

PS C236A/ Stat C239A

Problem Set 4

Due: October 2, 2009

Problem 1 Consider an observational study, where $Z_i = 1$ if unit i is in the treatment group and $Z_i = 0$ if unit i is in the control group. Let X be a vector of observed pretreatment covariates. Write $X_{Z=1}$ for the observed covariates of the units in the treatment group. Similarly, let $X_{Z=0}$ be the observed covariates in the control group. Let r_1 be outcome under treatment and r_0 be the outcome under control.

Assume the following:

$$\begin{aligned} r_0 &\perp\!\!\!\perp Z|X_{Z=1} \\ P(Z = 1|X_{Z=1}) &< 1 \end{aligned}$$

Suppose you know the propensity score $e(X) = P(Z = 1)$ for all units i .

With these assumptions, can conditioning on the propensity score estimate the ATT without bias? Prove it mathematically and describe your logic in words. What additional assumption would we need in order to estimate the ATE without bias?

Hint: First show that conditioning on the propensity score is equivalent to conditioning on $X_{Z=1}$. Then show that conditioning on the propensity score can produce unbiased ATT estimates under the assumptions above.

Solution:

First we need to show that $Z \perp\!\!\!\perp X|e(X)$, which is equivalent to showing $P(Z|X) = P(Z|e(X))$.

$$\begin{aligned} P(Z|e(X)) &= E[Z|e(X)] = E[E[Z|e(x), x]|e(x)] = E[P(Z|X, e(X))|e(x)] \\ &= E[e(x)|e(x)] = e(x) = P(Z|X) \end{aligned}$$

Now we need to show that conditioning on the propensity score and under the stated assumptions, that the ATT is identified.

We want to estimate $E((r_1 - r_0)|Z = 1)$, which is the ATT estimand. Note that conditioning on $Z = 1$ is equivalent to conditioning on $X|Z = 1$, which—as we proved above—is equivalent to conditioning on $e(X)|Z = 1$. We can observe without assumptions:

$$E(r_1|Z = 1) - E(r_0|Z = 0)$$

Because we assume that $r_0 \perp\!\!\!\perp Z$, $E(r_0|Z = 0)$ can be rewritten as $E(r_0|Z = 1)$. As a result,

$$E(r_1|Z = 1) - E(r_0|Z = 1) = E(r_1 - r_2|Z = 1)$$

Problem 2 Write a paragraph or two on your plans for your final paper. If you have several ideas and would like feedback on the feasibility/suitability of each of your ideas, that's fine too.

Problem 3: This problem is based on Sekhon's analysis of the voting irregularities in the 2004 election in Florida. There was a lot of speculation that "the optical voting machines that [were] used in a majority of Florida counties caused John Kerry to receive fewer votes than 'Direct Recording Electronic' (DRE) voting machines". The paper can be downloaded at:

