Day 3, Practical 2, Hely's solution

Helene Charlotte Wiese Rytgaard

June 8, 2023

1 Simulation setting 1

Task 1.

```
library(data.table)
sim.fun <- function(n) {

    # baseline covariates
    X0.1 <- runif(n, -2, 2)
    X0.2 <- rnorm(n)
    X0.3 <- rbinom(n, 1, 0.2)

    # baseline treatment (randomized)
    A0 <- rbinom(n, 1, 0.5)

    # follow-up covariates
    X1.1 <- rbinom(n, 1, plogis(-0.7 + 0.3*X0.3 + 0.8*A0))
    X1.2 <- rbinom(n, 1, plogis(0.25 - 0.55*X0.3))

    # follow-up treatment
    A1 <- rbinom(n, 1, prob=plogis(0.9 - 5*(1-A0) - 4.7*X1.1 - 4.8*X1.2))

    # outcome
    Y <-  rbinom(n, 1, prob=plogis(-0.9 - 0.2*A0 + 1.2*X1.1 - 0.1*A1 - 0.8*A1*(X1
    .1==0)))


    return(data.table(X0.1=X0.1, X0.2=X0.2, X0.3=X0.3,
              A0=A0,
              X1.1=X1.1, X1.2=X1.2,
              A1=A1,
              Y=Y))
}
```

Task 2:

```
sim.fun <- function(n, intervene=list()) {

    # baseline covariates
    X0.1 <- runif(n, -2, 2)
    X0.2 <- rnorm(n)
    X0.3 <- rbinom(n, 1, 0.2)

    # baseline treatment (randomized)
    if ("A0" %in% names(intervene)) {
    A0 <- intervene$A0
    } else {
    A0 <- rbinom(n, 1, 0.5)
    }

    # follow-up covariates
    X1.1 <- rbinom(n, 1, plogis(-0.7 + 0.3*X0.3 + 0.8*A0))
    X1.2 <- rbinom(n, 1, plogis(0.25 - 0.55*X0.3))

    # follow-up treatment
    if ("A1" %in% names(intervene)) {
    A1 <- intervene$A1(X1.1)
    } else {
    A1 <- rbinom(n, 1, prob=plogis(0.9 - 5*(1-A0) - 4.7*X1.1 - 4.8*X1.2))
    }

    # outcome
    Y <-  rbinom(n, 1, prob=plogis(-0.9 - 0.2*A0 + 1.2*X1.1 - 0.1*A1 - 0.8*A1*(X1
    .1==0)))

    if (length(names(intervene))>0) {
    return(mean(Y))
    } else {
    return(data.table(X0.1=X0.1, X0.2=X0.2, X0.3=X0.3,
              A0=A0,
              X1.1=X1.1, X1.2=X1.2,
              A1=A1,
              Y=Y))
    }
}
```

```
set.seed(12)
ate.itt <- sim.fun(intervene=list(A0=1), n=1e6) -
    sim.fun(intervene=list(A0=0), n=1e6)
ate.static <- sim.fun(intervene=list(A0=1, A1=function(X1.1) 1), n=1e6) -
    sim.fun(intervene=list(A0=0, A1=function(X1.1) 0), n=1e6)
ate.dynamic <- sim.fun(intervene=list(A0=1, A1=function(X1.1) 1*(X1.1==0)), n=1e6) -
    sim.fun(intervene=list(A0=0, A1=function(X1.1) 0), n=1e6)
message(paste0("ITT:     ", ate.itt))
message(paste0("static:  ", ate.static))
message(paste0("dynamic: ", ate.dynamic))
```

```
ITT:     -0.00931700000000002
static:  -0.063294
dynamic: -0.050709
```

Task 3.

```
set.seed(15)
(sim.data <- sim.fun(1000))
```

```
            X0.1        X0.2 X0.3 A0 X1.1 X1.2 A1 Y
   1:  0.40845618 -0.19620228    0  1    1    0  0 1
   2: -1.21982429  0.59503302    1  1    0    0  1 0
   3:  1.86583493 -1.60888231    0  1    0    0  0 0
   4:  0.60362212  0.04123507    0  1    0    0  1 0
   5: -0.53171243 -1.25139144    0  0    0    1  0 0
  ---
1996: -0.72294993 -0.43541869    0  1    0    0  0 1
1997: -0.13592934 -0.80204340    0  0    0    0  0 0
1998: -1.67193359 -1.57095686    0  0    0    1  0 0
1999:  0.68115948 -0.72905589    0  1    0    0  1 0
2000:  0.04187752  0.10649384    0  0    0    0  0 1
```

```
##-- naive approach 1:
fit.glm1 <- glm(Y ~ A1 + X1.1 + X1.2 + A0 + X0.1 + X0.2 + X0.3,
        family=binomial, data=sim.data)
sim.data[, EY1:=predict(fit.glm1, type="response", newdata=copy(sim.data)[, ‘:=‘(A0=1,
     A1=1)])]
sim.data[, EY0:=predict(fit.glm1, type="response", newdata=copy(sim.data)[, ‘:=‘(A0=0,
     A1=0)])]
message("naive est1 = ", sim.data[, mean(EY1-EY0)])
##-- naive approach 2:
fit.glm2 <- glm(Y ~ A1 + A0 + X0.1 + X0.2 + X0.3,
        family=binomial, data=sim.data)
sim.data[, EY1:=predict(fit.glm2, type="response", newdata=copy(sim.data)[, ‘:=‘(A0=1,
     A1=1)])]
sim.data[, EY0:=predict(fit.glm2, type="response", newdata=copy(sim.data)[, ‘:=‘(A0=0,
     A1=0)])]
message("naive est2 = ", sim.data[, mean(EY1-EY0)])
```

```
naive est1 = -0.251243724502775
naive est2 = -0.315160896785246
```

