Homework 4 Solutions

Question 1

(a)

Overlap is the condition that $0 < \mathrm{P}(D = 1 \mid X = x) < 1$ for all values of x. It means that there are no values of X for which all units are either treated or untreated. If that were the case, we would not be able to compare treated and untreated units with the same value of X, and, thus, we would not be able to identify ATE or ATT.

(b)

$$\begin{aligned}
\text{ATE} &= \mathbb{E}[Y(1) - Y(0)] \\
&= \mathbb{E}\Big[\mathbb{E}[Y(1) - Y(0)|X]\Big] \\
&= \mathbb{E}\Big[\mathbb{E}[Y(1)|X, D = 1] - \mathbb{E}[Y(0)|X, D = 0]\Big] \\
&= \mathbb{E}\Big[\mathbb{E}[Y|X, D = 1] - \mathbb{E}[Y|X, D = 0]\Big]
\end{aligned}$$

where the first equality holds by definition, the second equality uses the law of iterated expectations, the third equality holds by the unconfoundedness assumption, and the last equality holds by replacing potential outcomes with observed outcomes. Overlap is used in the third equality; without it, $\mathbb{E}[Y(1)|X, D = 1]$ and $\mathbb{E}[Y(0)|X, D = 0]$ would not exist for some values of X.

(c)

Starting from the expression for ATE that we derived in part (b), we have that

$$\begin{aligned}
\text{ATE} &= \mathbb{E}\Big[\mathbb{E}[Y|X, D = 1]\Big] - \mathbb{E}\Big[\mathbb{E}[Y|X, D = 0]\Big] \\
&= \mathbb{E}\left[\frac{\pi}{p(X)}\mathbb{E}[Y|X, D = 1]\middle|D = 1\right] - \mathbb{E}\left[\frac{1-\pi}{1-p(X)}\mathbb{E}[Y|X, D = 0]\middle|D = 0\right] \\
&= \mathbb{E}\left[\frac{\pi}{p(X)}Y\middle|D = 1\right] - \mathbb{E}\left[\frac{1-\pi}{1-p(X)}Y\middle|D = 0\right] \\
&= \mathbb{E}\left[\frac{D}{p(X)}Y\right] - \mathbb{E}\left[\frac{1-D}{1-p(X)}Y\right] \\
&= \mathbb{E}\left[\left(\frac{D}{p(X)} - \frac{1-D}{1-p(X)}\right)Y\right]
\end{aligned}$$

where the first equality holds by the law of iterated expectations, the second equality uses the rules for switching from an unconditional expectation to an expectation conditional on $D = 1$ and $D = 0$, the third equality holds from the law of iterated expectations, the fourth equality holds by converting from conditional to unconditional expectations based on the law of iterated expectations as we discussed in class, and the last equality holds by combining terms. This expression leads to a propensity score re-weighting estimator for ATE where we estimate $p(X)$ and then plug in our estimate into the above expression.

(d)

Consider the following candidate expression fo ATE:

$$\mathbb{E}\Big[\mathbb{E}[Y|X,D=1]-\mathbb{E}[Y|X,D=0]+\frac{D}{p(X)}(Y-\mathbb{E}[Y|X,D=1])-\frac{1-D}{1-p(X)}(Y-\mathbb{E}[Y|X,D=0])\Big].$$

Notice that

$$\mathbb{E}\Big[\mathbb{E}[Y|X,D=1]-\mathbb{E}[Y|X,D=0]\Big].$$

This is equal to ATE as we showed in part (b). Next, notice that

$$\begin{aligned}\mathbb{E}\left[\frac{D}{p(X)}(Y-\mathbb{E}[Y|X,D=1])\right] &= \mathbb{E}\left[\frac{\pi}{p(X)}(Y-\mathbb{E}[Y|X,D=1])\,\middle|\,D=1\right]\\ &= \mathbb{E}\left[\frac{\pi}{p(X)}\underbrace{(\mathbb{E}[Y|X,D=1]-\mathbb{E}[Y|X,D=1])}_{=0}\,\middle|\,D=1\right]=0\end{aligned}$$

Similarly,

$$\begin{aligned}\mathbb{E}\left[\frac{1-D}{1-p(X)}(Y-\mathbb{E}[Y|X,D=0])\right] &= \mathbb{E}\left[\frac{1-\pi}{1-p(X)}(Y-\mathbb{E}[Y|X,D=0])\,\middle|\,D=0\right]\\ &= \mathbb{E}\left[\frac{1-\pi}{1-p(X)}\underbrace{(\mathbb{E}[Y|X,D=0]-\mathbb{E}[Y|X,D=0])}_{=0}\,\middle|\,D=0\right]=0\end{aligned}$$

Thus, the expression in the first display is equal to ATE.

