

610 Final project

Ariane Stark, Minsu Kim, Diezhang Wu, Alona Muzikansky

11/6/2021

Alona

```
set.seed(123)
# simple substitution estimator (a.k.a. parameteric G-computation)
txt <- ObsData
control <- ObsData

txt$A <- 1
control$A <- 0

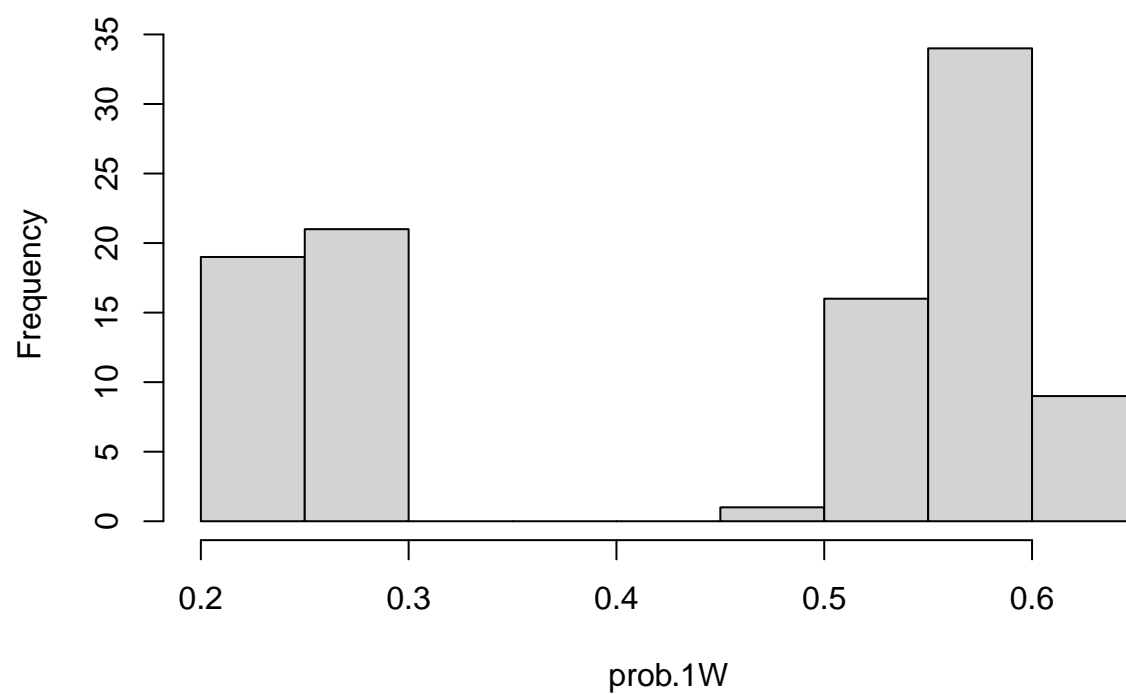
g.comp.reg <- glm(Y ~ W11 + W12 + W13 + W14 + W2 + A, family="binomial", data=ObsData)
pred.txt <- predict(g.comp.reg, newdata = txt, type = "response")
pred.control <- predict(g.comp.reg, newdata = control, type = "response")
psi.hat <- mean(pred.txt - pred.control)
psi.hat
```

```
## [1] 0.01454638
```

```
# IPTW estimator
prob.AW.reg <- glm(A ~ W11 + W12 + W13 + W14, family="binomial", data=ObsData)
prob.1W <- predict(prob.AW.reg, type= "response")
prob.0W <- 1 - prob.1W

hist(prob.1W)
```

