610 Final project

Ariane Stark, Minsu Kim, Diezhang Wu, Alona Muzikansky

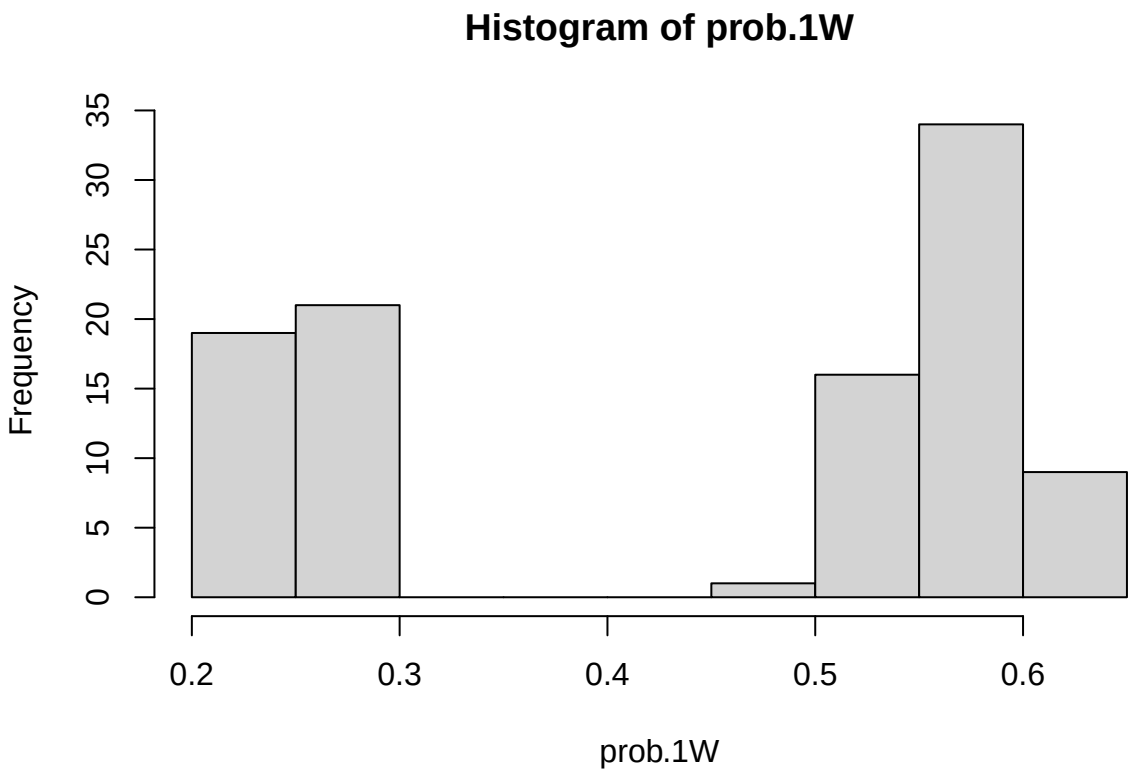
11/6/2021

Alona

```
# 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
```

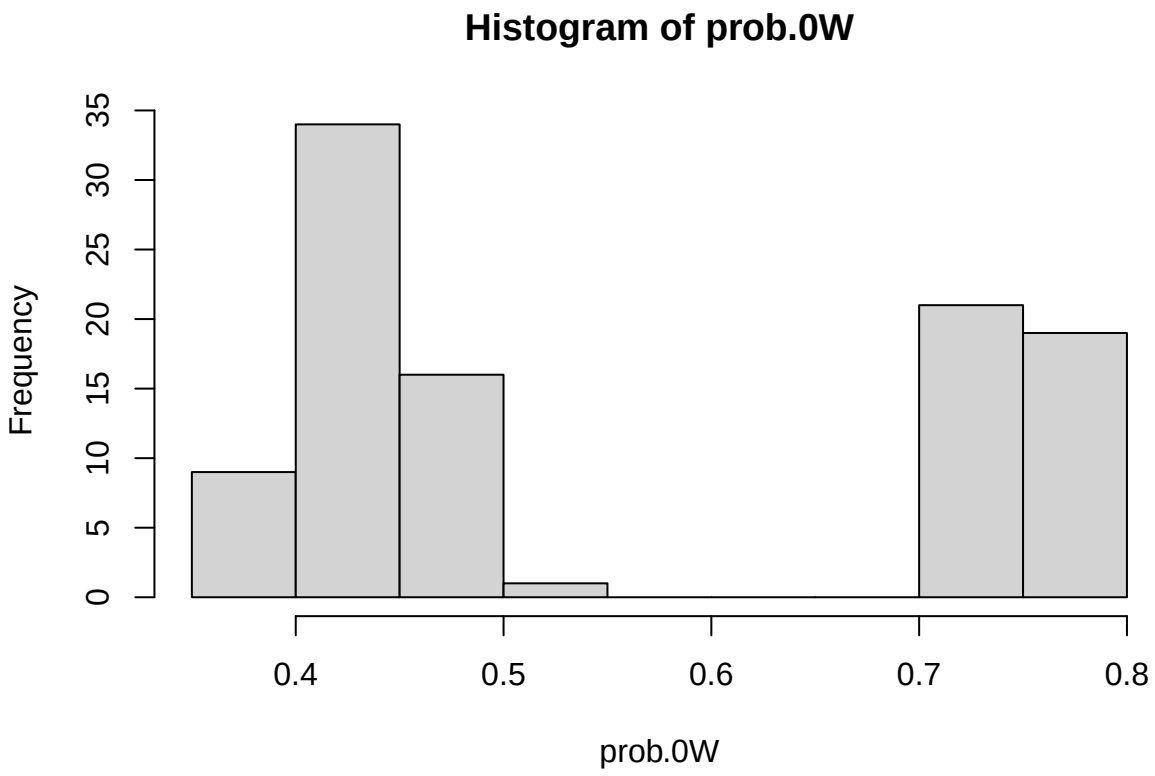
```
# IPTW estimator
prob.AW.reg <- glm(A ~ W11 + W12 + W13 + W14, family="binomial", data=ObsData)
