R Assignment 4- Part B

PH252D Fall 2013
Introduction to Causal Inference

Assigned: November 20, 2013
Due: November 27, 2013
Write-up: Please answer all questions and include relevant `R` code. You are encouraged to discuss the assignment in groups, but should not copy code or interpretations verbatim.

1 Background and Causal Road Map

"Russian Health Campaign Allows Train Users In Moscow To Pay In Squats"
The Huffington Post UK
"Want a free journey on the Tube in Moscow? Drop down and give 30 squats. In an effort to promote the upcoming Winter Olympic Games in Sochi, Moscow city officials and the Russian Olympic Committee are allowing subway riders to sweat it out to get to work. Instead of paying the regular 30 rubles (57p), commuters can now perform 30 squats at Vystavochnaya station..."
http://www.huffingtonpost.co.uk/2013/11/12/russia-moscow-train-squats_n_4260746.html

Consider a hypothetical intervention on the BART system, where riders will be given discounted tickets based on the number burpees they can properly complete (http://en.wikipedia.org/wiki/Burpee_(exercise)). For simplicity assume the minimum is 1 burpee and maximum is 7 burpees. The goal is to estimate the effect of burpees performed on rider's happiness, which is measured on a validated scale from 30-45. We have data on the the following variables

- $W1$: lifestyle with 1 as very active, 2 as somewhat active and 3 as sedentary
- $W2$: gender with 1 as female and 0 as male
- A: number of burpees completed with 1 as the minimum and 7 as the maximum
- Y: happiness which is a continuous scale from 30-45

2 Implement TMLE for the G-Computation estimand

Suppose we are interested in the difference in the expected counterfactual happiness if all riders completed 7 burpees ($A = 7$) and if all riders only completed 1 burpee ($A = 1$):

$$\Psi^F(P_{U,X}) = E_{U,X}(Y_7) - E_{U,X}(Y_1)$$

Under the necessary causal assumptions (i.e. the backdoor criteria and the positivity assumption), the corresponding statistical estimand is given by the G-Computation formula:

$$\begin{aligned}\Psi(P_0) &= E_0\big[E_0(Y|A=7,W) - E_0(Y|A=1,W)\big] \\ &= \sum_w \big[E_0(Y|A=7,W=w) - E_0(Y|A=1,W=w)\big] P_0(W=w)\end{aligned}$$

Please note that the following code is actually estimating a conditional parameter:

$$\Psi'(P_0) = \sum_w \big[E_0(Y|A=7,W=w) - E_0(Y|A=1,W)\big] P_0(W=w|A=1 \textbf{ or } A=7)$$

This is the G-Computation formula for a target population of BART riders, who complete $A = 1$ ***or*** $A = 7$ ***burpees. (This also has implications for the causal parameter and identifiability; everything is with respect to this subpopulation.) To recover the G-Computation formula for the target population of BART riders (equal to the marginal ATE under causal assumptions), we need to average over the marginal distribution of baseline covariates*** $P_0(W = w)$***.***

1. Set the seed to 252.
2. Import the data set `Rassign4.Fa2013.csv` and assign it to object `FullData`.
3. Create data frame `ObsData`, consisting riders completing $A = 1$ or $A = 7$ burpees:

   ```
   > ObsData<- FullData[FullData$A==1 | FullData$A==7, ]
   ```

4. Assign the number of riders in `ObsData` to `n`.
5. Create a new exposure variable `A.binary`, which equals 1 for riders completing $A = 7$ burpees and equals 0 for riders completing $A = 1$ burpee.
6. Use the `table` function to make sure your code is correct.
7. Implement `tmle` using SuperLearner with the default library for initial estimation of $\bar{Q}_0(A, W)$ and $g_0(A|W)$. Be sure to specify the outcome (`Y=ObsData$Y`), the exposure (`A=A.binary`) and the covariates (`W=subset(ObsData, select=c(W1,W2))`).
8. Comment on the point estimates from IPTW (part A) and from TMLE.

