R Lab 6 - Estimation, Part II: parametric g-computation, ICE representation of g-formula, and TMLE for intervention specific mean

Advanced Topics in Causal Inference

Assigned: November 2, 2020
Lab due: November 10, 2020 on bCourses. Please answer all questions and include relevant `R` code. You are encouraged to discuss the assignment in groups, but should not copy code or interpretations verbatim. Upload your own completed lab to bCourses.

Last lab:
Introduction to `ltmle` package.

Goals for this lab:
1. Implement the "traditional" longitudinal parametric g-computation estimator
2. Implement the g-computation estimator based on the ICE representation of the longitudinal g-computation formula.
3. Implement TMLE based on the ICE representation of the longitudinal g-computation formula (by hand).

Next lab:
TMLE for intervention specific mean, MSM, and dynamic regimes using the `ltmle` package.

---

1 Introduction and Motivation

Recall that in `R` Lab 3, we showed that under certain sequential randomization and positivity assumptions, we can write our target causal parameter, $\Psi(P_{U,X})$, as a parameter of our observed data distribution, $\Psi(P_0)$. Specifically:

$$\underbrace{E_{U,X}[Y_{\bar{a}}]}_{\Psi(P_{U,X})} \overbrace{=}^{\text{SRA and pos.}} \underbrace{\sum_{\bar{l}} \left[E_0[Y|\bar{A}=\bar{a},\bar{L}=\bar{l}] \times \prod_{t=1}^{K} P_0[L(t)|\bar{A}(t-1)=\bar{a}(t-1),\bar{L}(t-1)=\bar{l}(t-1)] \right]}_{\Psi(P_0)}$$

$\Psi(P_0)$ in this equation is called the longitudinal g-computation formula, and in this lab we will estimate this statistical parameter using the following three methods:

1. <u>**Parametric g-computation estimator**</u> - going forwards in time, we estimate the conditional distribution/density of non-intervention nodes, given the past, and plug those estimates into the parameter mapping $\Psi(P_0)$.

(a) Estimate $\bar{Q}(L(t)|\bar{L}(t-1), \bar{A}(t-1))$ the conditional distribution/density of each $L(t)$ given its past parents, for $t = 1, ..., K$

(b) Use these estimates of $\bar{Q}(L(t)|\bar{L}(t-1), \bar{A}(t-1))$ to "simulate counterfactual covariate histories" over time, setting $A(t) = a(t)$ for $t = 1, ..., K$

(c) Repeat this many times for each treatment of interest

(d) Summarize these regime-specific estimates according to the parameter of interest

2. **ICE** - this estimator is the g-computation formula represented as a series of iterated conditional expectations (ICEs). Essentially, we fit a series of regressions going backwards in time, where each regression uses the regression before it (evaluated at the treatment history of interest) as its outcome.

(a) At $t = K + 1$: estimate $\bar{Q}_{K+1}(\bar{a}(K), \bar{L}(K))$

- In other words, $E[Y|\bar{L}(K), \bar{A}(K) = \bar{a}(K)]$

(b) At $t = K$: estimate $\bar{Q}_K(\bar{a}(K-1), \bar{L}(K-1))$ by using the estimated $\bar{Q}_{K+1}(\bar{a}(K), \bar{L}(K))$ as an outcome and regressing it on $\bar{L}(K-1), \bar{A}(K-1) = \bar{a}(K-1)$

- In other words, take the expectation of $E[Y|\bar{L}(K), \bar{A}(K) = \bar{a}(K)]$ with respect to the distribution of $L(K)$ given $\bar{L}(K-1)$ and $\bar{A}(K-1) = \bar{a}(K-1)$
- Also denoted $E\big[E[Y|\bar{L}(K), \bar{A}(K) = \bar{a}(K)]|\bar{L}(K-1), \bar{A}(K-1) = \bar{a}(K-1)\big]$

(c) ...

(d) At $t = 2$: estimate $\bar{Q}_2(a(1), L(1))$ by using $\bar{Q}_3(\bar{a}(2), \bar{L}(2))$ as an outcome and regressing it on $L(1), A(1) = a(1)$

- In other words, take the expectation of $E\Big[\ldots\big[E[E[Y|\bar{L}(K), \bar{A}(K) = \bar{a}(K)]|\bar{L}(K-1), \bar{A}(K-1) = \bar{a}(K-1)]\big]\ldots\Big]$ with respect to the distribution of $L(2)$ given $L(1)$ and $A(1) = a(1)$

(e) Plug in the estimate of $\bar{Q}_2(a(1), L(1))$ into the parameter mapping $\Psi(P_0)$, i.e., take the empirical mean of the estimated $\bar{Q}_2(a(1), L(1))$

3. **TMLE** - similar to the ICE g-computation estimator, except that we update each regression at time t before using it as an outcome for the next regression corresponding to time $t - 1$.

