Day 2, Practical 1, Hely's solution

Helene Charlotte Wiese Rytgaard

September 28, 2021

Task 1: Simulation function from the practical from yesterday:

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

```
      id          X1          X2 X3 A Y
   1:    1  0.4084562  0.38996075  0 0 0
   2:    2 -1.2198243 -1.67449303  1 0 0
   3:    3  1.8658349 -2.22881407  0 1 1
   4:    4  0.6036221 -0.01388672  0 0 0
   5:    5 -0.5317124  0.57686435  0 0 0
  ---
 996:  996  1.6989517  0.14755236  0 1 1
 997:  997 -1.5151272  0.22514534  0 0 1
 998:  998 -1.4508899  0.31307290  0 0 1
 999:  999 -0.1766132 -1.60064177  0 0 0
1000: 1000  0.6122651  0.79204417  0 1 1
```

1 Implementing the targeting step for the treatment-specific mean $\Psi_a(P_0)$

Task 2:

```
target.fun <- function(d, a) {

    ##-- 1:
    fit.f <- glm(Y~A+X1+X2+X3, family=binomial, data=d)
    fit.pi <- glm(A~X1+X2+X3, family=binomial, data=d)

    ##-- 2:
    d[, pred.fa:=predict(fit.f, type="response", newdata=copy(d)[, A:=a])]
    d[, pred.f:=predict(fit.f, type="response", newdata=d)]

    ##-- 3:
    d[, pred.pi1:=predict(fit.pi, type="response", newdata=d)]

    ##-- 4:
    d[, Ha:=(A==a)/(pred.pi1^a*(1-pred.pi1)^{1-a})]

    ##-- 5:
    fit.tmle <- glm(Y ~ offset(qlogis(pred.f)) + Ha -1, family=binomial, data=d)
    eps.hat <- fit.tmle$coef
```

```
    ##-- 6:
    d[, pred.fa.1:=plogis(qlogis(pred.fa) + eps.hat/(pred.pi1^a*(1-pred.pi1)^{1-a}))]
    d[, pred.f.1:=plogis(qlogis(pred.f) + eps.hat*Ha)]

    ##-- 7:
    tmle.est <- d[, mean(pred.fa.1)]

    ##-- 8:
    print(paste0("eic solved at level = ", d[, mean(Ha*(Y-pred.f.1) + pred.fa.1 - tmle
    .est)]))

    ##-- 9:
    tmle.se <- d[, sqrt(mean((Ha*(Y-pred.f.1) + pred.fa.1 - tmle.est)^2)/nrow(d))]

    return(c(tmle.est=tmle.est, tmle.se=tmle.se))
}
```

Task 3.

```
(est.0 <- target.fun(d=sim.data, a=0))
(est.1 <- target.fun(d=sim.data, a=1))
```

```
[1] "eic solved at level = 0.000000000000003636503520998 34"
  tmle.est    tmle.se
0.69462449 0.02091243
[1] "eic solved at level = 7.30553088184628e-15"
 tmle.est   tmle.se
0.7605273 0.0210355
```

