Project 8 CSS Alex Schulte

```r
# Add to this package list for additional SL algorithms
pacman::p_load(
  tidyverse,
  ggthemes,
  ltmle,
  tmle,
  SuperLearner,
  tidymodels,
  caret,
  dagitty,
  ggdag,
  here)

#heart_disease <- read_csv(here('heart_disease_tmle.csv'))

heart_disease <- read.csv("/Users/alex/git/Computational-Social-Science-Projects/My Project 8/heart_dis
#View(heart_disease)
head(heart_disease)
```

```
##        age sex_at_birth simplified_race college_educ income_thousands      bmi
## 1 32.92951            0               1            2         91.32660 27.09986
## 2 53.92344            1               1            2         38.76890 27.61615
## 3 65.33631            1               3            2         35.48435 27.52499
## 4 16.82682            1               1            2         93.77726 24.88569
## 5 56.06874            1               1            2         85.74291 22.75805
## 6 57.15671            1               1            2         70.80389 24.04710
##   blood_pressure     chol blood_pressure_medication    bmi_2 blood_pressure_2
## 1       146.9862 211.4630                         0 27.08979         146.9800
## 2       125.7578 187.8418                         0 27.61077         125.7505
## 3       131.6789 197.0794                         0 27.52064         131.6626
## 4       131.2926 229.6352                         0 24.89534         131.2765
## 5       130.5946 208.0835                         0 22.76464         130.6103
## 6       129.8528 199.0600                         0 24.03932         129.8640
##     chol_2 blood_pressure_medication_2 mortality
## 1 211.5343                           0         1
## 2 187.7360                           0         0
## 3 196.9674                           0         1
## 4 229.6558                           0         0
## 5 207.9436                           0         0
## 6 199.0715                           0         1
```

Introduction

Heart disease is the leading cause of death in the United States, and treating it properly is an important public health goal. However, it is a complex disease with several different risk factors and potential treatments. Physicians typically recommend changes in diet, increased exercise, and/or medication to treat symptoms, but it is difficult to determine how effective any one of these factors is in treating the disease. In this project, you will explore SuperLearner, Targeted Maximum Likelihood Estimation (TMLE), and Longitudinal Targeted Maximum Likelihood Estimation (LTMLE). Using a simulated dataset, you will explore whether taking blood pressure medication reduces mortality risk.

Data

This dataset was simulated using R (so it does not come from a previous study or other data source). It contains several variables:

- **blood_pressure_medication**: Treatment indicator for whether the individual took blood pressure medication (0 for control, 1 for treatment)
- **mortality**: Outcome indicator for whether the individual passed away from complications of heart disease (0 for no, 1 for yes)
- **age**: Age at time 1
- **sex_at_birth**: Sex assigned at birth (0 female, 1 male)
- **simplified_race**: Simplified racial category. (1: White/Caucasian, 2: Black/African American, 3: Latinx, 4: Asian American,
5: Mixed Race/Other)
- **income_thousands**: Household income in thousands of dollars
- **college_educ**: Indicator for college education (0 for no, 1 for yes)
- **bmi**: Body mass index (BMI)
- **chol**: Cholesterol level
- **blood_pressure**: Systolic blood pressure
- **bmi_2**: BMI measured at time 2
- **chol_2**: Cholesterol measured at time 2
- **blood_pressure_2**: BP measured at time 2
- **blood_pressure_medication_2**: Whether the person took treatment at time period 2

For the "SuperLearner" and "TMLE" portions, you can ignore any variable that ends in "_2", we will reintroduce these for LTMLE.

SuperLearner

Modeling

Fit a SuperLearner model to estimate the probability of someone dying from complications of heart disease, conditional on treatment and the relevant covariates. Do the following:

1. Choose a library of at least 5 machine learning algorithms to evaluate. **Note**: We did not cover how to hyperparameter tune constituent algorithms within SuperLearner in lab, but you are free to do so if you like (though not required to for this exercise).
2. Split your data into train and test sets.
3. Train SuperLearner
4. Report the risk and coefficient associated with each model, and the performance of the discrete winner and SuperLearner ensemble