(a) At $t = K + 1$

i. Estimate $\bar{Q}^0_{K+1}(\bar{a}(K), \bar{L}(K))$.

ii. Target initial estimate of $\bar{Q}^0_{K+1}(\bar{a}(K), \bar{L}(K))$ and update to $\bar{Q}^\star_{K+1}(\bar{a}(K), \bar{L}(K))$ by "cleverly" incorporating an estimate of the treatment mechanism at K, $\prod_{j=1}^{K} g(A(j)|\bar{A}(j-1), \bar{L}(j))$

(b) At $t = K$

i. Using $\bar{Q}^\star_{K+1}(\bar{a}(K), \bar{L}(K))$ as an outcome, estimate $Q^0_K(\bar{a}(K-1), \bar{L}(K-1))$.

ii. Target $\bar{Q}_n, K(\bar{a}(K-1), \bar{L}(K-1))$ and update to $\bar{Q}^\star_K(\bar{a}(K-1), \bar{L}(K-1))$ by "cleverly" incorporating an estimate of the treatment mechanism at $K-1$, $\prod_{j=1}^{K-1} g(A(j)|\bar{A}(j-1), \bar{L}(j))$

(c) ...

(d) At $t = 2$

i. Using $\bar{Q}^\star_3(\bar{a}(2), \bar{L}(2))$ as an outcome, estimate $\bar{Q}^0_2(a(1), L(1))$.

ii. Target $\bar{Q}^0_2(a(1), L(1))$ and update to $Q^\star_2(a(1), L(1))$ by "cleverly" incorporating the treatment mechanism at time 1, $g(A(1)|L(1))$

(e) Plug in the targeted estimate of $\bar{Q}^\star_2(a(1), L(1))$ into the parameter mapping $\Psi(P_0)$, i.e., take the empirical mean of $\bar{Q}^\star_2(a(1), L(1))$

Figure 1: G-computation.

1.1 This lab

Recall that your GSR gave you data on one-thousand students, which we can treat as 1,000 i.i.d. copies of O. In `R` Lab 4 we used these data to estimate the effects of sleep using the IPTW estimator, and measured the estimator's performance.

In this lab we are continuing estimation (step 6 of the roadmap) of the effects of sleep by implementing the 3 estimators above. You'll also evaluate the estimators' performance (refer to `R` Lab 4 for definitions). In this lab, again you will answer questions for two of the data generating systems we've been working with in previous labs.

1.2 To turn in:

For each of the 2 data structures listed below, answer the following questions:

1. **Implement: (1) traditional longitudinal parametric g-computation estimator, (2) g-computation estimator based on the ICE representation of the longitudinal computation formula, (3) TMLE based on ICE representation of longitudinal g-computation**.
2. **Compare performance metrics of each of the estimators**. Specifically, evaluate the bias, variance, and mean-squared error (MSE) of each of the estimators.

Data Structure 0: $O = (L(1), A(1), L(2), A(2), Y)$ We are interested in the expected Y if everyone got treatment regime $\bar{a}(2) = 1$.

$$\Psi^F(P_{U,X}) = E_{U,X}[Y_{\bar{a}(2)=1}] = P_{U,X}(Y_{\bar{a}(2)=1} = 1) = 0.7922$$

Before we get started with estimation...

1. Load `DataStructure0.RData` using the `load()` function. Make sure you have specified the correct file path. You should see 4 new things come up in your global environment:

   - `ObsData0` - this is a dataframe of 1,000 observations that follows Data Structure 0 from the previous lab.
   - `Psi.F0` - this is the true $\Psi^F(P_{U,X})$ value for the target causal parameter $E_{U,X}[Y_{\bar{a}(2)=1}]$ (generated in lab 3).
   - `generate_data0` - this is the function that generates n copies of Data Structure 0.
   - `generate_data0_intervene` - this is the function that generates n intervened-on copies of Data Structure 0. We won't be using this function in this lab, so you can remove it using the `rm()` function.

     ```
     > rm(generate_data0_intervene)
     ```

2. Assign the number of students to `n`.

3. Create a new function, `bound()`, that takes as input a number `x` and bounds it to be between 0.1 and 1. That is: `x` is greater than 1, return 1; when `x` is smaller than 0.01, return .01.

   ```
   > bound = function(x){
   +   x[x < 0.01] = 0.01
   +   x[x > 1] = 0.99
   +   return(x)
   + }
   ```

   This function will be useful to prevent extreme weights on our clever covariate in the TMLE sections.

4. Also make sure you have the `bound()` function (defined in the previous data structure) ready and living in your global environment.

Parametric g-computation - Data Structure 0

1. Create a new function `param.gcomp0_fun()` that takes in `abar` as an argument. Within the function:

   (a) First, estimate the conditional distributions of all the non-intervention nodes, in this case, $L(1)$, $L(2)$, and Y. For this example, we need estimates of:

      i. $\bar{Q}(Y|\bar{L}(2), \bar{A}(2))$, or the conditional probability of Y, given the past. For example:

      ```
      > Q.Y.reg = glm(Y ~ L1 + A1 + L2 + A2, data = ObsData0, family = "binomial")
      ```

      ii. $\bar{Q}(L(2)|L(1), A(1))$, or the conditional probability of $L(2)$, given the past.

      iii. $\bar{Q}(L(1))$, the baseline covariate distribution. To estimate the distribution of $L(1)$, we can use the empirical distribution. *Hint:* go to the next step!

