Day 4, Practical 2, Hely's solution

Helene Charlotte Wiese Rytgaard

May 23, 2023

Task 1: Implementing the targeting algorithm.

1.

```
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(15)
(sim.data <- sim.fun(n=2000))
```

```
           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
```

2.

```
weight.truncation <- FALSE
fit.pi0 <- glm(A0 ~ X0.1+X0.2+X0.3, family=binomial, data=sim.data)
fit.pi1 <- glm(A1 ~ X0.1+X0.2+X0.3+A0+X1.1+X1.2, family=binomial, data=sim.data)
sim.data[, cum.pi0:=predict(fit.pi0, type="response")]
sim.data[, cum.pi1:=predict(fit.pi0, type="response")*predict(fit.pi1, type="response"
    )]
if (weight.truncation) {
    sim.data[cum.pi0<0.01, cum.pi0:=0.01]
    sim.data[cum.pi1<0.01, cum.pi1:=0.01]
}
sim.data[, H1:=(A0==1)/cum.pi0]
sim.data[, H2:=(A0==1)*(A1==1)/cum.pi1]
```

3.

```
fit.Q2 <- glm(Y ~ A1 + X1.1 + X1.2 + A0 + X0.1 + X0.2 + X0.3,
          family=binomial, data=sim.data)
sim.data[, pred.Q2:=predict(fit.Q2, newdata=copy(sim.data)[, A1:=1], type="response")]
```

4.

```
fit.tmle2 <- glm(Y ~ offset(qlogis(pred.Q2)), weights=H2,
         family=binomial, data=sim.data)
sim.data[, pred.Q2.star:=predict(fit.tmle2, type="response")]
##-- check that we solve:
sim.data[, mean(H2*(Y-pred.Q2.star))]
```

```
-2.22802486597902e-12
```

5.

```
fit.Q1 <- glm(pred.Q2.star ~ A0 + X0.1 + X0.2 + X0.3,
          family=quasibinomial, data=sim.data)
sim.data[, pred.Q1:=predict(fit.Q1, newdata=copy(sim.data)[, A0:=1], type="response")]
```

6.

```
fit.tmle1 <- glm(pred.Q2.star ~ offset(qlogis(pred.Q1)), weights=H1,
         family=quasibinomial, data=sim.data)
sim.data[, pred.Q1.star:=predict(fit.tmle1, type="response")]
##-- check that we solve:
sim.data[, mean(H1*(pred.Q2.star - pred.Q1.star))]
```

```
[1] -1.271552e-17
```

7.

```
(est.own.tmle <- sim.data[, mean(pred.Q1.star)])
```

```
[1] 0.2759537
```

8. Compute the standard error by evaluating the efficient influence function, i.e.,

```
(se.own.tmle <- sim.data[, sqrt(mean((H2*(Y-pred.Q2.star) +
                      H1*(pred.Q2.star - pred.Q1.star) +
                      pred.Q1.star - est.own.tmle)^2)/nrow(sim.data))])
```

```
[1] 0.1107792
```

Task 2: Using `ltmle` software.

1.

```
library(ltmle)
set.seed(15)
sim.data <- sim.fun(n=2000)
fit.ltmle <- ltmle(sim.data,
           Anodes=paste0("A",0:1),
           Lnodes=c(paste0("X0.", 1:3), paste0("X1.", 1:2)),
           Ynodes="Y",
           abar=c(1,1), variance.method="ic")
summary(fit.ltmle)
```

```
Qform not specified, using defaults:
formula for X1.1:
Q.kplus1 ~ X0.1 + X0.2 + X0.3 + A0
formula for Y:
Q.kplus1 ~ X0.1 + X0.2 + X0.3 + A0 + X1.1 + X1.2 + A1
```

```
gform not specified, using defaults:
formula for A0:
A0 ~ X0.1 + X0.2 + X0.3
formula for A1:
A1 ~ X0.1 + X0.2 + X0.3 + A0 + X1.1 + X1.2

Estimate of time to completion: < 1 minute
Warning message:
In CheckForVarianceWarning(inputs, g.ratio) :
  Variance estimate is based on influence curve only, which may be significantly anticonservative be
Estimator:  tmle
Call:
ltmle(data = sim.data, Anodes = paste0("A", 0:1), Lnodes = c(paste0("X0.",
    1:3), paste0("X1.", 1:2)), Ynodes = "Y", abar = c(1, 1),
    variance.method = "ic")

   Parameter Estimate:  0.15695
    Estimated Std Err:  0.051843
              p-value:  0.0024662
    95% Conf Interval: (0.055343, 0.25856)
```