Task 4:

```
target.fun <- function(d, a, weighted=FALSE) {

    ##-- 1:
    fit.f <- glm(Y~A+X1+X2+X3, family=binomial, data=d)
    fit.pi <- glm(A~X1+X2+X3, family=binomial, data=d)

    ##-- 2:
    d[, pred.fa:=predict(fit.f, type="response", newdata=copy(d)[, A:=a])]
    d[, pred.f:=predict(fit.f, type="response", newdata=d)]

    ##-- 3:
    d[, pred.pi1:=predict(fit.pi, type="response", newdata=d)]

    ##-- 4:
    d[, Ha:=(A==a)/(pred.pi1^a*(1-pred.pi1)^{1-a})]

    ##-- 5:
    if (!weighted) {
    fit.tmle <- glm(Y ~ offset(qlogis(pred.f)) + Ha -1, family=binomial, data=d)
    } else {
    fit.tmle <- glm(Y ~ offset(qlogis(pred.f)), weights=Ha, family=binomial, data=d)
    }
    eps.hat <- fit.tmle$coef
```

```
    ##-- 6:
    if (!weighted) {
    d[, pred.fa.1:=plogis(qlogis(pred.fa) + eps.hat/(pred.pi1^a*(1-pred.pi1)^{1-a}))]
    d[, pred.f.1:=plogis(qlogis(pred.f) + eps.hat*Ha)]
    } else {
    d[, pred.fa.1:=plogis(qlogis(pred.fa) + eps.hat)]
    d[, pred.f.1:=plogis(qlogis(pred.f) + eps.hat)]
    }

    ##-- 7:
    tmle.est <- d[, mean(pred.fa.1)]

    ##-- 8:
    print(paste0("eic solved at level = ", d[, mean(Ha*(Y-pred.f.1) + pred.fa.1 - tmle
    .est)]))

    ##-- 9:
    tmle.se <- d[, sqrt(mean((Ha*(Y-pred.f.1) + pred.fa.1 - tmle.est)^2)/nrow(d))]

    return(c(tmle.est=tmle.est, tmle.se=tmle.se))
}
```

```
target.fun(d=sim.data, a=0, weighted=TRUE)
target.fun(d=sim.data, a=1, weighted=TRUE)
```

```
[1] "eic solved at level = 0.000000000000098107623896582"
 tmle.est   tmle.se
[1] "eic solved at level = 0.000000000321081363349095"
  tmle.est    tmle.se
0.76090428 0.02087738
```

Note: Can remove warnings by using `quasibinomial` family rather than `binomial`.

2 Computing the variances of the ATE, the $\log \mathrm{RR}$ and the $\log \mathrm{OR}$

Task 5.

```
print(paste0("est ate = ", est.ate <- est.1["tmle.est"] - est.0["tmle.est"]))
print(paste0("est rr = ", est.rr <- est.1["tmle.est"] / est.0["tmle.est"]))
print(paste0("est or = ", est.or <- (est.1["tmle.est"] / (1-est.1["tmle.est"])) / (est
    .0["tmle.est"] / (1-est.0["tmle.est"]))))
```

```
[1] "est ate = 0.0659028514677774"
[1] "est rr = 1.09487550876992"
[1] "est or = 1.39618513268535"
```

Task 6. We can write the **log-RR** as

$$\log \Psi_{\mathrm{RR}}(P) = \log\left(\frac{\Psi_1(P)}{\Psi_0(P)}\right) = \log(\Psi_1(P)) - \log(\Psi_0(P)) = h(\Psi_1(P)) - h(\Psi_0(P)),$$

where h is defined as

$$h(\psi) = \log \psi.$$

We compute the derivative

$$h(\psi) = \frac{1}{\psi},$$

thus, we have that,

$$\phi^*_{1,h}(P) = \frac{1}{\Psi_1(P)}\phi^*_1(P) \quad \text{and} \quad \phi^*_{0,h}(P) = \frac{1}{\Psi_0(P)}\phi^*_0(P).$$

Similarly, we can write the **log odds ratio** as

$$\log \Psi_{\text{RR}}(P) = \log\left(\frac{\Psi_1(P)}{1-\Psi_1(P)}\right) - \log\left(\frac{\Psi_0(P)}{1-\Psi_0(P)}\right) = h(\Psi_1(P)) - h(\Psi_0(P)),$$

where h is defined as

$$h(\psi) = \log\left(\frac{\psi}{1-\psi}\right) = \log(\psi) - \log(1-\psi).$$

We compute the derivative as

$$\frac{d}{d\psi}h(\psi) = \frac{1}{\psi} + \frac{1}{1-\psi} = \frac{1-\psi+\psi}{\psi(1-\psi)} = \frac{1}{\psi(1-\psi)},$$

so that

$$\phi^*_{1,h}(P) = \frac{1}{\Psi_1(P)(1-\Psi_1(P))}\phi^*_1(P) \quad \text{and} \quad \phi^*_{0,h}(P) = \frac{1}{\Psi_0(P)(1-\Psi_0(P))}\phi^*_0(P).$$

