ZhaoY_Q1

Yongchao Zhao

December 12, 2018

Q1: Propensity Scores

All worlds should have 5 covariates, one binary treatment variable, and potential outcomes for one variable. The covariates should have a dependence structure. The response surface for World A can be linear and should satisfy all other assumptions as well. At least one of the response surfaces (that is, E[Y(0) | X] or E[Y(1) | X] in World B should be non-linear (with R2 less than .75) but the other assumptions should hold. World C can use the same response surface as World B but should violate one of the other assumptions. You should estimate a causal effect in each world both with linear regression and either propensity score matching or IPTW. You must present and discuss balance diagnostics and overlap plots for each of the three worlds.

(1) Description of a Hypothetical Real Life Scenario

The real life example we are going to use to simulate data is the Accelerated Study in Associate Programs (ASAP) of City University of New York (CUNY). The ASAP program aims to assist students in earning associate degrees within three years by providing a range of financial, academic, and personal supports including comprehensive and personalized advisement, career counseling, tutoring, waivers for tuition and mandatory fees, MTA MetroCards, and additional financial assistance to defray the cost of textbooks. The program takes first-time freshmen and continuing/transfer students (for continuing or transfer students, no more than 15 credits and a minimum GPA of 2.0), but to simplify the data simulation, we only include first-time fulltime (FTF) associate degree seeking students of ASAP program at one of the CUNY colleges in our current study.

The response variable is the cumulative GPA on graduation, and the five covariate variables are: gender, age, financial status (low-income or not), college admission average, and remediation needs of reading, writing or math.

The research question of interest focuses on the effect of the ASAP program on cumulative GPA on graduation for students who participated in it. The data for the comparison sample of students was pulled from the CUNY central data warehouse -IRDB - during a similar period of time.

(2) Data Generating Process

The simulated data will be generated by following requirements:
In total, 1000 observations will be generated.
(1) Treatment: 200 ASAP students in treatment group, 800 students in control group;

(2) Covariates:

- **gender**: 1 - male, 0 - female; based on historical data, the ratio of male students to female students for FTF student at the college is about 2:1;
- **age**: for treatment group, *age ~ N(19, 4)*, for control group, *age ~ N(23, 9)*;
- **pell**: indicates a student's financial status, eligible for Federal Pell grant means low-income household, 1- eligible, 0 - not eligible. Assume 90% of students in the treatment group are pell eligible while 60% are eligible in the control group;
- **caa**: college admission average, the score ranges from 50 to 100. For treatment group, *caa ~ N(65, 25)*, for control group, *caa ~ N(78, 49)*;
- **rem**: remediation needs in reading, writing or math. The value ranges from 0 (no remediation needs) to 3 (remediation needs for all three). If a student's caa score is greater than 85, set his/her remedial needs as 0, otherwise, randomly assign a value from 0 to 3.

(3) Response variable generating.

- For world A, generate potential outcomes of cumulative GPA assuming linear models for both E[Y(0) | X] and E[Y(1) | X]. The GPA scores range from 0 to 4. The expected treatment effect for everyone should be 0.5. The residual standard deviation of each potential outcome should be 0.2. Save two datasets as fullA (covariates and both potential outcomes) and obsA (covariates, treatment, and the observed outcome).
- For world B, generate potential outcomes of cumulative GPA with at least one of the response surfaces is non-linear (i.e., quadratic or square root transformation of the covariates). The GPA scores range from 0 to 4. The expected treatment effect for everyone should be 0.5. The residual standard deviation of each potential outcome should be 0.2. Save two datasets as fullB and obsB.
- For world C, the ignorability assumption does not hold. Assume we failed to control another important confounding variable - ethnicity. Further, generate potential outcomes as world B described, save two datasets as fullC and obsC.
  - **eth**: ethnicity variable, 1 - minority, 0 - non-minority. Assume 90% of students in the treatment group are minority while 75% are minority in the control group.

