

# HW1 Part 1

Sibei Zhang, Richard Tang & Alexander Quispe

2022-04-17

```
setwd('/Users/chenmingtang/Documents/GitHub/EC0293')
star <- read_sav("/Users/chenmingtang/Documents/GitHub/EC0293/data/Project_STAR/PROJECT_STAR/STAR_StudentData.sav")

## Failed to find G1READ_A
## Failed to find G1MATH_A
## Failed to find G2READ_A
## Failed to find G2MATH_A
## Failed to find G3READ_A
## Failed to find G3MATH_A

names(star)<-tolower(names(star))
options(stringAsFactors=FALSE)

star <- star[!(star$g1classtype== 3 ),]
star$g1classtype[star$g1classtype == 1] <- 1
star$g1classtype[star$g1classtype == 2] <- 0
```

## 1. average treatment effect in the randomized experiment

```
#ate
ate_est <- mean(star$g1tmathss[star$g1classtype==1], na.rm = TRUE) - mean(star$g1tmathss[star$g1classtype==0], na.rm = TRUE)
ate_est

## [1] 13.39563

#confidence intervals
ate_se <- sqrt(var(star$g1tmathss[star$g1classtype == 1], na.rm = TRUE) / sum(star$g1classtype == 1, na.rm = TRUE))
ate_se

## [1] 1.297031

print(paste0("95% CI: ", round(ate_est),
" +/- ", round(1.96 * ate_se)))

## [1] "95% CI: 13 +/- 3"
```

## 2 Describe your method(s) for systematically deleting some observations as a function of X's

```
names(star)[names(star) == 'g1classtype'] <- "W"
# copying old dataset, just in case
data.exp <- star
data <- star

# defining the group that we will be dropped with some high probability
grp <- ((data$W == 1) & # if treated AND...
      (
        (data$race == 2) | # belongs an older group OR
        (data$g4surban == 3) # more conservative
      )) | # OR
      ((data$W == 0) & # if untreated AND...
      (
        (data$race == 1) | # belongs a younger group OR
        (data$g4surban == 4) # more liberal
      ))

# Individuals in the group above have a small chance of being kept in the sample
prob.keep <- ifelse(grp, .15, .85)
keep.idx <- as.logical(rbinom(n=nrow(data), prob=prob.keep, size = 1))
```

```
## Warning in rbinom(n = nrow(data), prob = prob.keep, size = 1): NAs produced
```

```
# Dropping
star <- data[keep.idx,]
```

### 2.1 new difference-in-means point estimate of the treatment effect is significantly different

```
#ate
ate_est <- mean(star$g1tmathss[star$W==1], na.rm = TRUE) - mean(star$g1tmathss[star$W==0], na.rm = TRUE)
ate_est
```

```
## [1] 18.00031
```

```
#confidence intervals
ate_se <- sqrt(var(star$g1tmathss[star$W == 1], na.rm = TRUE) / sum(star$W == 1, na.rm = TRUE) + var(star$g1tmathss[star$W == 0], na.rm = TRUE) / sum(star$W == 0, na.rm = TRUE))
ate_se
```

```
## [1] 2.949309
```

```
print(paste0("95% CI: ", round(ate_est),
" +/- ", round(1.96 * ate_se)))
```

```
## [1] "95% CI: 18 +/- 6"
```

