R Lab 1 - Simulating Longitudinal Data

Advanced Topics in Causal Inference

Assigned: September 07, 2021
Lab due: September 14, 2021 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:
1. Review of `R` programming and good programming practice.
2. Create basic function to generate data with $O = (W, A, Y)$ structure.
3. Review g-computation, IPTW, and TMLE estimators for point treatment intervention.

Goals for this lab:
1. Explore different data structures, inspired by real-world experiments.
2. Simulate data generating systems that give rise to data we observe.
3. Think of studies in which these data generating systems may occur.

Next lab:
We will translate causal questions into target causal parameters and intervene on the Structural Causal Models (SCMs) to evaluate them.

---

1 Introduction and Motivation

1.1 What is data simulation?

Data simulation is the process of sampling repeatedly from specified data generating distributions. In practice, this means generating random numbers from known distributions (that may or may not depend on each other), and repeating this process a large number of times. In a simulation, unlike reality, you know the truth about the underlying processes that give rise to your data.

1.2 Why simulate?

There are many reasons to simulate. Here are a few:

1. **By simulating, we get an understanding of the performance of estimators.** Suppose you come up with an estimator – a function that takes in as input your observed data and gives as output an estimate of a particular statistical estimand, say the g-computation formula corresponding (under assumptions) to the population average treatment effect. For a given sample size and data-generating process, how well will your estimator actually estimate your target statistical estimand? How biased will it be? How variable?

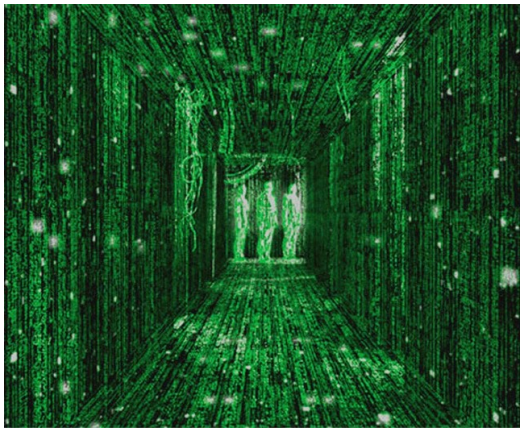

Figure 1: Simulation.

How is its behavior affected by things like data sparsity? What is its behavior like as the number of observations grows?

2. **Simulations are also helpful in evaluating your proposed approach to statistical inference.** Are your confidence intervals attaining their desired coverage? Across multiple repetitions of the same experiment, what percentage of the time will your proposed approach to confidence interval construction contain the true parameter value? Is your approach to hypothesis testing achieving its desired level of Type I error control? If the null hypothesis is true, what percentage of the time is your proposed approach to hypothesis testing end up rejecting the null?
3. **Simulations are useful for power analyses and sample size calculations.** You have a proposed study design and corresponding estimator. How big should your study be? For a given sample size and data generating process, how often will your estimator reject the null hypothesis? How big does your sample size need to be to ensure that the null hypothesis is rejected an acceptable proportion of the time?

2 This lab

2.1 Background

UC Berkeley School of Public Health is concerned about the amount of sleep students are getting. They've heard that lack of sleep is hurting students' academic performance and health. As part of a GSR project, you've been asked to study the effects of getting 8 or more hours of sleep on 1) students' statistics exam scores and 2) students' likelihood of getting sick.

Lucky for you, we have *perfect* knowledge of how sleep and other related variables work with each other to impact students' test performances and health outcomes. This means you'll know the generating system that gives rise to the variables you'd need to answer the scientific question of interest for your GSR.

For the first two data generating systems, you'll be asked questions related to the effect of 8 or more hours of sleep on students' statistics exam scores. For the last two data generating systems, we are interested in the effect of getting 8 or more hours of sleep on students' likelihood of becoming ill. In the next lab, we will evaluate these effects by intervening on the structural causal models presented below.

2.2 Link between observed data and causal model

In this lab we will simulate different data structures based on structural causal models (SCMs) that will help you answer your research questions on the effects of sleep. But what relates an SCM to the data we actually observe?

- In reality we might observe n copies of O, the observed ordered data structure of the above variables, from the observed data distribution, $P_0 \in \mathcal{M}$, (recall that $\mathcal{M}$ is a statistical model of which the true distribution, P_0, is an element).
- When working in the structural causal model framework, we assume that the observed data O were generated by sampling n times from a data generating system compatible with the causal model $\mathcal{M}^{\mathcal{F}}$. The distribution of the exogenous variables U together with the structural equations F identify the joint distribution of (U, X), $P_{U,X}$, and thus the distribution of the observed data O (which is defined as a subset of X).
- In this lab we will define several processes for generating the data (U, X), each of which defines a distribution $P_{U,X}$ of (U, X), and by extension, a distribution P_0 of O. If we have background knowledge about a system we want to study, we can incorporate that when we specify the data generating process we wish to study in a simulation. In this lab, we will then use each data generating process that we define to generate a single sample of the observed data: n i.i.d. copies of O (in general, when using simulations to, for example, study estimator performance, one will repeat this process many times).
- Because we know the true underlying data generating process in a simulation, we can also calculate the true value of parameters of both statistical estimands and causal parameters under specific interventions on parameters of $P_{U,X}$ (next lab!).

