# HAD7002 HW3

**Due June 12, 2024**

## Question 1

The file vitamin.csv contains numbers from a randomized experiment carried out in a South-east Asian country. Here infants were randomly assigned to receive a vitamin supplement (intervention) or placebo (control). The outcome was all-cause infant mortality.

```
## read dataset
vitamin <- read.csv("~/DLSPH/CHL7002/Assignments/HW3/vitamin.csv")
```

### (a) Suppose that we have the added information that the supplement was only available to those assigned to the intervention arm. What can we say about the numbers of "defiers", "always takers", "compliers" and "never-takers"?

Since the supplement was only available to those assigned to the intervention arm, so the potential outcome $A_0$ cannot take the value of 1, so there are no defiers and no always-takers. and since these are potential treatment variables, and we cannot know the number of never-takers and compliers.(we cannot assign a patient to treatment and control simultaneously and observe whether he/she adheres to the treatment assignment)

### (b) Calculate the intention-to-treat (ITT) effect estimate as a mortality risk difference, and a 95% confidence interval for this. What are the assumptions required for identification of the ITT effect? What can you say about the validity of these assumptions?

- ITT estimate: -0.0026 with 95% bootstrap CI of (-0.0044, $-9 \times 10^{-4}$)

This is an RCT setting, so (i)exchangeability/(ii)positivity/(iii)consistency assumptions are expected by design. Additional assumption that needs to get the ITT effect is:
- Assumption: (iv) Non-informative censoring: Since the ITT effect includes all individuals

based on their initial treatment assignment, if some individuals do not complete the follow-up, the estimate is biased. So we need the non-informative censoring assumption to ensure there is no selection bias due to loss to follow up. (or we need statistical methods to account for censoring. e.g., IPCW)

- comment on the validity of the assumptions:
  - Assumption (i) exchangeability is expected as the random assignment is used
  - Assumption (ii) positivity is met as the probability of receiving the treatment/control is greater than 0 (by design)
  - Assumption (iii) consisitency is satisfied if investigators follow the trail protocol
  - Non-informative censoring is not testable, could be violated.

```
ITT_dat<-vitamin%>%group_by(assigned_treatment)%>%summarise(risk =mean(death))
## point estimate risk difference
paste0("ITT effect estimate:",round(ITT_dat$risk[2]-ITT_dat$risk[1],4))
```

```
[1] "ITT effect estimate:-0.0026"
```

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

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

boot_data_output<-boot.data%>%group_by(assigned_treatment)%>%summarise(risk =mean(death))

## risk difference
  boot.est[i] <- boot_data_output$risk[2] - boot_data_output$risk[1]

}

#"95% confidence interval for ITT effect estimate:"
round(quantile(boot.est, probs = c(0.025, 0.975)),4)
```

```
   2.5%   97.5%
-0.0044 -0.0009
```

**(c) Calculate the IV estimate for the causal effect of received treatment on mortality, and a 95% confidence interval for this. How do you interpret this result? What are the assumptions required for identification of the IV estimand? What can you say about the validity of these assumptions?**

- Effect estimate: -0.003(95% CI: -0.006, -0.001)(codes provided in the following).
- Interpretation: the estimated average causal effect is the average causal effect in the complier population, as (i)there are no defiers or no always-takers, (ii)and never-takers do not contribute to the causal effect estimate since their treatment status always takes values of zero. So in the complier population, the risk difference is -0.004 <0, so the supplement has protective effect on the all-cause mortality risk in infants. The all-cause mortality risk in the treatment arm is 0.4% lower than that in the control arm. The 95% bootstrapped CI is [-0.006, -0.001], indicating the risk difference is of statistical significance.
- Assumptions required for identification of the IV estimand:

  IV could be causal or non causal/association;

  – i) Relevance: instrument must be associated with the treatment variable;

  wrong: no direct causal effect

  – ii) Exclusion restriction: there is no causal effect of the instrument on the outcome variable ($Y_{0a} = Y_{1a} = Y_a$);

  – iii) Exchangeability: the association between the instrument and outcome variable is not confounded.

- Validity of the assumptions:

  – i) the relevance assumption is satisfied as the supplement was only available to those assigned to the intervention arm. so $P(A = 1|Z = 0) = 0$ and $ P(A=1|Z=1)=0.8$ is always greater than $P(A = 1|Z = 0) = 0$.

  – ii) exclusion restriction is not violated as the treatment assignment is randomly assigned to infants, both the investigators and participants (infants) do not know whether the intervention received is the intervention or control arm.

  – iii) since this is an RCT, exchangeability is satisfied by randomized assignment.

```r
numerator <- mean(vitamin$death[vitamin$assigned_treatment==1]) -
  mean(vitamin$death[vitamin$assigned_treatment==0])
denominator <- mean(vitamin$received_treatment[vitamin$assigned_treatment==1]) -
  mean(vitamin$received_treatment[vitamin$assigned_treatment==0] )

paste("IV point estimate:", round(numerator/denominator,3) )
```

```
[1] "IV point estimate: -0.003"
```

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

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

numerator <- mean(
  boot.data$death[boot.data$assigned_treatment==1]) -
  mean(boot.data$death[boot.data$assigned_treatment==0])

denominator <- mean(
  boot.data$received_treatment[boot.data$assigned_treatment==1]) -
  mean(boot.data$received_treatment[boot.data$assigned_treatment==0] )
##  IV estimator
  boot.est[i] <- numerator/denominator

}

#IV 95% CI
round(quantile(boot.est, probs = c(0.025, 0.975)),3)
```

```
  2.5%  97.5%
-0.006 -0.001
```

## Question 2

Note: Observations with missing tax71 values are removed from the analysis. The working data contains 1476 subjects.

```r
## load data
library(causaldata)
Q2_dat <- nhefs_complete %>%
  select(qsmk,wt82_71,tax71) %>%
  mutate(tax71 = case_when(tax71 > 1.25 ~ "1",
                           is.na(tax71) ~ NA,
                           TRUE ~ "0")
         ) %>% na.omit()
# nhefs_codebook
```

**(a) We want to use an instrumental variable to estimate the effect of smoking cessation of weight gain. Instead of price of cigarettes, we will use tobacco tax in the state of residence in 1971 as the instrumental variable, which we dichotomize at $1.25. What can you say about the strength of this instrument? How can you quantify this?**

