HAD7002 HW2

Due May 21, 2024

Question 1

(a) Based on the Table 2.2 data, calculate the inverse probability of treatment weighted estimate, and verify that this is equivalent to the directly standardized estimate.

- Step 1: fit the treatment model. The model is specified as:

  $$\text{logit}(P(A=1|L,\theta)) = \theta_0 + \theta_1 L$$

  The following R codes are used to fit this model.

```
## fit treatment model
Q1_ps_model <- glm( a ~ l, family=binomial(link=logit), data=greek_data)
```

- Step 2: calculate the treatment weight. The propensity score is obtained from the fitted model using `predict` function call. The weight for individuals with $A=1$ is $\frac{1}{P(A_i=1|L_i)}$ and weight for individuals with $A=0$ is $\frac{1}{P(A_i=0|L_i)} = \frac{1}{1-P(A_i=1|L_i)}$. Since L only has two levels,and the exposure of interest is binary, there are 4 different combinations of A and L for an individual. So the weights can only take list all possible weights(see last row of the following table)

```
## get predicted treatment probs
treatment_pred <- predict(Q1_ps_model, type="response")

## calculate the weights
iptw_dat <- cbind(greek_data, treatment_pred) %>%
  mutate(weight = ifelse(a == 1,
                         1 / treatment_pred,
                         1 / (1 - treatment_pred))
         )
```

```
## display weights
iptw_dat %>% select(a,l,treatment_pred,weight) %>% unique()
```

```
   a l treatment_pred   weight
1  0 0           0.50 2.000000
5  1 0           0.50 2.000000
9  0 1           0.75 4.000000
12 1 1           0.75 1.333333
```

```
# The weighted sample size is `r
# sum((greek_data$a==1)/treatment_pred +
# (greek_data$a==0)/(1-treatment_pred))`
```

- Step 3: the IPTW estimator is given by:

$$\frac{1}{n}\sum_{i=1}^{n}\frac{A_i Y_i}{P(A_i = 1 \mid L_i)} - \frac{1}{n}\sum_{i=1}^{n}\frac{(1-A_i)Y_i}{P(A_i = 0 \mid L_i)}$$

  Using this equation, we get the $RD_{IPW} = 0.5 - 0.5 = 0$. And the result is the same as the direct standardization we saw in HW1.

```
data.frame(
  effect_a1 = mean((greek_data$a == 1) * greek_data$y / treatment_pred),
  effect_a0 = mean((greek_data$a == 0) * greek_data$y / (1 - treatment_pred))
) %>%
  mutate(risk_diff = round(effect_a1 - effect_a0, 2))
```

```
  effect_a1 effect_a0 risk_diff
1       0.5       0.5         0
```

We only have one covariate, the treatment model is a saturated model. So we obtained a weighted sample population where the size of the weighted sample in each treatment group ($A = 1$ or $A = 0$) is the sample as original total study sample. So we would expect the same results of the IPTW and normalized IPTW estimates. and following we calculated the normalized IPTW estimates for verification.

```
## normalized IPTW
iptw_dat %>%
  group_by(a) %>%
  summarise(risk = sum(y*weight)/sum(weight))
```

```
# A tibble: 2 x 2
      a  risk
  <dbl> <dbl>
1     0 0.500
2     1 0.500
```

(b) Use the bootstrap to calculate 95% confidence interval for the directly standardized mean difference in Assignment 1 Q2(a). Was the intervention effect statistically significant?

The main codes for computing the bootstrapped 95% confidence interval is provided below. The standardization estimator is applied for 1000 times, 1000 estimated standardized mean difference are obtained. For each iteration, we applied the standard_mean_f we defined(codes provided in the appendix) to get one estimate of standardized mean difference.

```
library(causaldata)
data(nhefs_complete)

Q2_dat <- nhefs_complete %>%
  select(qsmk,wt82_71,age,sex,race,wt71,education,smokeintensity) %>%
  mutate(education = as.factor(ifelse(education==5,1,0)))

input_dat <- Q2_dat
set.seed(1017)
boot_standard_means <- lapply(1:1000, function(x){
  boot_id <- sample(c(1:nrow(input_dat)), size=nrow(input_dat), replace=TRUE)
  boot_dat <- input_dat[boot_id,]
  standard_mean_f(dat = boot_dat)[[2]]
})

## quantiles
quantile_95ci <- quantile(boot_standard_means %>% unlist(),c(0.025,0.5,0.975))

## normal approx.
norm_approx_95ci <- mean(boot_standard_means%>% unlist())+ qnorm(c(0.025,0.5,0.975)) * sd(boo

rbind(quantile_95ci,norm_approx_95ci)
```

```
                     2.5%     50%    97.5%
quantile_95ci    2.349750 3.34000 4.270000
norm_approx_95ci 2.371023 3.33209 4.293157
```

From the bootstrap output, we see the 2.5%-97.5% quantile of the mean difference is (2.35,4.27), and the normal approximation based 95% CI is (2.37,4.29). The results are similar, and the CIs do not include 0 (null hypothesis: mean difference of weight change between two groups is 0), indicating the intervention effect is statistically significant. On average, the weight change in individuals who quit smoking during first questionnaire and 1982 is greater than that in individuals who did not quit smoking.

(c) Use the bootstrap to calculate 95% confidence intervals for the stratified (by sex) mean differences in Assignment 1 Q2(b). How can you use the bootstrap results to check if there was statistically significant effect modification?

