R Lab 5 - TMLE

Introduction to Causal Inference

Goals:
1. Review the causal roadmap.
2. Code TMLE for the G-computation estimand.
3. Understand the basics of the `tmle` package.
4. Use the `tmle` package to explore the double robustness of TMLE.

Next lab:
We will implement the non-parametric bootstrap to estimate the standard error of the estimators. We will also use the sample variance of the estimated influence curve to obtain inference for TMLE.

1 Background

Dr. Alan Grant: "T-Rex doesn't want to be fed. He wants to hunt. Can't just suppress 65 million years of gut instinct." - Michael Crichton

We are interested in estimating the causal effect of prior experience with Dinosaurs on survival on Isla Nublar, the location of the InGen lab. Suppose we have data on the following variables:

- W1: gender (1 for male; 0 for female)
- W2: intelligence (1 for smart; 0 for not)
- W3: handy/inventiveness (scale from 0 for none to 1 for MacGyver)
- W4: running speed (scale from 0 for slow to 3 for fast)
- A: prior Dinosaur experience (1 for yes; 0 for no)
- Y: survival (1 for yes; 0 for no)

Let $W = (W1, W2, W3, W4)$ be the vector of baseline covariates.

http://www.thesambarnes.com/web-project-management/account-management-for-the-web-project-manager-part-1/

2 Causal Road Map Rundown

1. **Specify the Question:**
   What is the causal effect of prior experience on survival in Jurassic Park?

2. **Specify the causal model:**
   - Endogenous nodes: $X = (W, A, Y)$, where $W = (W1, W2, W3, W4)$ is the set of baseline covariates (gender, intelligence, MacGyver-ness, running speed), A is prior Dinosaur experience and Y is survival. For simplicity, we have condensed the baseline characteristics into a single node.
   - Exogenous nodes: $U = (U_W, U_A, U_Y) \sim P_U$. We place no assumptions on the distribution P_U.
   - Structural equations F:

$$\begin{aligned} W &= f_W(U_W) \\ A &= f_A(W, U_A) \\ Y &= f_Y(W, A, U_Y) \end{aligned}$$

   We have made exclusion restrictions, but not placed any restrictions on the functional forms.

3. **Specify the causal parameter of interest:**
   We are interested in the causal effect of prior Dinosaur experience on survival on Isla Nublar (i.e. the causal risk difference or the average treatment effect):

$$\Psi^F(P_{U,X}) = P_{U,X}(Y_1 = 1) - P_{U,X}(Y_0 = 1) = E_{U,X}(Y_1) - E_{U,X}(Y_0)$$

   where Y_a is the counterfactual outcome (survival), if possibly contrary to fact, the subject had experience $A = a$.

4. **Specify the link between the SCM and the observed data:**
   We assume that the observed data $O = (W, A, Y) \sim P_0$ were generated by sampling n times from a data generating described by the SCM. The statistical model $\mathcal{M}$ for the set of allowed distributions of the observed data is non-parametric.

5. **Assess identifiability:**
   In the original SCM $\mathcal{M}^F$, the target causal parameter is not identified from the observed data distribution. We need make assumptions about the independence of exogenous errors: $U_A \perp\!\!\!\perp U_Y$ and (i) $U_A \perp\!\!\!\perp U_W$, or (ii) $U_Y \perp\!\!\!\perp U_W$. Then the backdoor criteria will hold conditionally on $W = (W1, W2, W3, W4)$. We use M^{F*} to denote the original SCM augmented by the assumptions needed for identifiability.

   To identify $E_{U,X}(Y_a)$ with the G-Computation formula, we also need the positivity assumption to hold

$$min_{a \in \mathcal{A}}\ P_0(A = a | W = w) > 0$$

   for all w for which $P_0(W = w) > 0$. In terms of our example, there must be a positive probability of being experienced and not experienced within strata of baseline covariates.

6. **Specify the statistical estimand:**
   The target parameter of the observed data distribution (which equals the causal parameter in the augmented causal model) is given by the G-Computation formula:

$$\Psi(P_0) = E_0\big[E_0(Y|A = 1, W = w) - E_0(Y|A = 0, W = w)\big]$$

   This is our statistical estimand.

7. **Estimate the chosen parameter of the observed data distribution:**

   (a) **Simple substitution estimator based on the G-Computation formula:**

$$\hat{\Psi}_{SS}(P_n) = \frac{1}{n} \sum_{i=1}^{n} \big(\bar{Q}_n(1, W_i) - \bar{Q}_n(0, W_i)\big)$$

where P_n is the empirical distribution and $\bar{Q}_n(A, W)$ is the estimate of the conditional mean outcome given the exposure (experience with Dinosaurs) and baseline covariates.
- Consistency of the simple (non-targeted) substitution estimator depends on consistent estimation of $\bar{Q}_0(A, W) = E_0(Y|A, W)$.

