Causal Inference HW 3

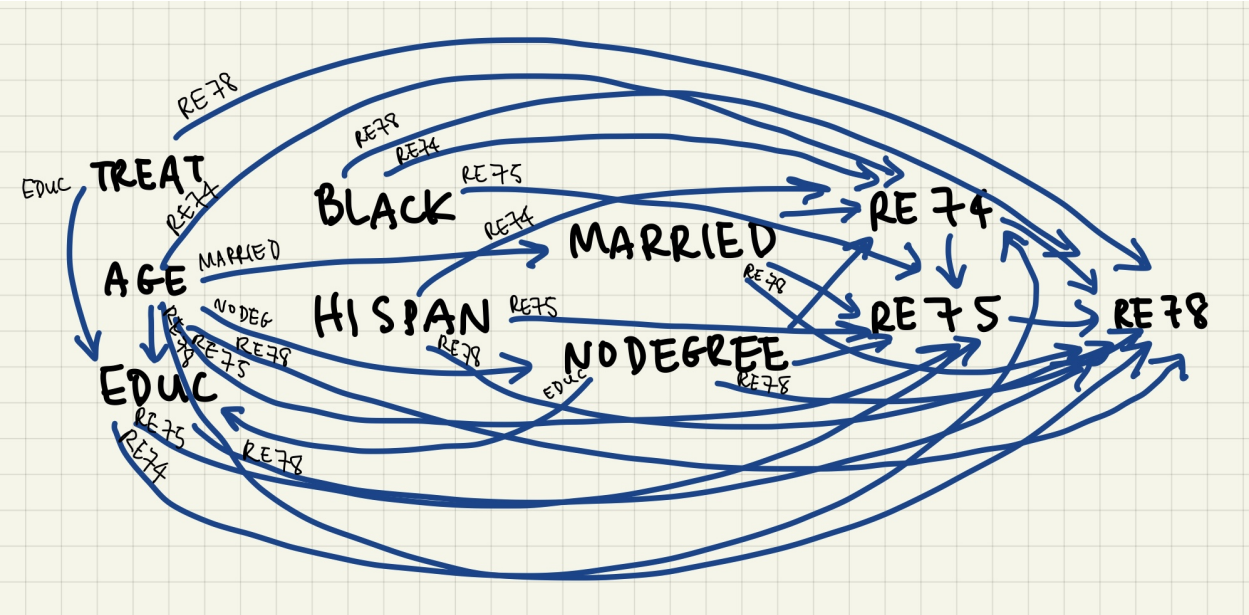
Adeline Shin

11/15/2020

Part I

1: DAGs

```
knitr::include_graphics("./IMG_0503.jpg")
```



In the DAG, there are the following arrows:

- treatment has arrows going to re78 and educ.
- age has arrows going to educ, re74, re75, re78, married, nodeg, educ
- educ has arrows going to re74, re75, re78
- black has arrows going to re74, re75, re78
- hispan has arrows going to re74, re75, re78
- married has arrows going to re74, re75, re78
- nodegree has arrows going to educ, re74, re75, re78
- re74 has arrows going to re75 and re78
- re75 has an arrow going to re78

2: Overall Covariate Balance

Loading the Data

```
hw3_data = read.csv("./hw3_data.csv")
hw3_data = hw3_data %>%
  mutate(
```

```
    treat = factor(treat),
    black = factor(black),
    hispan = factor(hispan),
    married = factor(married),
    nodegree = factor(nodegree)
  )
```

```
# Looking at overall covariate balance
names(hw3_data)
```

```
##  [1] "X"        "treat"    "age"      "educ"     "black"    "hispan"
##  [7] "married"  "nodegree" "re74"     "re75"     "re78"
```

```
X = hw3_data[, !(colnames(hw3_data) %in% c("X"))]
head(X)
```

```
##   treat age educ black hispan married nodegree re74 re75       re78
## 1     1  37   11     1      0       1        1    0    0  9930.0460
## 2     1  22    9     0      1       0        1    0    0  3595.8940
## 3     1  30   12     1      0       0        0    0    0 24909.4500
## 4     1  27   11     1      0       0        1    0    0  7506.1460
## 5     1  33    8     1      0       0        1    0    0   289.7899
## 6     1  22    9     1      0       0        1    0    0  4056.4940
```

```
vars = c("age", "educ", "black", "hispan", "married", "nodegree", "re74", "re75")

tabpresub = CreateTableOne(vars = vars, strata = "treat", data = hw3_data, test = FALSE)
print(tabpresub, smd = TRUE)
```

```
##                    Stratified by treat
##                     0                 1                 SMD
##   n                     429               185
##   age (mean (SD))     28.03 (10.79)     25.82 (7.16)     0.242
##   educ (mean (SD))    10.24 (2.86)      10.35 (2.01)     0.045
##   black = 1 (%)          87 (20.3)        156 (84.3)     1.671
##   hispan = 1 (%)         61 (14.2)         11 ( 5.9)     0.277
##   married = 1 (%)       220 (51.3)         35 (18.9)     0.721
##   nodegree = 1 (%)      256 (59.7)        131 (70.8)     0.235
##   re74 (mean (SD)) 5619.24 (6788.75) 2095.57 (4886.62)  0.596
##   re75 (mean (SD)) 2466.48 (3292.00) 1532.06 (3219.25)  0.287
```