The conditional averaged treatment effect are:

- for male: $E(Y_{a=1}|\text{sex} = \text{male}) - E(Y_{a=0}|\text{sex} = \text{male})$
- for female: $E(Y_{a=1}|\text{sex} = \text{female}) - E(Y_{a=0}|\text{sex} = \text{female})$

where Y is weight change (the wt82_71 variable). The conditional effect can be calculated using the standardized quantities as follows:

$$E[Y_a \mid V = v] = \sum_{\ell} E[Y \mid A = a, L = \ell, V = v] P(L = \ell \mid V = v) = E(E[Y \mid A = a, L = \ell, V = v])$$

where Y is the outcome, A is the exposure of interest, L is a vector of variables we want to adjust for for confounding effects. V is the potential effect modifier. In our case, V represents sex. $E[Y \mid A = a, L = \ell, V = v]$ can be estimated from regression model which is specified as:

$$E(Y|A, L, V, \theta) = \theta_0 + \theta_1 A + \theta_2 V + \theta_3 AV + {\theta_p}^T L$$

The following codes are used to define a function condition_standard_mean_f to calculate conditional ATE. and the bootstrap confidence interval can be obtained by repeating condition_standard_mean_f using bootstrapped samples

```
condition_standard_mean_f <- function(dat = Q2_dat,
                                      interaction = "qsmk*sex") {
  xvariables <- select(dat, -c(wt82_71))
  xvariables <- colnames(xvariables)
  ## fit the outcome model with interaction term qsmk*sex
  formula <- as.formula(paste0(
    "wt82_71 ~ ",
    paste(c(interaction,xvariables), collapse = "+")
  ))

  Q2_model <- glm(formula,
    family = gaussian(link = "identity"),
    data = dat
  )
  ## prediciton
  Q2_pred_dat <- rbind(
    select(dat, -c(qsmk, wt82_71)),
    select(dat, -c(qsmk, wt82_71))
  ) %>%
    mutate(qsmk = c(rep(0, nrow(dat)), rep(1, nrow(dat))))

  Q2_pred <- predict(Q2_model, Q2_pred_dat) %>%
    data.frame()

  outcome_pred_dat <- cbind(Q2_pred_dat, weight_gain_pred = Q2_pred$.)
  ## compute the conditional ATE
  mean_diff_table <- outcome_pred_dat %>%
    ## group by qsmk sex
    group_by(qsmk,sex) %>%
    summarise(mean_weight_change = mean(weight_gain_pred))%>%
    group_by(sex) %>%
    summarise(stratified_MD = diff(mean_weight_change))

 male_MD <-  mean_diff_table$stratified_MD[mean_diff_table$sex==1]
 female_MD <-  mean_diff_table$stratified_MD[mean_diff_table$sex==0]

 DID = male_MD - female_MD

 return(c(male_MD,female_MD,DID))

}
```

```
Q2_dat <- nhefs_complete %>%
  select(qsmk,wt82_71,age,sex,race,wt71,education,smokeintensity) %>%
  mutate(education = as.factor(ifelse(education==5,1,0)))

input_dat <- Q2_dat
set.seed(1017)
boot_stratified_standard_means <- lapply(1:1000, function(x){
  boot_id <- sample(c(1:nrow(input_dat)), size=nrow(input_dat), replace=TRUE)
  boot_dat <- input_dat[boot_id,]
  condition_standard_mean_f(dat = boot_dat)
})

saveRDS(boot_stratified_standard_means,file="data/boot_stratified_standard_means.rds")
```

```
boot_stratified_standard_means <- readRDS("data/boot_stratified_standard_means.rds")
## quantiles
CI_output <- apply(boot_stratified_standard_means %>% as.data.frame,1,quantile,c(0.025,0.5,0

colnames(CI_output) <- c('male mean difference','female mean difference','Difference in Diffe

CI_output
```

```
      male mean difference female mean difference Difference in Difference
2.5%              1.537904               2.362300                -2.480138
50%               3.037044               3.594377                -0.559725
97.5%             4.432109               4.865757                 1.320834
```

```
## normal approx.
# norm_approx_95ci <- mean(boot_standard_means%>% unlist())+ qnorm(c(0.025,0.975)) * sd(boot_
#
# rbind(quantile_95ci,norm_approx_95ci)
```

The upper table summarized the bootstrapped confidence interval of the mean difference for male, female, and their difference:
- for male: (1.54,4.43)
- for female: (2.36,4.87)
- difference: (-2.48,1.32) . the bootstrap confidence interval across 0, indicating the effect modification is not statistical significant.

(d) Implement the doubly robust estimator on Week 3 slide 11 for estimating the smoking cessation effect on weight gain. Calculate estimates for mean potential outcomes under smoking cessation and non-cessation, the mean difference, and bootstrap confidence intervals for these quantities.

The doubly robust estimator is defined as:

$$\frac{1}{n}\sum_{i=1}^{n}\left[E[Y_i \mid A_i = a, L_i; \hat{\phi}] + \frac{1_{\{A_i=a\}}}{P(A_i = a \mid L_i; \hat{\gamma})}(Y_i - E[Y_i \mid A_i = a, L_i; \hat{\phi}])\right]$$

where:

- A represent $qsmk$ status;
- L is the confound vector;

- $P(A_i = a \mid L_i; \hat{\gamma})$ is the the treatment model,estimated from a logistic model specified as:

```
Q1_ps_model <- glm(qsmk ~ age + race + sex + wt71 + education + smokeintensity,
                    data=Q2_dat, family=binomial(link=logit))
```

- $E[Y_i \mid A_i = a, L_i; \hat{\phi}]$, the outcome model which is specified as:

```
Q2_model <- glm(wt82_71 ~ qsmk + age + sex + race + wt71 + education + smokeintensity,
                family = gaussian(link = "identity"),
                data = Q2_dat)
```

