MY457: Problem Set Template

40714

Wed/19/Feb

1 Concepts

1.1 1.1

$(Y_1, Y_0) \perp D|X$ means that the potential outcomes Y_1 and Y_0 are independent of D (our treatment) **conditional on** X. Here, X represents all the potential **pre-treatment covariates**.

However, this does not mean that $(Y_1, Y_0) \perp X$, because **pre-treatment covariates can influence the potential outcomes** for both the treatment and control groups.

1.2 1.2

The underlying **assumptions for conditional ignorability** are that the potential outcomes Y_1 and Y_0 are **independent of treatment assignment** D, given the pre-treatment covariates. This means that, after controlling for these covariates, treatment assignment can be considered as-if random.

For **common support**, the assumption is that **all units have a non-zero probability of being assigned to either the treatment or control group**. If any units have a probability of zero, this assumption is violated, as there would be no comparable treated or control units for them.

1.3 1.3

When we use **matching**, it is possible to use the same control unit for multiple treated units. However, this is **not ideal**, as it is unlikely that a single control unit will be a perfect match for multiple treated individuals—especially as the dimensionality of pre-treatment covariates increases.

Using one control unit for multiple treated units means that this control was the **closest match** available, but it does not necessarily indicate **how close they actually are**. This issue becomes particularly problematic in **high-dimensional settings**, where finding truly comparable units is difficult.

As a result, we may end up **comparing potential outcomes for units that are not sufficiently similar**, making it difficult to determine whether differences in outcomes are due to the treatment itself or some **other unobserved factors**.

2 Simulations

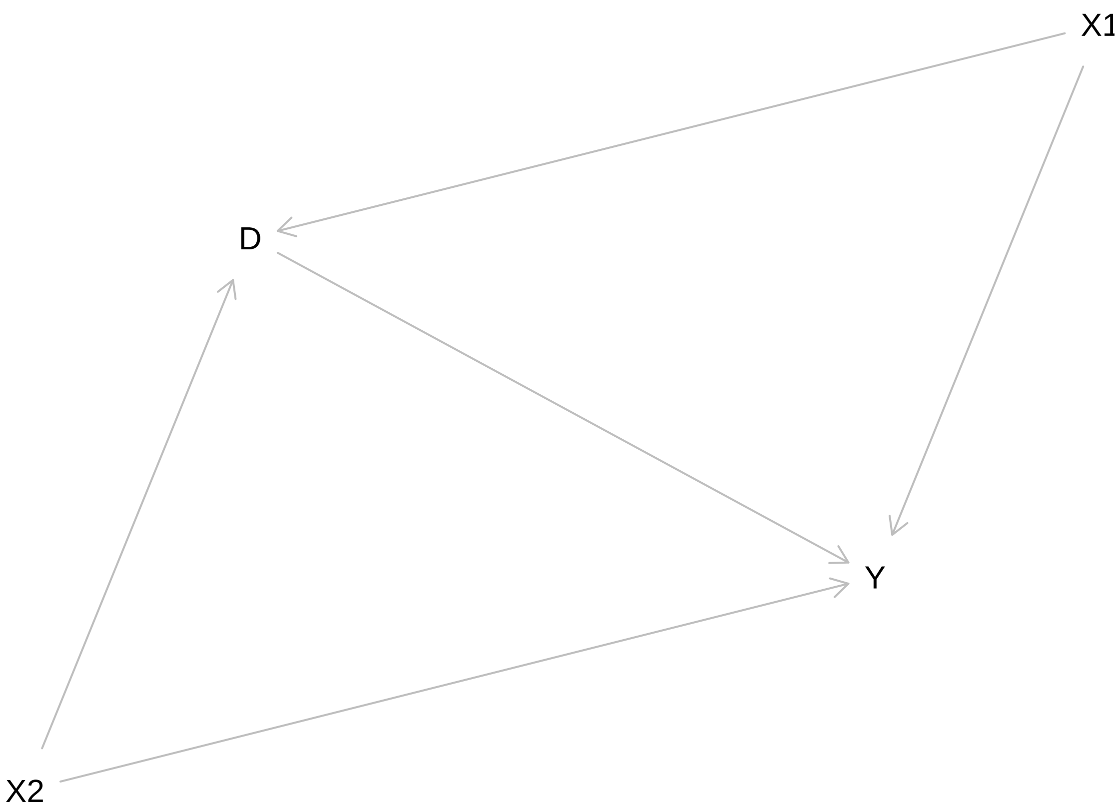
2.1 2.1

```
##   U1 U2         X1 X2       Y0       Y1 D        Y
## 1  0  0  -33.93682 31 51449.40 76449.40 0 51449.40
## 2  1  0 1542.34674 11 29126.12 54126.12 1 54126.12
## 3  0  1 1327.96530 24 45410.40 70410.40 0 45410.40
## 4  1  0 1701.86561 50 71204.07 96204.07 1 96204.07
## 5  1  0 1473.50274  6 26915.10 51915.10 0 26915.10
```

```
## 6  0  0  -71.43305 21 40129.27 65129.27 1 65129.27
```

```
library(dagitty)
dag <- dagitty('dag {
  D -> Y
  X1 -> D
  X2 -> D
  X1 -> Y
  X2 -> Y
}')

plot(dag)
```

