporary software development industry. In doing so, we may further elaborate on the implications of educating our students in the applications of VCS models in their future career aspirations. This fact becomes apparent when we take into account that 77% of respondents in the annual OSS survey reported that they have relied more upon the application of OSS within their respective places of work (Initiative, 2022). The rise of OSS applications, such as

Kubernetes, as well as other applications, such as React.js, Blender, and Node.js may attest to this advancement in the current industry-wide implementation of OSS in the production process.

When understanding the implementation of OSS in today's companies, we must understand the potential benefits a company may gain in choosing to open-source their software. By releasing the source code of their respective projects to the general public for consumption and development. In doing so, a company may push an application to the general public as a potential industry standard, or perhaps as a potential choice of a software development tool. Their incentive in participating in this practice lies in their gain of reputation for their development in said tool, thereby improving their brand name in the market. Furthermore, a social concept known as "coopetition" may occur. This phenomenon is defined as one in which competing companies may cooperate in the production process to solve a common problem, though may compete on varying features of products resulting from potential solutions to said problem (Guizani et. al., 2023). Finally, a company may wish to open-source applications it may care about, such as those it may utilize in its production process in more common software applications. Therefore, companies wishing to engage in the OSS environment should ideally seek to construct an "abandonment plan" for its most-used software applications, wherein said applications are released as “abandonware” by being open-source to the general public for universal use and development (Guizani et. al., 2023).

Through OSS development, which DVCS models may facilitate, the current software industry as a whole may drive innovation in all regions of software development, and, consequently, may become a mainstay in future trends in the practice of software engineering in its entirety. It is plausible that current students within fields adjacent to software engineering may look to improve their respective knowledge in DVCS models, such as github, to potentially contribute to these practices.

2.4. Improvement of Security Vulnerability Resolution by DVCS in OSS

In addition to assisting in the production process and maintenance of software applications, DVCS models, such as Github, may be utilized to draw attention to security issues in a project, as well as provide potential concrete software solutions to them through the Open Source Software (OSS) community. A key example of this may best be observed through the implementation of the Github Issues feature, which allows developers and consumers of any given OSS application alike to post security issues they may currently have in regards to the application of said OSS product, as well as openly provide solutions to said security issues. As of 2019, Github had tracked over 40 million users in its collective application of its services, as well as reported over 20 million security issues pertaining to its hosted software products in the OSS environment (Github, 2019). In accordance with our previous observations that Github has since surpassed a collective total of over 100 million users in consumption of its services, we may best understand this feature of DVCS models to potentially provide a net benefit to future years in the software development industry for providing aid to safety concerns through current and potential security failures.

Throughout a survey conducted on November 4, 2020, in which over 249043 security issues submitted through Github repositories were analyzed. When analyzed in the broader context of all issues raised in relevant repositories in the recorded snapshot of date specified, approximately 1.4% of all issues raised within github repositories recorded were related to security concerns (Buhlmann & Ghafari, 2022). The remaining concerns were non-security related, i.e. involved either in cryptography or identity management. Additionally, with few exceptions, the majority of security issues raised held a maximum of 3 comments, with the majority being resolved with no comments posted in github forums between software developers. Furthermore, security issues took a relatively long time to note in github issues at the initial creation of a Github repository, with the median time taken to raise the first security issue on recorded repositories recorded at a median of 655.50 days (Buhlmann & Ghafari, 2022). These issues were typically resolved, on average, 53.17 days after their first notice granted in github issues (Buhlmann & Ghafari, 2022). Resolved security issues were also analyzed to have been solved in repositories that appeared to have gained more traffic in popularity through the star system, in which github repositories may be given a star to denote approval in application and development of end users. These facts may point to the implementation of security maintenance and resolution through DVCS models through software development workflow in the future environment of the software development industry. We understand this, as previous qualitative research has shown that, while contributors in a DVCS model, such as Github, may be able to modify proposed unit test codes to the mainline, their granted access in doing so may be restricted by the broader community (Wermke et. al., 2022).

2.5. Previous Exposure to CSIS and Non-CSIS Students of DVCS Models

Given the capability of DVCS models to host information to collaborators in an open-source manner, these models hold purposes for those outside of the software development field as well as software developers. For examples of this application, we shall once again look to Github as our prime DVCS model. Within Github, users may compose ReadMe files, which are known as text-based files that may include documentation of repository software code, as well as text for other purposes. As such, ebooks, novels, and texts of other formats are hosted on Github by authors and other artists alongside software code by software developers. Therefore, we may take into account potential benefits for users of DVCS models who are not within standard CSIS fields.

2.6. Beneficial Improvements of Introduction of Students to DVCS Models

To further illustrate this phenomenon, we shall observe a study conducted in 2019 of Computer Science students who were assigned to complete a significant software project, using Github as a VCS to manage their collective code along the way. Through this study, students who made use of Github were found to be more involved in the project as opposed to those who did not use Github as their VCS model. Additionally, students across all teams were able to engage in social interpersonal and inter team activities, such as starring one another's repositories. From this phenomenon, we may further understand students who individually made contributions to team software output using Github as their VCS model were more likely to feel as though their contributions made an impact on the overall quality of the software product developed by their team than those who did not use Github as a VCS model (Hsing & Gennarelli, 2019). Through these social developments, we may analyze the potential benefits of introducing Github as a DVCS model to students not only within CSIS academic fields, but all other fields as well. Potentially, doing so may lead the way for students in creative fields to share ideas for potential artistic projects on an individual and collective basis.