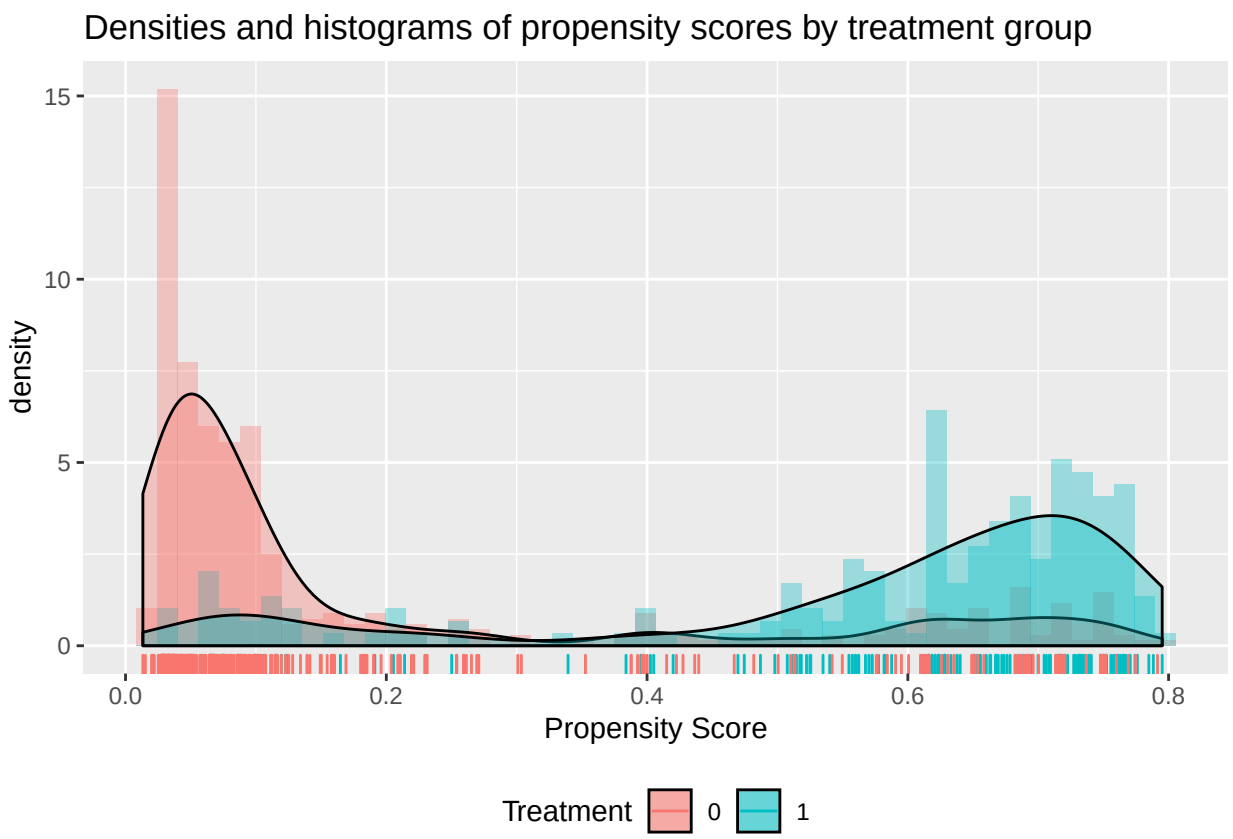
3: Propensity Scores

```
ps.model = glm(treat ~ age + educ + black + hispan + married + nodegree, data = hw3_data, family = bino

ps = predict(ps.model, type = "response")
```

4: Evaluating Propensity Scores and Trimming

```
# Check Overlap
prop.func = function(x, trt)
{
  # fit propensity score model
```

```
  propens.model <- glm(treat ~ age + educ + black + hispan + married + nodegree, data = hw3_data, family
  pi.x = predict(propens.model, type = "response")
  pi.x
}

check.overlap(x = hw3_data,
              trt = hw3_data$treat,
              type = "both",
              propensity.func = prop.func)
```



