HAD7002 HW4

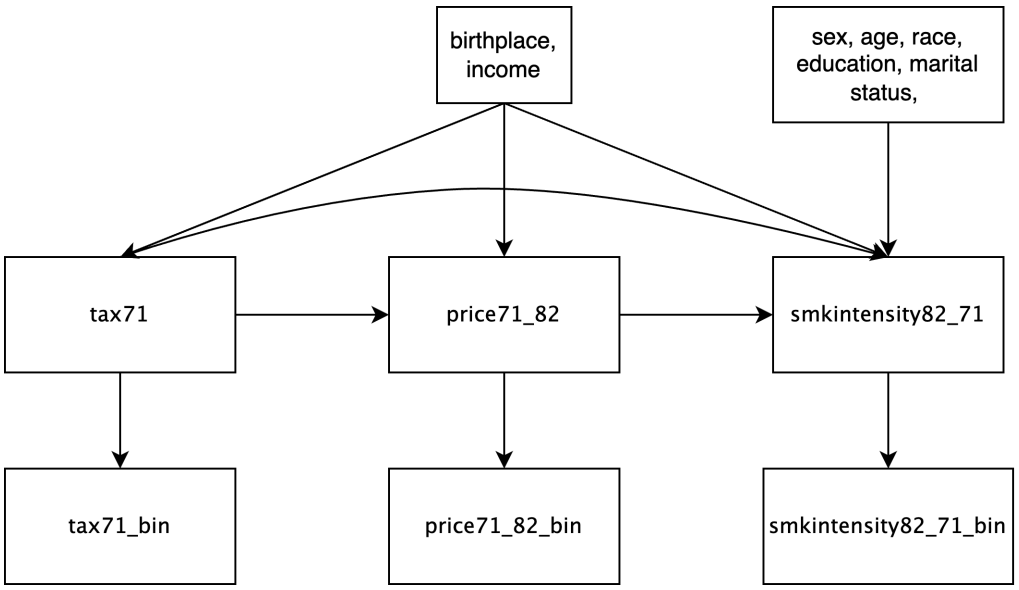
Due June 28, 2024

Question 1

(a) Propose a mediation question that can be studied based on the variables available in the NHEFS dataset. Choose a dichotomous exposure, mediator and outcome, dichotomizing continuous variables if needed, and present a DAG of the hypothesized causal mechanism. Please pay attention to the temporal ordering of the variables.

Hypothesized mediation question: An increase in tobacco taxes(exposure) will lead to an increase in tobacco prices (mediator), and thus will reduce smoking intensity(outcome)

the following DAG shows the hypothesized casual mechanism:



- **Exposure**:
  - tax71: tobacco tax in state of residence 1971 (US$2008)

  - tax71_bin: dichotomized tax71, using a threshold of 1

- **Mediator**:
  - price71_82: difference in avg tobacco price in state of residence 1971-1982 (us$2008)
  - price71_82_bin: dichotomized tax71, using a threshold of 0.3

- **Outcome**:
  - smkintensity 82_71: increase in number of cigarettes/day between 1971 and 1982.
  - smkintensity 82_71_bin: dichotomized tax71, using a threshold of 0

- **Confouding**:
  - income: Total family income in 1971 (11: <$1000, 12: $1000-1999, 13: $2000-2999, 14: $3000-3999, 15: $4000-4999, 16: $5000-5999, 17: $6000-6999, 18: $7000-9999, 19: $10000-14999, 20: $15000-19999, 21: $20000-24999, 22: $25000+)
  - birthplace: State code of birthplace.

- **Other covariates that affect the outcome**
  - sex: Sex (0: Male, 1: Female)
  - age: Age in 1971
  - race: Race in 1971 (0: White, 1: Black or other).
  - education: Level of education by 1971 (1: 8th grade or less, 2: HS dropout, 3: HS graduate, 4: College dropout, 5: College graduate or more).
  - marital: Marital status in 1971

(b) Adapting the code in mediation_formula.r (don't use the mediation package), calculate the natural direct and indirect effects, the total effect, and the proportion mediated, and 95% confidence intervals for these quantities using the bootstrap. Interpret the results.

```
## prepare dataset
## dichotomize variables

q1_dat <- nhefs_dat %>%
  mutate(smkintensity82_71_bin = case_when(
    smkintensity82_71 >=0 ~ "1",
    smkintensity82_71<0~ "0",
    is.na(smkintensity82_71)~NA

  ),
  price71_82_bin = case_when(
    price71_82 >=0.3 ~ "1",
```

```
    price71_82<0.3~ "0",
    is.na(price71_82)~NA

  ),
  tax71_bin = case_when(
   tax71 >=1 ~ "1",
    tax71<1~ "0",
    is.na(tax71)~NA
  )
  )

## subset
dat <- q1_dat %>%
  dplyr::select(tax71_bin,price71_82_bin,
         smkintensity82_71_bin,
         birthplace,income,
         sex,age,race,education,marital) %>% na.omit() %>%
  mutate(tax71_bin = as.factor(tax71_bin),
         price71_82_bin = as.factor(price71_82_bin),
         smkintensity82_71_bin = as.factor(smkintensity82_71_bin),
         marital = ifelse(marital==2,2,1)
         )

## frequency table
ftable(q1_dat$tax71_bin,q1_dat$price71_82_bin, q1_dat$smkintensity82_71_bin)
```

```
       0   1

0 0  218 208
  1   78  76
1 0   62  67
  1  380 387
```

```
ftable(q1_dat$tax71_bin,q1_dat$price71_82_bin, q1_dat$smkintensity82_71_bin)/nrow(q1_dat)
```

```
              0          1

0 0  0.13920817 0.13282248
  1  0.04980843 0.04853129
1 0  0.03959132 0.04278416
  1  0.24265645 0.24712644
```