2. Go back and change argument `weight.truncation` to `TRUE`.

3.

```
summary(fit.ltmle, estimator="iptw")$treatment$estimate
```

```
     iptw
0.1460717
```

4.

```
library(ltmle)
set.seed(15)
sim.data <- sim.fun(n=2000)
fit.g <- ltmle(sim.data,
           Anodes=paste0("A",0:1),
           Lnodes=c(paste0("X0.", 1:3), paste0("X1.", 1:2)),
           Ynodes="Y",
           gcomp=TRUE,
           abar=c(1,1), variance.method="ic")
summary(fit.g)
```

```
Qform not specified, using defaults:
formula for X1.1:
Q.kplus1 ~ X0.1 + X0.2 + X0.3 + A0
formula for Y:
Q.kplus1 ~ X0.1 + X0.2 + X0.3 + A0 + X1.1 + X1.2 + A1

gform not specified, using defaults:
```

```
formula for A0:
A0 ~ X0.1 + X0.2 + X0.3
formula for A1:
A1 ~ X0.1 + X0.2 + X0.3 + A0 + X1.1 + X1.2

Estimate of time to completion: < 1 minute
Warning message:
In CheckForVarianceWarning(inputs, g.ratio) :
  Variance estimate is based on influence curve only, which may be significantly anticonservative be
Estimator:  gcomp
Warning: inference for gcomp is not accurate! It is based on TMLE influence curves.
Call:
ltmle(data = sim.data, Anodes = paste0("A", 0:1), Lnodes = c(paste0("X0.",
    1:3), paste0("X1.", 1:2)), Ynodes = "Y", abar = c(1, 1),
    gcomp = TRUE, variance.method = "ic")

   Parameter Estimate:  0.22819
    Estimated Std Err:  0.055393
              p-value:  0.000037983
    95% Conf Interval: (0.11962, 0.33676)
```

True values of parameters:

```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: Estimating the static effect.

```r
set.seed(15)
sim.data <- sim.fun(n=2000)
fit.ltmle.static <- ltmle(sim.data,
              Anodes=paste0("A",0:1),
              Lnodes=c(paste0("X0.", 1:3), paste0("X1.", 1:2)),
              Ynodes="Y",
              abar=list(treatment=c(1,1), control=c(0,0)),
              variance.method="ic")
message("------------------")
message("tmle output:")
```

```r
message("---")
summary(fit.ltmle.static)$effect.measures$ATE
message("------------------")
message("ipw output:")
message("---")
summary(fit.ltmle.static, estimator="iptw")$effect.measures$ATE
```

```
Qform not specified, using defaults:
formula for X1.1:
Q.kplus1 ~ X0.1 + X0.2 + X0.3 + A0
formula for Y:
Q.kplus1 ~ X0.1 + X0.2 + X0.3 + A0 + X1.1 + X1.2 + A1

gform not specified, using defaults:
formula for A0:
A0 ~ X0.1 + X0.2 + X0.3
formula for A1:
A1 ~ X0.1 + X0.2 + X0.3 + A0 + X1.1 + X1.2

Estimate of time to completion: < 1 minute
Warning message:
In CheckForVarianceWarning(inputs, g.ratio) :
  Variance estimate is based on influence curve only, which may be significantly anticonservative be
------------------
tmle output:
---
$long.name
[1] "Additive Treatment Effect"

$estimate
[1] -0.215463

$std.dev
[1] 0.05409723

$pvalue
[1] 0.00006808408

$CI
           2.5%      97.5%
[1,] -0.3214916 -0.1094343

$log.std.err
[1] FALSE
------------------
ipw output:
---
$long.name
[1] "Additive Treatment Effect"

$estimate
```

```
[1] -0.2263073

$std.dev
[1] 0.04865195

$pvalue
[1] 0.000003294405

$CI
           2.5%      97.5%
[1,] -0.3216633 -0.1309512

$log.std.err
[1] FALSE
```

Task 4: Estimating the static effect with g-formula estimation.

