

Causal Inference II: an application of TMLE

Elena Petridou, Valérie Valckx & Victoria Zillo

Content

Introduction

- Dataset & Variables
- Research question & Target estimator

Method

- Assumptions
- Defining TMLE
- Step by step TMLE
- TMLE with Superlearner

Results

- Results for each method
- Interpretation & Discussion

Introduction

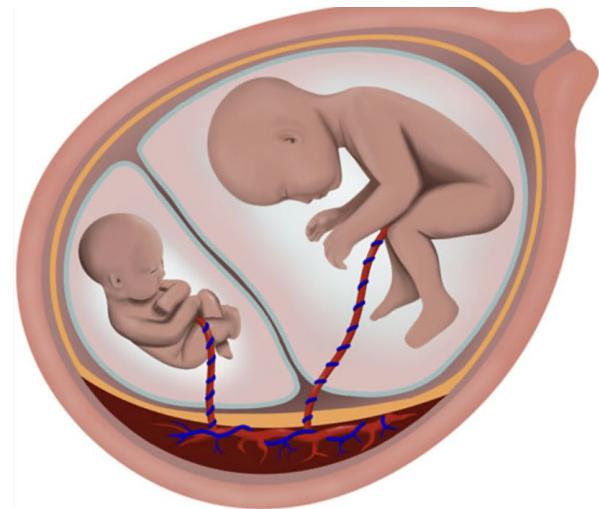
In some identical twin pregnancies, one baby grows much slower than the other → selective fetal growth restriction

This creates serious risks for both twins.

In severe cases, doctors may use an intervention to protect the healthier twin → selective reduction

We investigate whether this intervention actually improves the outcome for the larger twin.

To answer this, we use **Targeted Maximum Likelihood Estimation (TMLE)**, a modern and doubly robust causal method for observational data.



Dataset & variables

The dataset contains **264 pregnancies**, of which **252 remain** after removing rows with missing values (<5%).

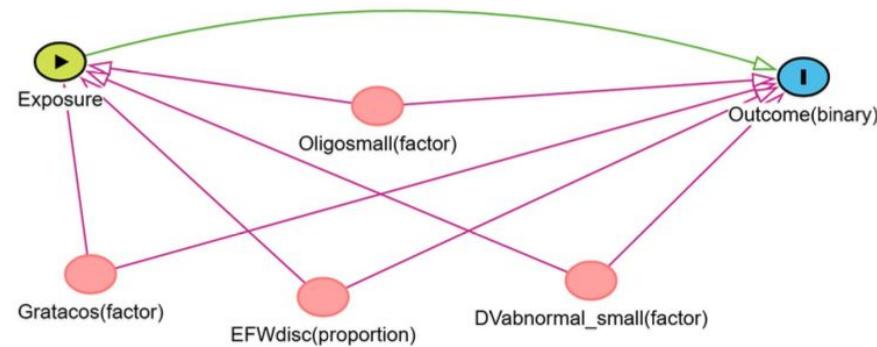
Treatment (A) is either **Conservative Management** or **Selective Reduction**.

Outcome (Y) is whether the larger twin had a **bad outcome**: death, birth before 32 weeks, or major neonatal morbidity.

We use **four covariates**:

- *Gratacos type*
- *DV abnormal small*
- *Oligohydramnios small*
- *EFW discordance* (continuous proportion)

These covariates are clinically important because they relate both to the decision to perform selective reduction and to the risk of a bad outcome.



Research question & target estimand

Research question:

Does selective reduction reduce the chance of a bad outcome for the larger twin?

Estimand:

The **Average Treatment Effect (ATE)** comparing:

- $Y(1)$: outcome if all received selective reduction
- $Y(0)$: outcome if all received conservative management