Next, we move to showing that the estimator that we get from this expression is doubly robust. Let $\tilde{p}(X)$ denote a parametric working model for $p(X)$, and let $\tilde{m}_1(X)$ and $\tilde{m}_0(X)$ denote parametric working models for $\mathbb{E}[Y|X,D=1]$ and $\mathbb{E}[Y|X,D=0]$, respectively. Then, the estimand that we get from the above expression when we plug in these working models is

$$\widetilde{ATE}=\mathbb{E}\Big[\tilde{m}_1(X)-\tilde{m}_0(X)+\frac{D}{\tilde{p}(X)}(Y-\tilde{m}_1(X))-\frac{1-D}{1-\tilde{p}(X)}(Y-\tilde{m}_0(X))\Big]$$

Case 1: $\tilde{m}_1(X)=\mathbb{E}[Y|X,D=1]$ and $\tilde{m}_0(X)=\mathbb{E}[Y|X,D=0]$, but it is not necessarily the case that $\tilde{p}(X)=p(X)$.

In this case, we have that

$$\widetilde{ATE}=\mathbb{E}\Big[\mathbb{E}[Y|X,D=1]-\mathbb{E}[Y|X,D=0]+\frac{D}{\tilde{p}(X)}(Y-\mathbb{E}[Y|X,D=1])-\frac{1-D}{1-\tilde{p}(X)}(Y-\mathbb{E}[Y|X,D=0])\Big]$$

Moreover, by essentially the same argument as above, we have that

$$\begin{aligned}\mathbb{E}\left[\frac{D}{\tilde{p}(X)}(Y-\mathbb{E}[Y|X,D=1])\right]&=\mathbb{E}\left[\frac{D}{\tilde{p}(X)}(\mathbb{E}[Y|X,D=1]-\mathbb{E}[Y|X,D=1])\right]=0\\ \mathbb{E}\left[\frac{1-D}{1-\tilde{p}(X)}(Y-\mathbb{E}[Y|X,D=0])\right]&=\mathbb{E}\left[\frac{1-D}{1-\tilde{p}(X)}(\mathbb{E}[Y|X,D=0]-\mathbb{E}[Y|X,D=0])\right]=0\end{aligned}$$

which implies that $\widetilde{ATE} = \mathbb{E}\Big[\mathbb{E}[Y|X, D=1] - \mathbb{E}[Y|X, D=0]\Big] = ATE$.

Case 2: $\tilde{p}(X) = p(X)$, but it is not necessarily the case that $\tilde{m}_1(X) = \mathbb{E}[Y|X, D=1]$ and $\tilde{m}_0(X) = \mathbb{E}[Y|X, D=0]$.

In this case, we have that

$$\widetilde{ATE} = \mathbb{E}\Big[\underbrace{\tilde{m}_1(X) - \tilde{m}_0(X)}_{A} + \underbrace{\frac{D}{p(X)}(Y - \tilde{m}_1(X))}_{B} - \underbrace{\frac{1-D}{1-p(X)}(Y - \tilde{m}_0(X))}_{C}\Big]$$

Notice that

$$\begin{aligned} B &= \mathbb{E}\left[\frac{\pi}{p(X)}(Y - \tilde{m}_1(X))\,\middle|\, D=1\right] \\ &= \mathbb{E}\left[\frac{\pi}{p(X)}(\mathbb{E}[Y|X, D=1] - \tilde{m}_1(X))\,\middle|\, D=1\right] \\ &= \mathbb{E}\left[\mathbb{E}[Y|X, D=1] - \tilde{m}_1(X)\right] \end{aligned}$$

and that

$$\begin{aligned} C &= \mathbb{E}\left[\frac{1-\pi}{1-p(X)}(Y - \tilde{m}_0(X))\,\middle|\, D=0\right] \\ &= \mathbb{E}\left[\frac{1-\pi}{1-p(X)}(\mathbb{E}[Y|X, D=0] - \tilde{m}_0(X))\,\middle|\, D=0\right] \\ &= \mathbb{E}\left[\mathbb{E}[Y|X, D=0] - \tilde{m}_0(X)\right] \end{aligned}$$