   (b) Use these estimates to generate (using Monte Carlo simulation) many "counterfactual" covariate and outcome histories over time, setting $A(t) = a(t)$ for $t = 1, 2$.

      i. Set `S` equal to $10,000$, the number of times we will simulate.

      ii. Sample `S` observations/rows, with replacement, from `ObsData0`.

      ```
      > ObsData0.MC = ObsData0[sample(1:n, S, replace = T),]
      ```

      iii. Draw $L_i(1) = l_i(1)$ for each individual. That is, set `l1` equal to `L1` that lives within the simulated data:

```
> l1 = ObsData0.MC$L1
```

iv. Draw $L_i(2) = l_i(2)$ for each individual.

A. From the estimated conditional distribution of $Q_n(L(2)|L(1), A(1))$, predict the probability of $L(2) = 1$, setting baseline exposure and covariate to $a(1)$ and $l(1)$, respectively. That is:

```
> Q.L2 = predict(Q.L2.reg,
+                type = "response",
+                newdata = data.frame(L1 = l1,
+                                     A1 = abar[1]))
```

B. Because $L(2)$ is binary, for each individual, draw an observation from a $Bernoulli(p_i = \bar{Q}(L_i(2)|L_i(1) = l_i(1), A_i(1) = a_i(1)))$ distribution.

```
> l2 = rbinom(S, size = 1, prob = Q.L2)
```

v. Draw $Y_i = y_i$ for each individual.

A. From the estimated conditional distribution of $Q_n(Y|\bar{L}(2), \bar{A}(2))$, predict the probability of $Y = 1$, setting time varying exposures and covariates to $\bar{a}(2)$ and $\bar{l}(2)$, respectively.

B. Because Y is binary, for each individual, draw an observation from a $Bernoulli(p_i = Q_n(Y_i|L_i(1) = l_i(1), A_i(1) = a_i(1), L_i(2) = l_i(2), A_i(2) = a_i(2)))$ distribution.

(c) Estimate the probability of $Y = 1$ with the empirical proportion of `y`. Have the function `param.gcomp0_fun()` return this value.

2. Estimate $E_{U,X}[Y_{\bar{a}=1}] = P_{U,X}(Y_{\bar{a}=1} = 1)$ using parametric g-computation by applying `param.gcomp0_fun()`, with the correct `abar` argument specification.

ICE g-computation - Data Structure 0

1. Create a new function `ICE.gcomp0_fun()` that takes in `abar` as an argument. Within the function:

(a) Create a dataframe called `newdata` where $A(1)$ and $A(2)$ have been set to $a(1)$ and $a(2)$, respectively:

```
> newdata = ObsData0
> newdata$A1 = abar[1]
> newdata$A2 = abar[2]
```

(b) For time $t = 3$:

i. Regress Y on observed history at time 3 using `glm()`.

```
> Q3.reg = glm(Y ~ L1 + A1 + L2 + A2, family = "binomial", data = ObsData0)
```

ii. Generate predicted values of Y, evaluated at each observed covariate value and exposure history of interest (i.e., $\bar{L}_i(2) = \bar{l}_i(2)$ and $\bar{A}_i(2) = \bar{a}(2)$, respectively). Set the predicted values equal to `Q3`.

```
> Q3 = predict(Q3.reg, type = "response", newdata = newdata)
```

(c) For time $t = 2$:

i. Using the predicted values from the prior step as a new outcome (i.e., `Q3`), regress the new outcome on the past at time 2.

ii. Generate new predicted values, evaluated at each observed covariate value and exposure history of interest (i.e., $L_i(1) = l_i(1)$ and $A_i(1) = a(1)$, respectively)

(d) Take the empirical mean of the final predicted outcome. This is the value that `ICE.gcomp0_fun()` should return.

2. Estimate $E_{U,X}[Y_{\bar{a}=1}] = P_{U,X}(Y_{\bar{a}=1} = 1)$ using parametric g-computation by applying `ICE.gcomp0_fun()`, with the correct `abar` argument specification.

TMLE - Data Structure 0

1. Create a new function `TMLE.gcomp0_fun()` that takes in `abar` as an argument. Within the function:

(a) Create a dataframe called `newdata` where $A(1)$ and $A(2)$ have been set to $a(1)$ and $a(2)$, respectively.