III. Methodology

3.1. Overview

To assess current academic trends on campus, and to determine social trends for academic curriculum in the Computer Science (CSC) department going forward, we conducted a survey throughout the CSIS department of Murray State University. Responses of said survey were collected for statistical analysis to determine student and faculty familiarity of VCS models currently on a personal basis, familiarity and application within their coursework, and enthusiasm of introducing VCS models to students earlier than their senior year. Each of these two surveys were hosted via Google Forms, and responses from each survey were generated into Google Spreadsheets. Each Google Spreadsheet was downloaded to the author's local personal computer for statistical analysis. Multiple linear regression analysis was utilized to determine the statistical significance of predictor variables on chosen response variables for each dataset. In both the CSIS student response dataset and the CSIS faculty response dataset, enthusiasm was chosen as our response variable to determine student and instructor attitudes on campus towards the idea of introducing VCS models to students earlier than their senior year. However, in the case of CSIS student responses, whether the student had ever learned a VCS model previously, whether the student is currently learning a VCS model in their courses now, and their current year of study on campus were selected as predictor variables. Choices for the year of study of surveyed students included freshman, sophomore, junior, and senior. In the case of CSIS faculty responses, whether the instructor had ever used a VCS model throughout their instructive capacities on campus, whether they were currently using a VCS model in their academic teaching capacities currently, and their academic tenure in years were selected as predictor variables. Response data from each of these two datasets were collected and analyzed separately in two different programs using R Studio version 4.2.2 (R Core Team, 2022).

3.2. Null Hypothesis

To test for statistical significance of each predictor variable's impact on the response variable, both data sets were examined. Multiple linear regression models were fit by selecting “Enthusiasm” as the response variable, and all other variables of each of the two datasets as the predictor variables. This was conducted through execution of the linear model, or, “lm” method used within R. From here, we determined the statistical significance of each predictor variable by using a level of significance $\alpha = 0.05$. Additionally, an Analysis of Variance (ANOVA) table was obtained for each multiple linear regression model. To determine statistical significance of each predictor variable chosen, we may state our null hypothesis such that each predictor variable chosen for each model taken has no correlation with the degree of enthusiasm each survey respondent held of the idea of introducing VCS models, such as Github, to students in the CSIS program at Murray State University earlier. To clarify, that means we will test H_o: $\beta_1=0$, or that the slope coefficient is zero. Likewise, in the dataset composed of instructor responses within the Murray State University CSIS department, whether the instructor had ever used a VCS model throughout their instructive capacities on campus, whether they were currently using a VCS model in their academic teaching capacities currently, and their academic tenure in years holding no statistically significant impact on their enthusiasm towards the idea of introducing VCS models to their students earlier than said student’s senior year was selected to be the null hypothesis for our study in relation to the dataset composed of instructor responses within the Murray State University CSIS department.

3.3. Application of R and Applied Libraries

To conduct a statistical analysis of all collected response data from both datasets, RStudio was used. Particularly, RStudio was used to conduct the statistical analysis for this study. We made use of the car, tidyverse, GGally, mosaic, and readxl packages in R to conduct this study. A detailed explanation of each package follows:

Car - This package was used to assist with constructing a Type II ANOVA table from our multiple linear regression model in this project (Fox & Weisberg, 2019).

Tidyverse - This package is a collection of several packages that exists to assist in the process of visually graphing, manipulating, and tidying our data. Packages that are part of the 'tidyverse' overload 'dplyr' for data manipulation and 'ggplot2' for visualization (Wickham et. al., 2019).

GGally - This package exists to extend graphical capabilities provided in 'ggplot2' for our statistical output data (Schloerke et. al., 2021).

Mosaic - This package implements high-level systems for working with statistical models. The main interest in this study was to use formula-based syntax for statistical modeling (Pruim et. al., 2017).

Readxl - This package exists to assist us in loading a spreadsheet of data from our local machine into the R programming language. We used it to import each of the two downloaded spreadsheets from Google spreadsheets consisting of each survey's distinct response data for students and instructors within the Murray State University CSIS department (Wickham & Bryan, 2023).

3.4. Process of Data Collection

For our study, we conducted two distinct surveys. The first considered student attitudes in the CSIS department on campus from their previous academic and personal experiences using VCS models, as well as their experience in the department on campus. The second survey considered both previous and personal experiences in using VCS models, as well as their experience in educating students on campus in accordance with their respective academic tenure. Surveys were composed on Google Forms and a composite URL link for each survey was sent throughout the CSIS department via email for both students and instructors. Survey participants were instructed to choose one survey in accordance with their role as student or academic instructor in the CSIS department. Survey responses were collected between Monday, March 26, and Thursday, April 4, 2024. Survey response data from each survey was generated and loaded into a Google spreadsheet for responses from each survey, then downloaded to my local machine for statistical analysis. Sample size of student survey participant response data was n = 38. Sample size of instructor survey participant response data was n = 3.