5. Create a confusion matrix and report your overall accuracy, recall, and precision

```r
# make sure packaages are here
library(SuperLearner)
library(tidyverse)
library(caret)

## Train/Test split. doing this first, makes more sense
# found this way to do it online, seems easier / fewer steps than our lab approach

set.seed(123)
train_index <- createDataPartition(heart_disease$mortality, p = 0.75, list = FALSE) #doing 3/4 like in
train_data <- heart_disease[train_index, ]
test_data <- heart_disease[-train_index, ]

## sl library with 5 I chose
sl_lib <- c("SL.mean", "SL.glmnet", "SL.ranger", "SL.glm", "SL.lm")

## Train SuperLearner
sl_fit <- SuperLearner(Y = train_data$mortality, X = train_data %>%
                       select(blood_pressure_medication, age, sex_at_birth, simplified_race,
                              income_thousands, college_educ, bmi, chol, blood_pressure), # these are t
                       SL.library = sl_lib, method = "method.NNloglik", family = binomial())
```

```
## Loading required namespace: ranger
```

```r
# showing  the risk and coefficient associated with each model overall
summary(sl_fit)
```

```
##                   Length Class  Mode
## call                  6  -none- call
## libraryNames          5  -none- character
## SL.library            2  -none- list
## SL.predict         7500  -none- numeric
## coef                  5  -none- numeric
## library.predict   37500  -none- numeric
## Z                 37500  -none- numeric
## cvRisk                5  -none- numeric
## family               13  family list
## fitLibrary            5  -none- list
## cvFitLibrary          0  -none- NULL
## varNames              9  -none- character
## validRows            10  -none- list
## method                3  -none- list
## whichScreen           9  -none- logical
## control               3  -none- list
## cvControl             4  -none- list
## errorsInCVLibrary     5  -none- logical
## errorsInLibrary       5  -none- logical
## metaOptimizer         5  -none- list
## env                   7  -none- environment
## times                 3  -none- list
```

```r
#now specifically
coefficients <- sl_fit$coef
risks <- sl_fit$coef
```

```r
print(coefficients)
```

```
##   SL.mean_All SL.glmnet_All SL.ranger_All    SL.glm_All     SL.lm_All
##     0.3188064     0.0000000     0.4970268     0.0000000     0.1841669
```

```r
print(risks)
```

```
##   SL.mean_All SL.glmnet_All SL.ranger_All    SL.glm_All     SL.lm_All
##     0.3188064     0.0000000     0.4970268     0.0000000     0.1841669
```

```r
#discrete winner
sl_fit$cvRisk[which.min(sl_fit$cvRisk)]
```

```
## SL.ranger_All
##     0.6446834
```

```r
sl_fit$coef[which.min(sl_fit$coef)]
```

```
## SL.glmnet_All
##             0
```

```r
# getting predictions and reporting confusion matrix
predictions <- predict(sl_fit, newdata = test_data %>%
                         select(blood_pressure_medication, age, sex_at_birth, simplified_race,
                                income_thousands, college_educ, bmi, chol, blood_pressure), type = "resp

predicted_class <- ifelse(predictions$pred >= 0.5, 1, 0)
conf_matrix <- table(Predicted = predicted_class, Actual = test_data$mortality)
confusionMatrix(conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##          Actual
## Predicted    0    1
##         0  392  168
##         1  874 1066
##
##                Accuracy : 0.5832
##                  95% CI : (0.5636, 0.6026)
##     No Information Rate : 0.5064
##     P-Value [Acc > NIR] : 7.925e-15
##
##                   Kappa : 0.1722
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.3096
##             Specificity : 0.8639
##          Pos Pred Value : 0.7000
##          Neg Pred Value : 0.5495
##              Prevalence : 0.5064
##          Detection Rate : 0.1568
##    Detection Prevalence : 0.2240
##       Balanced Accuracy : 0.5867
##
##        'Positive' Class : 0
```

```
##
```

```r
# Accuracy, Recall, and Precision
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
recall <- conf_matrix[2, 2] / sum(conf_matrix[2, ])
precision <- conf_matrix[2, 2] / sum(conf_matrix[, 2])
list(Accuracy = accuracy, Recall = recall, Precision = precision)
```

```
## $Accuracy
## [1] 0.5832
##
## $Recall
## [1] 0.5494845
##
## $Precision
## [1] 0.8638574
```

Discussion Questions

1. Why should we, in general, prefer the SuperLearner ensemble to the discrete winner in cross-validation? Or in other words, what is the advantage of "blending" algorithms together and giving them each weights, rather than just using the single best algorithm (with best being defined as minimizing risk)?