(b) Estimate the probability of receiving treatment $P_0(A(t) = 1|\bar{L}(t), \bar{A}(t-1)) = g_0(A(t) = 1|\bar{L}(t), \bar{A}(t-1))$ for $t = 1, 2$ using correctly specified parametric regression models. The correct model specifications are (refer back to `R` lab 3 for reference):

$$g_0(A(1) = 1|L(1)) = expit[\beta_0 + \beta_1 L(1)]$$
$$g_0(A(2) = 1|\bar{L}(2)) = expit[\beta_0 + \beta_1 L(1) + \beta_2 L(2)]$$

i. Use the `glm()` function, and specify the correct formula following the above model specifications. Also include the arguments `family = 'binomial'` for logistic regression and `data = ObsData0`. For example, for $t = 1$:

```
> gA1.reg = glm(A1 ~ L1, family = 'binomial', data = ObsData0)
```

Do the same for $t = 2$.

ii. Predict each subject's probability of receiving treatment at time t, given his or her observed treatment and covariate history i.e., $g_n(A_i(t) = 1|\bar{A}_i(t-1), \bar{L}_i(t))$. Assign to the variables `g.abar1.1` and `g.abar2.1` (respectively). For example, for $t = 2$:

```
> g.abar2.1 = predict(gA2.reg, type = "response")
```

Do the same for $t = 1$.

(c) For each timepoint $t = 1, 2$, create a logical variable that indicates which students had a treatment history $\bar{A}(t) = \bar{a}(t)$. For example for $t = 1$:

```
> I1 = ObsData0$A1 == abar[1]
```

Do the same for $t = 2$.

(d) For $t = 3$:

i. Estimate $\bar{Q}^0_{3,n}(\bar{a}(2), \bar{L}(2))$. This is the conditional probability of $Y = 1$ (i.e., `Q3`), given the past covariates and exposure of interest.

A. Regress Y on the *observed past*:

```
> Q3.reg = glm(Y ~ L1 + A1 + L2 + A2, data = ObsData0, family = "binomial")
```

B. Using the above model, predict $\bar{Q}^0_{3,n}$, the conditional probability on the logit scale for all subjects *at the exposure history we want* (i.e., $\bar{A}(2) = \bar{a}(2)$). Call this vector `logit.Q3`.

```
> logit.Q3 = predict(Q3.reg, type = "link", newdata = newdata)
```

ii. Update $\bar{Q}^0_{3,n}(\bar{a}(2), \bar{L}(2))$ to $\bar{Q}^\star_{3,n}(\bar{a}(2), \bar{L}(2))$.

A. Make the clever covariate, $H_{n,3}(\bar{A}(2), \bar{L}(2)) = \frac{\mathbb{I}[\bar{A}(2)=\bar{a}(2)]}{\prod_{t=1}^{2} g_n(A(t)|\bar{L}(t), \bar{A}(t-1))}$, where the denominator, $\prod_{t=1}^{2} g_n(A(t)|\bar{L}(t), \bar{A}(t-1))$, is bounded between 0.01 and 1:

```
> H3 = I2/bound(g.abar1.1 * g.abar2.1)
```

B. Fit a logistic regression of Y on the intercept, with $logit(\bar{Q}^0_{3,n})$ as an offset and clever covariate $H_{n,3}(\bar{A}(2), \bar{L}(2))$ as weights:

```
> Q3.reg.update = glm(Y ~ offset(logit.Q3), weights = H3, family = "binomial",
+                      data = ObsData0)
```

C. Generate $\bar{Q}^\star_{3,n}$, the updated predicted probabilities of $\bar{Q}^0_{3,n}$ using the updated model in the previous step:

```
> Q3.star = predict(Q3.reg.update, type = "response")
```

(e) For $t = 2$:

i. Estimate $\bar{Q}^0_{2,n}(a(1), L(1))$. This is the conditional expectation of $\bar{Q}^\star_{3,n}$, given the past covariates and exposure of interest.

A. Regress $\bar{Q}^\star_{3,n}$ on the *observed past*. Call this `Q2.reg`. Make sure to specify `data = ObsData0` and `family = 'quasibinomial'`.

B. Using the above model, predict $logit(\bar{Q}_{2,n})$, the conditional probabiity on the logit scale for all subjects at the exposure history we want (i.e., $A(1) = a(1)$). Call this vector `logit.Q2`.

ii. Update $\bar{Q}^0_{2,n}(a(1), L(1))$ to $\bar{Q}^\star_{2,n}(a(1), L(1))$

A. Make the clever covariate, $H_{n,2}(A(1), L(1)) = \frac{\mathbb{I}[A(1)=a(1)]}{g_n(A(1)|L(1))}$, where $g_n(A(1)|L(1))$ is bounded between 0.01 and 1. Call this `H2`.

B. Fit a logistic regression on the intercept, with $logit(\bar{Q}^0_{1,n})$ as an offset and clever covariate $H_{n,2}(A(1), L(1))$ as weights.

C. Generate $\bar{Q}^\star_{2,n}$, the updated predicted probabilities of $\bar{Q}_{2,n}$ using the updated model in the previous step.

(f) Take the mean of $\bar{Q}^\star_{2,n}$. This is your TMLE estimate! Have `TMLE.gcomp0_fun()` return this value.

2. Estimate $E_{U,X}[Y_{\bar{a}=1}] = P_{U,X}(Y_{\bar{a}=1} = 1)$ using TMLE by applying `TMLE.gcomp0_fun()`, with the correct `abar` argument specification.