2.3 To turn in:

For each of the 4 data structures listed below, answer the following questions:

1. **Write a function that generates n copies following the above data generating process.** The function should:
   (a) Take in as input `n`, the number of observations you want to generate
   (b) Output a dataframe with n i.i.d. copies of O.

   Hint: Use the function in `R` Lab 0 that simulates a data-generating process for $O = (W, A, Y)$ as a template!
2. **Generate $n = 1000$ i.i.d. copies of O and store in a dataframe.**
3. **Show the first 6 lines and summary statistics of O** using the `head()` and `summary()` functions, respectively.
4. **Describe a study/experiment that could be represented by this data generating system.**

3 Data Structures

Data Structure 1: $O = (W, A, L, \Delta, \Delta Y)$

This data generating system represents an example with a baseline covariate, a single intervention, one covariate in between the intervention and outcome, and an outcome that is subject to missingness.

Let's define the following variables:

U - exogenous errors
W - baseline covariate that is a standardized measure of how many hours a student naps
A - the exposure of interest is sleep status for the night's sleep before the test. We define A as a binary variable indicating whether or not student got 8 or more hours of sleep the night before the test[1]:

$$A = \begin{cases} 1, & \geq 8 \text{ hours of sleep the night before the test} \\ 0, & < 8 \text{ hours of sleep the night before the test} \end{cases}$$

L - a continuous covariate that measures the student's stress level the day of the test. The higher the value, the more the stress level.
Δ - a binary variable of missingness indicating whether or not we observe the outcome (e.g., whether or not the student turns in a statistics test). Specifically:

$$\Delta = \begin{cases} 1, & \text{if we observe the outcome} \\ 0, & \text{otherwise} \end{cases}$$

Y - student's statistics test score between 0 and 100.

Underlying data generating process, $P_{U,X}$:

Exogenous variables:

$$\begin{aligned} U_W &\sim Uniform(min = 0, max = 1) \\ U_A &\sim Uniform(min = 0, max = 1) \\ U_L &\sim Normal(\mu = 2, \sigma^2 = 1^2) \\ U_\Delta &\sim Uniform(min = 0, max = 1) \\ U_Y &\sim Normal(\mu = 72, \sigma^2 = 0.3^2) \end{aligned}$$

Structural equations F and endogenous variables:

$$\begin{aligned} W &= U_W \\ A &= \mathbb{I}\big[U_A < expit(0.01^*W)\big] \\ L &= W + A + U_L \\ \Delta &= \mathbb{I}\big[U_\Delta < expit(0.01^*(W + A + L))\big] \\ Y &= L + 5^*A + 3^*W - 0.25^*A^*W + U_Y \end{aligned}$$

1. **Set the seed to 252**.

2. **Write a function that simulates the specified data generating process**. Call this function `generate_data1()`:

[1] Can you think of some concerns or challenges that result from defining your exposure by categorizing an underlying continuous variable? There is an extensive discussion/debate in the literature about this. See for example:

- Petersen, Maya L. "Compound treatments, transportability, and the structural causal model: the power and simplicity of causal graphs." *Epidemiology* 22.3 (2011): 378-381.
- Hernán, Miguel A., and Tyler J. VanderWeele. "Compound treatments and transportability of causal inference." *Epidemiology (Cambridge, Mass.)* 22.3 (2011): 368.

(a) Create your exogenous variables, $U_W, U_A, U_L, U_\Delta, U_Y$, using the `rnorm()` and `runif()` functions. Careful with your parameters!

(b) Recall that the *expit* function is the inverse of the logistic function:

$$logit(x) = log\left(\frac{x}{1-x}\right)$$
$$expit(x) = \frac{1}{1+e^{-x}}$$

In `R`, the *expit* function is called `plogis()`.

(c) Use the `as.numeric()` function for endogenous indicator variables.
Example: $\mathbb{I}[X > 0]$ can be coded as `as.numeric(X>0)`.

(d) For a given subject, *if* $\Delta = 0$, *then* Y is missing (which we code as `NA`). *Otherwise*, keep Y as is. You may use the `ifelse()` function on `Y` to code this.
Example: Suppose x is a vector with the numbers $[3, 1, 0, -1]$. *If* an element of x is greater than 0, *then* return 1, *otherwise*, return 0.

```
> x = c(3, 1, 0, -1)
> ifelse(x > 0, yes = 1, no = 0)
```

```
[1] 1 1 0 0
```

(e) Make sure your function takes in an argument `n` (the number of copies of O) and returns back a dataframe with the endogenous variables.

3. **Generate $n = 1000$ copies and store in a dataframe.** Call this dataframe `ObsData1`.

4. **Show the first 6 lines and summary statistics of O.**

5. **Describe a study/experiment that could be represented by this data generating system.**

6. **Bonus:** What does this data generating process encode about the effect of Δ on Y? How would this be encoded in an SCM? Is it realistic in this setting?

