Milestone 3 - Report

Authors: Patrick Tung, Paul Vial, and Mengda (Albert) Yu

Date: 2019/04/12

Contents

1.0 Introduction

As the Master of Data science program is soon to end, we all like to reflect on the courses we have taken. Some courses we took were difficult and some were relatively easy, but the true question is, how was this affected by our prior experience. DSCI 512 is a programming and algorithms course in the MDS program at UBC which introduces fundamental algorithms such as sorting and searching, as well as data structures. This project is to analyze whether the level of programming experience prior to the MDS program affects an MDS student's self-perceived difficulty of DSCI 512 materials.

Question: Does their level of programming experience prior to the MDS program influence a person's self-perceived difficulty of DSCI 512 (Algorithms and Data Structures)?

We began with defining a null hypothesis and an alternative hypothesis, as shown below.

> **Null hypothesis:** The level of programming experience prior to the MDS program does not influence a person's self-perceived difficulty regarding DSCI 512.
>
> **Alternative hypothesis:** The level of programming experience prior to the MDS program influences a person's self-perceived difficulty regarding DSCI 512.

After extensive brainstorming, we decided that the variables we believe that are important are:

- Previous programming experience
- Sex
- Mathematics skill level
- Whether or not a student has friends or family with programming experience

Variable	Name	Type	Description
Confounder	`sex`	category	Female or Male
Confounder	`math_skill`	ordinal	Self-reported Math skills (Below average, Average, Above average)
Confounder	`friend_with_prog`	category	Friends who have jobs associated with programming (No, Yes)
Main Covariate	`prog_exp`	ordinal	Previous programming experience prior to the MDS in hour (None, Less than 100 hours, Less than 1000 hours, More than 1000 hours)
Outcome	`difficulty`	ordinal	Self-perceived difficulty (Easier than average, Average, More difficult than average)

Figure 1:

2.0 Data description

To gather the data, we created a survey and collected 56 observations from our fellow MDS students, DSCI 554 TAs, and lab instructor for self-perceived difficulty of the DSCI 512 course.

Table 1: Surveyed Variables

3.0 Key EDA

First, we did some preliminary investigations to understand the data and discover important patterns.

