

# Day 4, Practical 2, Hely's solution

Helene Charlotte Wiese Rytgaard

May 16, 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:

$A0 \sim X0.1 + X0.2 + X0.3$

formula for A1:

$A1 \sim X0.1 + X0.2 + X0.3 + A0 + X1.1 + X1.2$

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 b

Estimator: tmlle

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:

$Q.kplus1 \sim X0.1 + X0.2 + X0.3 + A0$

formula for Y:

$Q.kplus1 \sim X0.1 + X0.2 + X0.3 + A0 + X1.1 + X1.2 + A1$

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 b
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:

```

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:      -0.009317000000000002
static:   -0.063294
dynamic:  -0.050709

```

### Task 3: Estimating the static effect.

```

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:")

```

```

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:

$Q.kplus1 \sim X0.1 + X0.2 + X0.3 + A0$

formula for Y:

$Q.kplus1 \sim X0.1 + X0.2 + X0.3 + A0 + X1.1 + X1.2 + A1$

gform not specified, using defaults:

formula for A0:

$A0 \sim X0.1 + X0.2 + X0.3$

formula for A1:

$A1 \sim X0.1 + X0.2 + X0.3 + A0 + X1.1 + X1.2$

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 b

-----  
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.

```

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 b

-----

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.



```

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:

$Q.kplus1 \sim X0.1 + X0.2 + X0.3 + A0$

gform not specified, using defaults:

formula for A0:

$A0 \sim X0.1 + X0.2 + X0.3$

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.

```

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:

$Q.kplus1 \sim X0.1 + X0.2 + X0.3 + A0$

formula for Y:

$Q.kplus1 \sim X0.1 + X0.2 + X0.3 + A0 + X1.1 + X1.2 + A1$

gform not specified, using defaults:

formula for A0:

$A0 \sim X0.1 + X0.2 + X0.3$

formula for A1:

$A1 \sim X0.1 + X0.2 + X0.3 + A0 + X1.1 + X1.2$

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 b

\$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.

```
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.

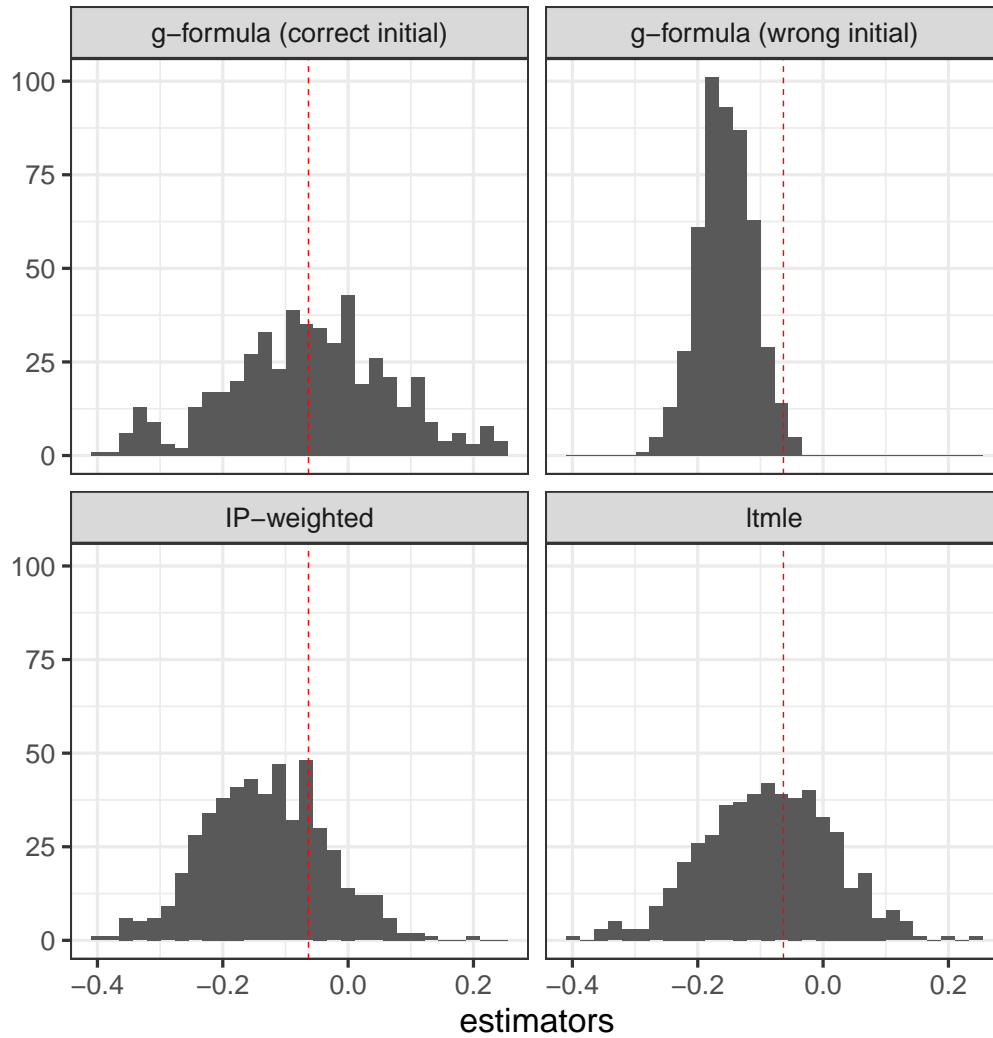


Figure 1

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

```

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

```

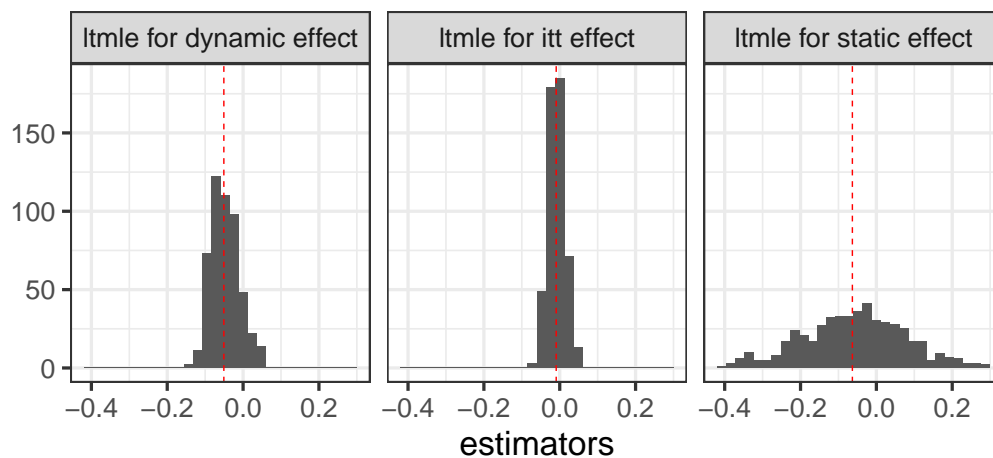


Figure 2

### 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	0.00000000
SL.mean_All	0.06689833	0.00000000
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