Solution:

```
> # 1. set the seed
> set.seed(252)


> # 2. write function to generate O = (W,A,L,Delta,Delta*Y)
> generate_data1 <- function(n){
+   # exogenous variables
+   U.W = runif(n, min=0, max=1)
+   U.A = runif(n, min=0, max=1)
+   U.L = rnorm(n, mean=2, sd=1)
+   U.Delta = runif(n, min=0, max=1)
+   U.Y = rnorm(n, mean=72, sd=0.3)
+
+   # endogenous variables
+   W = U.W
+   A = as.numeric(U.A < plogis(0.01*W))
+   L = W + A + U.L
+   Delta = as.numeric(U.Delta < plogis(0.01*(W + A + L)))
+   DeltaY = ifelse(Delta == 0, NA, L + 5*A + 3*W - 0.25*A*W + U.Y)
```

```
+
+   # store all variables in dataframe
+   O = data.frame(W, A, L, Delta, DeltaY)
+
+   return(O)
+ }

> # 3. generate 1000 observations and set equal to ObsData1
> ObsData1 = generate_data1(n = 1000)

> # 4. show head() and summary() functions on ObsData1
> head(ObsData1)

          W A         L Delta   DeltaY
1 0.8976079 1 5.4000270     0       NA
2 0.7121359 0 2.4393497     1 76.38029
3 0.3274262 0 0.9762165     0       NA
4 0.7678558 0 4.1686958     0       NA
5 0.6831118 0 2.7152281     0       NA
6 0.3716002 0 3.5146374     0       NA

> summary(ObsData1)

       W                   A                L               Delta
 Min.   :0.0001883   Min.   :0.000   Min.   :-0.755   Min.   :0.000
 1st Qu.:0.2828826   1st Qu.:0.000   1st Qu.: 2.171   1st Qu.:0.000
 Median :0.5176949   Median :1.000   Median : 3.023   Median :1.000
 Mean   :0.5136629   Mean   :0.505   Mean   : 3.031   Mean   :0.501
 3rd Qu.:0.7479553   3rd Qu.:1.000   3rd Qu.: 3.895   3rd Qu.:1.000
 Max.   :0.9999480   Max.   :1.000   Max.   : 6.973   Max.   :1.000

     DeltaY
 Min.   :72.13
 1st Qu.:76.31
 Median :79.72
 Mean   :79.21
 3rd Qu.:82.08
 Max.   :85.85
 NA's   :499
```

5. This data-generating system represents a scenario in which we have a single baseline covariate, intervention, covariate after the intervention, and outcome that is subject to missingness. Notice that there are 499 missing values for `DeltaY` in the summary output – this means 499 students didn't take the statistics test (uh oh!). Also notice that in this data generating example, whether or not the outcome is measured can depend on both baseline and time varying covariates (affected by the exposure) as well as the exposure itself. Thinking ahead, how do you think this might causes challenges?

Another example of a study where we might see this data generating process is in a study looking at the effect of a medication on a health-related outcome within Electronic Medical Records (EMR), where we collect demographic variables as baseline covariates, whether or not the patient took the drug as the treatment, physiological measures as the longitudinal covariate, and a health-related outcome (such as blood pressure), where some patients simply never appear in the EMR database for a record of their outcome.

6. Δ does not affect the underlying Y – we would encode this as an exclusion restriction. In other words, the process by which the test score is observed (i.e., turning in the test) does not affect the underlying test score, just whether or not it is observed.

Is this a realistic assumption? It depends on what Δ represents (i.e., what is missingness due to?). For example, it might not be realistic if it means forcing someone to class and take the test. Further if Δ is an indicator that a student turns in the test after he or she takes it, one could critique the time ordering encoded here – the true underlying test score for your completed test might actually affect whether you turn it in!

Data Structure 2: $O = (\bar{L}(K), \bar{A}(K), Y)$ for $K = 4$
$= (L(1), A(1), L(2), A(2), L(3), A(3), L(4), A(4), Y)$

This data structure represents a classic longitudinal experiment with time-varying treatment and covariates.

Variable definitions:

U - exogenous errors
$A(t)$ - the exposure of interest for this study is sleep status for the night's sleep starting on day t. In this example, we define $A(t)$ as a binary variable indicating whether or not student got 8 or more hours of sleep starting on day t:

$$A(t) = \begin{cases} 1, & \geq 8 \text{ hours of sleep starting on day } t \\ 0, & < 8 \text{ hours of sleep starting on day } t \end{cases}$$

$L(t)$ - a continuous covariate that measures the student's stress level on day t. Higher values correspond to more stress levels.
Y - the student's statistics test score between 0 and 100.

Underlying data generating process:

Exogenous variables:

$$U_{L(t)}, t = 1, ..., 4 \sim Normal(\mu = 0, \sigma^2 = 1)$$
$$U_{A(t)}, t = 1, ..., 4 \sim Uniform(min = 0, max = 1)$$
$$U_Y \sim Normal(\mu = 72, \sigma^2 = 3^2)$$