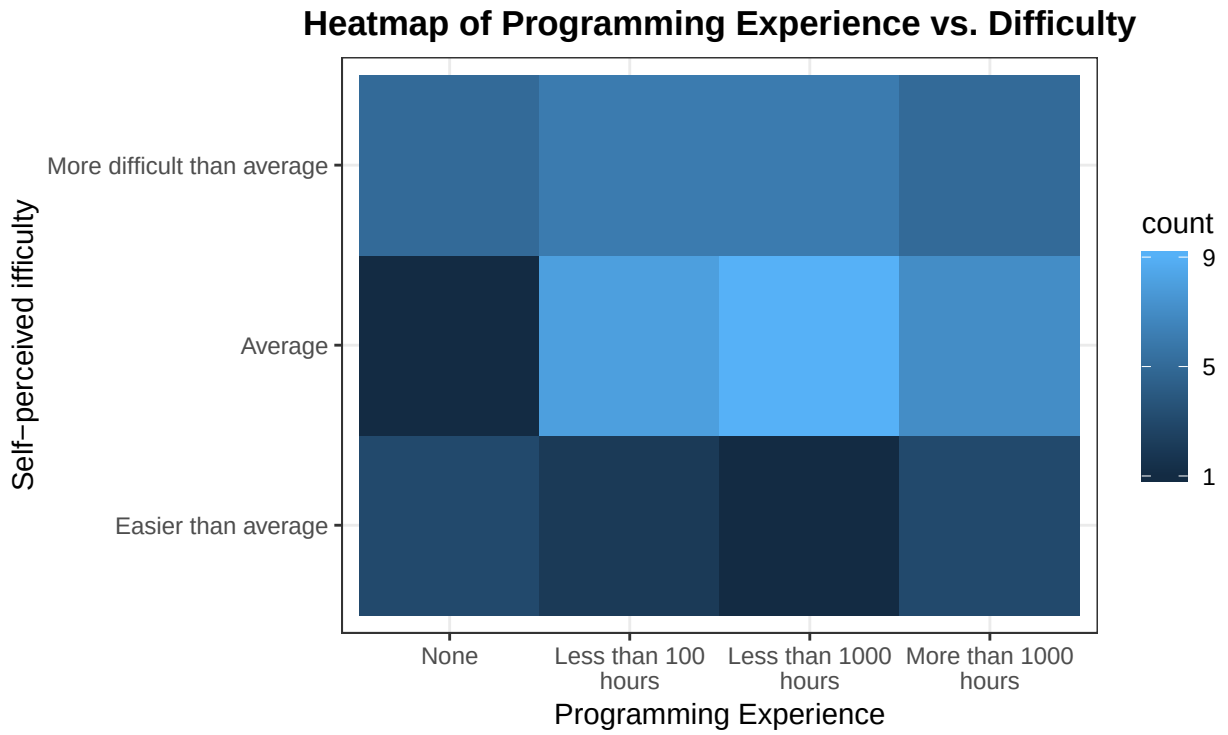

Figure 1.

It can be seen that the number of students who have been experiencing a harder time in DSCI 512 is greater than the number of students who found the course easier than average.

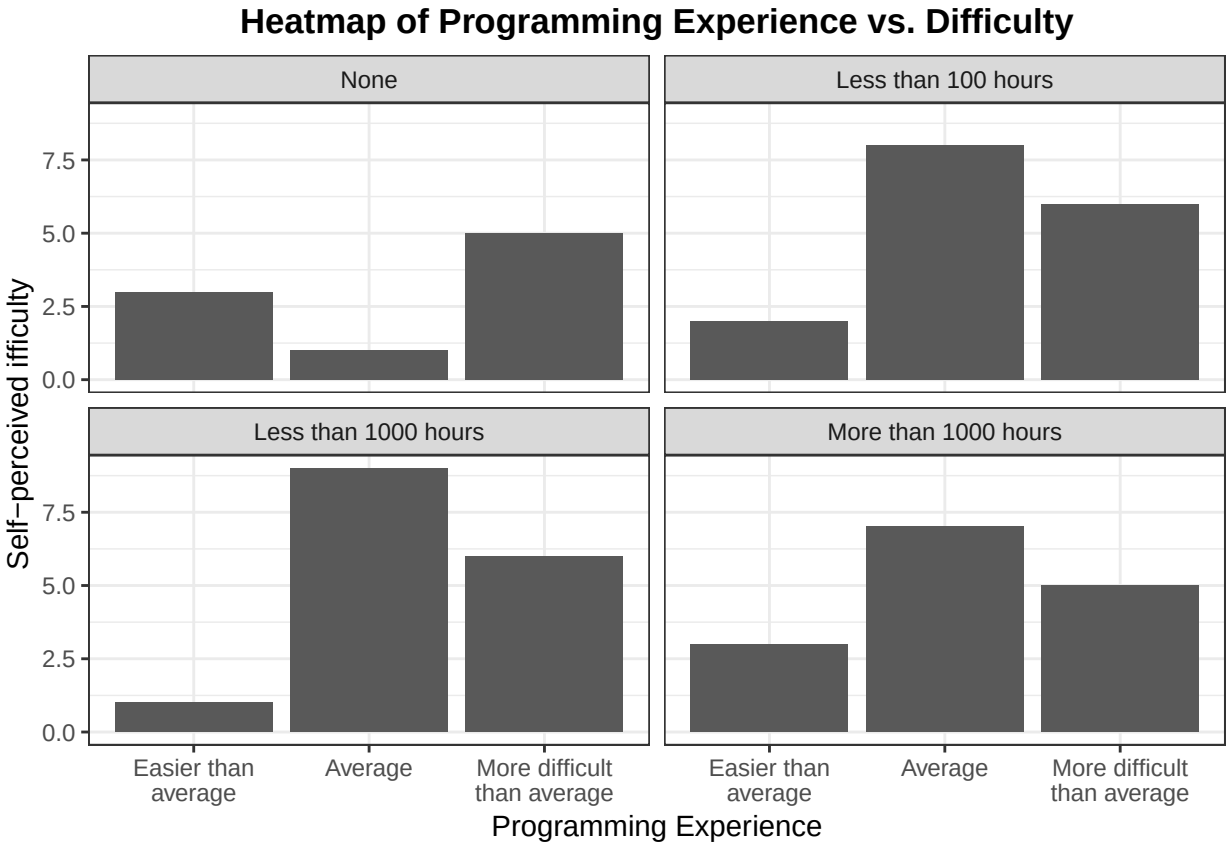

Figure 2.