<http://sekhon.berkeley.edu/papers/SekhonOpticalMatch.pdf>

- First, using Bush's vote percentage in 2004 as an outcome, run three linear models of the effect of using an electronic voting machine. In each model, include different explanatory variables. How does changing the model change the estimate of the effect of having an electronic voting machine as well as the significance of this estimate. Does the point estimate move? Should we be concerned, and if so, why?

```
> modell = lm(bush04 ~ etouch)
> summary(modell)
```

```
Call:
lm(formula = bush04 ~ etouch)
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-0.311761 -0.054553 -0.004049  0.077816  0.181817
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.60975    0.01441  42.327  <2e-16 ***
etouch      -0.06502    0.03045  -2.136  0.0365 *
---
Signif. codes:  0  ***  0.001  **  0.01  *  0.05  .  0.1  1
```

```
Residual standard error: 0.1039 on 65 degrees of freedom
Multiple R-squared:  0.06556, Adjusted R-squared:  0.05118
F-statistic:  4.56 on 1 and 65 DF,  p-value:  0.03649
```

```
> model2 = lm(bush04 ~ etouch + hisp00 + black00 + lowEduc00 +
              foreignBorn00 + income)
> summary(model2)
```

```
Call:
lm(formula = bush04 ~ etouch + hisp00 + black00 + lowEduc00 +
    foreignBorn00 + income)
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-0.112792 -0.049716 -0.009212  0.052389  0.209099
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.694e-01  1.021e-01  4.596  2.27e-05 ***
etouch      -3.032e-02  2.739e-02  -1.107  0.27280
hisp00       7.539e-03  3.112e-01  0.024  0.98075
black00     -5.521e-01  1.012e-01  -5.455  9.78e-07 ***
lowEduc00    2.122e+00  6.372e-01  3.330  0.00149 **
foreignBorn00 -8.344e-01  3.763e-01  -2.218  0.03039 *
income       4.541e-06  2.254e-06  2.015  0.04841 *
---
Signif. codes:  0  ***  0.001  **  0.01  *  0.05  .  0.1  1
```

```
Residual standard error: 0.07658 on 60 degrees of freedom
Multiple R-squared:  0.5313, Adjusted R-squared:  0.4844
F-statistic: 11.33 on 6 and 60 DF,  p-value:  1.991e-08
```

```
> model3 = lm(bush04 ~ etouch + black00 + white00 +
              income + votePer96.rep + votePer00.rep)
> summary(model3)
```

```
Call:
lm(formula = bush04 ~ etouch + black00 + white00 + income + votePer96.rep +
    votePer00.rep)
```

```
Residuals:
      Min       1Q   Median       3Q      Max
-0.0553277 -0.0143562  0.0001657  0.0107277  0.0685523
```

Coefficients:

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.235e-02  2.310e-01   0.356   0.7228
etouch       -8.489e-03  7.073e-03  -1.200   0.2348
black00      -1.181e-01  2.345e-01  -0.504   0.6163
white00      -3.505e-03  2.353e-01  -0.015   0.9882
income       -1.282e-06  7.274e-07  -1.762   0.0831 .
votePer96.rep -2.268e-01  1.040e-01  -2.181   0.0331 *
votePer00.rep  1.240e+00  8.303e-02  14.934  <2e-16 ***
```

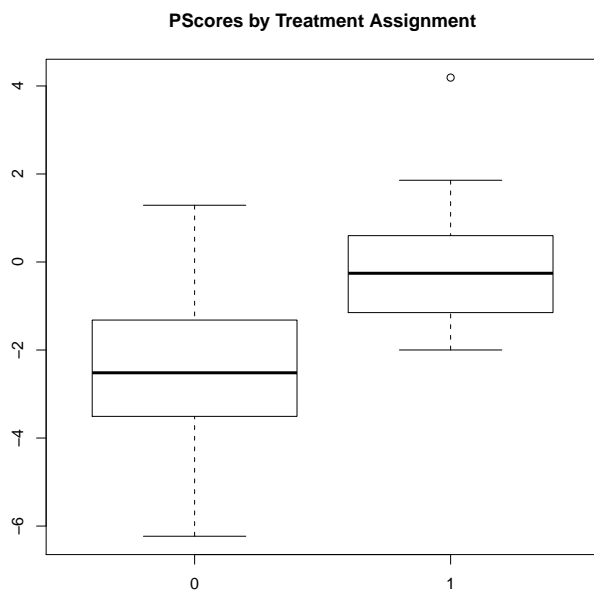
```
---
Signif. codes:  0  ***  0.001  **  0.01  *  0.05  .  0.1  1
```

```
Residual standard error: 0.02088 on 60 degrees of freedom
Multiple R-squared:  0.9652, Adjusted R-squared:  0.9617
F-statistic:  277 on 6 and 60 DF,  p-value: < 2.2e-16
```

We can see that the estimate moves around and the significance changes which is cause for concern. We know that etouch (our treatment indicator) is not orthogonal to the covariates and could be confounded. It also depends on what variables we include what the substantive outcome of our model is.

- b. Now, calculate a propensity score for assignment to treatment (defined as having an electronic voting machine). Provide some justification for your pscore model. Make boxplots showing the distribution of the propensity score for both treated and control groups.