Task 4. See Figure 1.

```
naive.est1 <- list()
naive.est2 <- list()

for (m in 1:500) {

    set.seed(15+m)
    sim.data <- sim.fun(n=1000)

    ##-- naive approach 1:
    fit.glm1 <- glm(Y ~ A1 + X1.1 + X1.2 + A0 + X0.1 + X0.2 + X0.3,
            family=binomial, data=sim.data)
    sim.data[, EY1:=predict(fit.glm1, type="response", newdata=copy(sim.data)[, ‘:=‘(
    A0=1, A1=1)])]
    sim.data[, EY0:=predict(fit.glm1, type="response", newdata=copy(sim.data)[, ‘:=‘(
    A0=0, A1=0)])]
    naive.est1[[m]] <- sim.data[, mean(EY1-EY0)]

    ##-- naive approach 2:
```

```
    fit.glm2 <- glm(Y ~ A1 + A0 + X0.1 + X0.2 + X0.3,
            family=binomial, data=sim.data)
    sim.data[, EY1:=predict(fit.glm2, type="response", newdata=copy(sim.data)[, ‘:=‘(
    A0=1, A1=1)])]
    sim.data[, EY0:=predict(fit.glm2, type="response", newdata=copy(sim.data)[, ‘:=‘(
    A0=0, A1=0)])]
    naive.est2[[m]] <- sim.data[, mean(EY1-EY0)]

}
```

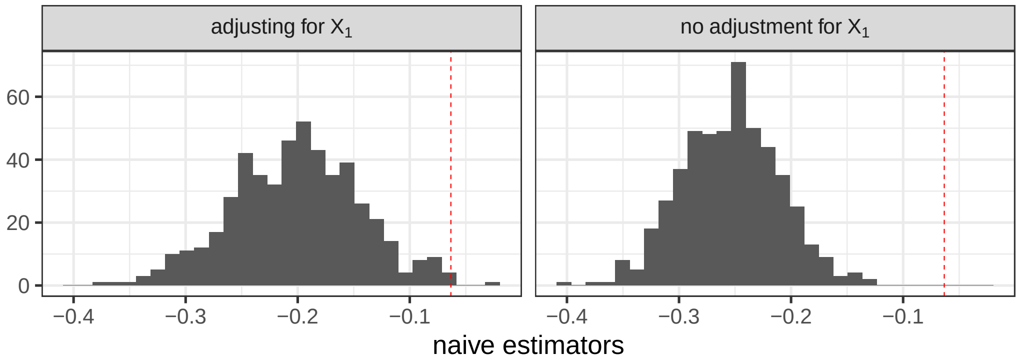

Figure 1

2 Simulation setting 2

Task 5.

```
library(data.table)
sim.fun2 <- function(n=1e6) {
  U <- rbinom(n, 1, prob=0.5)
  A <- rbinom(n, 1, prob=0.5)
  D2 <- D1 <- rbinom(n, 1, prob=plogis(1.3-1.8*U-1.1*A))
  D2[D1==0] <- rbinom(n, 1, prob=plogis(2.1-3.9*U))[D1==0]
  return(data.table(A=A,D1=D1,D2=D2))
}
```

Task 6.

```
set.seed(100)
head(dat2 <- sim.fun2())
```

```
   A D1 D2
1: 1  1  1
2: 1  0  1
3: 0  1  1
4: 0  1  1
5: 1  1  1
6: 1  1  1
```

6.1:

```
summary(fit.D1 <- glm(D1~A, data=dat2, family=binomial))$coefficients
```

```
              Estimate  Std. Error   z value Pr(>|z|)
(Intercept)  0.3346503 0.002869386  116.6279        0
A           -0.9172592 0.004113874 -222.9672        0
```