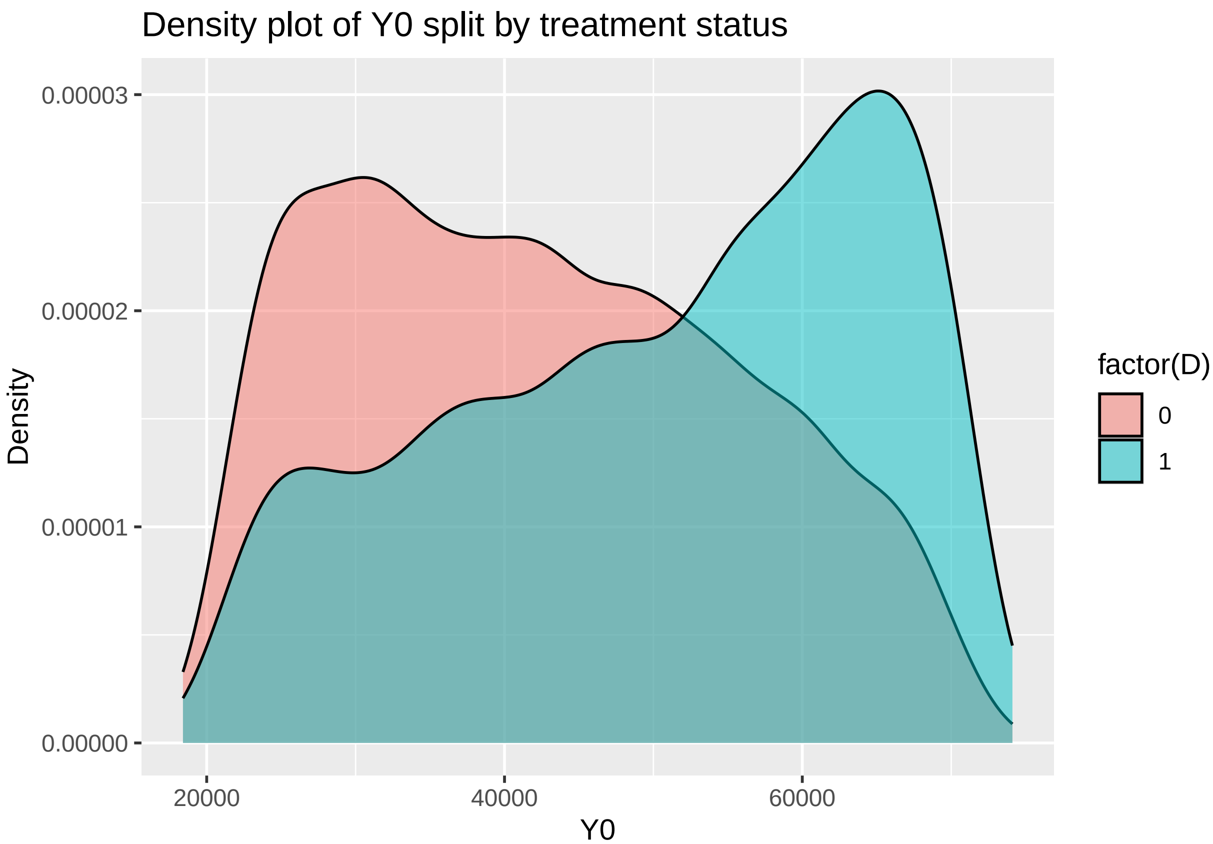


In the plot we can see that X1 and X2 are the pre treatment covarients and they effect both D and Y. D has a causal effect on Y but it's cofounded by X1 and X2.

2.2 2.2

```
library(ggplot2)

sim_data %>%
  ggplot(aes(x = Y0, fill = factor(D))) +
  geom_density(alpha = 0.5) +
  labs(title = "Density plot of Y0 split by treatment status",
       x = "Y0",
       y = "Density")
```



We cannot assume that $(Y_1, Y_0) \perp D$ because we introduced selection bias when we failed to randomize. So unless we condition on X (the pre treatment covarients) we cannot assume that $(Y_1, Y_0) \perp D$, only if we say that $((Y_1, Y_0) \perp D|X$ we can assume that the potential outcomes are independent from the treatment given the pre treatment covarients.

2.3 2.3

```
#install.packages("tableone")
library(tableone)

tableone <- CreateTableOne(vars = c("X1", "X2"), strata = "D", data = sim_data)
print(tableone, nonnormal = c("X1", "X2"))
```

```
##                      Stratified by D
##                       0                        1                          p
##   n                      5535                     4465
##   X1 (median [IQR]) 1443.17 [52.97, 1579.42] 1583.60 [1450.36, 2967.31] <0.001
##   X2 (median [IQR])   21.00 [10.00, 32.00]     33.00 [19.00, 43.00]     <0.001
##                      Stratified by D
##                       test
##   n
##   X1 (median [IQR]) nonnorm
##   X2 (median [IQR]) nonnorm
```

When we check the balance between X1 and X2 we can see that the means are not the same for the treatment and control groups as the p-value of < 0.001 shows significant difference between groups in both cases. This means that we have selection bias and we need to control for these covariates in order to estimate the true treatment effect. In other words some people have a higher chance of being treated then others. This is not a suprise given that we failed to truly randomize our assignment.

2.4 2.4

```
lm_naive <- lm(Y ~ D, data = sim_data)
#summary(lm_naive)
naive_ate <- coef(lm_naive)[2]

true_ate <- mean(sim_data$Y1 - sim_data$Y0)

cat("True ATE: ", true_ate, "\n")
```

```
## True ATE:  25000
```

```
cat("Naive ATE: ", naive_ate, "\n")
```

```
## Naive ATE:  33405.94
```

```
bias <- naive_ate - true_ate
cat("Selection Bias: ", bias, "\n")
```

```
## Selection Bias:  8405.941
```

Since we have simulated data we now both potential outcomes of a unit, hence we know the true ATE - which in this case is 25000. The naive ATE is 33405.94 which is higher than the true ATE. This is because we have selection bias in our data and we failed to control for the pre treatment covariates. The selection bias is 8405.94.

2.5 2.5

```
lm_x2 <- lm(Y ~ D + X2, data = sim_data)
x2_ate <- coef(lm_x2)[2]
cat("X2 ATE: ", x2_ate, "\n")
```

```
## X2 ATE:  25005.27
```

When we control for X2 we can see that the ATE is 25005.27 which is much closer to the true ATE of 25000. This is because we have controlled for the one of the known selection bias that was present in our data. This makes X2 a good control variable for our treatment effect.