```
> pscore = glm(etouch ~ foreignBorn00 + black00 + income + votePer00.rep + lowEduc00 +
> pdf(file = "./pscore.pdf")
> boxplot(pscore ~ etouch, main = "PScores by Treatment Assignment")
> dev.off()
```



The important (and difficult part) here was to find a pscore that still provided for some overlap between the two groups. This pscore shows a fair bit of separation, but if one includes all of the variables provided in the data set then one gets near perfect separation. Perfect separation leads to bad matches and R often has a warning that you have near-perfect separation. When providing justification for a pscore, it should provide decent balance on covariates and should have an overlap between the two groups.

- c. Write your own univariate nearest-neighbor matching function. In this function, include option to pass in a caliper. Run this function twice, once without an enforced caliper and once with an enforced caliper. How small must your caliper be before it changes your results in a significant way? [Hint: You can effectively not enforce a caliper if you pass in a very large number for your caliper]

```
> # part c)
>
> nn.att = function(x, treat, cal = 100) {
+
+   # make my treated and control groups and their respective indexes
+   index = 1:length(x)
+
+   tr = x[treat == 1]
+   index.tr = index[treat == 1]
+   co = x[treat == 0]
+   index.co = index[treat == 0]
+
+   # initialize my matched data set
+   match.data = NULL
+
+   # cycle through for each element in the treated group
+   # since I am trying to find att
+   for(i in 1:length(tr)) {
+
+       #calculate the distance
+       dis = abs(co - tr[i])
+
+       # find the minimum one(s)
+       match.index = which(dis == min(dis))
+
+       # create the matches. The rep() statements are for ties
+       # where we repeat the index for however many ties there are,
+       # repeat the treatment value for however many ties there are,
+       # the index of the nearest neighbor(s), the values for the
+       # control units, the distance of those units, and the weight
+       # for each unit. If there is only one nearest neighbor
+       # then this will only have one row, otherwise it will have
+       # a row for each neighbor
+
+       matches = cbind(rep(index.tr[i], length(match.index)),
+                       rep(tr[i], length(match.index)), index.co[match.index],
+                       co[match.index], dis[match.index],
+                       rep(1 / length(match.index), length(match.index)))
+
+       # attach the matches to the bottom of my matched data set
+       match.data = rbind(match.data, matches)
+   }
+
+   # find which rows are above the caliper
+   tooBig = which(match.data[,5] > (cal * sd(x)) )
+
+   dropped = NULL
}
```

```

+     if( length(tooBig) > 0) {
+
+         # make a dropped matrix
+         dropped = rbind(match.data[tooBig, c(1, 2)])
+         dropped = as.matrix(dropped)
+         colnames(dropped) = c("index.tr","xt")
+
+         # remove those rows which exceeded the caliper
+         match.data = match.data[-tooBig, ]
+     }
+
+     # give names to my match.data object and return that and the dropped
+     match.data = as.data.frame(match.data)
+     colnames(match.data) = c("index.tr","xt","index.co","xc","dis","wts")
+     return(list(match.data = match.data, dropped = dropped, index.tr = match.data[,
+         index.co = match.data[,3], weights = match.data[,6]))
+ }
>
> nn.pscore = nn.att(pscore, treat = etouch)
> nn.pscore
$match.data
  index.tr      xt index.co      xc      dis wts
6         6 0.90609878     44 0.6558173 0.2502814419 1
8         8 -0.42111576     48 -0.3059262 0.1151895654 1
11        11 1.85739044     17 1.2897144 0.5676759995 1
28        28 -0.14918133     48 -0.3059262 0.1567448638 1
30        30 -0.42967477     48 -0.3059262 0.1237485713 1
34        34 -1.32185876      1 -1.4235155 0.1016567029 1
35        35 -0.01598848      5 0.0444192 0.0604076747 1
42        42 0.29129027     59 0.1915047 0.0997855566 1
43        43 4.19047006     17 1.2897144 2.9007556174 1
45        45 -1.55469834     24 -1.5554805 0.0007821928 1
50        50 0.97388188     17 1.2897144 0.3158325588 1
51        51 -1.06094513     26 -1.0052401 0.0557050500 1
52        52 -0.25619069     48 -0.3059262 0.0497355054 1
56        56 -1.23886221     40 -1.2152564 0.0236057838 1
60        60 -1.99917165     25 -1.9672292 0.0319424703 1