Combining these results, we have that

$$\begin{aligned} \widetilde{ATE} &= \mathbb{E}\left[\tilde{m}_1(X) - \tilde{m}_0(X) + \mathbb{E}[Y|X, D=1] - \tilde{m}_1(X) - (\mathbb{E}[Y|X, D=0] - \tilde{m}_0(X))\right] \\ &= \mathbb{E}\Big[\mathbb{E}[Y|X, D=1] - \mathbb{E}[Y|X, D=0]\Big] = \text{ATE} \end{aligned}$$

The results from the two cases show that the estimator that we get based on $\widetilde{ATE}$ is doubly robust.

Question 2

In this question, we will estimate the ATT of a job training program using a number of different techniques that we discussed in class.

One thing to note: for regression, regression adjustment, propensity score re-weighting, and AIPW, your answer should be exactly the same as mine, but we may be get different estimates when we use machine learning, due to sample splitting, choosing tuning parameters, etc. The same is true for the bootstrap standard errors—they should be broadly similar, but we would not expect that mine and yours will be literally the same number.

(a)

To estimate the ATT using regression adjustment, we first need to calculate $\hat{\beta}_0$ from the regression of Y on X using untreated observations only. Once we have this estimate, we can compute

$$\widehat{\text{ATT}} = \frac{1}{n}\sum_{i=1}^{n}\frac{D_i}{\pi}Y_i - \frac{1}{n}\sum_{i=1}^{n}\frac{D_i}{\pi}X_i'\hat{\beta}_0$$

```
# load packages used later
library(pbapply)
library(ranger)

# load data
data <- as.data.frame(haven::read_dta("../data/jtrain_observational.dta"))
Y <- data$re78
D <- data$train
pi1 <- mean(D)
n <- nrow(data)
X <- model.matrix(~age + educ + black + hisp + married + re75 + unem75, data=data)
# run regression using untreated observations only
bet <- solve(t(X)%*%(X*as.numeric((1-D)/pi1)))%*%t(X)%*%as.numeric(Y*(1-D)/pi1)

att1 <- mean(D*Y/pi1)
att2 <- sum(apply(D*X/pi1,2,mean)*as.numeric(bet))
att <- att1 - att2

# report estimate of att
round(att,3)
```

```
[1] 0.859
```

The outcome is in 1000's of dollars, so this indicates that we are estimating that job training increased yearly earnings by \$859.

As a side-comment, it's not immediately clear if this should be interpreted as a large effect or not. One way to think about this is to compute: $\text{ATT}/\mathbb{E}[Y(0)|D=1]$ (i.e., the relative size of the ATT compared to what the average outcome would have been absent the treatment). Further, notice that this is equal to $\text{ATT}/(\mathbb{E}[Y|D=1] - \text{ATT})$ (which holds by adding and subtracting $\mathbb{E}[Y(1)|D=1]$ in the denominator). If we compute this, we get that we have estimated that yearly earnings as about 16% higher from job training relative to what they would have been in the absence of job training.

(b)

Notice that

$$\begin{aligned}
\sqrt{n}(\widehat{\text{ATT}} - \text{ATT}) &= \sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n}\frac{D_i}{\pi}Y_i - \frac{1}{n}\sum_{i=1}^{n}\frac{D_i}{\pi}X_i'\hat{\beta}_0\right) - \sqrt{n}\left(\mathbb{E}\left[\frac{D}{\pi}Y\right] - \mathbb{E}\left[\frac{D}{\pi}X'\right]\beta_0\right) \\
&= \sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n}\frac{D_i}{\pi}Y_i - \mathbb{E}\left[\frac{D}{\pi}Y\right]\right) - \sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n}\frac{D_i}{\pi}X_i' - \mathbb{E}\left[\frac{D}{\pi}X'\right]\right)\hat{\beta}_0 \\
&\quad - \mathbb{E}\left[\frac{D}{\pi}X'\right]\sqrt{n}(\hat{\beta}_0 - \beta_0) \\
&= \sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n}\frac{D_i}{\pi}Y_i - \mathbb{E}\left[\frac{D}{\pi}Y\right]\right) - \sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n}\frac{D_i}{\pi}X_i' - \mathbb{E}\left[\frac{D}{\pi}X'\right]\right)\beta_0 \\
&\quad - \mathbb{E}\left[\frac{D}{\pi}X'\right]\sqrt{n}(\hat{\beta}_0 - \beta_0) + o_p(1)
\end{aligned}$$