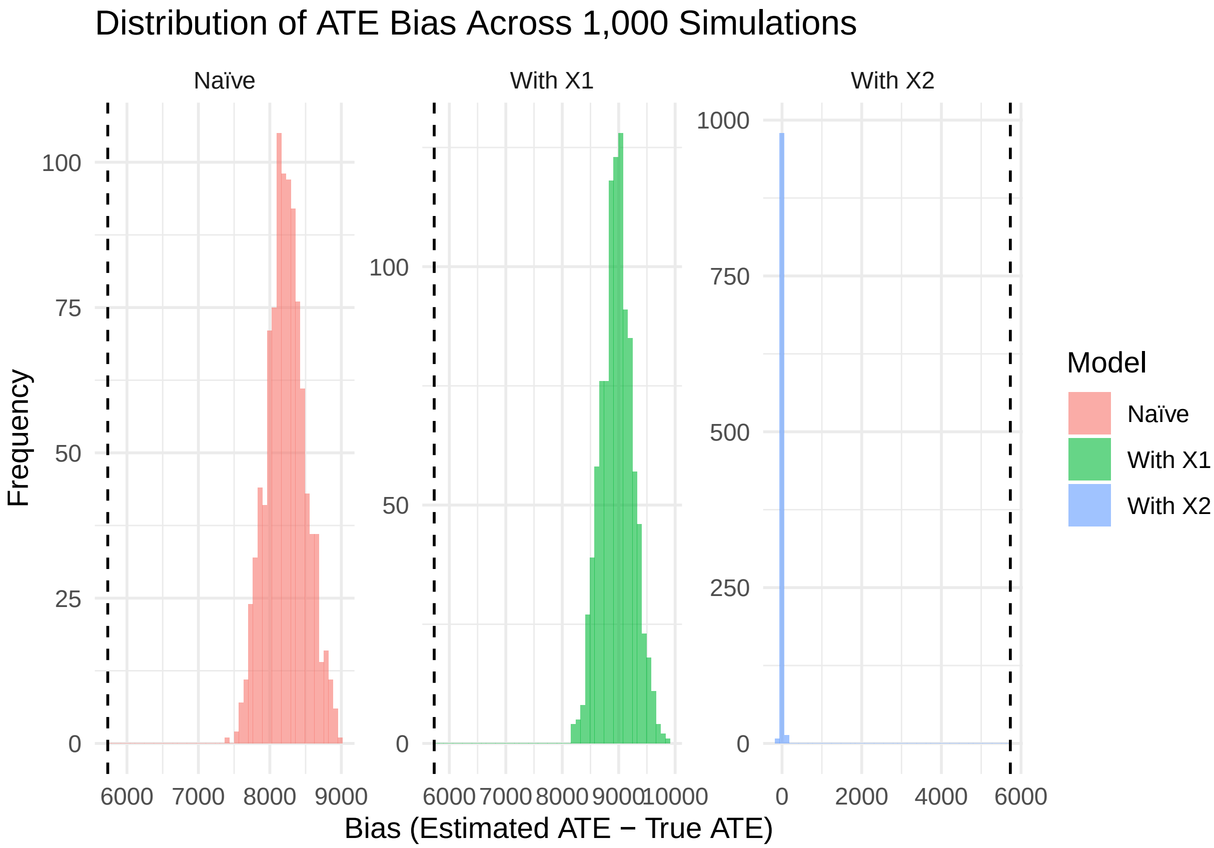
2.6 2.6

```
lm_x1 <- lm(Y ~ D + X1, data = sim_data)
x1_ate <- coef(lm_x1)[2]
cat("X1 ATE: ", x1_ate, "\n")
```

```
## X1 ATE:  34167.17
```

When we control for X1 we can see that the ATE is 34167.17 which is much further from the true ATE of 25000. This is because we have controlled for the one of the known selection bias that was present in our data. This makes X1 a bad control variable for our treatment effect.

2.7 2.7



After running the simulation 1000 times we can see that our results were not due to chance as both the naive and X1 models have a bias that is significantly different from 0 - with bias being around 8000 which in line what we saw above. The X2 model has a bias that is much closer to 0. This is because X2 is a good control variable for our treatment effect.

3 Replication

3.1 3.1

Urban and Niebler used the spliover adds from competitive states to not competitive states as their treatment. They argied that this way they can isolatethe effects of television adverts since if they would compare competitive states with noncompetitive states sthey could not control for speaches, rallys etc. I dont think it can be the ccase of selcetion on-observables since they are using a natural experiment to isolate the effect of the treatment and based on geogprahy certain parts of the neighboring states will be exposed to spillover adds. So probabal we would need to control for postcodes level variables (because they used postcodes to estimate donations) such as income edication, etc. (X) in order control fro pre treatment covariets.