(b) **Standard (unstabilized) inverse probability weighted estimator (IPTW)**:

$$\hat{\Psi}_{IPTW}(P_n) = \frac{1}{n}\sum_{i=1}^{n}\frac{\mathbb{I}(A_i = 1)}{g_n(A_i|W_i)}Y_i - \frac{1}{n}\sum_{i=1}^{n}\frac{\mathbb{I}(A_i = 0)}{g_n(A_i|W_i)}Y_i$$

where $g_n(A_i|W_i) = P_n(A_i|W_i)$ is an estimate of the treatment mechanism (i.e. the conditional probability of having Dinosaur experience, given the baseline covariates).
- Consistency of IPTW estimators depends on consistent estimation of $g_0(A|W)$.

(c) **Targeted maximum likelihood estimation (TMLE):**

$$\hat{\Psi}_{TMLE}(P_n) = \frac{1}{n}\sum_{i=1}^{n}\left(\bar{Q}_n^*(1, W_i) - \bar{Q}_n^*(0, W_i)\right)$$

where $\bar{Q}_n^*(A, W)$ denotes the updated estimate of the conditional mean outcome given the exposure and baseline covariates.
- Implementation requires estimation of both the conditional mean function $\bar{Q}_0(A, W)$ and the treatment mechanism $g_0(A|W)$.
- Double robust estimators are consistent if *either* $\bar{Q}_0(A, W)$ *or* $g_0(A|W)$ are estimated consistently.
- If both $\bar{Q}_0(A, W)$ and $g_0(A|W)$ are estimated consistently, TMLE will be efficient and achieve the lowest possible asymptotic variance over a large class of estimators.
- These asymptotic properties describe what happens when sample size goes to infinity and also translate into lower bias and variance in finite samples.

If we apply an estimator to our observed data (n i.i.d. copies of O drawn from P_0), we get an estimate (a number). The estimator is function of a random variable; so it is a random variable. It has a distribution, which we can study theoretically or using simulations.

Note: An estimator is *consistent* if the point estimates converge (in probability) to the estimand as sample size $n \to \infty$.

8. **Inference and interpret results:**
In the next lab, we will implement the non-parametric bootstrap for variance estimation for the three types of estimators. We will use the sample variance of the estimated influence curve to obtain inference for the TMLE.

3 Import and explore data set `RLab5.TMLE.csv`.

1. Use the `read.csv` function to import the data set and assign it to data frame `ObsData`.
2. Use the `head` and `summary` functions to explore the data.
3. Use the `nrow` function to count the number of subjects in the data set. Assign this number as `n`.

Solution:

```
> #  Import the data set and assign it to object ObsData; explore
> ObsData<- read.csv("RLab5.TMLE.csv")
> names(ObsData)


[1] "W1" "W2" "W3" "W4" "A"  "Y"


> head(ObsData)


  W1 W2         W3 W4 A Y
1  1  1 0.78873747  3 0 1
2  1  1 0.97580028  2 0 1
3  0  1 0.95099488  1 0 1
4  1  1 0.60374230  3 1 1
5  1  1 0.03030798  2 0 1
6  0  1 0.79658317  1 0 0


> summary(ObsData)


       W1               W2               W3                  W4
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0006814   Min.   :0.000
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.2474774   1st Qu.:1.000
 Median :1.0000   Median :1.0000   Median :0.5021037   Median :2.000
 Mean   :0.5158   Mean   :0.6524   Mean   :0.4981556   Mean   :1.735
 3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.7432143   3rd Qu.:2.000
 Max.   :1.0000   Max.   :1.0000   Max.   :0.9997801   Max.   :3.000
       A                Y
 Min.   :0.000   Min.   :0.0000
 1st Qu.:0.000   1st Qu.:0.0000
 Median :0.000   Median :1.0000
 Mean   :0.372   Mean   :0.6068
 3rd Qu.:1.000   3rd Qu.:1.0000
 Max.   :1.000   Max.   :1.0000


> # can get the dimensions
> dim(ObsData)


[1] 5000    6


> n<- nrow(ObsData)
```

4 Implement TMLE for the G-computation estimand

1. **Use SuperLearner to estimate $E_0(Y|A,W) = \bar{Q}_0(A,W)$, which is the conditional probability of surviving given the exposure (prior experience) and baseline covariates.**

   (a) Use the `library` function to load the `SuperLearner` package and then specify the SuperLearner library with the following algorithms: `SL.glm`, `SL.step` and `SL.glm.interaction`.

   ```
   > library("SuperLearner")
   ```

```
> # specify the library
> SL.library<- c("SL.glm", "SL.step", "SL.glm.interaction")
```

(b) Create data frame `X` consisting of the covariates $(W1, W2, W3, W4)$ and the intervention A.

i. Also create data frame `X1` where A has been set to 1.

ii. Also create data frame `X0` where A has been set to 0.

iii. Finally, create data frame `newdata` by stacking the data frames `X`, `X1`, `X0`:

```
> newdata<- rbind(X,X1,X0)
```

We will use `newdata` to obtain the expected outcome under the observed exposure $\bar{Q}_n(A, W)$, under the intervention $\bar{Q}_n(A = 1, W)$ and under the control $\bar{Q}_n(A = 0, W)$.