Structural equations F and endogenous variables:

$$L(1) = U_{L(1)}$$
$$A(1) = \mathbb{I}\big[U_{A(1)} < expit(0.001 * L(1))\big]$$
$$L(2) = A(1) + L(1) + U_{L(2)}$$
$$A(2) = \mathbb{I}\big[U_{A(2)} < expit(0.001 * (L(2) + A(1) + L(1)))\big]$$
$$L(3) = A(1) + L(1) + A(2) + L(2) + U_{L(3)}$$
$$A(3) = \mathbb{I}\big[U_{A(3)} < expit(0.001 * (L(1) + A(1) + L(2) + A(2) + L(3)))\big]$$
$$L(4) = A(1) + L(1) + A(2) + L(2) + A(3) + L(3) + U_{L(4)}$$
$$A(4) = \mathbb{I}\big[U_{A(4)} < expit(0.001 * (L(1) + A(1) + L(2) + A(2) + L(3) + A(3) + L(4)))\big]$$
$$Y = 0.3^*L(1) + A(1) + 0.5^*L(2) + A(2) + 0.7^*L(3) + A(3) + L(4) + A(4) - U_Y + 130$$

1. **Set the seed to 252.**
2. **Write a function `generate_data2()` that simulates the specified data generating process**. Use hints from the previous data structure to generate your endogenous and exogenous variables.

3. **Generate 1000 i.i.d. copies of O and store in a dataframe `ObsData2`.**
4. **Show the first 6 lines and summary statistics of O.**
5. **Describe a study/experiment that could be represented by this data generating system.**

Solution:

```
> #1. set the seed
> set.seed(252)


> # 2. write function to generate O = (L(1), A(1),...,Y)
> generate_data2 <- function(n){
+   # exogenous variables
+   U.L1 = rnorm(n, mean=0, sd=1)
+   U.A1 = runif(n, min=0, max=1)
+   U.L2 = rnorm(n, mean=0, sd=1)
+   U.A2 = runif(n, min=0, max=1)
+   U.L3 = rnorm(n, mean=0, sd=1)
+   U.A3 = runif(n, min=0, max=1)
+   U.L4 = rnorm(n, mean=0, sd=1)
+   U.A4 = runif(n, min=0, max=1)
+   U.Y = rnorm(n, mean=72, sd=3)
+
+   # endogenous variables
+   L1 = U.L1
+   A1 = as.numeric(U.A1 < plogis(0.001*L1))
+   L2 = A1 + L1 + U.L2
+   A2 = as.numeric(U.A2 < plogis(0.001*(L2+A1 + L1)))
+   L3 = A1 + L1 + A2 + L2 + U.L3
+   A3 = as.numeric(U.A3 < plogis(0.001*(L1 + A1 + L2 + A2 + L3)))
+   L4 = A1 + L1 + A2 + L2 + A3 + L3 + U.L4
+   A4 = as.numeric(U.A4 < plogis(0.001*(L1 + A1 + L2 + A2 + L3 + A3 + L4)))
+
+   Y = 0.3*L1 + A1 + 0.5*L2 + A2 + 0.7*L3 + A3 + L4 + A4 - U.Y + 130
+
+   O = data.frame(L1, A1, L2, A2, L3, A3, L4, A4, Y)
+
+   return(O)
+
+ }


> #3. generate 1000 observations and set equal to ObsData3
> ObsData2 = generate_data2(n=1000)


> #4. show head() and summary() functions on ObsData3
> head(ObsData2)


          L1 A1        L2 A2        L3 A3         L4 A4        Y
1  1.2680386  0 0.2710493  0 1.0204073  0  3.5358595  0 67.06123
2 -0.4470315  1 2.3191348  1 3.7767181  1  7.7017530  1 73.38921
```

```
3  0.4764182  1 2.1832829  1 6.0291888  1 11.4392710  1 73.07440
4 -0.1605500  1 0.4025755  0 0.3506208  1  0.6516012  0 56.70239
5 -0.8010446  1 1.6915839  1 3.0802601  0  6.0265842  1 69.76773
6  0.2079778  1 0.2083505  0 1.5280887  1  3.4192575  1 65.67612


> summary(ObsData2)


       L1                  A1               L2                A2
 Min.   :-3.556007   Min.   :0.000   Min.   :-4.3756   Min.   :0.000
 1st Qu.:-0.598577   1st Qu.:0.000   1st Qu.:-0.4662   1st Qu.:0.000
 Median : 0.006724   Median :0.000   Median : 0.4992   Median :1.000
 Mean   : 0.033832   Mean   :0.476   Mean   : 0.5019   Mean   :0.502
 3rd Qu.: 0.642610   3rd Qu.:1.000   3rd Qu.: 1.5211   3rd Qu.:1.000
 Max.   : 3.880840   Max.   :1.000   Max.   : 4.7066   Max.   :1.000
       L3                A3              L4                  A4
 Min.   :-7.0027   Min.   :0.00   Min.   :-12.80268   Min.   :0.000
 1st Qu.:-0.2439   1st Qu.:0.00   1st Qu.:  0.02516   1st Qu.:0.000
 Median : 1.4354   Median :1.00   Median :  3.29660   Median :0.000
 Mean   : 1.5300   Mean   :0.51   Mean   :  3.50466   Mean   :0.496
 3rd Qu.: 3.4183   3rd Qu.:1.00   3rd Qu.:  7.35755   3rd Qu.:1.000
 Max.   :11.3654   Max.   :1.00   Max.   : 24.23476   Max.   :1.000
       Y
 Min.   :37.88
 1st Qu.:59.08
 Median :64.68
 Mean   :64.90
 3rd Qu.:70.87
 Max.   :96.64
```

5. Here, we are getting measures of students' stress levels over time, how much sleep they're getting over time, and their test scores at the end of the study. This data structure also exemplifies any study in which $A(t)$ is a time-varying treatment or exposure and $L(t)$ is a time-varying covariate that is affected by prior exposure and affects future exposure. One common example is $L(t)$ as a time varying indication of disease severity or need for treatment. In other words, if a subject is sick, they are more likely to get the treatment, which affects subsequent sickness, propensity to treatment, and, ultimately, the outcome.