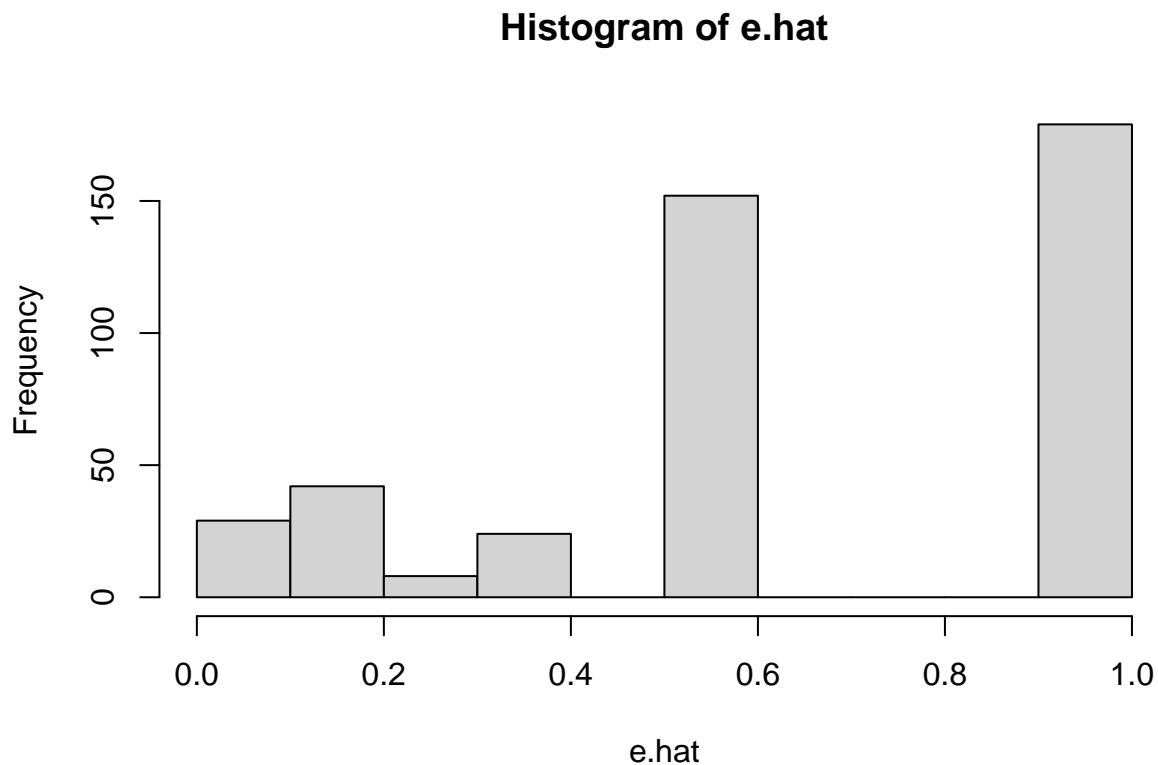
## 2.2 Propensity Score

```
star_2 <- star[complete.cases(star$gender, star$race, star$flagsgk, star$gktrace, star$g4surban, star$W),]

# Additional covariates
covariates <- c("gender", "race", "flagsgk", "gktrace", "g4surban")

# Estimate the propensity score e(X) via logistic regression using splines
fmla <- as.formula(paste0("~", paste0("bs(", covariates, ", df=3)", collapse="+")))
Z <- star_2$W
Y <- star_2$g1tmathss
XX <- model.matrix(fmla, star_2)
logit <- cv.glmnet(x=XX, y=Z, family="binomial")
e.hat <- predict(logit, XX, s = "lambda.min", type="response")

# Histogram
hist(e.hat)
```



### 3.1 The difference in means estimator (which would be appropriate in an RCT).

```
#ate
ate_est <- mean(star_2$gitmathss[star_2$W==1], na.rm = TRUE) - mean(star_2$gitmathss[star_2$W==0], na.rm = TRUE)

ate_est

## [1] 18.38701

#confidence intervals
ate_se <- sqrt(var(star_2$gitmathss[star_2$W == 1], na.rm = TRUE) / sum(star_2$W == 1, na.rm = TRUE) + var(star_2$gitmathss[star_2$W == 0], na.rm = TRUE) / sum(star_2$W == 0, na.rm = TRUE))

ate_se

## [1] 3.983468

print(paste0("95% CI: ", round(ate_est),
" +/- ", round(1.96 * ate_se)))

## [1] "95% CI: 18 +/- 8"
```

### 3.2 Simple linear regression, i.e., fit $Y \sim X + W$ and interpret the W-coefficient as the ATE.

```
fmla <- gitmathss ~ (W + gender + race + flagsgk + gktrace+ g4surban)
regbasic <- lm(fmla, data=star_2)
summary(regbasic)

##
## Call:
## lm(formula = fmla, data = star_2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -107.488  -26.209   -3.021   24.020  130.621
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  582.3855    16.1110   36.148  < 2e-16 ***
## W             10.6685     4.6338    2.302  0.02180 *
## gender       -11.1091     3.8383   -2.894  0.00400 **
## race         -17.2153     5.6297   -3.058  0.00237 **
## flagsgk             NA           NA      NA      NA
## gktrace         4.8977     6.8855    0.711  0.47728
## g4surban       -0.8236     2.6656   -0.309  0.75749
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.91 on 428 degrees of freedom
## Multiple R-squared:  0.0853, Adjusted R-squared:  0.07461
## F-statistic: 7.982 on 5 and 428 DF,  p-value: 3.297e-07
```

```
ols <- lm(fmla, data=star_2)
coeftest(ols, vcov=vcovHC(ols, type='HC2'))[2,]
```

```
##      Estimate Std. Error      t value    Pr(>|t|)
## 10.66848632   4.55986460   2.33964980  0.01976075
```