<u>**Estimator performance metrics**</u> - Data Structure 0 (refer back to `R` Lab 4 for bias, variance, and MSE definitions).

1. Set the number of iterations `B` to 5 (to start).
2. Create a matrix `estimates_data0` with `B` rows and `3` columns. Name the columns of the matrix `Param, ICE, TMLE`.
3. Within a for loop from `b` to `1:B`, do the following:
   (a) Redraw n copies of the data using the `generate_data0()` function you loaded earlier, and set equal to the object `ObsData0`.
   (b) Implement the parametric g-computation, ICE g-computation, and TMLE using the functions `param.gcomp0_fun()`, `ICE.gcomp0_fun()`, and `TMLE.gcomp0_fun()`, respectively, to estimate the causal parameter of interest, making sure you have specified the correct `abar` vector as an argument for each.
   (c) Save the estimates in the b^{th} row of the `estimates_data0` matrix.
4. When you are confident that your code is working, increase the number of iterations `B = 500` and rerun your code. *Warning: this may take a long time!*
5. For each estimator, estimate the:
   - Bias. *Hint:* use the `colMeans()` function.
   - Variance. *Hint:* use the `var()` function on the estimates to get the covariance matrix, and take the diagonal of that matrix using the `diag()` function to get each estimator's variance.
   - MSE. *Hint:* use the `colMeans()` function.

Data Structure 2: $O = (L(1), A(1), L(2), A(2), L(3), A(3), L(4), A(4), Y)$

We are interested in the difference in the expected test score if all students got 8 or more hours of sleep for all 4 nights before the test versus if all students got less than 8 hours of sleep for all 4 nights before the test is 13.5891 points.

$$\Psi^F(P_{U,X}) = E_{U,X}[Y_{\bar{a}(4)=1} - Y_{\bar{a}(4)=0}] = 13.5891$$

Before we get started with estimation...

1. Load `DataStructure2.RData` using the `load()` function. Make sure you have specified the correct file path. You should see 6 new things come up in your global environment:

   - `ObsData2` - this is a dataframe of 1,000 observations that follows Data Structure 2 from previous labs.
   - `Psi.F2` - this is the true $\Psi^F(P_{U,X})$ value for the target causal parameter $E_{U,X}[Y_{\bar{a}(4)=1}] - E_{U,X}[Y_{\bar{a}(4)=0}]$ (generated in lab 2).
   - `generate_data2` - this is the function that generates n copies of Data Structure 2.
   - `generate_data2_intervene`, `TrueMSMbeta1`, `TrueMSMbeta1_wts` - we won't be using these objects, so you can remove them from your global environment if you'd like.

2. Assign the number of students to `n`.

3. Create the function `rescale0to1()`, a function that takes as input a vector of numbers `Y` and returns the numbers rescaled between 0 and 1:

```
> rescale0to1 = function(Y) {
+   rescaleY = (Y - min(Y))/(max(Y) - min(Y))
+   return(rescaleY)
+ }
```

   We will apply this function on our outcome, Y, in the TMLE sections.

<u>Parametric g-computation</u> - Data Structure 2

1. Create a new function `param.gcomp2_fun()` that takes in `abar` as an argument. Within the function:

   (a) First, estimate the conditional distributions of all the non-intervention nodes, in this case, $L(1)$, $L(2)$, $L(3)$, $L(4)$ and Y. For this example, we need estimates of:

      i. $\bar{Q}(Y|\bar{L}(4), \bar{A}(4))$, or the conditional expectation of Y, given the past.
      ii. $\bar{Q}(L(t)|\bar{L}(t-1), \bar{A}(t-1))$, or the conditional expectation of $L(t)$, given the past, for $t = 2, 3, 4$.
      iii. $\bar{Q}(L(1))$, the baseline covariate distribution. To estimate the distribution of $L(1)$, we can use the empirical distribution. *Hint:* go to the next step!

   (b) Use these estimates to generate (using Monte Carlo simulation) many "counterfactual" covariate and outcome histories over time, setting $A(t) = a(t)$ for $t = 1, ..., 4$.

      i. Set `S` equal to $10,000$, the number of times we will simulate.
      ii. Sample `S` observations/rows, with replacement, from `ObsData2`.
      iii. Draw $L_i(1) = l_i(1)$ for each individual. That is, set `l1` equal to `L1` that lives within the simulated data.
      iv. Draw $L_i(2) = l_i(2)$ for each individual.

         A. From the estimated conditional distribution of $Q_n(L(2)|L(1), A(1))$, predict the expectation of $L(2)$, setting baseline exposure and covariate to $a(1)$ and $l(1)$, respectively. That is:

```
> Q.L2 = predict(Q.L2.reg,
+                newdata = data.frame(L1 = l1,
+                                     A1 = abar[1]))
```

B. Because $L(2)$ is continuous, for each individual, draw an observation from a $Normal(\mu_i = \bar{Q}_n(L_i(2)|L_i(1) = l_i(1), A_i(1) = a_i(1)), \sigma_i = \hat{\sigma}_{L(2)})$ distribution.

```
> l2 = rnorm(S, mean = Q.L2, sd = sd(ObsData2$L2))
```

v. Draw $L_i(3) = l_i(3)$ for each individual.