Answer: There are a few different reasons. The obvious one is improved prediction accuracy. By blending multiple algorithms, the ensemble can leverage the strengths of each component algorithm. Another important piece is robustness. An ensemble like SuperLearner is less likely to be overly influenced by peculiarities in the training data that might favor one particular model; this makes it more robust, especially when working with new or unseen data.

Targeted Maximum Likelihood Estimation

Causal Diagram

TMLE requires estimating two models:

1. The outcome model, or the relationship between the outcome and the treatment/predictors, $P(Y|(A, W)$.
2. The propensity score model, or the relationship between assignment to treatment and predictors $P(A|W)$

Using ggdag and daggity, draw a directed acylcic graph (DAG) that describes the relationships between the outcome, treatment, and covariates/predictors. Note, if you think there are covariates that are not related to other variables in the dataset, note this by either including them as freestanding nodes or by omitting them and noting omissions in your discussion.

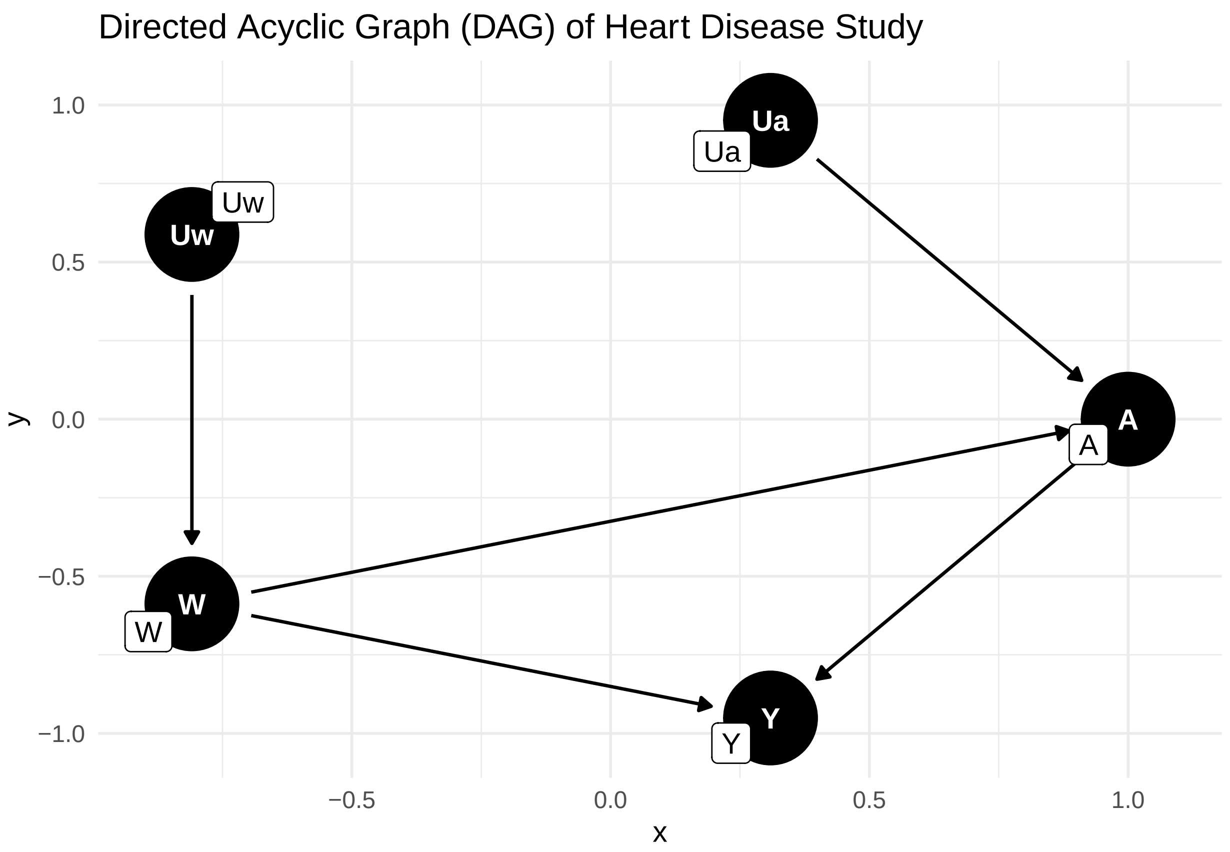
W = covariates (blood_pressure_medication, age, sex_at_birth, simplified_race, income_thousands, college_educ, bmi, chol, blood_pressure) A = treatment Y = outcome Uw = unobserved variables affecting covariates Ua = unobserved variables affecting treatment

```r
# Load necessary libraries
library(ggdag)
library(dagitty)

# Define DAG
#Note i'm using some code I learned previously in a biostats class
dag <- dagitty::dagitty("
dag {
  Uw [unobserved]
  Ua [unobserved]
  W -> A -> Y
  W -> Y
  Uw -> W
  Ua -> A
}
")

ggdag <- ggdag::ggdag(dag, text = TRUE, use_labels = "name", layout = "circle") +
  theme_minimal() +
  ggtitle("Directed Acyclic Graph (DAG) of Heart Disease Study")

print(ggdag)
```