### 3.3 AIPW with cross-fitting, using a machine learning method of your choice.

```
data <- star_2
# Available in randomized settings and observational settings with unconfoundedness+overlap

# A list of vectors indicating the left-out subset
n <- nrow(data)
n.folds <- 5
indices <- split(seq(n), sort(seq(n) %% n.folds))

# Preparing data
W <- data$W
Y <- data$gitmathss
covariates <- c("gender", "race", "flagsgk", "gktrace", "g4surban")

# Matrix of (transformed) covariates used to estimate  $E[Y|X, W]$ 
fmla.xw <- formula(paste("~ 0 +", paste0("bs(", covariates, ", df=3)", "*", W, collapse=" + ")))
XW <- model.matrix(fmla.xw, data)
# Matrix of (transformed) covariates used to predict  $E[Y|X, W=w]$  for each  $w$  in  $\{0, 1\}$ 
data.1 <- data
data.1$W <- 1
XW1 <- model.matrix(fmla.xw, data.1) # setting  $W=1$ 
data.0 <- data
data.0$W <- 0
XW0 <- model.matrix(fmla.xw, data.0) # setting  $W=0$ 

# Matrix of (transformed) covariates used to estimate and predict  $e(X) = P[W=1|X]$ 
fmla.x <- formula(paste("~ 0 +", paste0("bs(", covariates, ", df=3)", collapse=" + ")))
XX <- model.matrix(fmla.x, data)

# # (Optional) Not penalizing the main effect (the coefficient on W)
penalty.factor <- rep(1, ncol(XW))
#penalty.factor[colnames(XW) == W] <- 0

# Cross-fitted estimates of  $E[Y|X, W=1]$ ,  $E[Y|X, W=0]$  and  $e(X) = P[W=1|X]$ 
```

```

mu.hat.1 <- rep(NA, n)
mu.hat.0 <- rep(NA, n)
e.hat <- rep(NA, n)
for (idx in indices) {
  # Estimate outcome model and propensity models
  # Note how cross-validation is done (via cv.glmnet) within cross-fitting!
  outcome.model <- cv.glmnet(x=XW[-idx,], y=Y[-idx], family="gaussian", penalty.factor=penalty.factor)
  propensity.model <- cv.glmnet(x=XX[-idx,], y=W[-idx], family="binomial")

  # Predict with cross-fitting
  mu.hat.1[idx] <- predict(outcome.model, newx=XW1[idx,], type="response")
  mu.hat.0[idx] <- predict(outcome.model, newx=XW0[idx,], type="response")
  e.hat[idx] <- predict(propensity.model, newx=XX[idx,], type="response")
}

# Compute the summand in AIPW estimator
aipw.scores <- (mu.hat.1 - mu.hat.0
               + W / e.hat * (Y - mu.hat.1)
               - (1 - W) / (1 - e.hat) * (Y - mu.hat.0))

# Tally up results
ate.aipw.est <- mean(aipw.scores)
ate.aipw.se <- sd(aipw.scores) / sqrt(n)
ate.aipw.tstat <- ate.aipw.est / ate.aipw.se
ate.aipw.pvalue <- 2*(pnorm(1 - abs(ate.aipw.tstat)))
ate.aipw.results <- c(estimate=ate.aipw.est, std.error=ate.aipw.se, t.stat=ate.aipw.tstat, pvalue=ate.aipw.pvalue)
print(ate.aipw.results)

## estimate std.error t.stat pvalue
## 9.9868627 3.8838156 2.5714049 0.1160886