(We may note that the coefficient in front of A corresponds to:)

```
p.A0 <- 0.5*(plogis(1.3-1.8)+plogis(1.3))
p.A1 <- 0.5*(plogis(1.3-1.8-1.1)+plogis(1.3-1.1))
log((p.A1/(1-p.A1)) / (p.A0/(1-p.A0)))
```

```
[1] -0.9098137
```

(And the intercept coefficient to:)

```
qlogis(p.A0)
```

```
[1] 0.3297059
```

6.2:

```
summary(fit.D2 <- glm(D2~A, data=dat2[D1==0,], family=binomial))$coefficients
```

```
              Estimate  Std. Error    z value Pr(>|z|)
(Intercept) -0.6884959 0.004631504 -148.65492        0
A            0.2961096 0.005865281   50.48515        0
```

6.3:

```
dat2[, id:=1:.N]
dat21 <- copy(dat2)[, k:=1]
dat21$D <- dat21$D1
dat22 <- dat2[dat2$D1==0,][, k:=2]
dat22$D <- dat22$D2
head(dat2.stacked <- rbind(dat21, dat22)[, t:=factor(k)][order(id)])
```

```
   A D1 D2 id k D t
1: 0  1  1  1 1 1 1
2: 1  0  1  2 1 0 1
3: 1  1  1  3 1 1 1
4: 0  0  0  4 1 0 1
5: 1  0  1  5 1 0 1
6: 1  0  1  6 1 0 1
```

6.4:

```
summary(fit.D <- glm(D~A+t, data=dat2.stacked, family=binomial))$coefficients
```

```
               Estimate  Std. Error    z value Pr(>|z|)
(Intercept)  0.1322593 0.002595677   50.95367        0
A           -0.5117185 0.003311977 -154.50546        0
t2          -0.3365993 0.003509668  -95.90632        0
```

```
summary(fit.D <- glm(D~A*t, data=dat2.stacked, family=binomial))$coefficients
```

```
               Estimate  Std. Error   z value Pr(>|z|)
(Intercept)  0.3271974 0.002866385  114.1498        0
A           -0.9103901 0.004112876 -221.3512        0
t2          -1.0156933 0.005446742 -186.4772        0
A:t2         1.2064997 0.007163607  168.4207        0
```

6.5:

```
set.seed(10)
U <- rbinom(n, 1, prob=0.5)
A <- 1
D2.1 <- D1.1 <- rbinom(n, 1, prob=plogis(1.3-1.8*U-1.1*A))
D2.1[D1.1==0] <- rbinom(n, 1, prob=plogis(2.1-3.9*U))[D1.1==0]
A <- 0
D2.0 <- D1.0 <- rbinom(n, 1, prob=plogis(1.3-1.8*U-1.0*A))
D2.0[D1.0==0] <- rbinom(n, 1, prob=plogis(2.0-3.9*U))[D1.0==0]
message(paste0("Counterfactual risk difference: ", mean(D2.1-D2.0)))
```

```
Counterfactual risk difference: -0.106
```

```
set.seed(10)
U <- rbinom(n, 1, prob=0.5)
A <- 1
D2.1 <- D1.1 <- rbinom(n, 1, prob=plogis(1.3-1.8*U-1.1*A))
D2.1[D1.1==0] <- rbinom(n, 1, prob=plogis(2.1-3.9*U))[D1.1==0]
A <- 0
D2.0 <- D1.0 <- rbinom(n, 1, prob=plogis(1.3-1.8*U-1.0*A))
D2.0[D1.0==0] <- rbinom(n, 1, prob=plogis(2.0-3.9*U))[D1.0==0]
message(paste0("Counterfactual hazard ratio at 2nd time-point: ", mean(D2.1[D1.1==0])/
    mean(D2.0[D1.0==0])))
```

```
Counterfactual hazard ratio at 2nd time-point: 1.21615895286972
```

2.1 Treatment switching

Task 7.

```
library(data.table)
sim.fun2 <- function(n=1e6) {
  U <- rbinom(n, 1, prob=0.5)
  A1 <- A <- rbinom(n, 1, prob=0.5)
  D2 <- D1 <- rbinom(n, 1, prob=plogis(1.3-1.8*U-1.1*A))
  D2[D1==0] <- rbinom(n, 1, prob=plogis(2.1-3.9*U))[D1==0]
  A1[D1==0] <- rbinom(n, 1, prob=plogis(1.2+0.5*A))[D1==0]
  return(data.table(A=A,Aswitch=1*(A1!=A),D1=D1,D2=D2))
}
```

Task 8.

```
set.seed(100)
head(dat2 <- sim.fun2())
```

```
   A Aswitch D1 D2
1: 1       0  1  1
2: 1       1  0  1
3: 0       0  1  1
4: 0       0  1  1
5: 1       0  1  1
6: 1       0  1  1
```

```
summary(glm(D2~A+Aswitch, family=binomial, data=dat2[A==1]))$coefficients
```

```
              Estimate  Std. Error   z value Pr(>|z|)
(Intercept)  0.5770818 0.003103637 185.93729        0
Aswitch     -0.9519854 0.009647494 -98.67696        0
```