The instrumental variable method based on four assumptions, among which the relevance assumption **can be checked** from the data, remaining untestable assumptions need to be justified using domain knowledge. the following is done to test the relevance assumption.

```r
# P(A=1|Z=1)
paste("P(A=1|Z=1) estimate:", round(Q2_dat$qsmk[Q2_dat$tax71==1] %>% mean(),3) )
```

```
[1] "P(A=1|Z=1) estimate: 0.265"
```

```r
# P(A=1|Z=0)

paste("P(A=1|Z=0) estimate:", round(Q2_dat$qsmk[Q2_dat$tax71==0] %>% mean(),3) )
```

```
[1] "P(A=1|Z=0) estimate: 0.253"
```

```r
## P(A=1|Z=1) - P(A=1|Z=0)
round(mean(Q2_dat$qsmk[Q2_dat$tax71==1]) - mean(Q2_dat$qsmk[Q2_dat$tax71==0]),3)
```

```
[1] 0.012
```

The probability of quitting smoking is 0.265 among tax71=1 and 0.253 among these with tax71=1.the difference gives $P(\text{qsmk} = 1|\text{tax71} = 1) - P(\text{qsmk} = 1|\text{tax71} = 0) = 0.012$. The difference is greater than 0. However, the magnitude of the difference is small, and tax71 may be a weak IV.

**(b) Use the ratio-type IV estimator to estimate the effect of smoking cessation of weight gain, and use a method of your choice to obtain a 95% confidence interval for this. How do you interpret the result?**

The ratio-type IV estimator is written as:

$$\frac{E(Y|Z=1) - E(Y|Z=0)}{E(A|Z=1) - E(A|Z=0)}$$

where:
- Z is the instrumental variable, `tax71`
- A is the treatment received, smoking cessation `qsmk`
we can calculate these quantities empirically(or fit saturated model) from the data:

```
## E(Y|Z=1) estimate
paste("E(Y|Z=1) estimate:", round(Q2_dat$wt82_71[Q2_dat$tax71==1] %>% mean(),3) )
```

```
[1] "E(Y|Z=1) estimate: 2.756"
```

```
## E(Y|Z=0) estimate
paste("E(Y|Z=0) estimate:", round(Q2_dat$wt82_71[Q2_dat$tax71==0] %>% mean(),3) )
```

```
[1] "E(Y|Z=0) estimate: 2.659"
```

```
## E(A|Z=1) estimate
paste("E(A|Z=1) estimate:", round(Q2_dat$qsmk[Q2_dat$tax71==1] %>% mean(),3) )
```

```
[1] "E(A|Z=1) estimate: 0.265"
```

```
## E(A|Z=0) estimate
paste("E(A|Z=0) estimate:", round(Q2_dat$qsmk[Q2_dat$tax71==0] %>% mean(),3) )
```

```
[1] "E(A|Z=0) estimate: 0.253"
```

```
## ratio-type IV estimator estimate

numerator <- mean(Q2_dat$wt82_71[Q2_dat$tax71==1]) - mean(Q2_dat$wt82_71[Q2_dat$tax71==0])
denominator <- mean(Q2_dat$qsmk[Q2_dat$tax71==1]) - mean(Q2_dat$qsmk[Q2_dat$tax71==0] )

ratio_type_iv <- round(numerator/denominator,3)
paste("ratio-type IV estimate:", round(numerator/denominator,3) )
```

```
[1] "ratio-type IV estimate: 8.367"
```

```
## 95% CI of rati-type IV estimate:
set.seed(1017)
boot.est <- rep(NA, 2000)
for (i in 1:1000){

  boot.idx <- sample(1:dim(Q2_dat)[1], size = dim(Q2_dat)[1], replace = T)
  boot.data <- Q2_dat[boot.idx,]

numerator <- mean(boot.data$wt82_71[boot.data$tax71==1]) - mean(boot.data$wt82_71[boot.data$t
denominator <- mean(boot.data$qsmk[boot.data$tax71==1]) - mean(boot.data$qsmk[boot.data$tax7
##  ratio type IV estimator
  boot.est[i] <- numerator/denominator

}

quantile(na.omit(boot.est), probs = c(0.025, 0.975))
```

```
     2.5%     97.5%
-195.5034  140.8509
```

The ratio-type IV estimator gives an estimate of the mean difference of 8.367 (bootstrapped 95% CI: [-195.503, 140.851]). Individuals who quitting smoking is expected to gain 8.367 units weight. However, because tax71 is a weak IV, the confidence interval is wide (large variation).

**(c) Verify that you can replicate the result in Q2(b) using the 2-stage least squares method.**

First, we need to fit the treatment model (specified as follows) and get the predicted value:

$$logit(P(\text{qsmk} = 1)) = \alpha_0 + \alpha_1 \text{tax71}$$

```
treat_fit <- glm("qsmk~tax71",family="binomial", data=Q2_dat)
Q2_dat$pred_treat = predict(treat_fit, type="response")
```

then we need to fit the outcome model:

$$E(wt82\_71 \mid tax71) = \beta_0 + \beta_1 \hat{E}(qsmk \mid tax71)$$

```r
outcome_fit <- glm("wt82_71~pred_treat",family=gaussian(link = 'identity'), data=Q2_dat)
summary(outcome_fit)
```

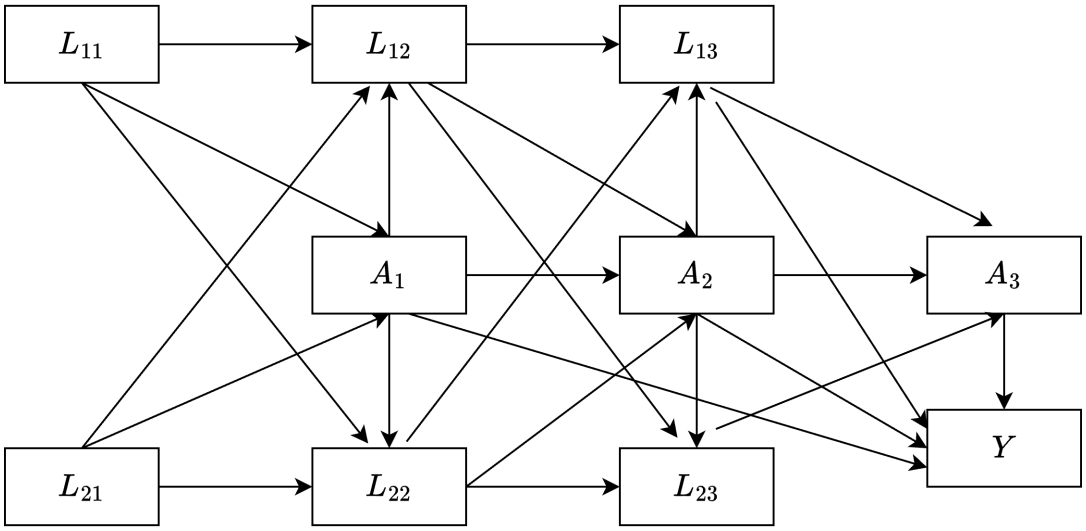
```
Call:
glm(formula = "wt82_71~pred_treat", family = gaussian(link = "identity"),
    data = Q2_dat)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.539     10.698    0.05    0.960
pred_treat     8.367     41.767    0.20    0.841

(Dispersion parameter for gaussian family taken to be 63.05915)

    Null deviance: 92952  on 1475  degrees of freedom
Residual deviance: 92949  on 1474  degrees of freedom
AIC: 10309

Number of Fisher Scoring iterations: 2
```

the two-stage least square estimator is the $\hat{\beta}_1 = 8.367$, which is equivalent to the ratio-type estimate in (b)