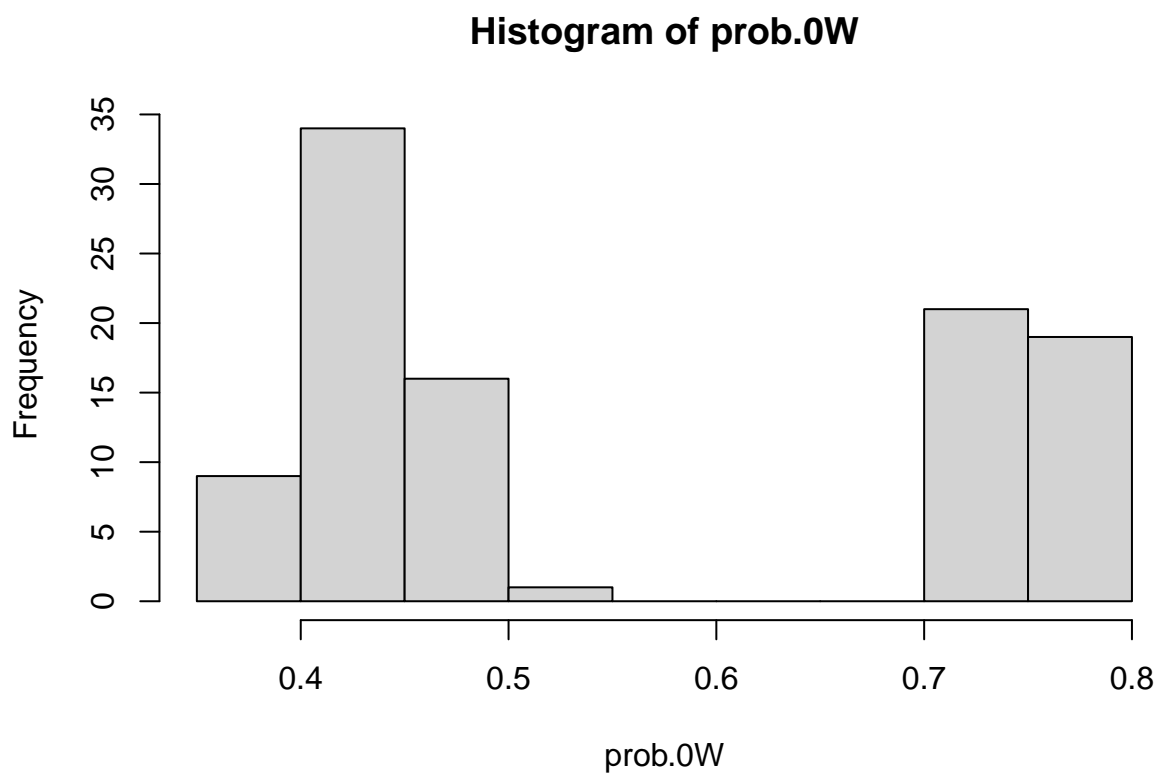
Histogram of prob.1W



```
summary(prob.1W)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.2216 0.2516 0.5358 0.4400 0.5727 0.6226
```

```
hist(prob.0W)
```



```
summary(prob.0W)
```

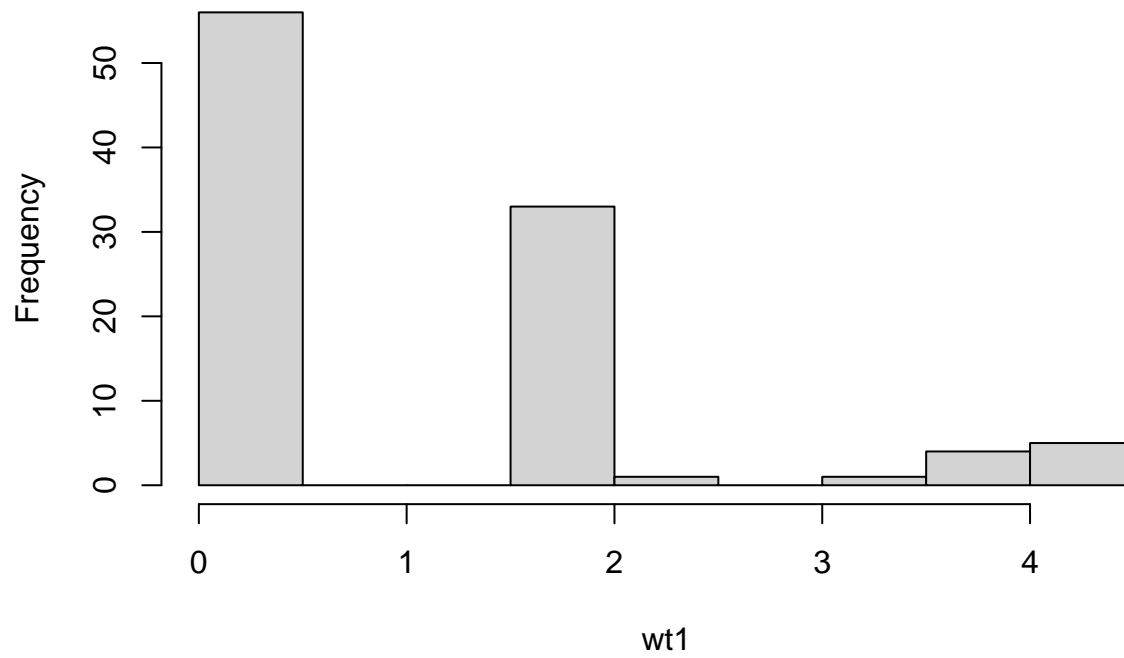
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.3774 0.4273 0.4642 0.5600 0.7484 0.7784
```