(3) R Code for Data Generating

```r
library(dplyr)
library(arm)

# data generating
set.seed(1)
treatment <- c(rep(1, 200),rep(0, 800))
gender <- sample(c(1,0), 1000, replace = T, prob = c(0.67, 0.33))
age <- c(rnorm(200, mean = 19, sd = 2), rnorm(800, mean = 23, sd = 3))
caa <- c(rnorm(200, mean = 65, sd = 5), rnorm(800, mean = 78, sd = 7))
pell <-  c(sample(c(1, 0), 200, replace = T, prob = c(0.9, 0.1)),
          sample(c(1, 0), 800, replace = T, prob = c(0.6, 0.4)))
eth <- c(sample(c(1, 0), 200, replace = T, prob = c(0.9, 0.1)),
          sample(c(1, 0), 800, replace = T, prob = c(0.75, 0.25)))
rem <-  rep(NA, 1000)
```

```r
  # for remediation needs
for (i in 1: 1000){
  if (caa[i] >= 85){
     rem[i] = 0
  } else {
    rem[i] = sample(c(0,1,2,3), 1, replace = F)
  }
}

  # create a dataset for later use
mydat <- data.frame(treatment, gender, age, pell, caa, rem, eth)
mydat %>% group_by(treatment) %>% dplyr::select(one_of(names(mydat))) %>%
       summarise_all(funs(mean(., na.rm = T)))
## # A tibble: 2 x 7
##   treatment gender   age  pell   caa   rem   eth
##       <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1         0  0.668  23.0 0.565  78.0  1.2  0.73
## 2         1  0.675  18.8 0.91   64.7  1.48 0.865

# for world A: both linear outcomes
set.seed(2)
y0_a <- gender - 0.04*age - pell + 0.08*caa - 1.5*rem + rnorm(1000, mean = 0,
sd = 0.2)
y0_a[y0_a > 4] = runif(length(y0_a[y0_a > 4]), 2, 4)
y0_a[y0_a < 1] = runif(length(y0_a[y0_a < 1]), 0, 4)
summary(y0_a)

##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.007468 2.097372 2.703564 2.630443 3.370887 3.994338

y1_a <- y0_a + rnorm(1000, mean = 0.5, sd = 0.2)
y1_a[y1_a > 4] = 4
summary(y1_a)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2285  2.5701  3.2080  3.0776  3.8781  4.0000

mean(y1_a - y0_a)

## [1] 0.4471861

fullA <- data.frame(gender, age, pell, caa, rem, y0_a, y1_a)
y_a <- (1 - treatment) *y0_a + treatment*y1_a
obsA <- data.frame(treatment, gender, age, pell, caa, rem, y = y_a)

# for world B: non-linear potential outcomes
set.seed(3)
y0_b <- 0.35*sqrt(caa) + 0.2*age - gender - 2*pell - 1.5*rem + rnorm(1000,
mean = 0, sd = 0.2)
y0_b[y0_b > 4] = runif(length(y0_b[y0_b > 4]), 2, 4)
y0_b[y0_b < 1] = runif(length(y0_b[y0_b < 1]), 1, 4)
```

```
summary(y0_b)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.002   2.236   2.845   2.771   3.364   3.996

y1_b <- y0_b + rnorm(1000, mean = 0.5, sd = 0.2)
y1_b[y1_b > 4] = 4
summary(y1_b)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.122   2.766   3.335   3.215   3.890   4.000

mean(y1_b - y0_b)

## [1] 0.4439877

fullB <- data.frame(gender, age, pell, caa, rem, y0_b, y1_b)
y_b <- (1 - treatment) *y0_b + treatment*y1_b
obsB <- data.frame(treatment, gender, age, pell, caa, rem, y = y_b)

# for world C: non-linear potential outcomes, another confounding covariate
set.seed(4)
y0_c <- 0.3*sqrt(caa) + 0.2*age + gender - 1.5*pell - rem - 2*eth +
rnorm(1000, mean = 0, sd = 0.2)
y0_c[y0_c > 4] = runif(length(y0_c[y0_c > 4]), 2, 4)
y0_c[y0_c < 1] = runif(length(y0_c[y0_c < 1]), 0, 4)
summary(y0_c)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0345  2.3355  2.9114  2.8229  3.4542  3.9957

y1_c <- y0_c + rnorm(1000, mean = 0.5, sd = 0.2)
y1_c[y1_c > 4] = 4
summary(y1_c)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2829  2.8114  3.4110  3.2619  3.9814  4.0000

mean(y1_c - y0_c)

## [1] 0.4389479

fullC <- data.frame(gender, age, pell, caa, rem, eth, y0_c, y1_c)
y_c <- (1 - treatment) *y0_c + treatment*y1_c
obsC <- data.frame(treatment, gender, age, pell, caa, rem, eth, y = y_c)
```