## Question 3

### (a)

**(i) Draw the longitudinal causal DAG following the simulation code.**




**(ii) Report the frequency table of the treatment sequences across the three visits using the simulated observational data**

```
simdat1 <- simdat %>%
  mutate(A1 = ifelse(A1==0,"A1=0","A1=1"),
         A2 = ifelse(A2==0,"A2=0","A2=1"),
         A3 = ifelse(A3==0,"A3=0","A3=1")
    )
# frequency table of the treatment sequences
ftable(simdat1$A1, simdat1$A2,simdat1$A3)
```

```
            A3=0 A3=1

A1=0 A2=0     45   57
     A2=1     42   76
A1=1 A2=0     43   55
     A2=1     52  130
```

```
## percentage
ftable(simdat1$A1, simdat1$A2,simdat1$A3)/nrow(simdat1)
```

```
             A3=0  A3=1

A1=0 A2=0  0.090 0.114
     A2=1  0.084 0.152
A1=1 A2=0  0.086 0.110
     A2=1  0.104 0.260
```

**(b)(MSMs) Based on the simulated code for the end-of-study outcome, please conduct a marginal structural models analysis using stabilized weights to estimate the expected absolute risk difference between always treated vs never treated on the outcome.**

so we are interested in

$$E(Y|A_1 = 1, A_2 = 1, A_3 = 1) - E(Y|A_1 = 0, A_2 = 0, A_3 = 0)$$

**(i) Explain and specify your choice of the marginal outcome model and the time-varying treatment assignment model.**

The causal DAG shows $A_1$, $A_2$ and $A_3$ have effect on the outcome, so the potential outcome could be modeled by the three treatments. Since the outcome is binary, so `logit` link is used. The marginal outcome model is specified as:

$$logit(P(Y = 1)) = \theta_0 + \sum_{i=1}^{3} \theta_i A_i$$

The treatment assignment models are specified based on the causal DAG/data generation mechanism. propensity score model is built for each visit:
- visit 1:

$$logit(P(A_1 = 1)) = \gamma_{10} + \gamma_{11} L_{11} + \gamma_{12} L_{21}$$

- visit 2:

$$logit(P(A_2 = 1)) = \gamma_{20} + \gamma_{21} L_{12} + \gamma_{22} L_{22} + \gamma_{23} A_1$$

- visit 3:

$$logit(P(A_3 = 1)) = \gamma_{30} + \gamma_{31} L_{13} + \gamma_{32} L_{23} + \gamma_{33} A_2$$

**(ii) assess and comment on the overlap of the treatment assignment probability or weights at each visit.**

```
## calculate stabilized weight
library(WeightIt)

Wmsm <- weightitMSM(
  list(A1 ~ L11 + L21,
       A2 ~ L12 + L22 + A1,
       A3 ~ L13 + L23 + A2),
  data = simdat,
  method = "ps",
  stabilize = TRUE)

Wmsm
```

```
A weightitMSM object
 - method: "glm" (propensity score weighting with GLM)
 - number of obs.: 500
 - sampling weights: none
 - number of time points: 3 (A1, A2, A3)
 - treatment:
    + time 1: 2-category
    + time 2: 2-category
    + time 3: 2-category
 - covariates:
    + baseline: L11, L21
    + after time 1: L12, L22, A1
    + after time 2: L13, L23, A2
 - stabilized; stabilization factors:
    + baseline: (none)
    + after time 1: A1
    + after time 2: A1, A2, A1:A2
```