```
wt1 <- as.numeric(ObsData$A==1)/prob.1W
wt0 <- as.numeric(ObsData$A==0)/prob.0W
summary(wt1)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.000 0.000 0.000 1.004 1.766 4.452
```

```
hist(wt1)
```

Histogram of wt1

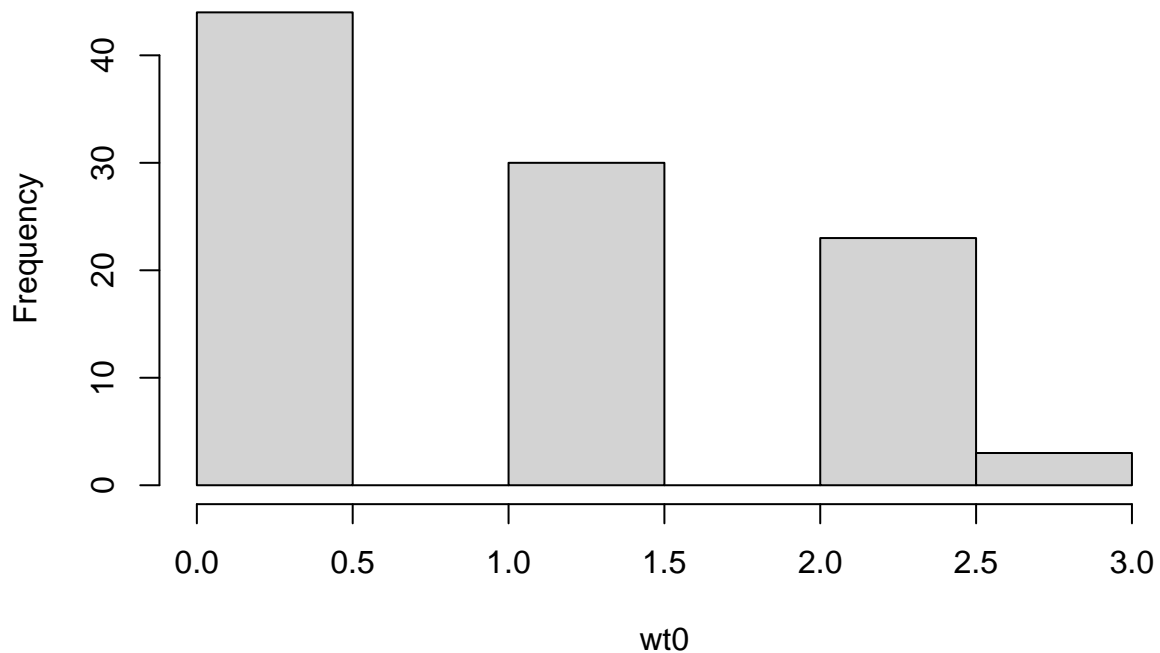


```
summary(wt0)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.000  0.000   1.312   0.998  2.074   2.594
```

```
hist(wt0)
```

Histogram of wt0



```
psi.iptw <- mean(wt1*ObsData$Y) - mean(wt0*ObsData$Y)
psi.iptw
```

```
## [1] -0.003532538
```

```
# Modified HT
```

```
psi.ht <- mean(wt1*ObsData$Y)/mean(wt1) - mean(wt0*ObsData$Y)/mean(wt0)
psi.ht
```

```
## [1] -0.00916455
```

```
# Unadjusted estimator
```

```
wt1.ua <- as.numeric(ObsData$A==1)/mean(ObsData$A == 1)
wt0.ua <- as.numeric(ObsData$A==0)/mean(ObsData$A == 0)
psi.unadj <- mean(wt1.ua*ObsData$Y) - mean(wt0.ua*ObsData$Y)
psi.unadj
```

