The effect of post-undergraduate training on passing first-screening for Data Science job applications

[redacted]

[redacted]

Background

The Berkeley MIDS degree is a relatively new Master's degree in Information and Data Science. There are publicly available, published data that compare before-and-after statistics of students that have completed the program. These statistics include mean before-and-after salary. However, from the research completed by the authors, we have not found an experiment conducted to-date that has shown causal effect of the MIDS degree with respect to increased success in obtaining a Data Science job for graduates of the program. In this study, we seek to investigate this causal effect.

Objective/Purpose

The purpose of this experiment is to investigate the causal effect of a Berkeley MIDS degree on an applicant's ability to obtain a Data Science job.

Methods

Study Design

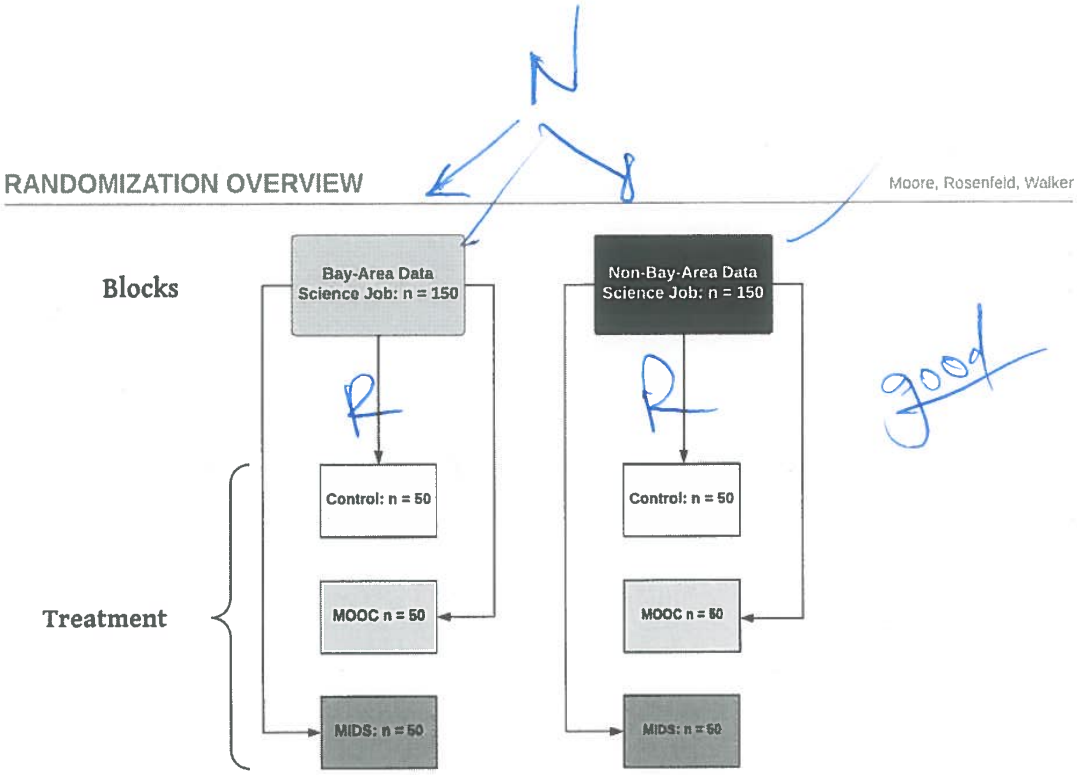
2x3

same

This study was a blocked, factorial randomized-controlled, single-blinded experiment. Factorial treatment composed of post-undergraduate training in Data Science (MOOC or MIDS) was compared to control in online applications withing Glassdoor to jobs within the Data Science professional space. This yielded a two-block (Bay-area Data Science job vs Non-Bay-area Data Science job) 3 treatment-arm factorial design with three possible randomizations per block (control, MOOC, or MIDS).

In total, 300 Data Science jobs on Glassdoor were randomized, with blocked randomization performed for Bay-area Data Science jobs and Non-Bay-area Data Science jobs as diagrammed and explained in further detail below in "Randomization" subsection.

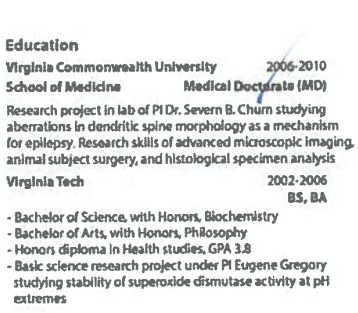
This can be/should be "unwound" some. Right now, it is pretty terse, and reads "jargony."



Details on Treatment

The treatment was applied at the level of the resume associated with the application. A single resume template was created that served as the blueprint for the arms of the experiment. All fields within the resume were identical between the arms of the experiment, with the exception of Education, which served as the treatment assignment. Education information was altered on the resumes such that the applicant either had no MOOC and no MIDS experience, just MOOC, or just MIDS. This information was carried-through on any required manual data input when applying for the jobs.

The result of this design was 3 resumes that were identical other than the education information, yielding a true apples-to-apples comparison between arms of the experiment.



Education

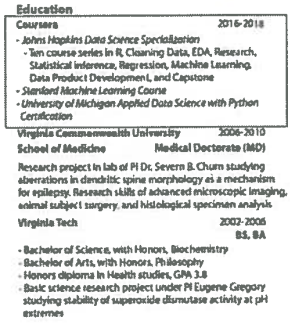
Virginia Commonwealth University 2006-2010
School of Medicine **Medical Doctorate (MD)**

Research project in lab of PI Dr. Severn B. Churn studying aberrations in dendritic spine morphology as a mechanism for epilepsy. Research skills of advanced microscopic imaging, animal subject surgery, and histological specimen analysis

Virginia Tech 2002-2006
BS, BA

- Bachelor of Science, with Honors, Biochemistry
- Bachelor of Arts, with Honors, Philosophy
- Honors diploma in Health studies, GPA 3.8
- Basic science research project under PI Eugene Gregory studying stability of superoxide dismutase activity at pH extremes

Control



Education

Coursera 2016-2018
- *Johns Hopkins Data Science Specialization*
  - Ten course series in R, Cleaning Data, EDA, Research, Statistical inference, Regression, Machine Learning, Data Product Development, and Capstone
- *Stanford Machine Learning Course*
- *University of Michigan Applied Data Science with Python Certification*

Virginia Commonwealth University 2006-2010
School of Medicine **Medical Doctorate (MD)**