(c) Estimate $\bar{Q}_0(A, W)$ by running `SuperLearner`. Call this object `Qinit`. Be sure to specify the `SL.library` and the appropriate `family`.

```
> Qinit<- SuperLearner(Y=ObsData$Y, X=X, newX=newdata, SL.library=SL.library,
+     family="binomial")
```

Including `newX=newdata` allows us to get the predicted outcomes for all subjects under their observed exposure (i.e. using (A, W) in `X`), under the treatment (i.e. using $(A = 1, W)$ in `X1`), and under the control (i.e. $(A = 0, W)$ in `X0`)

(d) The predicted probabilities of surviving are accessed with `Qinit$SL.predict`. This is a vector of length $3n$.

i. Assign the predicted probability of surviving, given the subject's observed experience A and baseline characteristics W to `QbarAW`:

```
> QbarAW <- Qinit$SL.predict[1:n]
```

ii. Assign the predicted probability of survival for each subject given $A = 1$ and W to `Qbar1W`:

```
> Qbar1W <- Qinit$SL.predict[(n+1):(2*n)]
```

iii. Assign the predicted probability of survival for each subject given $A = 0$ and W to `Qbar0W`:

```
> Qbar0W <- Qinit$SL.predict[(2*n+1):(3*n)]
```

(e) Evaluate the simple substitution estimator by plugging the estimates $\bar{Q}_n(1, W)$ and $\bar{Q}_n(0, W)$ into the target parameter mapping:

$$\hat{\Psi}_{SS}(P_n) = \frac{1}{n} \sum_{i=1}^{n} \bar{Q}_n(1, W_i) - \bar{Q}_n(0, W_i)$$

Note: This step is not part of the TMLE algorithm, but done for comparison.

2. **Estimate the treatment mechanism $g_0(A|W) = P_0(A|W)$, which is the conditional probability of having Dinosaur experience, given baseline covariates.** You are provided with background knowledge that the true conditional probability might depend on the following variables:

$$\{A,\ W1,\ W2,\ W3,\ W4,\ cos(\pi W2),\ sin(\pi W3),\ W4^2\}$$

(a) Create transformed variables `cosW2`, `sinW3` and `W4sq`.

```
> cosW2<- cos(pi*ObsData$W2)
```

(b) Create data frame `W` of baseline covariates and the transformed variables.

(c) Estimate $g_0(A|W)$ by running `SuperLearner`. Call this object `gHatSL`. Since we are estimating the treatment mechanism, specify `Y=ObsData$A` and the predictors as `X=W`. Use the same library.

```
> gHatSL<- SuperLearner(Y=ObsData$A, X=W, SL.library=SL.library, family="binomial")
```

(d) The predicted probability of being experienced, given the subject's baseline characteristics $g_n(A = 1|W)$, can be accessed with `gHatSL$SL.predict`

i. Assign the predicted probability of being experienced $g_n(A = 1|W)$ to `gHat1W`:

```
> gHat1W<- gHatSL$SL.predict
```

ii. Assign the predicted probability of not being experienced $g_n(A=0|W)$ to `gHat0W`.

iii. Look at the distribution of propensity scores $g_n(A=1|W)$ and $g_n(A=0|W)$.

iv. Generate the predicted probability of the observed exposure, given the subject's baseline characteristics $g_n(a|w)$.
- *Hint:* Create empty vector `gHatAW`. Among subjects with $A=1$, assign the predicted probabilities $g_n(A=1|W)$. Among subjects with $A=0$, assign the predicted probabilities $g_n(A=0|W)$.

(e) Evaluate the IPTW estimator by taking the empirical mean of the weighted observations:

$$\hat{\Psi}_{IPTW}(P_n) = \frac{1}{n}\sum_{i=1}^{n}\frac{\mathbb{I}(A_i=1)}{g_n(A_i|W_i)}Y_i - \frac{1}{n}\sum_{i=1}^{n}\frac{\mathbb{I}(A_i=0)}{g_n(A_i|W_i)}Y_i$$

- The first term is the empirical mean of the outcomes, where observations with $A_i=1$ (`as.numeric(ObsData$A==1)`) are weighted as the inverse of the predicted probability of the observed exposure, given the baseline covariates $1/g_n(A_i|W_i)$ and where observations with $A_i \neq 1$ are weighted 0.
- The second term is the empirical mean of the outcomes, where observations with $A_i=0$ (`as.numeric(ObsData$A==0)`) are weighted as the inverse of the predicted probability of the observed exposure, given the baseline covariates $1/g_n(A_i|W_i)$ and where observations with $A_i \neq 0$ are weighted 0.
- As before, this is not part of the TMLE algorithm, but implemented for comparison.

3. **Use these estimates to create the clever covariate:**

$$H_n(A,W) = \left(\frac{\mathbb{I}(A=1)}{g_n(A=1|W)} - \frac{\mathbb{I}(A=0)}{g_n(A=0|W)}\right)$$

(a) Calculate `H.AW` for each subject:

```
> H.AW<- as.numeric(ObsData$A==1)/gAW - as.numeric(ObsData$A==0)/gAW
```

For subjects with $A=1$, the clever covariate is 1 over the predicted probability of being experienced, given the baseline covariates. Among subjects with $A=0$, the clever covariate is -1 over the predicted probability of not being experienced, given the baseline covariates $g_n(A=0|W)$.

