Causal Inference: Introduction to Propensity Score Matching

Joseph Kush

12/3/2021

Purpose: This R script introduces propensity score matching methods for causal inference. Example school-level data is then generated, in which there is imbalance on baseline covariates between schools implementing treatment versus control. After examining the extent of imbalance, propensity scores are estimated and schools are matched. Finally, the effect of treatment on the outcome of interest (read) is estimated and compared to the known population ATT.

Data generation First, load necessary libraries and set seed for reproducibility.

```
library("cobalt")
```

```
##  cobalt (Version 4.3.1, Build Date: 2021-03-30 09:50:18 UTC)
```

```
library("MatchIt")
```

```
##
## Attaching package: ’MatchIt’
```

```
## The following object is masked from ’package:cobalt’:
##
##     lalonde
```

```
library("survey")
```

```
## Loading required package: grid
```

```
## Loading required package: Matrix
```

```
## Loading required package: survival
```

```
##
## Attaching package: ’survey’
```

```
## The following object is masked from ’package:graphics’:
##
##     dotchart
```

```r
set.seed(1234)
```

In this example, we generate school-level covariates such as enrollment, student:teacher ratio, and % of students receiving free and reduced-price meals for N = 1,000 schools.

```r
# Generate school-level data
sample_size <- 1000
schID <- 1:sample_size

enrollment <- as.integer(runif(n = sample_size, min = 100, max = 1000))
st_ratio <- rnorm(n = sample_size, mean = 0.2, sd = 0.05)
susp <- runif(n = sample_size, min = 0, max = 0.2)
farms <- rnorm(n = sample_size, mean = st_ratio, sd = 0.03)
sped <- rnorm(n = sample_size, mean = 0.15, sd = 0.03)
minority <- runif(n = sample_size, min = 0, max = 0.8)
ell <- rnorm(n = sample_size, mean = farms, sd = 0.02)
disable <- rnorm(n = sample_size, mean = 0.15, sd = 0.03)
math <- rnorm(n = sample_size, mean = 0.8, sd = 0.05)
```

Next, a binary treatment indicator is generated for each school according to a treatment-selection model. Logit odds (x) are created with a binary treatment indicator generated from a binomial distribution with a probability of exposure equal to $\frac{e^x}{1+e^x}$.

```r
# Create a binary treatment indicator
# First, create logit odds (hint: intercept sets treatment prevalence)
logit_treat <- 0.7 + 0*enrollment + 2.7*st_ratio + 1.9*susp + 1.3*farms +
  9*sped*minority + 1.8*ell + 2.6*disable + -3.7*math

# Next, convert logit odds into probability
prob_treat <- exp(logit_treat)/(1 + exp(logit_treat))

# Finally, generate binary treatment indicator from binomial
# distribution of 1 trial with P = prob_treat
treat <- rbinom(sample_size, 1, prob_treat)

# Create potential outcomes for reading scores
treatment_effect <- 2

read_0 <- (80 + 0*treatment_effect +
  0*enrollment + 2.7*st_ratio + 1.9*susp + 1.3*farms +
  9*sped*minority + 1.8*ell + 2.6*disable + -3.7*math +
  rnorm(n = sample_size, mean = 0, sd = 3)) / 100

read_1 <- (80 + 1*treatment_effect +
  0*enrollment + 2.7*st_ratio + 1.9*susp + 1.3*farms +
  9*sped*minority + 1.8*ell + 2.6*disable + -3.7*math +
  rnorm(n = sample_size, mean = 0, sd = 3)) / 100

read <- ifelse(treat == 1, read_1, read_0)
```

