

Growth Mindset on Student Achievements

Minh Thu Bui^a, Vindyani Herath^a

^aDepartment of Mathematics and Statistics, Boston University, Boston, USA

Abstract

A growth mindset is defined as the belief that intelligence can be developed and that people can become smarter through hard work, effective studying strategies, and help from the community. This project aims to assess the effectiveness of a growth mindset intervention in enhancing student achievement through a causal inference perspective. Unconfoundedness is the first assumption to consider for measuring causal effects with inverse probability weighting, Hajek, and doubly robust estimators. Due to the clustering nature of the dataset and potential unmeasured confounders, the causal forests method with clusters is considered to gain better insights into the causal effects. The nudge-like intervention method has a positive impact on students' measures of achievement and reinforces the idea that a growth mindset can be used to encourage students in their study progress.

1 Introduction

A growth mindset is defined as the belief that intelligence can be developed and that people can become smarter through hard work, effective study strategies, and help from the community [1]. From the National Study of Learning Mindsets, an experiment was conducted to assess the impact of the nudge-like intervention by encouraging a growth mindset in high school students' achievements. The objective of this study is to assess the effectiveness of mindset intervention in enhancing student achievement.

The dataset contains observational data from 10,391 students across 76 high schools with a simulated continuous outcome (Y) for a measure of achievement, a binary treatment variable (A) indicating receipt of the intervention, and ten covariates as described in Table 1.

Table 1: Description of variables in the study	
Outcome	A continuous measure of achievement
Treatment	A binary variable for receipt of intervention
StudentExpectation	Student's self-reported expectations for success in the future, a continuous variable, measured prior to random assignment
Race	Categorical variable for student race/ethnicity
Gender	Categorical variable for student identified gender
FirstGen	Categorical variable for student first-generation status
Urbancity	School-level categorical variable for urbanity of the school
FixedMindsets	School-level mean of students' fixed mindsets, reported prior to random assignment
AchievementLevels	School achievement level (measured by test scores and college preparation)
SchoolMinorityPercent	School racial/ethnic minority composition
SchoolPovertyPercent	School poverty concentration
SchoolSize	Total number of students in all four grade levels in the school

Section 2.1 of this paper focuses on exploratory data analysis, while section 2.2 explores average treatment effect estimation using four main approaches: Outcome Regression (OR) Estimator approach, Inverse Probability Weighting (IPW) approach, Hajek Estimator approach, and Doubly Robust (DR) Estimator approach. Section 2.3 of the paper discusses a novel approach known as 'Causal Forests' aimed at addressing issues related to clustered observations and selection bias.

2 Methods and Analysis

2.1 Exploratory Data Analysis:

In this section, we focus on including visualizations that provide us with insights into the data patterns and outcomes. As defined above, treatment A is a binary variable indicating whether the student received treatment or not. In the dataset, the number of students who received the intervention (3384) is only approximately half of the number of students who did not receive the intervention (7007). The boxplot below visualizes how the intervention might impact the outcome, which is the students' measurement of achievements, with the color blue representing no intervention and red means having intervention. Moreover, we also investigate the relationship between students' self-reported expectation of success and the intervention status.

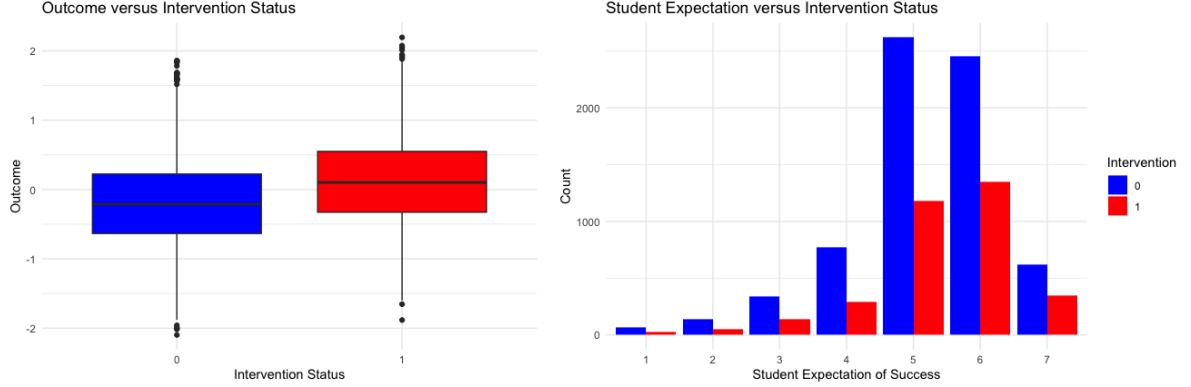


Figure 1: Outcomes (left) and Student Expectation (right) versus the Intervention Status.

From the plot on the left, we observe that the outcome mean for students who received the intervention is higher than that of students who did not receive the intervention. It is also noticeable that the outcome mean of the groups that did receive intervention is in the positive range compared to the other groups, which is in the negative range. This suggests that the intervention status might have a positive impact on the outcomes. From the plot on the right, we observe that a majority of the students self-reported that their expectation of success is between 5 and 6 ratings and this corresponds to the number of students who received and did not receive intervention are high in the two plot columns.

Next, we are interested in seeing whether the student's demographic information, specifically gender and first-generation status, can potentially impact the assignments of treatment. Note that in the *Gender* variable, 1 means Male, and 2 means Female. In the *FirstGen* variable, 0 means not being a first-generation student, and 1 means being a first-generation student.

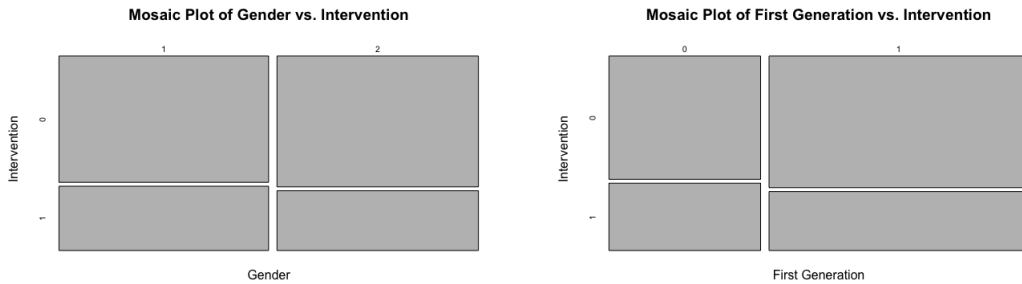


Figure 2: Gender (left) and First Generation Status (right) versus Intervention Status.

From Figure 2, the assignment of treatment based on gender does not seem to be drastically different and almost equal. However, for the right plot, the number of students who receive the intervention who are first-generation students is lower than that in the other group. This suggests that there might not be complete random in assignment of treatments to students in each school.

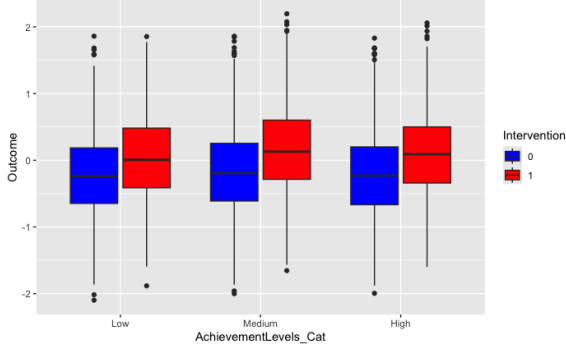


Figure 3: Schools' Achievement Levels versus Intervention Status and Outcomes.

We also investigate the relationship between schools' achievement levels and the outcomes given to two groups of students with the intervention and without the intervention in each school. Because the schools' achievement level variable is continuous, it was categorized into three groups: Low for schools with scores in the 25th percentile or lower, Middle for schools with scores that are in the range from 25th to 75th percentile, and High for schools with scores that are 75th percentile or higher. From Figure 3, we see that the results are consistent with the visualizations we obtained earlier for the mean outcomes for all of the levels: The mean outcomes for groups that received the

intervention are higher than the mean outcomes for groups that did not receive the intervention, regardless of the schools' achievement levels.

From the visualizations above, we can see that the assignment of treatments for students in different schools is not completely randomized, and with clustering based on schools in the dataset, there might be confounders that are not included in the dataset. Hence, there might be school-specific confounders that are affecting the causal effects. Moreover, with clustered data, traditional causal inference methods that do not account for the clusterings in the dataset cannot produce valid results for inference.