where the first line holds by definition, the second line adds and subtracts $\mathbb{E}[(D/\pi)X']\hat{\beta}_0$, and the third equality holds because $\hat{\beta}_0 \xrightarrow{p} \beta_0$ (and by the CMT). You can think of the three terms above as being the parts of the asymptotic distribution that are going to come from estimating $\mathbb{E}[Y|D=1]$, $\mathbb{E}[X|D=1]$, and β_0, respectively. Recalling that,

$$\sqrt{n}(\hat{\beta}_0 - \beta_0) = \mathbb{E}\left[\frac{(1-D)}{(1-\pi)}XX'\right]^{-1}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\frac{(1-D_i)}{(1-\pi)}X_i e_i + o_p(1)$$

we have that

$$\sqrt{n}(\widehat{ATT} - ATT) = \frac{1}{\sqrt{n}}\sum_{i=1}^{n}(A_i - B_i - C_i) + o_p(1)$$

where

$$\begin{aligned}
A_i &= \frac{D_i}{\pi}Y_i - \mathbb{E}\left[\frac{D}{\pi}Y\right] \\
B_i &= \left(\frac{D_i}{\pi}X_i' - \mathbb{E}\left[\frac{D}{\pi}X'\right]\right)\beta_0 \\
C_i &= \mathbb{E}\left[\frac{D}{\pi}X'\right]\mathbb{E}\left[\frac{(1-D)}{(1-\pi)}XX'\right]^{-1}\frac{(1-D_i)}{(1-\pi)}X_i e_i
\end{aligned}$$

Thus, $\sqrt{n}(\widehat{\text{ATT}} - \text{ATT}) \xrightarrow{d} \mathcal{N}(0, V)$ where $V = \mathbb{E}[(A-B-C)^2]$ (we can square here since ATT is a scalar). We can consistently estimate V by replacing all of the population averages by sample averages and replacing β_0 with its consistent estimate $\hat{\beta}_0$.

```
A2 <- mean(D/pi1*Y)
Ai <- D/pi1*Y - A2

XDp <- X*as.numeric(D/pi1)
B2 <- colMeans(XDp)
# sweep subtracts a vector from each row of a matrix
```

```
Bi <- sweep(XDp, 2, B2) %*% bet

C2 <- t(B2)
XUp <- X*as.numeric((1-D)/(1-pi1))
C3 <- t(X)%*%XUp/n
ehat <- Y - X%*%bet
XUpe <- XUp*as.numeric(ehat)
Ci <- as.numeric(C2%*%solve(C3)%*%t(XUpe))

V <- mean( (Ai-Bi-Ci)^2 )
se <- sqrt(V)/sqrt(n)
round(se,3)
```

```
[1] 0.903
```

This indicates that we cannot reject that job training had no effect earnings at conventional significance levels.

(c)

Let's move to computing standard errors using the bootstrap. Towards, this end let's write a function that takes in some data and computes an estimate of ATT (this is essentially just the same code that we used before).

```
compute.att <- function(data) {
  Y <- data$re78
  D <- data$train
  pi1 <- mean(D)
  X <- model.matrix(~age + educ + black + hisp + married + re75 + unem75, data=data)
  # run regression using untreated observations only
  bet <- solve(t(X)%*%(X*as.numeric((1-D)/pi1)))%*%t(X)%*%as.numeric(Y*(1-D)/pi1)

  att1 <- mean(D*Y/pi1)
  att2 <- sum(apply(D*X/pi1,2,mean)*as.numeric(bet))
  att <- att1 - att2
  att
}
```