$dropped
NULL

$index.tr
[1] 6 8 11 28 30 34 35 42 43 45 50 51 52 56 60

$index.co
[1] 44 48 17 48 48 1 5 59 17 24 17 26 48 40 25

$weights
[1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

>
> nn.pscore.call = nn.att(pscore, treat = etouch, cal = 1)
> nn.pscore.call
$match.data
  index.tr      xt index.co      xc      dis wts
6         6 0.90609878     44 0.6558173 0.2502814419 1
8         8 -0.42111576     48 -0.3059262 0.1151895654 1

```

```

11      11  1.85739044      17  1.2897144  0.5676759995  1
28      28 -0.14918133      48 -0.3059262  0.1567448638  1
30      30 -0.42967477      48 -0.3059262  0.1237485713  1
34      34 -1.32185876      1  -1.4235155  0.1016567029  1
35      35 -0.01598848      5   0.0444192  0.0604076747  1
42      42  0.29129027      59  0.1915047  0.0997855566  1
45      45 -1.55469834      24 -1.5554805  0.0007821928  1
50      50  0.97388188      17  1.2897144  0.3158325588  1
51      51 -1.06094513      26 -1.0052401  0.0557050500  1
52      52 -0.25619069      48 -0.3059262  0.0497355054  1
56      56 -1.23886221      40 -1.2152564  0.0236057838  1
60      60 -1.99917165      25 -1.9672292  0.0319424703  1

```

\$dropped

```

      index.tr      xt
[1,]      43  4.19047

```

\$index.tr

```
[1] 6 8 11 28 30 34 35 42 45 50 51 52 56 60
```

\$index.co

```
[1] 44 48 17 48 48 1 5 59 24 17 26 48 40 25
```

\$weights

```
[1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

>

```
> nn.pscore.cal.1 = nn.att(pscore, treat = etouch, cal = 0.1)
```

```
> nn.pscore.cal.1
```

\$match.data

```

      index.tr      xt index.co      xc      dis wts
8      8 -0.42111576      48 -0.3059262  0.1151895654  1
28      28 -0.14918133      48 -0.3059262  0.1567448638  1
30      30 -0.42967477      48 -0.3059262  0.1237485713  1
34      34 -1.32185876      1  -1.4235155  0.1016567029  1
35      35 -0.01598848      5   0.0444192  0.0604076747  1
42      42  0.29129027      59  0.1915047  0.0997855566  1
45      45 -1.55469834      24 -1.5554805  0.0007821928  1
51      51 -1.06094513      26 -1.0052401  0.0557050500  1
52      52 -0.25619069      48 -0.3059262  0.0497355054  1
56      56 -1.23886221      40 -1.2152564  0.0236057838  1
60      60 -1.99917165      25 -1.9672292  0.0319424703  1

```

\$dropped

```

      index.tr      xt
6      6  0.9060988
11     11  1.8573904
43     43  4.1904701
50     50  0.9738819

```

\$index.tr

```
[1] 8 28 30 34 35 42 45 51 52 56 60
```

\$index.co

```
[1] 48 48 48 1 5 59 24 26 48 40 25
```

\$weights

```
[1] 1 1 1 1 1 1 1 1 1 1 1
```

We see that we have to make the caliper pretty small to make it drop more than one or two observations.