```r
set.seed(15)
sim.data <- sim.fun(n=2000)
fit.g.static <- ltmle(sim.data,
              Anodes=paste0("A",0:1),
              Lnodes=c(paste0("X0.", 1:3), paste0("X1.", 1:2)),
              Ynodes="Y",
              abar=list(treatment=c(1,1), control=c(0,0)),
              gcomp=TRUE,
              variance.method="ic")
message("------------------")
message("g-formula output:")
message("---")
summary(fit.g.static)$effect.measures$ATE
```

```
Qform not specified, using defaults:
formula for X1.1:
Q.kplus1 ~ X0.1 + X0.2 + X0.3 + A0
formula for Y:
Q.kplus1 ~ X0.1 + X0.2 + X0.3 + A0 + X1.1 + X1.2 + A1

gform not specified, using defaults:
formula for A0:
A0 ~ X0.1 + X0.2 + X0.3
formula for A1:
A1 ~ X0.1 + X0.2 + X0.3 + A0 + X1.1 + X1.2

Estimate of time to completion: < 1 minute
Warning message:
In CheckForVarianceWarning(inputs, g.ratio) :
  Variance estimate is based on influence curve only, which may be significantly anticonservative be
------------------
g-formula output:
---
$long.name
```

```
[1] "Additive Treatment Effect"

$estimate
[1] -0.1435938

$std.dev
[1] 0.05750552

$pvalue
[1] 0.01252333

$CI
           2.5%       97.5%
[1,] -0.2563026 -0.03088509

$log.std.err
[1] FALSE
```

Task 5.

```
message("------------------")
message("tmle output:")
message("---")
summary(fit.ltmle.static)$effect.measures$ATE[c("estimate")]
message("------------------")
message("ipw output:")
message("---")
summary(fit.ltmle.static, estimator="iptw")$effect.measures$ATE[c("estimate")]
message("------------------")
message("g-formula output:")
message("---")
summary(fit.g.static)$effect.measures$ATE[c("estimate")]
```

```
------------------
tmle output:
---
$estimate
[1] -0.215463
------------------
ipw output:
---
$estimate
[1] -0.2263073
------------------
g-formula output:
---
$estimate
[1] -0.1435938
```

Task 6: Estimating the ITT effect.

```r
set.seed(15)
sim.data <- sim.fun(n=2000)
fit.ltmle.itt <- ltmle(sim.data,
                Anodes=paste0("A",0),
                Lnodes=c(paste0("X0.", 1:3), paste0("X1.", 1:2),  "A1"),
                Ynodes="Y",
                abar=list(treatment=c(1), control=c(0)))
summary(fit.ltmle.itt)$effect.measures$ATE
```

```
Qform not specified, using defaults:
formula for X1.1:
Q.kplus1 ~ X0.1 + X0.2 + X0.3 + A0

gform not specified, using defaults:
formula for A0:
A0 ~ X0.1 + X0.2 + X0.3

Estimate of time to completion: < 1 minute
$long.name
[1] "Additive Treatment Effect"

$estimate
[1] 0.03664088

$std.dev
[1] 0.02179757

$pvalue
[1] 0.09277035

$CI
             2.5%      97.5%
[1,] -0.006081571 0.07936332

$log.std.err
[1] FALSE
```

Task 7: Estimating the dynamic effect.

```r
set.seed(15)
sim.data <- sim.fun(n=2000)
fit.ltmle.dynamic <- ltmle(sim.data,
                Anodes=paste0("A",0:1),
                Lnodes=c(paste0("X0.", 1:3), paste0("X1.", 1:2)),
                Ynodes="Y",
                rule=list(treatment=function(row) c(1, ifelse(row["X1.1"]==1, 0, 1)),
                      control=function(row) c(0,0)),
                variance.method="ic")
summary(fit.ltmle.dynamic)$effect.measures$ATE
```

```
Qform not specified, using defaults:
formula for X1.1:
Q.kplus1 ~ X0.1 + X0.2 + X0.3 + A0
formula for Y:
Q.kplus1 ~ X0.1 + X0.2 + X0.3 + A0 + X1.1 + X1.2 + A1

gform not specified, using defaults:
formula for A0:
A0 ~ X0.1 + X0.2 + X0.3
formula for A1:
A1 ~ X0.1 + X0.2 + X0.3 + A0 + X1.1 + X1.2

Estimate of time to completion: < 1 minute
Warning message:
In CheckForVarianceWarning(inputs, g.ratio) :
  Variance estimate is based on influence curve only, which may be significantly anticonservative be
$long.name
[1] "Additive Treatment Effect"

$estimate
[1] 0.04175735

$std.dev
[1] 0.04672911

$pvalue
[1] 0.3715335

$CI
            2.5%     97.5%
[1,] -0.04983003 0.1333447

$log.std.err
[1] FALSE
```

Task 8.