There is a subtle issue about whether we should treat π as being known or estimated. Above I treated it like it was known. And, for this reason, I am going to draw bootstrap samples from the treated group and untreated group separately (it is not a big deal if you didn't do this though, just noting it so you can understand the code).

```
# now bootstrap
biters <- 1000
treated_data <- subset(data, train==1)
untreated_data <- subset(data, train==0)
n1 <- nrow(treated_data)
```

```r
n0 <- nrow(untreated_data)
boot_res <- pblapply(1:biters, function(b) {
  # draw new data with replacement

  boot_treated_rows <- sample(1:n1, size=n1, replace=TRUE)
  boot_treated <- treated_data[boot_treated_rows,]
  boot_untreated_rows <- sample(1:n0, size=n0, replace=TRUE)
  boot_untreated <- untreated_data[boot_untreated_rows,]
  boot_data <- rbind.data.frame(boot_treated, boot_untreated)

  # alternative code that doesn't treat pi1 as fixed
  #boot_rows <- sample(1:n, size=n, replace=TRUE)
  #boot_data <- data[boot_rows,]

  compute.att(boot_data)
})

# run bootstrap
boot_res <- do.call("rbind", boot_res)

# compute bootstrap standard errors
boot_se <- apply(boot_res, 2, sd)

round(boot_se, 3)
```

```
[1] 0.904
```

These standard errors are similar to the ones we computed before.

(d)

For this part, we are just going to run a regression of Y on D and X.

```r
X2 <- cbind(X,D)
bet2 <- solve(t(X2)%*%X2)%*%t(X2)%*%Y
round(bet2,3)
```

```
              [,1]
(Intercept) -0.061
age         -0.057
educ         0.604
black       -0.597
hisp         2.547
married      1.530
re75         0.788
unem75      -0.079
D            0.525
```

```
ehat <- Y - X2%*%bet2
X2e <- X2*as.numeric(ehat)
Omeg2 <- t(X2e)%*%X2e/n
Q2 <- t(X2)%*%X2/n
V2 <- solve(Q2)%*%Omeg2%*%solve(Q2)
se2 <- sqrt(diag(V2))/sqrt(n)
round(se2,3)
```

```
(Intercept)         age        educ       black        hisp     married
      1.588       0.025       0.096       0.462       1.271       0.517
       re75      unem75           D
      0.036       0.967       0.884
```

The estimated coefficient on D is somewhat closer to 0 than we computed in the first part while the standard errors are about the same. In some sense, this doesn't appear to matter much, because in both cases we are estimating a small positive (and not statistically significant effect) of job training. However, this is mostly a result of us not being able to precisely estimate effects of job training. That said, our earlier point estimate is about 64% larger than the one from the regression which is arguably meaningfully different even though the identifying assumptions are the same.

(e)

Next, we will estimate the ATT using propensity score re-weighting. Since we will use the bootstrap to compute standard errors, let's write a function that computes an estimate of the ATT given some data—we'll use this function to estimate the ATT itself and inside our bootstrap iterations.

```
compute.att_ipw <- function(data) {
  Y <- data$re78
  D <- data$train
  pi1 <- mean(D)

  logit_est <- glm(train ~ age + educ + black + hisp + married + re75 + unem75,
                   data=data, family=binomial(link="logit"))
  pscore <- predict(logit_est, type="response")
  att <- mean( ( (D/pi1) - (1-D)/pi1*pscore/(1-pscore) ) * Y )
  att
}
att_ipw <- compute.att_ipw(data)
```

and now code to compute bootstrapped standard errors

```
boot_res <- pblapply(1:biters, function(b) {
  # draw new data with replacement

  boot_treated_rows <- sample(1:n1, size=n1, replace=TRUE)
  boot_treated <- treated_data[boot_treated_rows,]
  boot_untreated_rows <- sample(1:n0, size=n0, replace=TRUE)
  boot_untreated <- untreated_data[boot_untreated_rows,]
```

```
  boot_data <- rbind.data.frame(boot_treated, boot_untreated)

  # alternative code that doesn't treat pi1 as fixed
  #boot_rows <- sample(1:n, size=n, replace=TRUE)
  #boot_data <- data[boot_rows,]

  compute.att_ipw(boot_data)
})

# run bootstrap
boot_res <- do.call("rbind", boot_res)

# compute bootstrap standard errors
boot_se <- apply(boot_res, 2, sd)

# report results
data.frame(att_ipw=round(att_ipw,3), se=round(boot_se, 3))
```

```
  att_ipw    se
1   0.458 1.082
```

This estimate is somewhat lower than regression adjustment and the bootstrap standard errors have a similar magnitude.

