simcausal R Package: Simplifying Simulation Studies of Causal Effects with Structural Equation Models

September 14, 2016

simcausal

- Simulations conducted with the package can help evaluate:
  - asymptotic properties of an estimator
  - finite sample bias
  - relative efficiency of two estimators

simcausal

- Relies on the logic of NPSEM for model specification
- A single pipeline for conducting a “typical” simulation study:
  1. Define the distribution of the observed data using the SEM
  2. Specify interventions (static, dynamic or stochastic)
  3. Simulate observed data or intervention-specific (counterfactual) data
  4. Evaluate the “gold standard” defined by various causal effects:
     - ★ Treatment-specific means
     - ★ The average treatment effects (ATE)
     - ★ Coefficients from working marginal structural models

simcausal Installation

- Preferred installation route:

```
devtools::install_github('osofr/simcausal', build_vignettes = FALSE)
```

- Can also install directly from CRAN (might not be the latest version):

```
install.packages("simcausal")
```

simcausal Roadmap for simulation with single time point

- Initiate specification of the SEM with **DAG.empty** function:

```r
library(simcausal)
D <- DAG.empty()
```

simcausal Roadmap for simulation with single time point

- Initiate specification of the SEM with **DAG.empty** function:

```r
library(simcausal)
D <- DAG.empty()
```

- Add a node (structural equation) to grow **DAG** object:

```r
D <- D + node("CVD", distr="rcat.b1", probs = c(0.5, 0.25, 0.25))
```

simcausal Roadmap for simulation with single time point

- Initiate specification of the SEM with **DAG.empty** function:

```
library(simcausal)
D <- DAG.empty()
```

- Add a node (structural equation) to grow **DAG** object:

```
D <- D + node("CVD", distr="rcat.b1", probs = c(0.5, 0.25, 0.25))
```

- The **node** function consists of the following arguments:
  - node name: **CVD** (cardiovascular disease)
  - node distribution: **distr** = "**rcat.b1**"
  - parameters of node distribution: **probs** = **c(0.5, 0.25, 0.25)**

`simcausal` Roadmap for simulation with single time point

- We proceed to add another node **A1C** (blood glucose):

simcausal Roadmap for simulation with single time point

- We proceed to add another node **A1C** (blood glucose):

```
D <- D + node("A1C", distr="rnorm", mean = 5 + (CVD > 1)*10 + (CVD > 2)*5)
```

- The above node is:
  - Normally distributed: **rnorm**
  - Its mean a function of previously simulated **CVD** values (sd=1)

simcausal Roadmap for simulation with single time point

- We proceed to add another node **A1C** (blood glucose):

```
D <- D + node("A1C", distr="rnorm", mean = 5 + (CVD > 1)*10 + (CVD > 2)*5)
```

- The above node is:
  - Normally distributed: **rnorm**
  - Its mean a function of previously simulated **CVD** values (sd=1)
- Proceed to add Bernoulli treatment **TI** (treatment intensification) and Bernoulli outcome **Y** (risk of heart attack within a year):

```
D <- D +
  node("TI", distr="rbern", prob = plogis(-0.5 - 0.3*CVD + 0.2*A1C)) +
  node("Y", distr="rbern", prob = plogis(-3 + 1.2*TI + 0.1*CVD + 0.3*A1C))
```

simcausal Roadmap for simulation with single time point

- We proceed to add another node **A1C** (blood glucose):

```
D <- D + node("A1C", distr="rnorm", mean = 5 + (CVD > 1)*10 + (CVD > 2)*5)
```

- The above node is:
  - Normally distributed: **rnorm**
  - Its mean a function of previously simulated **CVD** values (sd=1)
- Proceed to add Bernoulli treatment **TI** (treatment intensification) and Bernoulli outcome **Y** (risk of heart attack within a year):

```
D <- D +
  node("TI", distr="rbern", prob = plogis(-0.5 - 0.3*CVD + 0.2*A1C)) +
  node("Y", distr="rbern", prob = plogis(-3 + 1.2*TI + 0.1*CVD + 0.3*A1C))
```