Formally: $E[Y(1)] - E[Y(0)]$

Method - TMLE

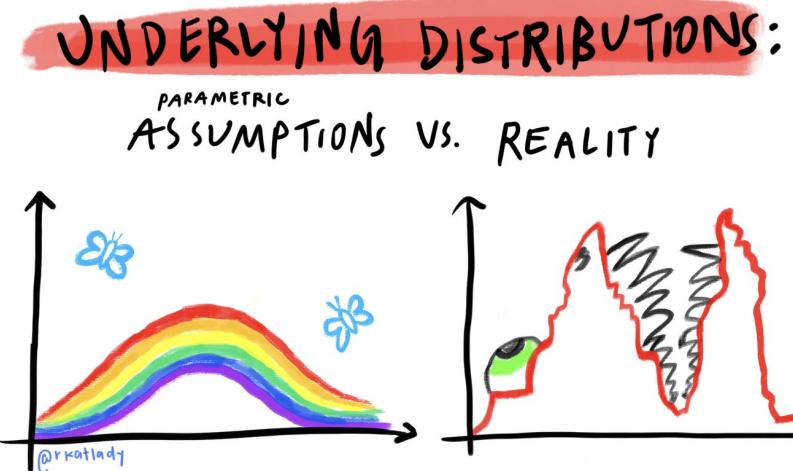
TMLE combines two components:

1. An **outcome model (Q)** predicting Y from A and the covariates.
2. A **treatment model (g)** (propensity score) predicting the probability of receiving selective reduction.

TMLE updates the initial outcome model using information from the treatment model → *doubly robust* estimate. This means that if either model is correctly specified, the causal estimate is still consistent.

Superlearner

TMLE also allows flexible machine-learning models, reducing the risk of misspecification.



TMLE assumptions

Consistency

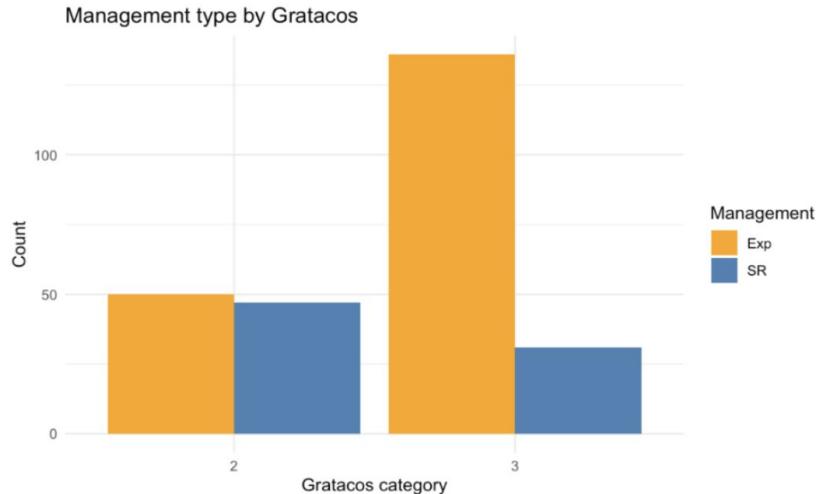
Not possible to check empirically but assume it holds as:

- Each treatment level corresponds to a well-defined intervention
- There is no interference between pregnancies

Positivity

For all covariates, both treatments occur at least once.

- Examining the distribution of the 2 management strategies across all covariates.
- In our data, both treatment types appear in all covariate strata.



TMLE Assumptions

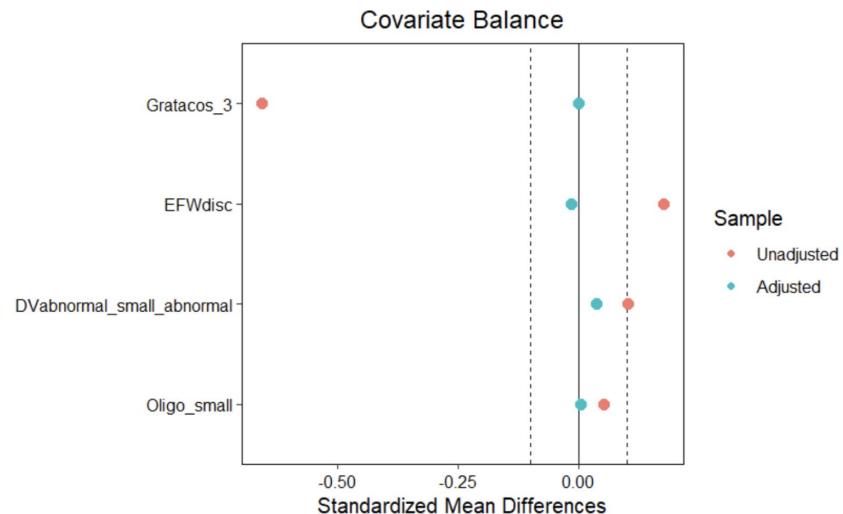
Conditional Exchangeability

After adjusting for the measured covariates, treatment assignment is comparable across groups.