Solution:

```
> # 1. Import the data
> FullData<- read.csv("Rassign4.Fa2013.csv")

> # 2. get the observations with A=1 or A=7
> ObsData<- FullData[FullData$A==1 | FullData$A==7, ]
> summary(ObsData)

       W1              W2               A               Y
 Min.   :1.000   Min.   :0.0000   Min.   :1.000   Min.   :31.49
 1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:1.000   1st Qu.:34.50
 Median :2.000   Median :0.0000   Median :7.000   Median :39.42
 Mean   :1.957   Mean   :0.3534   Mean   :4.414   Mean   :37.35
 3rd Qu.:3.000   3rd Qu.:1.0000   3rd Qu.:7.000   3rd Qu.:39.45
 Max.   :3.000   Max.   :1.0000   Max.   :7.000   Max.   :42.45

> # 3. number of indpt riders
> n<- nrow(ObsData)

> # 4. recode the exposure so that A.binary=1 corresponds to 7 burpees
> # and A.binary=0 corresponds to 1 burpee
> A.binary<- rep(0, n)
> A.binary[ObsData$A==7] <- 1
> # 5. check
> table(ObsData$A)

  1   7
100 132
```

```
> table(A.binary)

A.binary
  0   1
100 132

> # 6 load the tmle package
> library(tmle)
> #
> tmle.out <- tmle(Y=ObsData$Y, A=A.binary, W=subset(ObsData, select=c(W1,W2)))
> summary(tmle.out)

 Initial estimation of Q
        Procedure: SuperLearner
        Model:
                Y ~  SL.glm_All + SL.step_All + SL.glm.interaction_All

        Coefficients:
             SL.glm_All    0
            SL.step_All    0
        SL.glm.interaction_All    1

 Estimation of g (treatment mechanism)
        Procedure: SuperLearner
        Model:
                A ~  SL.glm_All + SL.step_All + SL.glm.interaction_All

        Coefficients:
             SL.glm_All    0
            SL.step_All    0.0911191
        SL.glm.interaction_All    0.9088809

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

 Estimation of g.Delta (missingness mechanism)
        Procedure: No missingness

 Bounds on g: ( 0.025 0.975 )

 Additive Effect
   Parameter Estimate:  7.0034
   Estimated Variance:  0.0032156
               p-value:  <2e-16
    95% Conf Interval: (6.8922, 7.1145)

> tmle.out$epsilon

          H0W           H1W
-0.0006569619  0.0006162064

> # the TMLE package uses a two-dimensional clever covariate.
```

The IPTW estimate without stabilized weights was 7.53 units. The IPTW estimate with stablized weights was 6.86 units. The point estimate (of the conditional parameter) from TMLE was 7.00 units. The true value is $\psi_0 = 6.99$ units.

To evaluate the performance of these estimators (e.g. bias and variance), we could draw another independent sample of size n, implement the 3 estimators and repeat many times. To evaluate the consistency of these estimators, we would draw multiple samples of increasing size n and implement the 3 estimators.

Solution: ***To estimate the standard (marginal) G-Computation estimand with TMLE, we would have to code it by hand.***

```
> # USING ALL THE DATA
> n<- nrow(FullData)
> #------------------------
> # 1.   Estimate Q_0(A,W) with Superlearner
> #----------------------------
> library("SuperLearner")
> # specify the library
> SL.library<- c("SL.glm", "SL.step", "SL.glm.interaction")

> # data frame X with baseline covariates and observed exposure
> X<-subset(FullData, select=c(A, W1, W2) )
> # create data frames with A=7 and A= 1
> X1 <- X0<-X
> X1$A<- 7 # will get pred outcome for ALL riders with A=7
> X0$A<- 1 # will get pred outcome for ALL riders with A=1
> # create newdata by stacking
> newdata<- rbind(X,X1,X0)

> # call superlearner - using all the data.
> Qinit<- SuperLearner(Y=FullData$Y, X=X, newX=newdata, SL.library=SL.library)

> # pred outcome given the observed A and W
> QbarAW <- Qinit$SL.predict[1:n]
> # pred outcome for each subject given A=7 and W
> Qbar1W <- Qinit$SL.predict[(n+1): (2*n)]
> # pred outcome for each subject given A=1 and W
> Qbar0W <- Qinit$SL.predict[(2*n+1): (3*n)]

> #  the simple substitution estimator would be
> PsiHat.SS<-mean(Qbar1W - Qbar0W)
> # note we are averaging over all W=w; not just for those BART riders with
> # A=1 or A=7
> PsiHat.SS

[1] 6.150732

> ########################
> # 2.  Estimate g_0(A|W)
> ########################
```