Task 7.

```
target.fun <- function(d, a) {

    ##-- 1:
    fit.f <- glm(Y~A+X1+X2+X3, family=binomial, data=d)
    fit.pi <- glm(A~X1+X2+X3, family=binomial, data=d)

    ##-- 2:
    d[, pred.fa:=predict(fit.f, type="response", newdata=copy(d)[, A:=a])]
    d[, pred.f:=predict(fit.f, type="response", newdata=d)]

    ##-- 3:
    d[, pred.pi1:=predict(fit.pi, type="response", newdata=d)]

    ##-- 4:
    d[, Ha:=(A==a)/(pred.pi1^a*(1-pred.pi1)^{1-a})]

    ##-- 5:
    fit.tmle <- glm(Y ~ offset(qlogis(pred.f)) + Ha -1, family=binomial, data=d)
    eps.hat <- fit.tmle$coef

    ##-- 6:
```

```
    d[, pred.fa.1:=plogis(qlogis(pred.fa) + eps.hat/(pred.pi1^a*(1-pred.pi1)^{1-a}))]
    d[, pred.f.1:=plogis(qlogis(pred.f) + eps.hat*Ha)]

    ##-- 7:
    tmle.est <- d[, mean(pred.fa.1)]

    ##-- 8:
    print(paste0("eic solved at level = ", d[, mean(Ha*(Y-pred.f.1) + pred.fa.1 - tmle
    .est)]))

    ##-- 9:
    tmle.se <- d[, sqrt(mean((Ha*(Y-pred.f.1) + pred.fa.1 - tmle.est)^2)/nrow(d))]

    return(list(tmle.est=tmle.est, tmle.se=tmle.se, eic=d[, (Ha*(Y-pred.f.1) + pred.fa
    .1 - tmle.est)]))
}
```

Task 8.

```
eic.0 <- target.fun(d=sim.data, a=0)[["eic"]]
eic.1 <- target.fun(d=sim.data, a=1)[["eic"]]
```

```
print(paste0("ate eic solved at level = ", (mean(eic.1 - eic.0))))
print(paste0("log-rr eic solved at level = ", (mean(1/est.1[["tmle.est"]]*eic.1 - 1/
    est.0[["tmle.est"]]*eic.0))))
print(paste0("log-or eic solved at level = ", (mean(1/(est.1[["tmle.est"]]*(1-est.1[["
    tmle.est"]]))*eic.1 -
                            1/(est.0[["tmle.est"]]*(1-est.0[["tmle.est"]]))*eic.0))))
```

```
[1] "ate eic solved at level = -0.000000000000002905924050109O9"
[1] "log-rr eic solved at level = -0.000000000000004274944523266 32"
[1] "log-or eic solved at level = -0.000000000000013130903569330 6"
```

```
print(paste0("ate var = ", (mean((eic.1 - eic.0)^2/nrow(sim.data)))))
print(paste0("log-rr var = ", (mean((1/est.1[["tmle.est"]]*eic.1 - 1/est.0[["tmle.est"
    ]]*eic.0)^2/nrow(sim.data)))))
print(paste0("log-or var = ", (mean((1/(est.1[["tmle.est"]]*(1-est.1[["tmle.est"]]))*
    eic.1 -
                    1/(est.0[["tmle.est"]]*(1-est.0[["tmle.est"]]))*eic.0)^2/nrow(sim.
    data)))))
```

```
[1] "ate var = 0.000861914952736296"
[1] "log-rr var = 0.00163750576793132"
[1] "log-or var = 0.022596156195927"
```

```
print(paste0("est ate = ", est.ate))
print(paste0("est log-rr = ", log(est.rr)))
print(paste0("est log-or = ", log(est.or)))
```

```
[1] "est ate = 0.0659028514677774"
[1] "est log-rr = 0.0906406661841214"
[1] "est log-or = 0.333743612084167"
```

Task 9.

```
library(tmle)
fit.tmle <- tmle(Y=sim.data$Y, A=sim.data$A,
         cbind(X1=sim.data$X1,X2=sim.data$X2,X3=sim.data$X3),
         gform=A~X1+X2+X3, ## treatment model
         Qform=Y~A+X1+X2+X3, ## outcome model
         family="binomial",
         cvQinit=FALSE)
```

```
fit.tmle
```

```
Additive Effect
  Parameter Estimate:  0.066263
  Estimated Variance:  0.00085811
             p-value:  0.023694
   95% Conf Interval: (0.0088482, 0.12368)

Additive Effect among the Treated
  Parameter Estimate:  0.072104
  Estimated Variance:  0.0009739
             p-value:  0.020862
   95% Conf Interval: (0.010938, 0.13327)

Additive Effect among the Controls
  Parameter Estimate:  0.059976
  Estimated Variance:  0.0009839
             p-value:  0.055869
   95% Conf Interval: (-0.0015039, 0.12146)

Relative Risk
  Parameter Estimate:  1.0954
             p-value:  0.024072
   95% Conf Interval: (1.012, 1.1856)

              log(RR):  0.091113
   variance(log(RR)):  0.0016311

Odds Ratio
  Parameter Estimate:  1.399
             p-value:  0.025166
   95% Conf Interval: (1.0427, 1.877)

              log(OR):  0.33574
   variance(log(OR)):  0.022488
```