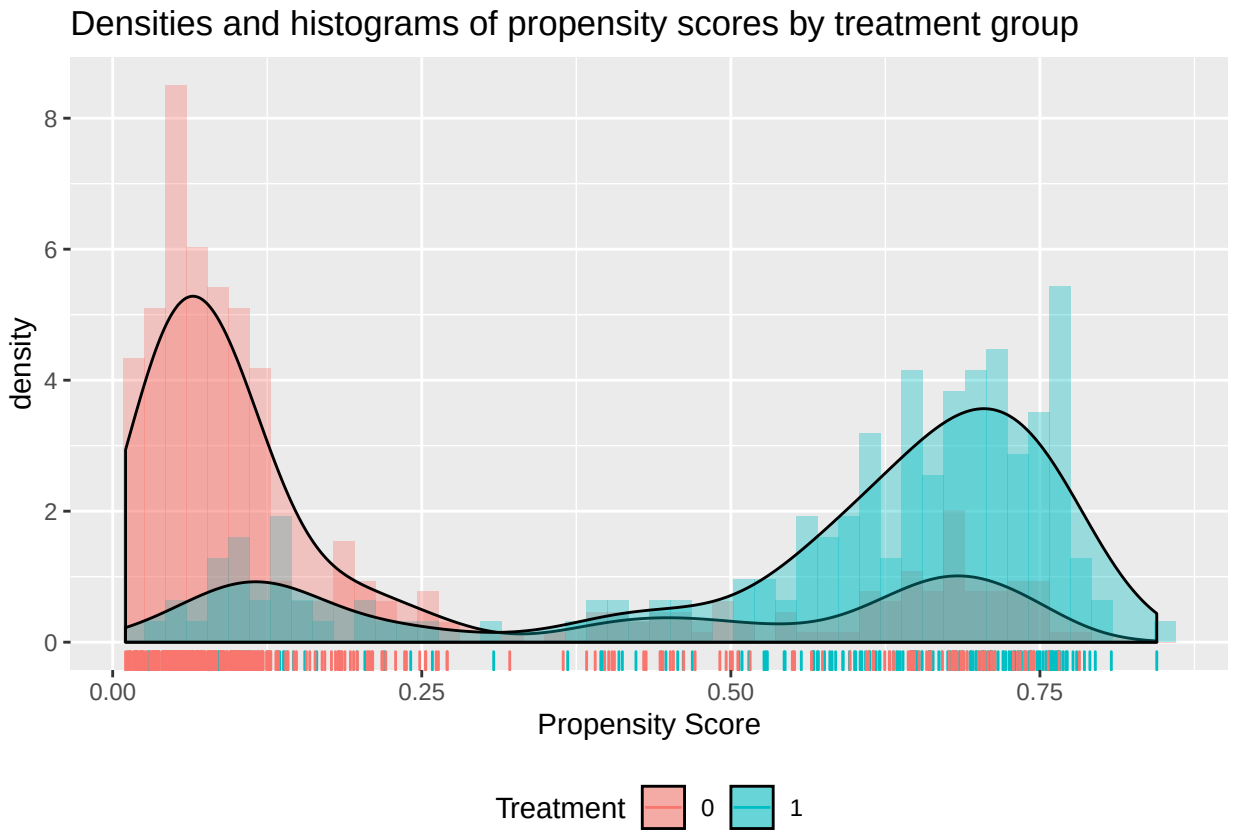

```
# Trim non-overlap data
# eliminate controls for whom the P(A=1|C) is less that the min(P(A=1|C)) found in the treated group
min(ps[hw3_data$treat==1])
ps[which(hw3_data$treat==0)]<= min(ps[hw3_data$treat==1])

# eliminate treated for whom the P(A=1|C) is greater that the max(P(A=1|C)) found in the control group
max(ps[hw3_data$treat==0])
ps[which(hw3_data$treat==1)]>= max(ps[hw3_data$treat==0])

data = hw3_data[ps>=min(ps[hw3_data$treat==1]) & ps <= max(ps[hw3_data$treat==0]),]
dim(hw3_data)
dim(data)
```

```
# Check overlap again for trimmed data
ps.model = glm(treat ~ age + educ + black + hispan + married + nodegree, data = data, family = binomial
summary(ps.model)
```

```
##
```

```
## Call:
## glm(formula = treat ~ age + educ + black + hispan + married +
##     nodegree, family = binomial, data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7367  -0.4878  -0.3397   0.8182   2.5783
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.337426   1.039229  -4.174    3e-05 ***
## age          0.007736   0.013351   0.579  0.56227
## educ         0.136880   0.066153   2.069  0.03853 *
## black1       3.036204   0.285473  10.636  < 2e-16 ***
## hispan1      0.909107   0.421742   2.156  0.03112 *
## married1    -0.847718   0.274155  -3.092  0.00199 **
## nodegree1    0.701467   0.337627   2.078  0.03774 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 712.31  on 563  degrees of freedom
## Residual deviance: 491.27  on 557  degrees of freedom
## AIC: 505.27
##
## Number of Fisher Scoring iterations: 5
```

```r
ps = predict(ps.model, type="response") #gets the propensity scores for each unit, based on the model

data$ps = ps

prop.func2 = function(x, trt)
{
  # fit propensity score model
  propens.model <- glm(treat ~ age + educ + black + hispan + married + nodegree + re74 + re75, data = da
  pi.x <- predict(propens.model, type = "response")
  pi.x
}

check.overlap(x = data,
              trt = data$treat,
              type = "both",
              propensity.func = prop.func2)
```



Trimming the data resulted in 50 subjects trimmed, but this results in greater efficiency, since it increases the proportion of subjects able to be matched based on propensity score. However, generalizability is hurt because we have a fewer number of participants, and we are only using a selected number of participants.