3.5. Procedure

In the case of CSIS student response data, our data was loaded to the local machine through our included readxl package. Additionally, questionnaire question text was assigned variable names. Whether the student was assigned to learn VCS models throughout their current academic courses was labeled “LearningVersionControl”. Whether the student is currently or has ever used a VCS model was labeled “UsingVersionControl”. Their enthusiasm at the idea of potentially being introduced to VCS models earlier than their senior year was labeled “Enthusiasm”. Finally, their year of study on campus was labeled by the variable name “Year”. Seeing that we have no need for the date nor time in which any survey participant responded to our survey, we drop all mention of the date and time in which each survey participant responded to our survey by using the “dplyr::select” method from the tidyverse package. We then used the “ggpairs” method to generate visual output data from the correlation for each predictor variable on our response variable, which in this case is enthusiasm. In the case of the enthusiasm variable, we visually determined the proportion of student survey respondents who noted they would be interested in being introduced to a VCS model earlier in their education.

Given that the graph of response data appears to approach a high proportional value near 5, we may assume the majority of students are interested in being introduced to VCS models earlier in their education. We performed the ggpairs method on the graph of correlation between our selected “Enthusiasm” response variable and each predictor variable separately for visual purposes. Our first graph depicting the correlation between the response variable and our “LearningVersionControl” predictor variable, in which student respondents recorded their current application of VCS models in their courses or lack thereof, is presented as follows:

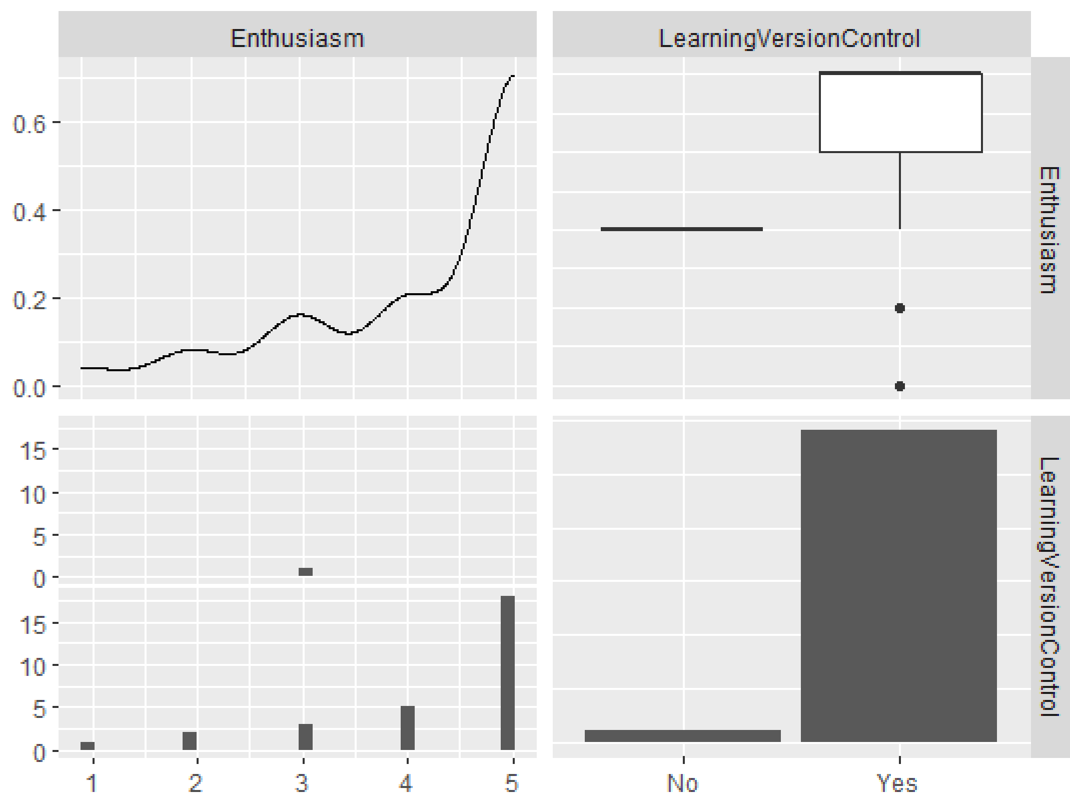


Figure 1: Enthusiasm vs LearningVersionControl

Our second graph depicting the correlation between our selected response variable of "Enthusiasm" in relation to our predictor variable titled "UsingVersionControl", in which students respond they have ever used a VCS model previously or not, is presented as follows:

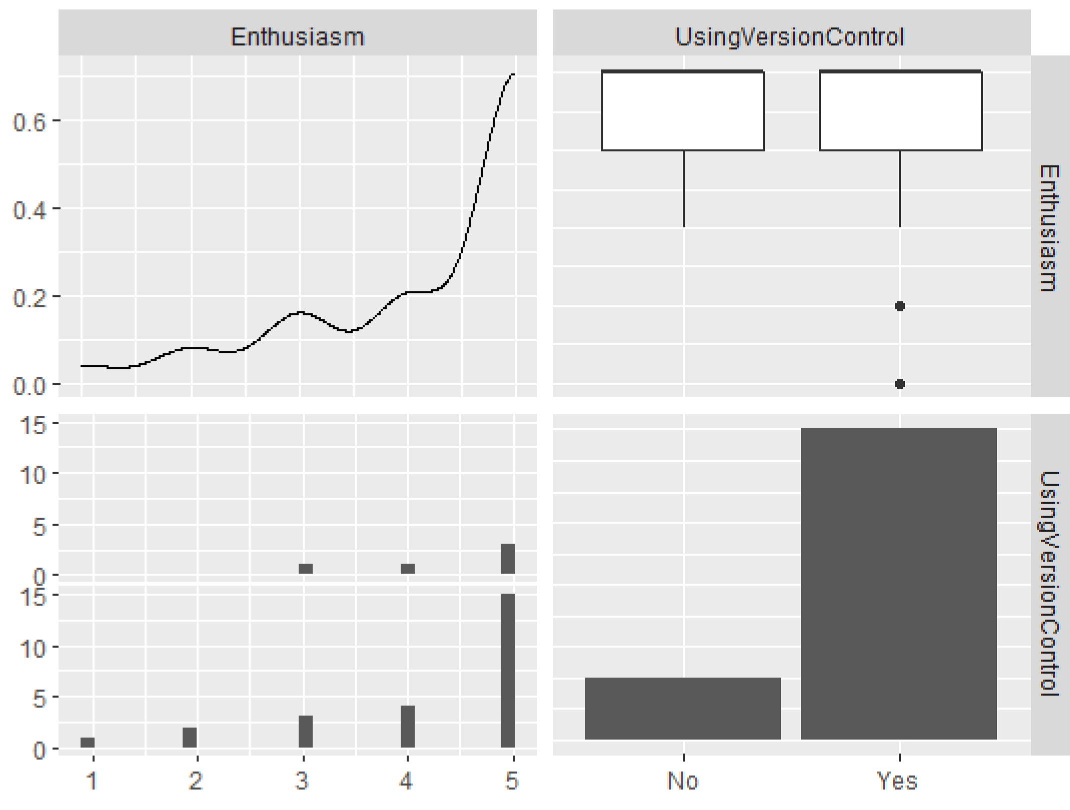


Figure 2: Enthusiasm vs UsingVersionControl

Finally, we present our final visual output, depicting the correlation between our selected response variable of “Enthusiasm”, and the year of study of each student surveyed:

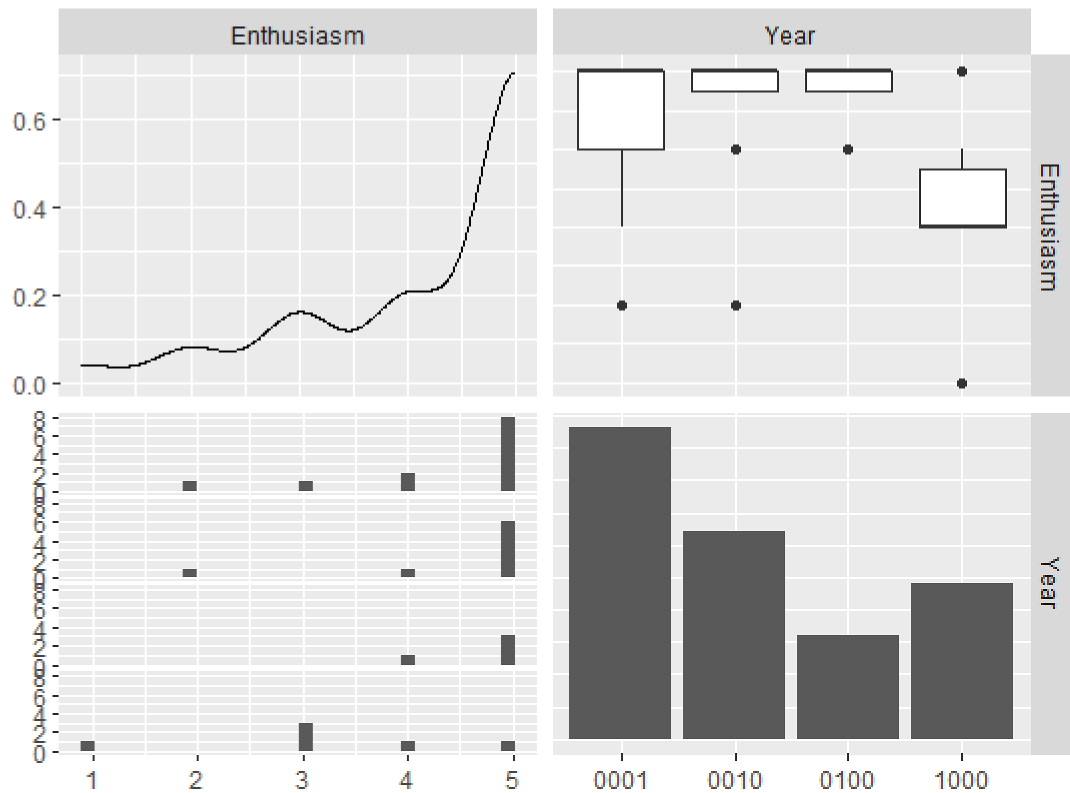


Figure 3: Enthusiasm vs Year

From here, we then create our linear regression model from the student survey response data. As previously mentioned, we select "Enthusiasm" as our response variable. We then select "LearningVersionControl", "UsingVersionControl", and "Year" as our predictor variables for our multiple linear regression model. Taking the summary of this model, we find the following statistical data from our student survey response dataset:

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	3.0000	0.9986	3.004	0.00614	**
LearningVersionControlYes	0.9387	1.1474	0.818	0.42134	
UsingVersionControlYes	-1.2312	0.5773	-2.133	0.04339	*
YearJunior	1.6386	0.5910	2.773	0.01058	*
YearSenior	1.6065	0.5621	2.858	0.00867	**
YearSophomore	2.0425	0.7086	2.883	0.00819	**

Figure 4: Summary of CSIS Student Response Data

According to the summary of our multiple linear regression model, we find the enthusiasm of students in the CSIS department of Murray State University towards the idea of introducing VCS models in our data to be found by the equation:

$$\text{Enthusiasm} = 3 + 0.9387\text{LearningVersionControlYes} - 1.2312\text{UsingVersionControlYes} + 1.6386\text{YearJunior} + 1.6065\text{YearSenior} + 2.0425\text{YearSophomore}$$

We find LearningVersionControlYes, or whether or not each student has learned VCS models previously, was not statistically significant in predicting their enthusiasm for being introduced to VCS models earlier in their education than their senior year (t=0.816,df=24,p=0.421). The variable UsingVersionControl is significant (t=-2.133,df=24,p=0.043). Since the year in school is a categorical variable with 4 levels, 3 indicator variables were used, with first-year as the baseline.

An ANOVA table of the model is displayed. The year was statistically significant (F = 3.616, df = 3, 24, p=0.028):

```
Anova Table (Type II tests)

Response: Enthusiasm
                        Sum Sq Df F value  Pr(>F)
LearningVersionControl  0.6674  1  0.6693 0.42134
UsingVersionControl     4.5349  1  4.5478 0.04339 *
Year                   10.8182  3  3.6164 0.02763 *
Residuals              23.9318 24
```

Figure 5: ANOVA Model Taken of CSIS Student Response Data

From here, we then conducted the same process of study on our dataset composed of responses from instructors within the CSIS department of Murray State University. As we only have 3 responses, we must first drop all responses in which survey participants did not currently use and have never used a VCS model. Unfortunately, this appeared to be the case for 2 of the 3 responses we received from our CSIS instructor dataset. Consequently, we were left with only 1 data point in which an instructor from the CSIS department of Murray State University had either previously used or is currently using a VCS model in their day-to-day academic capabilities in instructing students within their courses. According to the dataset value, one instructor had both used VCS models in the past and is currently using VCS models to educate their students in their courses currently. This instructor responded they have instructed students for 9 years and rated a 4 out of 5 on the enthusiasm scale.

IV. Findings

4.1. Report on Overall Multiple Linear Regression Analysis Findings and Potentially Statistically Significant Variables

In the case of our dataset composed of responses from the CSIS student survey respondents, our multiple linear regression model appears to produce the following regression equation:

Enthusiasm = 3 + 0.9387LearningVersionControlYes - 1.2312UsingVersionControlYes + 1.6386YearJunior + 1.6065YearSenior + 2.0425YearSophomore

This means that, if the student responded that they had learned VCS models in the past, they were, on average, statistically likely to score 0.9387 more points of enthusiasm. If they are currently using a VCS model, they were projected to score 1.2312 less enthusiasm points. Additionally, if they were a junior student, they were projected to score 1.6386 more enthusiasm points. If they were a senior, they were projected to score 1.6065 more enthusiasm points. Finally, if they were a sophomore, they were projected to score 2.0425 enthusiasm points.

Furthermore, according to our multiple linear regression model in the case of response data from CSIS student survey participants, whether or not the student survey had learned VCS models in the past was found not to have a statistically significant impact on their enthusiasm towards the idea of being introduced to VCS models earlier in their education than their senior year. Contrarily, whether or not said students were being introduced to VCS models in their courses currently as well as their current respective year of study was found to have a statistically significant impact on their enthusiasm towards the idea of being introduced to VCS models earlier.

Unfortunately, we were not able to collect enough information from instructors in the CSIS department of Murray State University to perform a multiple linear regression analysis for their aforementioned predictor variables on their enthusiasm towards our proposal. As only 1 of the 3 instructors had either previously used a VCS model or are currently using a VCS model in their courses, we did not have enough data to determine statistical significance from the data provided in their respective dataset.

