Bob Bell

10.28.2013

PH252D

**R Assignment #2**

# 2.1 Evaluate Positivity Assumption in closed form

W1 <- 1; W2 <- 1

plogis(-0.5 + W1 - 1.5\*W2)

[1] 0.2689414

W1 <- 1; W2 <- 0

plogis(-0.5 + W1 - 1.5\*W2)

[1] 0.6224593

W1 <- 0; W2 <- 1

plogis(-0.5 + W1 - 1.5\*W2)

[1] 0.1192029

W1 <- 0; W2 <- 0

plogis(-0.5 + W1 - 1.5\*W2)

[1] 0.3775407

All probabilities are between 0 and 1. Therefore, we do not violate the positivity assumption.

# 2.2 Evaluate Statistical Estimand in Closed Form

P0.W1W2 <- 0.25

W1 <- 0; W2 <- 0

E0.00 <- plogis(-0.75 + W1 - 2\*W2 + 2.5\*1 + 1\*W1) - plogis(-0.75 + W1 - 2\*W2 + 2.5\*0 + 0\*W1)

W1 <- 0; W2 <- 1

E0.01 <- plogis(-0.75 + W1 - 2\*W2 + 2.5\*1 + 1\*W1) - plogis(-0.75 + W1 - 2\*W2 + 2.5\*0 + 0\*W1)

W1 <- 1; W2 <- 0

E0.10 <- plogis(-0.75 + W1 - 2\*W2 + 2.5\*1 + 1\*W1) - plogis(-0.75 + W1 - 2\*W2 + 2.5\*0 + 0\*W1)

W1 <- 1; W2 <- 1

E0.11 <- plogis(-0.75 + W1 - 2\*W2 + 2.5\*1 + 1\*W1) - plogis(-0.75 + W1 - 2\*W2 + 2.5\*0 + 0\*W1)

Psi.P0 <- P0.W1W2\*(E0.00 + E0.01 + E0.10 + E0.11)

Psi.P0

[1] 0.506905

#3 Translate data generating process into simulations

#3.1 Set seed to 252

set.seed(252)

#3.2 Set number of draws to 100,000

n=100000

#3.3 Sample n i.i.d. observations of random variable O

UW1 <- runif(n, min=0, max=1)

UW2 <- runif(n, min=0, max=1)

UA <- runif(n, min=0, max=1)

UY <- runif(n, min=0, max=1)

W1 <- as.numeric(UW1<0.50)

W2 <- as.numeric(UW2<0.50)

A <- as.numeric(UA<plogis(-0.5+W1-1.5\*W2))

Y <- as.numeric(UY<plogis(-0.75 + W1 - 2\*W2 + 2.5\*A + A\*W1))

Or one could alternatively simulate with the following code:

W1 <- rbinom(n, size=1, prob=0.50)

W2 <- rbinom(n, size=1, prob=0.50)

A <- rbinom(n, size=1, prob=plogis(-0.5+W1-1.5\*W2))

Y <- rbinom(n, size=1, prob=plogis(-0.75 + W1 - 2\*W2 + 2.5\*A + A\*W1))

#3.4 Intervene to set exposure to standard of care and generate counterfactual outcomes Y.0

Y.0 <- as.numeric(UY<plogis(-0.75 + W1 - 2\*W2 + 2.5\*0 + 0\*W1))

#Intervene to set exposure to combination package and generate counterfactual outcomes Y.1

Y.1 <- as.numeric(UY<plogis(-0.75 + W1 - 2\*W2 + 2.5\*1 + 1\*W1))

Or one could alternatively simulate with the following code:

Y.0<- rbinom(n, size=1, prob=plogis(-0.75 + W1 - 2\*W2 + 2.5\*0 + 0\*W1))

Y.1<- rbinom(n, size=1, prob=plogis(-0.75 + W1 - 2\*W2 + 2.5\*1 + 1\*W1))

#Evaluate the causal parameter

Psi.F <- mean(Y.1-Y.0)

Psi.F

[1] 0.50707

When I evaluate with the alternative code (assuming I start with the same random number seed), I get the value:

[1] 0.50577

#3.5 Evaluate Positivity Assumption

A.00 <- as.numeric(UA<plogis(-0.5+0-1.5\*0))

mean(A.00)

[1] 0.37656

A.10 <- as.numeric(UA<plogis(-0.5+1-1.5\*0))

mean(A.10)

[1] 0.62032

A.01 <- as.numeric(UA<plogis(-0.5+0-1.5\*1))

mean(A.01)

[1] 0.11904

A.11 <- as.numeric(UA<plogis(-0.5+1-1.5\*1))

mean(A.11)