3.2 3.2

```
##   Treated TotAds NonComp   zip TotalPop MedianHHInc PerCapitaHHInc MaleOver65
## 1       0      0       1 01001    16475       45735          22490       1234
## 2       0      0       1 01002    36776       42567          18212        999
## 3       1   5883       1 01005     5079       50395          20518        273
## 4       0      0       1 01007    12997       52425          21923        500
## 5       0      0       1 01008     1234       52663          23680         56
```

```
## 6       0      0       1 01010     3350       50181          23697         158
##   FemaleOver65 PercentOver65 Rural Urban PercentWhite PercentBlack
## 1         2133    0.20437026     0     1    0.9576328  0.010440061
## 2         1451    0.06661954     0     1    0.8022351  0.047694147
## 3          382    0.12896240     1     0    0.9730262  0.005316007
## 4          636    0.08740479     1     0    0.9618374  0.005924444
## 5           60    0.09400324     1     0    0.9870340  0.001620746
## 6          208    0.10925373     1     0    0.9611940  0.003582089
##   PercentHispanic amount AmountRep AmountDem AmountRCand AmountDCand
## 1     0.012018209   3300         0      3300           0        3300
## 2     0.059767239 202005     13000    189005        9300      185405
## 3     0.007284899  20140         0     20140           0       19790
## 4     0.019850735  17330       250     17080           0       16547
## 5     0.009724474      0         0         0           0           0
## 6     0.012537314   3104       200      2904         200        2904
##   AmountRComm AmountDComm rep dem    meanrep   meandem         _merge1
## 1           0           0   6   6 0.00000000 1.0000000     matched (3)
## 2        3700        3600 349 349 0.06876791 0.9312321     matched (3)
## 3           0         350  33  33 0.00000000 1.0000000     matched (3)
## 4         250         533  40  40 0.02500000 0.9750000     matched (3)
## 5           0           0   0   0 0.00000000 0.0000000 master only (1)
## 6           0           0   6   6 0.16666667 0.8333333     matched (3)
##   StCtyFIPS StDMACode     _merge2 DMACode               State StFIPS
## 1     25013     25543 matched (3)     543 Massachusetts           25
## 2     25011     25543 matched (3)     543 Massachusetts           25
## 3     25027     25506 matched (3)     506 Massachusetts           25
## 4     25015     25543 matched (3)     543 Massachusetts           25
## 5     25013     25543 matched (3)     543 Massachusetts           25
## 6     25013     25543 matched (3)     543 Massachusetts           25
##                 DMAName Keep_L10 Keep_15S Keep_25S     _merge3
## 1 Springfield-Holyoke           1        1        1 matched (3)
## 2 Springfield-Holyoke           1        1        1 matched (3)
## 3 Boston-Manchester             1        1        1 matched (3)
## 4 Springfield-Holyoke           1        1        1 matched (3)
## 5 Springfield-Holyoke           1        1        1 matched (3)
## 6 Springfield-Holyoke           1        1        1 matched (3)
##              Market TotDem TotRep  Negative  Positive  Personal    Policy
## 1   Springfield, MA      0      0        NA        NA        NA        NA
## 2   Springfield, MA      0      0        NA        NA        NA        NA
## 3 Boston/Manchester   3362   2521 0.7263301 0.2736699 0.2982322 0.7017678
## 4   Springfield, MA      0      0        NA        NA        NA        NA
## 5   Springfield, MA      0      0        NA        NA        NA        NA
## 6   Springfield, MA      0      0        NA        NA        NA        NA
##     MeanDem   MeanRep     _merge4 Bush_00_sum Total_00_sum Bush04_sum
## 1        NA        NA matched (3)    88939.57       274085     103298
## 2        NA        NA matched (3)    88939.57       274085     103298
## 3 0.5714771 0.4285229 matched (3)   710893.19      2158151     868481
## 4        NA        NA matched (3)    88939.57       274085     103298
## 5        NA        NA matched (3)    88939.57       274085     103298
## 6        NA        NA matched (3)    88939.57       274085     103298
##   Total04_sum repvote00 repvote04     diff PollsterEverComp PollsterSeptComp
## 1      298442  32.44963  34.61242 2.162789                1                0
## 2      298442  32.44963  34.61242 2.162789                1                0
## 3     2315777  32.93992  37.50279 4.562872                1                0
```

```
## 4      298442  32.44963  34.61242 2.162789                1                0
## 5      298442  32.44963  34.61242 2.162789                1                0
## 6      298442  32.44963  34.61242 2.162789                1                0
##   PollsterComp SeptNotComp    Pop    Cont RCont   DCont    Inc   Ads  AdsSqrd
## 1            0           1 16.475   3.300  0.00   3.300 45.735 0.000  0.00000
## 2            0           1 36.776 202.005 13.00 189.005 42.567 0.000  0.00000
## 3            0           1  5.079  20.140  0.00  20.140 50.395 5.883 34.60969
## 4            0           1 12.997  17.330  0.25  17.080 52.425 0.000  0.00000
## 5            0           1  1.234   0.000  0.00   0.000 52.663 0.000  0.00000
## 6            0           1  3.350   3.104  0.20   2.904 50.181 0.000  0.00000
##    RAds  DAds    RDAds RAdsSqrd DAdsSqrd RepubVote TotVote RepubShare RepubTown
## 1 0.000 0.000 0.000000 0.000000  0.00000    103298  298442  0.3461242         0
## 2 0.000 0.000 0.000000 0.000000  0.00000    103298  298442  0.3461242         0
## 3 2.521 3.362 8.475601 6.355441 11.30304    868481 2315777  0.3750279         0
## 4 0.000 0.000 0.000000 0.000000  0.00000    103298  298442  0.3461242         0
## 5 0.000 0.000 0.000000 0.000000  0.00000    103298  298442  0.3461242         0
## 6 0.000 0.000 0.000000 0.000000  0.00000    103298  298442  0.3461242         0
##   DemTown        CDF      CDF2    density Zip_Number Pop_Tot per_hsgrads
## 1       1 0.66958880 0.7459357 1350.62401       1001   12293    86.75669
## 2       1 0.98109317 0.6854535  640.50602       1002   14232    95.15177
## 3       1 0.86186254 0.8162311  119.96511       1005    3348    85.33453
## 4       1 0.84799320 0.8439698  242.47543       1007    8577    89.86825
## 5       1 0.09666023 0.8465866   23.69753       1008     845    89.11243
## 6       1 0.66311210 0.8126329   95.96050       1010    2252    85.65719
##   per_collegegrads          geoid geoid2 geodisplaylabel perinstate
## 1         22.06947 8600000US01001   1001     ZCTA5 01001       85.0
## 2         68.52867 8600000US01002   1002     ZCTA5 01002       96.6
## 3         20.51971 8600000US01005   1005     ZCTA5 01005       99.3
## 4         31.38627 8600000US01007   1007     ZCTA5 01007       97.2
## 5         25.79882 8600000US01008   1008     ZCTA5 01008       91.6
## 6         27.79751 8600000US01010   1010     ZCTA5 01010       96.7
##   peroutofstate MergeCommuting
## 1          15.0    matched (3)
## 2           3.4    matched (3)
## 3           0.7    matched (3)
## 4           2.8    matched (3)
## 5           8.4    matched (3)
## 6           3.3    matched (3)
```

After we load the data we can created the binary Treated variables for non-competitive postcodes that have received more than 1000 adds. We then reorganize the columns so that the variables that we care about - Treated, TotAds, NonComp - are the first columns. We then print the 5 five rows of the data to see if everything is in order.

```
# How many treated vs not treated
dollars_data %>%
  group_by(Treated) %>%
  summarise(n = n())
```

```
## # A tibble: 2 x 2
##   Treated     n
##     <dbl> <int>
## 1       0 24165
## 2       1  6406
```

We can see that we have more non-treated observations than treated observations around 4 times more non-treated observations than treated observations.