(4) Methods and Estimand

The estimand of interest is **ATT (Average Treatment Effect on Treated)**.
Two methods will be used to estimate the treatment effect:
(1) linear regression: fit a regression of outcomes to the treatment and five confounding covariates;
(2) Propensity score matching

- step 1: select (potential) confounders represented in the model (five covariates in this example);
- step 2: estimate propensity scores: use logistic regression in this example;
- step 3: restructure the data: use k-1 with replacement weights and normalized IPTW ATT weights.
  - Perform one-to-one nearest neighbor matching with replacement using the estimated propensity score (apply the matching command in the arm package), the "cnts" variable in the output reflects the number of times each control observation was used as a match (the length is equal to the number of control observations). Use the output of this function to create a weight variable (equals one for treated observations, and equals the number of times used as a match for non-treated observations);
  - Apply the Inverse Probability of Treatment Weighting (IPTW) to re-weight the control group to look like the treatment group.
- step 4: check overlap and balance. The overlap checking is to make sure there is sufficient common support between treatment and control group, in other words, for each treatment group member there is a control group member that is sufficiently similar that we believe they can act as an empirical counterfactual. As for the balance examination, it aims to create a restructured dataset that looks like the output of a randomized experiment.
  - check overlap on the *unmatched* data using some diagnostic plots;
  - examining Balance. Build a function takes observed data frame created in step 2, a vector of covariate names, and weights from propensity score matching as inputs, the outputs should include: mean in unmatched treatment/control group, mean in matched treatment/control group, unmatched/matched mean difference (standardized for continuous variables, not standardized for binary variables), ratio of standard deviations across unmatched treated/control groups and ratio of standard deviations across matched treated/control groups;
- repeat steps 2-4 within the matching framework if needed;
- step 5: estimate average treatment effect on the restructured data.

```
# Method 1: Linear regression with five confounding variables
fit.a <- lm(y ~ ., data = obsA)
summary(fit.a)

##
## Call:
## lm(formula = y ~ ., data = obsA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.52889 -0.58478 -0.00649  0.54489  2.15923
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.430191   0.337898  10.152  < 2e-16 ***
```

```
## treatment    0.280971   0.085422   3.289  0.00104 **
## gender       0.211253   0.050929   4.148 3.64e-05 ***
## age         -0.010121   0.008221  -1.231  0.21855
## pell        -0.119897   0.052082  -2.302  0.02154 *
## caa         -0.001286   0.003528  -0.364  0.71562
## rem         -0.403444   0.021399 -18.854  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7575 on 993 degrees of freedom
## Multiple R-squared:  0.2914, Adjusted R-squared:  0.2871
## F-statistic: 68.05 on 6 and 993 DF,  p-value: < 2.2e-16
```

```
fit.b <- lm(y ~ ., data = obsB)
summary(fit.b)
```

```
##
## Call:
## lm(formula = y ~ ., data = obsB)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.64180 -0.54324  0.04193  0.51335  1.63979
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.188688   0.303095  10.520  < 2e-16 ***
## treatment    0.453074   0.076624   5.913 4.62e-09 ***
## gender      -0.077253   0.045683  -1.691 0.091139 .
## age          0.025079   0.007374   3.401 0.000698 ***
## pell        -0.205498   0.046718  -4.399 1.21e-05 ***
## caa         -0.006409   0.003164  -2.026 0.043079 *
## rem         -0.244400   0.019195 -12.733  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6795 on 993 degrees of freedom
## Multiple R-squared:  0.1855, Adjusted R-squared:  0.1806
## F-statistic:  37.7 on 6 and 993 DF,  p-value: < 2.2e-16
```

```
fit.c <- lm(y ~ .-eth, data = obsC)
summary(fit.c)
```

```
##
## Call:
## lm(formula = y ~ . - eth, data = obsC)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.34370 -0.57166  0.06376  0.56247  1.68696
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.223372   0.314488   7.070 2.92e-12 ***
## treatment    0.239602   0.079504   3.014  0.00265 **
## gender       0.227683   0.047400   4.803 1.80e-06 ***
## age          0.024756   0.007651   3.235  0.00125 **
## pell        -0.136174   0.048474  -2.809  0.00506 **
## caa          0.003820   0.003283   1.164  0.24490
## rem         -0.206796   0.019916 -10.383  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7051 on 993 degrees of freedom
## Multiple R-squared:  0.141,  Adjusted R-squared:  0.1358
## F-statistic: 27.17 on 6 and 993 DF,  p-value: < 2.2e-16

# Method 2: propensity score matching
 # way 1: logistic regression and k-1 with replacement matching
fit<- glm(treatment ~ ., data = mydat[,-7], family = binomial(link =
"logit"))
pscores <- data.frame(pscore = predict(fit, type = "response"), treatment =
fit$model$treatment)
match <- matching(z = mydat$treatment, score = pscores$pscore, replace = T)

wts <- data.frame(treatment = mydat$treatment, weight = NA)
wts[wts$treatment == 0, ]$weight = match$cnts
wts[wts$treatment == 1, ]$weight = 1
table(wts$weight, wts$treatment, useNA = "ifany")

##
##        0   1
##   0  750   0
##   1   25 200
##   2    7   0
##   3    3   0
##   4    2   0
##   6    3   0
##   7    4   0
##   10   1   0
##   11   1   0
##   12   2   0
##   24   1   0
##   29   1   0

# way 2: logistic regression and normalized IPTW weights for ATT
wts_iptw <- pscores
wts_iptw$weight_raw <- ifelse(wts_iptw$treatment == 0, wts_iptw$pscore / (1 -
wts_iptw$pscore), 1)
wts_iptw[wts_iptw$weight_raw > 50,]$weight_raw <- 50 # adjust extreme weights
```

```r
  # normalize the weights
t_sum <- sum(wts_iptw[wts_iptw$treatment == 1,]$weight_raw)
c_sum <- sum(wts_iptw[wts_iptw$treatment == 0,]$weight_raw)
wts_iptw$weight <- ifelse(wts_iptw$treatment == 0,
wts_iptw$weight_raw*t_sum/c_sum, wts_iptw$weight_raw)

tapply(wts_iptw$weight_raw, wts_iptw$treat, summary)
```

```
## $`0`
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  0.00000  0.00033  0.00294  0.35746  0.03288 50.00000
##
## $`1`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       1       1       1       1       1
```

```r
tapply(wts_iptw$weight_raw, wts_iptw$treat, sum)
```

```
##        0        1
## 285.9645 200.0000
```

