# Causal Inference: R Assignment #2

*Janelle Downing*

*October 27, 2014*

**Part 2: A specific data generating process**

1. Evaluate the postivity assumption in closed form for this data generating process.

```
#For P0(A = 1|W1;W2) = expit(-0.5 + W1 - 1.5*W2)
#0 < P0(A = 1|W1 = 1;W2 = 1) < 1
plogis(-.5 + 1 - 1.5*1)
```

```
## [1] 0.2689414
```

```
#0 < P0(A = 1|W1 = 1;W2 = 0) < 1
plogis(-.5 + 1 - 1.5*0)
```

```
## [1] 0.6224593
```

```
#0 < P0(A = 1|W1 = 0;W2 = 1) < 1
plogis(-.5 + 0 - 1.5*1)
```

```
## [1] 0.1192029
```

```
#0 < P0(A = 1|W1 = 0;W2 = 0) < 1
plogis(-.5 + 0 - 1.5*0)
```

```
## [1] 0.3775407
```

*Result: There are no violations of the positivity assumption.*

2. Evaluate the statistical estimand in closed form.

```
(plogis(-.75+1-2+2.5+1) - plogis(-.75 + 1 - 2))*0.25 + (plogis(-.75+1+2.5*1+1)- plogis(-.75+1))*.25 + (
```

```
## [1] 0.506905
```

**Part 3: Translate this data generating process into simulations**

1. First set the seed to 252.

```
set.seed(252)
```

2. Set the numer of draws n = 100,000

```
n = 100000
```

3. Sample n i.i.d. observations of random variable O.

```
#Endogenous factors
U.W1 = runif(n, 0, 1)
U.W2 = runif(n, 0, 1)
U.A = runif(n, 0, 1)
U.Y = runif(n, 0, 1)
```

```
#Exogenous factors
W.1 = as.numeric(U.W1 < 0.5)
W.2 = as.numeric(U.W2 < 0.5)
A = as.numeric(U.A < plogis(-0.5 + W.1 - 1.5*W.2))
Y = as.numeric(U.Y < plogis(-0.75 + W.1 - 2*W.2 + 2.5*A + A*W.1))
```

```
#Make dataframe
X <- data.frame(W.1, W.2, A, Y)
head(X)
```

```
##   W.1 W.2 A Y
## 1   0   0 1 1
## 2   0   0 1 1
## 3   1   0 1 1
## 4   0   0 1 1
## 5   0   0 1 1
## 6   1   0 1 1
```

```
sum(X)
```

```
## [1] 179149
```

4. Bonus: Intervene to set the exposure to the combination package (A = 1) and generate the counterfactual outcome Y1. Intervene to set the exposure to the standard of care (A = 0) and generate the counterfactual outcomes Y0.

```
Y.1 <- as.numeric(U.Y < plogis(-0.75 + W.1 - 2*W.2 + 2.5*1 + 1*W.1))
Y.0 <- as.numeric(U.Y < plogis(-0.75 + W.1 - 2*W.2 + 2.5*0 + 0*W.1))
```

Evaluate the causal parameter.

```
psi.f <- mean(Y.1) - mean(Y.0)
psi.f
```

```
## [1] 0.50707
```

5. Evaluate the positivity assumption.

```
A.W1W1 <- A[W.1==1 & W.2==1]
mean(A.W1W1)
```

```
## [1] 0.271355
```

```
A.W1W0 <- A[W.1==1 & W.2==0]
mean(A.W1W0)
```

```
## [1] 0.6221695
```

```
A.W0W1 <- A[W.1==0 & W.2==1]
mean(A.W0W1)
```

```
## [1] 0.1190666
```

```
A.W0W0 <- A[W.1==0 & W.2==0]
mean(A.W0W0)
```

```
## [1] 0.3756981
```