The following shows a summary of the weight ranges between the treated and control overlap, we see good overlap of the weights and there is no extreme weights. Also, `bal.tab` function call results showed that all the covariates are balanced after weighting. ($A_1$ and $A_2$ are not balanced, but this causes no issue as we will include treatment variables in MSM model)

```
## check the weight overlap
summary(Wmsm)
```

```
$A1
                Summary of weights

- Weight ranges:

          Min                                   Max
treated 0.4488 |-----------------|           2.0174
control 0.5167 |--------------------------| 2.7550

- Units with the 5 most extreme weights by group:

            254    380    158    277    159
 treated 1.6927 1.7488 1.7763 1.8656 2.0174
            189    339    313     68    381
 control 1.7619 1.7927 1.8113 1.9912  2.755

- Weight statistics:

        Coef of Var   MAD Entropy # Zeros
treated       0.242 0.187   0.028       0
control       0.289 0.219   0.038       0

- Mean of Weights = 1

- Effective Sample Sizes:

           Control Treated
Unweighted  220.    280.
Weighted    203.14  264.53

$A2
                Summary of weights

- Weight ranges:

          Min                                   Max
treated 0.5395  |--------------|            1.8656
control 0.4488 |--------------------------| 2.7550

- Units with the 5 most extreme weights by group:

            496    380    158    339    277
 treated 1.7173 1.7488 1.7763 1.7927 1.8656
```

```
            189    313     68    159    381
 control 1.7619 1.8113 1.9912 2.0174  2.755

- Weight statistics:

        Coef of Var   MAD Entropy # Zeros
treated       0.233 0.184   0.026       0
control       0.306 0.227   0.042       0

- Mean of Weights = 1

- Effective Sample Sizes:

           Control Treated
Unweighted  200.    300.
Weighted    182.93  284.65

$A3
                Summary of weights

- Weight ranges:

           Min                                   Max
treated 0.6173   |---------------|           1.9912
control 0.4488 |---------------------------| 2.7550

- Units with the 5 most extreme weights by group:

            380    189    158    277     68
 treated 1.7488 1.7619 1.7763 1.8656 1.9912
            496    339    313    159    381
 control 1.7173 1.7927 1.8113 2.0174  2.755

- Weight statistics:

        Coef of Var   MAD Entropy # Zeros
treated       0.240 0.190   0.027       0
control       0.303 0.222   0.041       0

- Mean of Weights = 1

- Effective Sample Sizes:
```

```
           Control Treated
Unweighted  182.    318.
Weighted    166.77  300.76

attr(,"class")
[1] "summary.weightitMSM"
```

```r
# check covariate overlap
library(cobalt)
bal.tab(Wmsm,
        stats = c("m"),
        thresholds = c(m = .1),
        which.time = .none)
```

```
Balance summary across all time points
             Times     Type Max.Diff.Adj        M.Threshold
prop.score 1, 2, 3 Distance       0.1858
L11              1   Binary       0.0097     Balanced, <0.1
L21              1  Contin.       0.0165     Balanced, <0.1
L12              2   Binary       0.0101     Balanced, <0.1
L22              2  Contin.       0.0714     Balanced, <0.1
A1               2   Binary       0.1317 Not Balanced, >0.1
L13              3   Binary       0.0103     Balanced, <0.1
L23              3  Contin.       0.0716     Balanced, <0.1
A2               3   Binary       0.1294 Not Balanced, >0.1

Balance tally for mean differences
                   count
Balanced, <0.1         6
Not Balanced, >0.1     2

Variable with the greatest mean difference
 Variable Max.Diff.Adj        M.Threshold
       A1       0.1317 Not Balanced, >0.1

Effective sample sizes
 - Time 1
           Control Treated
Unadjusted  220.    280.
Adjusted    203.14  264.53
 - Time 2
           Control Treated
```

```
Unadjusted  200.    300.
Adjusted    182.93  284.65
 - Time 3
           Control Treated
Unadjusted  182.    318.
Adjusted    166.77  300.76


Call:
svyglm(formula = Y ~ A1 + A2 + A3, design = msm_design, family = binomial(link = "logit"))

Survey design:
svydesign(~1, weights = Wmsm$weights, data = simdat)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.01162    0.20776   0.056  0.95542
A1          -0.02139    0.19160  -0.112  0.91115
A2           0.05682    0.19368   0.293  0.76938
A3           0.57636    0.19615   2.938  0.00345 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1.00224)

Number of Fisher Scoring iterations: 4
```

### (iii) Report the point estimate of the causal estimand and it's 95% CI.

The point estimate of the causal estimand (always treated versus never treated) is 0.148 with a bootstrapped 95% CI of [0.001, 1.157]. The following codes are used to get the estimate.

```r
## MSM model
library(survey)

# first create a survey object;
msm_design <- svydesign(~1, weights = Wmsm$weights, data = simdat)

fitMSM <- svyglm(Y ~ A1+A2+A3,
                 family=binomial(link = "logit"),
                 design = msm_design)
```

```r
summary(fitMSM)
```

```
Call:
svyglm(formula = Y ~ A1 + A2 + A3, design = msm_design, family = binomial(link = "logit"))

Survey design:
svydesign(~1, weights = Wmsm$weights, data = simdat)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.002544   0.205790   0.012  0.99014
A1          -0.030421   0.193696  -0.157  0.87526
A2           0.103279   0.195416   0.529  0.59738
A3           0.520473   0.197414   2.636  0.00864 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1.002391)

Number of Fisher Scoring iterations: 4
```

```r
##  point estimate of the causal estimand
always_treat <- predict(fitMSM, newdata = data.frame(A1=1,A2=1,A3=1),type = 'response')
never_treat <- predict(fitMSM, newdata = data.frame(A1=0,A2=0,A3=0),type='response')

always_treat - never_treat
```

```
  response     SE
1  0.14408 0.0372
```

```r
## compute bootstrap 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,]

  Wmsm <- weightitMSM(
```

```
  list(A1 ~ L11 + L21,
       A2 ~ L12 + L22 + A1,
       A3 ~ L13 + L23 + A2),
  data = boot.data,
  method = "ps",
  stabilize = TRUE)

  msm_design <- svydesign(~1, weights = Wmsm$weights, data = boot.data)

  fitMSM <- svyglm(Y ~ A1+A2+A3,
                   family=binomial(link = "logit"),
                   design = msm_design)

  boot.est[i] <- predict(fitMSM,
                         newdata = data.frame(A1=1,A2=1,A3=1))[1] -
    predict(fitMSM, newdata = data.frame(A1=0,A2=0,A3=0))[1]

}

# SE of ATE;


#95% CI
quantile(boot.est, probs = c(0.025, 0.975))
```

```
      2.5%      97.5%
0.00118781 1.15732359
```

**(c) (G-computation) Estimate the same causal estimand, the expected absolute risk difference between always treated vs never treated, but now using parametric g-computation.**

```
# preparing the data: long format
library(tidyr)
## load and manipulate data
colnames(simdat) <- c("L1_1" ,"L2_1" ,"A_1" ,
                      "L1_2", "L2_2", "A_2",
                      "L1_3", "L2_3" ,"A_3" ,
                      "Y")
```

```r
simdat_long <- simdat %>%
  mutate(id = rep(1:nrow(simdat))) %>%
  pivot_longer(cols = -c(Y,id),
               names_to = c("variable","visit"),
               names_sep = "_",
               values_to = "value") %>%
  pivot_wider(names_from = variable, values_from = value) %>%
  mutate(time = case_when(visit == 1 ~ 0,
                          visit == 2 ~ 1,
                          visit == 3 ~ 2))

# Y is only measured at the end-of-study,
# thus, when we pivot to long format visit 1's y will have a missing value;
simdat_long$Y[simdat_long$visit == 1] <- NA
simdat_long$Y[simdat_long$visit == 2] <- NA

# look at the new data;
head(simdat_long)
```

```
# A tibble: 6 x 7
      Y    id visit    L1     L2     A  time
  <dbl> <int> <chr> <dbl>  <dbl> <dbl> <dbl>
1    NA     1 1         0 -0.376     0     0
2    NA     1 2         1  0.864     1     1
3     1     1 3         1 -1.52      1     2
4    NA     2 1         1 -0.562     0     0
5    NA     2 2         1  0.142     1     1
6     1     2 3         0 -0.740     1     2
```

**(i) For this analysis, please explain and specify your choice of the conditional outcome model and the time-varying covariates models.**