- d. Plot the density of some key covariates before matching, after matching with a caliper, and after matching without a caliper. What can you conclude?

```
# before
pdf(file = "./before.pdf")
par(mfrow = c(2, 2))
plot(density(votePer00.rep[etouch == 1]),
     main = "Republican Vote Percentage '00 \n Before", col = "blue",
     lty = 1, ylim = c(0, 5))
points(density(votePer00.rep[etouch == 0]), type = "l", lty = 2, col = "red")
legend("topleft", c("Treated", "Control"), col = c("blue", "red"), lty = c(1, 2))

plot(density(black00[etouch == 1]), main = "Black Percentage '00 \n Before",
     col = "blue", lty = 1, ylim = c(0, 7))
points(density(black00[etouch == 0]), type = "l", lty = 2, col = "red")
legend("topright", c("Treated", "Control"), col = c("blue", "red"), lty = c(1, 2))

plot(density(turnout00[etouch == 1]), main = "Turnout '00 \n Before",
     col = "blue", lty = 1, ylim = c(0, 15), xlim = c(0.5, 0.9))
points(density(turnout00[etouch == 0]), type = "l", lty = 2, col = "red")
legend("topright", c("Treated", "Control"), col = c("blue", "red"), lty = c(1, 2))

plot(density(foreignBorn00[etouch == 1]),
     main = "Foreign Born Percentage '00 \n Before", col = "blue",
     lty = 1, ylim = c(0, 12))
points(density(foreignBorn00[etouch == 0]), type = "l", lty = 2, col = "red")
legend("topright", c("Treated", "Control"), col = c("blue", "red"), lty = c(1, 2))
dev.off()

# no caliper
pdf(file = "./noCal.pdf")
par(mfrow = c(2, 2))
plot(density(votePer00.rep[nn.pscore$index.tr]),
     main = "Republican Vote Percentage '00 \n No Caliper", col = "blue",
     lty = 1, ylim = c(0, 10))
points(density(votePer00.rep[nn.pscore$index.co]), type = "l", lty = 2, col = "red")
legend("topright", c("Treated", "Control"), col = c("blue", "red"), lty = c(1, 2))

plot(density(black00[nn.pscore$index.tr]),
     main = "Black Percentage '00 \n No Caliper", col = "blue",
     lty = 1, ylim = c(0, 7))
points(density(black00[nn.pscore$index.co]), type = "l", lty = 2, col = "red")
legend("topright", c("Treated", "Control"), col = c("blue", "red"), lty = c(1, 2))

plot(density(turnout00[nn.pscore$index.tr]),