Research project in lab of PI Dr. Severn B. Churn studying aberrations in dendritic spine morphology as a mechanism for epilepsy. Research skills of advanced microscopic imaging, animal subject surgery, and histological specimen analysis

Virginia Tech 2002-2006
BS, BA

- Bachelor of Science, with Honors, Biochemistry
- Bachelor of Arts, with Honors, Philosophy
- Honors diploma in Health studies, GPA 3.8
- Basic science research project under PI Eugene Gregory studying stability of superoxide dismutase activity at pH extremes

MOOC

Education

University of California, Berkeley 2017-2018
Master of Information and Data Science (MIDS)
- Masters degree in Data Science, graduation Summer 2018 with overall GPA 3.9
- Courses in Research Design, Statistics for Data Science, Data Engineering, Applied Machine Learning, Machine Learning at Scale, Big Data, Statistical Methods for Time Series, Experiments and Causation, and Capstone

Virginia Commonwealth University 2006-2010
School of Medicine **Medical Doctorate (MD)**

Research project in lab of PI Dr. Severn B. Churn studying aberrations in dendritic spine morphology as a mechanism for epilepsy. Research skills of advanced microscopic imaging, animal subject surgery, and histological specimen analysis

Virginia Tech 2002-2006
BS, BA

- Bachelor of Science, with Honors, Biochemistry
- Bachelor of Arts, with Honors, Philosophy
- Honors diploma in Health studies, GPA 3.8
- Basic science research project under PI Eugene Gregory studying stability of superoxide dismutase activity at pH extremes

MIDS

why is everything bold?

Inclusion Criteria

The following terms in isolation or in combination were input into the job search function in Glassdoor.com: Data, Scientist, Engineer, Business, Analyst, Natural Language Processing, NLP, Consultant, Machine Learning. In order to isolate Bay-area jobs, the "Location" filter was set to San Francisco, Oakland, Berkeley, Palo Alto, San Jose, Sunnyvale, or another Bay-area city/town. To isolate Non-Bay-area jobs, the Location filter was left blank, and jobs that appeared within the search that were within the Bay-area were excluded.

Exclusion Criteria

Jobs that met the following criteria were excluded from randomization:

1. Jobs containing "Data" in the title but are clearly not within Data Science (ex: "Data Entry Specialist")
2. Jobs requiring a PhD degree
3. Jobs requiring a Master's degree
4. Jobs with information fields in the application that are required in order to apply for the job, but were not common in job applications such that filling-in the information would have introduced covariate information into the randomization

Randomization

Randomization was coded in R using the "blocksdesign" package such that there were 2 replicates (Bay-area = 1, Non-Bay-area = 2) of 3 treatments (Control = 1, MOOC = 2, MIDS = 3). Each of the six end-points of randomization had 50 jobs within it, with a total of 300 jobs randomized to treatment in the experiment. The design of the study ensured equal assignment to each arm of the experiment for both blocks within the study. The job applications were all completed within Glassdoor.com by three experimenters using a standard technique. Any additional information required by the job application was standardized between the experimenters. This information included such items such as Disability status, Gender identification, Cover Letter, Objective, and others.

```
our_treatments = factor(1:3)
our_blocks_make = factor(rep(1:2, each = 150))
our_blocks = data.frame(our_blocks_make)
our_design <- design(our_treatments, our_blocks)
head(our_design$design)
```

```
##   our_blocks_make TF
## 1               1  2
## 2               1  3
## 3               1  1
## 4               1  3
## 5               1  1
## 6               1  2
```

```
tail(our_design$design)
```

```
##     our_blocks_make TF
## 295               2  1
## 296               2  2
## 297               2  3
## 298               2  3
```

The nature of the outcome, and the timeline of the study leads it to a HAZARD-MODEL type of analysis

```
## 299          2  3
## 300          2  2
```

Outcome Measure

The primary outcome measure was binary:

- 1 if the job responded with interest via email, phone call, or embedded job-search message prior to the close of the study on 8/8/2018
- 0 if the job did not respond with interest via email, phone call, or embedded job-search message prior to the close of the study on 8/8/2018

Pre-treatment Covariates

Due to the applicant's resume having a heavy domain focus on healthcare and biomedical research, it was hypothesized that a Data Science job posting by an organization within healthcare or biomedical research may be more likely to express interest in the candidate. Therefore, we tracked this covariate prior to treatment assignment in order to ensure that randomization properly allocated these jobs equally to the control, MOOC, and MIDS arms of the experiment. We also included a dummy variable for Healthcare job in our regression analysis.

Statistical Analysis

The goal of our statistical analysis was to estimate separate treatment effects for the two types of post-undergraduate treatments (MOOC and MIDS). To accomplish this, we will regressed our primary outcome on dummy variables for each of our possible sub-treatments, our blocking variable, and all possible interaction terms in a fully-saturated model. We calculated robust standard errors and confidence intervals, as well as obtained the estimate of the treatment effect and the corresponding p-value from the regression model.

$$Outcome = B_0 + B_1 Bay + B_2 MOOC + B_3 MIDS + B_4 Health + B_5(Bay * MOOC) + B_6(Bay * MIDS) + B_7(Bay * Health) + B_8(Health * MOOC) + B_9(Health * MIDS) + B_{10}(Bay * Health * MOOC) + B_{11}(Bay * Health * MIDS)$$

I often like to show the simplest treatment/control contrast, and then

Results

Data Clean-up

```
# Import csv with necessary data to show table
experiment_df <- read.csv([redacted]W241_Final_Data - Sheet1.csv")
experiment_dt <- data.table(experiment_df)
# View(experiment_dt) No response at all from a company was
# viewed as the same as a response declining to move forward
# with hiring our applicant. Therefore, all NAs are
# equivalent to a response value of 0 and will be coded as
# such
experiment_dt$Binary_Response[is.na(experiment_dt$Binary_Response)] <- 0
# View(experiment_dt)
```

fread( )

dt[is.na(Binary_Response), Binary_Resp := 0]

include other factors to compliment the core model.

So bold...