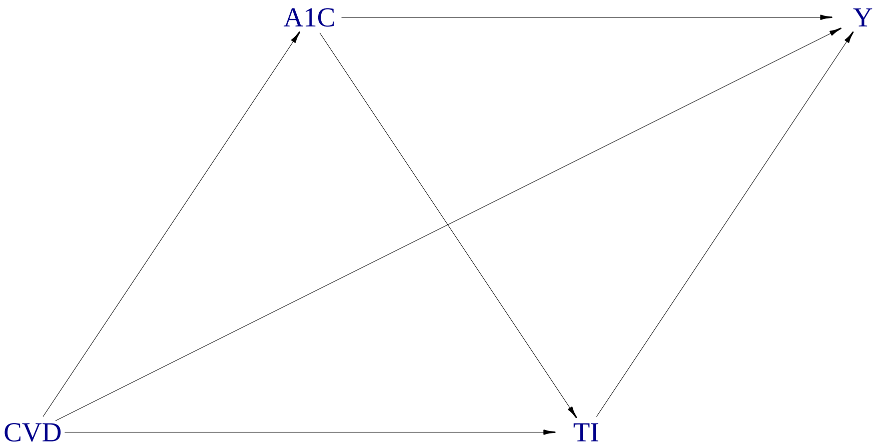
- Finish defining the SEM by calling **set.DAG**() (checks and locks the **DAG** object):

```
setD <- set.DAG(D)
```

simcausal Visualization

- **plotDAG** displays this data generating distribution (SEM):

```
plotDAG(setD, vertex_attrs = list(size = 15, label.cex = 1.5))
```



- Arrows denote node dependencies in the previously defined SEM

simcausal Roadmap for multiple time points

- Lets build on the previous example and consider a slightly more realistic scenario:
  - Allow **A1C**, **TI** and **Y** to change over time for $t = 0, \ldots, 7$, keeping **CVD** unchanged

simcausal Roadmap for multiple time points

- Lets build on the previous example and consider a slightly more realistic scenario:
  - Allow **A1C**, **TI** and **Y** to change over time for $t = 0, \ldots, 7$, keeping **CVD** unchanged
  - Suppose **A1C** at t depends on **TI** at $t - 1$, for each $t > 0$:

simcausal Roadmap for multiple time points

- Lets build on the previous example and consider a slightly more realistic scenario:
  - Allow **A1C**, **TI** and **Y** to change over time for $t = 0, \ldots, 7$, keeping **CVD** unchanged
  - Suppose **A1C** at t depends on **TI** at $t - 1$, for each $t > 0$:

```
D <- D + node("A1C", t = 1:7,
        distr="rnorm",
        mean = -TI[t-1]*10 + 5 + (CVD > 1)*10 + (CVD > 2)*5)
```

simcausal Roadmap for multiple time points

- Lets build on the previous example and consider a slightly more realistic scenario:
  - Allow **A1C**, **TI** and **Y** to change over time for $t = 0, \ldots, 7$, keeping **CVD** unchanged
  - Suppose **A1C** at t depends on **TI** at $t - 1$, for each $t > 0$:

```
D <- D + node("A1C", t = 1:7,
       distr="rnorm",
       mean = -TI[t-1]*10 + 5 + (CVD > 1)*10 + (CVD > 2)*5)
```

- $\mathbf{t = 1 : 7}$ - introduces a new argument to **node** function
- $\mathbf{TI[t-1]}$ - references the time-varying parent of **A1C** with respect to each time point **t**
- $\mathbf{sum(TI[0 : t-1])}$ - can be used to reference the sum of all past treatments

simcausal Roadmap for multiple time points

- Lets build on the previous example and consider a slightly more realistic scenario:
  - Allow **A1C**, **TI** and **Y** to change over time for $t = 0, \ldots, 7$, keeping **CVD** unchanged
  - Suppose **A1C** at t depends on **TI** at $t-1$, for each $t > 0$:

```
D <- D + node("A1C", t = 1:7,
       distr="rnorm",
       mean = -TI[t-1]*10 + 5 + (CVD > 1)*10 + (CVD > 2)*5)
```

- $\mathbf{t = 1 : 7}$ - introduces a new argument to **node** function
- $\mathbf{TI[t-1]}$ - references the time-varying parent of **A1C** with respect to each time point **t**
- $\mathbf{sum(TI[0 : t-1])}$ - can be used to reference the sum of all past treatments
- Above illustrates the ease of extending a SEM for single time point to multiple time points

simcausal Roadmap for multiple time points

- Lets build on the previous example and consider a slightly more realistic scenario:
  - Allow **A1C**, **TI** and **Y** to change over time for $t = 0, \ldots, 7$, keeping **CVD** unchanged
  - Suppose **A1C** at t depends on **TI** at $t - 1$, for each $t > 0$:

```r
D <- D + node("A1C", t = 1:7,
       distr="rnorm",
       mean = -TI[t-1]*10 + 5 + (CVD > 1)*10 + (CVD > 2)*5)
```

- $\mathbf{t = 1 : 7}$ - introduces a new argument to **node** function
- $\mathbf{TI[t - 1]}$ - references the time-varying parent of **A1C** with respect to each time point **t**
- $\mathbf{sum(TI[0 : t - 1])}$ - can be used to reference the sum of all past treatments
- Above illustrates the ease of extending a SEM for single time point to multiple time points
- We define time-varying nodes for **TI** and **Y** in a similar manner, where **Y** is treated as survival outcome (code not shown)