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

Here, $Y(t)$ is an indicator variable describing whether or not the student has become ill by time t; as such, it deterministically jumps to 1 and remains there once an individual has become ill. This is an example of a "survival" or "time to event" data structure. For outcomes such as this, once the event of interest has occurred, by definition, no future interventions can change it. We could thus truncate the observed data for an individual at the point that the event occurs (this is often done when defining survival data structures). However, in the longitudinal causal inference literature, for notational convenience, we often instead define the values of observed variables as something deterministic (such as last observed value) after the event occurs. This allows us to write down a data structure that extends through time $K + 1$ for all individuals (instead of a data structure where the number of time points varies by individual).

Definition of variables:

U - exogenous errors
$A(t)$ - the exposure of interest for this study is sleep status for the night's sleep starting on day t. In this

example, we define $A(t)$ as a binary variable indicating whether or not student got 8 or more hours of sleep starting on day t:

$$A(t) = \begin{cases} 1, & \geq 8 \text{ hours of sleep starting on day } t \\ 0, & < 8 \text{ hours of sleep starting on day } t \end{cases}$$

$L(t)$ - a continuous covariate that measures the student's stress level on day t. Higher values correspond to higher stress levels.
$Y(t)$ is an indicator variable describing whether or not the student became ill by time t:

$$Y(t) = \begin{cases} 1, & \text{student became sick at or before time } t \\ 0, & \text{otherwise} \end{cases}$$

Underlying data generating process:

Exogenous variables:

$$U_{L(t)}, t = 1, 2 \sim Normal(\mu = 0, \sigma^2 = 1)$$
$$U_{A(t)}, t = 1, 2 \sim Uniform(min = 0, max = 1)$$
$$U_{Y(t+1)}, t = 1, 2 \sim Uniform(min = 0, max = 1)$$

Structural equations F and endogenous variables:

$$L(1) = U_{L(1)}$$
$$A(1) = \mathbb{I}\big[U_{A(1)} < expit(.001^* L(1))\big]$$
$$Y(2) = \mathbb{I}\big[U_{Y(2)} < expit(L(1) - 2^* A(1) - 6)\big]$$
$$L(2) = \begin{cases} A(1) + L(1) + U_{L(2)} & \text{if } Y(2) = 0 \\ \texttt{NA} & \text{if } Y(2) = 1 \end{cases}$$
$$A(2) = \begin{cases} \mathbb{I}\big[U_{A(2)} < expit(.001 * (L(1) + A(1) + L(2)))\big] & \text{if } Y(2) = 0 \\ \texttt{NA} & \text{if } Y(2) = 1 \end{cases}$$
$$Y(3) = \begin{cases} \mathbb{I}\big[U_{Y(3)} < expit(L(1) - 2^* A(1) + L(2) - A(2))\big] & \text{if } Y(2) = 0 \\ 1 & \text{if } Y(2) = 1 \end{cases}$$

1. **Set the seed to 252.**
2. **Write a function that simulates the data generating process.** Call the function `generate_data3()`.
   (a) Create the exogenous variables as in the previous two data-generating processes.
   (b) Create $L(1)$, $A(1)$ and $Y(2)$ as usual.
   (c) If a subject has the event Y (i.e., gets sick) at time 2, then:
   - The values that we assign deterministically to subsequent covariates and exposures will not affect the values of target causal parameters or their identification or corresponding estimands, nor will they be used in estimators of these estimands. We could assign them to have value equal to last observed value before the event occured. However, for simplicity we here just set them to `NA`. That is: *if* $Y(2)$ is 1 for a given subject, *then* $L(2)$ and $A(2)$ should be `NA`. *Otherwise*, $L(2)$ and $A(2)$ should take on the value defined in the structural equations.
     Hint: use the `ifelse()` function to generate $L(2)$ and $A(2)$'s missing values.
   - Subjects should remain as having had the event at subsequent timepoints. So, *if* $Y(2)$ is 1 for a given subject, *then* $Y(3)$ should be 1, as well. *Otherwise*, $Y(3)$ should take on the value defined in the structural equations.
     Hint: use the `ifelse()` function to generate $Y(3)$'s values.