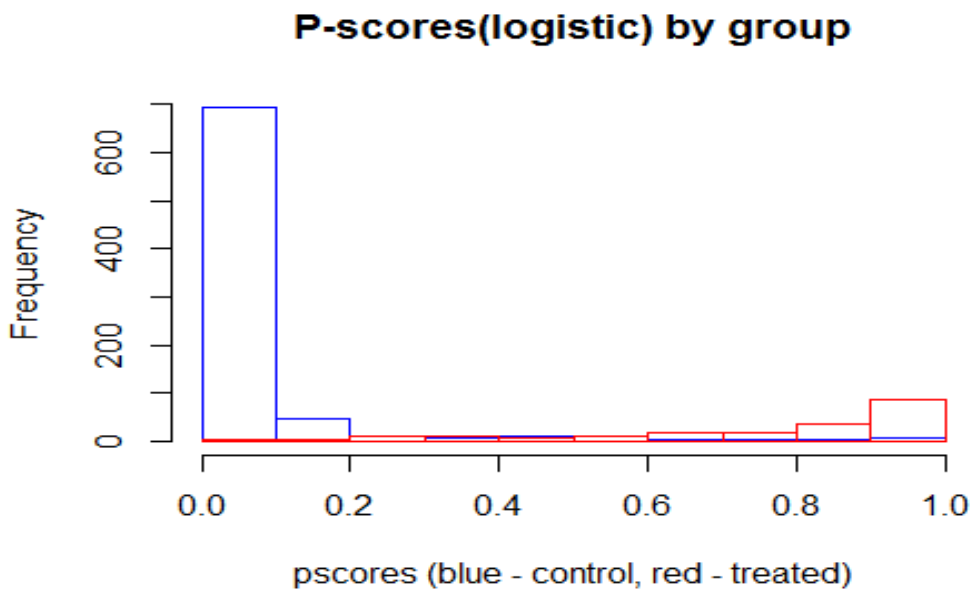
```r
tapply(wts_iptw$weight, wts_iptw$treat, summary)
```

```
## $`0`
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  0.00000  0.00023  0.00206  0.25000  0.02300 34.96937
##
## $`1`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       1       1       1       1       1
```