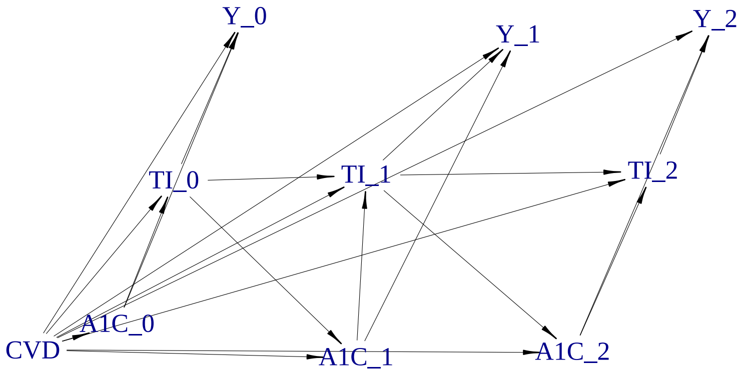
simcausal Visualization

- Visualize the data generating distribution in **setDI** by plotting just first 3 time points points:

simcausal Visualization

- Visualize the data generating distribution in **setDl** by plotting just first 3 time points points:

```
plotDAG(setDl, tmax = 2, xjitter = 0.36, yjitter = 0.08,
  edge_attrs = list(width = 0.5, arrow.width = 0.4, arrow.size = 0.8),
  vertex_attrs = list(size = 19, label.cex = 1.5))
```



simcausal Simulating data

- Function **sim** can be used for simulating and structuring the simulated data:

```
Odat_w <- sim(setDl, n = 5000, rndseed = 3)
```

simcausal Simulating data

- Function **sim** can be used for simulating and structuring the simulated data:

```
Odat_w <- sim(setDl, n = 5000, rndseed = 3)
```

- The default is to simulate data in wide format, with all covariates following censoring event ($\mathbf{Y} = \mathbf{1}$) set to **NA**:

simcausal Simulating data

- Function **sim** can be used for simulating and structuring the simulated data:

```
Odat_w <- sim(setDl, n = 5000, rndseed = 3)
```

- The default is to simulate data in wide format, with all covariates following censoring event ($\mathbf{Y} = \mathbf{1}$) set to **NA**:

```
head(Odat_w, 2)

##   ID CVD     A1C_0 TI_0 Y_0     A1C_1 TI_1 Y_1     A1C_2
## 1  1   1  5.146618    0   0  3.819087    0   0  4.964098
## 2  2   3 18.133367    1   0 10.562405    1   0 10.242300
##   TI_2 Y_2     A1C_3 TI_3 Y_3    A1C_4 TI_4 Y_4    A1C_5
## 1    1   0 -5.209156    0   0  6.78558    0   0 4.452179
## 2    1   0  8.637627    1   0 10.34994    1   0 9.783512
##   TI_5 Y_5     A1C_6 TI_6 Y_6     A1C_7 TI_7 Y_7
## 1    0   0  4.451178    1   0 -6.190455    0   0
## 2    0   0 19.424385    1   1        NA   NA  NA
```

simcausal Simulating data - long format

- Flag **wide** = **FALSE** can be used to structure simulated data in a long format:

```
Odat_l <- sim(setDl, n = 5000, wide = FALSE, rndseed = 3)
```

simcausal Simulating data - long format

- Flag **wide** = **FALSE** can be used to structure simulated data in a long format:

```
Odat_l <- sim(setDl, n = 5000, wide = FALSE, rndseed = 3)
```

```
head(Odat_l)

##   ID CVD t       A1C TI Y
## 1  1   1 0  5.146618  0 0
## 2  1   1 1  3.819087  0 0
## 3  1   1 2  4.964098  1 0
## 4  1   1 3 -5.209156  0 0
## 5  1   1 4  6.785580  0 0
## 6  1   1 5  4.452179  0 0
```

simcausal Defining interventions

- Use the function **action** to specify single/multiple time-point interventions (static, dynamic or stochastic)

simcausal Defining interventions

- Use the function **action** to specify single/multiple time-point interventions (static, dynamic or stochastic)

- For example, start by defining a new distribution for **TI**, using the **node** function:

```
newTRTp <- node("TI",t=1:7, distr="rbern",
        prob = ifelse(TI[t-1]==1,1,ifelse(A1C[t] >= theta,1,0)))
```

simcausal Defining interventions

- Use the function **action** to specify single/multiple time-point interventions (static, dynamic or stochastic)

- For example, start by defining a new distribution for **TI**, using the **node** function:

```
newTRTp <- node("TI",t=1:7, distr="rbern",
        prob = ifelse(TI[t-1]==1,1,ifelse(A1C[t] >= theta,1,0)))
```