- Outcome model:

```
# Outcome model (dichotomized outcome):
ymodel <- glm(smkintensity82_71_bin ~ tax71_bin*price71_82_bin +
               birthplace+income+
              sex+age+race+education+marital
              , family=binomial(link=logit), data=dat)
summary(ymodel)
```

```
Call:
glm(formula = smkintensity82_71_bin ~ tax71_bin * price71_82_bin +
    birthplace + income + sex + age + race + education + marital,
    family = binomial(link = logit), data = dat)

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)
(Intercept)                 0.5127222  0.5378017   0.953   0.3404
tax71_bin1                  0.0175734  0.2212014   0.079   0.9367
price71_82_bin1             0.0192923  0.1970843   0.098   0.9220
birthplace                  0.0083714  0.0040390   2.073   0.0382 *
income                     -0.0008511  0.0251034  -0.034   0.9730
sex1                        0.2052265  0.1116817   1.838   0.0661 .
age                        -0.0256193  0.0048570  -5.275 1.33e-07 ***
race1                       0.1622344  0.1680602   0.965   0.3344
education2                  0.0566067  0.1799395   0.315   0.7531
education3                  0.1931143  0.1715884   1.125   0.2604
education4                  0.1029321  0.2507582   0.410   0.6815
education5                 -0.4323372  0.2329312  -1.856   0.0634 .
marital                     0.0840555  0.1460316   0.576   0.5649
tax71_bin1:price71_82_bin1  0.0295944  0.2827066   0.105   0.9166
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1971.2  on 1421  degrees of freedom
Residual deviance: 1915.2  on 1408  degrees of freedom
AIC: 1943.2

Number of Fisher Scoring iterations: 4
```

- Mediator model

```r
# Mediator model:
zmodel <- glm(price71_82_bin ~ tax71_bin +
                birthplace + income,
  family = binomial(link = logit),
  data = dat
)
summary(zmodel)
```

```
Call:
glm(formula = price71_82_bin ~ tax71_bin + birthplace + income,
    family = binomial(link = logit), data = dat)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.138927   0.540702   2.106   0.0352 *
tax71_bin1   3.570599   0.182262  19.591   <2e-16 ***
birthplace  -0.064372   0.006396 -10.064   <2e-16 ***
income      -0.026062   0.027237  -0.957   0.3386
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1883.3  on 1421  degrees of freedom
Residual deviance: 1235.6  on 1418  degrees of freedom
AIC: 1243.6

Number of Fisher Scoring iterations: 5
```

```r
# Expected potential outcomes:

newdat11 <- dat %>% mutate(tax71_bin=1 %>% as.factor(),
                           price71_82_bin=1%>% as.factor())

predy11 <- predict(ymodel, newdata=newdat11, type='response')

newdat10 <- dat %>% mutate(tax71_bin=1 %>% as.factor(),
                           price71_82_bin=0%>% as.factor())
predy10 <- predict(ymodel, newdata=newdat10, type='response')
```

```r
newdat01 <-  dat %>% mutate(tax71_bin=0 %>% as.factor(),
                            price71_82_bin=1 %>% as.factor())
predy01 <- predict(ymodel, newdata=newdat01, type='response')

newdat00 <-  dat %>% mutate(tax71_bin=0 %>% as.factor(),
                            price71_82_bin=0 %>% as.factor())

predy00 <- predict(ymodel, newdata=newdat00, type='response')

# Expected potential mediators:

predz1 <- predict(zmodel, newdata=newdat11, type='response')
predz0 <- predict(zmodel, newdata=newdat00, type='response')
```

The total effect can be decomposed into the natural indirect effect and natural direct effect:

$$\begin{aligned} E[Y_a] - E[Y_{a^*}] &= E[Y_{aM_a}] - E[Y_{a^*M_{a^*}}] \\ &= (E[Y_{aM_a}] - E[Y_{aM_{a^*}}])(1) \\ &+ (E[Y_{aM_{a^*}}] - E[Y_{a^*M_{a^*}}])(2) \end{aligned}$$

where term (1) is the natural indirect effect and term (2) is the natural direct effect.

- So the natural indirect effect can be estimated as:

```r
# Calculate the natural indirect effect estimate:

nie <- mean(predy11 * predz1 + predy10 * (1.0 - predz1)) -
  mean(predy11 * predz0 + predy10 * (1.0 - predz0))
nie
```

```
[1] 0.007446979
```

- the natural direct effect can be estimated as:

```r
# Calculate the natural direct effect estimate:

nde <-
  mean(predy11 * predz0 + predy10 * (1.0 - predz0)) - mean(predy01 * predz0 + predy00 * (1.0
```

```
# Total effect:

te <- nde + nie
te
```

```
[1] 0.01335575
```

The mediated proportion is calculated as: NIE/TE

```
## proportion mediated

mediate_p <- (nie/te)*100

mediate_p
```

```
[1] 55.75861
```

- Bootstrapped CI:

```
# Bootstrap to get standard errors (m is the number of resamples):

m <- 1000

ndeb <- rep(NA, m)
nieb <- rep(NA, m)
teb <- rep(NA, m)
mediate_p1 <- rep(NA, m)

set.seed(1017)
for (i in 1:m) {

    bootidx <- sample(1:nrow(dat), nrow(dat), replace=TRUE)
    datb <- dat[bootidx,]

    # Outcome model:

    ymodel <- glm(smkintensity82_71_bin ~ tax71_bin*price71_82_bin +
                birthplace+income+
               sex+age+race+education+marital
               , family=binomial(link=logit), data=datb)
```

```r
# Mediator model:

zmodel <- glm(price71_82_bin ~ tax71_bin + birthplace + income,
              family=binomial(link=logit),
              data=datb)

# Expected potential outcomes:
##11
newdat11 <- datb %>% mutate(tax71_bin=1 %>% as.factor(),
                        price71_82_bin=1%>% as.factor())
predy11 <- predict(ymodel, newdata=newdat11, type='response')

##10
newdat10 <- datb %>% mutate(tax71_bin=1 %>% as.factor(),
                        price71_82_bin=0%>% as.factor())
predy10 <- predict(ymodel, newdata=newdat10, type='response')

##01
newdat01 <-  datb %>% mutate(tax71_bin=0 %>% as.factor(),
                        price71_82_bin=1 %>% as.factor())
predy01 <- predict(ymodel, newdata=newdat01, type='response')

##00
newdat00 <- datb %>% mutate(tax71_bin=0 %>% as.factor(),
                        price71_82_bin=0 %>% as.factor())
predy00 <- predict(ymodel, newdata=newdat00, type='response')

# Expected potential mediators:

predz1 <- predict(zmodel, newdata=newdat11, type='response')
predz0 <- predict(zmodel, newdata=newdat00, type='response')

# Calculate the natural direct effect estimate:
ndeb[i] <- mean(predy11 * predz0 + predy10 * (1.0 - predz0)) -
  mean(predy01 * predz0 + predy00 * (1.0 - predz0))

# Calculate the natural indirect effect estimate:

nieb[i] <- mean(predy11 * predz1 + predy10 * (1.0 - predz1)) -
  mean(predy11 * predz0 + predy10 * (1.0 - predz0))

# Total effect:
```

```
    teb[i] <- ndeb[i] + nieb[i]

    # proportion mediated
    mediate_p1[i] <- (ndeb[i]/ teb[i])*100

}
```

```
results <- rbind(
cbind(nde, nie, te, mediate_p),
colMeans(cbind( ndeb, nieb, teb,mediate_p1)),
sqrt(apply(cbind( ndeb, nieb, teb,mediate_p1), 2, var)),
apply(cbind( ndeb, nieb, teb,mediate_p1), 2, quantile, probs=0.025),
apply(cbind( ndeb, nieb, teb,mediate_p1), 2, quantile, probs=0.975)
)

rownames(results) <- c('Point estimate', 'Bootstrap mean', 'Bootstrap SE',
                       '95% CI lower bound', '95% CI upper bound')
colnames(results) <- c('nde','nie','te','mediated proportion')
round(results, 3)
```

```
                      nde    nie     te mediated proportion
Point estimate      0.006  0.007  0.013              55.759
Bootstrap mean      0.006  0.008  0.013             253.584
Bootstrap SE        0.041  0.031  0.027            3695.489
95% CI lower bound -0.072 -0.050 -0.043           -1062.069
95% CI upper bound  0.082  0.071  0.065            2235.185
```