The IPTW weights (from the treatment model) and the predicted weight gain (from the outcome model) were obtain as we did in HW1. The results were added to the working dataset. The following table lists the first 5 observations of the results.The **IPTweight** column are the weights,and the **weight_gain_pred** column indicates the predicted weight gain from the outcome model.

```
outcome_pred_dat %>% head(5)
```

```
    wt82_71 age sex race  wt71 education smokeintensity qsmk origin_qsmk
1 -10.093960  42   0    1 79.04         0             30    1           0
2   2.604970  36   0    0 58.63         0             20    1           0
3   9.414486  56   1    1 56.81         0             20    1           0
4   4.990117  68   0    1 59.42         0              3    1           0
5   4.989251  40   0    0 87.09         0             20    1           0
  IPTweight weight_gain_pred
1  1.148316         6.209740
2  1.307157         8.494406
3  1.166865         4.296175
4  1.488471         3.202845
5  1.420180         4.926471
```

Using these quantities, we can compute the doubly robust estimate as shown in the following code chunk.

```
DR_est <- outcome_pred_dat %>%
  mutate(
    ### indicator
    indicator = ifelse(origin_qsmk == qsmk,
                       IPTweight * (wt82_71 - weight_gain_pred),
                       0 ),
    DR = weight_gain_pred + indicator
  ) %>%
  group_by(qsmk) %>%
  summarise(DR_est = mean(DR))

### rearrange
all_estimates <- left_join(IPTW_est,
                           standardization_est,by = join_by(qsmk)) %>%
  left_join(DR_est,by = join_by(qsmk)) %>% data.frame()

rownames(all_estimates) <- paste0("qsmk=",as.character( all_estimates$qsmk))

Q1d_res <- all_estimates %>% select(-qsmk) %>% t() %>% as.data.frame() %>%
  mutate(`mean diff` = `qsmk=1`-`qsmk=0`)
Q1d_res
```

```
                  qsmk=1   qsmk=0 mean diff
IPTW_est        5.150621 1.800298  3.350323
standardize_est 5.119119 1.778652  3.340467
DR_est          5.174525 1.799172  3.375352
```

The first column is the mean weight change under smoking cessation, the second column is the mean weight change under non-cessation, the last column is the mean difference between $qsmk = 1$ and $qsmk = 0$ group. The doubly robust estimate is in the third row. we have:

- mean weight change under smoking cessation: $E(Y_1)_{DR} = 5.175$
- mean weight change under non-cessation: $E(Y_0)_{DR} = 1.799$
- mean difference: $E(Y_1)_{DR} - E(Y_0)_{DR} = 3.375$
  The IPTW estimate and model-based standardization estimate are also provided for comparison.(the IPTW_est and standardize_est rows)

Then bootstrap is used to get the confidence intervals.

```
input_dat <- nhefs_complete
set.seed(1017)
boot_doubly_robust <- lapply(1:1000, function(x){
  boot_id <- sample(c(1:nrow(input_dat)), size=nrow(input_dat), replace=TRUE)
  boot_dat <- input_dat[boot_id,]
  # iptw_boot <- doubly_robust_f(dat = boot_dat)[1,] ## iptw
  # stand_boot <- doubly_robust_f(dat = boot_dat)[2,] ## g-formula
  DR_boot <- doubly_robust_f(dat = boot_dat)[3,] ## extract DR est, others are not intereste
})

saveRDS(boot_doubly_robust,file="data/boot_doubly_robust.rds")
```

```
boot_doubly_robust <- readRDS("data/boot_doubly_robust.rds")

boot_doubly_robust_res <- boot_doubly_robust %>% bind_rows() %>% apply(2,quantile,c(0.025,0.5
boot_doubly_robust_res
```

```
        qsmk=1   qsmk=0 mean diff
2.5%  4.196181 1.378459  2.310639
50%   5.180089 1.801613  3.388432
97.5% 6.064114 2.213920  4.361990
```

The bootstrapped confidence intervals of the doubly robust estimator:
- for qsmk =1: (4.2,6.06)
- for qsmk =0: (1.38,2.21)
- for mean difference: (2.31,4.36)

Question 2

(a)(PS matching) Complete a 1:1 propensity score (PS) matching analysis to esti- mate the causal effect of being active in 1971 on the weight change between 1971 and 1982.

i) covariates distribution pre-matching

The covariates distributions are given in the following table. All SMDs are greater than 0.05, indicating potential unbalanced distribution between treatment groups (different active statuses).

```
## excercise has 3 levels
HW2Q2_dat <- nhefs_complete %>%
  select(active, qsmk,sex,race,age,school,
         smokeintensity,smokeyrs,exercise, wt71,wt82_71) %>%
  mutate(active_status = ifelse(active==2,0,1)) %>%
  select(-active)
```

```r
## pre-matching dist

library(tableone)
covariates <- select(HW2Q2_dat, -c(active_status, wt82_71))
baselines <- colnames(covariates)
baselines
```

```
[1] "qsmk"           "sex"            "race"           "age"
[5] "school"         "smokeintensity" "smokeyrs"       "exercise"
[9] "wt71"
```

```r
tab0 <- CreateTableOne(vars = baselines,
                       data = HW2Q2_dat,
                       strata = "active_status",
                       test = FALSE, #mute P-value calculation;
                       smd = TRUE,
                       addOverall = TRUE)
print(tab0, smd = TRUE, showAllLevels = FALSE)
```

```
                             Stratified by active_status
                              Overall       0             1             SMD
  n                            1566           149          1417
  qsmk (mean (SD))             0.26 (0.44)   0.30 (0.46)   0.25 (0.43)   0.110
  sex = 1 (%)                   804 (51.3)     84 (56.4)    720 (50.8)   0.112
  race = 1 (%)                  206 (13.2)     17 (11.4)    189 (13.3)   0.059
  age (mean (SD))             43.66 (11.99) 45.39 (12.33) 43.48 (11.94)  0.157
  school (mean (SD))          11.17 (3.07)  11.36 (3.32)  11.15 (3.04)   0.063
  smokeintensity (mean (SD)) 20.53 (11.77) 21.24 (11.41) 20.45 (11.81)  0.068
  smokeyrs (mean (SD))       24.59 (12.01) 25.63 (12.39) 24.48 (11.97)  0.095
  exercise (%)                                                           0.710
     0                          300 (19.2)     11 ( 7.4)    289 (20.4)
     1                          661 (42.2)     36 (24.2)    625 (44.1)
     2                          605 (38.6)    102 (68.5)    503 (35.5)
  wt71 (mean (SD))           70.83 (15.31) 71.72 (16.20) 70.74 (15.22)  0.062
```