3. **Generate 1000 copies of O and store in a dataframe called `ObsData3`.**
4. **Show the first 6 lines and summary statistics of O.**
5. **Describe a study/experiment that could be represented by this data generating system.**

Solution:

```
> #1. set the seed
> set.seed(252)


> #2. write function to generate O = (L(1), A(1), Y(2), ...Y(3))
> generate_data3 <- function(n){
+   # exogenous variables
+   U.L1 = rnorm(n, mean=0, sd=1)
+   U.A1 = runif(n, min=0, max=1)
+   U.Y2 = runif(n, min=0, max=1)
+   U.L2 = rnorm(n, mean=0, sd=1)
+   U.A2 = runif(n, min=0, max=1)
+   U.Y3 = runif(n, min=0, max=1)
+
+   # endogenous variables
+   L1 = U.L1
+   A1 = as.numeric(U.A1 < plogis(.001*L1))
+   Y2 = as.numeric(U.Y2 < plogis(L1 - 2*A1 - 6))
+   L2 = ifelse(Y2 == 1, NA, A1 + L1 + U.L2)
+   A2 = ifelse(Y2 == 1, NA, as.numeric(U.A2 < plogis(0.001*(L1 + A1 + L2))))
+   Y3 = ifelse(Y2 == 1, 1, as.numeric(U.Y3 < plogis(L1 - 2*A1 + L2 - A2)))
+
+   O = data.frame(L1, A1, Y2, L2, A2, Y3)
+
+   return(O)
+ }


> #3. generate 1000 observations and set equal to ObsData3
> ObsData3 = generate_data3(n=1000)


> #4. show head() and summary() functions on ObsData3
> head(ObsData3)


          L1 A1 Y2       L2 A2 Y3
1  1.2680386  0  0 2.567049  1  1
2 -0.4470315  1  0 2.846987  0  1
3  0.4764182  1  0 3.645934  1  0
4 -0.1605500  1  0 1.724645  1  0
5 -0.8010446  1  0 1.279684  0  0
6  0.2079778  1  0 1.279420  1  1


> summary(ObsData3)
```

```
       L1                  A1               Y2              L2
 Min.   :-3.556007   Min.   :0.000   Min.   :0.000   Min.   :-3.8943
 1st Qu.:-0.598577   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:-0.4887
 Median : 0.006724   Median :0.000   Median :0.000   Median : 0.4958
 Mean   : 0.033832   Mean   :0.476   Mean   :0.003   Mean   : 0.5089
 3rd Qu.: 0.642610   3rd Qu.:1.000   3rd Qu.:0.000   3rd Qu.: 1.5585
 Max.   : 3.880840   Max.   :1.000   Max.   :1.000   Max.   : 5.7955
                                                     NA's   :3
       A2               Y3
 Min.   :0.0000   Min.   :0.000
 1st Qu.:0.0000   1st Qu.:0.000
 Median :0.0000   Median :0.000
 Mean   :0.4995   Mean   :0.365
 3rd Qu.:1.0000   3rd Qu.:1.000
 Max.   :1.0000   Max.   :1.000
 NA's   :3
```

5. This data generating mechanism represents a survival structure in which we have a time-to-event outcome (i.e., time until illness), and $Y(t)$ is an indicator that the event has occured (i.e., the student got sick). Note that once a student becomes ill, $Y(t)$ is set to 1 thereafter. How we set $A(t)$ and $L(t)$ after a student becomes ill will not affect our estimates. Additionally, for this structure we have no loss to follow up (or censoring). An example of a study that would follow this data structure could be one investigating the effect of some binary treatment on time to death. For example, the subject is assigned to either one of two treatments at the first time point, denoted by $A(1)$. Then, at time $t = 2$, we define $Y(2)$ as the indicator of mortality by time 2, and, if the person is alive, we measure $L(2)$ at time 2. Then, it is indicated by $A(2)$ which treatment the subject recevies at the second time point. Once someone dies, $Y(t)$ is set to 1 thereafter.

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

We now introduce an example of a data structure with both a time to event (survival) outcome and a right censoring variable. Define $C(t)$ as an indicator variable whether or not the student dropped the class before day t. Thus, once $C(t)$ has jumped to 1 it remains there deterministically. Further, in general after a right censoring event (in this case, dropping the class) occurs, no more data is observed on an individual. However, similar to the survival example above, for notational convenience, we typically arbitrarily define the values of observed variables after a censoring event occurs (e.g., as deterministically equal to their last observed value). As for the survival example, this allows us to avoid having data structures of variable lengths, depending on the timing of failure time and censoring events, and instead to define a single data structure of length $K + 1$ that applies to all individuals, regardless of if and when they are censored or experience the failure event of interest.

Variable definitions:

U - exogenous errors
$A(t)$ - sleep status for the night's sleep starting on day t.$A(t)$ is a binary variable indicating whether or not student got 8 or more hours of sleep starting on day t:

$$A(t) = \begin{cases} 1, & \geq 8 \text{ hours of sleep starting on day } t \\ 0, & < 8 \text{ hours of sleep starting on day } t \end{cases}$$

$L(t)$ - a continuous covariate that measures the student's stress level on day t. Higher values of L correspond to higher stress levels.
$C(t)$ - an indicator variable describing whether or not the student dropped the class before day t.