- conditional outcome model: the conditional outcome model is specified based on the data generating mechanism

```r
Yprob<- expit (0.3*A3+0.1*A2-0.1*A1+0.1*L13-0.2*L23)
Y<- rbinom( n = samplesize, size = 1, prob = Yprob)
```

so the conditional model can be specified as:

$$logit(P(Y=1)) = \theta_0 + \sum_{t=1}^{3} \theta_t A_t + \gamma_1 L_{13} + \gamma_2 L_{23}$$

- time-varying covariates models: the data generating mechanism for $L_1$ is

```
## for L1
L12prob<- expit (0.5*A1+0.5*L11-0.2*L21)
L13prob<- expit (0.5*A2+0.5*L12-0.2*L22)

## for L2
meanL22<- 0.5*L21-0.5*A1-0.2*L11
meanL23<- 0.5*L22-0.5*A2-0.2*L12
```

so $L_1$ and $L_2$ can be modeled use lag1_$A$, lag1_$L_1$, and lag1_$L_2$, that is:

- for $L_1$:

$$logit(P(L_1 = 1)) = \theta_{01} + \theta_{11} lag1\_L_1 + \theta_{21} lag1\_L_2 + \theta_{31} lag1\_A$$

- for $L_2$:

$$E(L_2) = \theta_{02} + \theta_{12} lag1\_L_1 + \theta_{22} lag1\_L_2 + \theta_{32} lag1\_A$$

- Finally, the target treatment assignment model is specified as:

$$logit(P(A = 1) = L_1 + L_2 + lag1\_A$$

**(ii) Report the point estimate of the causal estimand and it's 95% CI.**

```
library(gfoRmula)

id <- 'id'
time_name <- 'time'
covnames <- c("L1", "L2", "A")
outcome_name <- 'Y'
covtypes <- c('binary', 'normal', 'binary')
histories <- c(lagged) #lagged feature to call for lagged value from the long format data;
histvars <- list(c('A', 'L1', 'L2'))

covparams <- list(
  covmodels = c(L1 ~ lag1_L1 + lag1_L2+ lag1_A,
                L2 ~ lag1_L1 + lag1_L2+ lag1_A,
                A ~ L1+L2+lag1_A))
```

```r
ymodel <- Y ~ lag1_A + lag2_A + A + L1 + L2


intvars <- list('A','A')
interventions <- list(list(c(static, rep(0, 3))),
                      list(c(static, rep(1, 3)))
                      )
int_descript <- c('Never treat', 'Always treat')

gform_eof <- gformula_continuous_eof(
  obs_data = simdat_long,
  id = id,
  time_name = time_name,
  covnames =covnames,
  outcome_name = outcome_name,
  covtypes = c("binary", "normal", "binary"),
  covparams = covparams,
  ymodel = ymodel,
  intvars = intvars,
  interventions = interventions,
  int_descript = int_descript,
  ref_int = 1,
  histories = c(lagged),
  histvars = list(c('A',"L1","L2")), #variables that are time-dependent;
  # basecovs = c("w1","w2"), #time-independent baseline var;
  nsimul = 1000,
  nsamples = 1000,
  parallel = TRUE,
  ncores = 6, #bootstrap features;
  seed = 1017)

summary(gform_eof)
```

```
PREDICTED RISK UNDER MULTIPLE INTERVENTIONS

Intervention     Description
0        Natural course
1        Never treat
2        Always treat

Sample size = 500, Monte Carlo sample size = 1000
Number of bootstrap samples = 1000
```

```
Reference intervention = 1
```

```
 k Interv. NP mean g-form mean    Mean SE Mean lower 95% CI Mean upper 95% CI
 2       0   0.588   0.5848332 0.02201400         0.5420992         0.6274840
 2       1      NA   0.4814057 0.04821448         0.3863248         0.5771198
 2       2      NA   0.6503573 0.03462426         0.5807810         0.7179872
 Mean ratio     MR SE MR lower 95% CI MR upper 95% CI Mean difference
   1.214845 0.1110801        1.037043        1.478846       0.1034274
   1.000000 0.0000000        1.000000        1.000000       0.0000000
   1.350955 0.1834461        1.050362        1.774989       0.1689516
      MD SE MD lower 95% CI MD upper 95% CI % Intervened On
 0.04196357      0.02036180       0.1871209             0.0
 0.00000000      0.00000000       0.0000000            89.5
 0.06984808      0.02814259       0.3015524            75.2
 Aver % Intervened On
              0.00000
             54.76667
             37.00000
```

```
gcomp_res <- gform_eof$result


gcomp_estimate <- gcomp_res$`Mean difference`[3]

gcomp_ci <- c(gcomp_res$`MD lower 95% CI`[3],gcomp_res$`MD upper 95% CI`[3])
```

The point estimate of the causal estimand and it's 95% bootstrapped CI using g computation:

```
## g computation point estimate
round(gcomp_estimate,3)
```

```
[1] 0.169
```

```
## 95% CI

round(gcomp_ci,3)
```

```
[1] 0.028 0.302
```