Randomization Check: Covariate Balance

The authors hypothesized that two covariates may strongly correlate with our outcome of interest. The first of these covariates was whether or not the job being advertised was within the greater "Bay Area". It is reasonable to suspect that the Berkeley academic brand may be more respected, revered, and present within the Bay Area. Therefore, we thought that a job's location within this area may explain a significant portion of the variance in the outcome. We chose to block on location of a job within or outside of the Bay Area to ensure that we achieved adequate allocation of jobs equally among the treatment groups within the two blocks. The second covariate of interest was a job's focus within the healthcare or biomedical domain. The fake applicants used in this experiment all had an obvious and strong domain background in healthcare and biomedical research, such that it is reasonable to assume that jobs within this domain would be more likely to pursue the applicant as a hire. Given that the healthcare domain status of a job would potentially strongly explain the variance in the outcome, we chose to track this covariate prior to assigning treatment. We check to see if this covariate is statistically significantly different between the treatment assignment groups. We have created a dummy variable "Health_Care" that is coded as 1 if the job is within healthcare or biomedical science and is coded as 0 otherwise.

this pipeline work probably need not be surfaced @ the report level

```
# Check to see if there is a statistically significant
# difference in Health_Care between the 6 different arms of
# the experiment To more easily form group in this section,
# we reassign a different treatment arm integer to the
# treatment groups in the Non-Bay-area block
experiment_dt_covariate_check <- experiment_dt
experiment_dt_covariate_check$Treatment[experiment_dt_covariate_check$Block ==
    2 & experiment_dt_covariate_check$Treatment == 1] <- 4
experiment_dt_covariate_check$Treatment[experiment_dt_covariate_check$Block ==
    2 & experiment_dt_covariate_check$Treatment == 2] <- 5
experiment_dt_covariate_check$Treatment[experiment_dt_covariate_check$Block ==
    2 & experiment_dt_covariate_check$Treatment == 3] <- 6
# To perform ANOVA test on difference of means in multiple
# groups, we assume that there is equal variance amongst the
# groups. Here we check this via a Bartlett test of
# homogeneity of variances
bartlett.test(experiment_dt_covariate_check$Health_Care, factor(experiment_dt_covariate_check$Treatm
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  experiment_dt_covariate_check$Health_Care and factor(experiment_dt_covariate_check$Treatme
## Bartlett's K-squared = 26.613, df = 5, p-value = 6.784e-05
```

We see from our Bartlett test of homogeneity of variances that we can reject the null hypothesis that the variances of Health_Care for our 6 groups are homogenous. This means we have a violation of the homogeneity of variance between groups with respect to proceeding with an ANOVA test on the difference in means for our covariate of interest, Health_Care. We proceed by fitting a heteroskedastic model and then performing an F-test for difference of group means for Health_Care. We now repeat the Bartlett test within each block (Bay and Non-Bay).

```
# Bartlett test for Bay block
bay_block_only <- experiment_dt_covariate_check[experiment_dt_covariate_check$Block ==
    1]
bartlett.test(bay_block_only$Health_Care, factor(bay_block_only$Treatment))
```

```
##
```

→ How can chain this to avoid making the intermediate object.

```
##  Bartlett test of homogeneity of variances
##
## data:  bay_block_only$Health_Care and factor(bay_block_only$Treatment)
## Bartlett's K-squared = 7.1017, df = 2, p-value = 0.0287
```

```
# Bartlett test for Non-Bay block
nonbay_block_only <- experiment_dt_covariate_check[experiment_dt_covariate_check$Block ==
    2]
bartlett.test(nonbay_block_only$Health_Care, factor(nonbay_block_only$Treatment))
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  nonbay_block_only$Health_Care and factor(nonbay_block_only$Treatment)
## Bartlett's K-squared = 9.9362, df = 2, p-value = 0.006957
```

Again we see that even when we analyze the variance for Health_Care within the treatment groups within each of our two blocks, that we can reject the null hypothesis that the variance for Health_Care within each of the three treatment arms in each block is homogenous.

Below we construct a table to show the means for Health_Care for each of the three treatment arms within the two blocks of our experiment. We see that the range is from 0.06 for Non-Bay Control to 0.26 for Bay Control.

```
mean_HC_bay_control <- mean(experiment_dt_covariate_check$Health_Care[experiment_dt_covariate_check$
    1])
mean_HC_bay_MOOC <- mean(experiment_dt_covariate_check$Health_Care[experiment_dt_covariate_check$Tre
    2])
mean_HC_bay_MIDS <- mean(experiment_dt_covariate_check$Health_Care[experiment_dt_covariate_check$Tre
    3])
mean_HC_nonbay_control <- mean(experiment_dt_covariate_check$Health_Care[experiment_dt_covariate_che
    4])
mean_HC_nonbay_MOOC <- mean(experiment_dt_covariate_check$Health_Care[experiment_dt_covariate_check$
    5])
mean_HC_nonbay_MIDS <- mean(experiment_dt_covariate_check$Health_Care[experiment_dt_covariate_check$
    6])
HC_mean_table <- matrix(c(mean_HC_bay_control, mean_HC_bay_MOOC,
    mean_HC_bay_MIDS, mean_HC_nonbay_control, mean_HC_nonbay_MOOC,
    mean_HC_nonbay_MIDS), ncol = 1, byrow = TRUE)
colnames(HC_mean_table) <- c("Mean")
rownames(HC_mean_table) <- c("Mean Bay Control", "Mean Bay MOOC",
    "Mean Bay MIDS", "Mean Non-Bay Control", "Mean Non-Bay MOOC",
    "Mean Non-Bay MIDS")
HC_mean_table <- as.table(HC_mean_table)
HC_mean_table
```

```
##                      Mean
## Mean Bay Control     0.26
## Mean Bay MOOC        0.20
## Mean Bay MIDS        0.10
## Mean Non-Bay Control 0.06
## Mean Non-Bay MOOC    0.08
## Mean Non-Bay MIDS    0.16
```

```
fit_gls_hetero <- gls(Health_Care ~ factor(Treatment), experiment_dt_covariate_check,
    weights = varIdent(form = ~1 | factor(Treatment)))
anova(fit_gls_hetero, type = "marginal")
```

```
## Denom. DF: 294
##                    numDF  F-value p-value
## (Intercept)            1 9.333392  0.0025
## factor(Treatment)      5 2.374358  0.0392
```

The results of our F-test on the difference in means for Health_Care between our six different arms of the experiment yields a p-value of 0.0392. Therefore, we reject the null hypothesis that states that there is not a difference between the six different arms of the experiment with respect to the mean for Health_Care. As we hypothesized that a job's healthcare domain status may explain the variance in our outcome of interest, we have shown that we have a failure of randomization to ensure that there is no statistically significant difference between the mean value of Health_Care between our treatment groups overall. This is initially concerning for evidence of covariate imbalance and a less than ideal randomization. However, at present this merely requires further investigation to see if there is covariate imbalance within each block (Bay and Non-Bay).