5: Covariate Balance in Trimmed Sample

```
tabpostsub = CreateTableOne(vars = vars, strata = "treat", data = data, test = TRUE)
print(tabpostsub, smd = TRUE)
```

```
##                   Stratified by treat
##                    0                 1                 p      test SMD
##   n                    380               184
##   age (mean (SD))    28.02 (11.17)     25.72 (7.05)     0.011       0.247
##   educ (mean (SD))   10.29 (2.77)      10.34 (2.02)     0.808       0.023
##   black = 1 (%)         87 (22.9)        155 (84.2)    <0.001       1.560
##   hispan = 1 (%)        61 (16.1)         11 ( 6.0)     0.001       0.326
##   married = 1 (%)      171 (45.0)         35 (19.0)    <0.001       0.580
##   nodegree = 1 (%)     241 (63.4)        130 (70.7)     0.109       0.154
##   re74 (mean (SD)) 5135.50 (6694.04) 2106.96 (4897.49) <0.001       0.516
##   re75 (mean (SD)) 2190.60 (3019.80) 1540.38 (3226.04)  0.019       0.208
```

The SMD of the variables educ, black, married, nodegree, re74, and re75 went down after trimming variables, while the rest of them went up. Some of the increases are not of concern, since the SMD is still below 0.2, but the race variables, married, re74, and re75 are still unbalanced since their SMD values are greater than 0.2.

6: Subclassification to Balance Covariates

The first attempt at subclassification using 5 subclasses:

```
#creating subclasses
subclass.breaks = quantile(ps, c(.20, .40, .60, .80)) # bins (initial try - modify as needed)
subclass = data$ps
subclass = as.numeric(data$ps>subclass.breaks[1])
subclass[which(data$ps>subclass.breaks[1]& data$ps<=subclass.breaks[2])] = 1
subclass[which(data$ps>subclass.breaks[2]&data$ps<=subclass.breaks[3])] = 2
subclass[which(data$ps>subclass.breaks[3]&data$ps<=subclass.breaks[4])] = 3
subclass[which(data$ps>subclass.breaks[4])] = 4
#looking at sample sizes within each subclass
table(data$treat, subclass) #violates overlap
```

```
##    subclass
##       0   1   2   3   4
##   0 110 103  91  40  36
##   1   3  10  21  73  77
```

The subclasses are not balanced in terms of their treatment groups, and the overlap is violated, so we need a different number of subclasses. Next, 4 subclasses will be used as an attempt for subclassification.

```
# Too much overlap: try different subclass cutoffs
subclass.breaks = quantile(data$ps, c(.25, .50, .75))
subclass = data$ps
subclass = as.numeric(data$ps>subclass.breaks[1])
subclass[which(data$ps>subclass.breaks[1]& data$ps<=subclass.breaks[2])]<- 1
subclass[which(data$ps>subclass.breaks[2]&data$ps<=subclass.breaks[3])]<- 2
subclass[which(data$ps>subclass.breaks[3])]<- 3
table(data$treat, subclass)
```

```
##    subclass
##       0   1   2   3
##   0 140 121  76  43
##   1   3  18  65  98
```

This is a little more balanced in terms of the number in each group, but there are only 3 subjects with treatment 1 in subgroup 0, so we will attempt subclassification with 3 subclasses.

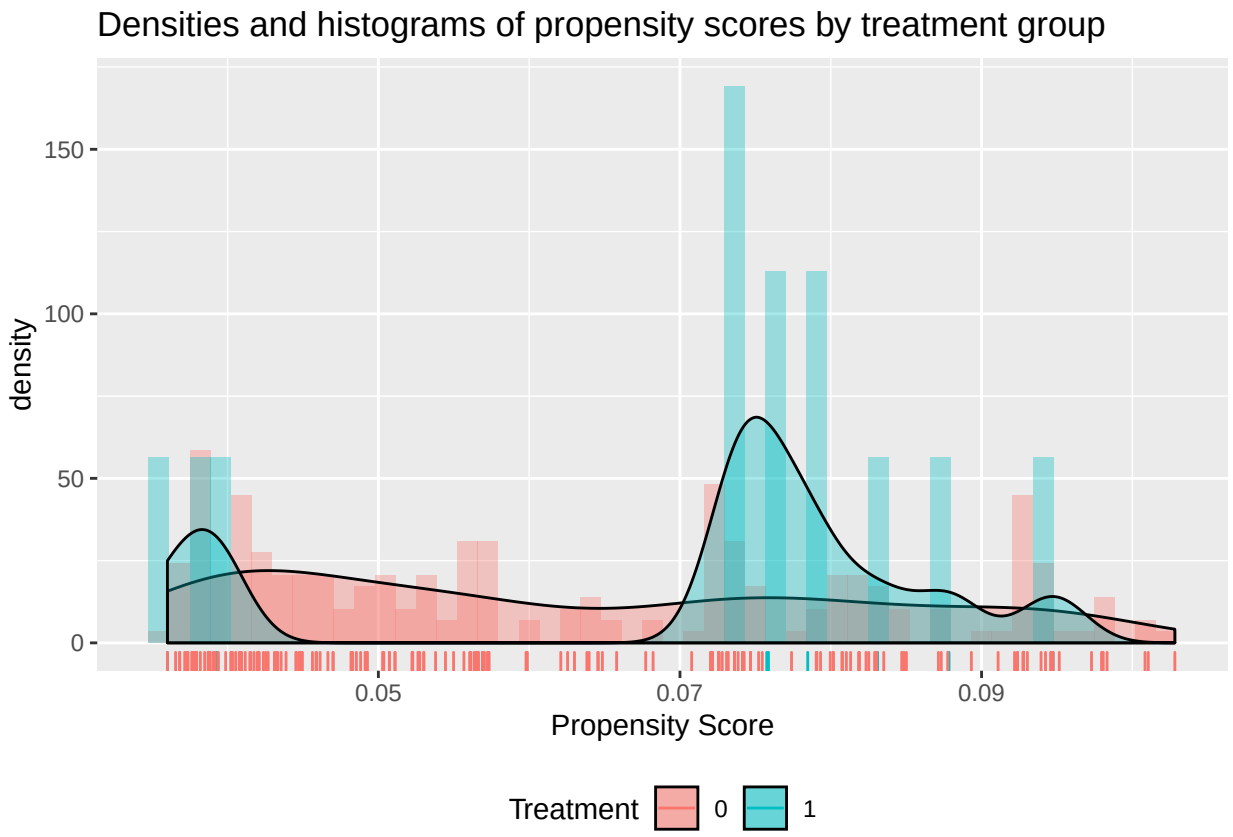
```
subclass.breaks = quantile(data$ps, c(.4, .66))
subclass = data$ps
subclass = as.numeric(data$ps>subclass.breaks[1])
subclass[which(data$ps>subclass.breaks[1]& data$ps<=subclass.breaks[2])]<- 1
subclass[which(data$ps>subclass.breaks[2])]<- 2
table(data$treat, subclass)
```

```
##    subclass
##       0   1   2
##   0 213 103  64
##   1  13  43 128
```