```

## Part 2

```
library('MatchIt')
```

```
## Warning: package 'MatchIt' was built under R version 4.1.1
```

```
library('optmatch')
```

```
## Warning: package 'optmatch' was built under R version 4.1.1
```

```
## Loading required package: survival
```

```
## Warning: package 'survival' was built under R version 4.1.1
```

```
data('lalonge')
head(lalonge)
```

```
##      treat age educ   race married nodegree re74 re75      re78
## NSW1     1  37  11  black        1         1   0   0 9930.0460
## NSW2     1  22   9 hispan        0         1   0   0 3595.8940
## NSW3     1  30  12  black        0         0   0   0 24909.4500
## NSW4     1  27  11  black        0         1   0   0 7506.1460
## NSW5     1  33   8  black        0         1   0   0 289.7899
## NSW6     1  22   9  black        0         1   0   0 4056.4940
```

```
sum(lalonde$treat)
```

```
## [1] 185
```

The number of total treated cases is 185.

```
mean(lalonde[which(lalonde$treat==0),'re78'])
```

```
## [1] 6984.17
```

The outcome variable in the control group is \$6984.17.

```
mean(lalonde[which(lalonde$treat==1),'re78'])
```

```
## [1] 6349.144
```

The outcome variable in the treatment group is \$6349.144.

## Estimating propensity score using logistic regression

```
PS_logit <- glm(treat ~ age + educ+race+married+nodegree+re74+re75, data = lalonde, family=binomial)
PS_Scores = fitted(PS_logit)
head(PS_Scores)
```

```
##      NSW1      NSW2      NSW3      NSW4      NSW5      NSW6
## 0.6387699 0.2246342 0.6782439 0.7763241 0.7016387 0.6990699
```

## Create mutually exclusive strata based on estimated PS

```
lalonde$PS_Scores <- PS_Scores
```

```
# create strata
Quintiles <- quantile(lalonde$PS_Scores, prob=seq(from=0,to=1,by=0.2),na.rm=TRUE)
lalonde$PS_Stratum <- cut(lalonde$PS_Scores, breaks = Quintiles, labels = 1:5, include.lowest = TRUE)
```

**Step 3 & 4: Group treated and control units into each stratum;  
Within each stratum, we calculate the difference in means between  
the treated and control**

```
# Fifth Stratum: PS_Stratum = 5
diff_5 = mean(lalonde[which(lalonde$treat==1&lalonde$PS_Stratum==5), 're78'])- mean(lalonde[which(lalonde$treat==0&lalonde$PS_Stratum==5), 're78'])

# Forth Stratum: PS_Stratum = 4
diff_4 = mean(lalonde[which(lalonde$treat==1&lalonde$PS_Stratum==4), 're78'])- mean(lalonde[which(lalonde$treat==0&lalonde$PS_Stratum==4), 're78'])

# Third Stratum: PS_Stratum = 3
diff_3 = mean(lalonde[which(lalonde$treat==1&lalonde$PS_Stratum==3), 're78'])- mean(lalonde[which(lalonde$treat==0&lalonde$PS_Stratum==3), 're78'])

# Second Stratum: PS_Stratum = 2
diff_2 = mean(lalonde[which(lalonde$treat==1&lalonde$PS_Stratum==2), 're78'])- mean(lalonde[which(lalonde$treat==0&lalonde$PS_Stratum==2), 're78'])

# First Stratum: PS_Stratum = 1
diff_1 = mean(lalonde[which(lalonde$treat==1&lalonde$PS_Stratum==1), 're78'])- mean(lalonde[which(lalonde$treat==0&lalonde$PS_Stratum==1), 're78'])
```

**To estimate the ATE, we weigh by the number of units in each stratum.**

```
total_units = nrow(lalonde)
ATE_1 = diff_1* nrow(lalonde[which(lalonde$PS_Stratum==1),])/total_units
ATE_2 = diff_2* nrow(lalonde[which(lalonde$PS_Stratum==2),])/total_units
ATE_3 = diff_3* nrow(lalonde[which(lalonde$PS_Stratum==3),])/total_units
ATE_4 = diff_4* nrow(lalonde[which(lalonde$PS_Stratum==4),])/total_units
ATE_5 = diff_5* nrow(lalonde[which(lalonde$PS_Stratum==5),])/total_units
ATE = ATE_1+ATE_2+ATE_3+ATE_4+ATE_5
ATE
```

```
## [1] -601.1986
```

```
mean(lalonde[which(lalonde$treat==1&lalonde$PS_Stratum==1), 're78'])
```

```
## [1] 0
```

```
mean(lalonde[which(lalonde$treat==0&lalonde$PS_Stratum==1), 're78'])
```

```
## [1] 10467.06
```



It's interesting to see ATE is negative. It is mostly due to stratum 1 has a oversized impact on the overall ATE; that is, for the stratum that is closest to 0 likelihood of enrolling in training programs, the real income of those who enrolled is significantly less than those who are not enrolled. This makes intuitive sense because those who are not enrolled have much higher income than those who enrolled (10467 vs. 0).