(b) Also evaluate the clever covariate at $A=1$ and $A=0$ for all subjects. Call the resulting values `H.1W` and `H.0W`, respectively.

4. **Update the initial estimates.**

(a) Run a logistic regression of the outcome Y on the clever covariate $H_n(A,W)$, using the logistic of the initial estimate as offset and suppressing the intercept.

```
> logitUpdate<- glm(ObsData$Y ~ -1 +offset(qlogis(QbarAW)) + H.AW, family='binomial')
```

- We suppress the intercept by including -1 on the right hand side.
- In R, logistic function is given by `qlogis(x)`.
- As always, including `family='binomial'` runs logistic regression.

(b) Let `eps` denote the resulting maximum likelihood estimate of the coefficient on the clever covariate `H.AW`.

```
> eps<- logitUpdate$coef
```

(c) Update the initial estimate of $\bar{Q}_n(A,W)$ according to the fluctuation model:

$$logit\left[\bar{Q}_n^*(A,W)\right] = logit\left[\bar{Q}_n(A,W)\right] + \epsilon_n H_n(A,W)$$
$$\bar{Q}_n^*(A,W) = expit\left[logit\left[\bar{Q}_n^0(A,W)\right] + \epsilon_n H_n(A,W)\right]$$

Create `QbarAW.star` by taking the inverse logit (i.e. the expit) of the offset (`logit(QbarAW)`) plus the coefficient ϵ_n times the clever covariate $H_n(A,W)$:

```
> QbarAW.star<- plogis(qlogis(QbarAW)+ eps*H.AW)
```

(d) Update the initial estimates of $\bar{Q}_n(1, W)$ and $\bar{Q}_n(0, W)$:

$$logit\big[\bar{Q}_n^*(1, W_i)\big] = logit\big[\bar{Q}_n(1, W_i)\big] + \epsilon_n H_n(1, W_i)$$
$$logit\big[\bar{Q}_n^*(0, W_i)\big] = logit\big[\bar{Q}_n(0, W_i)\big] + \epsilon_n H_n(0, W_i)$$

(e) *Optional*: Try updating again. What is updated ϵ_n?

5. **Substitute the updated fits into the target parameter mapping:**

$$\hat{\Psi}_{TMLE}(P_n) = \frac{1}{n}\sum_{i=1}^{n}\left[\bar{Q}_n^*(1, W_i) - \bar{Q}_n^*(0, W_i)\right]$$