The group without programming experience had the greatest proportion of people that found the course to be difficult. It makes sense that if students have no programming experiences, they are more likely to struggle with assignments and tests. It is also interesting to note that the most commonly reported level of difficulty was "Average" across the three other groups (less than 100, less than 1000 hours, More than 1000 hours), and that relatively few students found the course to be more difficult or less difficult than average.

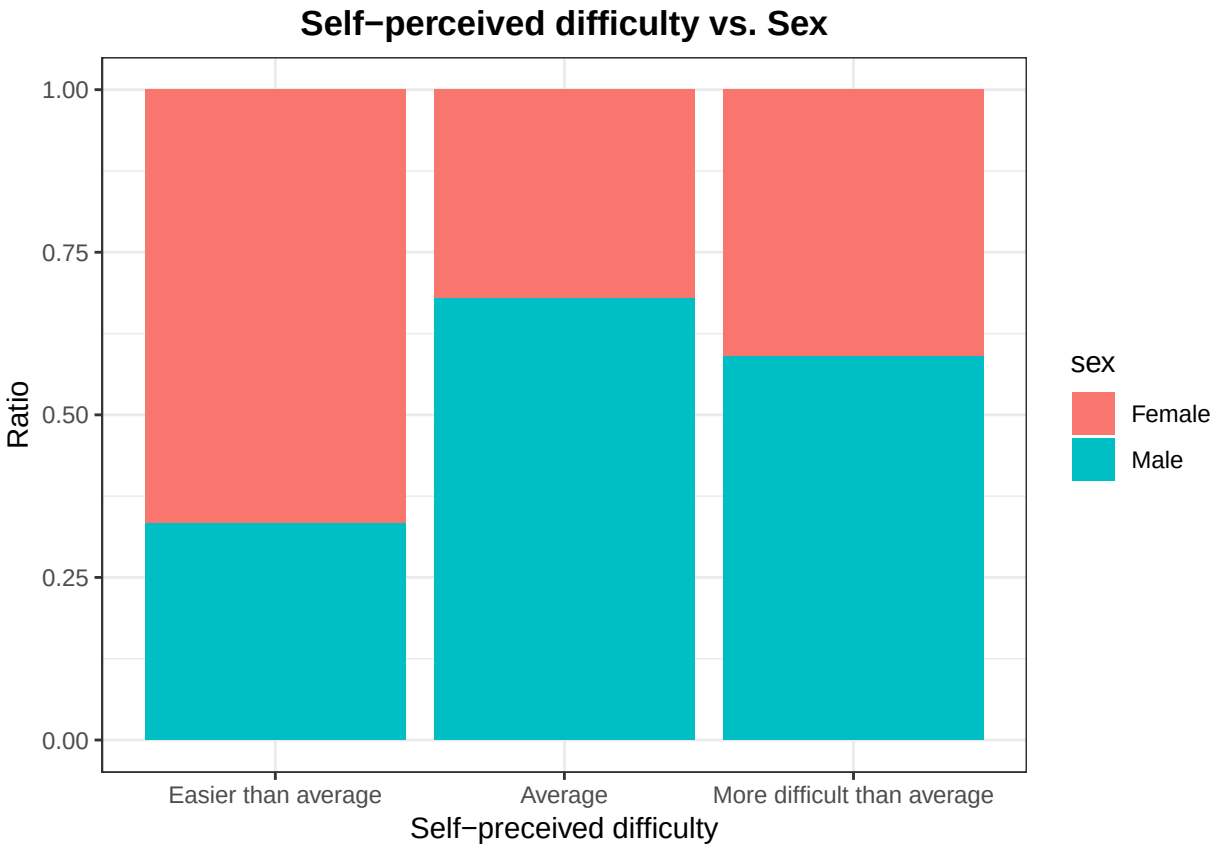

Figure 3.