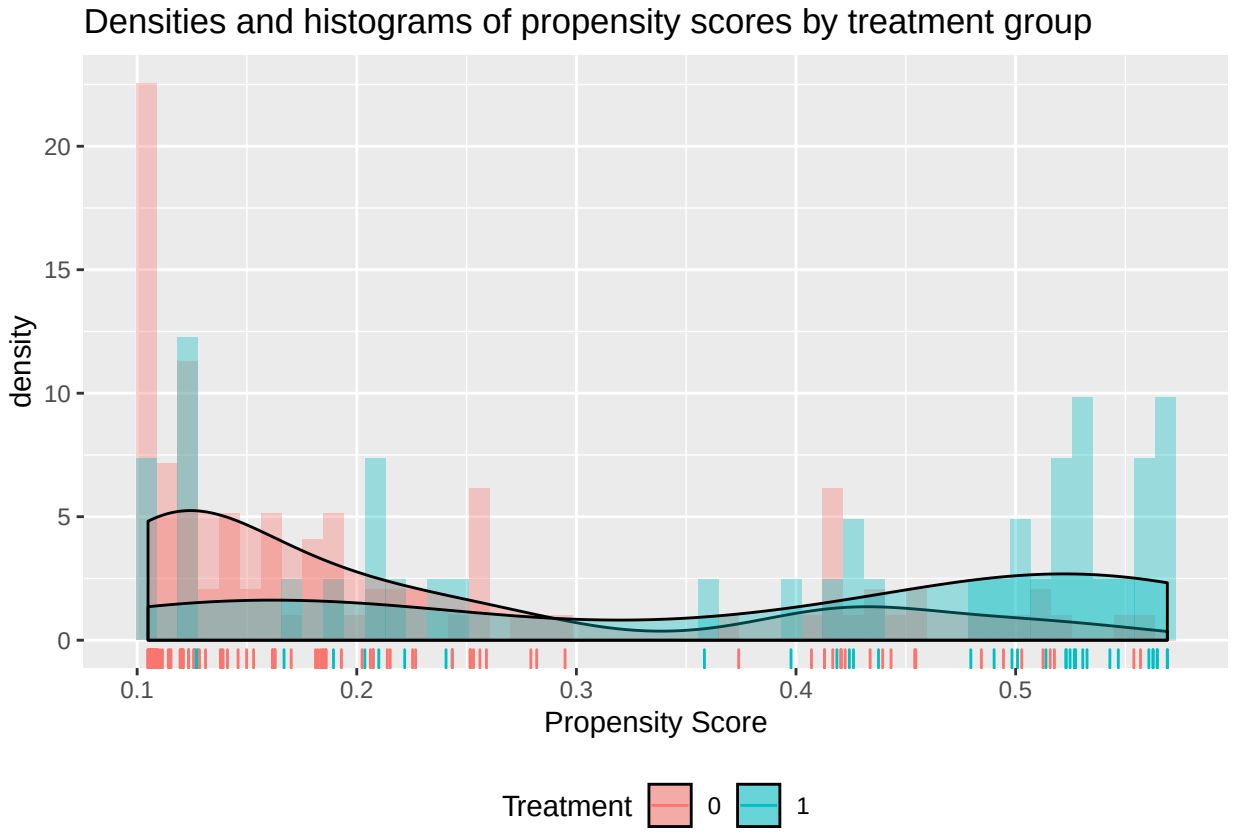
Only using 3 subgroups with the cutoffs of 0.4 and 0.66 gives a slightly more balanced set of subjects than any of the other previous subclassifications. The density graphs and histograms will be examined, as well as overall covariate balance.

```
#looking at propensity scores within subclasses
prop.func <- function(x, trt)
```