3 Simulation study

Task 10.

```
fit.tmle1.est <- list()
```

```
fit.tmle2.est <- list()
fit.tmle3.est <- list()
fit.tmle1.se <- list()
fit.tmle2.se <- list()
fit.tmle3.se <- list()

for (m in 1:500) {

    set.seed(m+110)
    sim.data <- sim.fun(1000)

    fit.tmle1 <- tmle(Y=sim.data$Y, A=sim.data$A,
              cbind(X1=sim.data$X1,X2=sim.data$X2,X3=sim.data$X3),
              gform=A~X1+X2+X3, ## treatment model
              Qform=Y~A+X1+X2+X3, ## outcome model
              family="binomial",
              cvQinit=FALSE)

    fit.tmle1.est[[m]] <- fit.tmle1$estimates$ATE$psi
    fit.tmle1.se[[m]] <- sqrt(fit.tmle1$estimates$ATE$var.psi)

    fit.tmle2 <- tmle(Y=sim.data$Y, A=sim.data$A,
              cbind(X1=sim.data$X1,X1.squared=sim.data$X1^2,X2=sim.data$X2,X3=sim.data
    $X3),
              gform=A~X1+X2+X3, ## treatment model
              Qform=Y~A+X1.squared+X2+X3, ## outcome model
              family="binomial",
              cvQinit=FALSE)

    fit.tmle2.est[[m]] <- fit.tmle2$estimates$ATE$psi
    fit.tmle2.se[[m]] <- sqrt(fit.tmle2$estimates$ATE$var.psi)

    fit.tmle3 <- tmle(Y=sim.data$Y, A=sim.data$A,
              cbind(X1=sim.data$X1,X2=sim.data$X2,X3=sim.data$X4),
              Q.SL.library=c("SL.glm", "SL.mean", "SL.gam"),
              g.SL.library=c("SL.glm", "SL.mean", "SL.gam"),
              family="binomial")

    fit.tmle3.est[[m]] <- fit.tmle3$estimates$ATE$psi
    fit.tmle3.se[[m]] <- sqrt(fit.tmle3$estimates$ATE$var.psi)

}
```