2.2 Average Treatment Effect (ATE) Estimation:

Under the three main identification assumptions: positivity, consistency, and conditional exchangeability, the average treatment effect, θ_{ATE} , is identified as

$$\theta_{ATE} = \mathbb{E}[\mathbb{E}[Y|A = 1, X]] - \mathbb{E}[\mathbb{E}[Y|A = 0, X]]. \quad (1)$$

This study uses 4 main approaches to estimate the causal effect of the intervention on the measure of achievement by calculating the average treatment effect (ATE): Outcome Regression (OR) estimator, Inverse probability weighting (IPW), Hajek estimator (Hajek), and Double robust estimator (DR). The simplest estimator for ATE is the OR estimator. The second approach, the IPW estimator does not require any modeling of the outcome mechanism and only requires modeling of the treatment assignment mechanism. The Hajek estimator addresses the main concern that we face with the IPW estimator which is not being invariant to the location transformation of the outcome by normalizing the weights. However, the approach introduces bias. The doubly robust (DR) estimation method combines both the OR estimation and IPW estimation to estimate the average causal effect. Table 2 illustrates the causal effect of the intervention on the measure of achievement by estimating the average treatment effect, their respective standard errors, and 95% bootstrap confidence intervals. While the OR estimator suggests a positive causal effect of the intervention on the measure of achievement, the other three approach a negative causal effect. As none of the 95% confidence intervals include zero, these estimates are considered significant. However, the traditional causal effect estimation methods might not provide valid results for inference as mentioned above in Section 2.1.

Table 2: Result reports for OR, IPW, Hajek, and DR estimators.

Method	ATE	Standard error	95% bootstrap confidence interval
OR Estimator	0.2788	0.0187	(0.2420, 0.3161)
Inverse Probability Weighting	-0.3348	0.0385	(-0.4309, -0.2832)
Hajek Estimator	-0.2119	0.0206	(-0.2397,-0.1907)
Doubly Robust Estimator	-0.3415	0.0740	(-0.5151, -0.2148)

However, our study faces two primary challenges. Firstly, the data under examination is categorized as observational rather than randomized. Figure 4 shows that students with a higher expectation of success are more likely to receive treatment, implying that the assignment of treatment is not randomized, confirming that we should treat this as an observational study. Another challenge is the clustered data

because from each school, multiple students are recruited to join the study. Thus, the causal forests method is introduced to measure causal effects in clustered datasets.

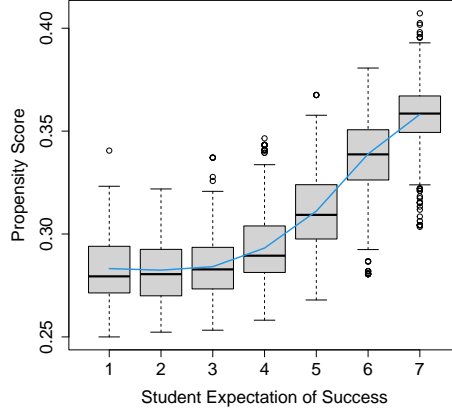


Figure 4: Propensity Score and Student Expectations.

2.3 Causal Forests

To examine the heterogeneity across different schools, we can estimate the conditional average treatment effect (CATE) for each subgroup. Additionally, the students included in this study are not independently sampled; instead, they are drawn from 76 randomly selected schools. Therefore, substantial heterogeneity exists across these schools. Such a situation could occur if there are unobserved features between schools that affect how well the treatment works. For example, some schools might have better leadership or students who are more open to the treatment. To make sure our findings apply to schools beyond the 76 we studied, we need to analyze the data carefully to deal with any unexplained differences between schools.

$$\theta_{CATE} = \mathbb{E}[Y_i^{(1)} - Y_i^{(0)} | X_i = x] \quad (2)$$

The validity of the estimate in (2) is conditional on being part of this subgroup.

Causal forests [2] which is a causal inference learning method that is an extension of random forests can be used to estimate CATE. The causal forests method is built from causal trees, where the causal trees learn a low-dimensional representation of treatment effect heterogeneity. This is simply the average of a large number of causal trees, where the trees differ due to sub-sampling. The splitting criterion of causal trees optimizes for finding splits associated with treatment heterogeneity. Causal forests can be implemented using the **grf** package in R. The implementation starts by fitting two separate regression forests for outcome regression and propensity score. It then makes out-of-bag predictions using the first two forests and uses them to grow a causal forest. The tuning parameters for causal forests were found by cross-validation. The final causal forest was trained by using those features that saw a reasonable number of splits in the first step. Table 3 illustrates the five most important variables that were used in implementing causal forests to estimate the causal effect of the intervention on the measure of achievement. The results are calculated by estimating the weighted sum of how many times a feature was split at each depth in the forest.

Table 3: Variable Importance	
Variable	Weighted Sum
Mean of students' fixed mindsets	0.1462
Student's self reported expectation	0.1251
School achievement level	0.1171
School size	0.1151
School racial/ethnic minority composition	0.1062

For the implementation of the package **grf**, we consider building causal forests with and without

clusters to capture the effect of clustering of schools on the outcome variable of students' measures of achievements.

Table 4: Causal Forests with and without clusters.

Method	ATE	Standard error	95% bootstrap confidence interval
Causal Forests with Clusters	0.2473	0.0202	(0.208, 0.286)
Causal Forests without Clusters	0.2530	0.0114	(0.231, 0.275)

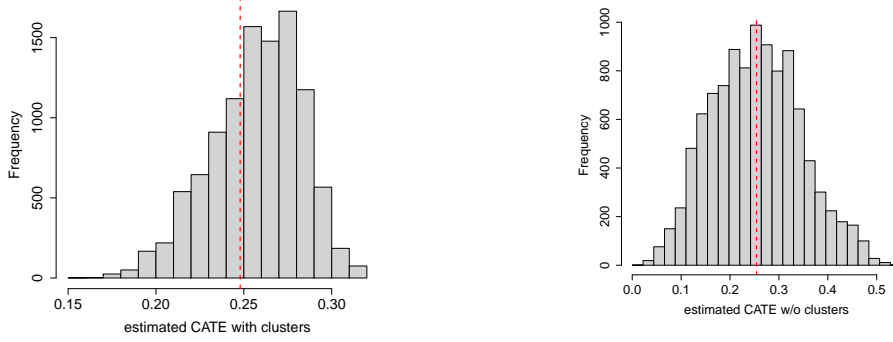


Figure 5: Estimated CATE with clusters (left) and without clusters (right).

From Table 4, it can be seen that the ATE estimates are not drastically different from each other. However, the range of the confidence intervals for causal forests without clusters are approximately half the size of that for causal forests with clusters. Moreover, Figure 5 indicates that the CATE estimates obtained without clusters exhibit more dispersion than those obtained with clusters. Hence, there might be treatment heterogeneity in the dataset that causes the differing range sizes for the two methods, with and without clusters.

To test for heterogeneity of causal forests, we perform a test calibration in R using the package `test.heterogeneity` on the data with clusters and without clusters. The test produces two results, one is `mean.prediction` and `differential.prediction`. A coefficient of 1 for `mean.prediction` indicates accurate mean forest prediction, while a coefficient of 1 for `differential.prediction` implies that the heterogeneity estimates from the forest is thoroughly calibrated. The p-value associated with the second estimate can test for heterogeneity presence with the p-value indicating significance means a rejection of the null hypothesis that there is no heterogeneity. In this case, the estimates of the p-value for `differential.prediction` is 0.3848 for the method with clusters and $3.031e - 05$ for the method without clusters. This indicates that when we consider causal forests with clusters, the problem with treatment heterogeneity is minimized and better calibrated than the causal forests without clusters. Hence, it suggests that there might be confounders that our current method does not account for, such as school-specific level confounders. For example, one school might have better facilities to implement the intervention process on students, leading to different causal effects compared to schools with lower ability to implement the intervention.

3 Conclusion

Given the design of the study with multiple participants recruited from each school and non-randomized assignments of treatment, the study is considered an observational study rather than a randomized study. Given the treatment heterogeneity, using the causal forests method to estimate Conditional Average Treatment Effects (CATE) proves more effective than using traditional causal measures such as OR, IPW, Hajek, and DR estimators. Causal forests offer a solution for addressing treatment heterogeneity and can be effectively used for causal effect estimation in observational studies..

With the causal forests with clusters in school IDs, we observe positive causal effects between the intervention and the students' achievement measures. Hence, the nudge-like intervention contributes to higher measures of students' achievements, thus proving that a growth mindset can indeed be instilled to develop and increase intelligence in students, contributing to better students' measurements of achievements.

References

- [1] Susan Athey and Stefan Wager. “Estimating treatment effects with causal forests: An application”. In: *Observational studies* 5.2 (2019), pp. 37–51.
- [2] Stefan Wager and Susan Athey. “Estimation and Inference of Heterogeneous Treatment Effects using Random Forests”. In: *Journal of the American Statistical Association* 113.523 (2018), pp. 1228–1242. DOI: [10 .1080/01621459 .2017 .1319839](https://doi.org/10.1080/01621459.2017.1319839). eprint: [https://doi.org/10 .1080/01621459 .2017 .1319839](https://doi.org/10.1080/01621459.2017.1319839). URL: <https://doi.org/10.1080/01621459.2017.1319839>.