V. Conclusion

5.1. Summary

According to our findings, we were able to determine potentially significant factors in considering the enthusiasm of the student body of Murray State University in regards to being introduced to VCS models in their CSIS education earlier than their scheduled senior year. Namely, we were able to determine that their current involvement in VCS model applications in their courses they are currently taking as well as their current year of study may play a role in their degree of enthusiasm towards our proposal. Potential causes may be student familiarity may breed excitement or reluctance in application of previous VCS models. Additionally, students may be in the process of participating in student internships, jobs, and/or early research opportunities that may necessitate familiarity with and application of VCS models. As we did not have sufficient data to conclude statistical significance of scheduled predictor variables in CSIS instructor dataset, we were not able to determine potential factors that may contribute to enthusiasm or lack thereof of faculty on campus towards the idea of introducing VCS models to their students earlier than their senior year.

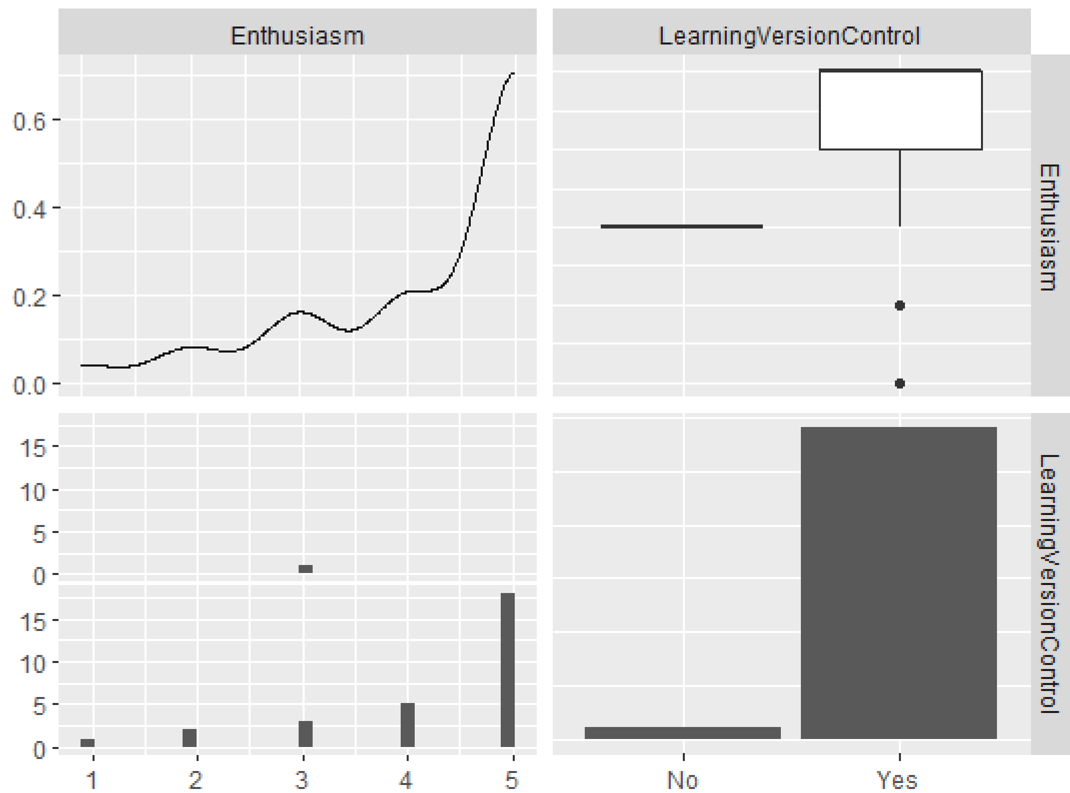
5.2. Future Work and Potential Improvements on Study

Potential improvements and modifications that could be made to our study may involve adding various predictor variables to our survey beyond those pertaining strictly to year of study and familiarity with VCS models. Namely, future surveys could inquire of students their prior career history in the software industry, as well as their possible previous experiences in early research opportunities. Other possible alternative survey questions could include previous student extracurricular opportunities within STEM fields, as well as previous Software Development hobbyist activity experiences, such as previous involvement in hackathons. Furthermore, future replications of our survey could expand the sample size of both datasets, especially those pertaining to the instructors within the CSIS program, to determine from a multiple linear regression model the statistical significance of proposed predictor variables.