- Checked by Love Plot
- Propensity score weighting and balance checks support this assumption

Correct specification of at least one of the two models

- Assuming it holds thanks to the use of Superlearner



Method - What is TMLE?

Targeted Maximum Likelihood Estimator

is a semiparametric double-robust method that improves
the chances of correct model specification by allowing for flexible estimation
using (nonparametric) machine-learning methods.

Target Estimator in the analysis is : ...

Method - TMLE step by step

STEP 1 - estimate outcome mechanism Q via outcome regression

$$\hat{Q}(A, W) = P(Y = 1 | A, W)$$

- For each woman in the sample, we get:
 - $Q(1, W)$: predicted probability of bad outcome in the larger twin under SR.
 - $Q(0, W)$: predicted probability of bad outcome in the larger twin under conservative management.

STEP 2 - estimate treatment mechanism g via clever covariates

$$H(A, W) = \frac{A}{\hat{g}(W)} - \frac{1 - A}{1 - \hat{g}(W)} \quad \text{where } \hat{g}(W) = P(A = 1 | W) \text{ is the propensity score: probability of receiving treatment (S.R) given covariates}$$

Method - TMLE step by step

STEP 3 - targeting: fit logistic regression with Y observed

Formula: $Y \sim -1 + \text{offset}(\text{qlogis}(QAW)) + \boldsymbol{\varepsilon}^* H(A, W)$

- Y is the observed outcome per subject
- $\text{offset}(\text{qlogis}(QAW))$ is a fixed term and it's $P(Y=1|A, W)$ on logit scale
- $H(A, W)$ clever covariate per subject

STEP 4 - predict updated outcomes $\text{logit}(Q^*(A, W)) = \text{logit}(Q(A, W)) + \epsilon H(A, W)$

- estimate mle $\hat{\boldsymbol{\varepsilon}} = (\hat{\varepsilon}_0, \hat{\varepsilon}_1)$ from the logistic model above and update:

$$\text{logit}(Q^*(A=1, W)) = \text{logit}(Q(A=1, W)) + \hat{\varepsilon}_1 * H(A=1, W) \text{ and } \text{logit}(Q^*(A=0, W)) = \text{logit}(Q(A=0, W)) + \hat{\varepsilon}_0 * H(A=0, W)$$

Method - TMLE step by step

STEP 5 - compute updated Average Treatment Effect (ATE)

- After updating the outcome predictions to $Q(A,W)$, TMLE computes:

$$\widehat{ATE} = \frac{1}{n} \sum_i \left[\tilde{Q}(1, W_i) - \tilde{Q}(0, W_i) \right]$$

- Variance and 95% CI
- In practice, we just extract them from the `tmle` object.

```
ATE_tmle  <- TMLE_SL$estimates$ATE$psi  
CI_tmle   <- TMLE_SL$estimates$ATE$CI
```

Method - TMLE with SuperLearner

What is SuperLearner?

SuperLearner is an algorithm that:

- Can use **non-parametric machine learning models**.
- Uses **cross-validation** to evaluate all models in the library.
- Does **not** simply choose the single model with the lowest MSE.
Instead, it **tests all weighted combinations** of the candidate models.
- Selects the **optimal weighted combination** that minimizes the cross-validated MSE.

Method - TMLE with SuperLearner

How is used in tmle() from tmle package on R?

By specifying the library argument in tmle() function , we allow Superlearner to estimate:

- the **outcome regression** – Step 1
 - the **clever covariates** – Step 2
- using **non-parametric machine learning models.**

Results - Only outcome regression

Recall: TMLE uses targeting to adjust the estimation of the **outcome mechanism Q** (from step 1) using information from the **treatment mechanism g** (from step 2)

Outcome mechanism Q was estimated in Causal Inference I via **outcome regression**

This estimate came out to **-0.115**

Interpretation:

“Selective reduction leads to an estimated 11.5% lower risk of a bad outcome (death, born before 32 weeks or neonatal morbidity) for the larger twin”

Results - tmle package with only GLMs

Now we adjust this estimate using the information obtained by estimating the **treatment mechanism g** (from step 2) via TMLE **targeting** (step 3)

The tmle package allows us to specify whichever methods we want for estimating **Q** (step 1) and for estimating **g** (step 2).