**(d) (LTMLE) Similar to (b) and (c), please estimate the same causal estimand using LTMLE and SuperLearner (you can just pick 1 algorithm). Report the point estimate of the causal estimand and it's 95% CI.**

```
library(ltmle)
# Step 1, if applicable remove variables we don't need;
colnames(simdat)
```

```
 [1] "L1_1" "L2_1" "A_1"  "L1_2" "L2_2" "A_2"  "L1_3" "L2_3" "A_3"  "Y"
```

```
# Step 2, fitting conventional tmle without superlearner (machine learning algorithm);
library(ltmle)
tmle_model <- ltmle(data = simdat,
                    Anodes = c("A_1","A_2","A_3"),
                    Lnodes = c("L1_1", "L2_1", "L1_2", "L2_2","L1_3", "L2_3"),
                    Ynodes = c("Y"),
                    survivalOutcome =FALSE,
                    SL.library = 'SL.xgboost',
                    gform = c("A_1 ~ L1_1 + L2_1",
                              "A_2 ~ L1_2 + L2_2 + A_1",
                              "A_3 ~ L1_3 + L2_3 + A_2"),
                    abar = list(c(1,1,1), c(0,0,0)))
```

```
Loading required namespace: SuperLearner

Qform not specified, using defaults:

formula for L1_2:

Q.kplus1 ~ L1_1 + L2_1 + A_1

formula for L1_3:

Q.kplus1 ~ L1_1 + L2_1 + A_1 + L1_2 + L2_2 + A_2

formula for Y:

Q.kplus1 ~ L1_1 + L2_1 + A_1 + L1_2 + L2_2 + A_2 + L1_3 + L2_3 +     A_3
```

```
Loading required package: nnls

Loading required namespace: xgboost

Estimate of time to completion: 1 minute
```

```r
tlme_res <- summary(tmle_model, estimator="tmle")
tlme_res
```

```
Estimator:  tmle
Call:
ltmle(data = simdat, Anodes = c("A_1", "A_2", "A_3"), Lnodes = c("L1_1",
    "L2_1", "L1_2", "L2_2", "L1_3", "L2_3"), Ynodes = c("Y"),
    survivalOutcome = FALSE, gform = c("A_1 ~ L1_1 + L2_1", "A_2 ~ L1_2 + L2_2 + A_1",
        "A_3 ~ L1_3 + L2_3 + A_2"), abar = list(c(1, 1, 1), c(0,
        0, 0)), SL.library = "SL.xgboost")

Treatment Estimate:
   Parameter Estimate:  0.60319
    Estimated Std Err:  0.042284
              p-value:  <2e-16
    95% Conf Interval: (0.52031, 0.68606)

Control Estimate:
   Parameter Estimate:  0.44566
    Estimated Std Err:  0.086492
              p-value:  2.5693e-07
    95% Conf Interval: (0.27614, 0.61518)

Additive Treatment Effect:
   Parameter Estimate:  0.15753
    Estimated Std Err:  0.096275
              p-value:  0.10179
    95% Conf Interval: (-0.031166, 0.34622)

Relative Risk:
   Parameter Estimate:  1.3535
  Est Std Err log(RR):  0.20635
              p-value:  0.14243
```

```
    95% Conf Interval: (0.90324, 2.0281)

Odds Ratio:
   Parameter Estimate:  1.8908
  Est Std Err log(OR):  0.39215
              p-value:  0.1043
    95% Conf Interval: (0.87668, 4.0779)
```

```
tlme_estimate <- tlme_res$effect.measures$ATE$estimate
tlme_ci <- tlme_res$effect.measures$ATE$CI
```

the point estimate of the causal estimand and it's 95% CI using TLME:

```
## tlme point estimate

tlme_estimate
```

```
[1] 0.1575286
```

```
## tlme boot 95%CI
tlme_ci
```

```
            2.5%     97.5%
[1,] -0.03116646 0.3462236
```

**(e) Compare and comment on your results from the three causal approaches.**

The following is a summary table of the casual estimates results of the three methods (codes in the following).

we see the point estimates are similar. But MSM gave wider confidence interval compared to the other two methods. and we see that the bootstrapped confidence interval from tlme crossed 0, MSM and g computation methods based confidence interval do not include 0.

```
summary_res <- data.frame(method = c("MSM","g computation", "tlme"),
           point_est = c(round(msm_estimate[1],3),
                         round(gcomp_estimate,3),
                         round(tlme_estimate,3)),
           boot_95CI = c(paste0("(",round(msm_ci[1],3),",",round(msm_ci[2],3),")"),
                         paste0("(",round(gcomp_ci[1],3),",",round(gcomp_ci[2],3),")"),
```

```
                          paste0("(",round(tlme_ci[1],3),",",round(tlme_ci[2],3),")")
                          )

)

print(summary_res)
```

```
         method point_est      boot_95CI
1           MSM     0.148  (0.001,1.157)
2 g computation     0.169  (0.028,0.302)
3          tlme     0.158 (-0.031,0.346)
```

## Question 4

**(a) For this question, you will be working with a simulated longitudinal dataset modified from the previous question.**

**(i) please report the frequency table of the treatment sequences across the three visits using the simulated data**

```
simdat_cen1 <- simdat_cen %>%
  mutate(A1 = ifelse(A1==0,"A1=0","A1=1"),
         A2 = ifelse(A2==0,"A2=0","A2=1"),
    A3 = case_when(is.na(A3)~ "A3 = censored",
                   A3 ==0 ~ "A3=0",
                   A3 ==1 ~ "A3=1"
                   )
    )
# freq table
ftable(simdat_cen1$A1, simdat_cen1$A2,simdat_cen1$A3)
```

```
           A3 = censored A3=0 A3=1

A1=0 A2=0             10   42   50
     A2=1             10   39   69
A1=1 A2=0              9   28   61
     A2=1             18   55  109
```

```
# percentage
ftable(simdat_cen1$A1, simdat_cen1$A2,simdat_cen1$A3)/nrow(simdat_cen1)
```

```
           A3 = censored  A3=0  A3=1

A1=0 A2=0           0.020 0.084 0.100
     A2=1           0.020 0.078 0.138
A1=1 A2=0           0.018 0.056 0.122
     A2=1           0.036 0.110 0.218
```

**(ii) and report the overall number of observations that are censored after visit 2 and the number of observations that are censored after visit 2 by the received treatment at visit 1 and visit 2.**

```
# overall number of observations that are censored after visit 2
sum(is.na(simdat_cen$A3))
```

```
[1] 47
```

```
#by the received treatment at visit 1 and visit 2.
cen_dat <- simdat_cen[is.na(simdat_cen$A3),] %>%
  mutate(A1 = ifelse(A1==0,"A1=0","A1=1"),
         A2 = ifelse(A2==0,"A2=0","A2=1"))

# number of censored obs by visit1 and visit2
table(cen_dat$A1,cen_dat$A2)
```

```
      A2=0 A2=1
  A1=0   10   10
  A1=1    9   18
```

```
## percentage of censored by visit 1 and visit2
table(cen_dat$A1,cen_dat$A2)/nrow(cen_dat)
```

```
           A2=0      A2=1
  A1=0 0.2127660 0.2127660
  A1=1 0.1914894 0.3829787
```

**(b) (MSMs with Censoring) Based on the simulated code for the end-of-study outcome, please conduct a marginal structural models analysis using stabilized weights and with adjustment of censoring to estimate the expected absolute risk difference between always treated vs never treated on the outcome.**

**(i) For this analysis, please explain and specify your choice of the marginal outcome model, the time-varying treatment assignment model, and the censoring model.**

The models are specified based on the data generating mechanism.

