Homework 5 Solutions

Additional Question 1

First, let's estimate the ATT. We will estimate $\hat{\beta}$ from the regression of Y on X using untreated observations only. Once we have this estimate, we can compute

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

```r
data <- as.data.frame(haven::read_dta("jtrain_observational.dta"))
Y <- data$re78
D <- data$train
p <- 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)/p)))%*%t(X)%*%as.numeric(Y*(1-D)/p)

att1 <- mean(D*Y/p)
att2 <- sum(apply(D*X/p,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: $ATT/\mathbb{E}[Y(0)|D=1]$ (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 $ATT/(\mathbb{E}[Y|D=1] - ATT)$ (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.

For the second part, notice that

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

where the first line holds by definition, the second line adds and subtracts $\mathbb{E}[(D/p)X']\hat{\beta}$, and the third equality holds because $\hat{\beta} \xrightarrow{p} \beta$ (and by the CMT). Recalling that,

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

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

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

Thus, $\sqrt{n}(\widehat{ATT} - ATT) \xrightarrow{d} N(0, V)$ 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 with its consistent estimate $\hat{\beta}$.

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

XDp <- X*as.numeric(D/p)
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-p))
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.

Next, 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
  p <- 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)/p)))%*%t(X)%*%as.numeric(Y*(1-D)/p)

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

There is a subtle issue about whether we should treat p 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)
n0 <- nrow(untreated_data)
library(pbapply)
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,]
```

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

  # alternative code that doesn't treat p 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] 0.939
```

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

For the last 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
```

```r
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.

Additional Question 2

```r
set.seed(1234)
library(randomForest)
data <- as.data.frame(haven::read_dta("jtrain_observational.dta"))

n <- nrow(data)
data$id <- 1:n
fold1 <- subset(data, (id%%2) == 0)
fold2 <- subset(data, (id%%2) == 1)

ml_att <- function(f1, f2) {

  # use f1 to estimate the first step models
  Dmod <- randomForest(as.factor(train) ~ age + educ + black + hisp +
                       married + re75 + unem75, data=f1)
  Ymod <- randomForest(re78 ~ age + educ + black + hisp +
                       married + re75 + unem75, data=f1)

  # get predictions with f2
  pscore <- predict(Dmod, newdata=f2, type="prob")[,2] # this gets p(d=1|x)
  out_reg <- predict(Ymod, newdata=f2)

  # compute att(k) with f2
  D <- f2$train
  p <- mean(D)
  Y <- f2$re78
  att1 <- mean(D/p*(Y-out_reg))
  att2 <- mean((1-D)/p * pscore/(1-pscore) * (Y-out_reg) )
  att <- att1-att2
  att
}

# cross splitting
ml1 <- ml_att(fold1,fold2)
# reverse roles
ml2 <- ml_att(fold2,fold1)
# average
mean(c(ml1,ml2))
```

```
## [1] 1.580708
```

This is larger than what we estimated on the previous question (about 0.85) though I noticed that my estimates do move somewhat if I run the code multiple times.

Additional Question 3

Yes, they are correct. Notice that, because there are only two periods, this regression is equivalent to

$$\Delta Y_{i2} = \Delta\theta_2 + \alpha D_{i2} + \Delta e_{i2}$$

Moreover,

$$\mathbb{E}[\Delta Y_2 | D_2 = 1] = \Delta\theta_2 + \alpha \tag{1}$$

$$\mathbb{E}[\Delta Y_2 | D_2 = 0] = \Delta\theta_2 \tag{2}$$

which implies that

$$\begin{aligned}
\alpha &= \mathbb{E}[\Delta Y_2 | D_2 = 1] - \mathbb{E}[\Delta Y_2 | D_2 = 0] \\
&= \mathbb{E}[\Delta Y_2 | D_2 = 1] - \mathbb{E}[\Delta Y_2(0) | D_2 = 0] \\
&= \mathbb{E}[\Delta Y_2 | D_2 = 1] - \mathbb{E}[\Delta Y_2(0) | D_2 = 1] \\
&= ATT
\end{aligned}$$

where the first line subtracts Equation 2 from Equation 1, the second equality writes the second term in terms of potential outcomes, and the third equality holds by the parallel trends assumption.