prob.1W <- predict(prob.AW.reg, type= "response")
prob.0W <- 1 - prob.1W

hist(prob.1W)
```




```
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)
```



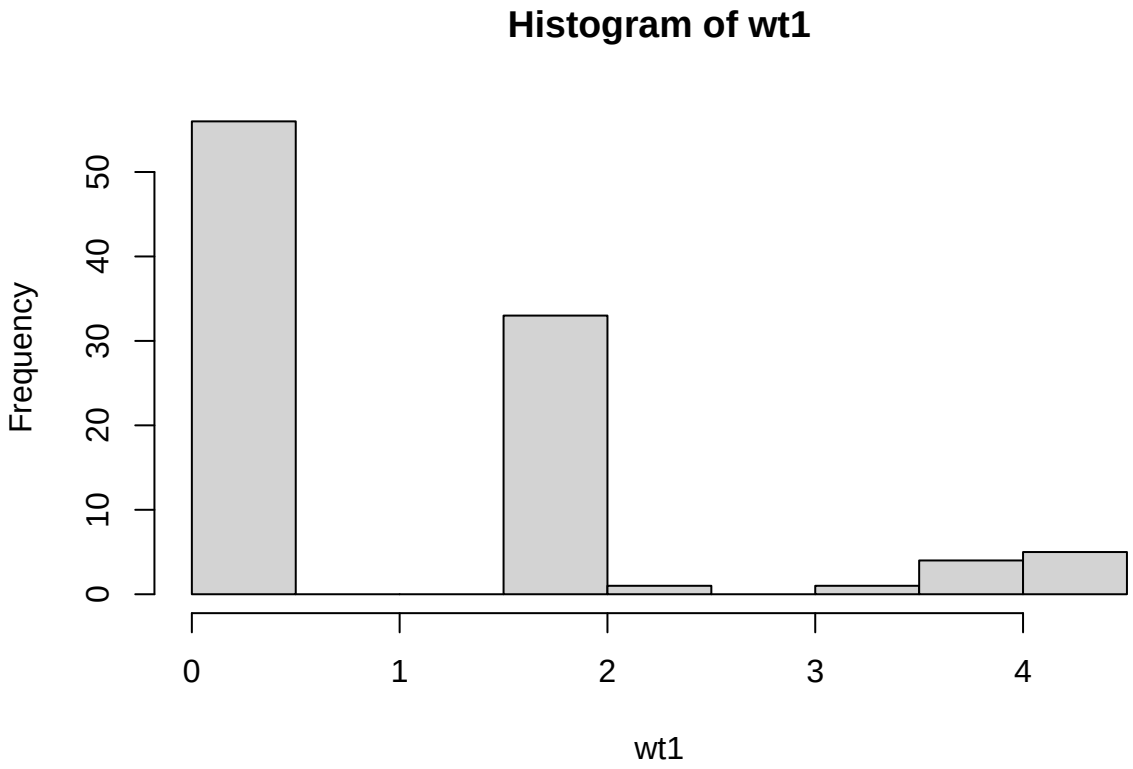

```
summary(prob.0W)
```

```
##    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
```