Since A has 7 levels, we could fit $g_0(A|W)$ with the multinomial regression given in Part A. Alternatively, we could use SuperLearner and fit A with an appropriate library (e.g. different multinomial regressions).

```
> library("nnet")
> gAW.reg <-multinom(A~ W1+W2, data=FullData)

# weights:  28 (18 variable)
initial  value 9729.550745
iter  10 value 8205.693797
iter  20 value 8032.285228
final  value 8018.904637
converged

> gAW.pred<- predict(gAW.reg, type="probs")

> # set up a vector for the predicted probabilities
> gAW <- rep(NA,n)
> # assign the appropriate predicted probabilities
> gAW[FullData$A==1] <- gAW.pred[FullData$A==1, "1"]
> gAW[FullData$A==2] <- gAW.pred[FullData$A==2, "2"]
> gAW[FullData$A==3] <- gAW.pred[FullData$A==3, "3"]
> gAW[FullData$A==4] <- gAW.pred[FullData$A==4, "4"]
> gAW[FullData$A==5] <- gAW.pred[FullData$A==5, "5"]
> gAW[FullData$A==6] <- gAW.pred[FullData$A==6, "6"]
> gAW[FullData$A==7] <- gAW.pred[FullData$A==7, "7"]

> # IPTW estimators given in part A.

> #------------
> # 3. Create the clever covariate H_n(A,W)$ for each subject
> # numerator is the indicator of the obsv txt.
> # denominator is the predicted probability of observed exp, given baseline cov
> #--------------
> # H.AW  = Ind(A=7)/P(A=7|W)  - Ind(A=1)/P(A=1|W)
> # so H.AW=0 for observations with A not equal to 1 or 7

> H.AW<- as.numeric(FullData$A==7)/gAW - as.numeric(FullData$A==1)/gAW
> # also want to evaluate the clever covariates at A=7 and A=1 for all subjects
> H.1W<- 1/(gAW.pred[,'7'])
> H.0W<- -1/(gAW.pred[,'1'])
> #
> #---------
> # 4. Update the initial estimate of Qbar_0(A,W)
> #------------
```

Thus far, we have considered binary outcomes when coding TMLE by hand. For binary outcomes and bounded continuous outcomes, the logistic fluctuation sub-model tends to behave more robustly than linear fluctuation sub-models. See Chapter 7 of Targeted Learning for discussion. For simplicity, we will use a linear fluctuation model:

$$\bar{Q}_n^{(0)}(A,W)(\epsilon) = \bar{Q}_n^{(0)}(A,W) + \epsilon H_n(A,W)$$

with the L2 loss function:

$$L(\bar{Q}) = \left(Y - \bar{Q}(A,W)\right)^2$$

```
> update<- glm(FullData$Y ~ -1 +offset(QbarAW) + H.AW)
> eps<- update$coef
> eps

       H.AW
0.003599122

> #  calc the predicted values for each subj under each txt
> QbarAW.star<- QbarAW + eps*H.AW
> Qbar1W.star<- Qbar1W + eps*H.1W
> Qbar0W.star<- Qbar0W + eps*H.0W

> # 5. Estimate Psi(P_0) as the emp mean of the difference in the pred
> # outcomes under A=1 and A=0
> PsiHat.TMLE<- mean(Qbar1W.star) - mean(Qbar0W.star)
> PsiHat.TMLE

[1] 7.061117
```

3 Evaluate the finite sample performance of TMLE

1. Set the seed to `set.seed(252)`. Create a vector `estimates` of size `R=500`.
2. Within a `for` loop, repeat the following `R=500` times.
   (a) Draw a sample of size 5000 independently from data generating process given in Section 2 of R assignment 4.
   (b) Create data frame `ObsData`, consisting riders completing $A = 1$ or $A = 7$ burpees.
   (c) Assign the number of riders in `ObsData` to `n`.
   (d) Create a new exposure variable `A.binary`, which equals 1 for riders completing $A = 7$ burpees and equals 0 for riders completing $A = 1$ burpee.
   (e) Implement `tmle` using SuperLearner with the default library for initial estimation of $\bar{Q}_0(A, W)$ and $g_0(A|W)$.
   (f) Save the resulting point estimate as a row in `estimates`
3. What is the average estimate? What is the bias, average deviation from the point estimate and the true value? How variable are the estimates?
4. Create a histogram of the point estimates.
   Hint: Use `hist` function.