Growth Mindset on Student Achievements

Minh Thu Bui, Vindyani Herath

Exploratory Data Analysis:

```
print(paste("The number of rows in the dataset is:", dim(data)[1]))
```

```
## [1] "The number of rows in the dataset is: 10391"
```

```
print(paste("The number of columns in the dataset is:", dim(data)[2]))
```

```
## [1] "The number of columns in the dataset is: 13"
```

```
print(paste("The number of students who received the intervention is:", sum(data$Intervention == 1)))
```

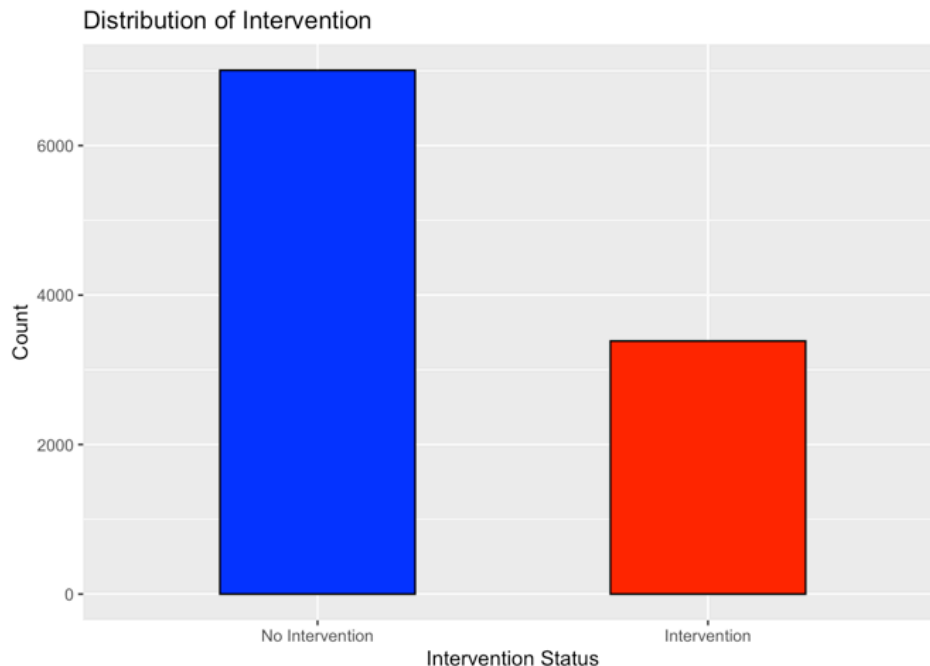
```
## [1] "The number of students who received the intervention is: 3384"
```

```
print(paste("The number of students who did not received the intervention is:", sum(data$Intervention == 0)))
```

```
## [1] "The number of students who did not received the intervention is: 7007"
```

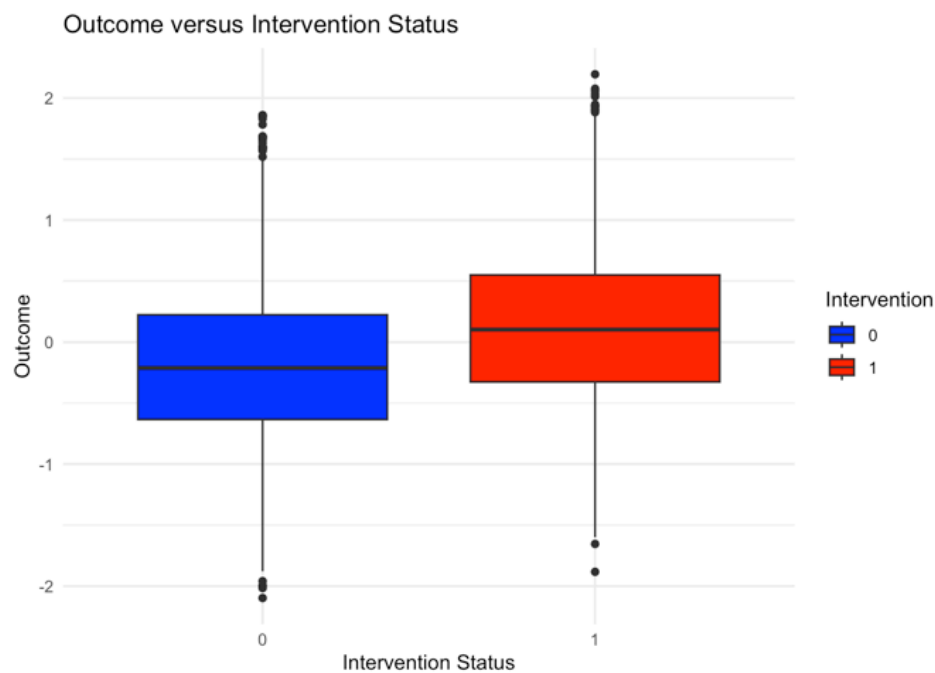
First, we want to see the distribution of people who received the intervention and who did not in the total dataset.

```
# Plot the number of students who receive intervention and the amount of people that did not.  
ggplot(data, aes(x = Intervention)) +  
  geom_bar(fill = c("blue", "red"), color = "black", width = 0.5) +  
  labs(x = "Intervention Status", y = "Count",  
       title = "Distribution of Intervention") +  
  scale_x_discrete(labels = c("No Intervention", "Intervention"))
```



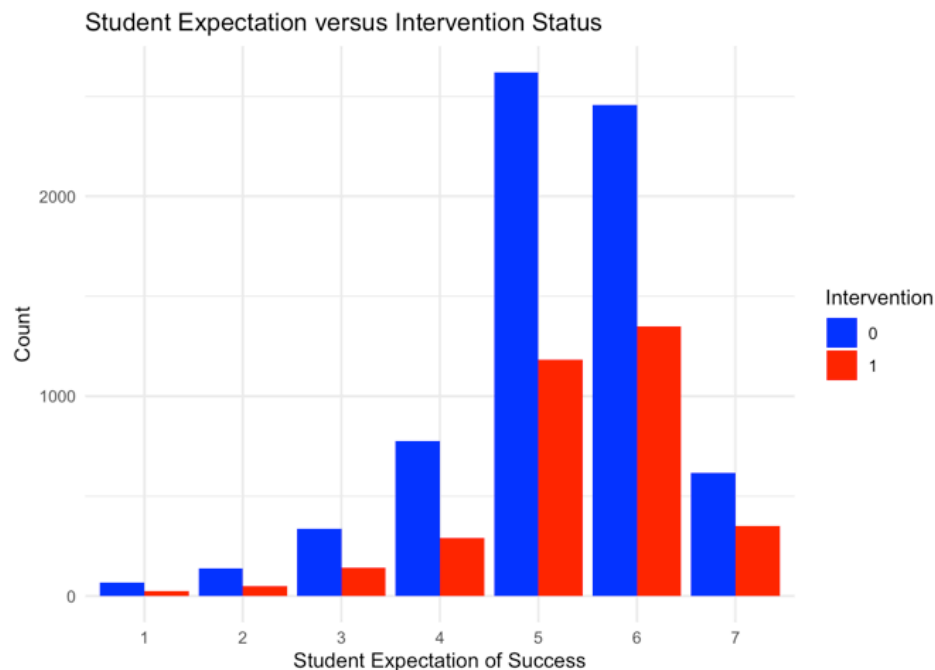
Next, we also want to investigate the relationship between the intervention status (whether they received it or not) and the outcome. It seems like the mean of the outcomes is higher and on the positive scale for people who received treatment while the mean is negative for the control group.

```
ggplot(data, aes(x = Intervention, y = Outcome, fill = Intervention)) +  
  geom_boxplot() +  
  labs(x = "Intervention Status", y = "Outcome", title = "Outcome versus Intervention Status") +  
  scale_fill_manual(values = c("blue", "red")) +  
  theme_minimal()
```



We also want to investigate whether the assignment of treatment is randomized or if there was some selection bias to consider in our analysis. First, we visualize to inspect whether students with a higher expectation of success appear to be more likely to receive treatment.

```
ggplot(data, aes(x = StudentExpectation, fill = Intervention)) +
  geom_bar(position = "dodge") +
  labs(x = "Student Expectation of Success", y = "Count",
       title = "Student Expectation versus Intervention Status") +
  scale_fill_manual(values = c("blue", "red")) +
  theme_minimal()
```