The next reasonable exploration is estimating the effect of ATT.

```
total_treated_units = nrow(lalonde[which(lalonde$treat==1),])
ATT_1 = diff_1* nrow(lalonde[which(lalonde$treat==1 & lalonde$PS_Stratum==1),])/total_treated_units
ATT_2 = diff_2* nrow(lalonde[which(lalonde$treat==1 & lalonde$PS_Stratum==2),])/total_treated_units
ATT_3 = diff_3* nrow(lalonde[which(lalonde$treat==1 & lalonde$PS_Stratum==3),])/total_treated_units
ATT_4 = diff_4* nrow(lalonde[which(lalonde$treat==1 & lalonde$PS_Stratum==4),])/total_treated_units
ATT_5 = diff_5* nrow(lalonde[which(lalonde$treat==1 & lalonde$PS_Stratum==5),])/total_treated_units
ATT = ATT_1+ATT_2+ATT_3+ATT_4+ATT_5
ATT
```

```
## [1] 1333.052
```

The ATT is a 1333.05 increase in household income after attending the job training program. The experimental benchmark is \$1,794, and the original paper estimates 1,608 using the PS Stratification. The gap is due to some technical details such as ML model choice in estimating PS.

In case when strata cannot estimate treatment effects, I will adjust the number of strata just so in each stratum there is meaningfully enough treatment and control observations.

### 3.

```
set.seed(1)
n = 1000
p = 20
X = matrix(rnorm(n * p), n, p)
propensity = pmax(0.2, pmin(0.8, 0.5 + X[,1]/3))
W = rbinom(n, 1, propensity)
Y = pmax(X[,1] + W * X[,2], 0) + rnorm(n)
```

### 4. PS using logsitic regression

```
library(glmnet)
logit <- cv.glmnet(x=X, y=W, family="binomial")
e.hat <- predict(logit, X, s = "lambda.min", type="response")
```

## IWP

```
z <- Y * (W/e.hat - (1-W)/(1-e.hat))
ate.est <- mean(z)
ate.se <- sd(z) / sqrt(length(z))
ate.tstat <- ate.est / ate.se
ate.pvalue <- 2*(pnorm(1 - abs(ate.est/ate.se)))
ate.results <- c(estimate=ate.est, std.error=ate.se, t.stat=ate.tstat, pvalue=ate.pvalue)
print(ate.results)
```

```
## estimate std.error t.stat pvalue
## 0.21348720 0.09242033 2.30995930 0.19020960
```

## Stratified

```
data <- cbind(X, W, Y)
df = as.data.frame(data)
```

```
# create strata
df$e.hat <- e.hat
Quintiles <- quantile(df$e.hat, prob=seq(from=0,to=1,by=0.2),na.rm=TRUE)
df$PS_Stratum <- cut(df$e.hat, breaks = Quintiles, labels = 1:5, include.lowest = TRUE)
```

```
# Fifth Stratum: PS_Stratum = 5
diff_5 = mean(df[which(df$W==1&df$PS_Stratum==5), 'Y']) - mean(df[which(df$W==0&df$PS_Stratum==5), 'Y'])

# Forth Stratum: PS_Stratum = 4
diff_4 = mean(df[which(df$W==1&df$PS_Stratum==4), 'Y']) - mean(df[which(df$W==0&df$PS_Stratum==4), 'Y'])
# Third Stratum: PS_Stratum = 3
diff_3 = mean(df[which(df$W==1&df$PS_Stratum==3), 'Y']) - mean(df[which(df$W==0&df$PS_Stratum==3), 'Y'])

# Second Stratum: PS_Stratum = 2
diff_2 = mean(df[which(df$W==1&df$PS_Stratum==2), 'Y']) - mean(df[which(df$W==0&df$PS_Stratum==2), 'Y'])

# First Stratum: PS_Stratum = 1
diff_1 = mean(df[which(df$W==1&df$PS_Stratum==1), 'Y']) - mean(df[which(df$W==0&df$PS_Stratum==1), 'Y'])
```