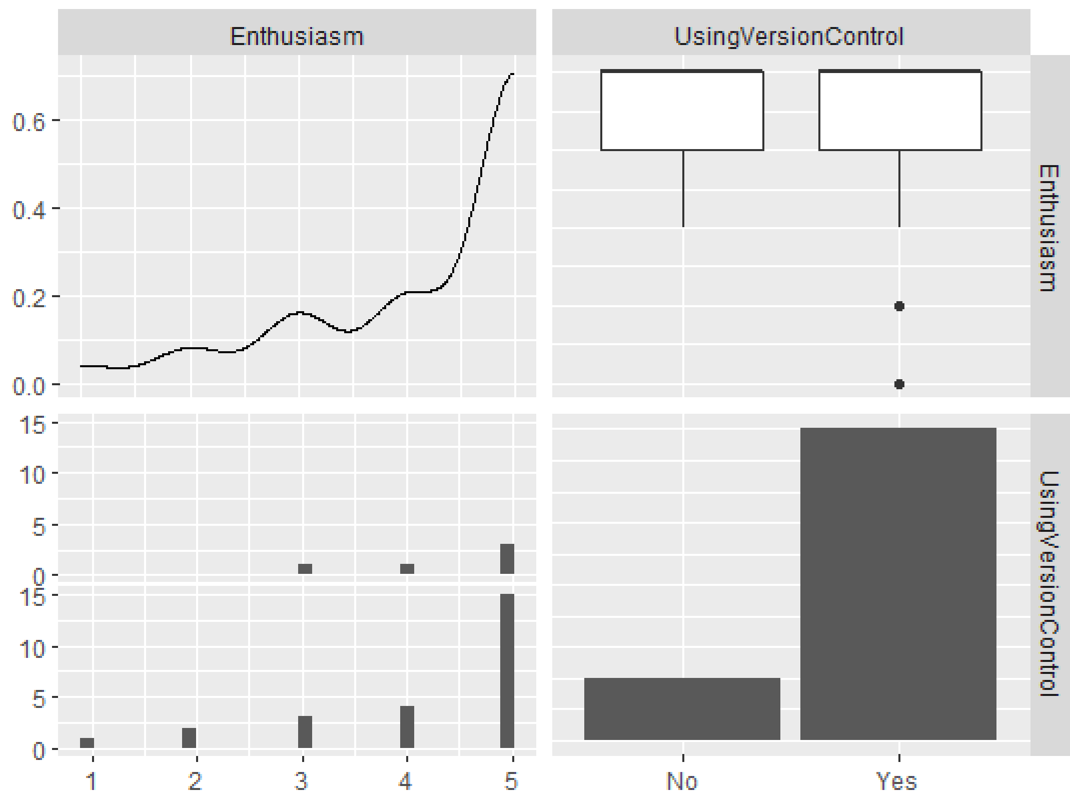
Appendix

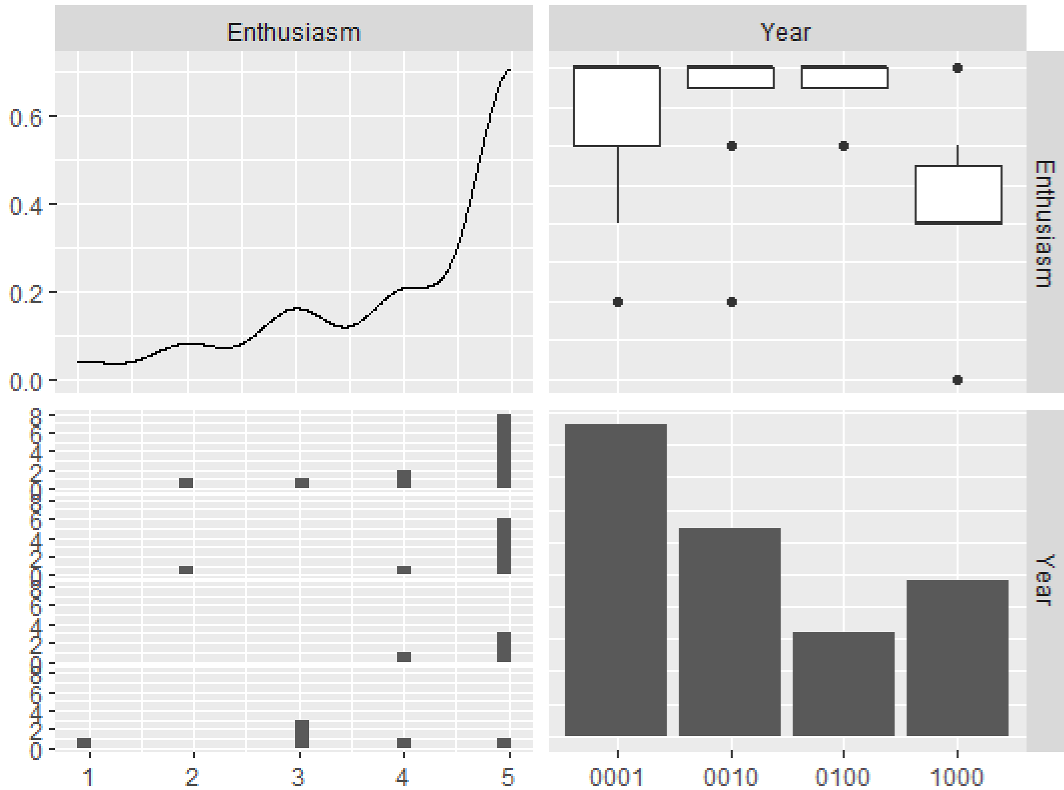
A. Graphical Representation of Collected CSIS Student Response Dataset in Correlation Between Selected Response and Predictor Variables

a. Graphical Representation of Enthusiasm vs LearningVersionControl from CSIS Student Response Dataset



b. Graphical Representation of Enthusiasm vs UsingVersionControl from CSIS Student Response Dataset





c. Graphical Representation of Enthusiasm vs Year from CSIS Student Response Dataset