3.3 3.3

```
## # A tibble: 2 x 2
##   Treated mean_contribution
##     <dbl>             <dbl>
## 1       0              19.4
## 2       1              19.7
```

```r
# Estimate Naive ATE
lm_naive <- lm(Cont ~ Treated, data = dollars_data)
naive_ate <- coef(lm_naive)[2]
cat("Naive ATE: ", naive_ate, "\n")
```

```
## Naive ATE:  0.3120867
```

When we estimate the naive ATE we can see that the naive ATE is 0.313 which is the difference in the mean contribution between the treated and non-treated observations. This is not the true ATE since we have selection bias in our data and we need to control for the pre treatment covariates.

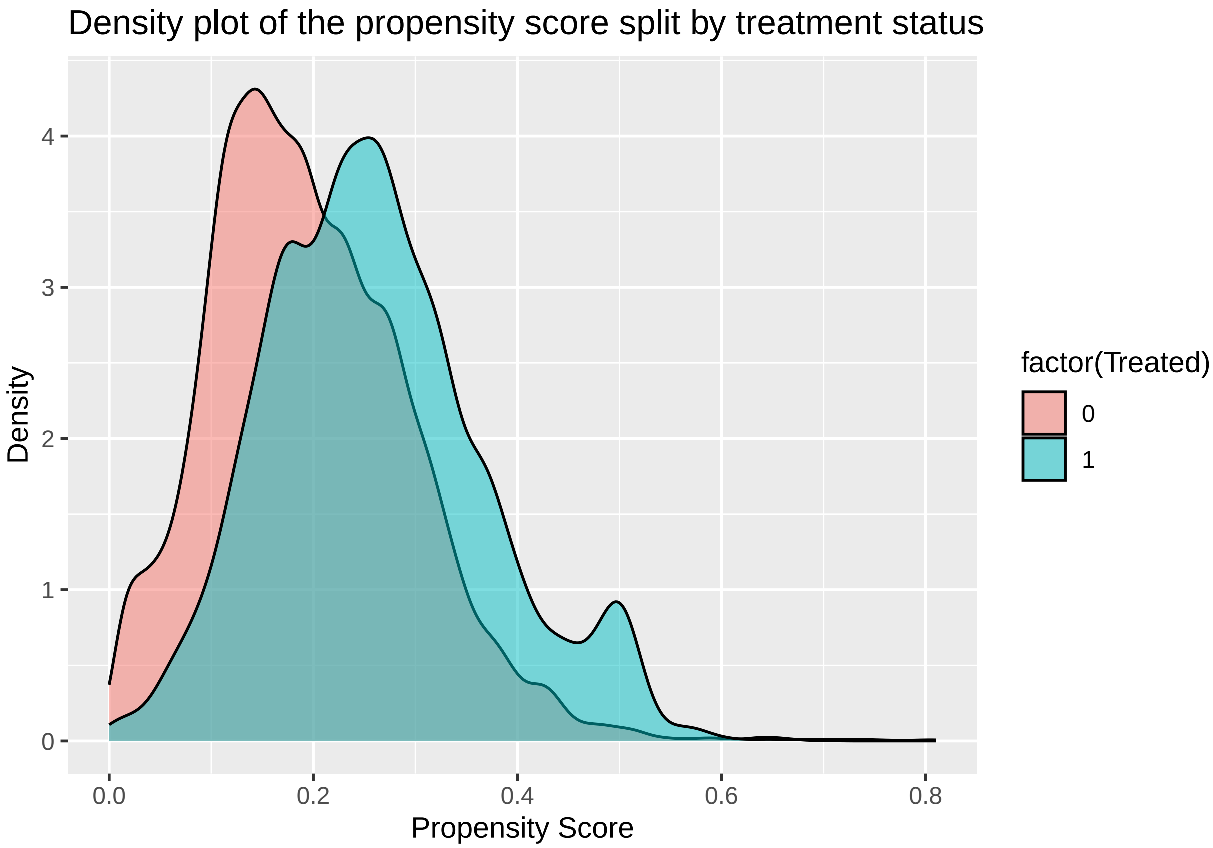
3.4 3.4

```
##      pscore Treated    Cont TotalPop MedianHHInc PercentOver65 PercentWhite
## 1 0.4391659       0   3.300    16475       45735    0.20437026    0.9576328
## 2 0.4309274       0 202.005    36776       42567    0.06661954    0.8022351
## 3 0.3469395       1  20.140     5079       50395    0.12896240    0.9730262
## 4 0.4695781       0  17.330    12997       52425    0.08740479    0.9618374
## 5 0.4799630       0   0.000     1234       52663    0.09400324    0.9870340
## 6 0.4708229       0   3.104     3350       50181    0.10925373    0.9611940
##   PercentBlack PercentHispanic Rural Urban repvote00 repvote04 RepubVote
## 1  0.010440061     0.012018209     0     1  32.44963  34.61242    103298
## 2  0.047694147     0.059767239     0     1  32.44963  34.61242    103298
## 3  0.005316007     0.007284899     1     0  32.93992  37.50279    868481
## 4  0.005924444     0.019850735     1     0  32.44963  34.61242    103298
## 5  0.001620746     0.009724474     1     0  32.44963  34.61242    103298
## 6  0.003582089     0.012537314     1     0  32.44963  34.61242    103298
##   RepubShare per_hsgrads per_collegegrads    density
## 1  0.3461242    86.75669         22.06947 1350.62401
## 2  0.3461242    95.15177         68.52867  640.50602
## 3  0.3750279    85.33453         20.51971  119.96511
## 4  0.3461242    89.86825         31.38627  242.47543
## 5  0.3461242    89.11243         25.79882   23.69753
## 6  0.3461242    85.65719         27.79751   95.96050
```

Here we manually selected the columns that we think will be the most useful for our model. These are Demographic columns such as TotalPop, ethinicity and Age, Political columns such as repvote00, repvote04, RepubVote, RepubShare and Socieconomic columns such as per_hsgrads, per_collegegrads. I would've like to use an algorithm to try to select column that contribute most to the model automatically instead selecting column manually but due to lack of time I had to do it manually. Also I dropped the columns that had missing values, which again is not ideal given that might be reason that some values are missing so ideally I would've used a package called MICE to impute the missing values - but had to drop them due to time constraints. Luckily only 50 ish rows had to be dropped out of more than 30,000 rows so it should not have a big impact on the results.