To estimate the ATE, we weigh by the number of units in each stratum.

```
total_units = 1000
ATE_1 = diff_1* nrow(df[which(df$PS_Stratum==1),])/total_units
ATE_2 = diff_2* nrow(df[which(df$PS_Stratum==2),])/total_units
ATE_3 = diff_3* nrow(df[which(df$PS_Stratum==3),])/total_units
ATE_4 = diff_4* nrow(df[which(df$PS_Stratum==4),])/total_units
ATE_5 = diff_5* nrow(df[which(df$PS_Stratum==5),])/total_units
ATE = ATE_1+ATE_2+ATE_3+ATE_4+ATE_5
ATE
```

```
## [1] 0.1714962
```

```
groundtruth_ATE = mean(df[which(df$W==1), 'Y']) - mean(df[which(df$W==0), 'Y'])
groundtruth_ATE
```

```
## [1] 0.5512573
```

I just picked 5 because research shows 5 strata can remove approximately 90% of the bias in the unadjusted estimate (Rosenbaum and Rubin, 1984). And the more strata doesn't always mean the better.

Repeat 20 times, calculate MSE, compare with groundtruth ATE

```
SE_IWP = rep(0,20)
SE_stratified = rep(0,20)
for (i in 1:20) {
  #simulate data
  set.seed(i+1)
  n = 1000
  p = 20
  X = matrix(rnorm(n * p), n, p)
  propensity = pmax(0.2, pmin(0.8, 0.5 + X[,1]/3))
  W = rbinom(n, 1, propensity)
  Y = pmax(X[,1] + W * X[,2], 0) + rnorm(n)

  #e.hat with logistic regression
  logit <- cv.glmnet(x=X, y=W, family="binomial")
  e.hat <- predict(logit, X, s = "lambda.min", type="response")

  #IWP
  z <- Y * (W/e.hat - (1-W)/(1-e.hat))
  ate.est <- mean(z)

  #stratified
  ata <- cbind(X, W, Y)
  df = as.data.frame(ata)
  df$e.hat <- e.hat
  Quintiles <- quantile(df$e.hat, prob=seq(from=0,to=1,by=0.2),na.rm=TRUE)
  df$PS_Stratum <- cut(df$e.hat, breaks = Quintiles, labels = 1:5, include.lowest = TRUE)

  diff_5 = mean(df[which(df$W==1&df$PS_Stratum==5), 'Y']) - mean(df[which(df$W==0&df$PS_Stratum==5), 'Y'])
  diff_4 = mean(df[which(df$W==1&df$PS_Stratum==4), 'Y']) - mean(df[which(df$W==0&df$PS_Stratum==4), 'Y'])
  diff_3 = mean(df[which(df$W==1&df$PS_Stratum==3), 'Y']) - mean(df[which(df$W==0&df$PS_Stratum==3), 'Y'])
  diff_2 = mean(df[which(df$W==1&df$PS_Stratum==2), 'Y']) - mean(df[which(df$W==0&df$PS_Stratum==2), 'Y'])
  diff_1 = mean(df[which(df$W==1&df$PS_Stratum==1), 'Y']) - mean(df[which(df$W==0&df$PS_Stratum==1), 'Y'])

  total_units = 1000
  ATE_1 = diff_1* nrow(df[which(df$PS_Stratum==1),])/total_units
  ATE_2 = diff_2* nrow(df[which(df$PS_Stratum==2),])/total_units
  ATE_3 = diff_3* nrow(df[which(df$PS_Stratum==3),])/total_units
  ATE_4 = diff_4* nrow(df[which(df$PS_Stratum==4),])/total_units
}
```

```

ATE_5 = diff_5* nrow(df[which(df$PS_Stratum==5),])/total_units
ATE = ATE_1+ATE_2+ATE_3+ATE_4+ATE_5

#groundtruth
groundtruth_ATE = mean(df[which(df$W==1), 'Y']) - mean(df[which(df$W==0), 'Y'])
SE_IWP[i] = (ate.est-groundtruth_ATE)^2
SE_stratified[i] = (ATE-groundtruth_ATE)^2
}

print(mean(SE_IWP))