ii) 1:1 matching

```r
library(MatchIt)
```

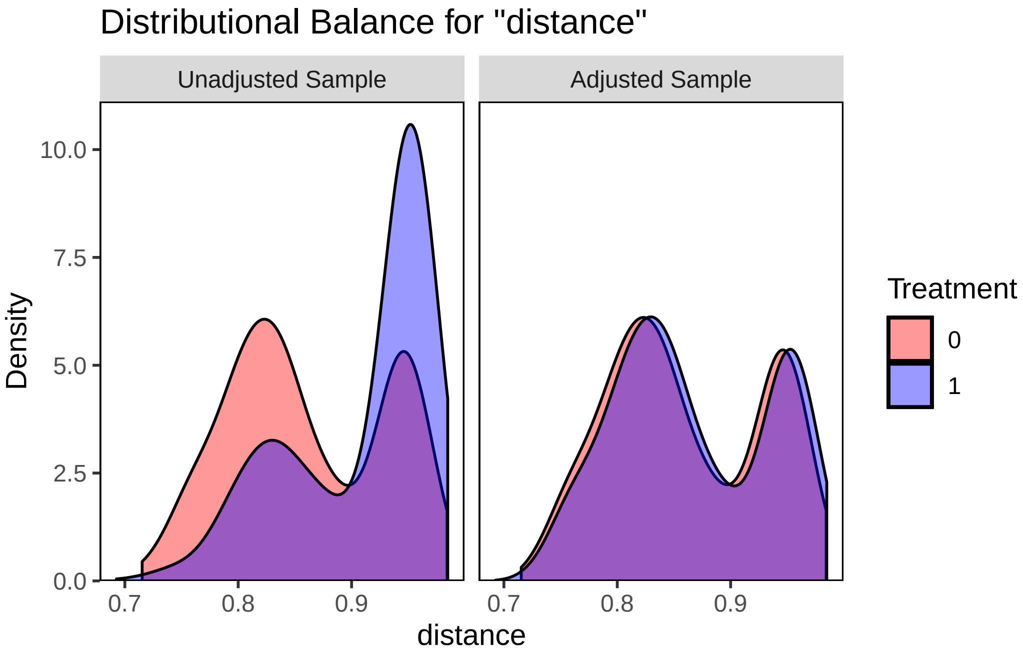
```r
set.seed(1017)

ps.formula <- as.formula(paste("active_status~",
                              paste(baselines, collapse = "+")))

PS.fit <- glm(ps.formula,family="binomial", data=HW2Q2_dat)


HW2Q2_dat$PS <- predict(PS.fit, newdata = HW2Q2_dat, type="response")

match.obj <- matchit(ps.formula, data =HW2Q2_dat,
                     distance = HW2Q2_dat$PS,
                     method = "nearest", #nearest neighbour;
                     replace=FALSE,
                     ratio = 1, #1:1 match;
                     caliper = .1)
```

The following plots are used to check the balance of the covariate distributions before and after matching: 1) the density plot of the propensity score in the matched samples have great overlap. 2) also, comparing the SMD values before and after matching, we see SMDs after matching are more around zero.

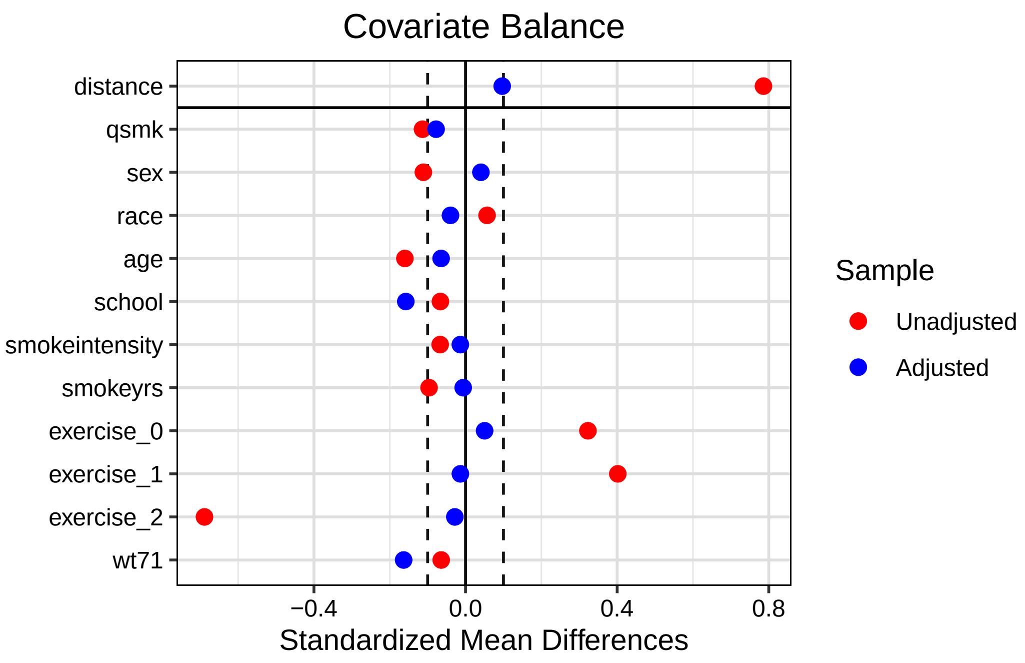
```r
library(cobalt)
```

```
Warning: package 'cobalt' was built under R version 4.3.2

 cobalt (Version 4.5.5, Build Date: 2024-04-02)


Attaching package: 'cobalt'

The following object is masked from 'package:MatchIt':

    lalonde
```

```r
bal.plot(match.obj,
         var.name="distance",
         which="both",
         type = "density",
         colors = c("red","blue"))
```



```
love.plot(match.obj,
          binary = "std",
          grid = TRUE,
          thresholds = c(m = .1),
          colors = c("red","blue"))
```