If specifying only linear methods via **SL.library** arguments, we can obtain the same estimate as the manual steps via tmle() function.

```
tmle_same_as_manual = tmle(Y= data_clean$event, A= data_clean$ManagementType, W= data_clean[, c("Gratacos", "EFWdisc", "DVabnormal_small", "Oligo_small")], family = "binomial", Q.SL.library = c("SL.glm"), g.SL.library = c("SL.glm"))
```

The estimate is approximately the same:

- **-0.0868**
- 95% Confidence Interval: **(-0.239, 0.066)**
- p-value: **0.266**
- standard error: **0.078**

Interpretation:

"The confidence interval contains zero, so we cannot say that at $\alpha=0.05$ level there is a significant effect of selective reduction on the outcome for the larger twin."

Results - tmle package with superlearner default methods

The default methods with the superlearner for tmle() are:

- For estimating **Q** (step 1):

```
Q.SL.library = c("SL.glm", "tmle.SL.dbarts2", "SL.glmnet")
```

- For estimating **g** (step 2):

```
g.SL.library = c("SL.glm", "tmle.SL.dbarts.k.5", "SL.gam")
```

The estimate when using these default models is:

- **-0.237**
- 95% Confidence Interval: **(-0.345, -0.130)**
- p-value: **<0.001**
- standard error: **0.054**

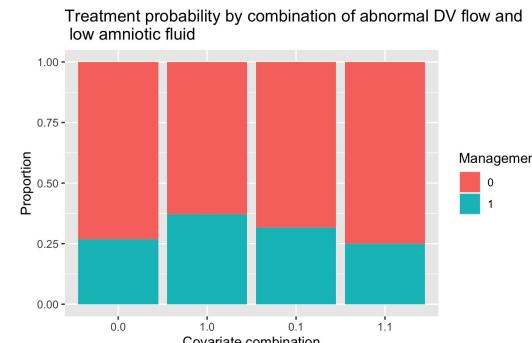
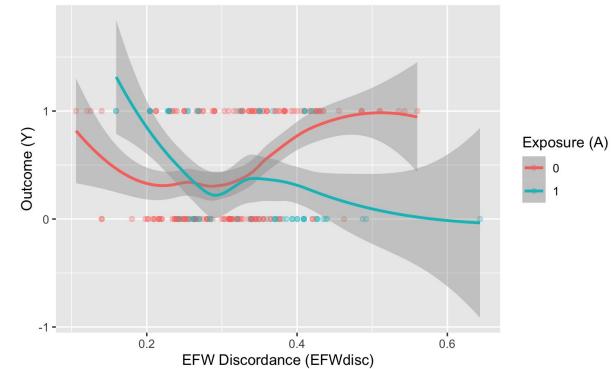
Interpretation:

*"Selective reduction leads to an estimated **23.7% lower probability of a bad outcome** (death, born before 32 weeks or neonatal morbidity) for the larger twin"*

Why such a different estimate? (1)

The difference could be due to:

1. Non-linearity in outcome mechanism Q : treatment-covariate interactions behave differently for different levels/values of the covariate
2. Non-linearity in treatment mechanism g : which could be caused by covariate-covariate interactions



Why such a different estimate? (2)

- Gain in **precision** - indicates the flexible models pick up more of the variation in the data
 - S.E. of linear tmle = 0.0781
 - S.E. of superlearner tmle = 0.0547
- Inspect the selected learners for each mechanism
 - For **Q**: only **Bayesian Additive Regression Trees (BART)** - flexible, non-linear method - selected
 - For **g**: also only **BART** selected

```
Initial estimation of Q
Procedure: cv-SuperLearner, ensemble
Model:
Y ~ SL.glm_All + tmle.SL.dbarts2_All + SL.glmnet_All
```

```
Coefficients:
SL.glm_All      0
tmle.SL.dbarts2_All    1
SL.glmnet_All      0
```

Cross-validated pseudo R squared : 0.045

```
Estimation of g (treatment mechanism)
Procedure: SuperLearner, ensemble
Model:
```

```
A ~ SL.glm_All + tmle.SL.dbarts.k.5_All + SL.gam_All
```

```
Coefficients:
SL.glm_All      0
tmle.SL.dbarts.k.5_All    1
SL.gam_All      0
```

Why such a different estimate? (3)

- Recall: ϵ (estimated coefficients) is the extent to which we need to adjust our estimates from **step 1**, so larger ϵ means more adjustment during targeting.
- Compare:

- Coefficients for linear methods tmle:

```
tmle_same_as_manual$epsilon  
H0W          H1W  
0.009039944 0.032192209
```

- Coefficients for default methods tmle (BART for both mechanisms)

```
TMLE_default$epsilon  
H0W          H1W  
0.04042470 -0.07460934
```

Interpretation:

“Estimated coefficients have a greater magnitude in the tmle model with non-linear methods. Thus, estimate from step 1 was adjusted more.”