Below we repeat our F-test within each block, Bay and Non-Bay, to see if there is evidence of covariate imbalance for Health_Care intrablock in addition to interblock.

```
# F-test for Bay
fit_gls_hetero_just_bay <- gls(Health_Care ~ factor(Treatment),
    bay_block_only, weights = varIdent(form = ~1 | factor(Treatment)))
anova(fit_gls_hetero_just_bay, type = "marginal")
```

```
## Denom. DF: 147
##                    numDF  F-value p-value
## (Intercept)            1 5.444440  0.0210
## factor(Treatment)      2 2.486848  0.0867
```

```
# F-test for Non-Bay
fit_gls_hetero_just_nonbay <- gls(Health_Care ~ factor(Treatment),
    nonbay_block_only, weights = varIdent(form = ~1 | factor(Treatment)))
anova(fit_gls_hetero_just_nonbay, type = "marginal")
```

```
## Denom. DF: 147
##                    numDF  F-value p-value
## (Intercept)            1 9.333333  0.0027
## factor(Treatment)      2 1.303824  0.2746
```

We see that the p-value for the F-test for the Bay block is 0.0867, approaching statistical significance at the alpha = 0.05 level but not achieving it. The p-value for the F-test for the Non-Bay block is 0.2746. Therefore, we fail to reject the null hypothesis for both the Bay and the Non-Bay blocks, which states that there is not a difference in the mean for Health_Care between the treatment arms within each block. This is evidence of covariate balance within each block, and therefore proper randomization.

Primary Outcome

The Raw Data

We briefly review the raw response data to get a picture of the overall counts of the responses for the control, MOOC, and MIDS treatment arms.

```
control_positive_response_count <- sum(experiment_dt$Binary_Response[experiment_dt$Treatment ==
    1])
control_positive_response_count_bay <- sum(experiment_dt$Binary_Response[experiment_dt$Treatment ==
    1 & experiment_dt$Block == 1])
```

```
control_positive_response_count_nonbay <- sum(experiment_dt$Binary_Response[experiment_dt$Treatment
    1 & experiment_dt$Block == 2])
MOOC_positive_response_count <- sum(experiment_dt$Binary_Response[experiment_dt$Treatment ==
    2])
MOOC_positive_response_count_bay <- sum(experiment_dt$Binary_Response[experiment_dt$Treatment ==
    2 & experiment_dt$Block == 1])
MOOC_positive_response_count_nonbay <- sum(experiment_dt$Binary_Response[experiment_dt$Treatment ==
    2 & experiment_dt$Block == 2])
MIDS_positive_response_count <- sum(experiment_dt$Binary_Response[experiment_dt$Treatment ==
    3])
MIDS_positive_response_count_bay <- sum(experiment_dt$Binary_Response[experiment_dt$Treatment ==
    3 & experiment_dt$Block == 1])
MIDS_positive_response_count_nonbay <- sum(experiment_dt$Binary_Response[experiment_dt$Treatment ==
    3 & experiment_dt$Block == 2])

table_of_positive_response_counts <- matrix(c(control_positive_response_count_bay,
    MOOC_positive_response_count_bay, MIDS_positive_response_count_bay,
    control_positive_response_count_nonbay, MOOC_positive_response_count_nonbay,
    MIDS_positive_response_count_nonbay, control_positive_response_count,
    MOOC_positive_response_count, MIDS_positive_response_count),
    ncol = 3, byrow = FALSE)
colnames(table_of_positive_response_counts) <- c("Positive Count Bay ",
    " Positive Count Non_Bay ", " Positive Count Total")
rownames(table_of_positive_response_counts) <- c("Control", "MOOC",
    "MIDS")
table_of_positive_response_counts <- as.table(table_of_positive_response_counts)
table_of_positive_response_counts
```

```
##         Positive Count Bay   Positive Count Non_Bay   Positive Count Total
## Control                  3                        1                      4
## MOOC                     0                        2                      2
## MIDS                     0                        2                      2
```

There were 4 positive responses in the control group, with 3 in the Bay and 1 outside the Bay. There were 2 positive responses for the MOOC training resume, with zero of these in the Bay and 2 outside the Bay. For the MIDS treatment arm, there were 2 positive responses, with none in the Bay and 2 outside the Bay.

Next, we calculated estimates of our treatment effects via three different methods.

1) Estimating Treatment Effects With Regression: Fully Saturated Model with Dummy Variable for Blocks

To form our fully saturated regression model to estimate our treatment effects, we will need to encode some additional indicator variables. We also create functions for generating robust standard error estimates and corresponding confidence intervals for the estimate of average treatment effect with robust standard errors.

```
experiment_dt$Bay[experiment_dt$Block == 1] <- 1
experiment_dt$Bay[experiment_dt$Block != 1] <- 0
experiment_dt$MOOC[experiment_dt$Treatment == 2] <- 1
experiment_dt$MOOC[experiment_dt$Treatment != 2] <- 0
experiment_dt$MIDS[experiment_dt$Treatment == 3] <- 1
experiment_dt$MIDS[experiment_dt$Treatment != 3] <- 0
# View(experiment_dt)
```

```
# Create function for CI output from ATE and robust SE input
se_robust_calculate_out_CI <- function(est_ATE, rob_SE) {
    # Calculate 95% CI and print
    lower_CI_bound_this_rm <- est_ATE - rob_SE * 1.96
    higher_CI_bound_this_rm <- est_ATE + rob_SE * 1.96
    return(c((round(as.numeric(lower_CI_bound_this_rm), 4)),
        (round(as.numeric(higher_CI_bound_this_rm), 4))))
}

# Create function for robust SE calculation with SE output in
# the case of nultiple coefficients from regression being
# used for estimate of ATE calc
se_robust_calculate_out_SE <- function(this_regression_model,
    coef_1, coef_2) {
    # Get robust SE and then extract treatment effect and SE
    this_regression_model$vcovHC <- vcovHC(this_regression_model,
        type = "HC0")
    se_this_rm <- sqrt(((diag(this_regression_model$vcovHC)[coef_1])) +
        ((diag(this_regression_model$vcovHC)[coef_2])) + (2 *
        ((this_regression_model$vcovHC)[coef_1, coef_2])))
    return(round(as.numeric(se_this_rm), 4))
}
```

what is the critical value that runs w/ N=300. pt

We now create a fully saturated model to obtain an estimate for the average treatment effect for a MIDS degree and an estimate for the average treatment effect for MOOC training. Our first method to estimate the average treatment effects within each of our blocks will be to use the fully saturated model with a dummy variable for our blocking unit ("Bay"). We then obtain estimates of our average treatment effects via the coefficients on our regression terms.

```
# Regress Binary_Outcome on fully saturated model variables
lm_outcome <- lm(Binary_Response ~ MIDS + MOOC + Bay + Health_Care +
    Bay:MOOC + Bay:MIDS + Bay:Health_Care + Health_Care:MOOC +
    Health_Care:MIDS + Bay:Health_Care:MOOC + Bay:Health_Care:MIDS,
    data = experiment_dt)

# Get full output of coefficients
coeftest(lm_outcome)
```

-1, to suppress intercept

I would suppress the intercept so that the reported coef are mean response rates in each cell.