```r
message("------------------")
message("tmle for static effect:")
message("---")
summary(fit.ltmle.static)$effect.measures$ATE[c("estimate", "std.dev")]
message("------------------")
message("tmle for dynamic effect:")
message("---")
summary(fit.ltmle.dynamic)$effect.measures$ATE[c("estimate", "std.dev")]
message("------------------")
message("tmle for itt effect:")
message("---")
summary(fit.ltmle.itt)$effect.measures$ATE[c("estimate", "std.dev")]
```

```
------------------
```

```
tmle for static effect:
---
$estimate
[1] -0.215463

$std.dev
[1] 0.05409723
------------------
tmle for dynamic effect:
---
$estimate
[1] 0.04175735

$std.dev
[1] 0.04672911
------------------
tmle for itt effect:
---
$estimate
[1] 0.03664088

$std.dev
[1] 0.02179757
```

Task 9: Simulation study targeting the static effect.

```
library(ltmle)
ate.ltmle.static <- list()
ate.ipw.static <- list()
ate.g.static <- list()
ate.g2.static <- list()

for (m in 1:500) {
    set.seed(15+m)
    sim.data <- sim.fun(n=2000)

    ##- 1.
    fit.ltmle <- ltmle(sim.data,
                Anodes=paste0("A",0:1),
                Lnodes=c(paste0("X0.", 1:3),
                 paste0("X1.", 1:2)),
                Ynodes="Y",
                abar=list(treatment=c(1,1), control=c(0,0)))
    ate.ltmle.static[[m]] <- summary(fit.ltmle)$effect.measures$ATE$estimate
    ate.ipw.static[[m]] <- summary(fit.ltmle, estimator="iptw")$effect.measures$ATE$
    estimate

    ##- 2.
    fit.g <- ltmle(sim.data, Anodes=paste0("A",0:1), Lnodes=c(paste0("X0.", 1:3),
    paste0("X1.", 1:2)), Ynodes="Y",
           abar=list(treatment=c(1,1), control=c(0,0)), gcomp=TRUE)
    ate.g.static[[m]] <- summary(fit.g)$effect.measures$ATE$estimate

    ##- 3.
```

```
    fit.g <- ltmle(sim.data, Anodes=paste0("A",0:1), Lnodes=c(paste0("X0.", 1:3),
    paste0("X1.", 1:2)), Ynodes="Y",
           Qform=c(X1.1="Q.kplus1~X0.1+X0.2+X0.3+A0",
               Y="Q.kplus1~X0.1+X0.2+X0.3+A0+X1.1+X1.2+A1*X1.1"),
           abar=list(treatment=c(1,1), control=c(0,0)), gcomp=TRUE)
    ate.g2.static[[m]] <- summary(fit.g)$effect.measures$ATE$estimate
}
```

Task 10. See Figure 1.

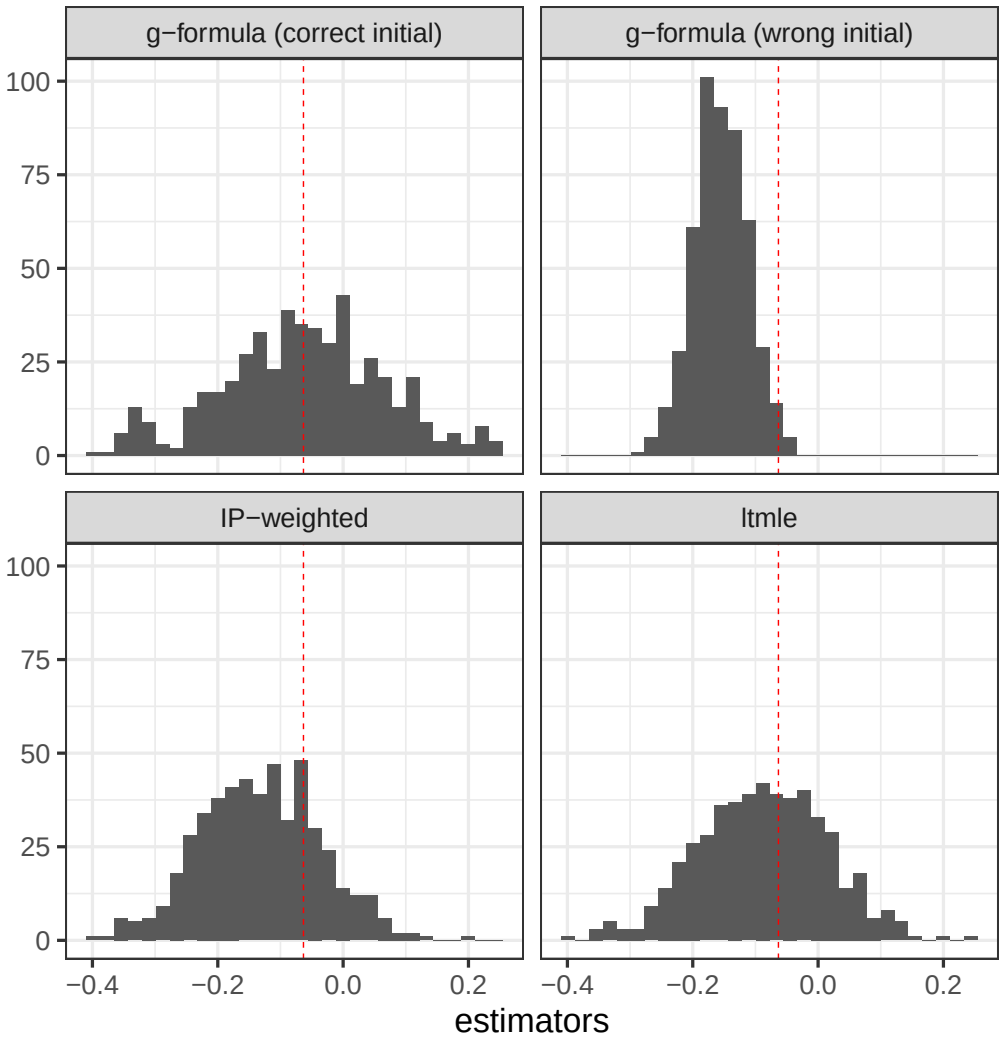

Figure 1

Task 11: Continuing the simulation study with TMLE targeting the dynamic effect and the ITT effect.