```
## [1] -0.02922078
```

```
# TMLE estimator
```

SS, IPTW and TMLE estimator with super learner

```

library("SuperLearner")
SL.library<- c('SL.glm', 'SL.glm.interaction', "SL.step",
              "SL.randomForest", "SL.step.forward", "SL.stepAIC", "SL.mean")

run.tmlle <- function(ObsData, SL.library){

  #-----
  # Simple substitution estimator
  #-----

  # dataframe X with baseline covariates and exposure
  X <- subset(ObsData, select=c(A, W11, W12, W13, W14,W2))

  # set the exposure=1 in X1 and the exposure=0 in X0
  X1 <- X0 <- X
  X1$A <- 1
  X0$A <- 0

  # Estimate  $E_0(Y/A, W)$  with Super Learner
  SL.outcome <- SuperLearner(Y=ObsData$Y, X=X, SL.library=SL.library,
                           family="binomial", cvControl=list(V=10))

  # get the expected outcome, given the observed exposure and covariates
  expY.givenAW <- predict(SL.outcome, newdata=ObsData)$pred
  # expected outcome, given A=1 and covariates
  expY.given1W <- predict(SL.outcome, newdata=X1)$pred
  # expected outcome, given A=0 and covariates
  expY.given0W <- predict(SL.outcome, newdata=X0)$pred

  # simple substitution estimator would be
  PsiHat.SS <- mean(expY.given1W - expY.given0W)

  #-----
  # Inverse probability of tx weighting
  #-----

  # Super Learner for the exposure mechanism  $P_0(A=1|W)$ 
  SL.exposure <- SuperLearner(Y=ObsData$A,
                             X=subset(ObsData, select= -c(A,Y,W2)),
                             SL.library=SL.library, family="binomial",
                             cvControl=list(V=10, stratifyCV = TRUE))

  # generate the predicted prob of being exposed, given baseline cov
  probA1.givenW <- SL.exposure$SL.predict
  # generate the predicted prob of not being exposed, given baseline cov
  probA0.givenW <- 1- probA1.givenW

  # clever covariate
  H.AW <- as.numeric(ObsData$A==1)/probA1.givenW - as.numeric(ObsData$A==0)/probA0.givenW

  # also want to evaluate the clever covariate at A=1 and A=0 for all participants
  H.1W <- 1/probA1.givenW
  H.0W <- -1/probA0.givenW

```

```

# IPTW estimate
PsiHat.IPTW <- mean(H.AW*ObsData$Y, na.rm = TRUE)

#-----
# Targeting & TMLE
#-----

# Update the initial estimator of  $E_0(Y|A,W)$ 
# run logistic regression of Y on H.AW using the logit of the estimates as offset
expY.givenAW <- expY.givenAW - 0.000001
logitUpdate<- glm( ObsData$Y ~ -1 +offset(qlogis(expY.givenAW)) +
                  H.AW, family='binomial')
epsilon <- logitUpdate$coef

# obtain the targeted estimates
expY.givenAW.star<- plogis( qlogis(expY.givenAW)+ epsilon*H.AW )
expY.given1W.star<- plogis( qlogis(expY.given1W)+ epsilon*H.1W )
expY.given0W.star<- plogis( qlogis(expY.given0W)+ epsilon*H.0W )

# TMLE point estimate
PsiHat.TMLE<- mean(expY.given1W.star - expY.given0W.star)

#-----
# Return a list with the point estimates, targeted estimates of  $E_0(Y|A,W)$ ,
# and the vector of clever covariates
#-----

estimates <- data.frame(cbind(PsiHat.SS=PsiHat.SS, PsiHat.IPTW, PsiHat.TMLE))
predictions <- data.frame(cbind(expY.givenAW.star, expY.given1W.star, expY.given0W.star))
colnames(predictions) <- c('givenAW', 'given1W', 'given0W')
list(estimates=estimates, predictions=predictions, H.AW=H.AW)
}

```

```

out <- run.tmle(ObsData = ObsData, SL.library = SL.library)

```

```

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge

```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
est <- out$estimates
est
```

```
##      PsiHat.SS PsiHat.IPTW PsiHat.TMLE
## 1 0.01058352 -0.07048078 0.001769326
```