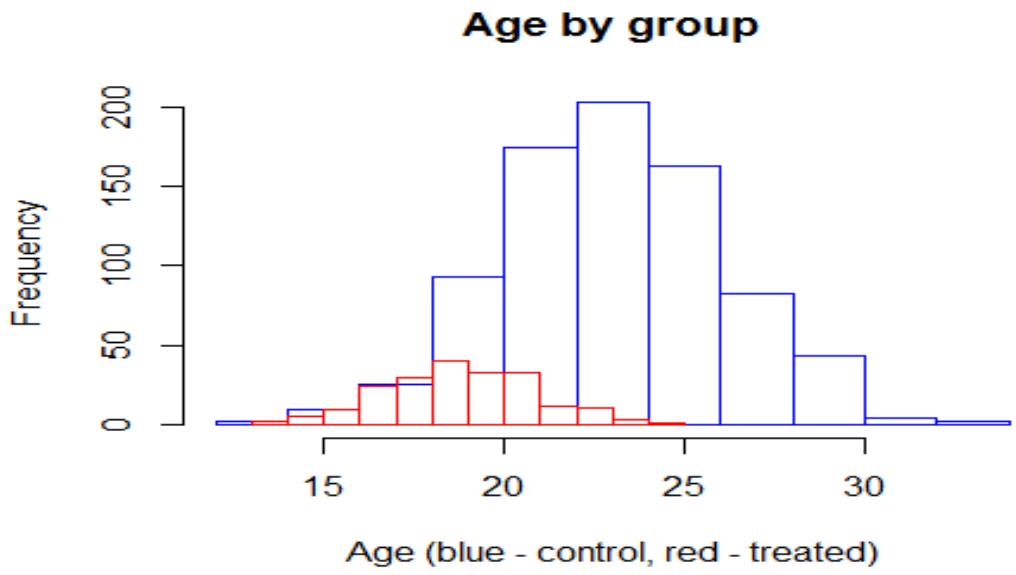
```r
tapply(wts_iptw$weight, wts_iptw$treat, sum)
```

```
##   0   1
## 200 200
```