Most male students reported average difficulty. The number of female students who felt the course was easy is greater than the number of male students who felt that way.

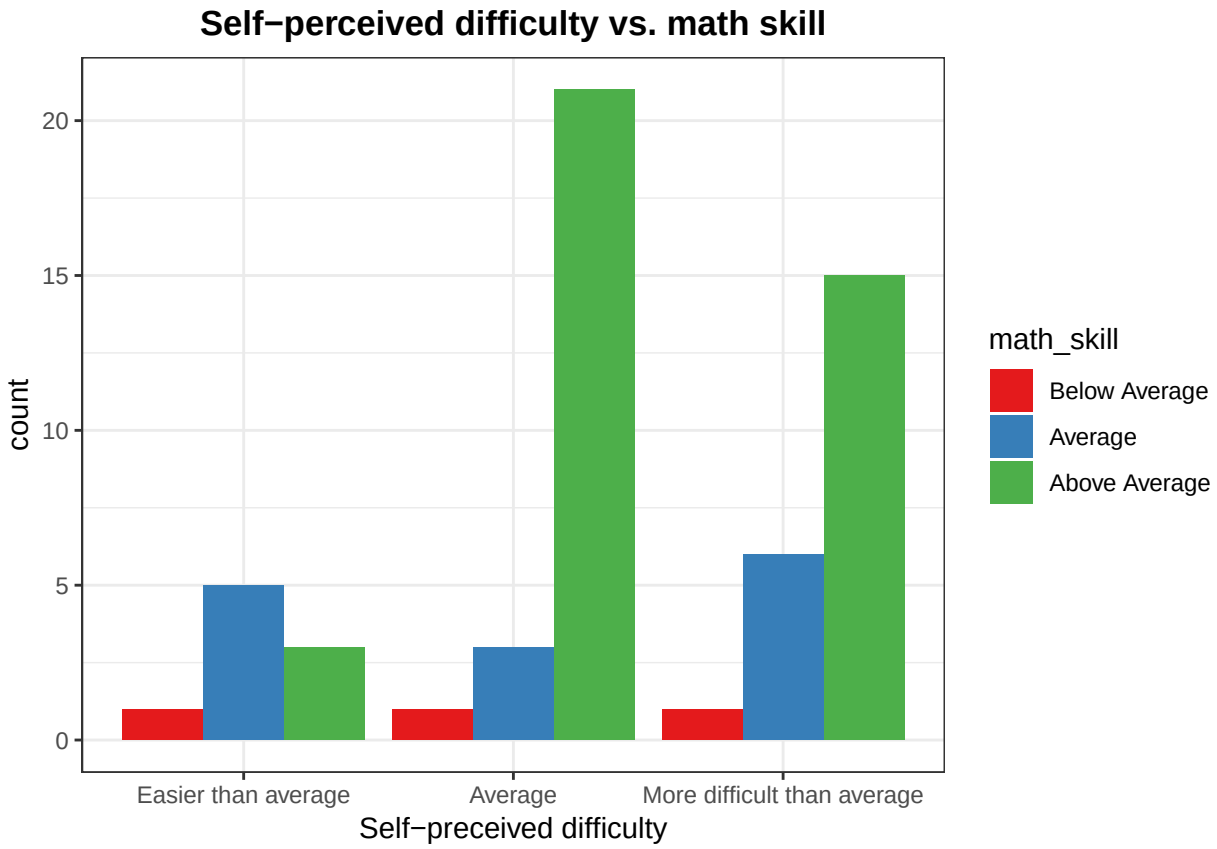

Figure 4.

In this figure, we observed that the most of students who have proficiency in math felt that the difficulty of DSCI 512 is average. It seems that the math skill does not affect the self-perceived difficulty of the course.