Solution:

```
> #-----------------
> # 1. Estimate Q_0(A,W) with SuperLearner
> #---------------------
> library("SuperLearner")
> # specify the library
> SL.library<- c("SL.glm", "SL.step", "SL.glm.interaction")


> # data frame X with baseline covariates and exposure
> X<-subset(ObsData, select=c(A, W1, W2, W3, W4))
> # create data frames with A=1 and A=0
> X1 <- X0<-X
> X1$A<-1          # under txt
> X0$A<- 0         # under control
> # create newdata by stacking
> newdata<- rbind(X,X1,X0)
> dim(newdata)


[1] 15000     5


> # call SuperLearner
> Qinit<- SuperLearner(Y=ObsData$Y, X=X, newX=newdata, SL.library=SL.library,
+   family="binomial")
> Qinit


Call:
SuperLearner(Y = ObsData$Y, X = X, newX = newdata, family = "binomial", SL.library = SL.library)



                           Risk      Coef
SL.glm_All             0.1761764 0.0000000
SL.step_All            0.1760120 0.1725645
SL.glm.interaction_All 0.1752254 0.8274355


> #get the predicted prob
> length(Qinit$SL.predict)
```

```
[1] 15000


> # pred prob of survival given A,W
> QbarAW <- Qinit$SL.predict[1:n]
> # predicted probability of survival for each subject given A=1 and W
> Qbar1W <- Qinit$SL.predict[(n+1): (2*n)]
> # predicted probability of survival for each subject given A=0 and W
> Qbar0W <- Qinit$SL.predict[(2*n+1): (3*n)]


> # the fitted value at the observed exposure should equal the fitted value
> # under when A=a
> tail(cbind(ObsData$A, QbarAW, Qbar1W, Qbar0W))


             QbarAW    Qbar1W     Qbar0W
[4995,] 0 0.8075776 0.9373462 0.80757756
[4996,] 1 0.2989520 0.2989520 0.09783428
[4997,] 0 0.8259809 0.9268409 0.82598095
[4998,] 0 0.8032400 0.9528324 0.80324004
[4999,] 0 0.6969058 0.8867991 0.69690581
[5000,] 1 0.2703483 0.2703483 0.11793546


> # note the simple substitution estimator would be
> PsiHat.SS<-mean(Qbar1W - Qbar0W)
> PsiHat.SS

[1] 0.1753176


> ##########
> # 2.  Estimate g_0(A|W) with SuperLearner
> #############
> # creating transformed variables
> cosW2<- cos(pi*ObsData$W2)
> sinW3<- sin(pi*ObsData$W3)
> W4sq<- ObsData$W4*ObsData$W4
> # add to data set
> ObsData<- data.frame(ObsData, cosW2, sinW3, W4sq)
> names(ObsData)


[1] "W1"    "W2"    "W3"    "W4"    "A"     "Y"     "cosW2" "sinW3" "W4sq"


> # creating data frame with only baseline cov and the transformed variables
> W<- subset(ObsData, select= -c(A,Y))


> # call SuperLearner
> gHatSL<- SuperLearner(Y=ObsData$A, X=W, SL.library=SL.library, family="binomial")
> gHatSL


Call:
SuperLearner(Y = ObsData$A, X = W, family = "binomial", SL.library = SL.library)
```

```
                            Risk      Coef
SL.glm_All              0.1209160 0.0000000
SL.step_All             0.1207556 0.5616181
SL.glm.interaction_All 0.1208965 0.4383819

> # generate the predicted prob of being experienced, given baseline cov
> gHat1W<- gHatSL$SL.predict
> # generate the predicted prob of not being experienced, given baseline cov
> gHat0W<- 1- gHat1W

> # summary of propensity scores
> summary(gHat1W)

       V1
 Min.   :0.01209
 1st Qu.:0.09484
 Median :0.20138
 Mean   :0.37200
 3rd Qu.:0.76625
 Max.   :0.97693

> summary(gHat0W)

       V1
 Min.   :0.02307
 1st Qu.:0.23375
 Median :0.79862
 Mean   :0.62800
 3rd Qu.:0.90516
 Max.   :0.98791

> # generate the predicted prob of the obs experience, given baseline cov
> gHatAW<- rep(NA, n)
> gHatAW[ObsData$A==1]<- gHat1W[ObsData$A==1]
> gHatAW[ObsData$A==0]<- gHat0W[ObsData$A==0]
> # check that the pred prob of the obs exposure equals the pred prob
> # when A=a
> tail(cbind(ObsData$A, gHatAW, gHat1W, gHat0W))

          gHatAW
4995 0 0.7558620 0.2441380 0.7558620
4996 1 0.8196462 0.8196462 0.1803538
4997 0 0.9738081 0.0261919 0.9738081
4998 0 0.9540158 0.0459842 0.9540158
4999 0 0.8357616 0.1642384 0.8357616
5000 1 0.9100534 0.9100534 0.0899466

> #IPTW estimator of the ATE is
> PsiHat.IPTW<- mean(as.numeric(ObsData$A==1)*ObsData$Y/gHatAW) -
```

```
+   mean(as.numeric(ObsData$A==0)*ObsData$Y/gHatAW)
> PsiHat.IPTW


[1] 0.1163912


> #------------
> # 3. Create the clever covariate H_n^*(A,W)$ for each subject
> # numerator is the indicator of the obsv txt.
> # denominator is the predicted probability of observed exp, given baseline cov
> #--------------
> H.AW<- as.numeric(ObsData$A==1)/gHat1W - as.numeric(ObsData$A==0)/gHat0W
> # equiv: H.AW<- (2*ObsData$A-1)/ gHatAW
>
> # also want to evaluate the clever covariates at A=1 and A=0 for all subjects
> H.1W<- 1/gHat1W
> H.0W<- -1/gHat0W


> tail(cbind(ObsData$A, gHatAW, gHat1W, gHat0W, H.AW, H.1W, H.0W))


          gHatAW
4995 0 0.7558620 0.2441380 0.7558620 -1.322993  4.096044  -1.322993
4996 1 0.8196462 0.8196462 0.1803538  1.220039  1.220039  -5.544658
4997 0 0.9738081 0.0261919 0.9738081 -1.026896 38.179746  -1.026896
4998 0 0.9540158 0.0459842 0.9540158 -1.048201 21.746601  -1.048201
4999 0 0.8357616 0.1642384 0.8357616 -1.196513  6.088711  -1.196513
5000 1 0.9100534 0.9100534 0.0899466  1.098837  1.098837 -11.117708


> #---------
> # 4. Update the initial estimate of Qbar_0(A,W)
> # logit( Qbar_n^0(eps) )= logit( Qbar_n^0) + epsHAW
> # run logistic regression of Y on H.AW using the logit of initQbarAW.predict as offset
> #------------
> logitUpdate<- glm(ObsData$Y ~ -1 +offset(qlogis(QbarAW)) + H.AW, family='binomial')
> eps<- logitUpdate$coef
> eps


       H.AW
-0.03545905


> #  calc the predicted values for each subj under each txt
> QbarAW.star<- plogis(qlogis(QbarAW)+ eps*H.AW)
> Qbar1W.star<- plogis(qlogis(Qbar1W)+ eps*H.1W)
> Qbar0W.star<- plogis(qlogis(Qbar0W)+ eps*H.0W)


> # since the clever cov is not changing, updating will not have any effect
> coef(glm(ObsData$Y ~ -1 +offset(qlogis(QbarAW.star)) + H.AW, family=binomial))


        H.AW
8.720348e-17
```

```
> # 5. Estimate Psi(P_0) as the emp mean of the difference in the pred
> # outcomes under A=1 and A=0
> PsiHat.TMLE<- mean(Qbar1W.star) - mean(Qbar0W.star)


> # comparing the estimates...
> c(PsiHat.SS, PsiHat.IPTW, PsiHat.TMLE)


[1] 0.1753176 0.1163912 0.1263828
```