- both the natural direct and natural indirect effects are small, and the bootstrapped confidence intervals include zero, indicates the effect of tabacco tax in 1971 on the change of smoking intensity during 1971-1982 is not statistically significant.
- the magnitude of the mediated proportion is extremely high, one explanation could be that the natural direct and indirect effects are in opposite direct, so they cancel out and the total effect(denominator of mediated proportion) is pulled close to zero.

Question 2

(a) Suppose you completed a causal analysis and obtained an observed risk ratio 1.3 with a 95%CI [1.1, 1.6]. Please conduct a sensitivity analysis using the bounding factor approach and example introduced in Slide 19 from the Week 8 lecture. The range of the two sensitivity parameters is provided in Table 1. Please fill Table 1 by calculating the bounding factors and please identify the combinations of RRAU {1.5, 2} and RRAU {1.5, 2} that lead to changes in the interpretation of the observed RR.

The bounding factor is given by:

$$BF = \frac{RR_{AU} + RR_{UY}}{RR_{AU} + RR_{UY} - 1}$$

```
# calculate bounding factor
RR_AU <- c(rep(1,3),rep(1.5,3),rep(2,3))
RR_UY <-  rep(c(1,1.5,2),3)
BF = (RR_AU*RR_UY)/(RR_AU+RR_UY-1)

BF_dat <- data.frame(RR_AU,RR_UY,BF)

BF_dat
```

```
  RR_AU RR_UY       BF
1   1.0   1.0 1.000000
2   1.0   1.5 1.000000
3   1.0   2.0 1.000000
4   1.5   1.0 1.000000
5   1.5   1.5 1.125000
6   1.5   2.0 1.200000
7   2.0   1.0 1.000000
8   2.0   1.5 1.200000
9   2.0   2.0 1.333333
```

The results are summarized in the following table:

Since $RR^{ture}_{AY|l} >= RR^{obs}_{AY|l}/BF$, so the interpretation of observed RR will change if bounding factor does not equal to 1, the following **four** combinations of RR_{AU} and RR_{UY} will lead to the change of RR interpretation:

```
BF_dat %>% filter(BF !=1)
```

bounding factor		RR_{UY}		
		1	**1.5**	**2**
RR_{AU}	**1**	**1**	**1**	**1**
	1.5	**1**	**1.125**	**1.2**
	2	**1**	**1.2**	**1.333**

Table 1: Values of the Bounding Factors

```
  RR_AU RR_UY       BF
1   1.5   1.5 1.125000
2   1.5   2.0 1.200000
3   2.0   1.5 1.200000
4   2.0   2.0 1.333333
```

(b) (Understand the impact of unmeasured confounding via simulation) Create the dataset simdat using the following function.

```r
# simulated data
sim.r <- function(samplesize = 500)
{
  set.seed(123)
  expit <- function(x){exp(x)/(1+exp(x))}
  #covariates;
  L1 <- runif(n=samplesize,0,1)
  L2 <- runif(n=samplesize,0,1)
  U <- rbinom(n=samplesize, size = 1, prob = 0.2)

  #treatment;
  Aprob <- expit(3*L1-3*L2+4*U)
  A <- rbinom(n=samplesize, size=1, prob=Aprob)

  #outcome;
  Yprob <- expit(3*A+3*L1-3*L2+4*U)
  Y <- rbinom(n = samplesize, size = 1, prob = Yprob)
  dat <- cbind(L1, L2, U, A, Y)
  dat <- data.frame(dat)
  return(dat)
}

simdat <- sim.r(samplesize = 500)
```

(i) Assume variable U is unmeasured, calculate the IPTW estimator for risk ratio (RR) with 95% bootstrap confidence interval using measured covariates L1 and L2 (hint, risk ratio is E[Y1]/E[Y0]).

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

$$\text{logit}(P(A = 1|L_1, L_2, \theta)) = \theta_0 + \theta_1 L_1 + \theta_2 L_2$$

  The following R codes are used to fit this model.

```
## fit treatment model
Q2_ps_model <- glm(A ~ L1+L2, family=binomial(link=logit), data=simdat)
```

- 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 $\frac{1}{P(A_i=1|L_i)}$ (L_i is the covariates vector for patient i, i.e., (L_{1i},L_{2i})) and weight for individuals with $A = 0$ is $\frac{1}{P(A_i=0|L_i)} = \frac{1}{1-P(A_i=1|L_i)}$ or calculate the stabilized version $\frac{P(A_i=1)}{P(A_i=1|L_i)} = \frac{P(A_i=0)}{1-P(A_i=1|L_i)}$ (we calculated the stabilized version):

```
## get predicted treatment probs
treatment_pred <- predict(Q2_ps_model, type="response")
## P(A)

PA1 = mean(simdat$A)
PA0 = 1- mean(simdat$A)

## calculate the weights
iptw_dat <- cbind(simdat, treatment_pred) %>%
  mutate(weight = ifelse(A == 1,
                         PA1 / treatment_pred,
                         PA0 / (1 - treatment_pred))
  )

## or use package
library(WeightIt)
```

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

```
library(cobalt)
```

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

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

```r
baselines <- c("L1","L2")
ps.formula <- as.formula(paste("A~",
                paste(baselines, collapse = "+")))

IPTW <- weightit(ps.formula,
                 data = iptw_dat,
                 method = "glm",
                 #using the default logistic regression;
                 stabilize = TRUE)
```