$Y(t)$ is an indicator variable describing whether or not the student became ill by time t:

$$Y(t) = \begin{cases} 1, & \text{student became sick at or before time } t \\ 0, & \text{otherwise} \end{cases}$$

Underlying data generating process:

Exogenous variables:

$$U_{L(t)}, t = 1, 2 \sim Normal(\mu = 0, \sigma^2 = 1^2)$$
$$U_{C(t)}, t = 1, 2 \sim Uniform(min = 0, max = 1)$$
$$U_{A(t)}, t = 1, 2 \sim Uniform(min = 0, max = 1)$$
$$U_{Y(t+1)}, t = 1, 2 \sim Uniform(min = 0, max = 1)$$

Structural equations F and endogenous variables:

$$L(1) = U_{L(1)}$$
$$C(1) = \mathbb{I}\big[U_{C(1)} < expit(.001^* L(1) - 2)\big]$$
$$A(1) = \begin{cases} \mathbb{I}\big[U_{A(1)} < expit(.001^* L(1))\big] & \text{if } C(1) = 0 \\ \texttt{NA} & \text{if } C(1) = 1 \end{cases}$$
$$Y(2) = \begin{cases} \mathbb{I}\big[U_{Y(2)} < expit(L(1) - 2^* A(1) - 6)\big] & \text{if } C(1) = 0 \\ \texttt{NA} & \text{if } C(1) = 1 \end{cases}$$
$$L(2) = \begin{cases} L(1) + A(1) + U_{L(2)} & \text{if } Y(2) = 0 \text{ and } C(1) = 0 \\ \texttt{NA} & \text{if } Y(2) = 1 \text{ or } C(1) = 1 \end{cases}$$
$$C(2) = \begin{cases} \mathbb{I}\big[U_{C(2)} < expit(.001^*(L(1) + A(1) + L(2)) - 2)\big] & \text{if } C(1) = 0 \text{ and } Y(2) = 0 \\ \texttt{NA} & \text{if } C(1) = 1 \text{ or } Y(2) = 1 \end{cases}$$
$$A(2) = \begin{cases} \mathbb{I}\big[U_{A(2)} < expit(0.001^*(L(1) + A(1) + L(2))\big] & \text{if } Y(2) = 0 \text{ and } C(2) = 0 \\ \texttt{NA} & \text{if } Y(2) = 1 \text{ or } C(2) = 1 \end{cases}$$
$$Y(3) = \begin{cases} \mathbb{I}\big[U_{Y(3)} < expit(L(1) - 2^* A(1) + L(2) - A(2))\big] & \text{if } Y(2) = 0 \text{ and } C(2) = 0 \\ 1 & \text{if } Y(2) = 1 \\ \texttt{NA} & \text{if } Y(2) = 0 \text{ and } C(2) = 1 \end{cases}$$

1. **Set the seed to 252.**
2. **Write a function `generate_data4()` that simulates the specified data generating process.**
   (a) Generate the endogenous variables, $L(1)$ and $C(1)$ as usual.
   (b) If a subject is censored at time t, we set the values of most variables after censoring has occurred deterministically. We could assign them to have a value equal to the last observed value before the event occurred. However, for simplicity, here we just set them to `NA`. An important exception is $Y(t)$ – once the failure event has been observed to occur, it is known to be deterministically equal to 1 thereafter.
   Hint: use the `ifelse()` function to generate missing values of variables after right censoring occurs.
   (c) As in the previous data structure, if a subject has the event (i.e., gotten sick) at time t (i.e., $Y(t) = 1$), then:
   - They should remain as having had had the event at subsequent timepoints. So, *if* $Y(t)$ is 1 for a given subject, *then* $Y(t+1), Y(t+2), ..., Y(K+1)$ should be 1, as well. *Otherwise*, $Y(t+1), Y(t+2), ..., Y(K+1)$ should take on the value defined in the structural equations.
   Hint: use the `ifelse()` function to generate $Y(t+1), Y(t+2), ..., Y(K+1)$'s values.

- Again, as in the previous data structure, the values that we assign deterministically to subsequent covariates and exposures will not affect the values of target causal parameters or their identification or corresponding estimands, nor will they be used in estimators of these estimands. We could assign them to have value equal to last observed value before the event occured. However, for simplicity we here just set them to `NA`. That is: *if* $Y(t)$ is 1 for a given subject, *then* $A(t+1), ...A(K)$ should be `NA`. *Otherwise*, $A(t+1), ...A(K)$ should take on the value defined in the structural equations. Same goes for generating L's after an event has occurred.
  Hint: use the `ifelse()` function to generate $L(t+1), A(t+1), ..., L(K), A(K)$'s missing values.

3. **Generate 1000 copies of O and store in a dataframe `ObsData4`.**
4. **Show the first 6 lines and summary statistics of `ObsData4`.**
5. **Describe a study/experiment that could be represented by this data generating system.**