     main = "Turnout '00 \n No Caliper", col = "blue", lty = 1,
     ylim = c(0, 15), xlim = c(0.5, 0.9))
points(density(turnout00[nn.pscore$index.co]), type = "l", lty = 2, col = "red")
legend("topright", c("Treated", "Control"), col = c("blue", "red"), lty = c(1, 2))

plot(density(foreignBorn00[nn.pscore$index.tr]),
     main = "Foreign Born Percentage '00 \n No Caliper", col = "blue",
     lty = 1, ylim = c(0, 8))
```

```

points(density(foreignBorn00[nn.pscore$index.co]), type = "l", lty = 2, col = "red")
legend("topright", c("Treated", "Control"), col = c("blue", "red"), lty = c(1, 2))
dev.off()

# 0.1 caliper
pdf(file = "./withCal.pdf")
par(mfrow = c(2, 2))
plot(density(votePer00.rep[nn.pscore.cal.1$index.tr]),
     main = "Republican Vote Percentage '00 \n Caliper = 0.1",
     col = "blue", lty = 1, ylim = c(0, 9))
points(density(votePer00.rep[nn.pscore.cal.1$index.co]), type = "l", lty = 2,
       col = "red")
legend("topright", c("Treated", "Control"), col = c("blue", "red"), lty = c(1, 2))

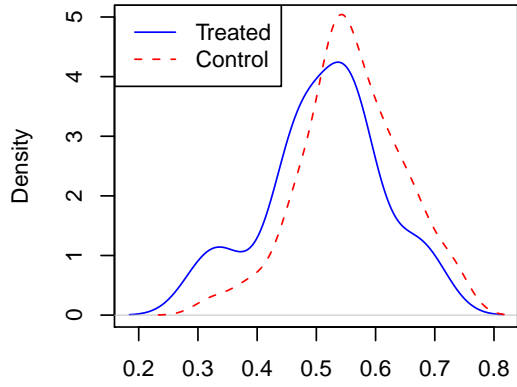
plot(density(black00[nn.pscore.cal.1$index.tr]),
     main = "Black Percentage '00 \n Caliper = 0.1", col = "blue", lty = 1,
     ylim = c(0, 10), xlim = c(-0.05, 0.3))
points(density(black00[nn.pscore.cal.1$index.co]), type = "l", lty = 2, col = "red")
legend("topright", c("Treated", "Control"), col = c("blue", "red"), lty = c(1, 2))

plot(density(turnout00[nn.pscore.cal.1$index.tr]),
     main = "Turnout '00 \n Caliper = 0.1", col = "blue", lty = 1,
     ylim = c(0, 18), xlim = c(0.5, 0.8))
points(density(turnout00[nn.pscore.cal.1$index.co]), type = "l", lty = 2, col = "red")
legend("topleft", c("Treated", "Control"), col = c("blue", "red"), lty = c(1, 2))

plot(density(foreignBorn00[nn.pscore.cal.1$index.tr]),
     main = "Foreign Born Percentage '00 \n Caliper = 0.1", col = "blue", lty = 1,
     ylim = c(0, 15), xlim = c(0, 0.3))
points(density(foreignBorn00[nn.pscore.cal.1$index.co]), type = "l", lty = 2,
       col = "red")
legend("topright", c("Treated", "Control"), col = c("blue", "red"), lty = c(1, 2))
dev.off()

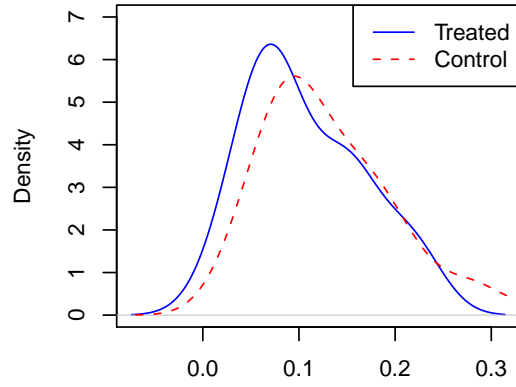
```