iii) estimate casual effect

```
library(geepack)
Match <- match.data(match.obj)
fit1 <- geeglm(wt82_71 ~ active_status,
               family=gaussian("identity"),
               data=Match,
               weights=weights,
               std.err = 'san.se',
               id=subclass,
               corstr="independence")
#sjPlot::tab_model(fit1)
```

the estimated casual effect is -1.08 (95% CI:[-2.94,0.78], P-value:0.26), however, the effect of the active status on the weight change is not statistical significant.

```
summary(fit1)
```

```
Call:
```

```
geeglm(formula = wt82_71 ~ active_status, family = gaussian("identity"), 
    data = Match, weights = weights, id = subclass, corstr = "independence", 
    std.err = "san.se")

 Coefficients:
              Estimate Std.err   Wald Pr(>|W|)    
(Intercept)     2.4989  0.7562 10.920 0.000951 ***
active_status  -1.0782  0.9488  1.291 0.255773    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation structure = independence 
Estimated Scale Parameters:

            Estimate Std.err
(Intercept)    66.54   7.294
Number of clusters:   295  Maximum cluster size: 2 
```

iv) unmatched observations

```
unmatched_dat <- HW2Q2_dat[match.obj$weights==0,]

tab_unmatched <- CreateTableOne(vars = baselines,
                       data = unmatched_dat,
                       strata = "active_status",
                       test = FALSE, #mute P-value calculation;
                       smd = TRUE,
                       addOverall = TRUE)
```

- the number of unmatched observations: 1270
- the percentage of unmatched observations:0.811 The following shows the distributions of the unmatched samples by active status. As there is only 1 subject in active=0 group, we see the distributions in two comparison groups are different.

```
print(tab_unmatched, smd = TRUE, showAllLevels = FALSE)
```

```
                            Stratified by active_status
                             Overall       0             1              SMD   
  n                           1270            1           1269               
  qsmk (mean (SD))            0.25 (0.43)   0.00 (NA)      0.25 (0.43)   NA
  sex = 1 (%)                    635 (50.0)      1 (100.0)    634 (50.0)   1.415
  race = 1 (%)                   174 (13.7)      0 (  0.0)    174 (13.7)   0.564
  age (mean (SD))              43.34 (11.86) 43.00 (NA)     43.34 (11.86)  NA
  school (mean (SD))           11.20 (3.04)  17.00 (NA)     11.19 (3.03)   NA
  smokeintensity (mean (SD)) 20.42 (11.92) 50.00 (NA)     20.40 (11.90)  NA
  smokeyrs (mean (SD))       24.34 (11.91) 15.00 (NA)     24.34 (11.92)  NA
  exercise (%)                                                           2.069
     0                           275 (21.7)      0 (  0.0)    275 (21.7)
     1                           590 (46.5)      0 (  0.0)    590 (46.5)
     2                           405 (31.9)      1 (100.0)    404 (31.8)
  wt71 (mean (SD))             70.92 (15.35) 76.20 (NA)     70.92 (15.35)  NA
```

(b) (TMLE steps) Using SuperLearner as the initial outcome model estimation algorithm (select 3 algorithms) and also using SuperLearner for the treatment model (select 3 algorithms). Please replicate the tutorial and perform step-by-step TMLE estimation to estimate the causal effect. Please report both the point estimate and its 95% confidence interval. We consider TMLE achieved convergence if $\hat{e} < 0.0001$.