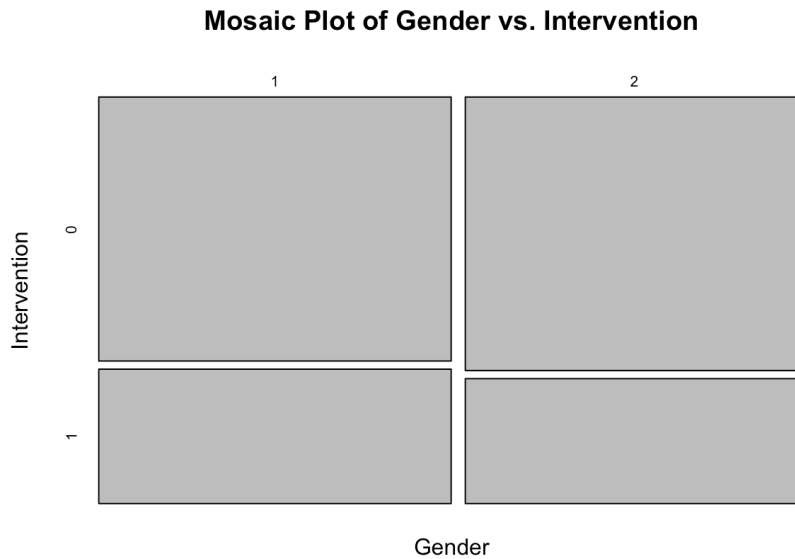
We also create a contingency table between the binary variable Gender and the binary variable for Intervention. We see that for Gender decoded as 1 (Male) tends to have higher chances of getting the treatment compared to the Gender decoded as 2 (Female).

```
table_data <- table(data$Gender, data$Intervention)
print(table_data)
```

```
##
##      0      1
## 1 3512 1788
## 2 3495 1596
```



```
mosaicplot(table_data, main="Mosaic Plot of Gender vs. Intervention", xlab="Gender", ylab="Intervention")
```

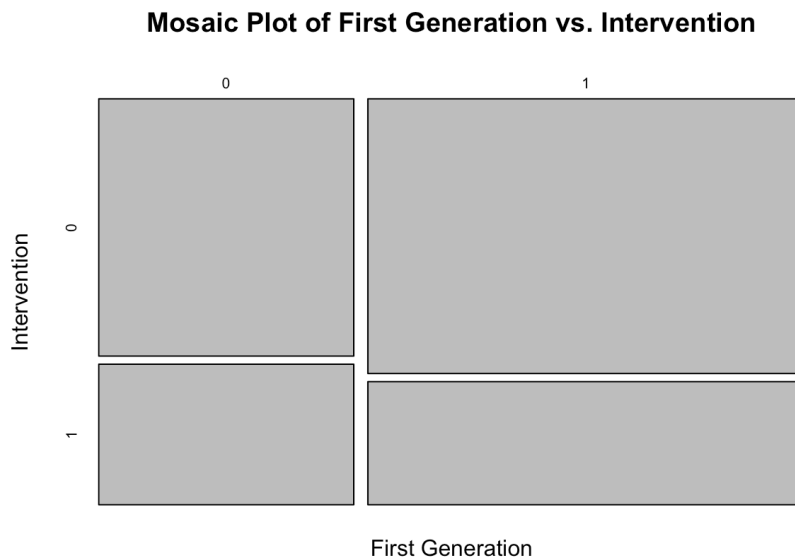


Then, the relationship between being first-generation versus being assigned the treatment shown in the table below. It seems like the randomization does occur here but there could be some sort of selection bias because the number of first-generation students receiving the treatment or not is almost double compared to the number of non-first generation students.

```
table_data <- table(data$FirstGen, data$Intervention)
print(table_data)
```

```
##
##      0      1
## 0 2480 1355
## 1 4527 2029
```

```
mosaicplot(table_data, main="Mosaic Plot of First Generation vs. Intervention", xlab="First Generation", ylab="Intervention")
```



```
#Checking whether school achievement level have a relationship with the intervention and outcome
#Achievement levels: low, 25th percentile or lower, middle, 25th - 75th percentile; high, 75th percentile or higher
summary(data$AchievementLevels)
```

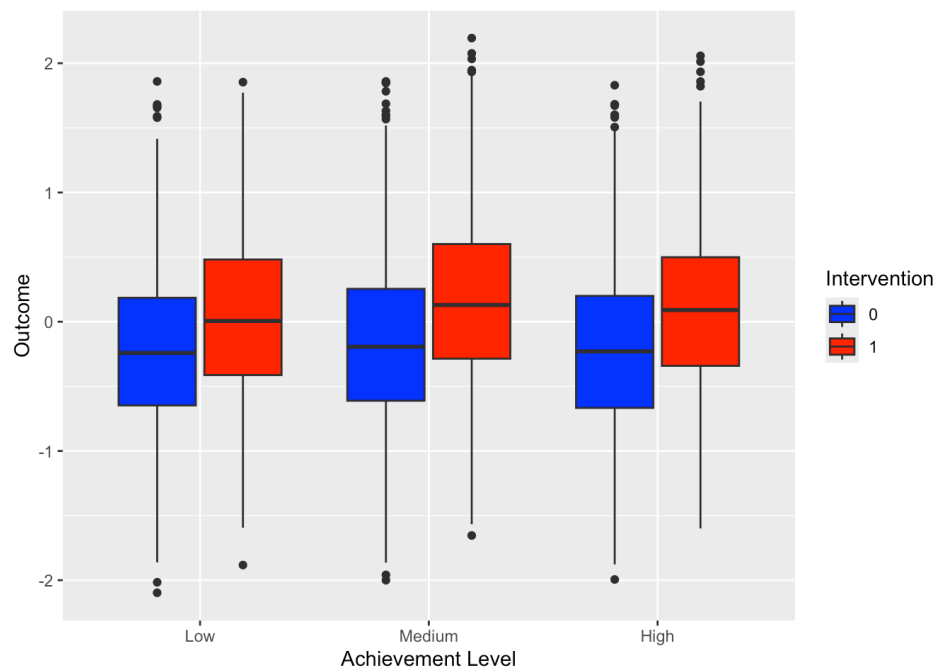
```
##      Min.   1st Qu.   Median     Mean  3rd Qu.    Max.
## -3.34782 -0.54451 -0.02251  0.05484  0.72684  2.17182
```

```
data$AchievementLevels_Cat <- ifelse(data$AchievementLevels > 0.72684, "High",
                                     ifelse(data$AchievementLevels < -0.54451, "Low", "Medium"))
data$AchievementLevels_Cat <- factor(data$AchievementLevels_Cat, levels = c("Low", "Medium", "High"))
table(data$AchievementLevels_Cat)
```

```
##
##      Low Medium   High
##    2532   5488   2371
```

```
# library
#library(ggplot2)

# grouped boxplot
ggplot(data, aes(x=AchievementLevels_Cat, y=Outcome, fill=Intervention)) +
  geom_boxplot() + scale_fill_manual(values = c("blue", "red")) + labs(x = "Achievement Level", y = "Outcome")
```



This dataset exhibits two methodological challenges. First, although the National Study itself was a randomized study, there seems to be some selection effects in the synthetic data used here. Second, the students in this study are not independently sampled; rather, they are all drawn from 76 randomly selected schools, and there appears to be considerable heterogeneity across schools. Such a situation could arise if there are unobserved school-level features that are important treatment effect modifiers; for example, some schools may have leadership teams who implemented the intervention better than others, or may have a student culture that is more receptive to the treatment [Athey and Wager, 2018].