The point estimate from the simple substitution estimator, using SuperLearner for $\bar{Q}_0(A, W)$, was 17.5%. The point estimate from IPTW, using SuperLearner for $g_0(A|W)$, was 11.6%. The point estimate from TMLE was 12.6%. The true value of the statistical estimand was 13%. To evaluate the performance of these estimators (e.g. bias and variance), we would draw another independent sample of size n, implement the 3 estimators (with the same SuperLearner library), and repeat 500 or so times.

5 The basics of the `tmle` package

1. **If you have not already, download and install the `tmle` package.**
2. **Load the package with `library("tmle")`.**
3. **Read the help file: `?tmle`:**
   - The basic input to the function `tmle` is the outcome `Y`, the intervention `A` and the baseline covariates `W` (need to be in a matrix or data frame).
   - The user can also specify mediating variables with `Z`, missing values with `Delta` and repeated observations with `id`.
   - The initial estimates of $\bar{Q}_0(A, W)$ can be supplied by the user in a nx2 matrix `Q`, can be estimated according to a user-specified regression formula `Qform`, or estimated with SuperLearner with library `Q.SL.library`.
   - The initial estimates of $g_0(A = 1|W)$ can be supplied by the user in a nx1 vector `g1W`, can be estimated according to a user-specified regression formula `gform`, or estimated with SuperLearner with library `g.SL.library`.
4. **Call the `tmle` function using the SuperLearner initial estimates for the conditional mean outcome $\bar{Q}_0(A, W)$ and for the treatment mechanism $g_0(A|W)$. Also, specify the family as binomial.**

   ```
   > tmle(Y=ObsData$Y, A=ObsData$A, W=W, Q=cbind(Qbar0W, Qbar1W), g1W=gHat1W, family="binomial")
   ```

   Recall that the data frame `W` of includes baseline covariates and the transformed variables.
5. **Use the `summary` and `names` functions to explore the output.**

Solution:

```
> ###########
> # tmle package
> ##########
> library("tmle")
```

```
> ?tmle

> tmle.QGiven.gGiven<- tmle(Y=ObsData$Y, A=ObsData$A, W=W,
+     Q=cbind(QbarOW, Qbar1W),g1W=gHat1W, family="binomial")
> summary(tmle.QGiven.gGiven)

 Initial estimation of Q
        Procedure: user-supplied values
 Estimation of g (treatment mechanism)
        Procedure: user-supplied values

 Estimation of g.Z (intermediate variable assignment mechanism)
        Procedure: No intermediate variable

 Estimation of g.Delta (missingness mechanism)
        Procedure: No missingness

 Bounds on g: ( 0.025 0.975 )

 Additive Effect
   Parameter Estimate:  0.1276
   Estimated Variance:  0.00029445
              p-value:  1.0351e-13
    95% Conf Interval: (0.093971, 0.16124)

 Relative Risk
   Parameter Estimate:  1.2301
              p-value:  3.444e-14
    95% Conf Interval: (1.166, 1.2978)

              log(RR):  0.20709
    variance(log(RR)):  0.00074632

 Odds Ratio
   Parameter Estimate:  1.724
              p-value:  7.646e-13
    95% Conf Interval: (1.4854, 2.0008)

              log(OR):  0.54463
    variance(log(OR)):  0.0057741

> names(tmle.QGiven.gGiven)

[1] "estimates" "Qinit"     "g"         "g.Z"       "g.Delta"   "Qstar"
[7] "epsilon"

> tmle.QGiven.gGiven$epsilon

        H0W         H1W
 0.03749054 -0.03477500
```

The `tmle` package uses a two-dimensional clever covariate for updating.

6 Use the `tmle` package to explore performance under model misspecification

1. **Implement `tmle` with the correctly specified model for $\bar{Q}_0(A,W)$ and for $g_0(A|W)$. Specify the regression formulas in `Qform` and `gform`.** The true conditional probability of surviving, given the exposure and baseline covariates, is described by the following parametric model

$$logit[\bar{Q}_0(A,W)] = \beta_0 + \beta_1 A + \beta_2 cos(\pi W2) + \beta_3 sin(\pi W3) + \beta_4 W4^2$$

   The true conditional probability of the exposure (having prior Dinosaur experience), given the baseline covariates, is described by the following parametric model

$$logit[g_0(A=1|W)] = \beta_0 + \beta_1 cos(\pi W2) + \beta_2 sin(\pi W3) + \beta_4 W1^* W4$$

```
> tmle.Qcorr.gcorr<- tmle(Y=ObsData$Y, A=ObsData$A, W=W,
+   Qform=Y~ A+cosW2+sinW3+W4sq, gform=A~cosW2+sinW3+W1:W4, family="binomial")
```

   - Using `W1:W4` ensures that only the interaction between $W1$ and $W4$ is included in the model for $g_0(A|W)$.