step 1: Initial G-formula/outcome model estimate using superlearner and get predicted outcome

```r
## the data
set.seed(1017)
library(xgboost)
library(tmle)

HW2Q2_dat <- nhefs_complete[, c("qsmk", "sex","race","age", "school","smokeintensity",
                             "smokeyrs", "exercise","active", "wt71", "wt82_71")] %>%
mutate(active = ifelse(active=="2", 0, 1)) %>%
  mutate(id = 1:nrow(nhefs_complete)) %>%
  mutate(
         Y =wt82_71,
         A = active) %>%
  select(-active,-wt82_71)
data2 <- HW2Q2_dat
data2.noY <- select(data2, -c(id,Y))
```

```r
Y.fit.sl<-SuperLearner(Y=data2$Y,X=data2.noY,cvControl =list(V =3),
                       SL.library=c("SL.xgboost","SL.glmnet","SL.randomForest"),
                       family=gaussian())
```

```
Loading required namespace: randomForest
```

```
# saveRDS(Y.fit.sl, file = "data/SL_TMLE_G1")
```

```
## actual predicition
#Y.fit.sl <- readRDS(file = "data/SL_TMLE_G1")
# predict Y under the observed A;
# predict Y under the observed A;
data2$init.Pred <- predict(Y.fit.sl, newdata = data2.noY)$pred
summary(data2$init.Pred)
```

```
       V1
 Min.   :-9.066
 1st Qu.: 0.983
 Median : 2.773
 Mean   : 2.647
 3rd Qu.: 4.482
 Max.   :12.101
```

```
#treated;
data2.noY$A <- 1
data2$Pred.Y1 <- predict(Y.fit.sl, newdata = data2.noY)$pred
summary(data2$Pred.Y1)
```

```
       V1
 Min.   :-9.066
 1st Qu.: 0.993
 Median : 2.734
 Mean   : 2.628
 3rd Qu.: 4.405
 Max.   :12.101
```

```
#control;
data2.noY$A <- 0
data2$Pred.Y0 <- predict(Y.fit.sl, newdata = data2.noY)$pred
summary(data2$Pred.Y0)
```

```
       V1
 Min.   :-8.94
 1st Qu.: 1.13
 Median : 3.19  
 Mean   : 2.84  
 3rd Qu.: 4.86  
 Max.   :11.22  
```

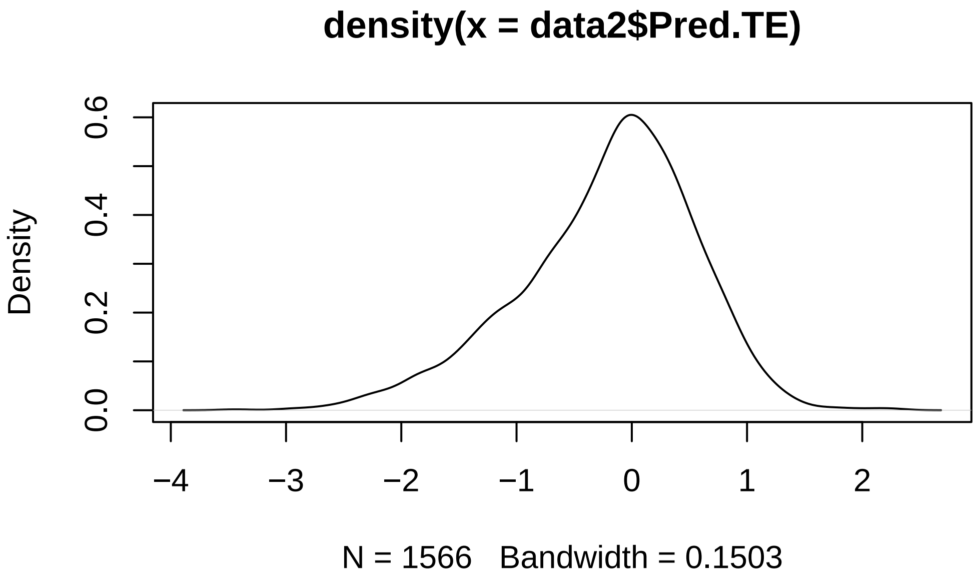
```
# Compute initial treatment effect estimate;
data2$Pred.TE <- data2$Pred.Y1 - data2$Pred.Y0
summary(data2$Pred.TE)
```

```
       V1         
 Min.   :-3.440  
 1st Qu.:-0.664  
 Median :-0.102  
 Mean   :-0.209  
 3rd Qu.: 0.311  
 Max.   : 2.233  
```

```
# Compute initial treatment effect estimate;
data2$Pred.TE <- data2$Pred.Y1 - data2$Pred.Y0
summary(data2$Pred.TE)
```

```
       V1         
 Min.   :-3.440  
 1st Qu.:-0.664  
 Median :-0.102  
 Mean   :-0.209  
 3rd Qu.: 0.311  
 Max.   : 2.233  
```

```
# Compute initial treatment effect estimate;
plot(density(data2$Pred.TE))
```



step 2: fit treatment model

```
### fit PS model
set.seed(1017)
#covariates are variables not including A & Y;

PS.fit.SL <- SuperLearner(Y=data2$A,
                          X=covariates,
                          cvControl = list(V = 3),
                          SL.library=c("SL.glm",
                                       "SL.glmnet",
                                       "SL.xgboost"),
                          method="method.CC_nloglik",
                          family="binomial")
```

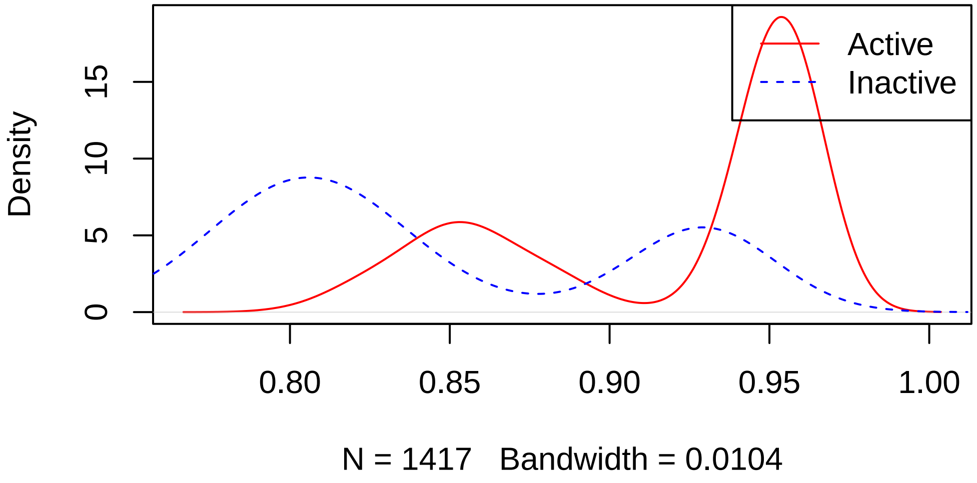
```
Loading required package: nloptr
```

```
#saveRDS(PS.fit.SL, file = "data/SL_TMLE_PS")
```

```
# predicted propensity scores
# predict A = 1 given covariates;
all.pred <- predict(PS.fit.SL, type = "response")
data2$PS.SL <- all.pred$pred

tapply(data2$PS.SL, data2$A, summary)
```

```
$`0`
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.730   0.798   0.819   0.845   0.919   0.952 

$`1`
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.798   0.866   0.947   0.918   0.956   0.973 
```

```
## check overlap
plot(density(data2$PS.SL[data2$A==1]),
     col = "red", main = "")
lines(density(data2$PS.SL[data2$A==0]),
      col = "blue", lty = 2)
legend("topright", c("Active","Inactive"),
       col = c("red", "blue"), lty=1:2)
```