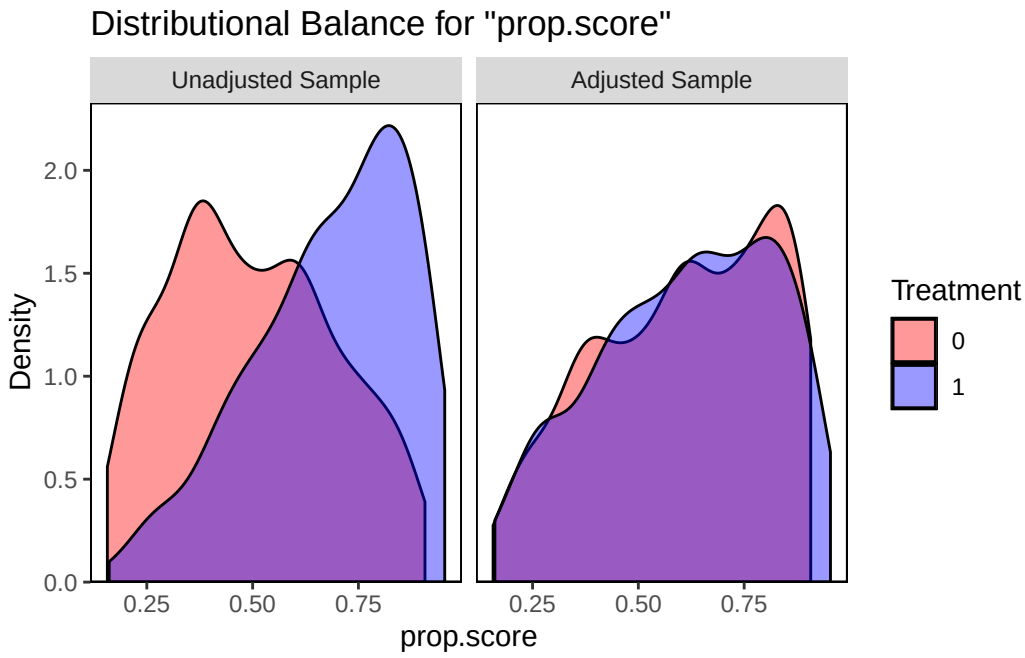
check the distribution of the propensity score between two groups:

```r
summary(IPTW)
```

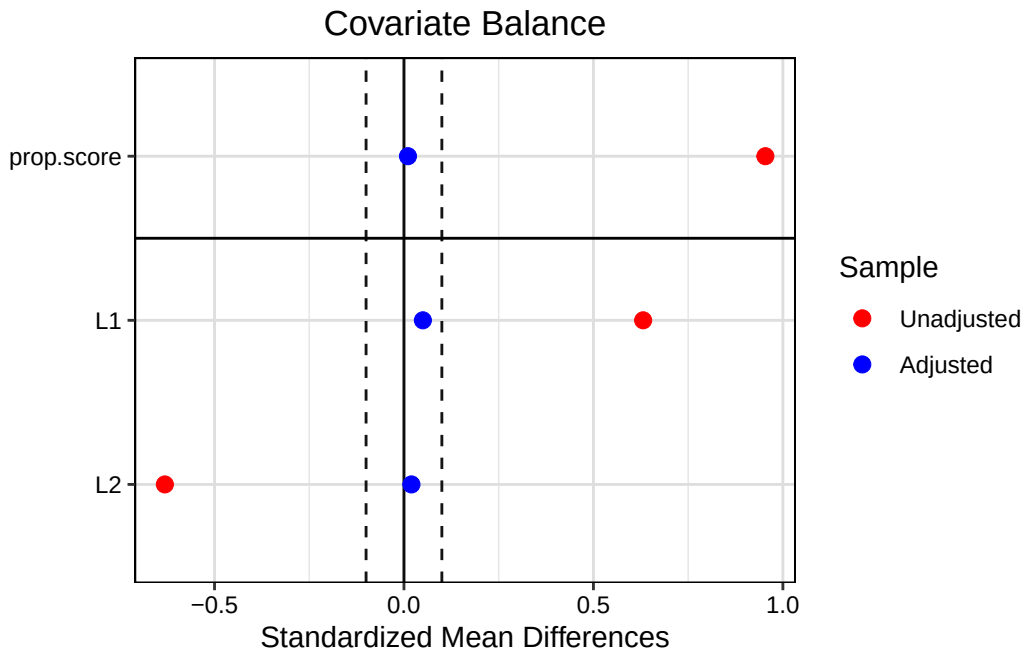
```
                Summary of weights

- Weight ranges:

           Min                                  Max
treated 0.6415  |----------------------|    3.8011
control 0.4601 |--------------------------| 4.1862

- Units with the 5 most extreme weights by group:

             41    282    395    408    234
 treated  2.483 2.6026 2.7287 3.4519 3.8011
            242     92    331    190    260
 control 2.9592  3.528 3.6958 4.0349 4.1862

- Weight statistics:

        Coef of Var   MAD Entropy # Zeros
treated       0.429 0.285   0.072       0
control       0.664 0.448   0.166       0

- Effective Sample Sizes:

           Control Treated
Unweighted  194.    306.
Weighted    134.89  258.58
```

```
bal.plot(IPTW,
         which="both",
         type = "density",
         colors = c("red","blue"))
```

```
No `var.name` was provided. Displaying balance for prop.score.
```




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




```
bal.tab(IPTW, un=TRUE, thresholds = c(m=0.1))
```

```
Balance Measures
               Type Diff.Un Diff.Adj    M.Threshold
prop.score Distance  0.9539   0.0106 Balanced, <0.1
L1          Contin.  0.6312   0.0499 Balanced, <0.1
L2          Contin. -0.6310   0.0197 Balanced, <0.1

Balance tally for mean differences
                    count
Balanced, <0.1          3
Not Balanced, >0.1      0

Variable with the greatest mean difference
 Variable Diff.Adj    M.Threshold
       L1   0.0499 Balanced, <0.1

Effective sample sizes
           Control Treated
Unadjusted  194.    306.
Adjusted    134.89  258.58
```

- Step 3: the IPTW estimator is given by: E[Y1]/E[Y0]) = E[Y1]/E[Y0]) = 0.92 / 0.452 = 1.914 with 95% bootstrapped CI: (1.653, 2.292).

```
risk_dat <- iptw_dat %>%
  group_by(A) %>%
  summarise(risk = sum(Y*weight)/ sum(weight) )


risk1 = risk_dat$risk[risk_dat$A==1] %>% round(3)
risk0 = risk_dat$risk[risk_dat$A==0]%>% round(3)

rr_est_q2 <- risk1/risk0

# rr_est_q2
```