**Republican Vote Percentage '00
Before**



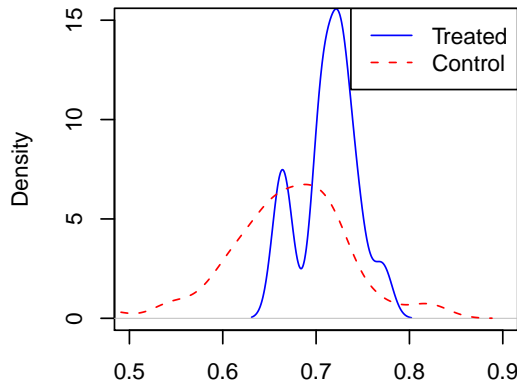
N = 15 Bandwidth = 0.04167

**Black Percentage '00
Before**



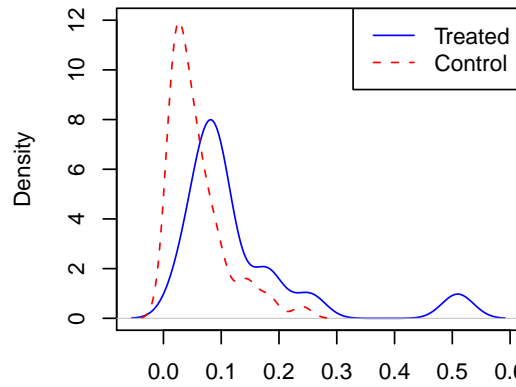
N = 15 Bandwidth = 0.03185

**Turnout '00
Before**



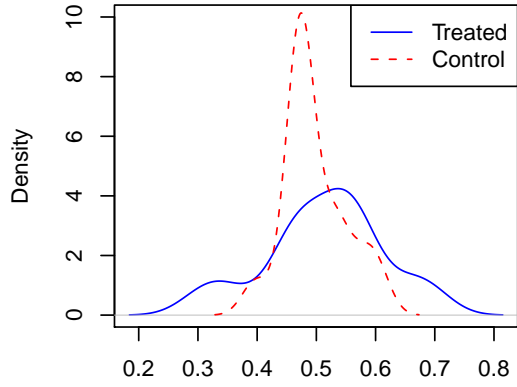
N = 15 Bandwidth = 0.01023

**Foreign Born Percentage '00
Before**



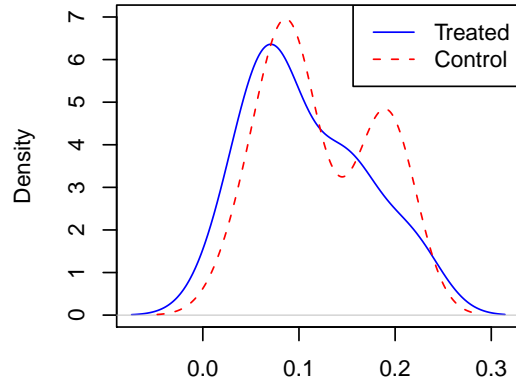
N = 15 Bandwidth = 0.02727

**Republican Vote Percentage '00
No Caliper**



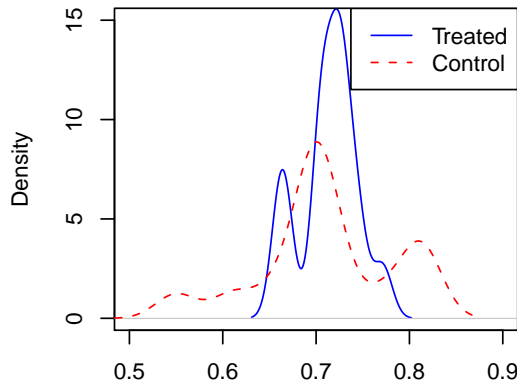
N = 15 Bandwidth = 0.04167

**Black Percentage '00
No Caliper**



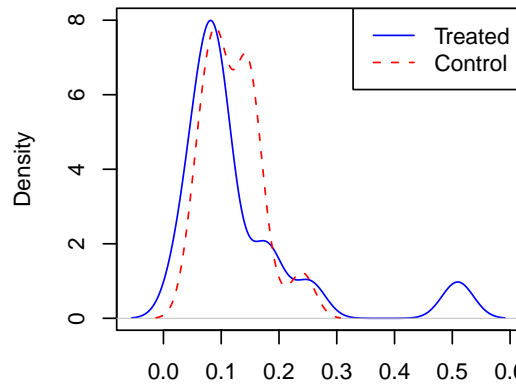
N = 15 Bandwidth = 0.03185

**Turnout '00
No Caliper**

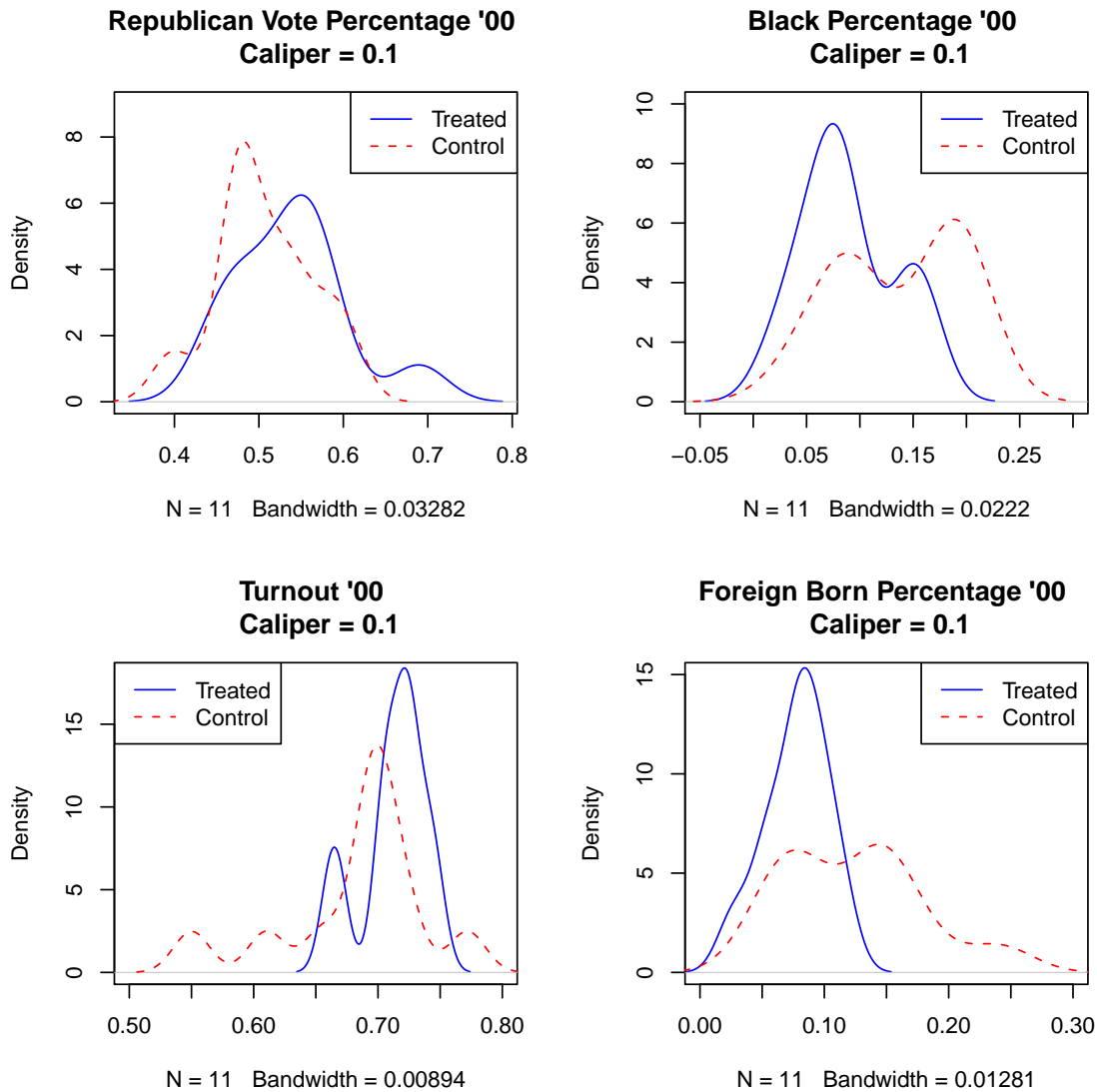


N = 15 Bandwidth = 0.01023

**Foreign Born Percentage '00
No Caliper**



N = 15 Bandwidth = 0.02727



We see that matching improves balance on some (although, not all) covariates. Enforcing the caliper improves the balance even more, dropping the unusual counties such as those with high percentages of foreign born people. Other pscors may do a better job of increasing balance. When looking for a pscore, our metric for determining what is a good pscore should be how much it improves balance.

- e. What is the difference between running your function with a caliper and without a caliper? Which method should we prefer? Why?

Running a matching method with a caliper means that only matches that are “close”, where “close” is defined as being within the caliper, are allowed. If there is no nearest neighbor within the caliper means that observation is dropped. This changes what you estimate, since it is no longer the average treatment effect of the treated if some treated observations are dropped. Whether or not we want to enforce a caliper depends on how much it changes the estimand and how bad the matches are without the enforced caliper. If we can drop a subset of the population that we can easily define, then we may wish to enforce the caliper. For example, if we knew that in this sample we were only dropping counties with major metropolitan centers, such as Broward and Miami-Dade counties, we could say the estimate is the average treatment effect for counties that are not major metropolitan centers. In general, though, we may wish to not enforce a caliper so long as the bias isn’t too bad due to imbalance.

- f. Using your matched data set, calculate a treatment effect (ATE, ATT, or ATC) of having an electronic voting machine. Which treatment effect did you choose to estimate? Why? How does it compare to your first three models?

```
att = mean(bush04[nn.pscore$index.tr] * nn.pscore$weights)
      - mean(bush04[nn.pscore$index.co] * nn.pscore$weights)
att
# 0.01305403
```

Here the “ATT” is estimated because there are so many more counties without the “eTouch” than there are with the “eTouch”. The estimate is of the opposite sign than the regression estimates.