Eg. I can figure out that nobody from non-bay control, health-care responded, but it's hard.

```
##
## t test of coefficients:
##
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             0.0212766  0.0234789  0.9062   0.3656
## MIDS                    0.0263425  0.0341780  0.7707   0.4415
## MOOC                    0.0222017  0.0333841  0.6650   0.5066
## Bay                     0.0057504  0.0353766  0.1625   0.8710
## Health_Care            -0.0212766  0.0958520 -0.2220   0.8245
## MOOC:Bay               -0.0492287  0.0496233 -0.9920   0.3220
## MIDS:Bay               -0.0533695  0.0494383 -1.0795   0.2813
## Bay:Health_Care         0.1480957  0.1089994  1.3587   0.1753
## MOOC:Health_Care       -0.0222017  0.1273896 -0.1743   0.8618
## MIDS:Health_Care       -0.0263425  0.1142065 -0.2307   0.8177
## MOOC:Bay:Health_Care   -0.1046175  0.1488623 -0.7028   0.4828
## MIDS:Bay:Health_Care   -0.1004767  0.1466081 -0.6853   0.4937
```

```
# Get treatment effect estimates for MIDS in Bay, MOOC in
# Bay, MIDS in Non-Bay, and MOOC in Non-Bay from model MIDS
# in Bay = MIDS:Bay + MIDS:Bay:Health_Care
te_MIDS_bay_saturated <- round((coeftest(lm_outcome)[7, 1] +
    coeftest(lm_outcome)[12, 1]), 4)
# MOOC in Bay = MOOC:Bay + MOOC:Bay:Health_Care
te_MOOC_bay_saturated <- round((coeftest(lm_outcome)[6, 1] +
    coeftest(lm_outcome)[11, 1]), 4)
# MIDS in Non-Bay = MIDS + MIDS:Health_Care
te_MIDS_nonbay_saturated <- round((coeftest(lm_outcome)[2, 1] +
    coeftest(lm_outcome)[10, 1]), 4)
# MOOC in Bay = MOOC:Bay + MOOC:Bay:Health_Care
te_MOOC_nonbay_saturated <- round((coeftest(lm_outcome)[3, 1] +
    coeftest(lm_outcome)[9, 1]), 4)

# Get robust SEs for all of our treatment effect estimates
# using pre-made function
rob_se_MIDS_bay_sat <- se_robust_calculate_out_SE(lm_outcome,
    7, 12)
rob_se_MOOC_bay_sat <- se_robust_calculate_out_SE(lm_outcome,
    6, 11)
rob_se_MIDS_nonbay_sat <- se_robust_calculate_out_SE(lm_outcome,
    2, 10)
rob_se_MOOC_nonbay_sat <- se_robust_calculate_out_SE(lm_outcome,
    3, 9)

# Get lower and upper bound of 95% CI for estimates of
# treatment effects
lower_rob_se_MIDS_bay_sat <- se_robust_calculate_out_CI(te_MIDS_bay_saturated,
    rob_se_MIDS_bay_sat)[1]
upper_rob_se_MIDS_bay_sat <- se_robust_calculate_out_CI(te_MIDS_bay_saturated,
    rob_se_MIDS_bay_sat)[2]
lower_rob_se_MOOC_bay_sat <- se_robust_calculate_out_CI(te_MOOC_bay_saturated,
    rob_se_MOOC_bay_sat)[1]
upper_rob_se_MOOC_bay_sat <- se_robust_calculate_out_CI(te_MOOC_bay_saturated,
    rob_se_MOOC_bay_sat)[2]
lower_rob_se_MIDS_nonbay_sat <- se_robust_calculate_out_CI(te_MIDS_nonbay_saturated,
    rob_se_MIDS_nonbay_sat)[1]
upper_rob_se_MIDS_nonbay_sat <- se_robust_calculate_out_CI(te_MIDS_nonbay_saturated,
    rob_se_MIDS_nonbay_sat)[2]
lower_rob_se_MOOC_nonbay_sat <- se_robust_calculate_out_CI(te_MOOC_nonbay_saturated,
    rob_se_MOOC_nonbay_sat)[1]
upper_rob_se_MOOC_nonbay_sat <- se_robust_calculate_out_CI(te_MOOC_nonbay_saturated,
    rob_se_MOOC_nonbay_sat)[2]

# Make table to display the estimates fo the average
# treatment effects, robust SEs, and CIs
sat_reg_table <- matrix(c(te_MIDS_bay_saturated, te_MOOC_bay_saturated,
    te_MIDS_nonbay_saturated, te_MOOC_nonbay_saturated, rob_se_MIDS_bay_sat,
    rob_se_MOOC_bay_sat, rob_se_MIDS_nonbay_sat, rob_se_MOOC_nonbay_sat,
    lower_rob_se_MIDS_bay_sat, lower_rob_se_MOOC_bay_sat, lower_rob_se_MIDS_nonbay_sat,
    lower_rob_se_MOOC_nonbay_sat, upper_rob_se_MIDS_bay_sat,
    upper_rob_se_MOOC_bay_sat, upper_rob_se_MIDS_nonbay_sat,
```

```
    upper_rob_se_MOOC_nonbay_sat), ncol = 4, byrow = FALSE)
colnames(sat_reg_table) <- c("Estimate of ATE ", "Robust SE ",
    "95% CI Lower ", " 95% CI Upper")
rownames(sat_reg_table) <- c("MIDS Bay", "MOOC Bay", "MIDS Non-Bay",
    "MOOC Non-Bay")
sat_reg_table <- as.table(sat_reg_table)
sat_reg_table
```

```
##              Estimate of ATE  Robust SE  95% CI Lower   95% CI Upper
## MIDS Bay             -0.1538     0.1001       -0.3500         0.0424
## MOOC Bay             -0.1538     0.1001       -0.3500         0.0424
## MIDS Non-Bay          0.0000     0.0000        0.0000         0.0000
## MOOC Non-Bay          0.0000     0.0000        0.0000         0.0000
```

```
# Make table to display overall estimate of ATEs including
# both blocks
overall_MIDS_sat_ate <- te_MIDS_bay_saturated * (150/300) + te_MIDS_nonbay_saturated *
    (150/300)
overall_MOOC_sat_ate <- te_MOOC_bay_saturated * (150/300) + te_MOOC_nonbay_saturated *
    (150/300)
overall_ate_both_blocks_sat_table <- matrix(c(overall_MIDS_sat_ate,
    overall_MOOC_sat_ate), ncol = 1, byrow = FALSE)
colnames(overall_ate_both_blocks_sat_table) <- c("Overall Estimate of ATE via Method (1)")
rownames(overall_ate_both_blocks_sat_table) <- c("MIDS", "MOOC")
overall_ate_both_blocks_sat_table <- as.table(overall_ate_both_blocks_sat_table)
overall_ate_both_blocks_sat_table
```

```
##      Overall Estimate of ATE via Method (1)
## MIDS                                -0.0769
## MOOC                                -0.0769
```