B. Summary & ANOVA Output

a. CSIS Student Survey Response Summary Data:

```
Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                 3.0000     0.9986   3.004  0.00614 **
LearningVersionControlYes   0.9387     1.1474   0.818  0.42134
UsingVersionControlYes     -1.2312     0.5773  -2.133  0.04339 *
YearJunior                  1.6386     0.5910   2.773  0.01058 *
YearSenior                  1.6065     0.5621   2.858  0.00867 **
YearSophomore               2.0425     0.7086   2.883  0.00819 **
```

b. CSIS Student Survey Response ANOVA Data:

```
Anova Table (Type II tests)

Response: Enthusiasm
                       Sum Sq Df F value  Pr(>F)  
LearningVersionControl  0.6674  1  0.6693 0.42134  
UsingVersionControl     4.5349  1  4.5478 0.04339 *
Year                   10.8182  3  3.6164 0.02763 *
Residuals              23.9318 24                  
```

C. R Code

a. R Code for CSIS Student Response Data

```
require(car)
require(tidyverse)
require(GGally)
require(mosaic)
require(readxl)

responseData <-
read_excel('C:\\Users\\gavin\\Downloads\\StudentResponseData.xlsx')

colnames(responseData) <-
c("Date","LearningVersionControl","UsingVersionControl","Enthusiasm","Year"
)

responseData <- dplyr::select(responseData,-Date)

#responseYearModel <- data_unite(responseData,new_column =
"Year",select=c("Sophomore","Junior","Senior"))
```

```
responseData <- responseData %>%
  drop_na()



ggpairs(responseData,columns=c("Enthusiasm","LearningVersionControl"))
ggpairs(responseData,columns=c("Enthusiasm","UsingVersionControl"))
ggpairs(responseData,columns=c("Enthusiasm","Year"))

happyModel <-
lm(Enthusiasm~LearningVersionControl+UsingVersionControl+Year,data=response
Data)

summary(happyModel)
Anova(happyModel)
```

b. R Code for CSIS Instructor Response Survey Data:

```
require(car)
require(tidyverse)
require(GGally)
require(mosaic)
require(readxl)

responseData <-
read_excel('C:\\Users\\gavin\\Downloads\\InstructorResponseData.xlsx')

colnames(responseData) <-
c("Date","CurrentUsing","HaveUsed","Enthusiasm","YearsTaught")

responseData <- dplyr::select(responseData,-Date)

responseData <- responseData %>%
  drop_na()
ggpairs(responseData)
```

happyInstructorModel <-

lm(Enthusiasm~CurrentUsing+HaveUsed+YearsTaught,data=responseData)

Works Cited

Bühlmann, N., & Ghafari, M. (2022, April). How do developers deal with security issue reports on github?. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing* (pp. 1580-1589).

De Alwis, B., & Sillito, J. (2009, May). Why are software projects moving from centralized to decentralized version control systems?. In *2009 ICSE Workshop on Cooperativ*

Fox J, Weisberg S (2019). An R Companion to Applied Regression, Third edition. Sage, Thousand Oaks CA. https://socialsciences.mcmaster.ca/jfox/Books/Companion/

Guizani, M., Castro-Guzman, A. A., Sarma, A., & Steinmacher, I. (2023, May). Rules of Engagement: Why and How Companies Participate in OSS. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)* (pp. 2617-2629). IEEE.

Kelleher, J. (2014, January). Employing git in the classroom. In *2014 World Congress on Computer Applications and Information Systems (WCCAIS)* (pp. 1-4). IEEE.

Lawrance, J., Jung, S., & Wiseman, C. (2013, March). Git on the cloud in the classroom. In *Proceeding of the 44th ACM technical symposium on Computer science education* (pp. 639-644).

O. S. Innitiative, “Ten takeaways from the 2022 state of open source survey,” 2022, [Online; accessed 12-Aug-2022]. [Online]. Available: https://opensource.org/ten-takeaways-from-the-2022-state-of open-source-survey

Perez De Rosso, S., & Jackson, D. (2013, October). What's wrong with git? A conceptual design analysis. In *Proceedings of the 2013 ACM international symposium on New ideas, new paradigms, and reflections on programming & software* (pp. 37-52).

R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/

R. Pruim, D. T. Kaplan and N. J. Horton The mosaic Package: Helping Students to ‘Think with Data’ Using R (2017). The R Journal, 9(1):77-102.

Schloerke B, Cook D, Larmarange J, Briatte F, Marbach M, Thoen E, Elberg A, Crowley J (2021). GGally: Extension to ‘ggplot2’. R package version 2.1.2, https://CRAN.R-project.org/package=GGally

Wermke, D., Wöhler, N., Klemmer, J. H., Fourné, M., Acar, Y., & Fahl, S. (2022, May). Committed to trust: A qualitative study on security & trust in open source software projects. In *2022 IEEE Symposium on Security and Privacy (SP)* (pp. 1880-1896). IEEE.

Wickham H, Averick M, Bryan J, Chang W, McGowan LD, Francois R, Grolemund G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V, Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). "Welcome to the tidyverse." Journal of Open Source Software, *4*(43), 1686. doi: 10.21105/joss.01686. https://doi.org/10.21105/joss.01686

Wickham H, Bryan J (2023). Readxl: Read Excel File. R package version 1.4.3, https://CRAN.R-project.org/package=readxl

Zolkifli, N. N., Ngah, A., & Deraman, A. (2018). Version control system: A review. *Procedia Computer Science*, *135*, 408-415.