Solution:

```
> #1. set the seed
> set.seed(252)


> #2. write function to generate O = (L(1), C(1), A(1), Y(2), ..., Y(3))
> generate_data4 <- function(n){
+   # exogenous variables
+   U.L1 = rnorm(n, mean=0, sd=1)
+   U.C1 = runif(n, min=0, max=1)
+   U.A1 = runif(n, min=0, max=1)
+   U.Y2 = runif(n, min=0, max=1)
+   U.L2 = rnorm(n, mean=0, sd=1)
+   U.C2 = runif(n, min=0, max=1)
+   U.A2 = runif(n, min=0, max=1)
+   U.Y3 = runif(n, min=0, max=1)
+
+   # endogenous variables
+   L1 = U.L1
+   C1 = as.numeric(U.C1 < plogis(.001*L1 - 2))
+   A1 = ifelse(C1 == 1, NA, as.numeric(U.A1 < plogis(.001*L1)))
+   Y2 = as.numeric(U.Y2 < plogis(L1 - 2*A1 - 6))
+   L2 = ifelse(Y2 == 1, NA, A1 + L1 + U.L2)
+   C2 = as.numeric(U.C2 < plogis(.001*(L1 + A1 + L2) - 2))
+   A2 = ifelse(Y2 == 1 | C2 == 1, NA, as.numeric(U.A2 < plogis(0.001*(L1 + A1 + L2))))
+   Y3 = ifelse(Y2 == 1, 1, as.numeric(U.Y3 < plogis(L1 - 2*A1 + L2 - A2)))
+
+   O = data.frame(L1, C1, A1, Y2, L2, C2, A2, Y3)
+
+   return(O)
+ }


> #3. generate 1000 observations and set equal to ObsData4
> ObsData4 = generate_data4(n = 1000)


> #4. show head() and summary() functions on ObsData4
> head(ObsData4)
```

```
          L1 C1 A1 Y2         L2 C2 A2 Y3
1  1.2680386  0  1  0  2.3688849  1 NA NA
2 -0.4470315  0  1  0 -0.0985898  0  1  0
3  0.4764182  0  0  0  0.3127658  0  1  1
4 -0.1605500  0  1  0  1.1823613  0  1  0
5 -0.8010446  1 NA NA         NA NA NA NA
6  0.2079778  0  1  0  1.5510319  1 NA NA
```

```
> summary(ObsData4)
```

```
       L1                  C1               A1               Y2
 Min.   :-3.556007   Min.   :0.000   Min.   :0.0000   Min.   :0.00000
 1st Qu.:-0.598577   1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.00000
 Median : 0.006724   Median :0.000   Median :0.0000   Median :0.00000
 Mean   : 0.033832   Mean   :0.103   Mean   :0.4816   Mean   :0.00223
 3rd Qu.: 0.642610   3rd Qu.:0.000   3rd Qu.:1.0000   3rd Qu.:0.00000
 Max.   : 3.880840   Max.   :1.000   Max.   :1.0000   Max.   :1.00000
                                     NA's   :103      NA's   :103
       L2                C2               A2               Y3
 Min.   :-4.0909   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
 1st Qu.:-0.5075   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
 Median : 0.4864   Median :0.0000   Median :1.0000   Median :0.0000
 Mean   : 0.4881   Mean   :0.1073   Mean   :0.5206   Mean   :0.3833
 3rd Qu.: 1.5833   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
 Max.   : 5.3100   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
 NA's   :105       NA's   :105      NA's   :201      NA's   :199
```

5. This is another longitudinal survival structure allowing for censoring. In the context of the current study, this data structure looks at the time until a student becomes sick in the class, taking into account whether or not a student dropped the class during the study. Again, once a student becomes ill, future $Y(t)$'s are set to 1, and whatever we set $A(t)$ and $L(t)$ to after a student becomes ill will not affect the true value of causal parameters defined by interventions on $\bar{A}$, identification, or estimators. Another example of a study that would fit this study design would be time until cardiovascular death after some treatment intervention, allowing for loss to follow up from the study.

Note that this is an alternative way of representing a missing data structure (in contrast to data structure 1). There, we defined a data generating process for an underlying variable (Y, in that case representing test score). And then defined the observed outcome ΔY as the product of the true underlying out come Y and an indicator that the outcome was measured (Δ).

An alterantive, used here, is to define the data generating process directly for the observed random variables. So for example, $L(t)$ is defined in the data generating process as a variable whose true value is only observed if a subject remains alive and uncensored.

Both approaches are valid, but in general when working with longitudinal data structures we will use the latter, as it simplifies our notation.

4 For Your Project: Simulations

Think through the following questions and apply them to the dataset you will use for your final project. This exercise is not required for R Lab 1 submission.

1. **Define your variables**
   (a) We have not yet worked on defining specific target casual parameters of interested. However, thinking ahead, what variables are you interested in intervening on? What are your outcome variables? When were your variables measured?
2. **Go through first step of roadmap (i.e., specify the causal model representing real background knowledge)**.
   (a) Draw a DAG to represent how you believe the variables relate to each other based on background knowledge. Do this for two timepoints.
   (b) Using those relationships drawn in the previous step, define your structural equations generically; in other words, don't assuming distributions or functional forms yet. Do you have any prior knowledge on the functional forms?
3. **Make histograms for continuous variables and tables for binary/categorical variables.**
   (a) What shapes do the distributions seem to take? Based on the shape, what known distribution do you think that variable's error term is drawn from?
   (b) If you've picked a distribution for an exogenous variable, how would you parameterize it?
4. **Is there missingness in your dataset?**
   (a) Which variables are subject to missingness? The covariates? treatment? outcome?
   (b) Is there censoring? In other words, are there people who, at a certain timepoint, have missing values for the rest of the study?
   (c) Based on *a priori* knowledge, do you expect missingness to depend on other variables? Which ones?
5. **Come up with more specific structural equations that relate the endogenous variables based on the previous two questions.**
6. **Create a function to simulate your data and generate $n = 1000$ copies of your O.**
   (a) Check the histograms and summary statistics of the variables in your simulated data and see how they match up to your real data. Over the course of the class you can refine your data generating structure to match the data at hand.

5 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?