Everything is placed into a data.frame.

```
# Combine all variables into a data frame
df <- data.frame(cbind(schID, enrollment, st_ratio, susp, farms, sped,
                       minority, ell, disable, math, treat, read))
summary(df)
```

```
##      schID          enrollment       st_ratio            susp          
##  Min.   :   1.0   Min.   :100.0   Min.   :0.05348   Min.   :0.0001799  
##  1st Qu.: 250.8   1st Qu.:332.0   1st Qu.:0.16839   1st Qu.:0.0500234  
##  Median : 500.5   Median :558.5   Median :0.20025   Median :0.0955312  
##  Mean   : 500.5   Mean   :556.1   Mean   :0.19942   Mean   :0.0991982  
##  3rd Qu.: 750.2   3rd Qu.:782.2   3rd Qu.:0.23108   3rd Qu.:0.1480819  
##  Max.   :1000.0   Max.   :999.0   Max.   :0.33529   Max.   :0.1998465  
##      farms              sped            minority              ell         
##  Min.   :0.02307   Min.   :0.05871   Min.   :0.0008865   Min.   :0.01205  
##  1st Qu.:0.16034   1st Qu.:0.12925   1st Qu.:0.2011978   1st Qu.:0.16036  
##  Median :0.20098   Median :0.15013   Median :0.3978692   Median :0.19845  
##  Mean   :0.20028   Mean   :0.14971   Mean   :0.4009244   Mean   :0.19980  
##  3rd Qu.:0.24007   3rd Qu.:0.16981   3rd Qu.:0.5997060   3rd Qu.:0.24207  
##  Max.   :0.36934   Max.   :0.24360   Max.   :0.7969689   Max.   :0.37539  
##     disable             math            treat            read       
##  Min.   :0.06121   Min.   :0.6400   Min.   :0.000   Min.   :0.7018  
##  1st Qu.:0.13091   1st Qu.:0.7672   1st Qu.:0.000   1st Qu.:0.7826  
##  Median :0.14946   Median :0.8021   Median :1.000   Median :0.8036  
##  Mean   :0.15003   Mean   :0.8022   Mean   :0.513   Mean   :0.8036  
##  3rd Qu.:0.16990   3rd Qu.:0.8365   3rd Qu.:1.000   3rd Qu.:0.8252  
##  Max.   :0.24885   Max.   :0.9809   Max.   :1.000   Max.   :0.8998  
```

```
head(df)
```

```
##   schID enrollment  st_ratio       susp     farms      sped  minority       ell
## 1     1        202 0.2492390 0.19271987 0.2200244 0.1384478 0.1342445 0.1924882
## 2     2        660 0.1387631 0.04135248 0.1357742 0.1654217 0.3716408 0.1177858
## 3     3        648 0.2354863 0.01723949 0.2321643 0.1592401 0.1663240 0.2630359
## 4     4        661 0.1945390 0.04320560 0.2303048 0.2051746 0.6150317 0.1939220
## 5     5        874 0.2891304 0.04792931 0.2394538 0.1978021 0.3383099 0.2227598
## 6     6        676 0.1878278 0.03943219 0.1564585 0.1392541 0.6416494 0.1821018
##     disable      math treat      read
## 1 0.1362781 0.8050001     0 0.7831675
## 2 0.1479290 0.7342882     0 0.7797560
## 3 0.1410184 0.8387209     0 0.7617098
## 4 0.1362928 0.7748310     1 0.7868684
## 5 0.1708369 0.8553123     1 0.7726073
## 6 0.1542474 0.7170435     1 0.7996595
```

```
# ------------------------------------------------------------------------ #
```

Examine baseline imbalance and conduct matching We start by fitting a linear model to our data. Because method = NULL, no propensity score matching is done (i.e., baseline model)

```
# Because method = NULL, no propensity score matching
# is done (i.e., baseline model)
baseline <- matchit(treat ~ enrollment + st_ratio + susp + farms +
               sped + minority + ell + disable + math,
               data = df, method = NULL)
```