We then fit a logistic regression model to estimate the propensity score and add it to the dataset. We then reorganize the columns so that the propensity score is the first column. We then print the first 5 rows of the data to see if everything is in order.

3.5 3.5



After plotting the distribution of the propensity score for the treated and non-treated observations we can see that the propensity score is not well balanced between the treated and non-treated observations - aka we have observations that are treated but should have been in the control and vice versa. This is not ideal since if we want to do propensity score mathcing we need to make sure that the propensity score is well balanced between the treated and non-treated observations.

3.6 3.6

```
##
## Call:
## matchit(formula = Treated ~ pscore, data = selected_data, method = "nearest",
##     replace = TRUE)
##
## Summary of Balance for All Data:
##          Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance        0.2655        0.1947          0.5466     1.7390    0.1788
## pscore          0.2622        0.1956          0.6152     1.2546    0.1788
##          eCDF Max
## distance   0.2636
## pscore     0.2636
##
## Summary of Balance for Matched Data:
##          Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
```

```
## distance        0.2655        0.2655          0.0003     1.0027         0
## pscore          0.2622        0.2621          0.0005     1.0043         0
##          eCDF Max Std. Pair Dist.
## distance    0.002          0.0006
## pscore      0.002          0.0008
##
## Sample Sizes:
##               Control Treated
## All           24136.      6397
## Matched (ESS)  3734.75    6397
## Matched        4933.      6397
## Unmatched     19203.         0
## Discarded         0.         0
```

After performing the propensity score matching we can see that we ended up with only 11,000 observations (5000 control and 6000 treated) which is not ideal since we lost more than 1/3 of our data. This is because the propensity score was not well balanced between the treated and non-treated observations.

3.7 3.7

```
##                                  Stratified by Treated
##                                   0                      1
##   n                                   24136                  6397
##   TotalPop (mean (SD))              9280.11 (13106.79)    7635.83 (12014.15)
##   MedianHHInc (mean (SD))          39620.79 (16239.60)   41032.90 (16397.19)
##   PercentOver65 (mean (SD))            0.14 (0.07)           0.14 (0.07)
##   PercentWhite (mean (SD))             0.84 (0.21)           0.90 (0.18)
##   PercentBlack (mean (SD))             0.08 (0.17)           0.05 (0.14)
##   PercentHispanic (mean (SD))          0.07 (0.15)           0.03 (0.08)
##   Rural (mean (SD))                    0.60 (0.49)           0.65 (0.48)
##   Urban (mean (SD))                    0.40 (0.49)           0.35 (0.48)
##   repvote00 (mean (SD))               52.06 (9.09)          48.83 (9.84)
##   repvote04 (mean (SD))               55.05 (9.45)          51.12 (10.10)
##   RepubVote (mean (SD))           411140.74 (423095.39) 393487.50 (438955.54)
##   RepubShare (mean (SD))               0.55 (0.09)           0.51 (0.10)
##   per_hsgrads (mean (SD))             78.55 (12.34)         81.08 (10.98)
##   per_collegegrads (mean (SD))        18.06 (13.70)         19.28 (14.21)
##   density (mean (SD))               1173.30 (4434.49)      919.40 (3076.02)
##                                  Stratified by Treated
##                                   p      test
##   n
##   TotalPop (mean (SD))            <0.001
##   MedianHHInc (mean (SD))         <0.001
##   PercentOver65 (mean (SD))        0.193
##   PercentWhite (mean (SD))        <0.001
##   PercentBlack (mean (SD))        <0.001
##   PercentHispanic (mean (SD))     <0.001
##   Rural (mean (SD))               <0.001
##   Urban (mean (SD))               <0.001
##   repvote00 (mean (SD))           <0.001
##   repvote04 (mean (SD))           <0.001
##   RepubVote (mean (SD))            0.003
##   RepubShare (mean (SD))          <0.001
##   per_hsgrads (mean (SD))         <0.001
##   per_collegegrads (mean (SD))    <0.001
```

```
##   density (mean (SD))          <0.001
```

```
##                                Stratified by Treated
##                                 0                      1
##   n                                 4933                   6397
##   TotalPop (mean (SD))           8186.97 (11944.12)    7635.83 (12014.15)
##   MedianHHInc (mean (SD))       40757.37 (16176.27)   41032.90 (16397.19)
##   PercentOver65 (mean (SD))         0.14 (0.07)           0.14 (0.07)
##   PercentWhite (mean (SD))          0.89 (0.17)           0.90 (0.18)
##   PercentBlack (mean (SD))          0.06 (0.14)           0.05 (0.14)
##   PercentHispanic (mean (SD))       0.04 (0.09)           0.03 (0.08)
##   Rural (mean (SD))                 0.64 (0.48)           0.65 (0.48)
##   Urban (mean (SD))                 0.36 (0.48)           0.35 (0.48)
##   repvote00 (mean (SD))            49.51 (9.15)          48.83 (9.84)
##   repvote04 (mean (SD))            51.92 (9.43)          51.12 (10.10)
##   RepubVote (mean (SD))        415649.17 (395572.01) 393487.50 (438955.54)
##   RepubShare (mean (SD))            0.52 (0.09)           0.51 (0.10)
##   per_hsgrads (mean (SD))          80.43 (11.63)         81.08 (10.98)
##   per_collegegrads (mean (SD))     18.96 (14.44)         19.28 (14.21)
##   density (mean (SD))             929.93 (3098.92)      919.40 (3076.02)
##                                Stratified by Treated
##                                 p      test
##   n
##   TotalPop (mean (SD))           0.015
##   MedianHHInc (mean (SD))        0.372
##   PercentOver65 (mean (SD))      0.880
##   PercentWhite (mean (SD))       0.001
##   PercentBlack (mean (SD))       0.112
##   PercentHispanic (mean (SD))   <0.001
##   Rural (mean (SD))              0.114
##   Urban (mean (SD))              0.114
##   repvote00 (mean (SD))         <0.001
##   repvote04 (mean (SD))         <0.001
##   RepubVote (mean (SD))          0.005
##   RepubShare (mean (SD))        <0.001
##   per_hsgrads (mean (SD))        0.002
##   per_collegegrads (mean (SD))   0.248
##   density (mean (SD))            0.857
```