CV Superlearner

```
X<- subset(ObsData, select= -Y )

CV.SL.out<- CV.SuperLearner(Y=ObsData$Y, X=X,
                           SL.library=SL.library, family='binomial',
                           cvControl = list(V = 5),
                           innerCvControl = list(list(V = 20)))

summary(CV.SL.out)
```

```
##
## Call:
## CV.SuperLearner(Y = ObsData$Y, X = X, family = "binomial", SL.library = SL.library,
##   cvControl = list(V = 5), innerCvControl = list(list(V = 20)))
##
## Risk is based on: Mean Squared Error
##
## All risk estimates are based on V = 5
##
##           Algorithm      Ave      se      Min      Max
##      Super Learner 0.10958 0.025377 0.020846 0.16398
##      Discrete SL 0.11054 0.026048 0.022500 0.16285
##      SL.glm_All 0.11662 0.025487 0.035708 0.17882
## SL.glm.interaction_All 0.26011 0.043939 0.150000 0.45000
##      SL.step_All 0.11476 0.025447 0.026246 0.17515
##      SL.randomForest_All 0.11643 0.024761 0.032584 0.16285
##      SL.step.forward_All 0.11452 0.026196 0.022500 0.17515
##      SL.stepAIC_All 0.10819 0.025367 0.022500 0.17000
##      SL.mean_All 0.10819 0.025367 0.022500 0.17000
```

```
CV.SL.out$whichDiscrete
```

```
## $'1'
## [1] "SL.mean_All"
##
## $'2'
## [1] "SL.randomForest_All"
```



```
##
## $'3'
## [1] "SL.stepAIC_All"
##
## $'4'
## [1] "SL.randomForest_All"
##
## $'5'
## [1] "SL.mean_All"
```

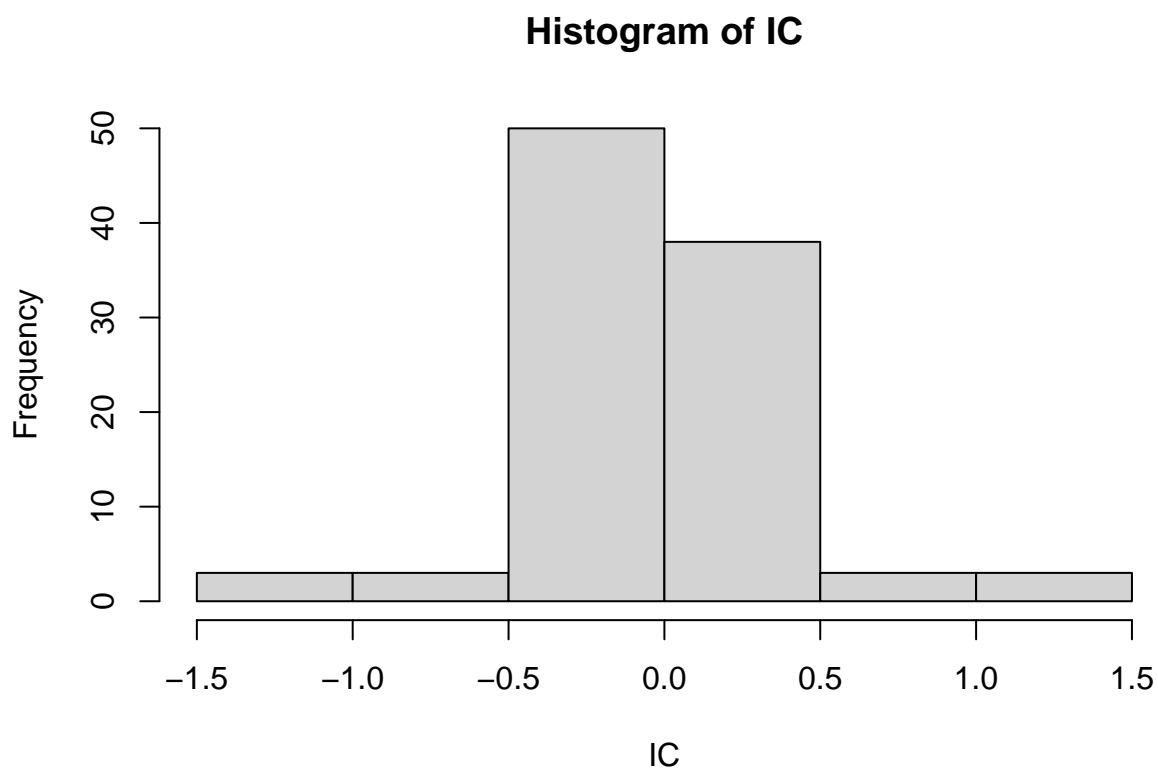
Influence Curve

```
n <- nrow(ObsData)
# clever covariate
H.AW <- out$H.AW