```r
library(ltmle)
ate.ltmle.static <- list()
ate.se.ltmle.static <- list()
ate.ltmle.itt <- list()
ate.se.ltmle.itt <- list()
ate.ltmle.dynamic <- list()
ate.se.ltmle.dynamic <- list()

for (m in 1:500) {
    set.seed(15+m)
    sim.data <- sim.fun(n=2000)

    ##- 1.
    fit.ltmle <- ltmle(sim.data,
                Anodes=paste0("A",0:1),
                Lnodes=c(paste0("X0.", 1:3), paste0("X1.", 1:2)),
                Ynodes="Y",
                abar=list(treatment=c(1,1), control=c(0,0)),
                Qform=c(X1.1="Q.kplus1~X0.1+X0.2+X0.3+A0",
                    Y="Q.kplus1~X0.1+X0.2+X0.3+A0+X1.1+X1.2+A1*X1.1"),
                variance.method="ic")
    ate.ltmle.static[[m]] <- summary(fit.ltmle)$effect.measures$ATE$estimate
    ate.se.ltmle.static[[m]] <- summary(fit.ltmle)$effect.measures$ATE$std.dev

    ##- 2.
    fit.ltmle.itt <- ltmle(sim.data,
                Anodes=paste0("A",0),
                Lnodes=c(paste0("X0.", 1:3), paste0("X1.", 1:2),  "A1"),
                Ynodes="Y",
                Qform=c(X1.1="Q.kplus1~X0.1+X0.2+X0.3+A0",
                    Y="Q.kplus1~X0.1+X0.2+X0.3+A0+X1.1+X1.2+A1*X1.1"),
                abar=list(treatment=c(1), control=c(0)),
                variance.method="ic")
    ate.ltmle.itt[[m]] <- summary(fit.ltmle.itt)$effect.measures$ATE$estimate
    ate.se.ltmle.itt[[m]] <- summary(fit.ltmle.itt)$effect.measures$ATE$std.dev

    ##- 3.
    fit.ltmle.dynamic <- ltmle(sim.data,
                    Anodes=paste0("A",0:1),
                    Lnodes=c(paste0("X0.", 1:3), paste0("X1.", 1:2)),
                    Ynodes="Y",
                    Qform=c(X1.1="Q.kplus1~X0.1+X0.2+X0.3+A0",
                        Y="Q.kplus1~X0.1+X0.2+X0.3+A0+X1.1+X1.2+A1*X1.1"),
                    rule=list(treatment=function(row) c(1, ifelse(row["X1.1"]==1, 0, 1)
    ),
                      control=function(row) c(0,0)),
                    variance.method="ic")
    ate.ltmle.dynamic[[m]] <- summary(fit.ltmle.dynamic)$effect.measures$ATE$estimate
    ate.se.ltmle.dynamic[[m]] <- summary(fit.ltmle.dynamic)$effect.measures$ATE$std.
    dev
}
```

Task 12. See Figure 2.