- marginal outcome model is specified as:

$$logit(P(Y=1) = \alpha_0 + \sum_{i=0}^{3} \alpha_i A_i$$

- time-varying treatment model at each visit:
  - visit 1:

$$logit(P(A_1=1)) = \beta_{01} + \beta_{11} L_{11} + \beta_{21} L_{21}$$

  - visit 2:

$$logit(P(A_2=1)) = \beta_{02} + \beta_{12} L_{12} + \beta_{22} L_{22} + \gamma_1 A_1$$

- visit 3:

$$logit(P(A_3=1)) = \beta_{03} + \beta_{13} L_{13} + \beta_{23} L_{23} + \gamma_2 A_2$$

- censoring model is:

$$logit(P(C=1)) = \alpha_0 + \alpha_1 L_{12} + \alpha_2 L_{22} + \alpha_3 A_2$$

**(ii) Please assess and comment on the overlap of the treatment assignment probability or weights at each visit.**

- First we calcuate the treatment assignment probability as follows:

```r
# calculating weights manually
# IPT weights;
library(survey)
colnames(simdat_cen) <- c("L1_1", "L2_1",
                          "A_1", "L1_2",
                          "L2_2", "A_2",
                          "L1_3", "L2_3",
                          "A_3", "C",
                          "Y")
tmodel1 <- glm(A_1 ~ L1_1 + L2_1,
               family=binomial(link=logit), data = simdat_cen)
tmodel2 <- glm(A_2  ~ L1_2 + L2_2 +A_1,
               family=binomial(link=logit), data = simdat_cen)
tmodel3 <- glm(A_3  ~   L1_3 + L2_3 +A_2,
               family=binomial(link=logit), data = simdat_cen)
# Stabilizer;
smodel1 <- glm(A_1 ~ 1, family=binomial(link=logit), data = simdat_cen)
smodel2 <- glm(A_2 ~ A_1, family=binomial(link=logit), data = simdat_cen)
smodel3 <- glm(A_3 ~ A_2, family=binomial(link=logit), data = simdat_cen)
# censoring model
cmodel <- glm(C ~ L1_2 + L2_2 + A_2, family=binomial(link=logit), data = simdat_cen)
csmodel <- glm(C ~ A_2, family=binomial(link=logit), data = simdat_cen)

# calculating treat weights;
t_numerator <- predict(smodel1, type = "response", newdata = simdat_cen)*
  predict(smodel2, type = "response", newdata = simdat_cen)*
  predict(smodel3, type = "response", newdata = simdat_cen)
t_denominator <- predict(tmodel1, type = "response", newdata = simdat_cen)*
  predict(tmodel2, type = "response", newdata = simdat_cen)*
  predict(tmodel3, type = "response", newdata = simdat_cen)

iptw <- t_numerator/t_denominator

# calculating censoring weights;
c_numerator <- predict(csmodel, type = "response", newdata = simdat_cen)
c_denominator <- predict(cmodel, type = "response", newdata = simdat_cen)

ipcw <- c_numerator/c_denominator

w <- iptw*ipcw

censor_design <- svydesign(id=~1, weights = ~ na.omit(w), data = na.omit(simdat_cen))
```

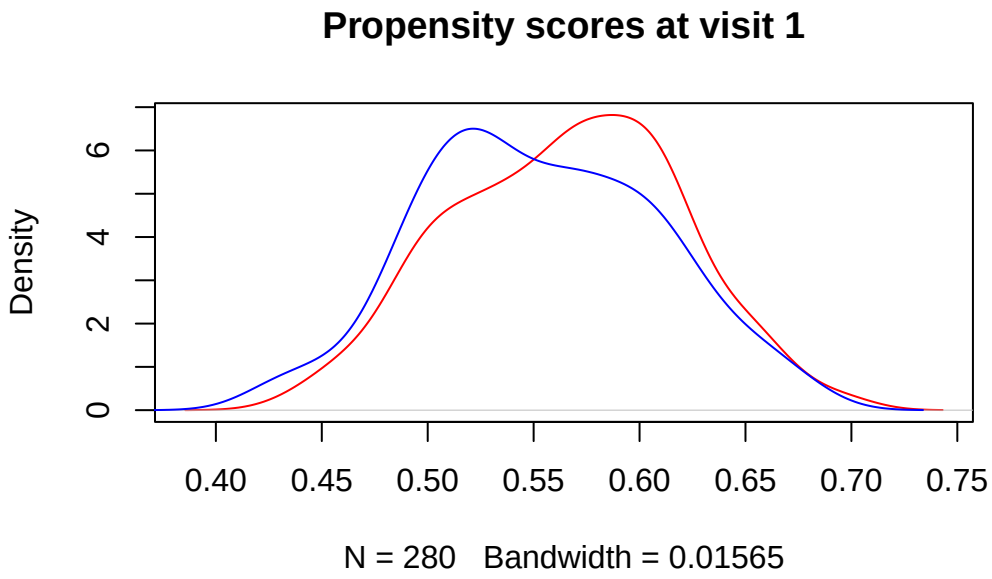
```r
censor_mod <- svyglm(Y ~ A_1+A_2+A_3,
                     design = censor_design,
                     family = binomial(link='logit'))
summary(censor_mod)
```

```
Call:
svyglm(formula = Y ~ A_1 + A_2 + A_3, design = censor_design,
    family = binomial(link = "logit"))

Survey design:
svydesign(id = ~1, weights = ~na.omit(w), data = na.omit(simdat_cen))

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.4327     0.2300  -1.881   0.0606 .
A_1           0.2342     0.2145   1.091   0.2757
A_2           0.4746     0.2153   2.204   0.0280 *
A_3           0.3269     0.2178   1.501   0.1341
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1.001939)

Number of Fisher Scoring iterations: 4
```

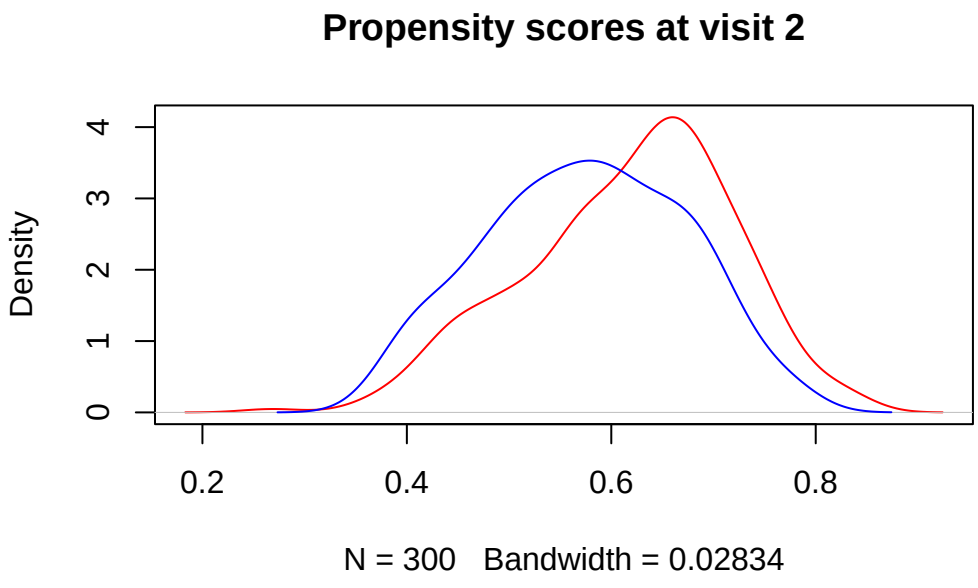
- Then we evaluate overlap
  The density plots of the treatment assignment probabilities for each visit are shown below. These plots indicate a good overlap of the propensity scores between the treatment and control groups. Also, there are no extreme values.

```r
propensity_score_dat <- simdat_cen %>%
  mutate(A_1 = as.factor(A_1),
         A_2 = as.factor(A_2),
         A_3 = as.factor(A_3)
         )

propensity_score_dat$pred_A1 <- predict(tmodel1, type = "response", newdata = simdat_cen)

propensity_score_dat$pred_A2 <- predict(tmodel2, type = "response", newdata = simdat_cen)

propensity_score_dat$pred_A3 <- predict(tmodel3, type = "response", newdata = simdat_cen)
```