Viewing the coefficients from the fully saturated regression model, we see that there is not statistical significance to any of our coefficients. In order to obtain the estimate of the ATE for each of our treatments, we added coefficients from the table as detailed in the code cell above. Next, we obtained standard errors and 95% confidence intervals for these estimates. The estimate of the ATE for MIDS in the Bay area is -0.1538 with a robust SE of 0.1001 and a 95% confidence interval of -0.3500 to 0.0424. We fail to reject the null hypothesis that the treatment effect of a MIDS degree in the Bay area is equal to zero as our CI contains zero. We see an identical estimate of the average treatment effect, robust SE, and CI for MOOC in the Bay area. Again, we fail to reject the null hypothesis that the treatment effect of MOOC training in the Bay area is equal to zero as our CI contains zero. In this model, we run into issues with perfect collinearity between MIDS and MIDS:Health_Care as well as with MOOC:Bay and MOOC:Bay:Health_Care. This creates an estimate of the ATE that is zero, robust SE of zero, and therefore no CI for both the MIDS Non-Bay and the MOOC Non-Bay estimates.

The overall estimate of the ATE for MIDS and MOOC are both -0.0769.

2) Estimating Treatment Effects With Regression: Separate Regression For Each Block

To confirm the estimates of our treatment effects above, we calculated estimates of the average treatment effect for MIDS Bay, MOOC Bay, MIDS Non-Bay, and MOOC Non-Bay by regressing our outcome on either block in isolation. This used a fully saturated model, but in this model there was no need for a dummy variable for Bay as there was when we calculated our estimate of the average treatment effects via method (1).