A. From the estimated conditional distribution of $Q_n(L(3)|\bar{L}(2), \bar{A}(2))$, predict the expectation of $L(3)$, setting baseline exposure and covariate to $\bar{a}(2)$ and $\bar{l}(2)$, respectively.

B. Because $L(3)$ is continuous, for each individual, draw an observation from a $Normal(\mu_i = \bar{Q}(L_i(3)|\bar{L}_i(2) = \bar{l}_i(2), \bar{A}_i(2) = \bar{a}_i(2)), \sigma_i = \hat{\sigma}_{L(3)})$ distribution.

vi. Draw $L_i(4) = l_i(4)$ and $Y_i = y_i$ for each individual, extending the steps used to draw $l_i(2)$ and $l_i(3)$ above.

(c) Estimate the expectation of Y with the empirical mean of `y`. Have the function `param.gcomp2_fun()` return this value.

2. Estimate $E_{U,X}[Y_{\bar{a}=1} - Y_{\bar{a}=0}]$ using parametric g-computation by applying `param.gcomp0_fun()`, with the correct `abar` argument specifications.

ICE g-computation - Data Structure 2

1. Create a new function `ICE.gcomp2_fun()` that takes in `abar` as an argument. Within the function:

(a) Create a dataframe called `newdata` where $A(t)$ has been set to $a(t)$ for $t = 1, ..., 4$:

```
> newdata = ObsData2
> newdata$A1 = abar[1]
> newdata$A2 = abar[2]
> newdata$A3 = abar[3]
> newdata$A4 = abar[4]
```

(b) Rescale Y from `ObsData2` to be between 0 and 1 using the `rescale0to1()` function, and set it equal to the variable `Y.scaled`.

(c) For time $t = 5$:

i. Regress rescaled Y on *observed history* at time 5 using `glm()`.

ii. Generate predicted values of rescaled Y, evaluated at each observed covariate value and exposure history of interest (i.e., $\bar{L}_i(4) = \bar{l}_i(4)$ and $\bar{A}_i(4) = \bar{a}(4)$, respectively). Set the predicted values equal to `Q5`.

(d) For time $t = 4, 3, 2$:

i. Using the predicted values from the prior step as a new outcome, regress the new outcome on the *observed past* at time t.

ii. Generate new predicted values, evaluated *at each observed covariate value and exposure history of interest* (i.e., $\bar{L}_i(t-1) = \bar{l}_i(t-1)$ and $\bar{A}_i(t-1) = \bar{a}(t-1)$, respectively).

(e) Take the empirical mean of the final predicted outcome. This is the value that `ICE.gcomp2_fun()` should return.

2. Estimate $E_{U,X}[Y_{\bar{a}=1} - Y_{\bar{a}=0}]$ using parametric g-computation by applying `ICE.gcomp2_fun()`, with the correct `abar` argument specification.

TMLE - Data Structure 2

1. Create a new function `TMLE.gcomp2_fun()` that takes in `abar` as an argument. Within the function:

(a) Create a dataframe called `newdata` where $A(t)$ has been set to $a(t)$ for $t = 1, ..., 4$.

(b) Estimate the probability of each person's observed exposure at time t, i.e., $P_0(A(t)|\bar{L}(t), \bar{A}(t-1)) = g_0(A(t)|\bar{L}(t), \bar{A}(t-1))$ for $t = 1, ..., 4$

i. First, estimate the probability of receiving treatment at time t $P_0(A(t) = 1|\bar{L}(t), \bar{A}(t-1)) = g_0(A(t) = 1|\bar{L}(t), \bar{A}(t-1))$ for $t = 1, ..., 4$ using correctly specified parametric regression models. The correct model specifications are (refer back to `R` lab 3 for reference):

$$g_0(A(1) = 1|L(1)) = expit[\beta_0 + \beta_1 L(1)]$$
$$g_0(A(2) = 1|\bar{L}(2), A(1)) = expit[\beta_0 + \beta_1 L(1) + \beta_2 A(1) + \beta_3 L(2)]$$
$$g_0(A(3) = 1|\bar{L}(3), \bar{A}(2)) = expit[\beta_0 + \beta_1 L(1) + \beta_2 A(1) + \beta_3 L(2) + \beta_4 A(2) + \beta_5 L(3)]$$
$$g_0(A(4) = 1|\bar{L}(4), \bar{A}(3)) = expit[\beta_0 + \beta_1 L(1) + \beta_2 A(1) + \beta_3 L(2) + \beta_4 A(2) + \beta_5 L(3) + \beta_6 A(3) + \beta_7 L(4)]$$

ii. Use the `glm()` function, and specify the arguments `family = 'binomial'` for logistic regression and `data = ObsData2`.