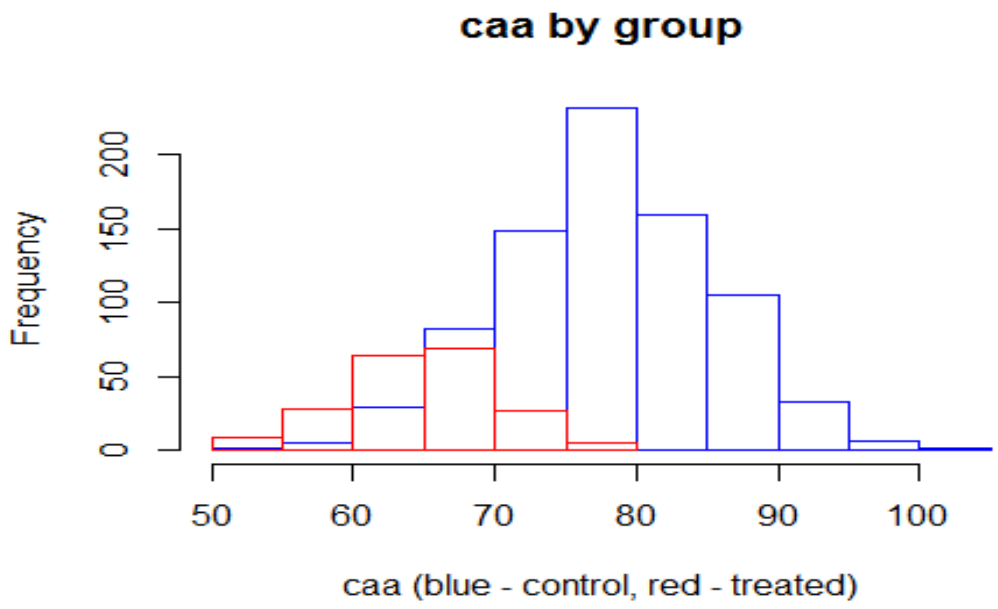
```r
# examining the overlap
  # overlap by propensity scores
hist(pscores$pscore[pscores$treatment == 0], border = "blue", main = "P-
scores(logistic) by group", xlab = "pscores (blue - control, red - treated)")
hist(pscores$pscore[pscores$treatment == 1], border = "red", add =T)
```




```
   # overlap by age
hist(mydat$age[mydat$treatment == 0], border = "blue", main = "Age by group",
xlab = "Age (blue - control, red - treated)")
hist(mydat$age[mydat$treatment == 1], border = "red", add = T)
```




```
   # overlap by caa
hist(obsB$caa[obsB$treatment == 0], border = "blue", main = "caa by group",
xlab = "caa (blue - control, red - treated)")
hist(obsB$caa[obsB$treatment == 1], border = "red", add = T)
```




```
# examining the balance
check_balance <- function(df, covs, wts_df){

      treated <- df[df$treatment == 1, ]
      control <- df[df$treatment == 0, ]

      treated_wts<- wts_df[wts_df$treatment == 1, "weight"]
      control_wts<- wts_df[wts_df$treatment == 0, "weight"]

      bi_var <- ifelse(sapply(covs, function(x) length(unique(df[, x]))) ==
2, "Y", "N")
      weighted_sd <- function(x, w) sqrt(sum(w*(x - weighted.mean(x,
w))^2)/sum(w))

      mn1 <- sapply(covs, function(x) mean(treated[, x]))
      mn0 <- sapply(covs, function(x) mean(control[, x]))

      mn1.m <- sapply(covs, function(x) weighted.mean(treated[, x],
treated_wts))
      mn0.m <- sapply(covs, function(x) weighted.mean(control[, x],
control_wts))

      diff <- ifelse(bi_var == "Y", mn1-mn0, (mn1-mn0)/sapply(covs,
function(x) sd(treated[, x])))
      diff.m <- ifelse(bi_var == "Y", mn1.m-mn0.m, (mn1.m-mn0.m)/sapply(covs,
function(x) sd(treated[, x])))

      ratio <- ifelse(bi_var == "N", sapply(covs, function(x) sd(control[,
x])) / sapply(covs, function(x) sd(treated[, x])), NA)
```

```
      ratio.m <- ifelse(bi_var == "N", sapply(covs, function(x)
weighted_sd(control[, x], control_wts)) / sapply(covs, function(x)
weighted_sd(treated[, x], treated_wts)), NA)

      data.frame(mn1, mn0, mn1.m,mn0.m, diff, diff.m, ratio, ratio.m)
}

confounders <- c("gender", "age","pell", "caa", "rem")
(balance_k1 <- round(check_balance(obsA, confounders, wts), 2))

##          mn1   mn0 mn1.m mn0.m  diff diff.m ratio ratio.m
## gender  0.68  0.67  0.68  0.60  0.01   0.08    NA      NA
## age    18.76 23.00 18.76 18.14 -2.07   0.30  1.51    1.17
## pell    0.91  0.56  0.91  0.91  0.35   0.00    NA      NA
## caa    64.71 78.01 64.71 65.98 -2.60  -0.25  1.45    0.98
## rem     1.49  1.20  1.49  1.29  0.25   0.17  1.03    1.03

(balance_iptw <- round(check_balance(mydat, confounders, wts_iptw),2))

##          mn1   mn0 mn1.m mn0.m  diff diff.m ratio ratio.m
## gender  0.68  0.67  0.68  0.65  0.01   0.02    NA      NA
## age    18.76 23.00 18.76 17.64 -2.07   0.55  1.51    1.18
## pell    0.91  0.56  0.91  0.93  0.35  -0.02    NA      NA
## caa    64.71 78.01 64.71 64.28 -2.60   0.08  1.45    1.03
## rem     1.49  1.20  1.49  1.19  0.25   0.26  1.03    0.92

# estimate the ATT effect
data.a.k1 <- cbind(obsA, pscores = pscores$pscore, weight = wts$weight)
data.a.iptw <- cbind(obsA, pscores = pscores$pscore, weight =
wts_iptw$weight)


data.b.k1 <- cbind(obsB, pscores = pscores$pscore, weight = wts$weight)
data.b.iptw <- cbind(obsB, pscores = pscores$pscore, weight =
wts_iptw$weight)


data.c.k1 <- cbind(obsC, pscores = pscores$pscore, weight = wts$weight)
data.c.iptw <- cbind(obsC, pscores = pscores$pscore, weight =
wts_iptw$weight)

estimate.a1 <- lm(y ~ treatment + weight, data = data.a.k1)
estimate.a2 <- lm(y ~ treatment + weight, data = data.a.iptw)

estimate.b1 <- lm(y ~ treatment + weight, data = data.b.k1)
estimate.b2 <- lm(y ~ treatment + weight, data = data.b.iptw)

estimate.c1 <- lm(y ~ treatment + weight, data = data.c.k1)
estimate.c2 <- lm(y ~ treatment + weight, data = data.c.iptw)
```