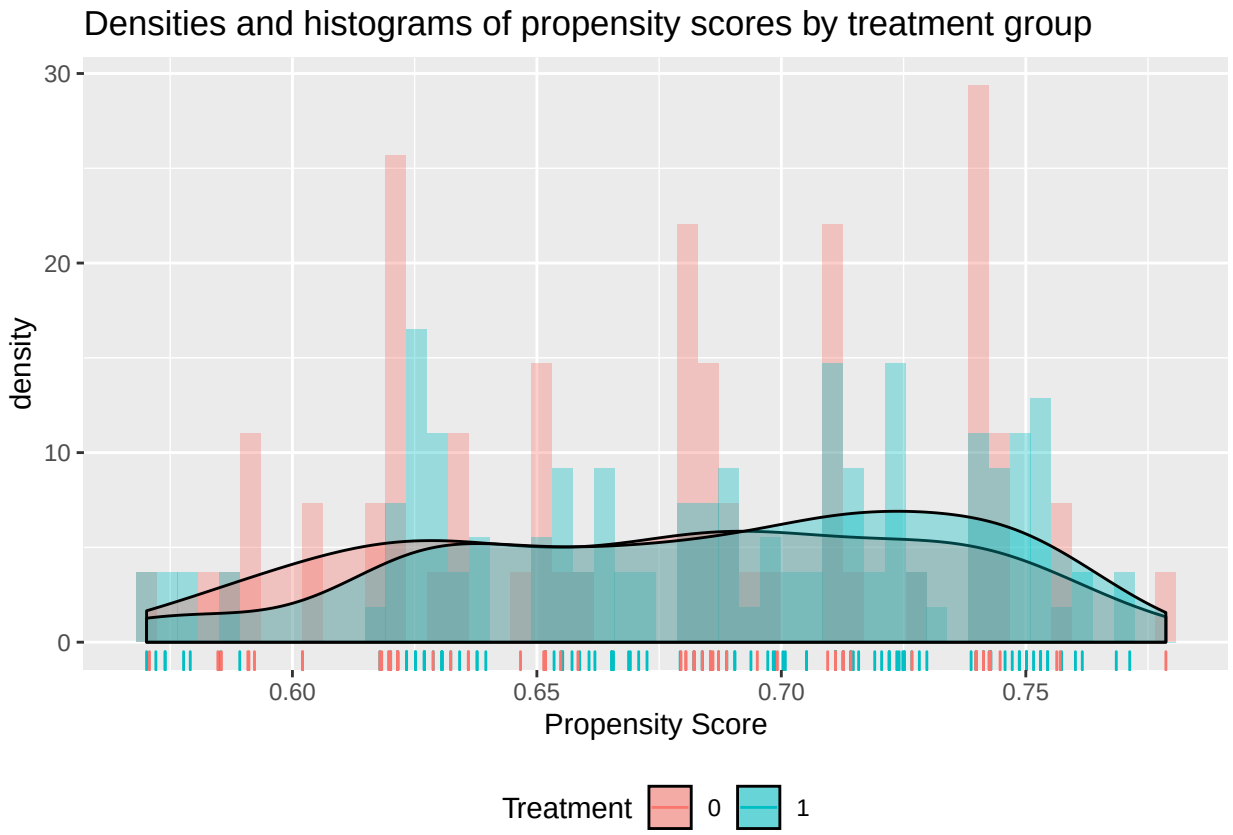
```
{
  data$ps[which(data$ps <= subclass.breaks[1])]
}
#data$ps <-ps
check.overlap(x = data[which(data$ps <=subclass.breaks[1]),],
              trt = data$treat[which(data$ps <= subclass.breaks[1])],
              type = "both",
              propensity.func = prop.func)
```




```
prop.func <- function(x, trt)
{
  data$ps[which(data$ps>subclass.breaks[1]&data$ps<=subclass.breaks[2])]
}
#data$ps <-ps
check.overlap(x = data[which(data$ps>subclass.breaks[1]&data$ps<=subclass.breaks[2]),],
              trt = data$treat[which(data$ps>subclass.breaks[1]&data$ps<=subclass.breaks[2])],
              type = "both",
              propensity.func = prop.func)
```




```
prop.func <- function(x, trt)
 {
   data$ps[which(data$ps>subclass.breaks[2])]
 }
check.overlap(x = data[which(data$ps>subclass.breaks[2]),],
              trt = data$treat[which(data$ps>subclass.breaks[2])],
              type = "both",
              propensity.func = prop.func)
```




```
# Check Covariate Balance
names(data)
```

```
##  [1] "X"        "treat"    "age"      "educ"     "black"    "hispan"
##  [7] "married"  "nodegree" "re74"     "re75"     "re78"     "ps"
```

```
tab_s0 <- CreateTableOne(vars = vars, strata = "treat", data = data[which(subclass==0),], test = FALSE)
tab_s1 <- CreateTableOne(vars = vars, strata = "treat", data = data[which(subclass==1),], test = FALSE)
tab_s2 <- CreateTableOne(vars = vars, strata = "treat", data = data[which(subclass==2),], test = FALSE)

## Show table with SMD
print(tab_s0, smd = TRUE)
```

```
##                     Stratified by treat
##                      0                 1                 SMD
##   n                      213               13
##   age (mean (SD))     30.57 (11.42)     24.85 (5.89)     0.630
##   educ (mean (SD))    10.19 (2.91)      10.85 (2.03)     0.260
##   black = 1 (%)           0 ( 0.0)          0 ( 0.0)    <0.001
##   hispan = 1 (%)         23 (10.8)          2 (15.4)     0.136
##   married = 1 (%)       140 (65.7)          5 (38.5)     0.567
##   nodegree = 1 (%)      122 (57.3)          4 (30.8)     0.554
##   re74 (mean (SD)) 6673.01 (7347.84) 1561.90 (3105.31)  0.906
##   re75 (mean (SD)) 2527.59 (3082.17) 1617.18 (2855.90)  0.306
```

```
print(tab_s1, smd = TRUE)
```

```
##                     Stratified by treat
##                      0                 1                 SMD
```

```
##   n                     103               43
##   age (mean (SD))    24.91 (9.44)      28.02 (7.48)     0.365
##   educ (mean (SD))   10.43 (2.67)       9.65 (2.56)     0.296
##   black = 1 (%)         23 (22.3)         27 (62.8)     0.897
##   hispan = 1 (%)        38 (36.9)          9 (20.9)     0.358
##   married = 1 (%)       27 (26.2)         24 (55.8)     0.631
##   nodegree = 1 (%)      75 (72.8)         35 (81.4)     0.205
##   re74 (mean (SD)) 3801.02 (5275.56) 3499.30 (7017.59)  0.049
##   re75 (mean (SD)) 2205.35 (3260.67) 2667.46 (4255.48)  0.122
```

```
print(tab_s2, smd = TRUE)
```

```
##                   Stratified by treat
##                    0                 1                 SMD
##   n                     64               128
##   age (mean (SD))    24.55 (10.83)     25.03 (6.89)     0.053
##   educ (mean (SD))   10.38 (2.43)      10.52 (1.75)     0.070
##   black = 1 (%)         64 (100.0)       128 (100.0)   <0.001
##   hispan = 1 (%)         0 (  0.0)         0 (  0.0)   <0.001
##   married = 1 (%)        4 (  6.2)         6 (  4.7)    0.069
##   nodegree = 1 (%)      44 ( 68.8)        91 ( 71.1)    0.051
##   re74 (mean (SD)) 2166.18 (4792.50) 1694.58 (4062.77)  0.106
##   re75 (mean (SD)) 1045.30 (1990.49) 1153.95 (2766.88)  0.045
```