Task 11. See Figure 1 and comments further below.

```
message(paste0("EY1 = ", E.Y1 <- sim.fun(1e6, a=1)))
message(paste0("EY0 = ", E.Y0 <- sim.fun(1e6, a=0)))
message(paste0("ATE = ", ATE <- E.Y1 - E.Y0))
```

```
EY1 = 0.749266
EY0 = 0.683064
ATE = 0.066202
```

```
setwd("~/Undervisning/TMLE/beamer/day2/")
```

```
library(ggplot2)

pdat <- data.table(estimator=c(rep("TMLE estimator (misspecified initial)",
                   length(fit.tmle1.est)),
                   rep("TMLE estimator (correctly specified initial)",
                   length(fit.tmle2.est)),
                   rep("TMLE estimator (simple super learner for initial)",
                   length(fit.tmle3.est))),
           est=c(unlist(fit.tmle1.est),
             unlist(fit.tmle2.est),
             unlist(fit.tmle3.est)))

ggplot(pdat) +
    theme_bw(base_size=25) +
    geom_boxplot(aes(x=est)) +
    facet_wrap(. ~ estimator, ncol=2) +
    geom_vline(aes(xintercept=ATE), linetype="dashed", color="red") +
    xlab(expression(hat(psi)[n])) + ylab("")
```

```
message(paste0("bias tmle1 = ", mean(unlist(fit.tmle1.est))-ATE))
message(paste0("bias tmle2 = ", mean(unlist(fit.tmle2.est))-ATE))
message(paste0("bias tmle3 = ", mean(unlist(fit.tmle3.est))-ATE))
message(paste0("variance tmle1 = ", var(unlist(fit.tmle1.est))))
message(paste0("variance tmle2 = ", var(unlist(fit.tmle2.est))))