$$Outcome = B_0 + B_1 MIDS + B_2 MOOC + B_3 Health + B_4(Health : MOOC) + B_5(Health : MIDS)$$

```
# Need to engineer our dumy variables again
bay_block_only$MOOC[bay_block_only$Treatment == 2] <- 1
bay_block_only$MOOC[bay_block_only$Treatment != 2] <- 0
bay_block_only$MIDS[bay_block_only$Treatment == 3] <- 1
bay_block_only$MIDS[bay_block_only$Treatment != 3] <- 0
nonbay_block_only$MOOC[nonbay_block_only$Treatment == 5] <- 1
nonbay_block_only$MOOC[nonbay_block_only$Treatment != 5] <- 0
nonbay_block_only$MIDS[nonbay_block_only$Treatment == 6] <- 1
nonbay_block_only$MIDS[nonbay_block_only$Treatment != 6] <- 0

# Regress Binary_Outcome on fully saturated model variables
# for Bay block
lm_outcome_bay <- lm(Binary_Response ~ MIDS + MOOC + Health_Care +
    Health_Care:MOOC + Health_Care:MIDS, data = bay_block_only)
# Get full output of coefficients for Bay block model
coeftest(lm_outcome_bay)
```

```
##
## t test of coefficients:
##
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.027027   0.022366  1.2084 0.228876
## MIDS             -0.027027   0.030192 -0.8952 0.372187
## MOOC             -0.027027   0.031032 -0.8710 0.385231
## Health_Care       0.126819   0.043863  2.8912 0.004433 **
## MOOC:Health_Care -0.126819   0.065097 -1.9482 0.053341 .
## MIDS:Health_Care -0.126819   0.077699 -1.6322 0.104824
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Get treatment effect estimates for MIDS Bay and MOOC Bay
# from model MIDS in Bay = MIDS + MIDS:Health_Care
te_MIDS_bay_subdf <- round((coeftest(lm_outcome_bay)[2, 1] +
    coeftest(lm_outcome_bay)[6, 1]), 4)
# MOOC in Bay = MOOC + MOOC:Health_Care
te_MOOC_bay_subdf <- round((coeftest(lm_outcome_bay)[3, 1] +
    coeftest(lm_outcome_bay)[5, 1]), 4)

# Get robust SEs for all of our treatment effect estimates
# using pre-made function
rob_se_MIDS_bay_subdf <- se_robust_calculate_out_SE(lm_outcome_bay,
    2, 6)
rob_se_MOOC_bay_subdf <- se_robust_calculate_out_SE(lm_outcome_bay,
    3, 5)

# Get lower and upper bound of 95% CI for estimates of
# treatment effects
lower_rob_se_MIDS_bay_subdf <- se_robust_calculate_out_CI(te_MIDS_bay_subdf,
    rob_se_MIDS_bay_subdf)[1]
upper_rob_se_MIDS_bay_subdf <- se_robust_calculate_out_CI(te_MIDS_bay_subdf,
    rob_se_MIDS_bay_subdf)[2]
lower_rob_se_MOOC_bay_subdf <- se_robust_calculate_out_CI(te_MOOC_bay_subdf,
    rob_se_MOOC_bay_subdf)[1]
```

```
upper_rob_se_MOOC_bay_subdf <- se_robust_calculate_out_CI(te_MOOC_bay_subdf,
    rob_se_MOOC_bay_subdf)[2]

# Regress Binary_Outcome on fully saturated model variables
# for Non-Bay block
lm_outcome_nonbay <- lm(Binary_Response ~ MIDS + MOOC + Health_Care +
    Health_Care:MOOC + Health_Care:MIDS, data = nonbay_block_only)
# Get full output of coefficients for Non-Bay block model
coeftest(lm_outcome_nonbay)
```

```
##
## t test of coefficients:
##
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.021277   0.026622  0.7992   0.4255
## MIDS               0.026342   0.038753  0.6798   0.4978
## MOOC               0.022202   0.037853  0.5865   0.5584
## Health_Care       -0.021277   0.108682 -0.1958   0.8451
## MOOC:Health_Care  -0.022202   0.144441 -0.1537   0.8781
## MIDS:Health_Care  -0.026342   0.129493 -0.2034   0.8391
```

```
# Get treatment effect estimates for MIDS Non-Bay and MOOC
# Non-Bay from model MIDS in Non-Bay = MIDS +
# MIDS:Health_Care
te_MIDS_nonbay_subdf <- round((coeftest(lm_outcome_nonbay)[2,
    1] + coeftest(lm_outcome_nonbay)[6, 1]), 4)
# MOOC in Non-Bay = MOOC + MOOC:Health_Care
te_MOOC_nonbay_subdf <- round((coeftest(lm_outcome_nonbay)[3,
    1] + coeftest(lm_outcome_nonbay)[5, 1]), 4)

# Get robust SEs for all of our treatment effect estimates
# using pre-made function
rob_se_MIDS_nonbay_subdf <- se_robust_calculate_out_SE(lm_outcome_nonbay,
    2, 6)
rob_se_MOOC_nonbay_subdf <- se_robust_calculate_out_SE(lm_outcome_nonbay,
    3, 5)

# Get lower and upper bound of 95% CI for estimates of
# treatment effects
lower_rob_se_MIDS_nonbay_subdf <- se_robust_calculate_out_CI(te_MIDS_nonbay_subdf,
    rob_se_MIDS_nonbay_subdf)[1]
upper_rob_se_MIDS_nonbay_subdf <- se_robust_calculate_out_CI(te_MIDS_nonbay_subdf,
    rob_se_MIDS_nonbay_subdf)[2]
lower_rob_se_MOOC_nonbay_subdf <- se_robust_calculate_out_CI(te_MOOC_nonbay_subdf,
    rob_se_MOOC_nonbay_subdf)[1]
upper_rob_se_MOOC_nonbay_subdf <- se_robust_calculate_out_CI(te_MOOC_nonbay_subdf,
    rob_se_MOOC_nonbay_subdf)[2]

# Make table to display the estimates fo the average
# treatment effects, robust SEs, and CIs
subdf_reg_table <- matrix(c(te_MIDS_bay_subdf, te_MOOC_bay_subdf,
    te_MIDS_nonbay_subdf, te_MOOC_nonbay_subdf, rob_se_MIDS_bay_subdf,
    rob_se_MOOC_bay_subdf, rob_se_MIDS_nonbay_subdf, rob_se_MOOC_nonbay_subdf,
    lower_rob_se_MIDS_bay_subdf, lower_rob_se_MOOC_bay_subdf,
```

```
    lower_rob_se_MIDS_nonbay_subdf, lower_rob_se_MOOC_nonbay_subdf,
    upper_rob_se_MIDS_bay_subdf, upper_rob_se_MOOC_bay_subdf,
    upper_rob_se_MIDS_nonbay_subdf, upper_rob_se_MOOC_nonbay_subdf),
    ncol = 4, byrow = FALSE)
colnames(subdf_reg_table) <- c("Estimate of ATE ", "Robust SE ",
    "95% CI Lower ", " 95% CI Upper")
rownames(subdf_reg_table) <- c("MIDS Bay", "MOOC Bay", "MIDS Non-Bay",
    "MOOC Non-Bay")
subdf_reg_table <- as.table(subdf_reg_table)
subdf_reg_table
```

```
##              Estimate of ATE  Robust SE  95% CI Lower   95% CI Upper
## MIDS Bay             -0.1538     0.1001       -0.3500        0.0424
## MOOC Bay             -0.1538     0.1001       -0.3500        0.0424
## MIDS Non-Bay          0.0000     0.0000        0.0000        0.0000
## MOOC Non-Bay          0.0000     0.0000        0.0000        0.0000
```

```
# Make table to display overall estimate of ATEs including
# both blocks
overall_MIDS_subdfs_ate <- te_MIDS_bay_subdf * (150/300) + te_MIDS_nonbay_subdf *
    (150/300)
overall_MOOC_subdfs_ate <- te_MOOC_bay_subdf * (150/300) + te_MOOC_nonbay_subdf *
    (150/300)
overall_ate_both_blocks_subdf_table <- matrix(c(overall_MIDS_subdfs_ate,
    overall_MOOC_subdfs_ate), ncol = 1, byrow = FALSE)
colnames(overall_ate_both_blocks_subdf_table) <- c("Overall Estimate of ATE via Method (2)")
rownames(overall_ate_both_blocks_subdf_table) <- c("MIDS", "MOOC")
overall_ate_both_blocks_subdf_table <- as.table(overall_ate_both_blocks_subdf_table)
overall_ate_both_blocks_subdf_table
```

```
##      Overall Estimate of ATE via Method (2)
## MIDS                                -0.0769
## MOOC                                -0.0769
```

Viewing the coefficients from the fully saturated regression model for the Bay block, we see that there is not statistical significance to any of our coefficients except for Health_Care. This tells us that Health_Care explains a significant portion of the variance in a positive vs negative response from a Data Science job within the Bay-area in the context of our experiment. However, we remind the reader that the applicant used in this experiment had a background dominated by healthcare and biomedical domain experiences. This would not generalize to the population of all MIDS students.

In order to obtain the estimate of the ATE for each of our treatments, we added coefficients from the table as detailed in the code cell above. Next, we obtained standard errors and 95% confidence intervals for these estimates. Calculating these estimates via Method (2) directly separates our blocks and obtains separate estimates for each block (Bay and Non-Bay). Previously we had pooled the data and included a dummy variable for our blocking variable of Bay. We confirm via Method (2) that all of the estimates are the same as they were in Method (1).