The following codes are used to obtain the 95% bootstrapped CI:

```
set.seed(1017)
boot.est <- rep(NA, 1000)
for (i in 1:1000){

  boot.idx <- sample(1:dim(simdat)[1], size = dim(simdat)[1], replace = T)
  boot.data <- simdat[boot.idx,]

  Q2_ps_model <- glm(A ~ L1+L2, family=binomial(link=logit), data=boot.data)

  treatment_pred <- predict(Q2_ps_model, type="response")

## P(A)

PA1 = mean(boot.data$A)
PA0 = 1- mean(boot.data$A)

## calculate the weights
iptw_dat <- cbind(boot.data, treatment_pred) %>%
  mutate(weight = ifelse(A == 1,
                         PA1 / treatment_pred,
                         PA0 / (1 - treatment_pred))
  )

risk_dat <- iptw_dat %>%
  group_by(A) %>%
```

```
  summarise(risk = sum(Y*weight)/ sum(weight) )


risk1 = risk_dat$risk[risk_dat$A==1]
risk0 = risk_dat$risk[risk_dat$A==0]

rr_est <- risk1/risk0

  boot.est[i] <- rr_est

}


rr_ci_q2 <- round(quantile(boot.est, probs = c(0.025, 0.975)),3)
```

(ii) Calculate the E-value given your IPTW RR estimates and provide a statement and explanation on the obtained E-value.

The E-value is calculated by:

$$RR^{obs} + \sqrt{(RR^{obs} \times (RR^{obs} - 1)}$$

```
# E-value
Eval <- rr_est_q2+ sqrt(rr_est_q2*(rr_est_q2-1))
Eval
```

```
[1] 3.236876
```

The E-value is defined as the lower bound of the Bounding Factor, which indicates the minimum strength of association that unmeasured counfounder(s) would need to have with both the treatment and outcome to fully explain away a specific treatment-outcome association, conditional on the measured covariates. The calculated E-value is 3.237, meaning that to explain away the association, a hypothetical confounder would need be associated with a 3.237 higher use of the treatment A and a 3.237 greater risk of Y.

To see how likely this is, we can compare the E-value with the RRs of known confounders in relation to the outcome. E-value which is significantly higher than the RRs of known confounders may indicate the robustness.

(iii) Recalculate the IPTW RR estimator with 95% bootstrap confidence interval using all simulated covariates L1, L2, and U. Compare and comment on the new RR estimates to i) the RR estimates without U and ii) the calculated E-value.

```
rm(list=ls())
```

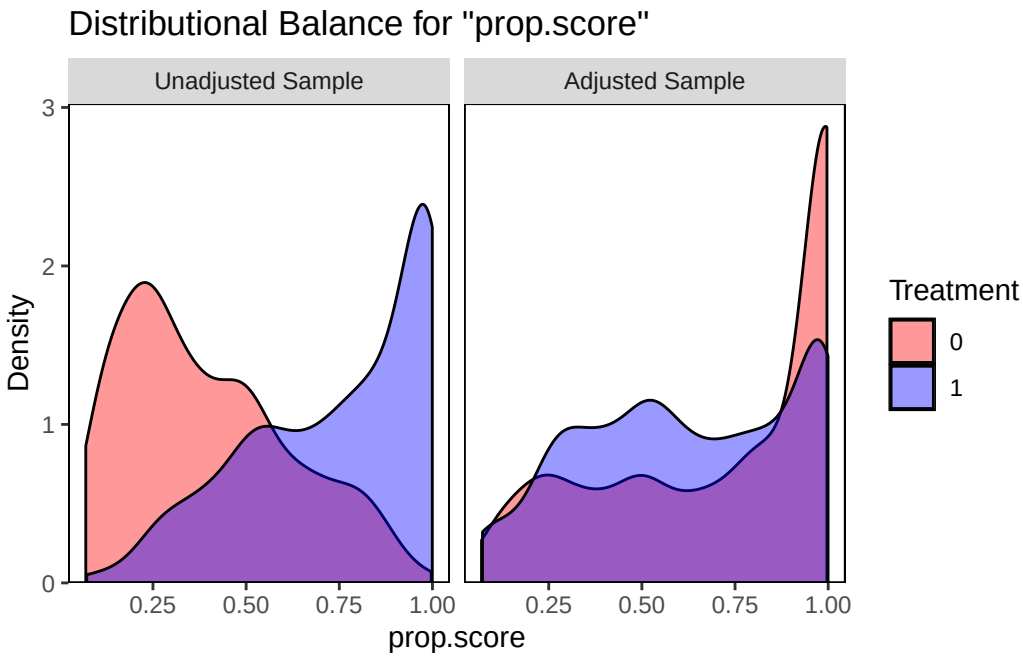
```
# simulated data
sim.r <- function(samplesize = 500)
{
  set.seed(123)
  expit <- function(x){exp(x)/(1+exp(x))}
  #covariates;
  L1 <- runif(n=samplesize,0,1)
  L2 <- runif(n=samplesize,0,1)
  U <- rbinom(n=samplesize, size = 1, prob = 0.2)

  #treatment;
  Aprob <- expit(3*L1-3*L2+4*U)
  A <- rbinom(n=samplesize, size=1, prob=Aprob)

  #outcome;
  Yprob <- expit(3*A+3*L1-3*L2+4*U)
  Y <- rbinom(n = samplesize, size = 1, prob = Yprob)
  dat <- cbind(L1, L2, U, A, Y)
  dat <- data.frame(dat)
  return(dat)
}

simdat <- sim.r(samplesize = 500)
```

```
## fit treatment model
## use package
library(WeightIt)
library(cobalt)
baselines <- c("L1","L2","U")
ps.formula <- as.formula(paste("A~",
                paste(baselines, collapse = "+")))

IPTW <- weightit(ps.formula,
                 data = simdat,
                 method = "glm",
                 #using the default logistic regression;
```