(f)

Next, we will estimate the ATT using the AIPW/doubly robust approach discussed in class.

```
compute.att_dr <- function(data) {
  Y <- data$re78
  D <- data$train
  pi1 <- mean(D)
  X <- model.matrix(~age + educ + black + hisp + married + re75 + unem75, data=data)
  bet <- solve(t(X)%*%(X*as.numeric((1-D)/pi1)))%*%t(X)%*%as.numeric(Y*(1-D)/pi1)
  m <- X%*%bet

  logit_est <- glm(train ~ age + educ + black + hisp + married + re75 + unem75,
                   data=data, family=binomial(link="logit"))
  pscore <- predict(logit_est, type="response")
  att <- mean( ( (D/pi1) - (1-D)/pi1*pscore/(1-pscore) ) * ( Y - m ) )
  att
}
att_dr <- compute.att_dr(data)
```

and now code to compute bootstrapped standard errors

```
boot_res <- pblapply(1:biters, function(b) {
  # draw new data with replacement

  boot_treated_rows <- sample(1:n1, size=n1, replace=TRUE)
  boot_treated <- treated_data[boot_treated_rows,]
  boot_untreated_rows <- sample(1:n0, size=n0, replace=TRUE)
  boot_untreated <- untreated_data[boot_untreated_rows,]
  boot_data <- rbind.data.frame(boot_treated, boot_untreated)

  # alternative code that doesn't treat pi1 as fixed
  #boot_rows <- sample(1:n, size=n, replace=TRUE)
  #boot_data <- data[boot_rows,]

  compute.att_dr(boot_data)
})

# run bootstrap
boot_res <- do.call("rbind", boot_res)

# compute bootstrap standard errors
boot_se <- apply(boot_res, 2, sd)

# report results
data.frame(att_dr=round(att_dr,3), se=round(boot_se, 3))
```

```
  att_dr    se
1   0.62 0.977
```

Here the estimate is in between regression adjustment and propensity score re-weighting. The bootstrap standard errors are similar to the ones we computed before.

(g)

Next, we will estimate the ATT using machine learning.

```
set.seed(1234)

# create folds
K <- 5
data$fold <- sample(1:K, n, replace=TRUE)

# inner function to compute an att for f2 using f1 to estimate the
# preliminary models
ml_att_inner <- function(this_fold_data, other_folds_data) {
  # estimate nuisance function using other folds
  Dmod <- ranger(as.factor(train) ~ age + educ + black + hisp +
                                    married + re75 + unem75,
                 data=other_folds_data, probability=TRUE)
```

```
  Ymod <- ranger(re78 ~ age + educ + black + hisp +
                        married + re75 + unem75,
                 data=other_folds_data)

  # get predictions with this fold data
  pscore <- predict(Dmod, data=this_fold_data, probability=TRUE)$predictions[,"1"]
  out_reg <- predict(Ymod, data=this_fold_data)$predictions

  # compute att(k) with this fold data
  D <- this_fold_data$train
  p <- mean(D)
  Y <- this_fold_data$re78
  psi1 <- (D/pi1)*(Y-out_reg)
  psi2 <- (1-D)/pi1 * pscore/(1-pscore) * (Y-out_reg)
  att <- mean(psi1 - psi2)
  inf_func <- psi1 - psi2
  list(att=att, inf_func=inf_func)
}

# cross splitting
ml_att <- numeric(K)
ml_inf_func <- c()
for (k in 1:K) {
  fold_k <- subset(data, fold==k)
  other_folds <- subset(data, fold!=k)
  fold_k_res <- ml_att_inner(fold_k, other_folds)
  ml_att[k] <- fold_k_res$att
  ml_inf_func <- c(ml_inf_func, fold_k_res$inf_func)
}
att_ml <- mean(ml_att)
se_ml <- sd(ml_inf_func)/sqrt(n)
data.frame(att_ml=round(att_ml,3), se_ml=round(se_ml, 3))
```

```
  att_ml se_ml
1 -0.585 0.885
```

This estimate has the opposite sign of the other estimates, though it is not statistically different from 0.