3) Estimating Treatment Effects With Difference In Means

For our third method to estimate the average treatment effect of MIDS and the average treatment effect of MOOC, we look at the estimates derived via difference in means.

```
# Bay ATE estimate for MIDS via difference in means
bay_ATE_diff_means_MIDS <- mean(experiment_dt$Binary_Response[experiment_dt$Block ==
    1 & experiment_dt$Treatment == 3]) - mean(experiment_dt$Binary_Response[experiment_dt$Block ==
    1 & experiment_dt$Treatment == 1])
# Non-Bay ATE estimate for MIDS via difference in means
nonbay_ATE_diff_means_MIDS <- mean(experiment_dt$Binary_Response[experiment_dt$Block ==
    2 & experiment_dt$Treatment == 3]) - mean(experiment_dt$Binary_Response[experiment_dt$Block ==
    2 & experiment_dt$Treatment == 1])
# Overall ATE estimate for MIDS via difference in means
MIDS_overall_ATE_diff_means <- (bay_ATE_diff_means_MIDS * (150/300)) +
    (nonbay_ATE_diff_means_MIDS * (150/300))

# Bay ATE estimate for MOOC via difference in means
bay_ATE_diff_means_MOOC <- mean(experiment_dt$Binary_Response[experiment_dt$Block ==
    1 & experiment_dt$Treatment == 2]) - mean(experiment_dt$Binary_Response[experiment_dt$Block ==
    1 & experiment_dt$Treatment == 1])
# Non-Bay ATE estimate for MOOC via difference in means
nonbay_ATE_diff_means_MOOC <- mean(experiment_dt$Binary_Response[experiment_dt$Block ==
    2 & experiment_dt$Treatment == 2]) - mean(experiment_dt$Binary_Response[experiment_dt$Block ==
    2 & experiment_dt$Treatment == 1])
# Overall ATE estimate for MOOC via difference in means
MOOC_overall_ATE_diff_means <- (bay_ATE_diff_means_MOOC * (150/300)) +
    (nonbay_ATE_diff_means_MOOC * (150/300))

# Make table to display the estimates for the average
# treatment effects
diff_means_ate_table <- matrix(c(bay_ATE_diff_means_MIDS, bay_ATE_diff_means_MOOC,
    nonbay_ATE_diff_means_MIDS, nonbay_ATE_diff_means_MOOC),
    ncol = 1, byrow = FALSE)
colnames(diff_means_ate_table) <- c("Estimate of ATE")
rownames(diff_means_ate_table) <- c("MIDS Bay", "MOOC Bay", "MIDS Non-Bay",
    "MOOC Non-Bay")
diff_means_ate_table <- as.table(diff_means_ate_table)
diff_means_ate_table
```

```
##              Estimate of ATE
## MIDS Bay               -0.06
## MOOC Bay               -0.06
## MIDS Non-Bay            0.02
## MOOC Non-Bay            0.02
```

```
# Make table to display overall estimate of ATEs including
# both blocks
overall_ate_both_blocks_diff_means_table <- matrix(c(MIDS_overall_ATE_diff_means,
    MOOC_overall_ATE_diff_means), ncol = 1, byrow = FALSE)
colnames(overall_ate_both_blocks_diff_means_table) <- c("Overall Estimate of ATE via Method (3)")
rownames(overall_ate_both_blocks_diff_means_table) <- c("MIDS",
    "MOOC")
overall_ate_both_blocks_diff_means_table <- as.table(overall_ate_both_blocks_diff_means_table)
overall_ate_both_blocks_diff_means_table
```

```
##      Overall Estimate of ATE via Method (3)
## MIDS                                  -0.02
## MOOC                                  -0.02
```

Of note, a limitation of using difference in means is that we are not able to obtain robust standard errors and confidence intervals. However, we are able to obtain estimates of our average treatment effects in order to compare them to those we obtained via our two regression methods above. The estimate of the ATE for MIDS in the Bay is -0.06. The estimate of the ATE for MOOC training in the Bay is also -0.06. This is a slightly negative treatment effect, and our previous work has already shown us that this treatment effect is not statistically significantly different from zero. Both MIDS and MOOC outside the Bay have an estimated average treatment effect of 0.02. Again, this is a slight positive effect, but our previous work showed that this is not statistically significantly different from zero. The overall estimate of the ATE for both MIDS and MOOC via difference in means is -0.02.

Discussion

There are several limitations to this study. The first is the very low response rate. Out of 300 applications, there were 8 total positive responses spread across the 6 conditions. This is not enough data to draw conclusions. One reason for this low response rate is the narrow time window between the applications and the end of the study. In some cases, this was as little as one week. Many jobs take more than one week to express interest in applicants. A longer time window would likely have increased this response rate. Secondly, this study only used a front-door approach to job applications. Many jobs only hire through internal referrals, and this contributed to the low response rate. Finally, the applicant was coming straight out of academic, from a healthcare background, with no data science or tech job experience. This candidate would see a different job response rate than a more traditional applicant.

We did not measure the level of manipulation in this study. The education section was in the upper left corner of the resume. However, we do not know to what extent employers read, noticed, or cared about this section when evaluating the applicant. There was also a potential for crossover. In other words, to have an internet presence, our applicant was a real person. The education section of LinkedIn was set to private, but there was one page from the University of Minnesota that listed the MIDS degree. An employer who received the control resume and was doing an internet search on the applicant would likely find this page, learn about the degree, and effectively receive the same treatment as control.

This study was designed to study the effect of a MIDS degree on passing first-screening for Data Science job applications. This is different than studying the benefit of a MIDS degree in the job market, and does not account for the benefits of the wide network for referrals, career services, or the additional jobs available to people with a Masters.

One surprise in this study was the high number of data science jobs that require a Masters or PhD, particularly in the Bay Area. In this study, we only applied for jobs that did not have this as a requirement. This meant we excluded far more jobs that we applied to. In future studies, this decision could be reexamined. One possible effect of a MIDS degree that we did not study was the jobs that are available by gaining a Masters degree.

Finally, there are concerns about generalizability. Our applicant has an MD, is coming straight from academia, and has no tech experience. This is unusual for a MIDS student, and are all factors that could lead to different response rates to job applications. Additionally, data science jobs that don't require Masters are generally entry level jobs. At least one employer expressed concern about the expected salary. This raises the possibility that the perception of a income expectation mismatch further lowered the response rate, and is likely to occur for an applicant without an MD.

10

Conclusion

In future iterations of this study, we look to address some of these shortcomings. Namely, with respect to response rate, we intend to make a more parsable resume, with specific highlights on keywords and terms that may pass through a first round machine sorter and increase the time window for a response following an application. A larger sample size of job applications, although time intensive, would also yield more power for our study. With a larger sample size and more power, we would also be capable of studying alternative outcomes including type of response (e.g. warmness of text within an email) or time to positive/negative response after an application is submitted.

Further generalizability of our study can be met by increasing the number of job applicants and controlling for different aspects of each applicant. Although the majority of jobs require a Master's Degree or PhD, we can also apply to these jobs and control for requirements with specific covariates.

In sum, we fail to find sufficient evidence to support the hypothesis that a MIDS degree will change the rate of passing first-screening for data science job applications. However, our findings are not statistically significant, and concerns about response rate, manipulation, and generalizability lead us to believe further study is necessary.

I appreciate the care that this team put into the design, execution, and analysis of this design. I'm sure that sending all those applications was tedious. While the model checks that you included were comprehensive for the purposes of a lab-notebook write up, for a report, I would come up a bunch. Also well, I might locate all of my feature engineering in one place, so the models can stand on their own.