Modeling:

```
set.seed(1)
sample <- sample.int(n = nrow(data), size = floor(0.5*nrow(data)), replace = F)
train <- data[sample, ]
test <- data[-sample, ]
```

Outcome Regression Approach:

```
compute_OR_ATE <- function(train_data, test_data){
  model1 <- lm(Outcome ~ Race + Gender + FirstGen + Urbanicity + FixedMindsets + AchievementLevels + SchoolMinorityPercent
              + SchoolPovertyPercent + SchoolSize, data = train_data[train_data$Intervention == 1,])
  model0 <- lm(Outcome ~ Race + Gender + FirstGen + Urbanicity + FixedMindsets + AchievementLevels + SchoolMinorityPercent
              + SchoolPovertyPercent + SchoolSize, data = train_data[train_data$Intervention == 0,])

  mu1 <- mean(as.matrix(predict(model1, newdata = test_data, type = "response")))
  mu0 <- mean(as.matrix(predict(model0, newdata = test_data, type = "response")))

  return(mu1 - mu0)
}
compute_OR_ATE(train, test)
```

```
## [1] 0.278849
```

Propensity Score Approach:

```
ate_OR <- function(train_data, test_data) {
  propensity_model <- glm(formula = Intervention ~ StudentExpectation + Race + Gender + FirstGen + Urbanicity +
    FixedMindsets + AchievementLevels + SchoolMinorityPercent + SchoolPovertyPercent + SchoolSize, family = binomial(
    link = "logit"), data = train_data)
  train_data$propensity_score <- predict(propensity_model, type = "response")

  propensity_treatment <- lm(Outcome ~ Race + Gender + FirstGen + Urbanicity + FixedMindsets + AchievementLevels + SchoolMinorityPercent
    + SchoolPovertyPercent + SchoolSize, data = train_data[train_data$Intervention == 1,])
  propensity_control <- lm(Outcome ~ Race + Gender + FirstGen + Urbanicity + FixedMindsets + AchievementLevels + SchoolMinorityPercent
    + SchoolPovertyPercent + SchoolSize, data = train_data[train_data$Intervention ==
    0,])
  test_data$propensity_score <- predict(propensity_model, newdata = test_data, type = "response")
  mu1 <- mean(as.matrix(predict(propensity_treatment, newdata = test_data)))
  mu0 <- mean(as.matrix(predict(propensity_control, newdata = test_data)))
  ATE_x = mu1 - mu0
  return(ATE_x)
}
ate_OR(train, test)
```

```
## [1] 0.278849
```

```
n_bootstrap <- 1000
# Bootstrap procedure
set.seed(123)
bootstrap <- function(train_data, test_data, n_bootstrap = 1000){
  results <- numeric(n_bootstrap)
  for (i in 1:n_bootstrap){
    bootstrap_sample <- train_data[sample(nrow(train_data), replace = TRUE), ]
    results[i] <- ate_OR(bootstrap_sample, test_data)
  }
  return(results)
}

bootstrap_results <- bootstrap(train, test, n_bootstrap)
#Standard Error
OR_SE <- sd(bootstrap_results)
OR_SE
```

```
## [1] 0.01867229
```

```
# Compute confidence intervals
confidence_interval <- quantile(bootstrap_results, c(0.025, 0.975))
confidence_interval
```

```
##      2.5%      97.5%
## 0.2420477 0.3161149
```

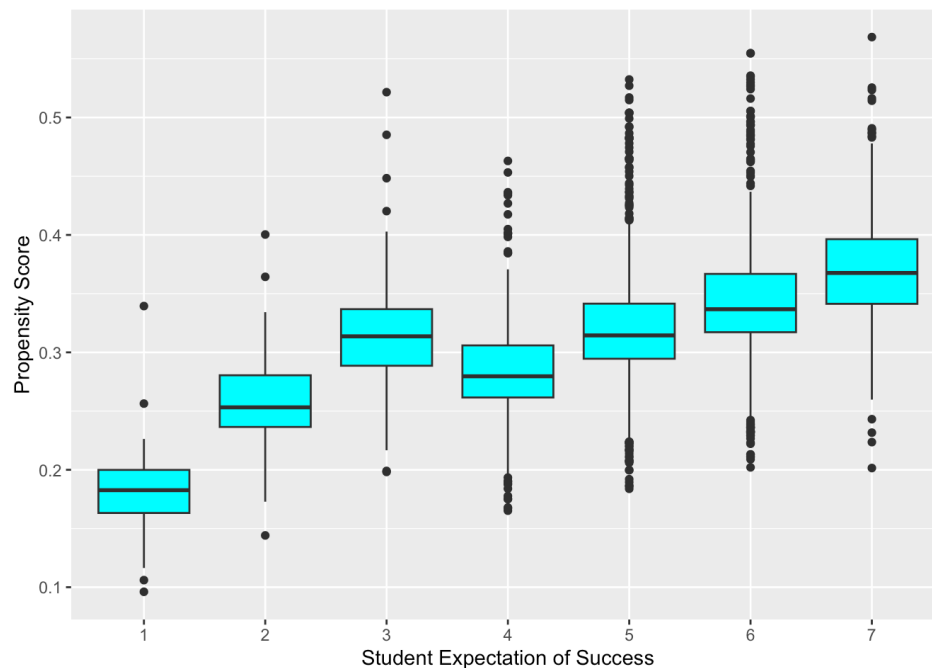
```
cat("Estimated ATE:", ate_OR(train,test), "\n", "95% Confidence Interval:", confidence_interval, "\n",
    "Standard Error:", OR_SE)
```

```
## Estimated ATE: 0.278849
## 95% Confidence Interval: 0.2420477 0.3161149
## Standard Error: 0.01867229
```

```
#Propensity Score
compute_propensity <- function(train_data, test_data){
  ps_model <- glm(formula = Intervention ~ StudentExpectation + Race + Gender
    + FirstGen + Urbanicity + FixedMindsets + AchievementLevels +
    SchoolMinorityPercent + SchoolPovertyPercent + SchoolSize,
    family = binomial(link = "logit"), data = train_data)

  #train_data$ps <- predict(ps_model, new_data = train_data, type = "response")
  test_data$ps <- predict(ps_model, newdata = test_data, type = "response")
  return(test_data$ps)
}

test$ps <- compute_propensity(train,test)
# grouped boxplot
ggplot(test, aes(x=StudentExpectation, y=ps)) +
  geom_boxplot(fill = "cyan") +
  labs(x = "Student Expectation of Success", y = "Propensity Score")
```



Inverse Weighting Approach:

```
# IPW estimator
test$Intervention <- as.numeric(test$Intervention)
compute_IPW_ATE <- function(train_data, test_data){
  ps_model <- glm(formula = Intervention ~ StudentExpectation + Race + Gender
    + FirstGen + Urbanicity + FixedMindsets + AchievementLevels +
    SchoolMinorityPercent + SchoolPovertyPercent + SchoolSize,
    family = binomial(link = "logit"), data = train_data)

  test_data$ps <- predict(ps_model, newdata = test_data, type = "response")
  m1_ipw <- mean((test_data$Intervention/ test_data$ps) * test_data$Outcome)
  m0_ipw <- mean(((1 - test_data$Intervention)/(1 - test_data$ps))*test_data$Outcome)
  return(m1_ipw - m0_ipw)
}

compute_IPW_ATE(train,test)
```

```
## [1] -0.3347782
```

```

set.seed(123)
bootstrap_ipw <- function(train_data, test_data, n_bootstrap = 5000){
  results <- numeric(n_bootstrap)
  for (i in 1:n_bootstrap){
    bootstrap_sample <- train_data[sample(nrow(train_data), replace = TRUE), ]
    results[i] <- compute_IPW_ATE(bootstrap_sample, test_data)
  }
  return(results)
}

bootstrap_results_ipw <- bootstrap_ipw(train, test, n_bootstrap)
IPW_SE <- sd(bootstrap_results_ipw)
IPW_SE

```

```
## [1] 0.03849028
```

```

#Confidence Intervals
IPW_CI <- quantile(bootstrap_results_ipw, c(0.025, 0.975))
IPW_CI

```

```

##          2.5%          97.5%
## -0.4308998 -0.2832002

```

Doubly Robust Estimation:

```

# DR estimator
test$Intervention <- as.numeric(test$Intervention)
compute_DR_ATE <- function(train_data, test_data){
  ps_model <- glm(formula = Intervention ~ StudentExpectation + Race + Gender
                  + FirstGen + Urbanicity + FixedMindsets + AchievementLevels +
                  SchoolMinorityPercent + SchoolPovertyPercent + SchoolSize,
                  family = binomial(link = "logit"), data = train_data)
  model1 <- lm(Outcome ~ Race + Gender + FirstGen + Urbanicity + FixedMindsets + AchievementLevels + SchoolMinorityPercent
              + SchoolPovertyPercent + SchoolSize, data = train_data[train_data$Intervention == 1,])
  model0 <- lm(Outcome ~ Race + Gender + FirstGen + Urbanicity + FixedMindsets + AchievementLevels + SchoolMinorityPercent
              + SchoolPovertyPercent + SchoolSize, data = train_data[train_data$Intervention == 0,])

  mu1 <- predict(model1, newdata = test_data, type = "response")
  mu0 <- predict(model0, newdata = test_data, type = "response")
  OR_est <- mean(mu1 - mu0)

  test_data$ps <- predict(ps_model, newdata = test_data, type = "response")
  M1 <- mean(test_data$Intervention*(test_data$Outcome - mu1)/test_data$ps)
  M2 <- mean(((1 - test_data$Intervention)* (test_data$Outcome - mu0))/(1 - test_data$ps))
  return(OR_est+ M1 - M2)
}
DR_ATE <- compute_DR_ATE(train, test)
DR_ATE