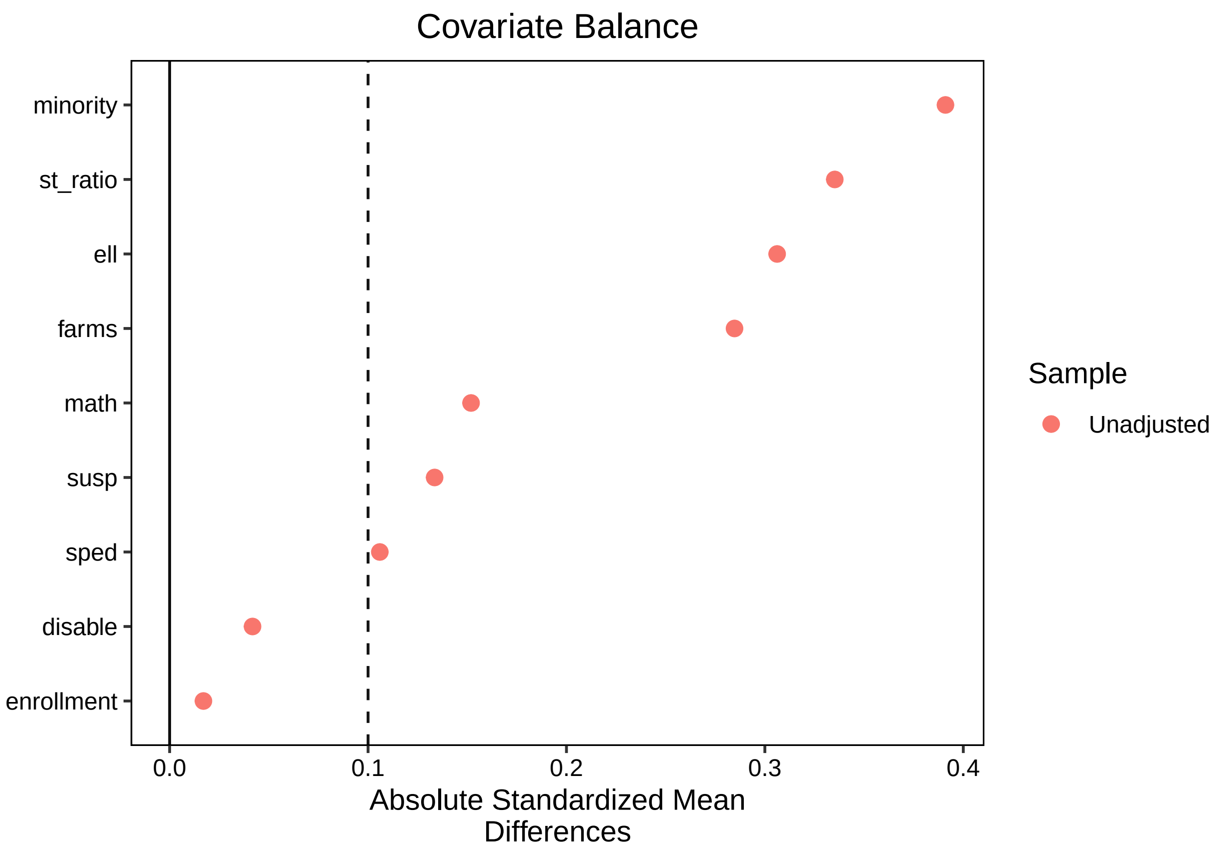
Let's examine standardized mean differences in covariates

```
bal.tab(baseline, s.d.denom = "treat", m.threshold = 0.1)
```

```
## Call
##  matchit(formula = treat ~ enrollment + st_ratio + susp + farms +
##     sped + minority + ell + disable + math, data = df, method = NULL)
##
## Balance Measures
##                 Type Diff.Un     M.Threshold.Un
## distance   Distance  0.6102
## enrollment  Contin.  0.0169     Balanced, <0.1
## st_ratio    Contin.  0.3444 Not Balanced, >0.1
## susp        Contin.  0.1335 Not Balanced, >0.1
## farms       Contin.  0.2918 Not Balanced, >0.1
## sped        Contin.  0.1065 Not Balanced, >0.1
## minority    Contin.  0.3799 Not Balanced, >0.1
## ell         Contin.  0.3122 Not Balanced, >0.1
## disable     Contin.  0.0416     Balanced, <0.1
## math        Contin. -0.1522 Not Balanced, >0.1
##
## Balance tally for mean differences
##                    count
## Balanced, <0.1         2
## Not Balanced, >0.1     7
##
## Variable with the greatest mean difference
##  Variable Diff.Un     M.Threshold.Un
##  minority  0.3799 Not Balanced, >0.1
##
## Sample sizes
##     Control Treated
## All     487     513
```

We can also plot our values.

```
# Plot SMDs
love.plot(baseline, s.d.denom = "pooled", abs = T,
          thresholds = 0.1, var.order = "unadjusted", drop.distance = T)
```



Now we will estimate propensity scores and match accordingly.