2. **Implement `tmle` with a misspecified model for $\bar{Q}_0(A,W)$ and a correctly specified model for $g_0(A|W)$.**

3. **Implement `tmle` with a correctly specified model for $\bar{Q}_0(A,W)$ and a misspecified model for $g_0(A|W)$.**

4. **Implement `tmle` with a misspecified model for $\bar{Q}_0(A,W)$ and a misspecified model for $g_0(A|W)$.**

5. **Implement `tmle` using SuperLearner with the default library for initial estimates of $\bar{Q}_0(A,W)$ and $g_0(A|W)$.**

6. **Compare the resulting point estimates for $\Psi(P_0)$ for this sample of $n = 5,000$ subjects.**

Solution:

```
> # both correctly specified
> tmle.Qcorr.gcorr<- tmle(Y=ObsData$Y, A=ObsData$A, W=W,
+   Qform=Y~A+cosW2+sinW3+W4sq, gform=A~cosW2+sinW3+W1:W4, family="binomial")


> # misspecified Qbar
> tmle.Qmiss.gcorr<- tmle(Y=ObsData$Y, A=ObsData$A, W=W,
+   Qform=Y~A, gform=A~cosW2+sinW3+W1:W4, family="binomial")


> # misspecified g
> tmle.Qcorr.gmiss<- tmle(Y=ObsData$Y, A=ObsData$A, W=W,
+   Qform=Y~A+cosW2+sinW3+W4sq, gform=A~W1, family="binomial")


> # misspecified Q and g
> tmle.Qmiss.gmiss<- tmle(Y=ObsData$Y, A=ObsData$A, W=W,
+   Qform=Y~A, gform=A~W1, family="binomial")


> # estimated with SuperLearner
> tmle.Qsl.gsl<- tmle(Y=ObsData$Y, A=ObsData$A, W=W, family="binomial")
```

```
> #----------------
> # corresponding simple substitution estimates
> #----------------
> # running simple subs est: Q correct
> Qcorr<- glm(Y~A+cosW2+sinW3+W4sq, data=ObsData, family="binomial")
> SS.Qcorr<- mean(predict(Qcorr, newdata=X1, type="response") -
+   predict(Qcorr, newdata=X0, type="response"))
> # running simple subs est: Q miss
> Qmiss<- glm(Y~A, data=ObsData, family="binomial")
> SS.Qmiss<- mean(predict(Qmiss, newdata=X1, type="response") -
+   predict(Qmiss, newdata=X0, type="response"))
> # using the superlearner fits from tmle.Qsl.gsl$Qinit$Q
> SS.Qsl<- mean(tmle.Qsl.gsl$Qinit$Q[,"Q1W"] - tmle.Qsl.gsl$Qinit$Q[,"Q0W"])


> #----------------
> # corresponding the IPTW estimates
> #-----------------
> # iptw with correctly spec g
> gcorr<- glm(A~cosW2+sinW3+W1:W4, data=ObsData, family="binomial")
> gcorr.1W<- predict(gcorr, type="response")
> IPTW.gcorr<- mean(ObsData$A*ObsData$Y/gcorr.1W -
+   (1-ObsData$A)*ObsData$Y/(1-gcorr.1W) )
> # iptw with mispec g
> gmiss<- glm(A~W1, data=ObsData, family="binomial")
> gmiss.1W<- predict(gmiss, type="response")
> IPTW.gmiss<- mean(ObsData$A*ObsData$Y/gmiss.1W -
+   (1-ObsData$A)*ObsData$Y/(1-gmiss.1W) )
> # iptw using the superlearner fit tmle.Qsl.gsl$g$g1W
> gsl.1W<- tmle.Qsl.gsl$g$g1W
> IPTW.gsl<- mean(ObsData$A*ObsData$Y/gsl.1W -
+   (1-ObsData$A)*ObsData$Y/(1-gsl.1W))


> #---------
> # Compare the point estimates.
> #----------
> est<- data.frame(rbind(c(SS.Qcorr, SS.Qmiss, SS.Qcorr, SS.Qmiss, SS.Qsl),
+   c(IPTW.gcorr, IPTW.gcorr, IPTW.gmiss, IPTW.gmiss, IPTW.gsl),
+   c(tmle.Qcorr.gcorr$estimates$ATE$psi,
+   tmle.Qmiss.gcorr$estimates$ATE$psi,tmle.Qcorr.gmiss$estimates$ATE$psi,
+   tmle.Qmiss.gmiss$estimates$ATE$psi, tmle.Qsl.gsl$estimates$ATE$psi)))
> colnames(est)<-c("Q-corr.g-corr", "Q-miss.g-corr", "Q-corr.g-miss",
+  "Q-miss.g-miss", "Q-SL.g-SL")
> rownames(est)<-c("Simp Subs", "IPTW", "TMLE")
> round(est, 4)


          Q-corr.g-corr Q-miss.g-corr Q-corr.g-miss Q-miss.g-miss Q-SL.g-SL
Simp Subs        0.1274       -0.2112        0.1274       -0.2112    0.1274
IPTW             0.1143        0.1143       -0.2148       -0.2148    0.1180
TMLE             0.1284        0.1330        0.1275       -0.2148    0.1265
```