step 3: determine a working model to fluctuate/update the initial estimator. Estiamte the clever covarate H

```r
## 4. clever covariate H

data2$H.A1L <- (data2$A) / data2$PS.SL
data2$H.A0L <- (1-data2$A) / (1- data2$PS.SL)
data2$H.AL <- data2$H.A1L - data2$H.A0L
summary(data2$H.AL)
```

```
       V1
 Min.   :-20.673
 1st Qu.:  1.043
 Median :  1.054
 Mean   :  0.218
 3rd Qu.:  1.146
 Max.   :  1.253
```

```r
tapply(data2$H.AL, data2$A, summary)
```

```
$`0`
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -20.67  -12.31   -5.51   -8.10   -4.95   -3.70

$`1`
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.03    1.05    1.06    1.09    1.15    1.25
```

step 4: estimate the fluctuation parameter

```r
eps_mod <- glm(Y ~ -1 + H.AL +
                 offset((init.Pred)),
               family = gaussian(link=identity),
               data = data2)
epsilon <- coef(eps_mod)
epsilon
```

```
    H.AL
-0.01867
```

step 5: update the fluctuation parameter till convergency

```
data2$Pred.Y1.update1 <- ((data2$Pred.Y1) +
                                  epsilon*data2$H.AL)
data2$Pred.Y0.update1 <- ((data2$Pred.Y0) +
                                  epsilon*data2$H.AL)
summary(data2$Pred.Y1.update1)
```

```
       V1
 Min.   :-9.085
 1st Qu.: 0.996
 Median : 2.732
 Mean   : 2.624
 3rd Qu.: 4.397
 Max.   :12.081
```

```
summary(data2$Pred.Y0.update1)
```

```
       V1
 Min.   :-8.96
 1st Qu.: 1.12
 Median : 3.19
 Mean   : 2.83
 3rd Qu.: 4.84
 Max.   :11.20
```

```
data2$update.Pred <- ifelse(data2$A==1, data2$Pred.Y1.update1, data2$Pred.Y0.update1)

eps_mod1 <- glm(Y ~ -1 + H.AL +
                  offset((update.Pred)),
                family = gaussian(link=identity),
                data = data2)
epsilon1 <- coef(eps_mod1)
epsilon1
```

```
      H.AL
-5.925e-17
```

we can stop here because the latest $\hat{e}$ is smaller than the pre-specified threshold 0.0001.

step 6: estimate ATE

```
data2$Pred.Y1.update2 <- ((data2$Pred.Y1.update1) +
                                   epsilon1*data2$H.AL)
data2$Pred.Y0.update2 <- ((data2$Pred.Y0.update1) +
                                   epsilon1*data2$H.AL)
summary(data2$Pred.Y1.update2)
```

```
       V1
 Min.   :-9.085
 1st Qu.: 0.996
 Median : 2.732
 Mean   : 2.624
 3rd Qu.: 4.397
 Max.   :12.081
```

```
summary(data2$Pred.Y0.update2)
```

```
       V1
 Min.   :-8.96
 1st Qu.: 1.12
 Median : 3.19
 Mean   : 2.83
 3rd Qu.: 4.84
 Max.   :11.20
```

```
ATE <- data2$Pred.Y1.update2 -  data2$Pred.Y0.update2
summary(ATE)
```

```
       V1
 Min.   :-3.440
 1st Qu.:-0.664
 Median :-0.102
 Mean   :-0.209
 3rd Qu.: 0.311
 Max.   : 2.233
```

```
ATE.TMLE<-mean(ATE)
```

```
## estimate confidence interval
ci.estimate <- function(data = data2){

  # transform predicted outcomes back to original scale
  EY1 <- mean(data$Pred.Y1.update2, na.rm = TRUE)
  EY0 <- mean(data$Pred.Y0.update2, na.rm = TRUE)
  # ATE efficient influence curve
  D1 <- data$A/data$PS.SL*
    (data$Y - data$Pred.Y1.update2) +
    data$Pred.Y1.update2 - EY1
  D0 <- (1 - data$A)/(1 - data$PS.SL)*
    (data$Y - data$Pred.Y0.update2) +
    data$Pred.Y0.update2 - EY0
  EIC <- D1 - D0
  # ATE variance
  n <- nrow(data)
  varHat.IC <- var(EIC, na.rm = TRUE)/n
  # ATE 95% CI
  ATE.TMLE.CI <- c(ATE.TMLE - 1.96*sqrt(varHat.IC),
                   ATE.TMLE + 1.96*sqrt(varHat.IC))
  return(ATE.TMLE.CI)
}

print(c(mean(ATE), ci.estimate(data = data2)))
```

```
[1] -0.2086 -1.1670  0.7498
```