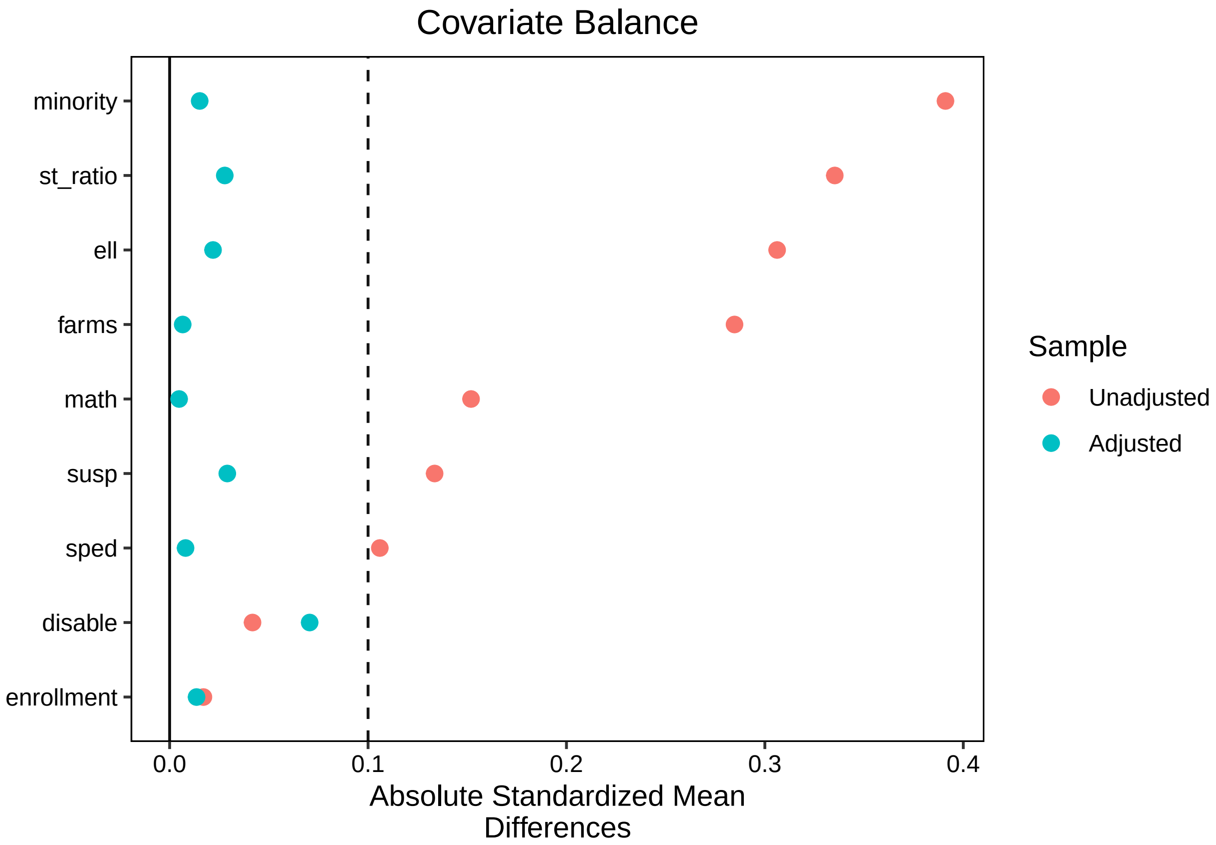
```
# Conduct propensity score matching
m <- matchit(treat ~ enrollment + st_ratio + susp + farms +
               sped + minority + ell + disable + math,
             data = df, method = "nearest", replace = T,
             ratio = 2, caliper = 0.1, std.caliper = T)
```

Examine our output: how do things look after matching?

```
summary(m)
```

```
##
## Call:
## matchit(formula = treat ~ enrollment + st_ratio + susp + farms +
##     sped + minority + ell + disable + math, data = df, method = "nearest",
##     replace = T, caliper = 0.1, std.caliper = T, ratio = 2)
##
## Summary of Balance for All Data:
##            Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance          0.5526        0.4713          0.6102     0.9129    0.1650
## enrollment      558.2300      553.7639          0.0169     1.0231    0.0116
## st_ratio          0.2071        0.1914          0.3444     0.8998    0.0953
## susp              0.1030        0.0952          0.1335     1.0004    0.0391
## farms             0.2082        0.1919          0.2918     0.9079    0.0777
## sped              0.1512        0.1481          0.1065     0.9806    0.0316
## minority          0.4435        0.3561          0.3799     1.1258    0.1100
## ell               0.2087        0.1904          0.3122     0.9262    0.0887
## disable           0.1506        0.1494          0.0416     1.0201    0.0117
## math              0.7984        0.8062         -0.1522     0.9928    0.0339
##            eCDF Max
## distance     0.2767
## enrollment   0.0360
## st_ratio     0.1515
## susp         0.0760
## farms        0.1323
## sped         0.0829
## minority     0.2041
## ell          0.1540
## disable      0.0369
## math         0.0546
##
##
## Summary of Balance for Matched Data:
##            Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance          0.5505        0.5505         -0.0002     0.9985    0.0018
## enrollment      556.9470      553.3900          0.0135     1.0143    0.0111
## st_ratio          0.2068        0.2081         -0.0286     1.0133    0.0188
## susp              0.1029        0.1012          0.0291     0.9588    0.0220
## farms             0.2079        0.2083         -0.0067     0.9414    0.0126
## sped              0.1510        0.1507          0.0081     0.9151    0.0179
## minority          0.4414        0.4381          0.0147     1.1498    0.0277
## ell               0.2083        0.2096         -0.0223     0.9289    0.0117
## disable           0.1507        0.1486          0.0702     0.9673    0.0194
## math              0.7990        0.7992         -0.0048     0.9752    0.0205
##            eCDF Max Std. Pair Dist.
## distance     0.0118          0.0080
## enrollment   0.0422          1.1444
## st_ratio     0.0639          0.9364
## susp         0.0658          1.1200
## farms        0.0521          1.0198
## sped         0.0501          1.1161
## minority     0.0668          0.7967
## ell          0.0354          0.9958
## disable      0.0570          1.1115
## math         0.0521          1.1190
##
## Percent Balance Improvement:
##            Std. Mean Diff. Var. Ratio eCDF Mean eCDF Max
## distance             100.0       98.4      98.9     95.7
## enrollment            20.4       38.0       3.6    -17.5
## st_ratio              91.7       87.5      80.2     57.9
## susp                  78.2   -11201.2      43.8     13.4
## farms                 97.7       37.5      83.8     60.7
## sped                  92.4     -353.1      43.2     39.6
## minority              96.1      -17.8      74.8     67.3
## ell                   92.9        3.8      86.8     77.0
## disable              -68.9      -67.1      -65.7     -54.4
## math                  96.9     -244.8       39.5       4.6
##
## Sample Sizes:
##               Control Treated
## All            487.       513
## Matched (ESS)  239.89     509
## Matched        363.       509
## Unmatched      124.         4
## Discarded        0.         0
```

```
love.plot(m, s.d.denom = "pooled", abs = T, thresholds = 0.1,
          var.order = "unadjusted", drop.distance = T)
```