```
message("------------------")
message("coverage: tmle for static effect:")
message("---")
mean(ate.static<=unlist(ate.ltmle.static)+1.96*unlist(ate.se.ltmle.static) &
     ate.static>=unlist(ate.ltmle.static)-1.96*unlist(ate.se.ltmle.static))
message("------------------")
message("coverage: tmle for itt effect:")
message("---")
mean(ate.itt<=unlist(ate.ltmle.itt)+1.96*unlist(ate.se.ltmle.itt) &
     ate.itt>=unlist(ate.ltmle.itt)-1.96*unlist(ate.se.ltmle.itt))
message("------------------")
message("coverage: tmle for dynamic effect:")
message("---")
mean(ate.dynamic<=unlist(ate.ltmle.dynamic)+1.96*unlist(ate.se.ltmle.dynamic) &
     ate.dynamic>=unlist(ate.ltmle.dynamic)-1.96*unlist(ate.se.ltmle.dynamic))
```

```
------------------
coverage: tmle for static effect:
---
[1] 0.568
------------------
coverage: tmle for itt effect:
---
[1] 0.938
------------------
coverage: tmle for dynamic effect:
---
[1] 0.928
```

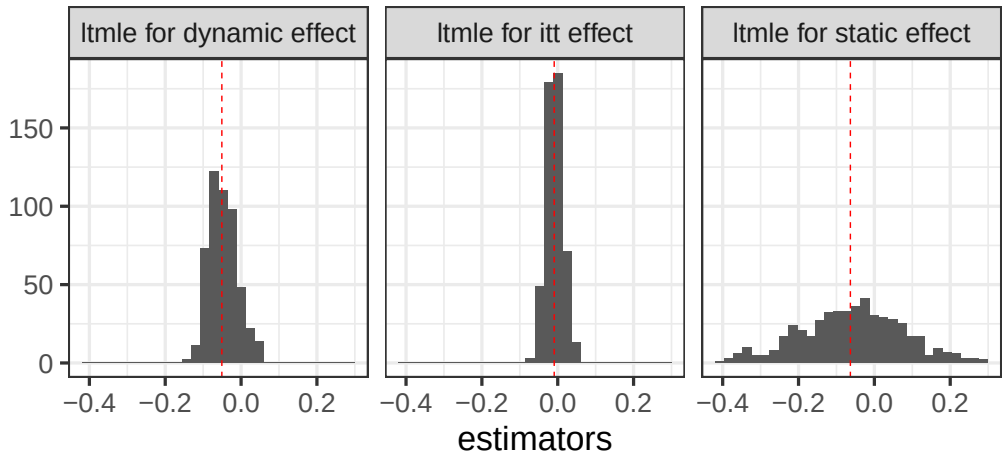

Figure 2

Task 13: Using super learning.

1.

```
fit.ltmle.sl <- ltmle(sim.data, Anodes=paste0("A",0:1),
              Lnodes=c(paste0("X0.", 1:3), paste0("X1.", 1:2)),
              Ynodes="Y",
              SL.library=list(Q=c("SL.glm", "SL.mean", "SL.glm.interaction",
                      "SL.glmnet", "SL.gam"),
                      g=c("SL.glm", "SL.mean", "SL.glmnet",
                      "SL.gam")),
              rule=list(treatment=function(row) c(1, ifelse(row["X1.1"]==1, 0, 1)),
                control=function(row) c(0,0)), variance.method="ic")
```

2.

```
fit.ltmle.sl$fit$Q[[1]]
fit.ltmle.sl$fit$g
```

```
$X1.1
                             Risk       Coef
SL.glm_All              0.02662852 0.00000000
SL.mean_All             0.02792894 0.04233679
SL.glm.interaction_All 0.02660383 0.50686160
SL.glmnet_All                 NA 0.00000000
SL.gam_All              0.02660866 0.45080161

$Y
                            Risk       Coef
SL.glm_All              0.2166817 0.00000000
SL.mean_All             0.2385468 0.00000000
SL.glm.interaction_All 0.2186063 0.09852719
SL.glmnet_All           0.2158281 0.90147281
SL.gam_All              0.2167009 0.00000000
[[1]]
[[1]]$A0
                   Risk Coef
SL.glm_All    0.2501241    0
SL.mean_All   0.2497703    1
SL.glmnet_All 0.2497703    0
SL.gam_All    0.2502473    0

[[1]]$A1
                    Risk      Coef
SL.glm_All    0.02687716 0.0000000
SL.mean_All   0.06689833 0.0000000
SL.glmnet_All 0.02684041 0.6521786
SL.gam_All    0.02688544 0.3478214


[[2]]
[[2]]$A0
                   Risk Coef
```

```
SL.glm_All    0.2501241    0
SL.mean_All   0.2497703    1
SL.glmnet_All 0.2497703    0
SL.gam_All    0.2502473    0

[[2]]$A1
                    Risk      Coef
SL.glm_All    0.02687716 0.0000000
SL.mean_All   0.06689833 0.0000000
SL.glmnet_All 0.02684041 0.6521786
SL.gam_All    0.02688544 0.3478214
```

1 `ltmle` with right-censored data

Task 14.