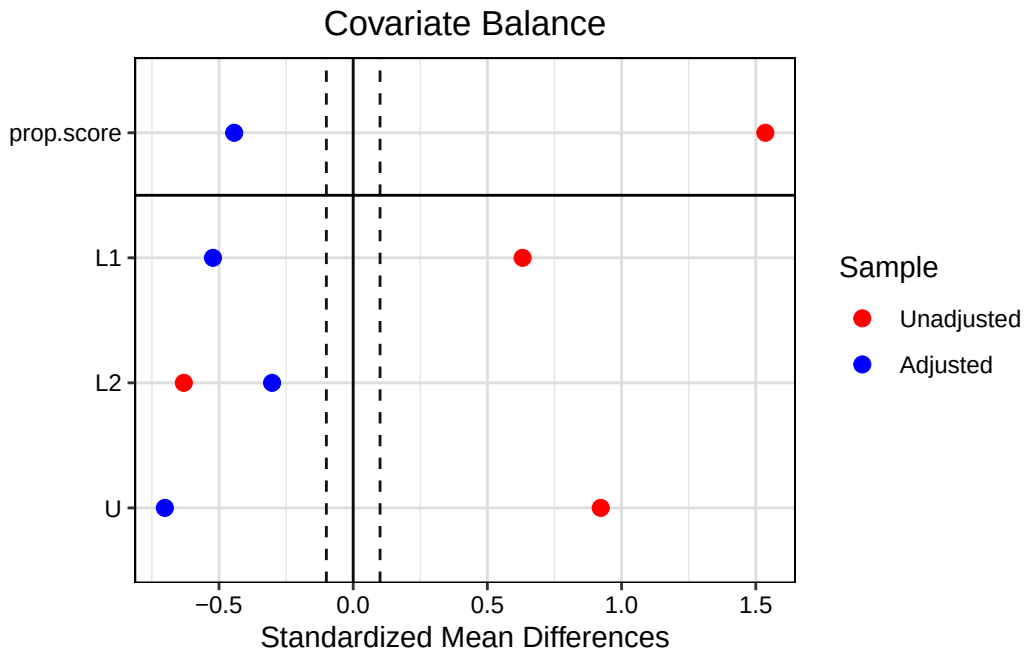
```
                stabilize = TRUE)

summary(IPTW)
```

```
                Summary of weights

- Weight ranges:

           Min                                   Max
treated 0.6123 ||                             8.4512
control 0.4171 |---------------------------| 117.9447

- Units with the 5 most extreme weights by group:

            392    469    379    395      234
 treated 2.8228 2.9569 3.8344  5.386   8.4512
             92    331    190    260      321
 control 3.2058 3.3851 3.7905 3.9576 117.9447

- Weight statistics:

        Coef of Var   MAD Entropy # Zeros
treated       0.697 0.389   0.147       0
control       5.937 0.971   1.681       0

- Effective Sample Sizes:

           Control Treated
Unweighted  194.    306.
Weighted      5.38  206.15
```

```
bal.plot(IPTW,
         which="both",
         type = "density",
         colors = c("red","blue"))
```




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




```
bal.tab(IPTW, un=TRUE, thresholds = c(m=0.2))
```

```
Balance Measures
               Type Diff.Un Diff.Adj        M.Threshold
prop.score Distance  1.5358  -0.4433
L1          Contin.  0.6312  -0.5224 Not Balanced, >0.2
L2          Contin. -0.6310  -0.3021 Not Balanced, >0.2
U            Binary  0.3053  -0.2323 Not Balanced, >0.2

Balance tally for mean differences
                   count
Balanced, <0.2         0
Not Balanced, >0.2     3

Variable with the greatest mean difference
 Variable Diff.Adj        M.Threshold
       L1  -0.5224 Not Balanced, >0.2

Effective sample sizes
           Control Treated
Unadjusted  194.    306.
Adjusted      5.38  206.15
```

We trimed the weights as we see extreme values.

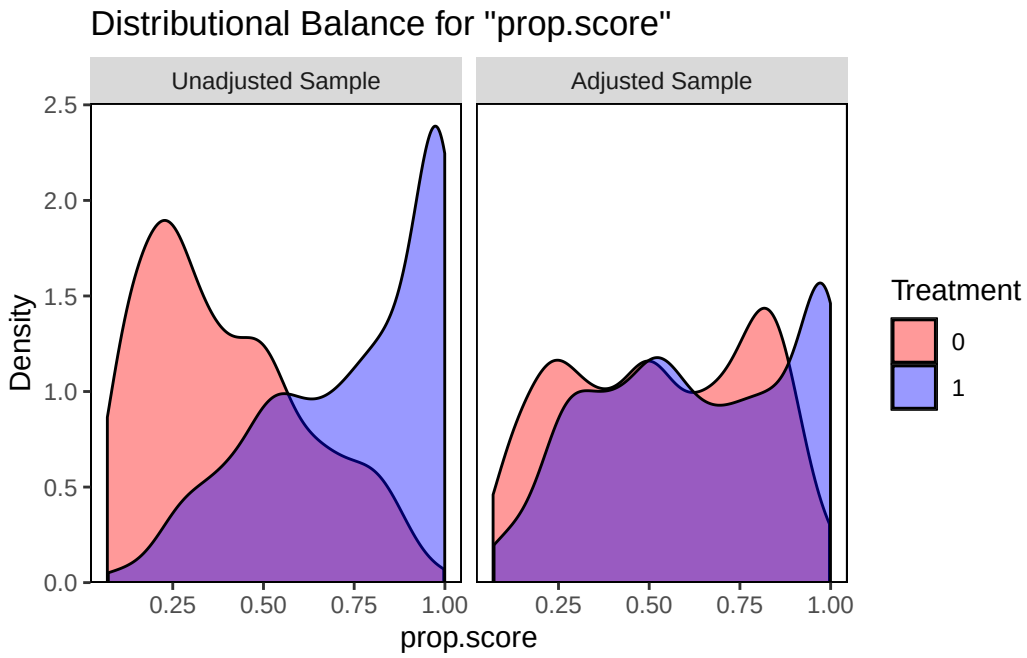
```
IPTW.trim <- trim(IPTW, at = .99)
```

```
Trimming weights to 99%.
```