Task 9.

```
A1 <- A <- rep(1, n)
D2.11 <- D1.11 <- rbinom(n, 1, prob=plogis(1.3-1.8*U-1.1*A))
D2.11[D1.11==0] <- rbinom(n, 1, prob=plogis(2.1-3.9*U))[D1.11==0]
A1[D1.11==0] <- 1

A1 <- A <- rep(1, n)
D2.10 <- D1.10 <- rbinom(n, 1, prob=plogis(1.3-1.8*U-1.1*A))
D2.10[D1.10==0] <- rbinom(n, 1, prob=plogis(2.1-3.9*U))[D1.10==0]
A1[D1.10==0] <- 0

mean(D2.11 - D2.10)
```

```
[1] -0.000303
```

2.2 Right-censoring

Task 10.

```
library(data.table)
sim.fun2 <- function(n=1e6) {
  U <- rbinom(n, 1, prob=0.5)
  A <- rbinom(n, 1, prob=0.5)
  D2 <- D1 <- rbinom(n, 1, prob=plogis(1.3-1.8*U-1.1*A))
  C1 <- rbinom(n, 1, prob=plogis(1.3))
  D2[D1==0 & C1==0] <- rbinom(n, 1, prob=plogis(2.1-3.9*U))[D1==0 & C1==0]
  return(data.table(A=A,D1=D1,C1=C1,D2=D2))
}
```

Task 11.

```
set.seed(100)
head(dat2 <- sim.fun2())
```

```
   A D1 C1 D2
1: 1  1  1  1
2: 1  0  0  0
3: 0  1  1  1
4: 0  1  1  1
5: 1  1  0  1
6: 1  1  1  1
```

11.1:

```
summary(fit.D1 <- glm(D1~A, data=dat2, family=binomial))$coefficients
```

```
               Estimate  Std. Error   z value Pr(>|z|)
(Intercept)  0.3346503 0.002869386  116.6279        0
A           -0.9172592 0.004113874 -222.9672        0
```

11.2:

```
summary(fit.D2 <- glm(D2~A, data=dat2[D1==0 & C1==0,], family=binomial))$coefficients
```

```
              Estimate Std. Error   z value      Pr(>|z|)
(Intercept) -0.6803444 0.01002981 -67.83221  0.000000e+00
A            0.2937851 0.01269486  23.14206 1.747762e-118
```

11.3:

```
dat2[, id:=1:.N]
dat21 <- copy(dat2)[, k:=1]
dat21$D <- dat21$D1
dat22 <- dat2[dat2$D1==0 & dat2$C1==0,][, k:=2]
dat22$D <- dat22$D2
head(dat2.stacked <- rbind(dat21, dat22)[, t:=factor(k)][order(id)])
```

```
   A D1 C1 D2 id k D t
1: 1  1  1  1  1 1 1 1
2: 1  0  0  0  2 1 0 1
3: 1  0  0  0  2 2 0 2
4: 0  1  1  1  3 1 1 1
5: 0  1  1  1  4 1 1 1
6: 1  1  0  1  5 1 1 1
```

11.4:

```
summary(fit.D <- glm(D~A+t, data=dat2.stacked, family=binomial))$coefficients
```

```
              Estimate  Std. Error    z value Pr(>|z|)
(Intercept)  0.2777191 0.002789808   99.54772        0
A           -0.8004605 0.003888827 -205.83597        0
t2          -0.3092774 0.006580645  -46.99805        0
```

```
summary(fit.D <- glm(D~A*t, data=dat2.stacked, family=binomial))$coefficients
```

```
               Estimate  Std. Error    z value Pr(>|z|)
(Intercept)  0.3346503 0.002869386  116.62786        0
A           -0.9172592 0.004113874 -222.96724        0
t2          -1.0149947 0.010432186  -97.29454        0
A:t2         1.2110443 0.013344786   90.75037        0
```

11.5:

```
set.seed(10)
U <- rbinom(n, 1, prob=0.5)
A <- 1
D2.1 <- D1.1 <- rbinom(n, 1, prob=plogis(1.3-1.8*U-1.1*A))
D2.1[D1.1==0] <- rbinom(n, 1, prob=plogis(2.1-3.9*U))[D1.1==0]
A <- 0
D2.0 <- D1.0 <- rbinom(n, 1, prob=plogis(1.3-1.8*U-1.0*A))
D2.0[D1.0==0] <- rbinom(n, 1, prob=plogis(2.0-3.9*U))[D1.0==0]
mean(D2.1-D2.0)
```

```
[1] -0.106
```