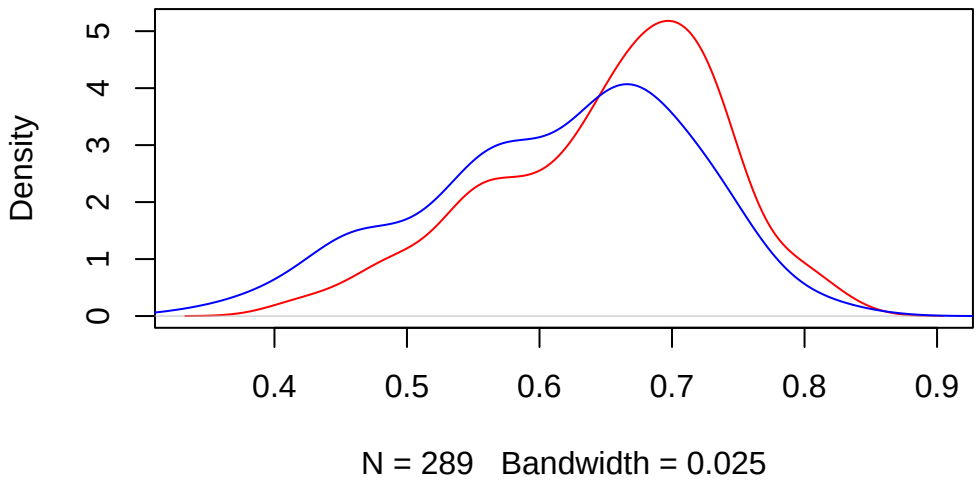
```r
plot(density(propensity_score_dat$pred_A1[propensity_score_dat$A_1==1]),col='red',main="Prope
lines(density(propensity_score_dat$pred_A1[propensity_score_dat$A_1==0]),col='blue')
```




```r
plot(density(propensity_score_dat$pred_A2[propensity_score_dat$A_2==1]),col='red',main="Prope
lines(density(propensity_score_dat$pred_A2[propensity_score_dat$A_2==0]),col='blue')
```




```
plot(density(na.omit(propensity_score_dat$pred_A3)[na.omit(propensity_score_dat$A_3==1)]),col
lines(density(na.omit(propensity_score_dat$pred_A3)[na.omit(propensity_score_dat$A_3==0)]),co
```



**(iii) Report the point estimate of the causal estimand and it's 95% CI.**

```
## point estimate
always_treat <- predict(censor_mod, type='response', newdata = data.frame(A_1=1,A_2=1,A_3=1))
never_treat <- predict(censor_mod, type='response', newdata = data.frame(A_1=0,A_2=0,A_3=0))

## point estimates
always_treat - never_treat
```

```
  response    SE
1  0.37252 0.037
```

```
## 95% boot CI
set.seed(1017)
boot.est <- rep(NA, 1000)
for (i in 1:1000){

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

tmodel1 <- glm(A_1 ~ L1_1 + L2_1,
               family=binomial(link=logit), data =  boot.data)
tmodel2 <- glm(A_2  ~ L1_2 + L2_2 +A_1,
               family=binomial(link=logit), data =  boot.data)
tmodel3 <- glm(A_3  ~   L1_3 + L2_3 +A_2,
               family=binomial(link=logit), data =  boot.data)
# Stabilizer;
smodel1 <- glm(A_1 ~ 1, family=binomial(link=logit), data =  boot.data)
smodel2 <- glm(A_2 ~ A_1, family=binomial(link=logit), data =  boot.data)
smodel3 <- glm(A_3 ~ A_2, family=binomial(link=logit), data =  boot.data)
# censoring model
cmodel <- glm(C ~ L1_2 + L2_2 + A_2, family=binomial(link=logit), data =  boot.data)
csmodel <- glm(C ~ A_2, family=binomial(link=logit), data =  boot.data)

# calculating treat weights;
t_numerator <- predict(smodel1, type = "response", newdata =  boot.data)*
  predict(smodel2, type = "response", newdata =  boot.data)*
  predict(smodel3, type = "response", newdata =  boot.data)
t_denominator <- predict(tmodel1, type = "response", newdata =  boot.data)*
  predict(tmodel2, type = "response", newdata =  boot.data)*
  predict(tmodel3, type = "response", newdata =  boot.data)
```

```r
iptw <- t_numerator/t_denominator

# calculating censoring weights;
c_numerator <- predict(csmodel, type = "response", newdata =  boot.data)
c_denominator <- predict(cmodel, type = "response", newdata =  boot.data)

ipcw <- c_numerator/c_denominator

w <- iptw*ipcw

censor_design <- svydesign(id=~1, weights = ~ na.omit(w), data =  na.omit(boot.data))
censor_mod <- svyglm(Y ~ A_1+ A_2+A_3,
                     design = censor_design,
                     family = binomial(link='logit'))

APO_11 <- predict(censor_mod, type='response',newdata = data.frame(A_1=1,A_2=1,A_3=1))
APO_00 <- predict(censor_mod,type='response', newdata = data.frame(A_1=0,A_2=0,A_3=0))

##  IV estimator
  boot.est[i] <- APO_11 - APO_00

}

#95% CI
quantile(boot.est, probs = c(0.025, 0.975))
```

```
      2.5%      97.5%
0.08830597 0.40863581
```