TMLE Estimation

Use the `tmle` package to estimate a model for the effect of blood pressure medication on the probability of mortality. Do the following:

1. Use the same SuperLearner library you defined earlier
2. Use the same outcome model and propensity score model that you specified in the DAG above. If in your DAG you concluded that it is not possible to make a causal inference from this dataset, specify a simpler model and note your assumptions for this step.
3. Report the average treatment effect and any other relevant statistics

Note: Assuming no unobserved confounders.

```r
# Load necessary libraries
library(SuperLearner)
library(tmle)
library(tidyverse)

#library
sl_lib <- c("SL.mean", "SL.glmnet", "SL.ranger", "SL.glm", "SL.lm")

# Define the outcome, treatment, and covariates
Y <- "mortality"
A <- "blood_pressure_medication"
W <- c("age", "sex_at_birth", "simplified_race", "income_thousands", "college_educ", "bmi", "chol", "bl

# Run the TMLE
tmle_fit <- tmle(
  Y = heart_disease[[Y]],
  A = heart_disease[[A]],
  W = heart_disease[W],
  family = "binomial",  # assuming Y is binary
  Q.SL.library = sl_lib,
  g.SL.library = sl_lib
)
```

```
## Growing trees.. Progress: 45%. Estimated remaining time: 1 minute, 36 seconds.
## Growing trees.. Progress: 99%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 26%. Estimated remaining time: 6 minutes, 8 seconds.
## Growing trees.. Progress: 39%. Estimated remaining time: 2 minutes, 36 seconds.
```

```r
summary(tmle_fit)
```

```
##  Initial estimation of Q
##   Procedure: cv-SuperLearner, ensemble
##   Model:
##       Y ~  SL.mean_All + SL.glmnet_All + SL.ranger_All + SL.glm_All + SL.lm_All
##
##   Coefficients:
##       SL.mean_All    0
##     SL.glmnet_All    0
##     SL.ranger_All    0.7058342
##        SL.glm_All    0
##         SL.lm_All    0.2941658
##
##   Cross-validated pseudo R squared :  0.0735
##
##  Estimation of g (treatment mechanism)
##   Procedure: SuperLearner, ensemble
##   Model:
##       A ~  SL.mean_All + SL.glmnet_All + SL.ranger_All + SL.glm_All + SL.lm_All
##
##   Coefficients:
##       SL.mean_All    0
##     SL.glmnet_All    0.8282546
##     SL.ranger_All    0.1717454
##        SL.glm_All    0
##         SL.lm_All    0
```

```
##
##  Estimation of g.Z (intermediate variable assignment mechanism)
##   Procedure: No intermediate variable
##
##  Estimation of g.Delta (missingness mechanism)
##   Procedure: No missingness, ensemble
##
##  Bounds on g: (0.0054, 1)
##
##  Bounds on g for ATT/ATE: (0.0054, 0.9946)
##
##  Additive Effect
##    Parameter Estimate:  -0.35472
##    Estimated Variance:  6.3518e-05
##               p-value:  <2e-16
##     95% Conf Interval:  (-0.37034, -0.3391)
##
##  Additive Effect among the Treated
##    Parameter Estimate:  -0.3212
##    Estimated Variance:  0.00014491
##               p-value:  <2e-16
##     95% Conf Interval:  (-0.34479, -0.29761)
##
##  Additive Effect among the Controls
##    Parameter Estimate:  -0.37031
##    Estimated Variance:  5.6292e-05
##               p-value:  <2e-16
##     95% Conf Interval:  (-0.38501, -0.3556)
##
##  Relative Risk
##    Parameter  Estimate:  0.37419
##    Variance(log scale):  0.00087535
##                p-value:  <2e-16
##      95% Conf Interval:  (0.35311, 0.39653)
##
##  Odds Ratio
##     Parameter  Estimate:  0.20573
##     Variance(log scale):  0.0017347
##                 p-value:  <2e-16
##       95% Conf Interval:  (0.1896, 0.22322)
```

```r
print(paste("Estimated Average Treatment Effect (ATE):", tmle_fit$estimates$ATE))
```

```
## [1] "Estimated Average Treatment Effect (ATE): -0.354719232459076"
## [2] "Estimated Average Treatment Effect (ATE): 6.35178624803227e-05"
## [3] "Estimated Average Treatment Effect (ATE): c(-0.370339771921485, -0.339098692996667)"
## [4] "Estimated Average Treatment Effect (ATE): 0"
## [5] "Estimated Average Treatment Effect (ATE): NA"
## [6] "Estimated Average Treatment Effect (ATE): c(NA, NA)"
## [7] "Estimated Average Treatment Effect (ATE): c(-Inf, NA)"
## [8] "Estimated Average Treatment Effect (ATE): c(NA, Inf)"
```

```r
print(paste("95% Confidence Interval for ATE:", tmle_fit$CI$ATE[1], "to", tmle_fit$CI$ATE[2]))
```

```
## [1] "95% Confidence Interval for ATE:  to "
```