# targeted predictions
expY.AW.star <- out$predictions[, 'givenAW']
expY.1W.star <- out$predictions[, 'given1W']
expY.0W.star <- out$predictions[, 'given0W']
# point estimate
PsiHat.TMLE <- est$PsiHat.TMLE

# plug-in
IC <- H.AW*(ObsData$Y - expY.AW.star) + expY.1W.star - expY.0W.star - PsiHat.TMLE
summary(IC)
```

```
##          V1
## Min.      :-1.28288
## 1st Qu.   :-0.13426
## Median   :-0.09063
## Mean      : 0.00000
## 3rd Qu.   : 0.16248
## Max.      : 1.35461
```

```
hist(IC)
```



```
# estimate sigma^2 with the variance of the IC divided by n
varHat.IC <- var(IC)/n
varHat.IC
```

```
##           [,1]
## [1,] 0.001530302
```

```
# standard error estimate
se <- sqrt(varHat.IC)
se
```

```
##           [,1]
## [1,] 0.03911908
```

```
##### TMLE
```

```
# obtain 95% two-sided confidence intervals TMLE:
alpha <- 0.05
c(PsiHat.TMLE+qnorm(alpha/2, lower.tail=T)*se,
  PsiHat.TMLE+qnorm(alpha/2, lower.tail=F)*se)
```

```
## [1] -0.07490266 0.07844131
```

```
# calculate the pvalue tmle
2* pnorm( abs(PsiHat.TMLE /se), lower.tail=F )
```

```
##           [,1]
## [1,] 0.9639246
```

```
##### IPTW
```

```
PsiHat.IPTW <- est$PsiHat.IPTW
```

```
# obtain 95% two-sided confidence intervals TMLE:
```

```
alpha <- 0.05
c(PsiHat.IPTW+qnorm(alpha/2, lower.tail=T)*se,
  PsiHat.IPTW+qnorm(alpha/2, lower.tail=F)*se)
```

```
## [1] -0.147152765 0.006191208
```

```
# calculate the pvalue tmle
2* pnorm( abs(PsiHat.IPTW /se), lower.tail=F )
```

```
##           [,1]
## [1,] 0.07159288
```