Why such a different estimate? (4)

Bottom Line:

*Regardless of where the model misspecification is coming from, **flexible models** are a good idea as they are able to model **non-linear relationships** and **interactions** better than linear models, as they make **no assumptions** about the underlying distribution of the data. This could explain why BART was selected by superlearner, as it is good at modelling interactions.*

Results - tmle package with additional superlearner methods (1)

Add even more flexible estimators, as well as linear models with interactions:

```
Q_extras = c("SL.glm", "SL.glm.interaction", "SL.step", "SL.step.interaction",
"tmle.SL.dbarts2", "SL.glm.interaction", "SL.gam", "SL.randomForest", "SL.rpart",
"SL.xgboost")
g_extras = c("SL.glm", "SL.glm.interaction", "SL.step", "SL.step.interaction",
"tmle.SL.dbarts.k.5", "SL.glm.interaction", "SL.gam", "SL.randomForest", "SL.rpart",
"SL.xgboost")
```
TMLE_extra_models= tmle(Y= data_clean$event, A= data_clean$ManagementType, W=
data_clean[, c("Gratacos", "EFWdisc", "DVabnormal_small", "Oligo_small")], family =
"binomial", Q.SL.library = Q_extras,g.SL.library = g_extras)
```
``
```

The estimate now is:

- **-0.265**
- 95% Confidence Interval: **(-0.355 -0.175)**
- p-value: **<0.001**
- standard error: **0.046**

Interpretation:

*"Selective reduction leads to an estimated **26.5% lower probability of a bad outcome** (death, born before 32 weeks or neonatal morbidity) for the larger twin"*

Results - tmle package with additional superlearner methods (2)

- Further gain in **precision**
 - S.E. of tmle with default learners = 0.0547
 - S.E. with additional learners = 0.046

Results - tmle package with additional superlearner methods (3)

For estimating outcome mechanism \mathbf{Q}

```
Initial estimation of Q
Procedure: cv-SuperLearner, ensemble
Model:
Y ~ SL.glm_All + SL.glm.interaction_All + SL.step_All +
SL.step.interaction_All + tmle.SL.dbarts2_All + SL.glm.interaction_All + SL.gam_All
+ SL.randomForest_All + SL.rpart_All + SL.xgboost_All

Coefficients:
SL.glm_All      0
SL.glm.interaction_All    0
SL.step_All      0
SL.step.interaction_All   0.1132083
tmle.SL.dbarts2_All    0
SL.glm.interaction_All    0
SL.gam_All      0
SL.randomForest_All   0.2008734
SL.rpart_All     0.09112411
SL.xgboost_All    0.5947942

Cross-validated pseudo R squared : 0.1663
```

Results - tmle package with additional superlearner methods (4)

For estimating treatment mechanism **g**

```
Estimation of g (treatment mechanism)
Procedure: SuperLearner, ensemble
Model:
A ~ SL.glm_All + SL.glm.interaction_All + SL.step_All +
SL.step.interaction_All + tmle.SL.dbarts.k.5_All + SL.glm.interaction_All +
SL.gam_All + SL.randomForest_All + SL.rpart_All + SL.xgboost_All
```