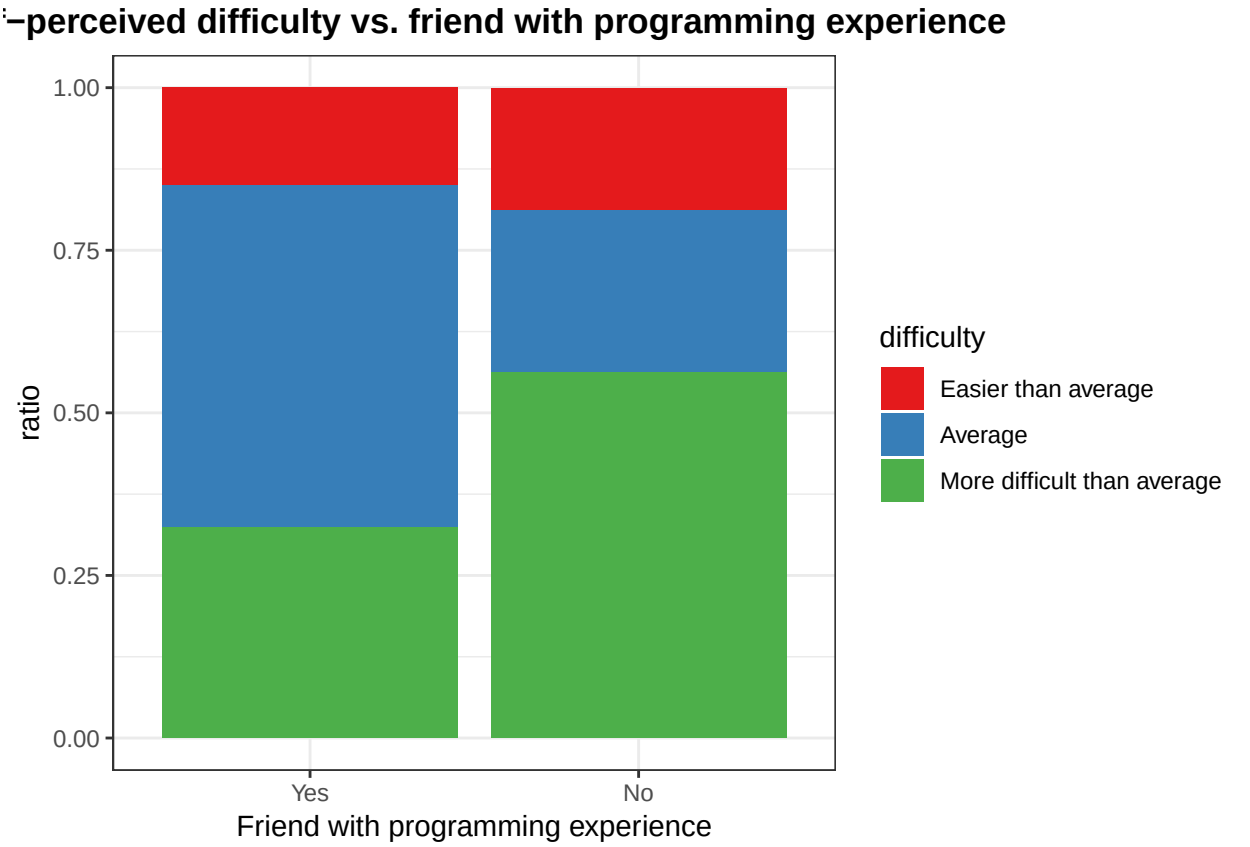

Figure 5.

It is interesting to note that the students who have no friends or family with programming experience are more likely to feel more difficult with the course materials. If a student has friends with programming experience, he/she tends to think that the difficulty of the course is average.

4.0 Analysis

To analyze our data, we implemented two methods of testing: (1) Ordinal Logistic Regression Test and (2) Likelihood Ratio Test with multiple Ordinal Regression Models.