6. Evaluate the statistical estimand and assign the value to Psi.PO

```
meanY.A1W1W1 <- mean(Y[A==1 & W.1==1 & W.2==1])
meanY.A1W1W0 <- mean(Y[A==1 & W.1==1 & W.2==0])
meanY.A1W0W1 <- mean(Y[A==1 & W.1==0 & W.2==1])
meanY.A1W0W0 <- mean(Y[A==1 & W.1==0 & W.2==0])
meanY.A0W1W1 <- mean(Y[A==0 & W.1==1 & W.2==1])
meanY.A0W1W0 <- mean(Y[A==0 & W.1==1 & W.2==0])
meanY.A0W0W1 <- mean(Y[A==0 & W.1==0 & W.2==1])
meanY.A0W0W0 <- mean(Y[A==0 & W.1==0 & W.2==0])
meanW1 <- mean(W.1)
meanW2 <- mean(W.2)
Psi.P0 <- (meanY.A1W1W1 - meanY.A0W1W1)*(meanW1*meanW2) + (meanY.A1W1W0 - meanY.A0W1W0)*(meanW1)*(1-mea
Psi.P0
```

```
## [1] 0.5039042
```

7. Interpret Psi.PO The difference in the strata-specific probability of survival under the intervention and under the control, averaged with respect to the distribution of access to healthcare facilities and conflict history is 0.504. Under the randomization assumption (if it held), (P0) could be interpreted as the causal risk difference: the probability of survival through the 2 years would be 50.4% higher under the intervention than without the intervention.

**Part 4: The simple substitution estimator based on the G-Computation formula**

1. Set the number of iterations R to 500 and the number of observations n to 200. Do not reset the seed.

```
R = 500
n = 200
```

2. Create a R = 500 by 4 matrix estimates to hold the resulting estimates obtained at each iteration.

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

3. Inside a for loop from r equals 1 to R (500), do the following.

```
for (i in 1:R){

        #a. Sample n i.i.d. observations of O
        W1 <-rbinom(n, size=1, prob=0.5)
        W2 <- rbinom(n, size=1, prob=0.5)
        A <- rbinom(n, size=1, prob=plogis(-0.5 + W1 -1.5*W2))
        Y <- rbinom(n, size=1, prob=plogis(-.75 + W1 - 2*W2 + 2.5*A + A*W1))

        #b. Create a data frame Obs of the resulting observed data.
        Obs <- data.frame(W1, W2, A, Y)

        #c. Copy the data set Obs into two new data frames txt and control.
        txt <- Obs
        control <- Obs
        #Then set A=1 for all unnits in txt and set A=0 for all units in the control.
        txt$A <- 1
        control$A<- 0

        #d. Estimator 1
        est1 <- glm(Y ~ A, family = 'binomial', data=Obs)

        #e. Estimator 2
        est2 <- glm(Y ~ A + W1, family = 'binomial', data=Obs)

        #f. Estimator 3
        est3 <- glm(Y ~ A + W2, family = 'binomial', data=Obs)

        #g. Estimator 4
        est4 <- glm(Y ~ A*W1 + A*W2, family = 'binomial', data=Obs)

        #h. Expected (mean) outcome for each unit under the intervention
        Y1.predict.est1<- predict(est1, newdata = txt, type='response')
        Y1.predict.est2<- predict(est2, newdata = txt, type='response')
        Y1.predict.est3<- predict(est3, newdata = txt, type='response')
        Y1.predict.est4<- predict(est4, newdata = txt, type='response')

        #i. Expected (mean) outcome for each unit under the control
        Y0.predict.est1<- predict(est1, newdata = control, type='response')
        Y0.predict.est2<- predict(est2, newdata = control, type='response')
        Y0.predict.est3<- predict(est3, newdata = control, type='response')
        Y0.predict.est4<- predict(est4, newdata = control, type='response')

        #j. Estimate Psi.P0
```

```
        psi.hat1 <- mean(Y1.predict.est1) - mean(Y0.predict.est1)
        psi.hat2 <- mean(Y1.predict.est2) - mean(Y0.predict.est2)
        psi.hat3 <- mean(Y1.predict.est3) - mean(Y0.predict.est3)
        psi.hat4 <- mean(Y1.predict.est4) - mean(Y0.predict.est4)

        #k. Assign resulting values as a row in the matrix estimates.
        estimates[i,] <- c(psi.hat1, psi.hat2, psi.hat3, psi.hat4)
}
```

```
## 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
```