```
set.seed(100)
head(sim.data2 <- sim.fun2())
```

Task 15.

```
(risk.switch <- sim.fun2(n=1e6, intervene=list(A0=0, A1=1)))
(risk.not.switch <- sim.fun2(n=1e6, intervene=list(A0=0, A1=0)))
```

```
[1] 0.732033
[1] 0.732059
```

Task 16.

```
sim.data2[, (paste0("C", 1)):=BinaryToCensoring(is.censored=get(paste0("C", 1)))]
```

```
ltmle.fit2 <- ltmle(sim.data2,
            Anodes=paste0("A", 0:1),
            Lnodes=paste0("X", 0:1),
            Cnodes="C1",
            Ynodes=paste0("D", 1:2),
            abar=list(treatment=c(0,1), control=c(0,0)),
            survivalOutcome=TRUE)
```

Task 17.

```
summary(ltmle.fit2)
```

```
Estimator:  tmle
Call:
ltmle(data = sim.data2, Anodes = paste0("A", 0:1), Cnodes = "C1",
    Lnodes = paste0("X", 0:1), Ynodes = paste0("D", 1:2), survivalOutcome = TRUE,
    abar = list(treatment = c(0, 1), control = c(0, 0)))

Treatment Estimate:
   Parameter Estimate:  0.66787
    Estimated Std Err:  0.020457
```

```
              p-value:  <2e-16
    95% Conf Interval: (0.62778, 0.70797)

Control Estimate:
   Parameter Estimate:  0.72594
    Estimated Std Err:  0.049985
              p-value:  <2e-16
    95% Conf Interval: (0.62797, 0.82391)

Additive Treatment Effect:
   Parameter Estimate:  -0.058066
    Estimated Std Err:  0.051609
              p-value:  0.26054
    95% Conf Interval: (-0.15922, 0.043085)

Relative Risk:
   Parameter Estimate:  0.92001
  Est Std Err log(RR):  0.071808
              p-value:  0.24564
    95% Conf Interval: (0.79923, 1.059)

Odds Ratio:
   Parameter Estimate:  0.75917
  Est Std Err log(OR):  0.25668
              p-value:  0.28306
    95% Conf Interval: (0.45904, 1.2555)
```

Task 18.

```
sim.data2[, X0.squared:=X0^2]
setcolorder(sim.data2, c(names(sim.data2)[1],
             "X0.squared",
             names(sim.data2)[-c(1,ncol(sim.data2))]))
ltmle.fit2 <- ltmle(sim.data2,
            Anodes=paste0("A", 0:1),
            Lnodes=paste0("X", 0:1),
            Cnodes="C1",
            Ynodes=paste0("D", 1:2),
            abar=list(treatment=c(0,1), control=c(0,0)),
            survivalOutcome=TRUE)
```

```
summary(ltmle.fit2)
```

```
Estimator:  tmle
Call:
ltmle(data = sim.data2, Anodes = paste0("A", 0:1), Cnodes = "C1",
    Lnodes = paste0("X", 0:1), Ynodes = paste0("D", 1:2), survivalOutcome = TRUE,
    abar = list(treatment = c(0, 1), control = c(0, 0)))

Treatment Estimate:
   Parameter Estimate:  0.72423
```

```
    Estimated Std Err:  0.022258
              p-value:  <2e-16
    95% Conf Interval: (0.68061, 0.76786)

Control Estimate:
   Parameter Estimate:  0.71872
    Estimated Std Err:  0.05146
              p-value:  <2e-16
    95% Conf Interval: (0.61786, 0.81958)

Additive Treatment Effect:
   Parameter Estimate:  0.0055139
    Estimated Std Err:  0.053037
              p-value:  0.9172
    95% Conf Interval: (-0.098436, 0.10946)

Relative Risk:
   Parameter Estimate:  1.0077
  Est Std Err log(RR):  0.073737
              p-value:  0.91745
    95% Conf Interval: (0.87208, 1.1644)