Solution:

```
> # ----------------
> # generateData - function to generate the observed data + counterfactuals
> # ----------------
> # this does (should) NOT need to be in the for loop.
> generateData<- function(n){
+
```

```
+   W1<- as.integer(runif(n, 1,4)  ) # lifestyle 1,2,3
+   W2<- rbinom(n, size=1, prob= runif(n, 0.02, 0.7)) # gender
+   A<- 1+ rbinom(n, size=6, prob=plogis(0.35 -0.3*W1 +0.5*(1-W2) )) #burpees
+   U.Y<- rnorm(n, 0, sd=0.01)
+   Y<- 30 +1.5*W1 +3*log(A)+.3*(1-W2)*A + U.Y # happiness
+
+   #  the counterfactuals
+   Y.1<- 30 +1.5*W1 +3*log(1)+.3*(1-W2)*1 + U.Y #
+   Y.2<- 30 +1.5*W1 +3*log(2)+.3*(1-W2)*2 + U.Y #
+   Y.3<- 30 +1.5*W1 +3*log(3)+.3*(1-W2)*3 + U.Y #
+   Y.4<- 30 +1.5*W1 +3*log(4)+.3*(1-W2)*4 + U.Y #
+   Y.5<- 30 +1.5*W1 +3*log(5)+.3*(1-W2)*5 + U.Y #
+   Y.6<- 30 +1.5*W1 +3*log(6)+.3*(1-W2)*6 + U.Y #
+   Y.7<- 30 +1.5*W1 +3*log(7)+.3*(1-W2)*7 + U.Y #
+
+   data.frame(W1,W2,A,Y,Y.1, Y.2, Y.3, Y.4,Y.5, Y.6, Y.7)
+ }
> # ----------------

> set.seed(252)
> R=500
> estimates<- rep(NA, R) # for G-Comp estimates of subpop doing A=1 or A=7 burpees
> # for loop time
> for(r in 1:R){
+   FullData<- generateData(n=5000)
+   ObsData<- FullData[FullData$A==1 | FullData$A==7, ]
+   # number of indpt riders
+   n<- nrow(ObsData)
+
+   # recode the exposure so that A.binary=1 corresponds to 7 burpees
+   # and A.binary=0 corresponds to 1 burpee
+   A.binary<- rep(0, n)
+   A.binary[ObsData$A==7] <- 1
+   # tmle
+   tmle.out <- tmle(Y=ObsData$Y, A=A.binary, W=subset(ObsData, select=c(W1,W2)))
+   estimates[r]<- tmle.out$estimates$ATE$psi
+ }
> true<- generateData(n=100000)
> psi.F <- mean(true$Y.7) - mean(true$Y.1)
> psi.F

[1] 6.990234

> # equal to Psi.P0 - under causal assumptions (all exogenous factors are independent)
> # so the back door criteria holds conditional on W=(W1,W2)

> # average value
> mean(estimates)

[1] 6.993928

> # bias
> mean(estimates- psi.F)
```

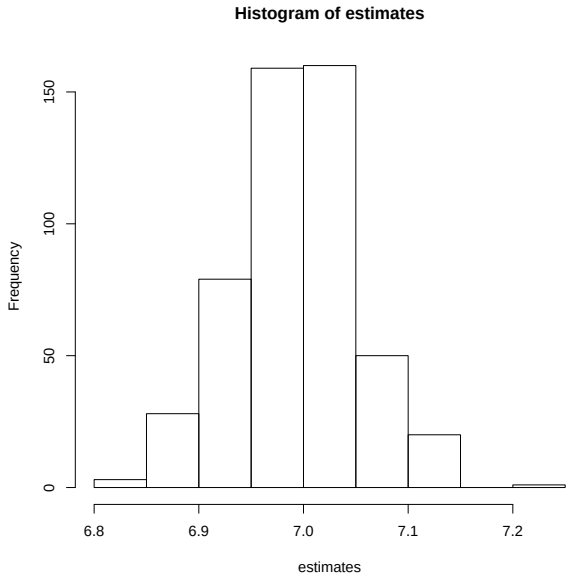
```
[1] 0.003693489

> # variance
> var(estimates)

[1] 0.00344688

> # create pdf of the histogram
> pdf(file="HistAss4b.pdf")
> hist(estimates)
> dev.off()
```

You could also try this when the conditional mean function $\bar{Q}_0(A, W)$ is correctly specified, the treatment mechanism $g_0(A|W)$ is misspecified and vice versa.



Solution Fig. 1: Histogram of point estimates.

