Day 3, Practical 2, Hely's solution

Helene Charlotte Wiese Rytgaard

May 22, 2023

1 Simulation setting 1

Task 1.

```r
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:

```r
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)
```

```r
    # 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))
    }
}
```

```r
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.

```r
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
```

```r
##-- 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 = -0.181898731224723
naive est2 = -0.241852261901675
```

Task 4. See Figure 1.

```r
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)]

}
```
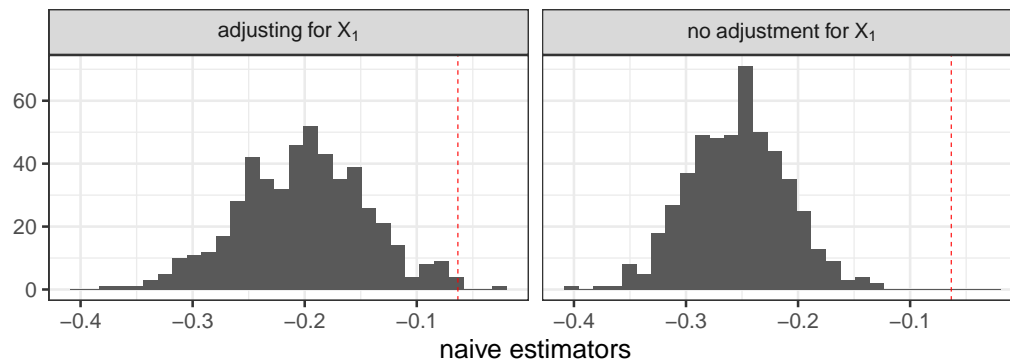

Figure 1

2 Simulation setting 2

Task 5.

```
library(data.table)
sim.fun2 <- function(n=1e6) {
  U <- rbinom(n, 1, prob=plogis(0.5))
  A <- rbinom(n, 1, prob=plogis(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: 0  0  0
2: 0  1  1
3: 1  0  0
4: 1  0  0
5: 1  0  0
6: 0  0  0
```

6.1:

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

```
              Estimate   Std. Error    z value Pr(>|z|)
(Intercept)  0.1290582  0.003260927   39.57716        0
A           -0.9191126  0.004256537 -215.92966        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

6.2:
```
summary(fit.D2 <- glm(D2~A, data=dat2[D1==0,], family=binomial))$coefficients
```

```
              Estimate   Std. Error    z value Pr(>|z|)
(Intercept) -0.9848795  0.005346553 -184.20832        0
A            0.2617649  0.006261530   41.80527        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)])
```

```
   A D1 D2 id k D t
1: 0  0  0  1 1 0 1
2: 0  1  1  2 1 1 1
3: 1  0  0  3 1 0 1
4: 1  0  0  4 1 0 1
5: 1  0  0  5 1 0 1
6: 0  0  0  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) -0.0998452  0.002914334  -34.26003  3.091798e-257
A           -0.5330474  0.003435493 -155.15894   0.000000e+00
t2          -0.3301824  0.003483821  -94.77594   0.000000e+00
```

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

```
            Estimate  Std. Error    z value  Pr(>|z|)
(Intercept)  0.1290582 0.003260927   39.57716          0
A           -0.9191126 0.004256537 -215.92966          0
t2          -1.1139377 0.006262529 -177.87346          0
A:t2         1.1808775 0.007571319  155.96722          0
```

6.5:

```
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] -0.115641
```

2.1 Treatment switching

Task 7.

```
library(data.table)
sim.fun2 <- function(n=1e6) {
  U <- rbinom(n, 1, prob=plogis(0.5))
  A1 <- A <- rbinom(n, 1, prob=plogis(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: 0       1  0  0
2: 0       0  1  1
3: 1       0  0  0
4: 1       0  0  0
5: 1       0  0  0
6: 0       1  0  0
```

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

```
            Estimate  Std. Error   z value  Pr(>|z|)
(Intercept)  0.2504214 0.002702747   92.6544          0
Aswitch     -0.9904240 0.008748468 -113.2111          0
```

Task 9.

6

```
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] -0.000056

2.2 Informative censoring

Task 10.

```
library(data.table)
sim.fun2 <- function(n=1e6) {
  U <- rbinom(n, 1, prob=plogis(0.5))
  A <- rbinom(n, 1, prob=plogis(0.5))
  D2 <- D1 <- rbinom(n, 1, prob=plogis(1.3-1.8*U-1.1*A))
  C1 <- rbinom(n, 1, prob=plogis(1.3-1.8*U))
  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: 0  0  0  0
2: 0  1  1  1
3: 1  0  0  0
4: 1  0  0  0
5: 1  0  1  0
6: 0  0  0  0
```

11.1:

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

```
              Estimate  Std. Error     z value Pr(>|z|)
(Intercept)  0.1290582 0.003260927    39.57716        0
A           -0.9191126 0.004256537 -215.92966        0
```

11.2:

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

7

```
             Estimate  Std. Error      z value      Pr(>|z|)
(Intercept) -2.13099978 0.007723776 -275.901300 0.000000e+00
A            0.07560283 0.009103530    8.304782 1.000029e-16
```

11.3:

```
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)])
```

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

11.4

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

```
             Estimate  Std. Error     z value       Pr(>|z|)
(Intercept)  0.01507786 0.003051071    4.941826 0.0000007739414
A           -0.72585475 0.003770841 -192.491464 0.0000000000000
t2          -1.62254722 0.004608921 -352.044945 0.0000000000000
```

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

```
             Estimate  Std. Error    z value Pr(>|z|)
(Intercept)  0.1290582 0.003260927   39.57716        0
A           -0.9191126 0.004256537 -215.92966        0
t2          -2.2600580 0.008383935 -269.57009        0
A:t2         0.9947154 0.010049496   98.98162        0
```

11.5:

```
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] -0.116882
```