(5) Assumptions Required

In order to yield a valid causal estimate for the estimand ATT by using propensity score matching, the most important assumptions required include:

- **Ignorability of the treatment assignment**. Ignorablity assumption assumes all confounders have been measured. Propensity score is a one-number summary of all the confounding covariates, thus, if ignorability holds, we should be able to get unbiased treatment effect estimates with properly estimated propensity score. In our example, if the five covariates (gender, age, pell, caa, rem) are the only confounding covariates, we can say that the ignorability assumption is satisfied;
- **Sufficient overlap**. To make a valid causal inference effect estimate, the treatment and control groups should have sufficient overlap, or common support. Otherwise, we have to either restrict inferences to the region of overlap, or rely on a model to extrapolate outside this region;
- **Appropriate balance achieved from the propensity score model**. Imbalance occurs if the distributions of the predictors are not similar across groups. Imbalance creates problems primarily because it forces us to rely more on the correctness of the model than we would have to if the samples were balanced. Therefore, in order to make a valid causal inference effect estimate, the potential confounders should achieve a good balance through the selected best propensity score estimation model before making the estimation;
- **SUTVA (Stable Unit Treatment Value Assumption)**. No interference across units and no hidden versions of the treatment. In other words, each subject's potential outcome is defined in terms of only his or her own treatment assignment. In our example, it means that one student's participation in the ASAP program would not affect another student's cumulative GPA score on graduation. SUTVA also implies that there are no hidden versions of treatments, that is, all subjects receive the same well-defined treatment.

(6) Results

Results of estimates and standard error of the three methods for each world are shown below. For world A, because the outcomes are simulated from a linear equation, the linear regression yields the closest estimated ATT effect as expected; For World B, the estimated effects form both propensity score matching weights are similar, because the outcomes are simulated non-linear, linear regression is not a good choice, and the estimate is probably extrapolated; Finally, in world C, because the propensity scores failed to consider one influential confounding covariate, both estimates of ATT are not reliable.

In terms of the estimate from the *Logit regression + IPTW weights method* in world B, the causal interpretation is:

For students who participated in the ASAP program among this analysis sample, their cumulative GPA on graduation were about 0.3 points higher than had they not attended to the ASAP program.