Covariate Balance for Each Subclass:

- For subclass 0, the variables black and hispan are balanced, but the remainder have an SMD higher than 0.2, which is not ideal. This subclass was the one with the least balance in terms of treatment, though.
- For subclass 1, the variables nodegree, re74, and re75 have an SMD indicating covariate balance, with age, educ, and hispan not too far off. The variables black and married have particularly high SMDs, but the SMD for black is still much lower than the overall SMD.
- In subclass 2, all of the SMDs are below 0.2, indicating balance among all of the variables.

7: Marginal ACE of Participation on Wages

```
ACE0 <- mean(data$re78[which(subclass==0 & data$treat==1)])-mean(data$re78[which(subclass==0 & data$trea
ACE1 <- mean(data$re78[which(subclass==1 & data$treat==1)])-mean(data$re78[which(subclass==1 & data$trea
ACE2 <- mean(data$re78[which(subclass==2 & data$treat==1)])-mean(data$re78[which(subclass==2 & data$trea

ace <- (nrow(data[which(subclass==0),])/nrow(data))*ACE0 + (nrow(data[which(subclass==1),])/nrow(data))

v01 <- var(data$re78[which(subclass==0 & data$treat==1)])
v00 <- var(data$re78[which(subclass==0 & data$treat==0)])

v11 <- var(data$re78[which(subclass==1 & data$treat==1)])
v10 <- var(data$re78[which(subclass==1 & data$treat==0)])

v21 <- var(data$re78[which(subclass==2 & data$treat==1)])
v20 <- var(data$re78[which(subclass==2 & data$treat==0)])

n0 <- nrow(data[which(subclass==0),])
n1 <- nrow(data[which(subclass==1),])
n2 <- nrow(data[which(subclass==2),])
```

```
n01 <- nrow(data[which(subclass==0& data$treat==1),])
n11 <- nrow(data[which(subclass==1& data$treat==1),])
n21 <- nrow(data[which(subclass==2& data$treat==1),])
n00 <- nrow(data[which(subclass==0& data$treat==0),])
n10 <- nrow(data[which(subclass==1& data$treat==0),])
n20 <- nrow(data[which(subclass==2& data$treat==0),])

varace = (n1)^2/nrow(data)^2*((v11/n11)+(v10/n10)) + (n2)^2/nrow(data)^2*((v21/n21)+(v20/n20)) + (n0)^2/

sdace<-sqrt(varace)

CIL=ace-sdace*2
CIU=ace+sdace*2

# p-value calculation
y_obs = data$re78
a = data$treat
group_means_df = aggregate(y_obs, list(a), mean)
group_counts = aggregate(y_obs, list(a), length)
t_obs = group_means_df[1, 2] - group_means_df[2, 2]

a = c(rep(1, 184), rep(0, 380)) # numbers taken from group_counts
a_bold = genperms(a)
```

```
## Too many permutations to use exact method.
## Defaulting to approximate method.
## Increase maxiter to at least 1.70290195518569e+153 to perform exact estimation.
```

```
rdist = rep(NA, times = ncol(a_bold))
for (i in 1:ncol(a_bold)) {
  a_tilde = a_bold[, i]
  rdist[i] = mean(y_obs[a_tilde == 1]) - mean(y_obs[a_tilde == 0])
}

pval = mean(rdist >= abs(t_obs))
pval
```

```
## [1] 0.2759
```