After comparing the balance in the pre-treatment covariates before and after matching we can see that the p-values of many covariates that were significant before are now not significant after matching. This is no good since we want to make sure that the pre-treatment covariates are well balanced between the treated and non-treated observations because if they are not well balanced we cannot assume that the potential outcomes are independent of the treatment given the pre-treatment covariates. In other words we introduced selection bias in our data when we matched and hence we cannot assume that the potential outcomes are independent of the treatment given the pre-treatment covariates.

3.8 3.8

```
## Estimated ATT (Difference-in-Means): 0.56
```

3.9 3.9

When evaluating the research design used in this study, several concerns arise regarding the validity of the causal inference drawn.

A key concern is whether the researcher identified the correct set of pre-treatment covariates (X) to effectively control for confounding factors. For instance, individuals living near state borders are more likely to be exposed to spillover campaign ads from the neighboring state. However, these individuals might work, shop, or socialize across the border, meaning their exposure to campaign activity is not limited to their state of residence. This cross-border exposure introduces unmeasured confounding, as the interactions and political messaging they encounter while in the neighboring state remain unaccounted for.

Another potential problem lies in the propensity score matching employed. The imbalance observed in the propensity scores indicates that after matching we most likely introduce selection bias when we match. This is problematic because the matching process should ensure that the treated and control groups are comparable on observed characteristics.

To address this, the researchers should have considered alternative approaches, such as:

Inverse Probability Weighting (IPW), Covariate Adjustment or Doubly Robust Estimation - Combining IPW and regression adjustment provides additional protection against model misspecification.