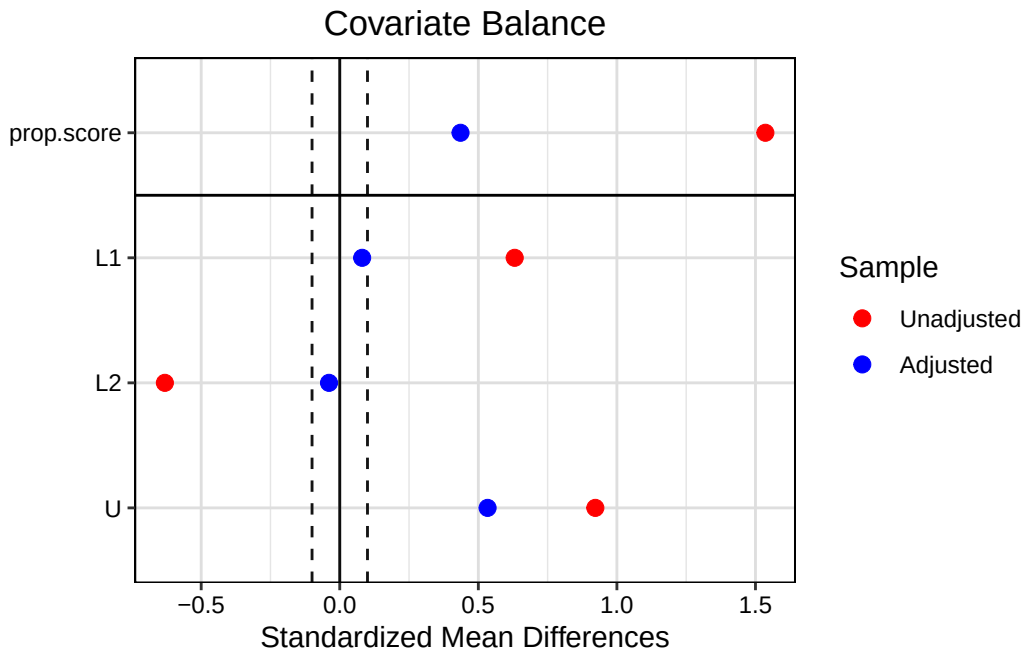
check the distribution of the propensity score between two groups after trimming, the distributions are more balanced after trimming.

```
bal.plot(IPTW.trim,
         which="both",
         type = "density",
         colors = c("red","blue"))
```

```
No `var.name` was provided. Displaying balance for prop.score.
```




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




```
bal.tab(IPTW.trim, un=TRUE, thresholds = c(m=0.2))
```

```
Balance Measures
               Type Diff.Un Diff.Adj    M.Threshold
prop.score Distance  1.5358   0.4358
L1          Contin.  0.6312   0.0808 Balanced, <0.2
L2          Contin. -0.6310  -0.0388 Balanced, <0.2
U            Binary  0.3053   0.1766 Balanced, <0.2

Balance tally for mean differences
                   count
Balanced, <0.2         3
Not Balanced, >0.2     0

Variable with the greatest mean difference
 Variable Diff.Adj    M.Threshold
        U   0.1766 Balanced, <0.2

Effective sample sizes
           Control Treated
Unadjusted  194.    306.
Adjusted    124.57  236.64
```

```r
## fit treatment model
Q2_ps_model <- glm(A ~ L1+L2+U, family=binomial(link=logit), data=simdat)

## get predicted treatment probs
treatment_pred <- predict(Q2_ps_model, type="response")
## P(A)

PA1 = mean(simdat$A)
PA0 = 1- mean(simdat$A)

## calculate the weights
iptw_dat <- cbind(simdat, treatment_pred) %>%
  mutate(weight = ifelse(A == 1,
                         PA1 / treatment_pred,
                         PA0 / (1 - treatment_pred))
  )


iptw_dat <- simdat %>% mutate(weight_trim = IPTW.trim$weights,  weight_nottrim = iptw_dat$wei

risk_dat <- iptw_dat %>%
  group_by(A) %>%
  summarise(risk_trimed = sum(Y*weight_trim)/ sum(weight_trim),
            risk_nottrimed = sum(Y*weight_nottrim)/ sum(weight_nottrim),
            )


risk_trim1 = risk_dat$risk_trimed[risk_dat$A==1] %>% round(3)
risk_trim0 = risk_dat$risk_trimed[risk_dat$A==0]%>% round(3)

rr_est1_trimed <- risk_trim1/risk_trim0

Eval_trim <- rr_est1_trimed+ sqrt(rr_est1_trimed*(rr_est1_trimed-1))


risk_nottrim1 = risk_dat$risk_nottrimed[risk_dat$A==1] %>% round(3)
risk_nottrim0 = risk_dat$risk_nottrimed[risk_dat$A==0]%>% round(3)

rr_est1_nottrimed <- risk_nottrim1/risk_nottrim0

Eval_nottrim <- rr_est1_nottrimed+ sqrt(rr_est1_nottrimed*(rr_est1_nottrimed-1))
## rr calculated from trimmed weights
```

```
rr_est1_trimed
```

```
[1] 1.837675
```

```
## rr calculated from untrimmed weights
rr_est1_nottrimed
```

```
[1] 1.268741
```