iii. Predict each subject's probability of the exposure at time t, given their observed exposure and observed covariate history, i.e., $g_n(A_i(t) = 1|\bar{A}_i(t-1), \bar{L}_i(t))$, for $t = 1, ..., 4$.

iv. Obtain the conditional probabilities of each subject's observed exposure, $g_n(A_i(t)|\bar{A}_i(t-1), \bar{L}_i(t))$. That is, for each timepoint ($t = 1, ..., 4$):
  - Among subjects who got $A(t) = 1$ at time t, assign the predicted probability as $g_n(A_i(t) = 1|\bar{A}_i(t-1), \bar{L}_i(t))$.
  - Similarly, among subjects who got $A(t) = 0$ at time t, assign the predicted probability as $g_n(A_i(t) = 0|\bar{A}_i(t-1), \bar{L}_i(t))$.

  For example, for timepoint 1:
```
> g.abar1 = (ObsData2$A1 == 1) * g.abar1.1 + (ObsData2$A1 == 0) * (1 - g.abar1.1)
```
  Repeat for $t = 2, 3, 4$.

(c) For each timepoint $t = 1, ..., 4$, create a logical variable that indicates which students had a treatment history $\bar{A}(t) = \bar{a}(t)$. For example for $t = 1$:
```
> I1 = ObsData2$A1 == abar[1]
```
Do the same for $t = 2, 3, 4$.

(d) For $t = 5$:

i. Rescale Y from `ObsData2` to be between 0 and 1 using the `rescale0to1()` function, and set it equal to the variable `Y.scaled`.

ii. Estimate $\bar{Q}^0_{5,n}(\bar{a}(4), \bar{L}(4))$. This is the conditional expectation of Y (i.e., `Y.scaled`), given the past covariates and exposure of interest.

  A. Regress rescaled Y on the *observed past*, $\bar{A}(4)$ and $\bar{L}(4)$:
```
> Q5.reg = glm(Y.scaled ~ L1 + A1 + L2 + A2 + L3 + A3 + L4 + A4,
+               data = ObsData2, family = "quasibinomial")
```
  B. Using the above model, predict $logit(\bar{Q}_{5,n})$, the conditional outcome for all subjects on the logit scale *at the exposure history we want* (i.e., $\bar{A}(4) = \bar{a}(4)$). Call this vector `logit.Q5`.
```
> logit.Q5 = predict(Q5.reg, type = "link", newdata = newdata)
```

iii. Update $\bar{Q}^0_{5,n}(\bar{a}(4), \bar{L}(4))$ to $\bar{Q}^\star_{5,n}(\bar{a}(4), \bar{L}(4))$

  A. Make the clever covariate, $H_{n,5}(\bar{A}(4), \bar{L}(4)) = \frac{\mathbb{I}[\bar{A}(4)=\bar{a}(4)]}{\prod_{t=1}^{4} g_n(A(t)|\bar{L}(t),\bar{A}(t-1))}$, where the denominator $\prod_{t=1}^{4} g_n(A(t)|\bar{L}(t), \bar{A}(t-1))$ is bounded between 0.01 and 1:
```
> H5 = I4/bound(g.abar1 * g.abar2 * g.abar3 * g.abar4)
```
  B. Fit a logistic regression of Y on the intercept, with $logit(\bar{Q}^0_{5,n})$ as an offset and clever covariate $H_{n,5}(\bar{A}(4), \bar{L}(4))$ as weight:
```
> Q5.reg.update = glm(Y.scaled ~ offset(logit.Q5), weights = H5,
+                     family = "quasibinomial")
```
  C. Generate $\bar{Q}^\star_{5,n}$, the predicted probabilities of the updated model in the previous step:
```
> Q5.star = predict(Q5.reg.update, type = "response")
```

(e) For $t = 4$:

i. Estimate $\bar{Q}^0_{4,n}(\bar{a}(3), \bar{L}(3))$. This is the conditional expectation of $\bar{Q}^\star_{5,n}$ (i.e., `Q5.star`), given the past covariates and exposure of interest.

A. Regress $\bar{Q}^{\star}_{5,n}$ on the *observed past*, $\bar{A}(3)$ and $\bar{L}(3)$:

```
> Q4.reg = glm(Q5.star ~ L1 + A1 + L2 + A2 + L3 + A3,
+              data = ObsData2, family = "quasibinomial")
```

B. Using the above model, predict $logit(\bar{Q}_{4,n})$, the conditional outcome for all subjects *at exposure history we want* (i.e., $\bar{A}(3) = \bar{a}(3)$). Call this vector `logit.Q4`.

```
> logit.Q4 = predict(Q4.reg, newdata = newdata, type = "link")
```

ii. Update $\bar{Q}^{0}_{4,n}(\bar{a}(3), \bar{L}(3))$ to $\bar{Q}^{\star}_{4,n}(\bar{a}(3), \bar{L}(3))$