message(paste0("variance tmle3 = ", var(unlist(fit.tmle3.est))))
message(paste0("coverage tmle1 = ", mean(ATE>=unlist(fit.tmle1.est)-1.96*unlist(fit.
    tmle1.se) &
                     ATE<=unlist(fit.tmle1.est)+1.96*unlist(fit.tmle1.se))))
message(paste0("coverage tmle2 = ", mean(ATE>=unlist(fit.tmle2.est)-1.96*unlist(fit.
    tmle2.se) &
                     ATE<=unlist(fit.tmle2.est)+1.96*unlist(fit.tmle2.se))))
message(paste0("coverage tmle3 = ", mean(ATE>=unlist(fit.tmle3.est)-1.96*unlist(fit.
    tmle3.se) &
                     ATE<=unlist(fit.tmle3.est)+1.96*unlist(fit.tmle3.se))))
```

```
bias tmle1 = 0.00141930436233162
bias tmle2 = 0.000594083168194162
bias tmle3 = 0.000852184104645035
variance tmle1 = 0.000896527774711243
variance tmle2 = 0.000557961015161351
variance tmle3 = 0.000579126141824112
coverage tmle1 = 0.938
coverage tmle2 = 0.958
coverage tmle3 = 0.95
```

Comments for Task 11. The simulation study illustrates a number of points:

1. The TMLE estimator is consistent when either π or f is estimated consistently.
2. Inference for the TMLE estimator is only valid when based on correctly specified models (or SL capturing same information well enough).

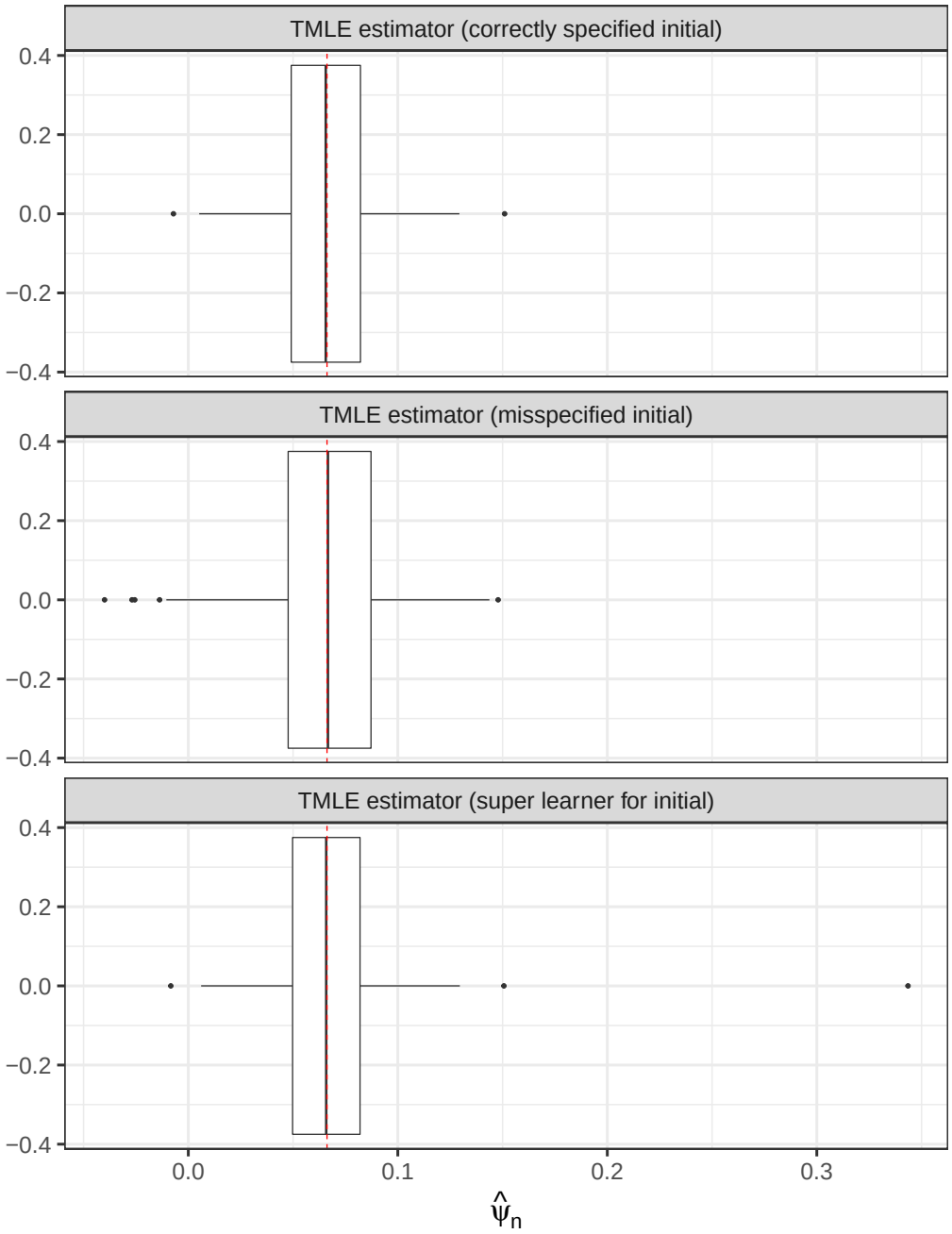
TMLE estimator (correctly specified initial)
TMLE estimator (misspecified initial)
TMLE estimator (super learner for initial)
0.4
0.2
0.0
−0.2
−0.4
0.0
0.1
0.2
0.3
$\hat{\psi}_n$