World	Method	ATT Estimate	Standard Error
A	Unmatched	**0.19**	-
A	Linear regression	**0.28**	**0.085**
A	Logit + K1 weights	**0.20**	**0.072**
A	Logit+IPTW weights	**0.19**	**0.072**
B	Unmatched	**0.29**	-
B	Linear regression	**0.45**	**0.077**
B	Logit + K1 weights	**0.29**	**0.060**
B	Logit+IPTW weights	**0.30**	**0.059**
C	Unmatched	**-0.02**	-
C	Linear regression	**0.24**	**0.080**
C	Logit + K1 weights	**0.02**	**0.061**
C	Logit+IPTW weights	**0.01**	**0.060**

```
result <- data.frame(world = rep(c("A", "B", "C"), each = 4),
                     method = rep(c("Unmatched", "Linear", "Logit + K1
weights", "Logit+IPTW weights"), 3), effect = NA, sd = NA)
# unmatched estimates
result$effect[1] = diff(tapply(obsA$y, obsA$treatment, mean))
result$effect[5] = diff(tapply(obsB$y, obsB$treatment, mean))
result$effect[9] = diff(tapply(obsC$y, obsC$treatment, mean))

# linear estimates and standard error
result$effect[2] = coef(fit.a)["treatment"]
result$effect[6] = coef(fit.b)["treatment"]
result$effect[10] = coef(fit.c)["treatment"]
result$sd[2] = summary(fit.a)$coefficients["treatment","Std. Error"]
result$sd[6] = summary(fit.b)$coefficients["treatment","Std. Error"]
result$sd[10] = summary(fit.c)$coefficients["treatment","Std. Error"]

# logit regression and k1 weights
result$effect[3] = coef(estimate.a1)["treatment"]
result$effect[7] = coef(estimate.b1)["treatment"]
result$effect[11] = coef(estimate.c1)["treatment"]
result$sd[3] = summary(estimate.a1)$coefficients["treatment","Std. Error"]
result$sd[7] = summary(estimate.b1)$coefficients["treatment","Std. Error"]
result$sd[11] = summary(estimate.c1)$coefficients["treatment","Std. Error"]

# logit regression and IPTW weights
result$effect[4] = coef(estimate.a2)["treatment"]
result$effect[8] = coef(estimate.b2)["treatment"]
result$effect[12] = coef(estimate.c2)["treatment"]
result$sd[4] = summary(estimate.a2)$coefficients["treatment","Std. Error"]
result$sd[8] = summary(estimate.b2)$coefficients["treatment","Std. Error"]
result$sd[12] = summary(estimate.c2)$coefficients["treatment","Std. Error"]
```

```
print(result)
##    world              method       effect         sd
## 1      A           Unmatched  0.186256542         NA
## 2      A              Linear  0.280970617 0.08542196
## 3      A Logit + K1 weights  0.200759321 0.07209284
## 4      A Logit+IPTW weights  0.190941396 0.07153787
## 5      B           Unmatched  0.290732612         NA
## 6      B              Linear  0.453074477 0.07662359
## 7      B Logit + K1 weights  0.288303530 0.05982718
## 8      B Logit+IPTW weights  0.300171674 0.05930755
## 9      C           Unmatched -0.020462936         NA
## 10     C              Linear  0.239601786 0.07950363
## 11     C Logit + K1 weights  0.016335362 0.06089001
## 12     C Logit+IPTW weights  0.008103346 0.06037867
```

(7) Discuss the Bias

To use linear regression models for purposes of inference or prediction, the most important assumption is that there must be a linear relationship between the outcome variable and the covariates. Moreover, using linear regression to estimate a causal effect has a risk of extrapolating over parts of the covariate space with no data. For example, in world B and C, because the outcomes and predictors are not linear related, applying a linear regression would not yield a good estimate. Even there is a linear relation between outcomes and covariates in world A, the estimated effect might still suffer the extrapolation issue as a result of unmatched data.

In contrast, propensity score matching is a more robust strategy for causal inference estimation. It reduces the bias by assembling comparable samples in which confounding covariates are balanced across groups, further, limiting the data sample to where there is sufficient overlap makes a big difference in what can be reliably estimated. For our example, in both world A and world B, when applying the propensity score matching, we see overlap and balance between treatment and control groups, thus, the results tend to be more reliable if other assumptions hold. Other than overlap and balance, the ignorability is also critical for an unbiased estimate, although it is generally hard to verify in real life practices. As the world C demonstrated, failing to include an influential covariate could lead to a questionable estimate.

(8) Conclusion

In general, linear regression might not be a good strategy for estimating causal inference effect, since it estimates the average effect of a combination of covariates, and is not aiming to estimate the causal effects.

The fundamental problem of causal inference is that the causal effect cannot be directly measured as we can never observe both Y(0) and Y(1), thus the efforts of different strategies are aiming to create comparable groups to gain a substitution of their counterfactual outcomes. In terms of propensity score matching, assuming all confounding covariates are controlled, the propensity score, as a one-number summary of all controlled

covariates, is used to assign weights for all individuals in the sample, and hope to obtain a well-balanced comparable groups to better estimate the causal effect unbiasedly.

Although Propensity Score Matching (PSM) has been an enormously popular method of preprocessing data for causal inference, recent research (*King & Nielsen, 2018*) claims it might actually lead to the opposite of its intended goals - increasing imbalance, inefficiency, model dependence, and bias. In fact, the more balanced the data, or the more balanced it becomes by pruning some observations through matching, the more likely PSM will degrade inferences - a problem referred to as the PSM paradox. Actually, we can sort of tracing this paradox in our current example.