4.1 Method 1 - Ordinal Logistic Regression Test

We decided to apply ordinal regression to test whether the main exposure `prog_exp` with some confounders have a significant impact on our outcome `difficulty`. The original regression is used to facilitate the interaction of dependent variables (having multiple ordered levels) with one or more independent variables. We used the polr from `MASS` package to estimate the OLR model. We also had `Hess = TRUE` to return the observed information matrix from optimization for standard error.

```
# fit ordered logit regression model
m <- polr(difficulty~ prog_exp + sex+math_skill + friend_with_prog, data=clean_data, Hess=TRUE)
# view a summary of the model
summary(m)
```

```
## Call:
```

```
## polr(formula = difficulty ~ prog_exp + sex + math_skill + friend_with_prog,
##     data = clean_data, Hess = TRUE)
##
## Coefficients:
##                                 Value Std. Error  t value
## prog_expLess than 100 hours  -0.17284     0.8932 -0.19350
## prog_expLess than 1000 hours  0.03418     0.9174  0.03726
## prog_expMore than 1000 hours -0.16640     0.9370 -0.17759
## sexMale                       0.33583     0.5400  0.62194
## math_skillAverage             0.33076     1.3670  0.24197
## math_skillAbove Average       0.88570     1.2939  0.68450
## friend_with_progNo            0.77935     0.6380  1.22162
##
## Intercepts:
##                                     Value   Std. Error t value
## Easier than average|Average         -0.6713  1.5074    -0.4453
## Average|More difficult than average  1.5127  1.5184     0.9963
##
## Residual Deviance: 111.0194
## AIC: 129.0194
```

The output of coefficients table contains the values of each coefficients, standard error and t values. Residual deviance and AIC are useful for later model comparison.

We calculated the p-values by comparing the t-value against the standard normal distribution and assumed that our data set is large enough.

```
# create table
ctable <- coef(summary(m))
# calculate p-values
p <- pnorm(abs(ctable[, "t value"]), lower.tail = FALSE) * 2
ctable <- cbind(ctable, "p value" = p)
ctable %>% kable()
```

	Value	Std. Error	t value	p value
prog_expLess than 100 hours	-0.1728359	0.8931988	-0.1935021	0.8465657
prog_expLess than 1000 hours	0.0341831	0.9173926	0.0372611	0.9702768
prog_expMore than 1000 hours	-0.1663964	0.9369797	-0.1775880	0.8590465
sexMale	0.3358303	0.5399724	0.6219398	0.5339814
math_skillAverage	0.3307642	1.3669667	0.2419695	0.8088038
math_skillAbove Average	0.8856973	1.2939238	0.6845050	0.4936563
friend_with_progNo	0.7793453	0.6379616	1.2216178	0.2218522
Easier than average|Average	-0.6713122	1.5074144	-0.4453401	0.6560740
Average|More difficult than average	1.5127481	1.5184046	0.9962747	0.3191167

Table 2. P-values of coefficients and intercepts

We can observe from table 6 that all p-values are greater than a typical threshold 0.05, which indicates that no significant difference exists. In other words, the different level of programming experience prior to the MDS program does not affect the MDS students' self-perceived difficulty of DSCI 512. The other exposure variables neither influence the outcome variable.

We also applied Anova type 3 from `Car` R package that can be used in ordinal regression model to verify the results above.

```r
# perform Anova type 3 test on the ordinal regression model
Anova(m, type = 3)
```

```
## Analysis of Deviance Table (Type III tests)
##
## Response: difficulty
##                   LR Chisq Df Pr(>Chisq)
## prog_exp           0.14187  3     0.9864
## sex                0.38773  1     0.5335
## math_skill         1.09944  2     0.5771
## friend_with_prog   1.51981  1     0.2176
```

It can be clearly seen that the results also suggest that neither one of variables has a significant impact on the outcome variable `difficulty`.

4.2 Method 2 - Likelihood Ratio Test with Ordinal Logistic Regression

Here we compared the fit of several different models. We have forgone any multiple comparison corrections due to the high p-values resulting from **all** of the models below. If any of these models had p-values less than 0.05, we would have adjusted the family-wise error rate via a Bonferroni correction.

First, we compared the null model to the model from Method 1 above:

```r
# Null model, no predictors
olr.M0 <- polr(difficulty~1, data=clean_data)
```

```r
# Full model (additive)
olr.M1 <- polr(difficulty~sex+math_skill+friend_with_prog+prog_exp, data=clean_data)
```

```r
# Are all variables good predictors?
lrtest(olr.M1, olr.M0)
```

```
## Likelihood ratio test
##
## Model 1: difficulty ~ sex + math_skill + friend_with_prog + prog_exp
## Model 2: difficulty ~ 1
##   #Df LogLik Df  Chisq Pr(>Chisq)
## 1   9 -55.51
## 2   2 -57.17 -7 3.3203     0.8539
```

Based on the p-value, this model does not provide a better fit than the null model. Next, we tried a model using only our main independent variable, `prog_exp`:

```r
# Reduced model, with our main variable
olr.M2 <- polr(difficulty~prog_exp, data=clean_data)
```

```r
# How does the model without any confounding variables perform?
lrtest(olr.M2, olr.M0)
```

```
## Likelihood ratio test
##
## Model 1: difficulty ~ prog_exp
## Model 2: difficulty ~ 1
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   5 -56.951
## 2   2 -57.170 -3 0.4385     0.9322
```

This model provided an even worse fit compared to the null model.