Figure 1

4 Extra simulation study

Here we remove X_1 as a confounder, i.e., the distribution of A depends not on X_1.

```
library(data.table)
new.sim.fun2 <- function(n, a=NULL) {
    X1 <- runif(n, -2, 2)
    X2 <- rnorm(n)
    X3 <- rbinom(n, 1, 0.2)
    if (length(a)>0) {
    A  <- a
    } else {
    A  <- rbinom(n, 1, prob=plogis(-0.25 + 0.4*X2 + 0.25*X3))
    }
    Y <- rbinom(n, 1, prob=plogis(-0.9 + 1.9*X1^2 + 0.6*X2 + 0.5*A))
    if (length(a)>0) {
    return(mean(Y))
    } else {
    return(data.table(id=1:n,X1=X1,X2=X2,X3=X3,A=A,Y=Y))
    }
}
```

```
fit.tmle1.est <- list()
fit.tmle2.est <- list()
fit.tmle3.est <- list()
fit.tmle1.se <- list()
fit.tmle2.se <- list()
fit.tmle3.se <- list()
fit.g.glm1 <- list()
fit.g.glm2 <- list()

for (m in 1:500) {

    set.seed(m+110)
    sim.data <- new.sim.fun2(1000)

    fit.f <- glm(Y~A+X1+X2+X3, family=binomial, data=sim.data)
    fit.f2 <- glm(Y~A+X1.squared+X2+X3, family=binomial, data=sim.data[, X1.squared:=
    X1^2])

    ##-- g-formula (section 3.1);
    sim.data[, pred.f1:=predict(fit.f, type="response", newdata=copy(sim.data)[, A
    :=1])]
    sim.data[, pred.f0:=predict(fit.f, type="response", newdata=copy(sim.data)[, A
    :=0])]
    fit.g.glm1[[m]] <- sim.data[, mean(pred.f1 - pred.f0)]

    ##-- g-formula (section 3.2);
    sim.data[, pred.f1:=predict(fit.f2, type="response", newdata=copy(sim.data)[, A
    :=1])]
    sim.data[, pred.f0:=predict(fit.f2, type="response", newdata=copy(sim.data)[, A
    :=0])]
    fit.g.glm2[[m]] <- sim.data[, mean(pred.f1 - pred.f0)]

    fit.tmle1 <- tmle(Y=sim.data$Y, A=sim.data$A,
```

```
              cbind(X1=sim.data$X1,X2=sim.data$X2,X3=sim.data$X3),
              gform=A~X1+X2+X3, ## treatment model
              Qform=Y~A+X1+X2+X3, ## outcome model
              family="binomial",
              cvQinit=FALSE)

    fit.tmle1.est[[m]] <- fit.tmle1$estimates$ATE$psi
    fit.tmle1.se[[m]] <- sqrt(fit.tmle1$estimates$ATE$var.psi)

    fit.tmle2 <- tmle(Y=sim.data$Y, A=sim.data$A,
              cbind(X1=sim.data$X1,X1.squared=sim.data$X1^2,X2=sim.data$X2,X3=sim.data
    $X3),
              gform=A~X1+X2+X3, ## treatment model
              Qform=Y~A+X1.squared+X2+X3, ## outcome model
              family="binomial",
              cvQinit=FALSE)

    fit.tmle2.est[[m]] <- fit.tmle2$estimates$ATE$psi
    fit.tmle2.se[[m]] <- sqrt(fit.tmle2$estimates$ATE$var.psi)

    fit.tmle3 <- tmle(Y=sim.data$Y, A=sim.data$A,
              cbind(X1=sim.data$X1,X1.squared=sim.data$X1^2,X2=sim.data$X2,X3=sim.data
    $X3),
              Q.SL.library=c("SL.glm", "SL.mean", "SL.gam"),
              g.SL.library=c("SL.glm", "SL.mean", "SL.gam"),
              family="binomial")

    fit.tmle3.est[[m]] <- fit.tmle3$estimates$ATE$psi
    fit.tmle3.se[[m]] <- sqrt(fit.tmle3$estimates$ATE$var.psi)

}
```

```
setwd("~/Undervisning/TMLE/beamer/day2/")
library(ggplot2)

pdat <- data.table(estimator=c(rep("TMLE estimator (misspecified initial)",
                   length(fit.tmle1.est)),
                   rep("TMLE estimator (correctly specified initial)",
                   length(fit.tmle2.est)),
                   rep("TMLE estimator (super learner for initial)",
                   length(fit.tmle3.est)),
                   rep("g-formula estimator (misspecified)",
                   length(fit.g.glm1)),
                   rep("g-formula estimator (correctly specified)",
                   length(fit.g.glm2))),
           est=c(unlist(fit.tmle1.est),
             unlist(fit.tmle2.est),
             unlist(fit.tmle3.est),
             unlist(fit.g.glm1),
             unlist(fit.g.glm2)))

ggplot(pdat) +
    theme_bw(base_size=25) +
    geom_boxplot(aes(x=est)) +
    facet_wrap(. ~ estimator, ncol=2) +
```

```
    geom_vline(aes(xintercept=ATE), linetype="dashed", color="red") +
    xlab(expression(hat(psi)[n])) + ylab("")
```

```
message(paste0("bias tmle1 = ", mean(unlist(fit.tmle1.est))-ATE))
message(paste0("bias tmle2 = ", mean(unlist(fit.tmle2.est))-ATE))
message(paste0("bias tmle3 = ", mean(unlist(fit.tmle3.est))-ATE))
message(paste0("variance tmle1 = ", var(unlist(fit.tmle1.est))))
message(paste0("variance tmle2 = ", var(unlist(fit.tmle2.est))))
message(paste0("variance tmle3 = ", var(unlist(fit.tmle3.est))))
message(paste0("coverage tmle1 = ", mean(ATE>=unlist(fit.tmle1.est)-1.96*unlist(fit.
    tmle1.se) &
                    ATE<=unlist(fit.tmle1.est)+1.96*unlist(fit.tmle1.se))))
message(paste0("coverage tmle2 = ", mean(ATE>=unlist(fit.tmle2.est)-1.96*unlist(fit.
    tmle2.se) &
                    ATE<=unlist(fit.tmle2.est)+1.96*unlist(fit.tmle2.se))))
message(paste0("coverage tmle3 = ", mean(ATE>=unlist(fit.tmle3.est)-1.96*unlist(fit.
    tmle3.se) &
                    ATE<=unlist(fit.tmle3.est)+1.96*unlist(fit.tmle3.se))))
```

```
bias tmle1 = -0.000951310654890167
bias tmle2 = 0.0000248405937115781
bias tmle3 = 0.0000897877462508467
variance tmle1 = 0.000902600222738588
variance tmle2 = 0.000629997544529498
variance tmle3 = 0.0006320532376377
coverage tmle1 = 0.946
coverage tmle2 = 0.94
coverage tmle3 = 0.936
```