(h)

For this problem, since the treatment is randomly assigned, we can just run a regression of Y on D and ignore X.

```
# load data
library(estimatr)
exp_data <- as.data.frame(haven::read_dta("../data/jtrain_experimental.dta"))
```

```r
exp_att_reg <- lm_robust(re78 ~ train, data=exp_data)
summary(exp_att_reg)
```

```
Call:
lm_robust(formula = re78 ~ train, data = exp_data)

Standard error type:  HC2

Coefficients:
            Estimate Std. Error t value  Pr(>|t|) CI Lower CI Upper  DF
(Intercept)    4.555     0.3401  13.393 1.391e-34   3.8864    5.223 443
train          1.794     0.6710   2.674 7.769e-03   0.4756    3.113 443

Multiple R-squared:  0.01782 ,  Adjusted R-squared:  0.01561
F-statistic: 7.151 on 1 and 443 DF,  p-value: 0.007769
```

Here the estimated ATT is about \$1790, which is much notably higher than our previous estimates, and it is statistically significant. Relative to this, the previous estimates were not horrible, but none of them would have recovered the qualitatively correct result. To me, this suggests that the version of unconfoundedness that we used was not correct—probably we needed to control for more variables, some of which may be unobserved. And a good lesson from this is that using advanced approaches (e.g., machine learning) is not a fix for a failure of the identifying assumptions.

Question 3

The starting point for this question is what we called Decomposition 2 in class:

$$\begin{aligned}\alpha = \mathbb{E}\Big[w(D,X)\Big(\mathbb{E}[Y|X,D=1] - \mathbb{E}[Y|X,D=0]\Big)\Big] \\ + \mathbb{E}[w(D,X)\Big(\mathbb{E}[Y|X,D=0] - \mathrm{L}_0(Y|X)\Big)]\end{aligned}$$

where

$$w(D,X) = \frac{D(1-\mathrm{L}(D|X))}{\mathbb{E}[(D-\mathrm{L}(D|X))^2]}$$

Under unconfoundedness, $\mathbb{E}[Y|X,D=1] - \mathbb{E}[Y|X,D=0] = CATE(X)$, so we mainly need to show that the second term is equal to 0 if either of the conditions in the problem hold. Condition (ii), that $\mathbb{E}[Y|X,D=0] = \mathrm{L}_0(Y|X)$, immediately implies that it is equal to 0. For condition (i), that

$p(X) = \mathrm{L}(D|X)$, ignoring the denominator of the weights, we have that

$$\begin{aligned}
&\mathbb{E}\Big[D(1-\mathrm{L}(D|X))\Big(\mathbb{E}[Y|X,D=0]-\mathrm{L}_0(Y|X)\Big)\Big] \\
&\qquad = \mathbb{E}\Big[D(1-p(X))\Big(\mathbb{E}[Y|X,D=0]-\mathrm{L}_0(Y|X)\Big)\Big] \\
&\qquad = \mathbb{E}\Big[p(X)(1-p(X))\Big(\mathbb{E}[Y|X,D=0]-\mathrm{L}_0(Y|X)\Big)\Big] \\
&\qquad = \mathbb{E}\Big[p(X)(1-\pi)\Big(\mathbb{E}[Y|X,D=0]-\mathrm{L}_0(Y|X)\Big)\Big|D=0\Big] \\
&\qquad = \mathbb{E}\Big[p(X)Y\Big|D=0\Big](1-\pi) - \mathbb{E}\Big[\mathrm{L}(D|X)\mathrm{L}_0(Y|X)\Big)\Big|D=0\Big](1-\pi) \\
&\qquad = \mathbb{E}\Big[p(X)Y\Big|D=0\Big](1-\pi) - \mathbb{E}\Big[\mathrm{L}(D|X)Y\Big|D=0\Big](1-\pi) = 0
\end{aligned}$$

where the first equality holds by condition (i), the second equality uses the law of iterated expectations (and that $\mathbb{E}[D|X] = p(X)$), the third equality uses one of the rules we discussed in class for switching from an unconditional expectation to an expectation conditional on $D = 0$, the fourth equality splits the expectation into two parts and applies the law of iterated expectations on the first term, the fifth equality holds by the property of linear projections that we derived in class in Equation (3) in the course notes, and the last equality holds by condition (i). This shows that the second term is equal to 0 under condition (i).

That the weights are non-negative under condition (i) follows essentially immediately. The denominator is positive since it involves the expectation of a quadratic term. Under condition (i), the numerator must be positive because $\mathrm{L}(D|X) = p(X) \leq 1$, which implies that the numerator is non-negative.