A. Make the clever covariate, $H_{n,4}(\bar{A}(3), \bar{L}(3)) = \frac{\mathbb{I}[\bar{A}(3)=\bar{a}(3)]}{\prod_{t=1}^{3} g_n(A(t)|\bar{L}(t),\bar{A}(t-1))}$, where the denominator $\prod_{t=1}^{3} g_n(A(t)|\bar{L}(t), \bar{A}(t-1))$ is bounded between 0.01 and 1:

```
> H4 = I3/bound(g.abar1 * g.abar2 * g.abar3)
```

B. Fit a logistic regression of $\bar{Q}^{\star}_{5,n}$ on the intercept, with $logit(\bar{Q}^{0}_{4,n})$ as an offset and clever covariate $H_{n,4}(\bar{A}(3), \bar{L}(3))$ as weight:

```
> Q4.reg.update = glm(Q5.star ~ offset(logit.Q4), weights = H4, family = "quasibinomial")
```

C. Generate $\bar{Q}^{\star}_{4,n}$, the predicted probabilities of the updated model in the previous step:

```
> Q4.star = predict(Q4.reg.update, type = "response")
```

(f) For $t = 3, 2$:

i. Estimate $\bar{Q}^{0}_{t,n}(\bar{a}(t-1), \bar{L}(t-1))$. This is the conditional expectation of $\bar{Q}^{\star}_{t+1,n}$, given the past covariates and exposure of interest.

A. Regress $\bar{Q}^{\star}_{t+1,n}$ on the *observed past*, $\bar{A}(t-1)$ and $\bar{L}(t-1)$

B. Using the above model, predict $logit(\bar{Q}_{t,n})$, the conditional outcome on the logit scale for all subjects *at exposure history we want* (i.e., $\bar{A}(t-1) = \bar{a}(t-1)$)

ii. Update $\bar{Q}^{0}_{t,n}(\bar{a}(t-1), \bar{L}(t-1))$ to $\bar{Q}^{\star}_{t,n}(\bar{a}(t-1), \bar{L}(t-1))$

A. Make the clever covariate, $H_{n,t}(\bar{A}(t-1), \bar{L}(t-1)) = \frac{\mathbb{I}[\bar{A}(t-1)=\bar{a}(t-1)]}{\prod_{j=1}^{t-1} g_n(A(j)|\bar{L}(j),\bar{A}(j-1))}$

B. Fit a logistic regression of $\bar{Q}^{\star}_{t+1,n}$ on the intercept, with $logit(\bar{Q}^{0}_{t,n})$ as an offset and clever covariate $H_{n,t}(\bar{A}(t-1), \bar{L}(t-1))$ as weight.

C. Generate $\bar{Q}^{\star}_{t,n}$, the predicted probabilities of the updated model in the previous step.

(g) Take the mean of $\bar{Q}^{\star}_{2,n}$ and back-transform this number to the original scale of Y. For example, if `meanQ2.star` is the mean of $\bar{Q}^{\star}_{2,n}$, this would be the code to back-transform the estimate:

```
> meanQ.star.back = meanQ2.star*(max(ObsData2$Y) - min(ObsData2$Y)) + min(ObsData0$Y)
```

This is your TMLE estimate! Have `TMLE.gcomp2_fun()` return this value.

2. Estimate $E_{U,X}[Y_{\bar{a}=1} - Y_{\bar{a}=0}]$ using TMLE by applying `TMLE.gcomp2_fun()`, with the correct `abar` argument specifications.

Estimator performance metrics - Data Structure 2 (refer back to `R` Lab 4 for bias, variance, and MSE definitions).

1. Set the number of iterations `B` to 5 (to start).

2. Create a matrix `estimates_data2` with `B` rows and `3` columns. Name the columns of the matrix `Param, ICE, TMLE`.

3. Within a for loop from `b` to `1:B`, do the following:

(a) Redraw n copies of the data using the `generate_data2()` function you loaded earlier, and set equal to the object `ObsData2`.

(b) Implement the parametric g-computation, ICE g-computation, and TMLE using the functions `param.gcomp2_fun()`, `ICE.gcomp2_fun()`, and `TMLE.gcomp2_fun()`, respectively, to estimate the causal parameter of interest, making sure you have specified the correct `abar` vector as an argument for each.

(c) Save the estimates in the b^{th} row of the `estimates_data2` matrix.

4. When you are confident that your code is working, increase the number of iterations `B = 500` and rerun your code.

5. For each estimator, estimate the bias, variance, and MSE.

2 For Your Project: Estimation via forms of g-computation

Think through the following questions and apply them to the dataset you will use for your final project.

1. Implement the parametric g-computation, ICE representation of g-computation, and TMLE estimators on your real data and simulated data.
2. For the simulated data only: obtain bias, variance, MSE for the three estimators based on the true value of the causal parameter you generated in `R` Lab 2.

3 Optional Feedback

You may attach responses to these questions to your lab. Thank you in advance!

1. Did you catch any errors in this lab? If so, where?
2. What did you learn in this lab?
3. Do you think that this lab met the goals listed at the beginning?
4. What else would you have liked to review? What would have helped your understanding?
5. Any other feedback?