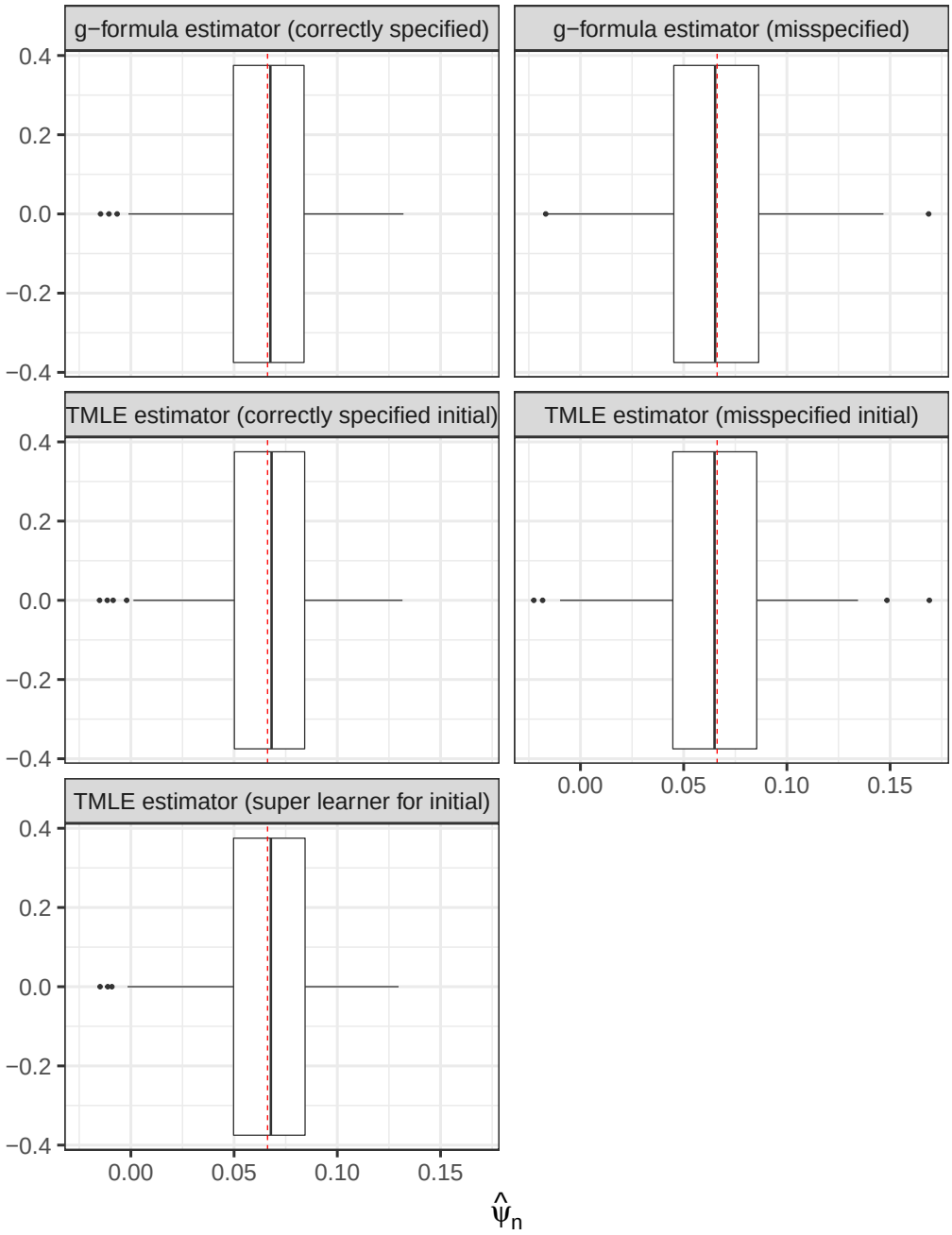
g−formula estimator (correctly specified)
g−formula estimator (misspecified)
TMLE estimator (correctly specified initial)
TMLE estimator (misspecified initial)
TMLE estimator (super learner for initial)
0.4
0.2
0.0
−0.2
−0.4
0.00
0.05
0.10
0.15
$\hat{\psi}_n$

Figure 2