Odds Ratio:
   Parameter Estimate:  1.0278
  Est Std Err log(OR):  0.26267
              p-value:  0.9168
    95% Conf Interval: (0.61423, 1.7199)
```

```
ltmle.fit2$fit$Q[[1]]
```

```
$D1
               Estimate Std. Error    t value      Pr(>|t|)
(Intercept)  1.56596540 0.06922717  22.620677 4.787834e-101
X0          -0.08085172 0.02985490  -2.708156  6.823564e-03
X0.squared  -0.42284781 0.02997600 -14.106214  3.823172e-43
A0          -0.67167523 0.07179688  -9.355215  2.164993e-20

$X1
                 Estimate  Std. Error       t value      Pr(>|t|)
(Intercept)  0.4849548188 0.003442044  140.89153226 1.250570e-305
X0          -0.3660083765 0.002033794 -179.96337273  0.000000e+00
X0.squared  -1.1552932320 0.002111787 -547.06905922  0.000000e+00
A0          -0.0001867727 0.004182044   -0.04466063  9.644038e-01

$D2
               Estimate Std. Error    t value             Pr(>|t|)
(Intercept)  0.6512907  0.4119586  1.5809615 0.114813911874227001
X0          -0.3734311  0.1645689 -2.2691480 0.023883589133921471
X0.squared  -1.1560105  0.1562277 -7.3995218 0.000000000001073994
A0           0.3850730  0.3195341  1.2051075 0.228997450239672690
```

```
X1           0.1034213  0.4280875  0.2415892 0.809243706916227379
A1          -0.1715239  0.4212253 -0.4072024 0.684114853083405361
```

```
ltmle.fit2$fit$g[[1]]
```

```
$A0
               Estimate Std. Error   t value  Pr(>|t|)
(Intercept) -0.09470835 0.06663170 -1.421371 0.1553652
X0          -0.03949804 0.03927721 -1.005623 0.3147190
X0.squared   0.03702366 0.03792029  0.976355 0.3290068

$C1
              Estimate Std. Error    t value     Pr(>|t|)
(Intercept) -1.1960037  0.1324855 -9.0274285 8.300060e-19
X0           0.0334300  0.0597762  0.5592526 5.761095e-01
X0.squared   0.7983833  0.0644685 12.3840838 5.850710e-33
A0          -1.2015567  0.1534715 -7.8291862 1.202877e-14

$A1
              Estimate Std. Error   t value          Pr(>|t|)
(Intercept)  1.2032820  0.2539590  4.738095 0.000003168252174
X0          -1.3645602  0.2240725 -6.089815 0.000000003039204
X0.squared   0.3558945  0.1492711  2.384216 0.017660315374184
A0           0.4106480  0.2809646  1.461565 0.144778651394808
X1          -0.4976473  0.3091209 -1.609879 0.108346808036168
```

Task 19.

```
ltmle.fit2.missQ <- ltmle(sim.data2,
              Anodes=paste0("A", 0:1),
              Lnodes=c("X0", "X0.squared", "X1"),
              Cnodes="C1",
              Ynodes=paste0("D", 1:2),
              Qform=c(D1="Q.kplus1~X0+A0",
                  X1="Q.kplus1~X0+A0",
                  D2="Q.kplus1~X0+A0+X1+A1"),
              gform=c(A0="A0~X0",
                  C1="C1~X0.squared+A0",
                  A1="A1~X0+A0+X1"),
              abar=list(treatment=c(0,1), control=c(0,0)),
              survivalOutcome=TRUE)
summary(ltmle.fit2.missQ)
```

```
Treatment Estimate:
   Parameter Estimate:  0.71255
    Estimated Std Err:  0.02244
              p-value:  <2e-16
    95% Conf Interval: (0.66857, 0.75653)

Control Estimate:
   Parameter Estimate:  0.74247
```

```
    Estimated Std Err:  0.05218
              p-value:  <2e-16
    95% Conf Interval: (0.6402, 0.84474)

Additive Treatment Effect:
   Parameter Estimate:  -0.029919
    Estimated Std Err:  0.055011
              p-value:  0.58653
    95% Conf Interval: (-0.13774, 0.077901)

Relative Risk:
   Parameter Estimate:  0.9597
  Est Std Err log(RR):  0.074517
              p-value:  0.58097
    95% Conf Interval: (0.82929, 1.1106)

Odds Ratio:
   Parameter Estimate:  0.85981
  Est Std Err log(OR):  0.28525
              p-value:  0.59646
    95% Conf Interval: (0.49159, 1.5039)
```