The average causal effect of participating in a job training on wages is 410.2817085, which is positive, so we can say that solely based on the ACE, job training has a positive effect on wage. The confidence interval for the average causal effect is (-1140.4114691, 1960.9748861), which is a wide range, so we cannot be completely confident that the average causal effect is positive. The p-value of the average causal effect is 0.2759, which is greater than 0.05, so at a 0.05 significance level, we fail to reject the null hypothesis and say that job training does not have a positive effect on wage.

8: Marginal ACE of Training and Salary Accounting for Confounders

```
hw3_data$re78_bin = as.numeric(hw3_data$re78 > hw3_data$re75)
reg_adj = glm(re78_bin ~ treat + age + educ + black + hispan + married, data = hw3_data, family = binom
summary(reg_adj)
```

```
##
```

```
## Call:
## glm(formula = re78_bin ~ treat + age + educ + black + hispan +
##     married, family = binomial, data = hw3_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8657  -1.3532   0.7859   0.9071   1.4247
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.781618   0.488844   1.599   0.1098
## treat1       0.254973   0.239455   1.065   0.2870
## age         -0.024898   0.009343  -2.665   0.0077 **
## educ         0.071599   0.033818   2.117   0.0342 *
## black1      -0.476507   0.237024  -2.010   0.0444 *
## hispan1     -0.242958   0.288849  -0.841   0.4003
## married1     0.016932   0.201950   0.084   0.9332
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 783.47  on 613  degrees of freedom
## Residual deviance: 764.60  on 607  degrees of freedom
## AIC: 778.6
##
## Number of Fisher Scoring iterations: 4
```

9: Advantages and Disadvantages

The regression-based approach to confounding adjustment and the subclassification can be very powerful when attempting to compare two unbalanced groups, because if we balance the groups on just the propensity score, then all covariates in the model are also balanced. This allows for direct comparison of two groups that otherwise would not have been able to be compared. However, we must make sure that the data is balanced within particular subclasses before proceeding with analysis, and this can sometimes be tough to do depending on the data.

Part II

a: Non-parametric structural equations

1.

- $\mathrm{Y} = f_Y(\mathrm{L}, \mathrm{A}, \epsilon_Y)$
- $\mathrm{A} = f_A(\mathrm{L}, \epsilon_A)$
- $\mathrm{L} = f_L(\epsilon_L)$

2.

- $\mathrm{Y} = f_Y(\mathrm{U}, \mathrm{A}, \epsilon_Y)$
- $\mathrm{A} = f_A(\mathrm{L}, \epsilon_A)$

- L = f_L(U, ϵ_L)
- U = $f_U(\epsilon_U)$

3.

- Y = f_Y(U, ϵ_Y)
- A = $f_A(\epsilon_A)$
- L = f_L(A, U, ϵ_L)
- U = $f_U(\epsilon_U)$

4.

- Y = f_Y(L, A, ϵ_Y)
- A = f_A(U, ϵ_A)
- L = f_L(U, ϵ_L)
- U = $f_U(\epsilon_U)$

5.

- Y = f_Y(U_1, A, ϵ_Y)
- A = f_A(U_2, ϵ_A)
- L = f_L(U_1, U_2, ϵ_L)
- U_1 = $f_U(\epsilon_{U_1})$
- U_2 = $f_U(\epsilon_{U_2})$

b: Does conditioning on L properly adjust for confounding if we used the definition of confounder based on the backdoor criterion? Justify your answer.

1. No, because A has an arrow going to Y so that does not make A conditionally independent of Y given L.
2. No, because U has arrows going to both L and Y, and A still has an arrow going to Y, so neither variable is conditionally independent of Y given L.
3. Yes, because L is no longer a collider in the path from A to U to Y.
4. ??? No, because conditioning on L does not account for A as a confounder on the path from U to A to Y.
5. ??? No, because conditioning on L does not account for A as a confounder on the path from U2 to A to Y.
6.