Compute bootstrapped CI:

```
## not trimed CI
set.seed(1017)
boot.est_nottrim <- rep(NA, 1000)
for (i in 1:1000){

  boot.idx <- sample(1:dim(simdat)[1], size = dim(simdat)[1], replace = T)
  boot.data <- simdat[boot.idx,]

  Q2_ps_model <- glm(A ~ L1+L2+U, family=binomial(link=logit), data=boot.data)

  treatment_pred <- predict(Q2_ps_model, type="response")

  ## P(A)

  PA1 = mean(boot.data$A)
  PA0 = 1- mean(boot.data$A)

  ## calculate the weights
  iptw_dat <- cbind(boot.data, treatment_pred) %>%
    mutate(weight = ifelse(A == 1,
                           PA1 / treatment_pred,
                           PA0 / (1 - treatment_pred))
    )

  risk_dat <- iptw_dat %>%
    group_by(A) %>%
    summarise(risk = sum(Y*weight)/ sum(weight) )
```

```
  risk1 = risk_dat$risk[risk_dat$A==1]
  risk0 = risk_dat$risk[risk_dat$A==0]

  rr_est <- risk1/risk0

  boot.est_nottrim[i] <- rr_est

}

rr_ci_q2_nottrim<-round(quantile(boot.est_nottrim,probs =c(0.025,0.975)),3)
```

```
## trimed CI
set.seed(1017)
boot.est_trim <- rep(NA, 1000)

for (i in 1:1000){

  boot.idx <- sample(1:dim(simdat)[1], size = dim(simdat)[1], replace = T)
  boot.data <- simdat[boot.idx,]

  baselines <- c("L1","L2","U")
ps.formula <- as.formula(paste("A~",
                paste(baselines, collapse = "+")))

IPTW <- weightit(ps.formula,
                 data = boot.data,
                 method = "glm",
                 #using the default logistic regression;
                 stabilize = TRUE)

IPTW.trim <- trim(IPTW, at = .99)
iptw_dat <- boot.data %>%
  mutate(weight_trim = IPTW.trim$weights,
         weight_nottrim = IPTW$weights)
risk_dat <- iptw_dat %>%
  group_by(A) %>%
  summarise(risk_trimed = sum(Y*weight_trim)/ sum(weight_trim),
            risk_nottrimed = sum(Y*weight_nottrim)/ sum(weight_nottrim),
            )


risk_trim1 = risk_dat$risk_trimed[risk_dat$A==1] %>% round(3)
```

```r
risk_trim0 = risk_dat$risk_trimed[risk_dat$A==0]%>% round(3)

rr_est1_trimed <- risk_trim1/risk_trim0



  boot.est_trim[i] <- rr_est1_trimed

}


rr_ci_q2_trim <- round(quantile(boot.est_trim, probs = c(0.025, 0.975)),3)
```

```r
## fit treatment model
Q2_ps_model <- glm(A ~ L1+L2, family=binomial(link=logit), data=simdat)

## get predicted treatment probs
treatment_pred <- predict(Q2_ps_model, type="response")
## P(A)

PA1 = mean(simdat$A)
PA0 = 1- mean(simdat$A)

## calculate the weights
iptw_dat <- cbind(simdat, treatment_pred) %>%
  mutate(weight = ifelse(A == 1,
                         PA1 / treatment_pred,
                         PA0 / (1 - treatment_pred))
  )

## or use package
library(WeightIt)
library(cobalt)
baselines <- c("L1","L2")
ps.formula <- as.formula(paste("A~",
                               paste(baselines, collapse = "+")))

IPTW <- weightit(ps.formula,
                 data = iptw_dat,
                 method = "glm",
                 #using the default logistic regression;
                 stabilize = TRUE)
```

```r
iptw_dat <- simdat %>% mutate(weight = IPTW$weights) ### use package
iptw_dat <- cbind(simdat, treatment_pred) %>%          ## hand calclulation
  mutate(weight = ifelse(A == 1,
                         PA1 / treatment_pred,
                         PA0 / (1 - treatment_pred))
  )

risk_dat <- iptw_dat %>%
  group_by(A) %>%
  summarise(risk = sum(Y*weight)/ sum(weight) )


risk1 = risk_dat$risk[risk_dat$A==1] %>% round(3)
risk0 = risk_dat$risk[risk_dat$A==0]%>% round(3)

rr_est_q2 <- risk1/risk0

set.seed(1017)
boot.est <- rep(NA, 1000)
for (i in 1:1000){

  boot.idx <- sample(1:dim(simdat)[1], size = dim(simdat)[1], replace = T)
  boot.data <- simdat[boot.idx,]

  Q2_ps_model <- glm(A ~ L1+L2, family=binomial(link=logit), data=boot.data)

  treatment_pred <- predict(Q2_ps_model, type="response")

  ## P(A)

  PA1 = mean(boot.data$A)
  PA0 = 1- mean(boot.data$A)

  ## calculate the weights
  iptw_dat <- cbind(boot.data, treatment_pred) %>%
    mutate(weight = ifelse(A == 1,
                           PA1 / treatment_pred,
                           PA0 / (1 - treatment_pred))
    )

  risk_dat <- iptw_dat %>%
    group_by(A) %>%
```

```
    summarise(risk = sum(Y*weight)/ sum(weight) )


  risk1 = risk_dat$risk[risk_dat$A==1]
  risk0 = risk_dat$risk[risk_dat$A==0]

  rr_est <- risk1/risk0

  boot.est[i] <- rr_est

}


rr_ci_q2 <- round(quantile(boot.est, probs = c(0.025, 0.975)),3)

# E-value
Eval <- rr_est_q2+ sqrt(rr_est_q2*(rr_est_q2-1))
```

The results are summarized in the following table

```
Q2_summary <- data.frame(
  model = c("model1","model2","model3"),
  covariates = c("L1,L2", "L1,L2,U(trimed weight)","L1,L2,U(not trimed weight)"),
  RR = c(round(rr_est_q2,3), round(rr_est1_trimed,3),round(rr_est1_nottrimed,3)),
  `boot_95%_CI` = c(
    paste0("[", rr_ci_q2[1], ",", rr_ci_q2[2], "]"),
    paste0("[", rr_ci_q2_trim[1], ",", rr_ci_q2_trim[2], "]"),
      paste0("[", rr_ci_q2_nottrim[1], ",", rr_ci_q2_nottrim[2], "]")
  ),
  Evalue = c(Eval,Eval_trim,Eval_nottrim)
)
Q2_summary
```

```
   model                 covariates    RR   boot_95._CI   Evalue
1 model1                      L1,L2 1.914 [1.653,2.292] 3.236876
2 model2     L1,L2,U(trimed weight) 1.859 [1.561,2.198] 3.078391
3 model3 L1,L2,U(not trimed weight) 1.269 [1.134,2.086] 1.852662
```

From the results, we see that after including U, the RR estimate and E-value gets smaller as U is a coufounder, part of the effect of A on Y is due to U. Model 3 should not be used as the covariates are not balanced in two comparison groups, thus the estimate is not causal effect.

E-value: model 2 has an E-value of 3.078, model 1 has a slightly higher E-value (3.236). However, we know from the data generation mechanism, U is a confounder, the E-value of model 1 suggests that model 1 is robust to unmeasured confounding even though U is not included in model 1.

Appendix

Use `WeightIt` package to calculate the weights in Q2 and verify the hand calculation is correct.

```
## hand calculation
Q2_ps_model <- glm(A ~ L1+L2, family=binomial(link=logit), data=simdat)
treatment_pred <- predict(Q2_ps_model, type="response")

## P(A)

PA1 = mean(simdat$A)
PA0 = 1- mean(simdat$A)

## calculate the weights
iptw_dat <- cbind(simdat, treatment_pred) %>%
  mutate(weight_hand = ifelse(A == 1,
                          PA1 / treatment_pred,
                          PA0 / (1 - treatment_pred))
  )


## use package
library(WeightIt)
baselines <- c("L1","L2")
ps.formula <- as.formula(paste("A~",
                paste(baselines, collapse = "+")))

IPTW <- weightit(ps.formula,
                 data = iptw_dat,
                 method = "glm",
                 #using the default logistic regression;
                 stabilize = TRUE)


iptw_dat_check <- iptw_dat %>%
  mutate(weight_pkg = IPTW$weight)
```

```r
# weight_hand col and weight_pkg  match
head(iptw_dat_check)
```

```
         L1         L2 U A Y treatment_pred weight_hand weight_pkg
1 0.2875775 0.35360608 0 1 1      0.5984291   1.0226776  1.0226776
2 0.7883051 0.36644144 0 0 1      0.8401904   2.4278894  2.4278894
3 0.4089769 0.28710013 0 1 1      0.7073127   0.8652467  0.8652467
4 0.8830174 0.07997291 1 1 1      0.9331355   0.6558533  0.6558533
5 0.9404673 0.36545427 1 1 1      0.8864758   0.6903742  0.6903742
6 0.0455565 0.17801381 0 0 1      0.5553092   0.8725163  0.8725163
```