[1] 0.26971

Alternative code:

Pos.A.00 <- mean(rbinom(n, size=1, prob=plogis(-0.5+0-1.5\*0)))

Pos.A.00

[1] 0.37869

Pos.A.10 <- mean(rbinom(n, size=1, prob=plogis(-0.5+1-1.5\*0)))

Pos.A.10

[1] 0.62209

Pos.A.01 <- mean(rbinom(n, size=1, prob=plogis(-0.5+0-1.5\*1)))

Pos.A.01

[1] 0.11995

Pos.A.11 <- mean(rbinom(n, size=1, prob=plogis(-0.5+1-1.5\*1)))

Pos.A.11

[1] 0.26948

There are no positivity violations.

(Note here and for the rest of the code, I assume the original (not alternative) code based on explicitly coding background variables (U’s).)

#3.6 Evaluate Statistical Estimand Psi.P0

> new\_W1<-0; new\_W2<-0

> sim\_E0.00 <- (as.numeric(UY<plogis(-0.75 + new\_W1 - 2\*new\_W2 + 2.5\*1 + 1\*new\_W1)) - as.numeric(UY<plogis(-0.75 + new\_W1 - 2\*new\_W2 + 2.5\*0 + 0\*new\_W1)))\*mean(W1)\*mean(W2)

> new\_W1<-0; new\_W2<-1

> sim\_E0.01 <- (as.numeric(UY<plogis(-0.75 + new\_W1 - 2\*new\_W2 + 2.5\*1 + 1\*new\_W1)) - as.numeric(UY<plogis(-0.75 + new\_W1 - 2\*new\_W2 + 2.5\*0 + 0\*new\_W1)))\*mean(W1)\*mean(W2)

> new\_W1<-1; new\_W2<-0

> sim\_E0.10 <- (as.numeric(UY<plogis(-0.75 + new\_W1 - 2\*new\_W2 + 2.5\*1 + 1\*new\_W1)) - as.numeric(UY<plogis(-0.75 + new\_W1 - 2\*new\_W2 + 2.5\*0 + 0\*new\_W1)))\*mean(W1)\*mean(W2)

> new\_W1<-1; new\_W2<-1

> sim\_E0.11 <- (as.numeric(UY<plogis(-0.75 + new\_W1 - 2\*new\_W2 + 2.5\*1 + 1\*new\_W1)) - as.numeric(UY<plogis(-0.75 + new\_W1 - 2\*new\_W2 + 2.5\*0 + 0\*new\_W1)))\*mean(W1)\*mean(W2)

> Psi.P0 <- mean(sim\_E0.00 + sim\_E0.01 + sim\_E0.10 + sim\_E0.11)

> Psi.P0

[1] 0.5054506

3.7. Interpretation: Our analysis was based on giving every child in our population sample the standard of care (A=0) and then rolling back time and giving every child in our population sample the intervention package (A=1). We take the mean difference of the counterfactual outcomes within strata of W1 and W2. Y is a binary variable, where I interpret Y=1 as survival and Y=0 as mortality. The mean difference across all strata, according to the G-Computation formula, is ~0.505. This represents approximately a 50% increase in the average survival status among children at the end of two years due to the combination/intervention package. This interpretation assumes identifiability of target causal quantity.

#4 Simple Substitution Estimator based on G-Computation

#4.1 Set number of iterations R and number of observations n

R <- 500

n <- 200

#4.2 Create Rx4 matrix estimates

estimates <- matrix(NA, nrow=R, ncol=4)

#4.3 For Loop

for(r in 1:R){

#4.3.a. sample of n=200 i.i.d. observations of O

UW1 <- runif(n, min=0, max=1)

UW2 <- runif(n, min=0, max=1)

UA <- runif(n, min=0, max=1)

UY <- runif(n, min=0, max=1)

W1 <- as.numeric(UW1<0.50)

W2 <- as.numeric(UW2<0.50)

A <- as.numeric(UA<plogis(-0.5+W1-1.5\*W2))

Y <- as.numeric(UY<plogis(-0.75 + W1 - 2\*W2 + 2.5\*A + A\*W1))

# 4.3.b.Create dataframe Obs of resulting observed data

Obs<- data.frame(W1,W2,A,Y)

#4.3.c. Copy the original dataset Obs into two new dataframes txt and control.

txt<- control <- Obs

# set A=1 in the txt dataframe and A=0 in control dataframe

txt$A <-1

control$A <- 0

#4.3.d. Estimator 1

reg.model1<- glm(Y ~ A, family='binomial', data=Obs)