The standardized mean differences decreased for ever covariate except for 'disable' (although SMD < 0.1).

Estimate outcomes Great - it looks like we have achieved balance on our covariates. Now let's estimate the treatment effect on reading outcomes. We begin by retaining matched units only and including weights (as we matched with replacement).

```
match_data <- match.data(m)
```

```
# Then use weights, as matching with replacement
mwr_data <- svydesign(ids=~1, weights =~ weights, data = match_data)
```

Finally, we estimate the ATT (unadjusted such that treatment is the only predictor).

```
outcome_unadj <- svyglm(read ~ treat, mwr_data, family=gaussian())
```

Before we examine our estimates, the potential outcomes ATT = 0.0188. This can be interpretted as the true effect of treatment (as we generated the data and know both potential outcomes)

```
pop_ATT <- by(read_1, treat, mean)[[2]] - by(read_0, treat, mean)[[2]]
pop_ATT
```

```
## [1] 0.01888618
```

Now we can examine bias in our propensity score matched estimate of the treatment effect.

```
# Matching estimates
round(summary(outcome_unadj)$coef, digits = 4)
```

```
##             Estimate Std. Error  t value Pr(>|t|)
## (Intercept)   0.7954     0.0019 420.1098        0
## treat         0.0177     0.0023   7.6145        0
```

```
bias_outcome_unadj <- (coef(outcome_unadj)["treat"] - pop_ATT) / pop_ATT
bias_outcome_unadj
```

```
##       treat
## -0.06115827
```

Not bad, about 6.1% bias. Out of curisoity, what would our estimated treatment effect be had we not used propensity score matching, but fit a regression model in which we adjusted for covariates including treatment?

```
# Compare to estimates from basic regression adjustment
reg_adj <- lm(read ~ treat + enrollment + st_ratio + susp + farms +
                sped + minority + ell + disable + math, data = df)
round(summary(reg_adj)$coef, digits = 4)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.8030     0.0173 46.3652   0.0000
## treat         0.0167     0.0020  8.4388   0.0000
## enrollment    0.0000     0.0000  0.5974   0.5504
## st_ratio     -0.0153     0.0383 -0.4007   0.6887
## susp          0.0144     0.0165  0.8710   0.3840
## farms         0.0873     0.0581  1.5016   0.1335
## sped          0.0658     0.0319  2.0608   0.0396
## minority      0.0174     0.0043  4.0874   0.0000
## ell          -0.0240     0.0487 -0.4933   0.6219
## disable       0.0630     0.0328  1.9182   0.0554
## math         -0.0580     0.0188 -3.0860   0.0021
```

```
bias_reg_adj <- (coef(reg_adj)["treat"] - pop_ATT) / pop_ATT
bias_reg_adj
```

```
##      treat
## -0.1146654
```

```
# -------------------------------------------------------------------------- #
```

While our PSM estimate is 6.1% biased, the regression adjustment estimate is 11.5% biased.