- Next, we define two actions (dynamic interventions) based on **TI** above, where each action is indexed by a value of **theta**:

```
setDl <- setDl +
  action("early.switch", nodes = c(newTRTp), theta = 4) +
  action("late.switch", nodes = c(newTRTp), theta = 10)
```

simcausal Defining interventions

- Use the function **action** to specify single/multiple time-point interventions (static, dynamic or stochastic)

- For example, start by defining a new distribution for **TI**, using the **node** function:

```
newTRTp <- node("TI",t=1:7, distr="rbern",
        prob = ifelse(TI[t-1]==1,1,ifelse(A1C[t] >= theta,1,0)))
```

- Next, we define two actions (dynamic interventions) based on **TI** above, where each action is indexed by a value of **theta**:

```
setDl <- setDl +
  action("early.switch", nodes = c(newTRTp), theta = 4) +
  action("late.switch", nodes = c(newTRTp), theta = 10)
```

- Use the same **sim** function with argument **actions** = "**early.switch**" to simulate counterfactual data for that intervention or a collection of interventions

simcausal Defining and evaluating true causal effects

- simcausal can be used for defining and evaluating various causal quantities of interest:
  - Counterfactual node means
  - Average treatment effects (ATE)
  - Coefficients of a working marginal structural model

simcausal Defining and evaluating true causal effects

- simcausal can be used for defining and evaluating various causal quantities of interest:
  - Counterfactual node means
  - Average treatment effects (ATE)
  - Coefficients of a working marginal structural model
- To define the causal target as the counterfactual mean of **Y** over time under intervention "**early.switch**":

```
lDAG <- set.targetE(setDl, outcome="Y", t=0:7, param="early.switch")
```

- ATE can be defined by changing **param** argument to "**early.switch** − **late.switch**" or "**early.switch**/**late.switch**"

simcausal Defining and evaluating true causal effects

- simcausal can be used for defining and evaluating various causal quantities of interest:
  - Counterfactual node means
  - Average treatment effects (ATE)
  - Coefficients of a working marginal structural model
- To define the causal target as the counterfactual mean of **Y** over time under intervention "**early**.**switch**":

```
lDAG <- set.targetE(setDl, outcome="Y", t=0:7, param="early.switch")
```

- ATE can be defined by changing **param** argument to "**early**.**switch** − **late**.**switch**" or "**early**.**switch**/**late**.**switch**"
- To evaluate a previously defined causal quantity use **eval**.**target**:

```
eval.target(lDAG, n = 5000)$res
```

simcausal Defining and evaluating true causal effects

- simcausal can be used for defining and evaluating various causal quantities of interest:
  - Counterfactual node means
  - Average treatment effects (ATE)
  - Coefficients of a working marginal structural model
- To define the causal target as the counterfactual mean of **Y** over time under intervention "**early.switch**":

```
lDAG <- set.targetE(setDl, outcome="Y", t=0:7, param="early.switch")
```

- ATE can be defined by changing **param** argument to "**early.switch** − **late.switch**" or "**early.switch**/**late.switch**"
- To evaluate a previously defined causal quantity use **eval.target**:

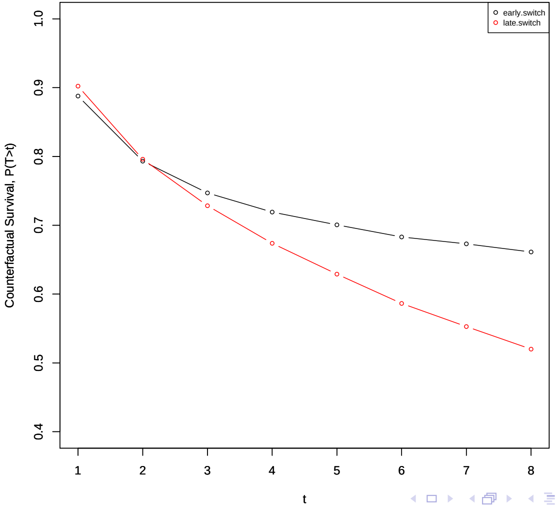
```
eval.target(lDAG, n = 5000)$res
```

- Coefficients of a working marginal structural model are defined with **set.targetMSM**:
  - MSMs can be linear, logistic, or any other type
  - For survival can model hazard or survival function

simcausal Causal effects (survival)

- Counterfactual survival curves for two dynamic interventions:

```
plotSurvEst(surv = list(early.switch = surv_th1, late.switch = surv_th2),
        xindx = 1:8, ylab = "Counterfactual Survival, P(T>t)", ylim = c(0.4,1))
```


early.switch
late.switch
Counterfactual Survival, P(T>t)
t