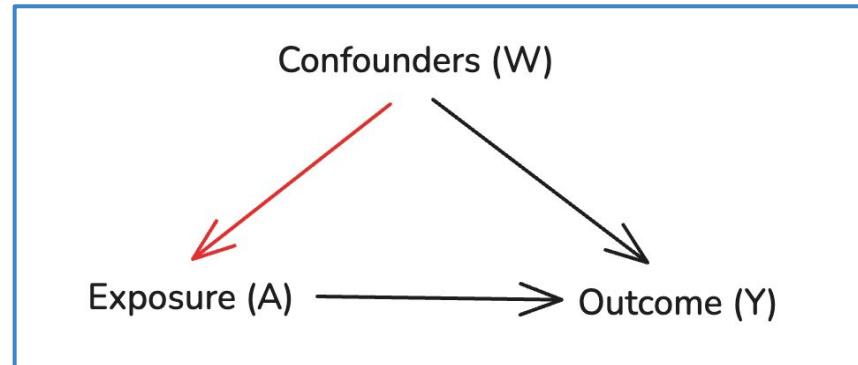
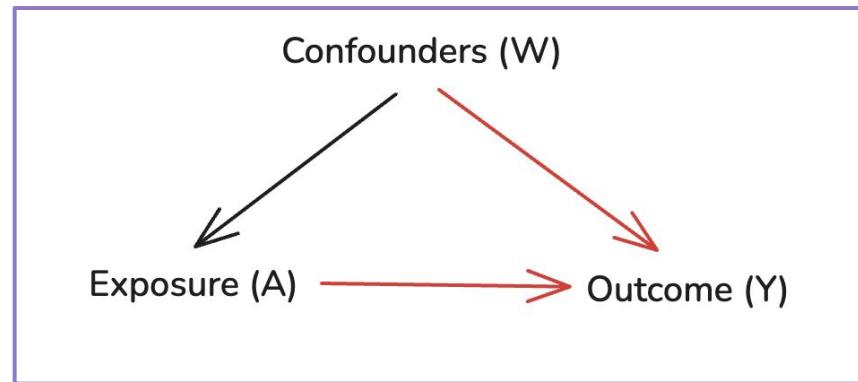
Coefficients:
SL.glm_All 0
SL.glm.interaction_All 0
SL.step_All 0
SL.step.interaction_All 0
tmle.SL.dbarts.k.5_All 0
SL.glm.interaction_All 0
SL.gam_All 0
SL.randomForest_All 0.6745099
SL.rpart_All 0
SL.xgboost_All 0.3254901

Conclusion - Comparison of all estimates

Method	Estimated ATE	95% Confidence Interval	Standard Error
Outcome Regression only	-0.115	(-0.233, 0.061)	
TMLE with GLMs only	-0.087	(-0.239, 0.066)	0.078
TMLE with superlearner default methods	-0.237	(-0.345, -0.130)	0.054
TMLE with additional flexible methods	-0.265	(-0.355 -0.175)	0.046

Conclusion - Takeaways (1)

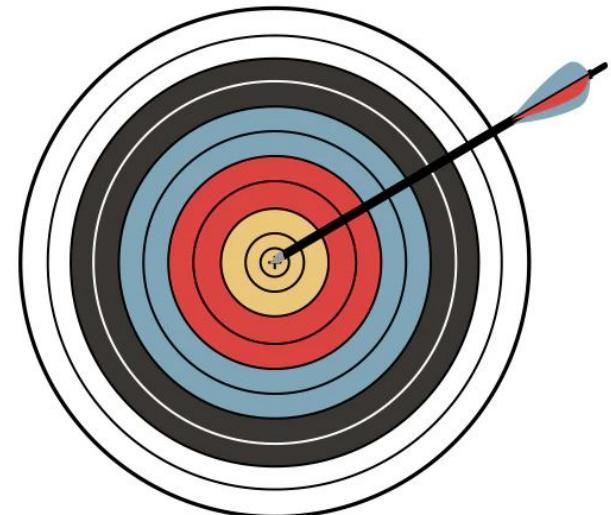
- TMLE helps us **adjust for confounding** by both:
 - Estimating the **outcome mechanism Q**
 - And the **treatment mechanism g**
- Then uses information from g to “**target**” the estimate from Q such that it is **unbiased for our estimand of interest** (in our case ATE)
- It is a **doubly robust** method for estimating various estimands of interest (in our case ATE)
 - Doubly robust = correct model specification assumption is “softened” as 2 models are used and only one needs to be correctly specified



Conclusion - Takeaways (2)

- TMLE can be applied using *tmle* package from CRAN using just 1 convenient function `tmle()`
 - The resulting object contains many useful estimates that are directly accessible
- Superlearner allows the model to create suitable ensemble learners from a wide array of statistical models via cross-validation

What do you think? Would you use TMLE? Why (not)?



**Thank you for your
attention! Questions?
Comments?**