Discussion Questions

1. What is a "double robust" estimator? Why does it provide a guarantee of consistency if either the outcome model or propensity score model is correctly specified? Or in other words, why does mispecifying one of the models not break the analysis? **Hint**: When answering this question, think about how your introductory statistics courses emphasized using theory to determine the correct outcome model, and in this course how we explored the benefits of matching.

Answer: As discussed in lab, double robustness means that if we either: 1. Fit the right model to estimate the expected outcome correctly OR 2. Fit the model to estimate the probability of treatment correctly THEN the final TMLE estimator will be consistent. Consistency means that as the sample size grows to infinity, the bias will drop to 0.

LTMLE Estimation

Now imagine that everything you measured up until now was in "time period 1". Some people either choose not to or otherwise lack access to medication in that time period, but do start taking the medication in time period 2. Imagine we measure covariates like BMI, blood pressure, and cholesterol at that time for everyone in the study (indicated by a "_2" after the covariate name).

Causal Diagram

Update your causal diagram to incorporate this new information. **Note**: If your groups divides up sections and someone is working on LTMLE separately from TMLE then just draw a causal diagram even if it does not match the one you specified above.

Hint: Check out slide 27 from Maya's lecture, or slides 15-17 from Dave's second slide deck in week 8 on matching.

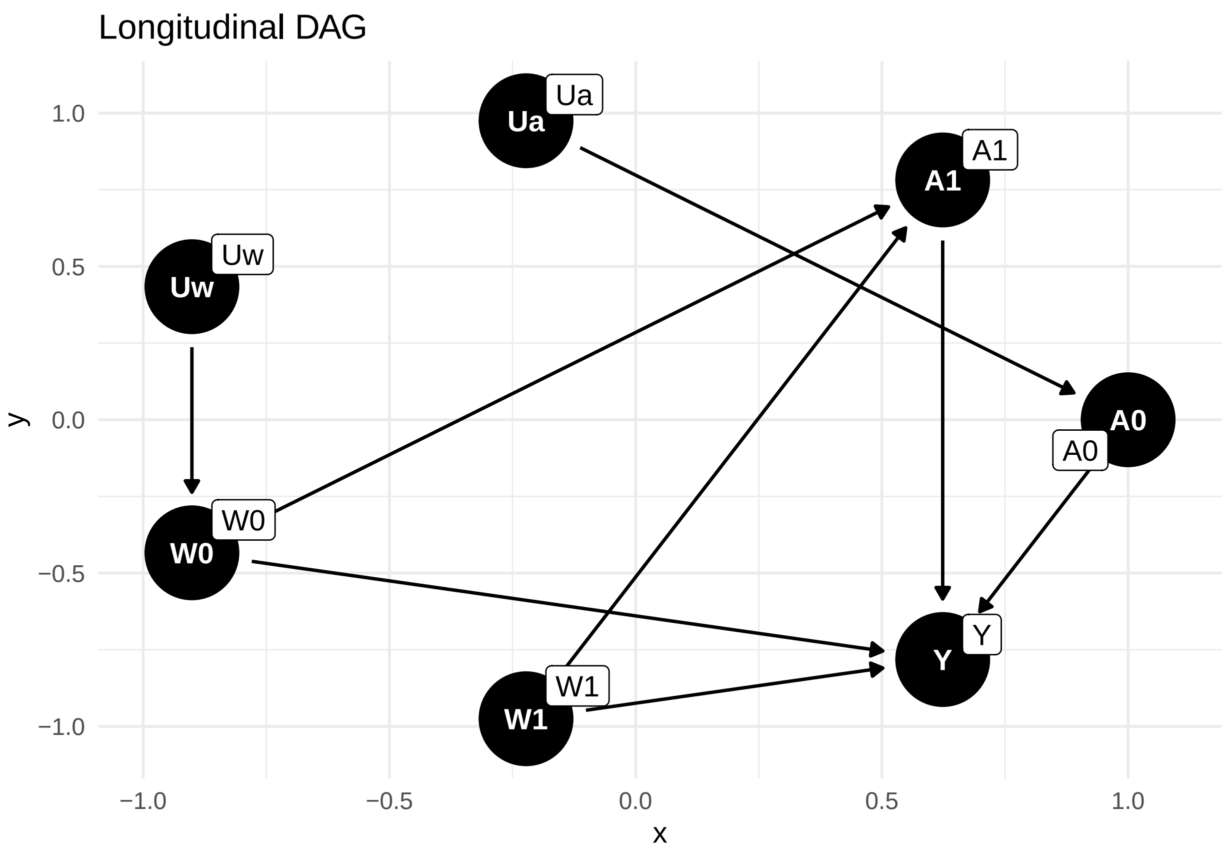
Hint: Keep in mind that any of the variables that end in "_2" are likely affected by both the previous covariates and the first treatment when drawing your DAG.

```r
# DAG for LTMLE

# Define DAG. note i'm sure some code I learned previously in a biostats class
dag <- dagitty::dagitty("
dag {
  Uw [unobserved]
  Ua [unobserved]
  W1-> A1 -> Y
  W1 -> Y
  W0 -> Y
  W0 -> A1
  A0 -> Y