```

```
## [1] -0.3415459
```

```

bootstrap_dr <- function(train_data, test_data, n_bootstrap = 1000){
  results <- numeric(n_bootstrap)
  for (i in 1:n_bootstrap){
    bootstrap_sample <- train_data[sample(nrow(train_data), replace = TRUE), ]
    results[i] <- compute_DR_ATE(bootstrap_sample, test_data)
  }
  return(results)
}
set.seed(123)
bootstrap_results_dr <- bootstrap_dr(train, test, n_bootstrap)
DR_SE <- sd(bootstrap_results_dr)
DR_SE

```

```
## [1] 0.07402885
```

```
#Confidence Intervals
DR_CI <- quantile(bootstrap_results_dr, c(0.025, 0.975))
DR_CI
```

```
##          2.5%          97.5%
## -0.5150961 -0.2147740
```

Hajek Estimator:

```
hajek_func <- function(data1, data2) {
  hajek_model <- glm(formula = Intervention ~ StudentExpectation + Race + Gender
    + FirstGen + Urbanicity + FixedMindsets + AchievementLevels +
    SchoolMinorityPercent + SchoolPovertyPercent + SchoolSize,
    family = binomial(link = "logit"), data = data1)
  data1$hajek <- predict(hajek_model, type = "response")
  data2$hajek <- predict(hajek_model, newdata = data2, type = "response")
  mu1_hajek <- mean((data2$Intervention/ data2$hajek)/mean((data2$Intervention/ data2$hajek)) * data2$Outcome)
  mu0_hajek <- mean((1 - data2$Intervention)/ (1-data2$hajek))/mean((1 - data2$Intervention)/ (1-data2$hajek))
  * data2$Outcome)
  ATE_x_hajek = mu1_hajek - mu0_hajek
  return(ATE_x_hajek)
}
ATE_x_hajek <- hajek_func(train, test)
print(paste("ATE for Hajek estimator is:", ATE_x_hajek))
```

```
## [1] "ATE for Hajek estimator is: -0.211971949529218"
```

```
suppressWarnings (bootstrap_hajek <- replicate(n_bootstrap , {sample_boot <- train[sample(nrow(train), replace =
TRUE), ]
  hajek_func(sample_boot, test)}))
hajek_ci <- quantile(bootstrap_hajek, c(0.025, 0.975))
hajek_ci
```

```
##          2.5%          97.5%
## -0.2396494 -0.1906684
```

```
hajek_se <- sd(bootstrap_hajek)
print(paste("Standard errors for Hajek estimator is:", hajek_se))
```

```
## [1] "Standard errors for Hajek estimator is: 0.0206401801665278"
```

Sensitivity Analysis

```

eta0 <- c(1/2, 1/1.7, 1/1.5, 1/1.3, 1, 1.3, 1.5, 1.7, 2)
etal <- c(1/2, 1/1.7, 1/1.5, 1/1.3, 1, 1.3, 1.5, 1.7, 2)
ATE_calculation <- function(train_data, test_data, mu1, mu0, eta0_val, etal_val) {
  test_data$Intervention <- as.numeric(test_data$Intervention)
  treatment <- mean(test_data$Intervention * mu1 + (1 - test_data$Intervention)*(mu1/etal_val))
  control <- mean(test_data$Intervention * mu0 * eta0_val + (1 - test_data$Intervention)*mu0)
  ATE_val <- treatment - control
  return(ATE_val)
}

sensitivity_analysis_ATE <- function(train_data, test_data, eta0_val, etal_val){
  modell <- glm(Outcome ~ Race + Gender + FirstGen + Urbanicity + FixedMindsets + AchievementLevels + SchoolMinorityPercent
    + SchoolPovertyPercent + SchoolSize, family = gaussian, data = train_data[train_data$Intervention
    == 1,])
  model0 <- glm(Outcome ~ Race + Gender + FirstGen + Urbanicity + FixedMindsets + AchievementLevels + SchoolMinorityPercent
    + SchoolPovertyPercent + SchoolSize, family = gaussian, data = train_data[train_data$Intervention
    == 0,])
  mu1_sen <- mean(as.matrix(predict(modell, newdata = test_data, type = "response")))
  mu0_sen <- mean(as.matrix(predict(model0, newdata = test_data, type = "response")))
  ATE_vals <- matrix(NA, length(eta0_val), length(etal_val), dimnames = list(eta0_val, etal_val))

  for (i in 1:length(eta0)) {
    for (j in 1:length(etal)) {
      ATE_vals[i, j] <- ATE_calculation(train_data, test_data, mu1_sen, mu0_sen, eta0_val[i], etal_val[j])
    }
  }
  return(ATE_vals)
}
suppressWarnings(ATE_values <- sensitivity_analysis_ATE(train, test, eta0, etal))
#View(ATE_values)

n_bootstrap <- 1000
eta0_str <- as.character(eta0)
etal_str <- as.character(etal)

bootstrap_results <- array(dim = c(length(eta0), length(etal), n_bootstrap),
  dimnames = list(eta0_val = eta0_str, etal_val = etal_str, Sample = NULL))
set.seed(123)
suppressWarnings(for (b in 1:n_bootstrap) {
  sample_boot <- train[sample(nrow(train), replace = TRUE), ]
  ATE_vals_boot <- sensitivity_analysis_ATE(sample_boot, test, eta0, etal)
  bootstrap_results[,b] <- ATE_vals_boot
})
se_matrix <- apply(bootstrap_results, c(1, 2), sd)
#View(se_matrix)

library(knitr)
library(kableExtra)

```

```

##
## Attaching package: 'kableExtra'

```

```

## The following object is masked from 'package:dplyr':
##
##   group_rows

```

```

kable(ATE_values, caption = "ATE estimates") %>%
  kable_styling(bootstrap_options = c("striped", "scale_down"))

```

ATE estimates

	0.5	0.588235294117647	0.666666666666667	0.769230769230769	1	1.3	1.5	1.7
0.5	0.1277089	0.1373063	0.1437045	0.1501028	0.1597001	0.1670827	0.1703638	0.1728729
0.588235294117647	0.1487352	0.1583326	0.1647308	0.1711290	0.1807264	0.1881089	0.1913901	0.1938992
0.666666666666667	0.1674252	0.1770226	0.1834208	0.1898190	0.1994164	0.2067990	0.2100801	0.2125892
0.769230769230769	0.1918660	0.2014634	0.2078616	0.2142598	0.2238572	0.2312398	0.2345209	0.2370300

1	0.2468578	0.2564551	0.2628534	0.2692516	0.2788490	0.2862315	0.2895127	0.2920218
1.3	0.3183471	0.3279444	0.3343427	0.3407409	0.3503383	0.3577208	0.3610020	0.3635111
1.5	0.3660066	0.3756040	0.3820022	0.3884005	0.3979978	0.4053804	0.4086615	0.4111706
1.7	0.4136662	0.4232635	0.4296618	0.4360600	0.4456573	0.4530399	0.4563211	0.4588302
2	0.4851555	0.4947528	0.5011511	0.5075493	0.5171466	0.5245292	0.5278104	0.5303195

```
kable(se_matrix, caption = "Bootstrapped Standard Errors") %>%
  kable_styling(bootstrap_options = c("striped", "scale_down"))
```

Bootstrapped Standard Errors

	0.5	0.588235294117647	0.666666666666667	0.769230769230769	1	1.3	1.5	1.7
0.5	0.0111221	0.0125688	0.0135408	0.0145174	0.0159889	0.0171253	0.0176313	0.0180187
0.588235294117647	0.0115551	0.0129577	0.0139051	0.0148601	0.0163041	0.0174222	0.0179209	0.0183029
0.666666666666667	0.0120206	0.0133780	0.0142998	0.0152321	0.0166467	0.0177454	0.0182361	0.0186123
0.769230769230769	0.0127291	0.0140225	0.0149072	0.0158065	0.0171774	0.0182467	0.0187255	0.0190929
1	0.0146522	0.0157977	0.0165937	0.0174113	0.0186723	0.0196660	0.0201133	0.0204576
1.3	0.0176202	0.0185935	0.0192806	0.0199948	0.0211105	0.0220007	0.0224042	0.0227159
1.5	0.0197862	0.0206635	0.0212877	0.0219403	0.0229670	0.0237917	0.0241671	0.0244577
1.7	0.0220510	0.0228468	0.0234164	0.0240145	0.0249609	0.0257254	0.0260746	0.0263452
2	0.0255758	0.0262720	0.0267733	0.0273024	0.0281449	0.0288299	0.0291441	0.0293881

Causal Forests:

From our dataset, notice that all of the observations are pooled from uneven clusters based on school ID. Thus, this will change our inferential approach as in finding an optimal way to quantify the causal effects accurately given the information. From the paper, for example, in our setting, do we want to fit a model that accurately reflects heterogeneity in our available sample of $J = 76$ schools, or a model that will generalize to students from other schools also? Should we give more weight in our analysis to schools from which we observe more students? The approach they choose in the paper is to assume that we want a predictive model that generalizes to more than J schools with equal weights to any new school added to the dataset. In other words, we want a predictive model that can predict the causal effect when we add a new observation from a new school to the data.

```
library(grf)
```



```

data = read.csv("synthetic_data.csv")
data$schoolid = factor(data$schoolid)

names(data) <- c("SchoolID", "Intervention", "Outcome", "StudentExpectation", "Race", "Gender", "FirstGen",
                "Urbanicity", "FixedMindsets", "AchievementLevels", "SchoolMinorityPercent", "SchoolPovertyPercent", "SchoolSize")

DF = data[,-1]
school.id = as.numeric(data$SchoolID)

school.mat = model.matrix(~ SchoolID + 0, data = data)
school.size = colSums(school.mat)

# It appears that school ID does not affect pscore. So ignore it in modeling, and just treat it as source of per-
cluster error.
w.lm = glm(Intervention ~ ., data = data[,-3], family = binomial)
#summary(w.lm)

W = DF$Intervention
Y = DF$Outcome
X.raw = DF[,-(1:2)]

Race.exp = model.matrix(~ factor(X.raw$Race) + 0)
Urbanicity.exp = model.matrix(~ factor(X.raw$Urbanicity) + 0)

X = cbind(X.raw[, -which(names(X.raw) %in% c("Race", "Urbanicity"))], Race.exp, Urbanicity.exp)

#
# Grow a forest. Add extra trees for the causal forest.
# Y: outcome (Outcome), W: treatment (Intervention)

Y.forest = regression_forest(X, Y, clusters = school.id, equalize.cluster.weights = TRUE)
Y.hat = predict(Y.forest)$predictions
W.forest = regression_forest(X, W, clusters = school.id, equalize.cluster.weights = TRUE)
W.hat = predict(W.forest)$predictions

cf.raw = causal_forest(X, Y, W,
                      Y.hat = Y.hat, W.hat = W.hat,
                      clusters = school.id,
                      equalize.cluster.weights = TRUE)
varimp = variable_importance(cf.raw)
selected.idx = which(varimp > mean(varimp))
sort(varimp); selected.idx

```

```

## [1] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
## [6] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0007521587 0.0011291143
## [11] 0.0017124521 0.0051428981 0.0055631918 0.0129009937 0.0150802038
## [16] 0.0155414239 0.0175869456 0.0179558106 0.0260828633 0.0314975582
## [21] 0.0363396990 0.0375087018 0.0414399149 0.0993921531 0.1083040050
## [26] 0.1124552250 0.1218523419 0.1325341354 0.1592282098

```

```
## [1] 1 2 4 5 6 7 8 9 13
```

```

cf = causal_forest(X[,selected.idx], Y, W,
                  Y.hat = Y.hat, W.hat = W.hat,
                  clusters = school.id,
                  equalize.cluster.weights = TRUE,
                  tune.parameters = "all")

```

```

# Tau is average treatment effect
tau.hat = predict(cf)$predictions
mean(tau.hat)

```

```
## [1] 0.2579778
```

```

#
# Estimate ATE
#
ATE = average_treatment_effect(cf)
ATE

```

```
## estimate std.err
## 0.24848322 0.02030945
```

```
paste("95% CI for the ATE:", round(ATE[1], 3),
      "+/-", round(qnorm(0.975) * ATE[2], 3))
```

```
## [1] "95% CI for the ATE: 0.248 +/- 0.04"
```

```
print(paste("Confidence interval for ATE is (0.208, 0.286)"))
```

```
## [1] "Confidence interval for ATE is (0.208, 0.286)"
```

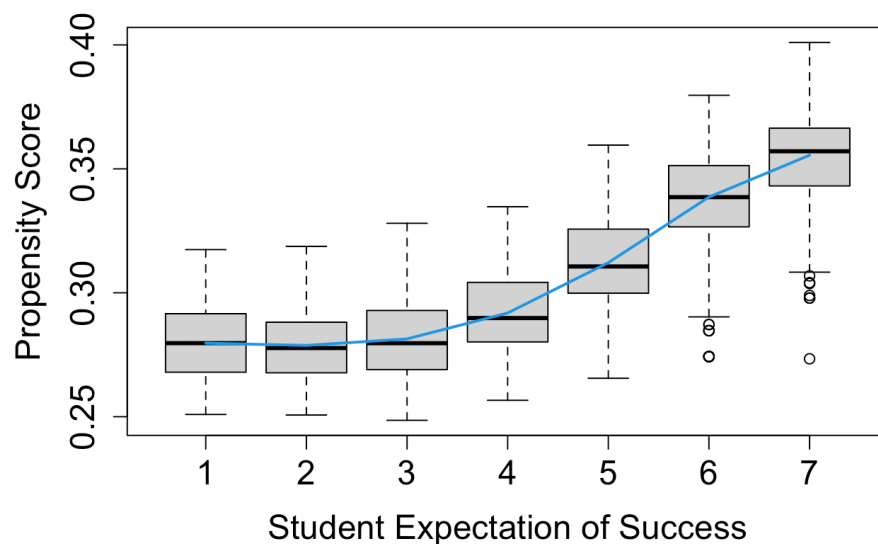
```
test_calibration(cf)
```

```
##
## Best linear fit using forest predictions (on held-out data)
## as well as the mean forest prediction as regressors, along
## with one-sided heteroskedasticity-robust (HC3) SEs:
##
##               Estimate Std. Error t value Pr(>t)
## mean.forest.prediction    0.998741   0.082805  12.061 <2e-16 ***
## differential.forest.prediction 0.385498   0.639263   0.603 0.2732
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

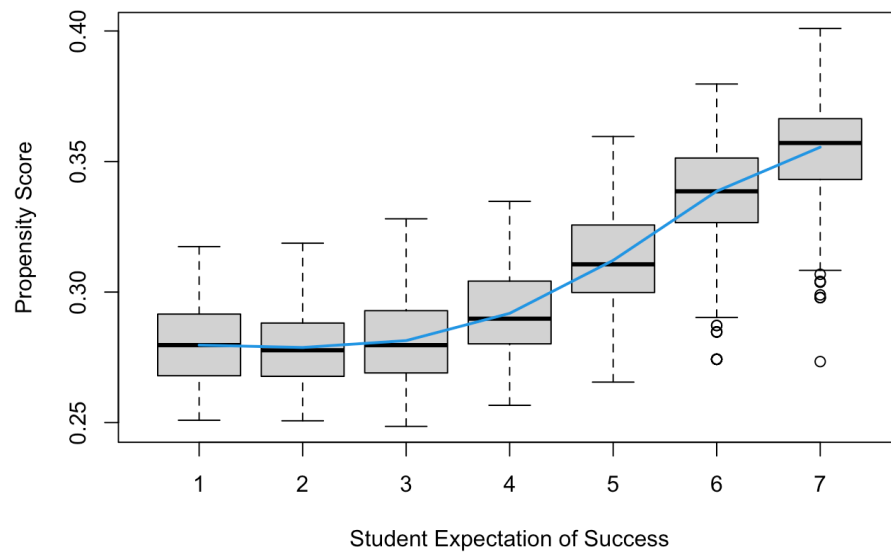
```
#
# Look at variation in propensity scores
#

DF = X
DF$W.hat = cf$W.hat

pardef = par(mar = c(5, 4, 4, 2) + 0.5, cex.lab=1.5, cex.axis=1.5, cex.main=1.5, cex.sub=1.5)
boxplot(W.hat ~ StudentExpectation, data = DF, ylab = "Propensity Score", xlab = "Student Expectation of Success")
lines(smooth.spline(X$StudentExpectation, cf$W.hat), lwd = 2, col = 4)
```

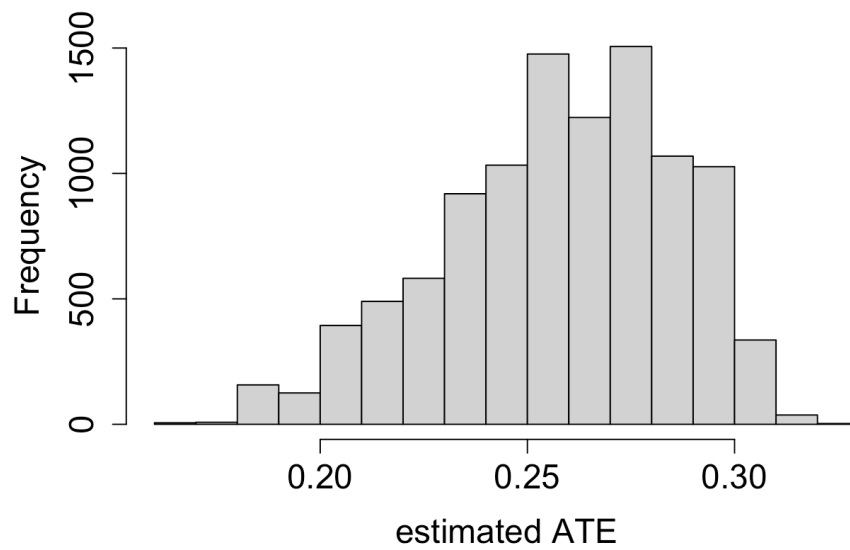