Solution: Here is the analogous code for the standard (marginal) G-Computation formula.

```
> set.seed(252)
> R= 500
> estimates.b<- rep(NA, R)
> # for loop time
> for(r in 1:R){
+   FullData<- generateData(n=5000)
+   n<- nrow(FullData)
+   # 1.   Estimate Q_0(A,W) with Superlearner
+   # data frame X with baseline covariates and observed exposure
+   X<-subset(FullData, select=c(A, W1, W2) )
+
+   # create data frames with A=7 and A= 1
```

```
+   X1 <- X0<-X
+   X1$A<- 7 # will get pred outcome for ALL riders with A=7
+   X0$A<- 1 # will get pred outcome for ALL riders with A=1
+   # create newdata by stacking
+   newdata<- rbind(X,X1,X0)
+   Qinit<- SuperLearner(Y=FullData$Y, X=X, newX=newdata, SL.library=SL.library)
+
+   # pred outcome given the observed A and W
+   QbarAW <- Qinit$SL.predict[1:n]
+   # pred outcome for each subject given A=7 and W
+   Qbar1W <- Qinit$SL.predict[(n+1): (2*n)]
+   # pred outcome for each subject given A=1 and W
+   Qbar0W <- Qinit$SL.predict[(2*n+1): (3*n)]
+
+   # 2.  Estimate g_0(A|W)
+   gAW.reg <- multinom(A~ W1+W2, data=FullData)
+   gAW.pred<- predict(gAW.reg, type="probs")
+
+   # set up a vector for the predicted probabilities
+   gAW <- rep(NA,n)
+   # assign the appropriate predicted probabilities
+   gAW[FullData$A==1] <- gAW.pred[FullData$A==1, "1"]
+   gAW[FullData$A==2] <- gAW.pred[FullData$A==2, "2"]
+   gAW[FullData$A==3] <- gAW.pred[FullData$A==3, "3"]
+   gAW[FullData$A==4] <- gAW.pred[FullData$A==4, "4"]
+   gAW[FullData$A==5] <- gAW.pred[FullData$A==5, "5"]
+   gAW[FullData$A==6] <- gAW.pred[FullData$A==6, "6"]
+   gAW[FullData$A==7] <- gAW.pred[FullData$A==7, "7"]
+
+   # 3. H.AW  = Ind(A=7)/P(A=7|W)  - Ind(A=1)/P(A=1|W)
+   H.AW<- as.numeric(FullData$A==7)/gAW - as.numeric(FullData$A==1)/gAW
+
+   # also want to evaluate the clever covariates at A=7 and A=1 for all subjects
+   H.1W<- 1/(gAW.pred[,'7'])
+   H.0W<- -1/(gAW.pred[,'1'])
+
+   # 4. Update the initial estimate of Qbar_0(A,W)
+   update<- glm(FullData$Y ~ -1 +offset(QbarAW) + H.AW)
+   eps<- update$coef
+
+   #  calc the predicted values for each subj under each txt
+   QbarAW.star<- QbarAW + eps*H.AW
+   Qbar1W.star<- Qbar1W + eps*H.1W
+   Qbar0W.star<- Qbar0W + eps*H.0W
+
+   # 5. Estimate Psi(P_0) as the emp mean of the difference in the pred
+   # outcomes under A=1 and A=0
+   estimates.b[r]<- mean(Qbar1W.star) - mean(Qbar0W.star)
+ }

> # average value
> mean(estimates.b)

[1] 7.057271
```

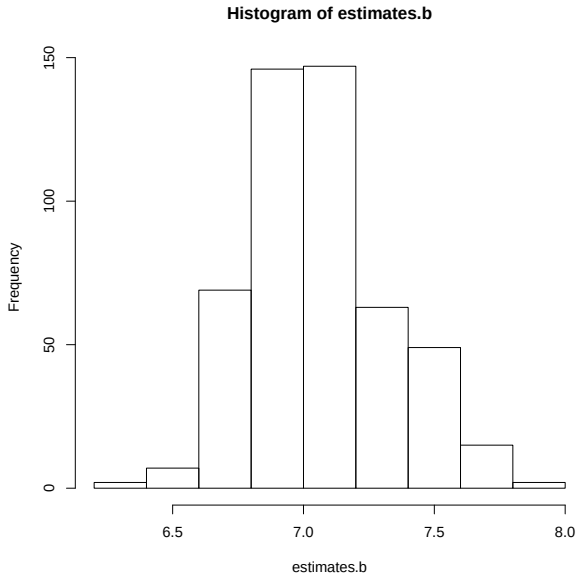
```
> # bias
> mean(estimates.b- psi.F)

[1] 0.06703677

> # variance
> var(estimates.b)

[1] 0.07155768

> # create pdf of the histogram
> pdf(file="HistAss4b2.pdf")
> hist(estimates.b)
> dev.off()
```



Solution Fig. 2: Histogram of point estimates.