Lastly, we tried modelling an interaction between `friend_with_prog`, and `prog_exp`. The rationale here is that the effect of having a friend or family member with programming experience may vary based on personal programming experience. For example, a student with more than 1000 hours of programming experience may not benefit from having a friend or family member with programming experience because they already have the programming that the friend or family member might otherwise help them with. On the other hand, a student with no programming experience probably stands to benefit far more if they have a friend or family member who can help them with coding.

```
# Modeling a possible interaction
olr.M3 <- polr(difficulty~friend_with_prog*prog_exp, data=clean_data)
```

```
# How does the interaction model perform?
lrtest(olr.M3, olr.M0)
```

```
## Likelihood ratio test
##
## Model 1: difficulty ~ friend_with_prog * prog_exp
## Model 2: difficulty ~ 1
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   9 -53.868
## 2   2 -57.170 -7 6.6033     0.4713
```

Again, this does not yield a better fit than the null model.

5.0 Discussion

5.1 Findings

The main aspect of our analysis is to investigate the influence of prior programming experience on the outcome, self-perceived difficulty of DSCI 512, based on the assumption that the relationship between each pair of outcome groups is the same. Before the analysis, we identified three other confounding variables, sex, mathematics skill level and whether or not a student has friends or family with programming experience. In this report, we analyzed the relationship between the main exposure `prog_exp` and the outcome `difficulty`, as well as determined the casual effect of those confounders.

We performed two methods of testing, basic ordinal logistic regression test and likelihood ratio test between multiple pairs of models.

In the ordinal logistic regression, we found that none of exposure variable has p-values greater than 0.05 in table 2, which indicates that we do not evidence to reject the null hypothesis. The Anova type 3 test also provides similar results.

Similarly, when we compared multiple alternative models via the likelihood ratio test, we did not find any that were a significantly better fit than the null model. Therefore, this method also indicates that we do not evidence to reject the null hypothesis.

5.2 Survey design

Design and Assumption

We put a lot of thought into which variables to include as potential confounders. Spending the proper amount of time on this before distributing the survey helped ensure that we did not realize additional potential confounders during our analysis when it would be too late to gather data to control for them. This diligence

helped our end goal of reaching a conclusion free of spurious findings. We also made an assumption that relationship between each pair of outcome groups is the same.

We also streamlined our survey questions to lighten the cognitive load on participants. We believe this helped us maximize our sample size. Unfortunately, this involved lowering the number of intervals in our observed variables to a point where we could only analyze them as categorical variables. This problem is discussed further under **Future Directions**, below.

Limitations

One of the biggest problems that we discovered after performing the analysis is that the amount of data we collected is simply not enough to make conclusive claims. Perhaps it would have been helpful if we decided to collect data from previous cohorts of the MDS program. It might also be better if we continued our research to allow future MDS cohorts to reflect and take the survey.

Another issue of our survey is that the level of "self-reported" information is very subjective. An "Average" difficulty might mean something different to two different students. Therefore, it is quite difficult to evaluate the results of our research.

Future Directions

Originally when we were designing our survey, we thought it was very logical to make our variables categorical and ordinal, even our response variable (i.e. self-perceived difficulty of DSCI 512). However, while we were performing analyses and tests with our data, we realized that because our variables were not numerical, we lost a lot of flexibility with our analysis. If, for example, our response variable was numerical, we could have performed more tests such as simple ANOVA test. Furthermore, if we found that numerical data does not work with our analysis, we could have binned them to become categorical. We feel that only using categorical data limited our ability to perform different analysis, and if we were to perform similar research in the future, this is definitely something we would change.

6.0 References

1. How to use Multinomial and Ordinal Logistic Regression in R ?
2. Ordinal logistic regression