4 Code appendix

```r
# this chunk contains code that sets global options for the entire .Rmd.
# we use include=FALSE to suppress it from the top of the document, but it will still appear in the app
knitr::opts_chunk$set(echo=FALSE, warning=FALSE, message=FALSE, linewidth=60)

# you can include your libraries here:
library(tidyverse)

# and any other options in R:
options(scipen=999)

set.seed(123)
n_obs <- 10000
tau <- 25000
# Create the dataset
U1 <- rbinom(n_obs, 1, 0.5)
U2 <- rbinom(n_obs, 1, 0.5)
X2 <- sample(1:50, n_obs, replace = TRUE)
X1 <- 10 + 1500*U1 + 1500*U2 + rnorm(n_obs, mean = 0, sd = 100)
Y0 <- 20000 + 1000*X2 + 1500*U2 + rnorm(n_obs, mean = 0, sd = 1000)
Y1 <- Y0 + tau
prob_d <- pnorm(X2, mean = 50, sd = 25) + 0.5*U1
prob_d <- pmin(prob_d, 1)
D <- rbinom(n_obs, 1, prob_d)
sim_data <- data.frame(U1, U2, X1, X2, Y0, Y1, D)
sim_data$Y <- ifelse(sim_data$D== 1, Y1, Y0)

head(sim_data)
library(dagitty)
dag <- dagitty('dag {
  D -> Y
  X1 -> D
  X2 -> D
  X1 -> Y
  X2 -> Y
}')

plot(dag)
library(ggplot2)

sim_data %>%
  ggplot(aes(x = Y0, fill = factor(D))) +
  geom_density(alpha = 0.5) +
  labs(title = "Density plot of Y0 split by treatment status",
       x = "Y0",
       y = "Density")
#install.packages("tableone")
library(tableone)

tableone <- CreateTableOne(vars = c("X1", "X2"), strata = "D", data = sim_data)
print(tableone, nonnormal = c("X1", "X2"))
```

```r
lm_naive <- lm(Y ~ D, data = sim_data)
#summary(lm_naive)
naive_ate <- coef(lm_naive)[2]

true_ate <- mean(sim_data$Y1 - sim_data$Y0)

cat("True ATE: ", true_ate, "\n")
cat("Naive ATE: ", naive_ate, "\n")

bias <- naive_ate - true_ate
cat("Selection Bias: ", bias, "\n")

lm_x2 <- lm(Y ~ D + X2, data = sim_data)
x2_ate <- coef(lm_x2)[2]
cat("X2 ATE: ", x2_ate, "\n")

lm_x1 <- lm(Y ~ D + X1, data = sim_data)
x1_ate <- coef(lm_x1)[2]
cat("X1 ATE: ", x1_ate, "\n")
library(ggplot2)

# Define number of simulations
n_sim <- 1000
n_obs <- 10000
tau <- 25000

# Store biases for each model
bias_naive <- numeric(n_sim)
bias_x2 <- numeric(n_sim)
bias_x1 <- numeric(n_sim)

# Monte Carlo simulation loop
for (i in 1:n_sim) {

  # Generate fresh data
  U1 <- rbinom(n_obs, 1, 0.5)
  U2 <- rbinom(n_obs, 1, 0.5)
  X2 <- sample(1:50, n_obs, replace = TRUE)
  X1 <- 10 + 1500 * U1 + 1500 * U2 + rnorm(n_obs, mean = 0, sd = 100)
  Y0 <- 20000 + 1000 * X2 + 1500 * U2 + rnorm(n_obs, mean = 0, sd = 1000)
  Y1 <- Y0 + tau
  prob_d <- pnorm(X2, mean = 50, sd = 25) + 0.5 * U1
  prob_d <- pmin(prob_d, 1)
  D <- rbinom(n_obs, 1, prob_d)
  Y <- ifelse(D == 1, Y1, Y0)

  # Store data in a dataframe
  sim_data <- data.frame(U1, U2, X1, X2, Y0, Y1, D, Y)

  # Compute true ATE (from counterfactuals)
  true_ate <- mean(sim_data$Y1 - sim_data$Y0)

  # Estimate ATE under different models
```

```r
  naive_ate <- coef(lm(Y ~ D, data = sim_data))[2]
  x2_ate <- coef(lm(Y ~ D + X2, data = sim_data))[2]   # Adding X2 to base model
  x1_ate <- coef(lm(Y ~ D + X1, data = sim_data))[2]   # Adding X1 only (corrected for 2.6)

  # Store biases
  bias_naive[i] <- naive_ate - true_ate
  bias_x2[i] <- x2_ate - true_ate
  bias_x1[i] <- x1_ate - true_ate
}

# Create a data frame for visualization
bias_data <- data.frame(
  Bias = c(bias_naive, bias_x2, bias_x1),
  Model = rep(c("Naïve", "With X2", "With X1"), each = n_sim)
)

# Plot histograms of the biases
ggplot(bias_data, aes(x = Bias, fill = Model)) +
  geom_histogram(alpha = 0.6, position = "identity", bins = 50) +
  facet_wrap(~ Model, scales = "free") +
  geom_vline(aes(xintercept = mean(Bias)), color = "black", linetype = "dashed") +
  labs(
    title = "Distribution of ATE Bias Across 1,000 Simulations",
    x = "Bias (Estimated ATE - True ATE)",
    y = "Frequency"
  ) +
  theme_minimal()
library(readstata13)

# Load the dta data
dollars_data <- read.dta13("dollars_on_the_sidewalk.dta")

# define binary Treated variable for TotAds > 1000 and in Noncomp = 1 states
dollars_data$Treated <- ifelse(dollars_data$TotAds > 1000 & dollars_data$NonComp == 1, 1, 0)

#colnames(dollars_data)

# put Noncomp and Treated in the first columns
dollars_data <- dollars_data %>%
  select(Treated, TotAds, NonComp, everything())

head(dollars_data)
# How many treated vs not treated
dollars_data %>%
  group_by(Treated) %>%
  summarise(n = n())
# Estimate the mean contribution for each level of the Treated variable
dollars_data %>%
  group_by(Treated) %>%
  summarise(mean_contribution = mean(Cont, na.rm = TRUE))
# Estimate Naive ATE
lm_naive <- lm(Cont ~ Treated, data = dollars_data)
naive_ate <- coef(lm_naive)[2]
```

```r
cat("Naive ATE: ", naive_ate, "\n")
# Select desired columns and remove missing values before logistic regression
selected_data <- dollars_data %>%
  select(Treated, Cont, TotalPop, MedianHHInc, PercentOver65, PercentWhite, PercentBlack,
         PercentHispanic, Rural, Urban, repvote00, repvote04, RepubVote, RepubShare,
         per_hsgrads, per_collegegrads, density) %>%
  na.omit()

# Fit a logistic regression for the propensity score
ps_model <- glm(Treated ~ TotalPop + MedianHHInc + PercentOver65 + PercentWhite +
                  PercentBlack + PercentHispanic + Rural + Urban + repvote00 + repvote04 +
                  RepubVote + RepubShare + per_hsgrads + per_collegegrads + density,
                data = selected_data, family = binomial(link = "logit"))

# Add the propensity score to the dataset
selected_data$pscore <- predict(ps_model, type = "response")

# Reorganize the columns so pscor is the first column
selected_data <- selected_data %>%
  select(pscore, everything())

selected_data %>%
  head()
# Plot the distribution of the propensity score for the treated and non-treated observations
selected_data %>%
  ggplot(aes(x = pscore, fill = factor(Treated))) +
  geom_density(alpha = 0.5) +
  labs(title = "Density plot of the propensity score split by treatment status",
       x = "Propensity Score",
       y = "Density")
# Perform propensity score matching
library(MatchIt)

# Create a MatchIt object
match_model <- matchit(Treated ~ pscore, data = selected_data, method = "nearest", replace = TRUE)

# Summary of the matching
summary(match_model)

# Match the data
matched_data <- match.data(match_model)
# Check balance in pre-treatment covariates before matching
print(tableone_before <- CreateTableOne(vars = c("TotalPop", "MedianHHInc", "PercentOver65", "PercentWh
                                          "PercentBlack", "PercentHispanic", "Rural", "Urban",
                                          "repvote00", "repvote04", "RepubVote", "RepubShare",
                                          "per_hsgrads", "per_collegegrads", "density"),
                                 strata = "Treated", data = selected_data))

# Check balance in pre-treatment covariates after matching
print(tableone_after <- CreateTableOne(vars = c("TotalPop", "MedianHHInc", "PercentOver65", "PercentWhi
                                         "PercentBlack", "PercentHispanic", "Rural", "Urban",
                                         "repvote00", "repvote04", "RepubVote", "RepubShare",
                                         "per_hsgrads", "per_collegegrads", "density"),
```

```r
                                  strata = "Treated", data = matched_data))


# Subset the treated group and their matched controls
treated_units <- matched_data %>% filter(Treated == 1)
control_units <- matched_data %>% filter(Treated == 0)

# Calculate the mean outcome for each group
mean_treated <- mean(treated_units$Cont, na.rm = TRUE)
mean_control <- mean(control_units$Cont, na.rm = TRUE)

# Estimate ATT as the difference-in-means
att_estimate <- mean_treated - mean_control

# print
cat("Estimated ATT (Difference-in-Means):", round(att_estimate, 2), "\n")
# this chunk generates the complete code appendix.
# eval=FALSE tells R not to re-run (``evaluate'') the code here.
```