```
boxplot(W.hat ~ StudentExpectation, data = DF, ylab = "Propensity Score", xlab = "Student Expectation of Success")
lines(smooth.spline(X$StudentExpectation, cf$W.hat), lwd = 2, col = 4)
```



```
#
# Make some plots...
#

pardef = par(mar = c(5, 4, 4, 2) + 0.5, cex.lab=1.5, cex.axis=1.5, cex.main=1.5, cex.sub=1.5)
hist(tau.hat, xlab = "estimated ATE", main = "")
```



Causal Forest non-clustering-

robustness:

```
#
# Analysis ignoring clusters
#

cf.noclust = causal_forest(X[,selected.idx], Y, W,
                           Y.hat = Y.hat, W.hat = W.hat,
                           tune.parameters = "all")

ATE.noclust = average_treatment_effect(cf.noclust)
ATE.noclust
```

```
## estimate std.err
## 0.25383522 0.01144392
```

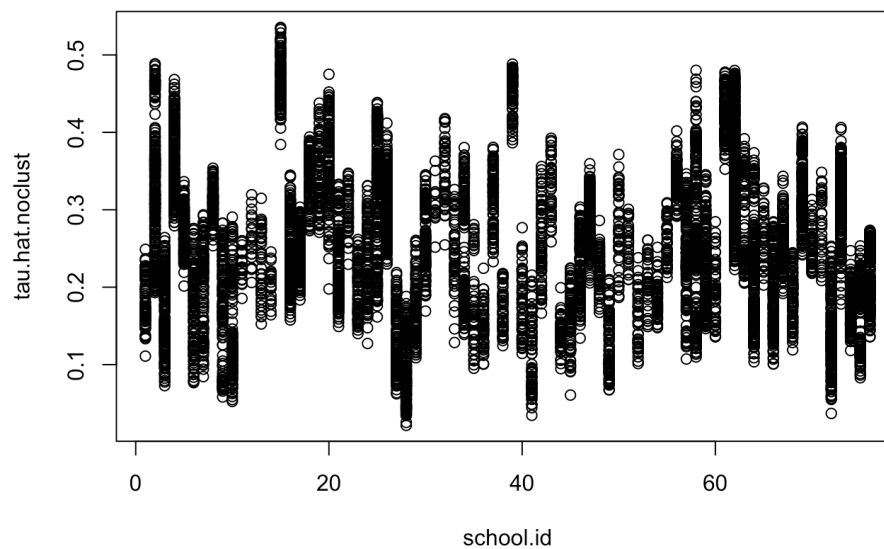
```
paste("95% CI for the ATE:", round(ATE.noclust[1], 3),
      "+/-", round(qnorm(0.975) * ATE.noclust[2], 3))
```

```
## [1] "95% CI for the ATE: 0.254 +/- 0.022"
```

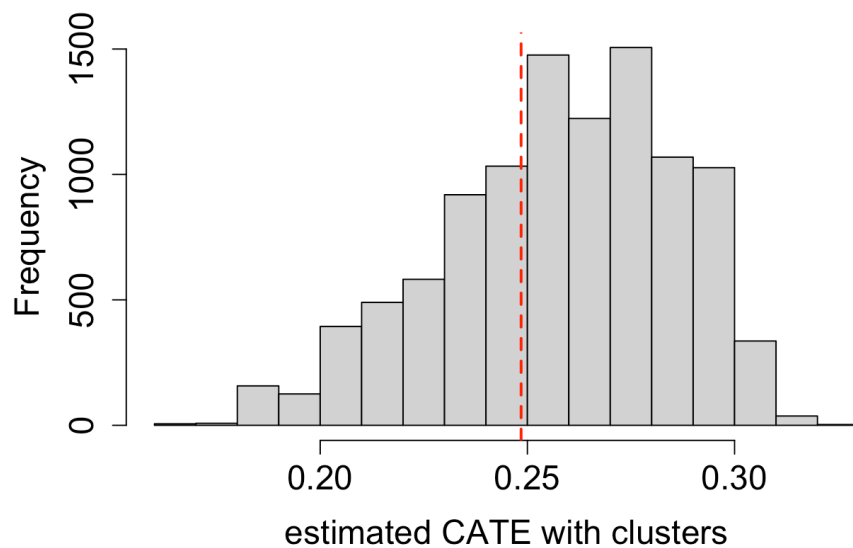
```
test_calibration(cf.noclust)
```

```
##
## Best linear fit using forest predictions (on held-out data)
## as well as the mean forest prediction as regressors, along
## with one-sided heteroskedasticity-robust (HC3) SEs:
##
##
##               Estimate Std. Error t value    Pr(>t)
## mean.forest.prediction      1.008263   0.044874 22.4687 < 2.2e-16 ***
## differential.forest.prediction 0.506361   0.122926  4.1192 1.915e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

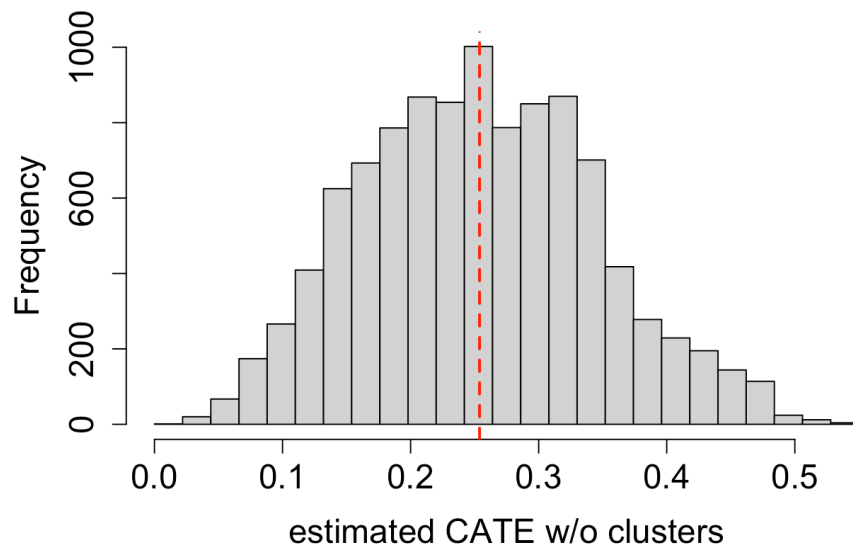
```
tau.hat.noclust = predict(cf.noclust)$predict
plot(school.id, tau.hat.noclust)
```



```
pardef = par(mar = c(5, 4, 4, 2) + 0.5, cex.lab=1.5, cex.axis=1.5, cex.main=1.5, cex.sub=1.5)
hist(tau.hat, xlab = "estimated CATE with clusters", main = "")
abline(v = ATE[1], col = 'red', lwd = 2, lty = 'dashed')
```



```
pardef = par(mar = c(5, 4, 4, 2) + 0.5, cex.lab=1.5, cex.axis=1.5, cex.main=1.5, cex.sub=1.5)
hist(tau.hat.noclust, xlab = "estimated CATE w/o clusters", main = "", breaks = seq(-0.0, 0.55, by = 0.55 / 25))
abline(v = ATE.noclust[1], col = 'red', lwd = 2, lty = 'dashed')
```



```
# Comparing the test_calibration
test_calibration(cf)
```

```
##
## Best linear fit using forest predictions (on held-out data)
## as well as the mean forest prediction as regressors, along
## with one-sided heteroskedasticity-robust (HC3) SEs:
##
##               Estimate Std. Error t value Pr(>t)
## mean.forest.prediction    0.998741   0.082805  12.061 <2e-16 ***
## differential.forest.prediction 0.385498   0.639263   0.603 0.2732
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
test_calibration(cf.noclust)
```

```
##
## Best linear fit using forest predictions (on held-out data)
## as well as the mean forest prediction as regressors, along
## with one-sided heteroskedasticity-robust (HC3) SEs:
##
##
##          Estimate Std. Error t value    Pr(>t)
## mean.forest.prediction      1.008263   0.044874 22.4687 < 2.2e-16 ***
## differential.forest.prediction 0.506361   0.122926  4.1192 1.915e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```