(c) (TMLE using tmle R package) Estimate and report TMLE causal effect estimator using the tmle R package. Please use the same three algorithms used in part (b) for the outcome and treatment models.

```
set.seed(1017)
SL.library = c("SL.randomForest",
               "SL.glmnet",
               "SL.xgboost")

tmle.fit <- tmle(Y = data2$Y,
```

```r
                 A = data2$A,
                 W = covariates,
                 family = "gaussian",
                 V.Q = 3, #outcome model;
                 V.g = 3, #treatment model;
                 Q.SL.library = SL.library,
                 g.SL.library = SL.library)

#saveRDS(tmle.fit, file = "data/SL_TMLE")
```

```r
#tmle.fit <- readRDS(file = "data/SL_TMLE")
summary(tmle.fit)
```

```
 Initial estimation of Q
     Procedure: cv-SuperLearner, ensemble
     Model:
         Y ~  SL.randomForest_All + SL.glmnet_All + SL.xgboost_All

     Coefficients:
     SL.randomForest_All    0.3142
       SL.glmnet_All    0.6858
      SL.xgboost_All    0

     Cross-validated R squared :  0.1027

 Estimation of g (treatment mechanism)
     Procedure: SuperLearner, ensemble
     Model:
         A ~  SL.randomForest_All + SL.glmnet_All + SL.xgboost_All

     Coefficients:
     SL.randomForest_All    0
       SL.glmnet_All    0.6453
      SL.xgboost_All    0.3547

 Estimation of g.Z (intermediate variable assignment mechanism)
     Procedure: No intermediate variable

 Estimation of g.Delta (missingness mechanism)
     Procedure: No missingness, ensemble
 Bounds on g: (0.0172, 1)

 Bounds on g for ATT/ATC: (0.0172, 0.9828)

 Marginal Mean under Treatment (EY1)
   Parameter Estimate:  2.61
   Estimated Variance:  0.04142
              p-value:  <2e-16
    95% Conf Interval:  (2.2111, 3.009)

 Marginal Mean under Comparator (EY0)
   Parameter Estimate:  2.657
   Estimated Variance:  0.1628
              p-value:  4.552e-11
    95% Conf Interval:  (1.866, 3.4475)

 Additive Effect
   Parameter Estimate:  -0.04669
   Estimated Variance:  0.1951
              p-value:  0.9158
    95% Conf Interval:  (-0.91235, 0.81897)

 Additive Effect among the Treated
   Parameter Estimate:  1.609
   Estimated Variance:  0.2673
              p-value:  0.001851
    95% Conf Interval:  (0.59616, 2.6227)

 Additive Effect among the Controls
   Parameter Estimate:  -4.024
   Estimated Variance:  1.81
              p-value:  0.002781
    95% Conf Interval:  (-6.661, -1.3871)
```

The effect of active on weight change from the tmle R package is -0.047 with 95% CI of (-0.912, 0.819)

(d) Compare and comment on the estimation results from part (a) to (c). Which approach do you prefer and why?

The following table summarized the estimation results from PSM and tmle methods. we see that the estimated effects from both methods are in the same direction. However, the PSM method gives greater magnitude (-1.08), and the tmle estimates from (b) and (c) are similar (around 0). All the 95% CIs across 0, indicating active status effect is not statistical significant. I would prefer the TMLE approach, because the PSM estimate would be biased if the treatment model is misspecified. The TMLE is more robust as the correct specification of either outcome or treatment model would give consistent estimate. Also, we see the data has imbalanced sample size in two comparison groups (inactive group has 149 subjects and active group has 1417 subjects), some subjects in the active group have to be removed because no matched can be found from the inactive group. So the study samples become less representative of the population of interest.

method	point estimate	95%CI
(a)PSM	-1.08	(-2.94,0.78)
(b)TMLE steps	-0.21	(-1.17, 0.75)
(c)tmle pacakge	-0.05	(-0.91,0.82)

Appendix

```r
## Question 1 (b) function for standardization
library(causaldata)
data(nhefs_complete)

Q2_dat <- nhefs_complete %>%
  select(qsmk,wt82_71,age,sex,race,wt71,education,smokeintensity) %>%
  mutate(education = as.factor(ifelse(education==5,1,0)))

standard_mean_f <- function(dat = Q2_dat) {
  xvariables <- select(dat, -c(wt82_71))
  xvariables <- colnames(xvariables)

  formula <- as.formula(paste0(
    "wt82_71 ~ ",
    paste(xvariables, collapse = "+")
  ))

  Q2_model <- glm(formula,
    family = gaussian(link = "identity"),
    data = dat
  )
  ## prediciton
  Q2_pred_dat <- rbind(
    select(dat, -c(qsmk, wt82_71)),
    select(dat, -c(qsmk, wt82_71))
  ) %>%
    mutate(qsmk = c(rep(0, nrow(dat)), rep(1, nrow(dat))))

  Q2_pred <- predict(Q2_model, Q2_pred_dat) %>%
    data.frame()

  outcome_pred_dat <- cbind(Q2_pred_dat, weight_gain_pred = Q2_pred$.)

  mean_diff_table <- outcome_pred_dat %>%
    group_by(qsmk) %>%
    summarise(mean_weight_change = mean(weight_gain_pred))

  effect <- round(mean_diff_table$mean_weight_change[2] -
    mean_diff_table$mean_weight_change[1], 2)
  return(list(mean_diff_table, effect))
}
```