#4.3.e. Estimator 2

reg.model2<- glm(Y ~ A + W1, family='binomial', data=Obs)

#4.3.f. Estimator 3

reg.model3<- glm(Y ~ A + W2, family='binomial', data=Obs)

#4.3.g. Estimator 4

reg.model4<- glm(Y ~ A + W1 + W2 + A:W1 + A:W2, family='binomial', data=Obs)

#4.h For each estimator predict expected outcome for each individual in the sample under the treatment

Y1\_model1.predict<- predict(reg.model1, newdata = txt, type='response')

Y1\_model2.predict<- predict(reg.model2, newdata = txt, type='response')

Y1\_model3.predict<- predict(reg.model3, newdata = txt, type='response')

Y1\_model4.predict<- predict(reg.model4, newdata = txt, type='response')

#4.i For each estimator predict expected outcome for each individual in the sample under the control

Y0\_model1.predict<- predict(reg.model1, newdata = control, type='response')

Y0\_model2.predict<- predict(reg.model2, newdata = control, type='response')

Y0\_model3.predict<- predict(reg.model3, newdata = control, type='response')

Y0\_model4.predict<- predict(reg.model4, newdata = control, type='response')

#4.j take the mean of the predicted outcomes

psi.hat1 <- mean(Y1\_model1.predict - Y0\_model1.predict)

psi.hat2 <- mean(Y1\_model2.predict - Y0\_model2.predict)

psi.hat3 <- mean(Y1\_model3.predict - Y0\_model3.predict)

psi.hat4 <- mean(Y1\_model4.predict - Y0\_model4.predict)

#4.k Assign estimates to matrix estimates

estimates[r,] <- c(psi.hat1, psi.hat2, psi.hat3, psi.hat4)

}

#5 Performance of the estimators

#5.1 Average value of each estimator

meanEst1 <- mean(estimates[,1])

meanEst2 <- mean(estimates[,2])

meanEst3 <- mean(estimates[,3])

meanEst4 <- mean(estimates[,4])

meanEst1

[1] 0.6519154

meanEst2

[1] 0.6243597

meanEst3

[1] 0.5653346

meanEst4

[1] 0.5025224

#5.2 Bias of each estimator

bias1<- mean(estimates[,1] - Psi.P0)

bias2<- mean(estimates[,2] - Psi.P0)

bias3<- mean(estimates[,3] - Psi.P0)

bias4<- mean(estimates[,4] - Psi.P0)

bias1

[1] 0.1464649

bias2

[1] 0.1189092

bias3

[1] 0.05988406

bias4

[1] -0.002928145

#5.3 Variance of each estimator

var1<- var(estimates[,1])

var2<- var(estimates[,2])

var3<- var(estimates[,3])

var4<- var(estimates[,4])

var1

[1] 0.003049491

var2

[1] 0.003401361

var3

[1] 0.004598521

var4

[1] 0.005604145

#5.4 MSE for each estimator

mse1<- mean( (estimates[,1]-Psi.P0)^2)

mse2<- mean( (estimates[,2]-Psi.P0)^2)

mse3<- mean( (estimates[,3]-Psi.P0)^2)

mse4<- mean( (estimates[,4]-Psi.P0)^2)

mse1

[1] 0.02449535

mse2

[1] 0.01753395

mse3

[1] 0.008175424

mse4

[1] 0.005601511

#5.5

I expected the simplest substitution estimator (#1) and the most complicated (#4 with multiple interaction terms) to do the worst and best, respectively. This analysis confirmed my intuition. Here, I define best estimator as the one that has the minimum MSE (compared to the other estimators) with respect to the true Psi.P0. The estimator with the lowest MSE was Estimator #4.

However, I was surprised by the differences in the MSE’s of Estimators #2 and #3 related to the inclusion of either covariate. I initially expected an estimator including the availability of healthcare facilities (#2) to better estimate the target causal parameter given by the true, unknown data distribution P0. The elements of the prevention package (i.e., antimalarial tablets, mosquito nets, capability building with community workers, vaccination, etc.) seemed to be interventions that would take place given some measure of health infrastructure.

To see that the working MSM including only the covariate for recent conflict (#3) had a lower MSE than the one including only health facilities (#2) was interesting. But now it makes sense. If I return to the initial question of preventing malnutrition in Sahel, recent conflict is likely to be a better predictor for malnutrition than health care facilities. In a conflict-ridden situation, children (in addition to others in their surrounding locales) may not have access to food. And having a healthcare facility is less likely to prevent malnutrition as it exists to serve the health needs of the community (and not necessarily nutrition concerns).