Consistency of the simple substitution estimator depends on consistent estimation of $\bar{Q}_0(A, W)$. Consistency of IPTW estimators depends on consistent estimation of $g_0(A|W)$. TMLE is double robust! Even when the

conditional mean function $\bar{Q}_0(A,W) = E_0(Y|A,W)$ is misspecified (by ignoring covariates) or $g_0(A|W)$ is misspecified (by only using $W1$), we obtain a consistent estimate of $\Psi(P_0)$. If both $\bar{Q}_0(A,W)$ and $g_0(A,W)$ are consistently estimated, then TMLE will achieve lowest asymptotic possible variance over a large class of estimators.

Formally, an estimator is *consistent* if the point estimates converge (in probability) to the estimand as sample size $n \to \infty$. This is an asymptotic property. Here, we only have one sample of size $n = 5,000$. To evaluate the consistency of TMLE, we would need to do multiple runs at increasing samples sizes, e.g. $n = 500$, $n = 5,000$, $n = 50,000$, $n = 500,000$.

Note: There is a new TMLE package for point treatment as well as longitudinal problems: `ltmle`.

Solution:

Appendix: A specific data generating process

The following code was used to generate the data set `RLab5.TMLE.csv`. In this data generating process (one of many compatible with the SCM $\mathcal{M}^F$), *all exogenous errors are independent.*

```
> #-------
> # generateData - function to generate the data
> # input: number of draws
> # output: ObsData + counterfactuals
> #--------
> generateData<- function(n){
+   W1 <- rbinom(n, size=1, prob=0.5) #male
+   W2 <- rbinom(n, size=1, prob=0.65) #smart
+   W3 <- runif(n, min=0, max=1) # MacGyver
+   W4 <- rbinom(n, size=3, prob=plogis(0.5-W2 + W3)) #running speed
+   cosW2 <- cos(pi*W2)
+   sinW3 <- sin(pi*W3)
+   W4sq <- W4*W4
+   A<- rbinom(n, size=1, prob= plogis(-1+2*cosW2+2*sinW3-0.5*W1*W4))
+   Y<- rbinom(n, size=1, prob= plogis(-3+A-2*cosW2+2.5*sinW3+.25*W4sq))
+
+   # counterfactual
+   Y.1<- rbinom(n, size=1, prob= plogis(-3+1-2*cosW2+2.5*sinW3+.25*W4sq))
+   Y.0<- rbinom(n, size=1, prob= plogis(-3+0-2*cosW2+2.5*sinW3+.25*W4sq))
+
+   # return data.frame
+   data.frame(W1,W2,W3,W4,cosW2, sinW3, W4sq, A,Y, Y.1, Y.0)
+ }


> #------------
> # Creation of RLab5.TMLE.csv
> #-----
> set.seed(252)
> ObsData<- generateData(n=5000)
```

```
> ObsData<- subset(ObsData, select=c(W1,W2,W3,W4,A,Y) )
> write.csv(ObsData, file="RLab5.TMLE.csv", row.names=F)
> #------------
```

We could obtain the true value of the causal parameter $\Psi^F(P_{U,X})$ by drawing a huge number of observations and taking the difference in the means of the counterfactual outcomes.

```
> set.seed(252)
> TrueData<- generateData(n=100000)
> # Simply take the difference in mean the counterfactuals
> Psi.F<- mean(TrueData$Y.1) - mean(TrueData$Y.0)
> Psi.F

[1] 0.13245
```

The causal risk difference $\Psi^F(P_{U,X})$ is 13.2%. The counterfactual probability of survival would be 13.2% higher if all subjects had prior Dinosaur experience than if none were experienced.

Since we know the true conditional mean $\bar{Q}_0(A, W)$, we could evaluate the statistical estimand $\Psi(P_0)$ by drawing a huge number of observations and taking the difference of the mean predicted outcomes under the intervention and the control.

```
> # Recall TrueData consists of 100,000 observations
> Qbar.true<- glm(Y~A+cosW2+sinW3+W4sq, family="binomial", data=TrueData)
> txt<- control <- TrueData
> txt$A=1; control$A=0
> Psi.P0<- mean( predict(Qbar.true, newdata=txt, type="response")) -
+   mean(predict(Qbar.true, newdata=control, type="response"))
> Psi.P0

[1] 0.1303093
```

Note: If we increased n enough, these values should be identical. The slight difference, here, is due to estimation.