**Part 5: Performance of Estimators** 1. What is the average value of each estimator?

```
colnames(estimates)<- c('psi.hat1', 'psi.hat2', 'psi.hat3', 'psi.hat4')
summary(estimates)
```

```
##     psi.hat1         psi.hat2         psi.hat3         psi.hat4
##  Min.   :0.4821   Min.   :0.3916   Min.   :0.3745   Min.   :0.2996
##  1st Qu.:0.6104   1st Qu.:0.5831   1st Qu.:0.5206   1st Qu.:0.4539
##  Median :0.6542   Median :0.6231   Median :0.5642   Median :0.5016
##  Mean   :0.6505   Mean   :0.6228   Mean   :0.5638   Mean   :0.5027
##  3rd Qu.:0.6901   3rd Qu.:0.6650   3rd Qu.:0.6115   3rd Qu.:0.5554
##  Max.   :0.8278   Max.   :0.8326   Max.   :0.7688   Max.   :0.7639
```

*As you see above, the mean of each estimator is 0.650, 0.620, 0.564, and 0.503 respectively*

2. Estimate the bias of each estimator. Bias of Estimator #1

```
bias1 <- mean(estimates[,"psi.hat1"] - Psi.P0)
bias1
```

```
## [1] 0.1465755
```

Bias of Estimator #2

```r
bias2 <- mean(estimates[,"psi.hat2"] - Psi.P0)
bias2
```

```
## [1] 0.1188526
```

Bias of Estimator #3

```r
bias3 <- mean(estimates[,"psi.hat3"] - Psi.P0)
bias3
```

```
## [1] 0.0598829
```

Bias of Estimator #4

```r
bias4 <- mean(estimates[,"psi.hat4"] - Psi.P0)
bias4
```

```
## [1] -0.001240282
```

3. Estimate the variance of each estimator.

```r
var1 <- var(estimates[,"psi.hat1"])
var1 #Var of Estimator #1
```

```
## [1] 0.003255506
```

```r
var2 <- var(estimates[,"psi.hat2"])
var2 #Var of Estimator #2
```

```
## [1] 0.00380953
```

```r
var3 <- var(estimates[,"psi.hat3"])
var3 #Var of Estimator #3
```

```
## [1] 0.004864633
```

```r
var4 <- var(estimates[,"psi.hat4"])
var4 #Var of Estimator #4
```

```
## [1] 0.005983574
```

4. Estimate the mean squared error of each estimator.

```r
mse1 <- bias1^2 + var1
mse1
```

```
## [1] 0.02473988
```

```
mse2 <- bias2^2 + var2
mse2
```

```
## [1] 0.01793548
```

```
mse3 <- bias3^2 + var3
mse3
```

```
## [1] 0.008450595
```

```
mse4 <- bias4^2 + var4
mse4
```

```
## [1] 0.005985112
```

5. Briefly comment on the performance of the estimators. Which estimator has the lowest MSE over the R = 500 iterations? Are you surprised?

*The 4th estimator had the lowest MSE over the 500 iterations, and is not surprising. The bias of this estimator was quite a bit smaller than the others, so this MSE makes sense mathematically. From a conceptual standpoint, it makes sense that the estimator with the lowest MSE has the most parameters if we believe that those parameters are actually getting us closer to the truth. Since intuitively it makes sense that the intervention is conditional on the history of conflict in the area and the access to healthcare, the low MSE is a reflection that our intuition was correct.*