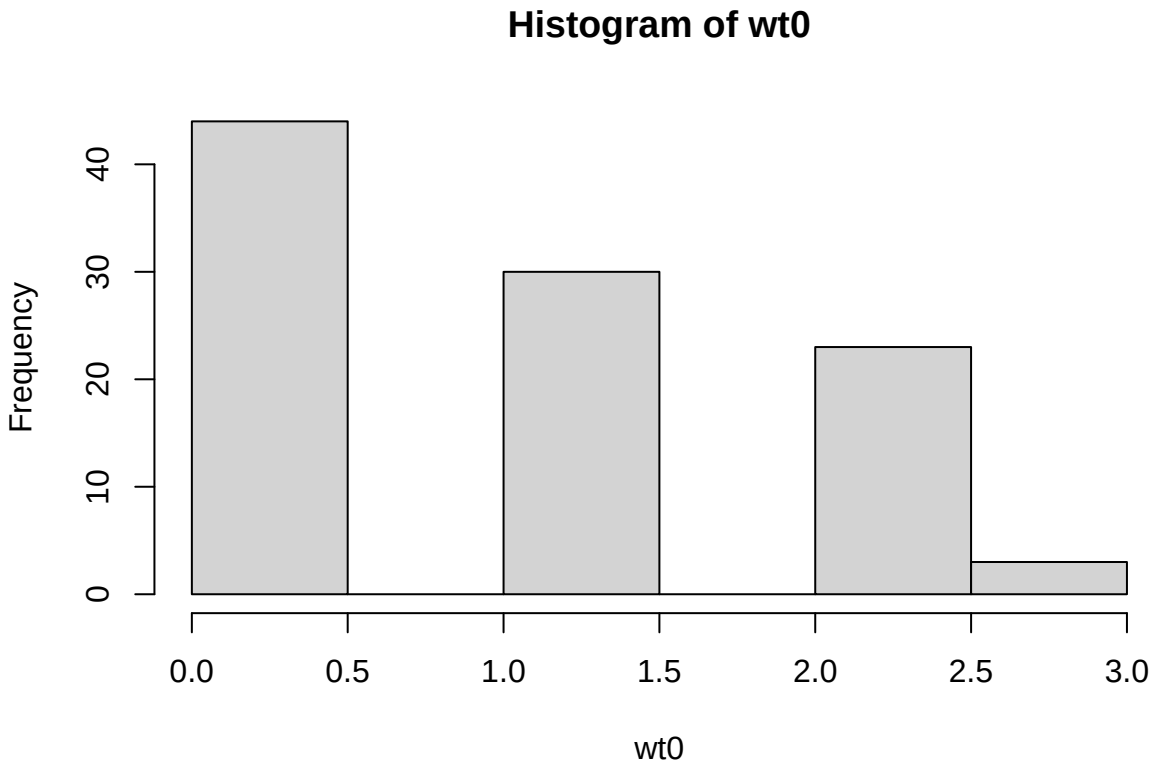
```
hist(wt1)
```




```
summary(wt0)
```

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

```
hist(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

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

```r
run.tmle <- 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$A <- 1     # exposed ('good guy')
  X0$A <- 0 # unexposed (not a 'good guy')

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

  # 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 txt 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")

  # 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)

  #-----------------------------------------
  # 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 esimates as offset
  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 withthe 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 -0.06509653 0.005638869
```