```

```
## [1] 0.1359569
```

```
print(mean(SE_stratified))
```

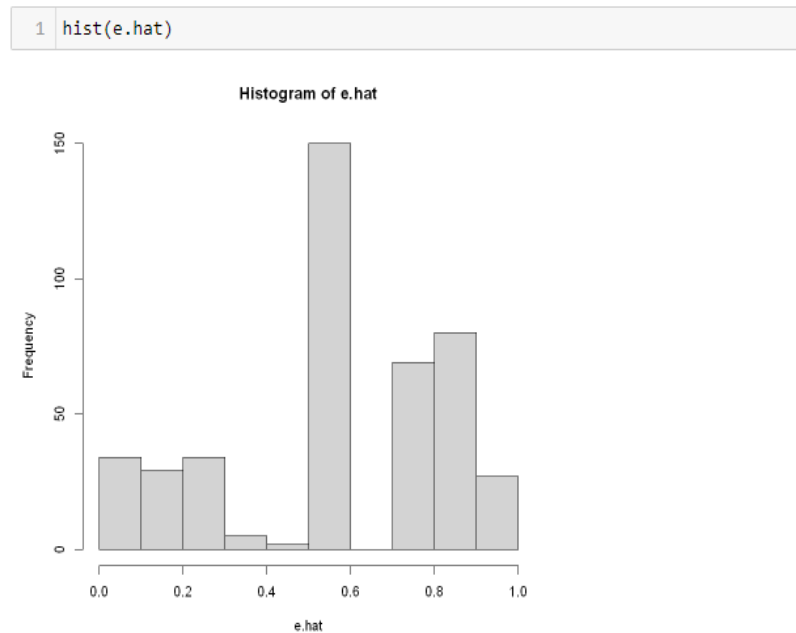
```
## [1] 0.002375719
```

IWP yields a MSE of 0.136 and the stratified yields 0.00238, which is 57 times larger.

The stratified method recovers more accurate ATE estimation in terms of MSE. This is because the precise matching process increases imbalance, inefficiency, model dependence, bias and fails to reduce the imbalance (King and Nielsen, 2019). In contrast, PS Stratification offers a better alternative to PS Matching by averaging across a set of quasi-RCTs within each stratum and adjusting for the covariate imbalance by tweaking the score.

## Part 1

1. Sample restriction: We decided to use the Star Project and restrict the sample to students in small and regular class, so we dropped the Regular + AIDE class group. Our treatment variable is if the student belongs to the Small Class and our outcome variable is the math score. The ATE we found is 13.39 and the 95% CI is "13.39 +/- 3".
2. **Describe your method(s) for systematically deleting some observations as a function of X's and treatment status**
  - a. We decided to delete observations in the next two groups. First, students that are treated, black and live in the rural area. Second, we dropped students in the control group that are white and live in urban areas.
  - b. Histogram of the Propensity Score



3. **In the modified dataset, report point estimates and confidence intervals for the ATE using the following 3 methods.**
  - a. The difference in means estimator: ATE = 16.2, 95% CI = "16.2 +/- 7"
  - b. Simple linear regression: ATE = 17.31, 95% CI = "17.31 +/- 8"
  - c. AIPW with cross-fitting = ATE = 16.2, 95% CI = "16.2 +/- 7.4"

## Part 2

1. To estimate the ATE, we weigh by the number of units in each stratum and aggregate the effect across all strata. Conceptually, PS stratification is considered a meta-analysis of a set of quasi-RCTs within each stratum (Austin, 2011). We can think of the treatment assignment process as a random assignment conditional on observed covariates, aka the unconfoundedness assumption, meaning the potential outcomes are independent of the treatment status (Austin, 2011). Therefore, propensity stratification provides consistent estimates of the ATE, when we have unconfoundedness assumption and a reasonable number of strata.

5.

IWP yields a MSE of 0.136 and the stratified yields 0.00238, which is 57 times larger.

The stratified method recovers more accurate ATE estimation in terms of MSE. This is because the precise matching process increases imbalance, inefficiency, model dependence, bias and fails to reduce the imbalance (King and Nielsen, 2019). In contrast, PS Stratification offers a better alternative to PS Matching by averaging across a set of quasi-RCTs within each stratum and adjusting for the covariate imbalance by tweaking the score.