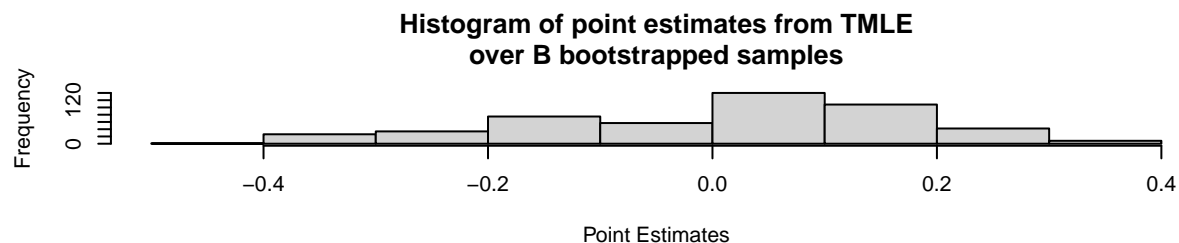
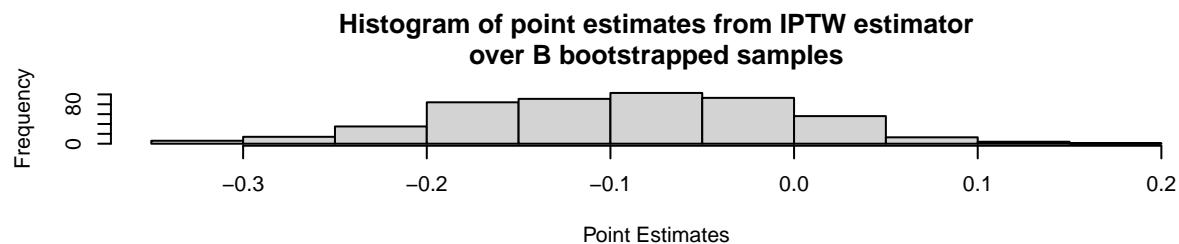
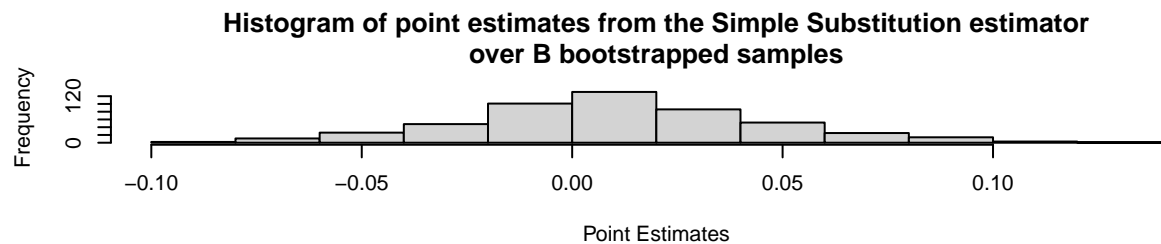
Non-parametric bootstrap

```
load('boot_par.Rdata')
```

```
summary(estimates)
```

```
##      SimpSubs      IPTW      TMLE
## Min.      :-0.088893 Min.      :-0.32983 Min.      :-0.402617
## 1st Qu.: -0.012123 1st Qu.: -0.15514 1st Qu.: -0.122351
## Median : 0.007485 Median : -0.08800 Median : 0.038971
## Mean    : 0.010020 Mean     :-0.09301 Mean     : 0.008726
## 3rd Qu.: 0.032206 3rd Qu.: -0.03223 3rd Qu.: 0.127827
## Max.    : 0.129803 Max.     : 0.17234 Max.     : 0.376132
##                                     NA's      :9
```

```
par(mfrow=c(3,1))
hist(estimates[,1], main="Histogram of point estimates from the Simple Substitution estimator
over B bootstrapped samples", xlab="Point Estimates")
hist(estimates[,2], main="Histogram of point estimates from IPTW estimator
over B bootstrapped samples", xlab="Point Estimates")
hist(estimates[,3], main="Histogram of point estimates from TMLE
over B bootstrapped samples", xlab="Point Estimates")
```



```
#-----
# 95% Confidence intervals assuming a normal dist & via quantiles
#-----
create.CI <- function(pt, boot, alpha=0.05){
  Zquant <- qnorm(alpha/2, lower.tail=F)
  CI.normal <- c(pt - Zquant*sd(boot,na.rm = TRUE),
                 pt + Zquant*sd(boot,na.rm = TRUE) )
  CI.quant <- quantile(boot, prob=c(0.025,0.975) ,na.rm=TRUE)
  out <- data.frame(rbind(CI.normal, CI.quant))
  colnames(out) <- c('CI.lo', 'CI.hi')
  out
}

# IMPORTANT - POINT OF CONFUSION FOR PAST STUDENTS
# The point estimate 'pt' is from the original dataset

# Simple Subs - note the bias because of misspecified regression? Will it converge fast enough?
est$PsiHat.SS

## [1] 0.01058352

create.CI(pt=est$PsiHat.SS, boot=estimates[, "SimpSubs"])

##           CI.lo      CI.hi
## CI.normal -0.05910387 0.08027091
## CI.quant  -0.05987893 0.08625440
```

```

# IPTW
est$PsiHat.IPTW

## [1] -0.07048078

create.CI(pt=est$PsiHat.IPTW, boot=estimates[, "IPTW"])

##           CI.lo      CI.hi
## CI.normal -0.2417312 0.1007696
## CI.quant  -0.2721505 0.0668197

# TMLE
est$PsiHat.TMLE

## [1] 0.001769326

create.CI(pt=est$PsiHat.TMLE, boot=estimates[, "TMLE"])

##           CI.lo      CI.hi
## CI.normal -0.3229254 0.3264641
## CI.quant  -0.3615892 0.2772986

# Compare to IC estimate
c(PsiHat.TMLE+qnorm(alpha/2, lower.tail=T)*se,
  PsiHat.TMLE+qnorm(alpha/2, lower.tail=F)*se)

## [1] -0.07490266  0.07844131

```