  Uw -> W0
  Ua -> A0
}
")

ggdag <- ggdag::ggdag(dag, text = TRUE, use_labels = "name", layout = "circle") +
  theme_minimal() +
  ggtitle("Longitudinal DAG")

# Print the DAG
print(ggdag)
```



Note: not sequenced very well, but I'm not sure how to change that

LTMLE Estimation

Use the `ltmle` package for this section. First fit a "naive model" that **does not** control for the time-dependent confounding. Then run a LTMLE model that does control for any time dependent confounding. Follow the same steps as in the TMLE section. Do you see a difference between the two estimates?

```
colnames(heart_disease)
```

```
##  [1] "age"                         "sex_at_birth"
##  [3] "simplified_race"             "college_educ"
##  [5] "income_thousands"            "bmi"
##  [7] "blood_pressure"              "chol"
##  [9] "blood_pressure_medication"   "bmi_2"
## [11] "blood_pressure_2"            "chol_2"
## [13] "blood_pressure_medication_2" "mortality"
```

```
#naive model

data_obs <-
  heart_disease %>%
  rename(Y = mortality, A = blood_pressure_medication, W1=age, W2=sex_at_birth, W3=simplified_race, W4=

data_obs_ltmle <-
  data_obs %>%
  select(W1, W2, W3,W4,W5,W6,W7,W8, A, Y)

# implement ltmle
# ----------
result <- ltmle(data_obs_ltmle, # dataset
                Anodes = "A",    # vector that shows treatment
                Ynodes = "Y",    # vector that shows outcome
                abar = 1)
```

```
## Qform not specified, using defaults:

## formula for Y:

## Q.kplus1 ~ W1 + W2 + W3 + W4 + W5 + W6 + W7 + W8 + A

##

## gform not specified, using defaults:

## formula for A:

## A ~ W1 + W2 + W3 + W4 + W5 + W6 + W7 + W8

##

## Estimate of time to completion: < 1 minute
```

```
# view
result
```

```
## Call:
## ltmle(data = data_obs_ltmle, Anodes = "A", Ynodes = "Y", abar = 1)
##
## TMLE Estimate:  0.2041733
```

```
#with time dependent confounding

# Create and rename
data <- heart_disease %>%
  rename(Y = mortality,
         A1 = blood_pressure_medication,
         A2 = blood_pressure_medication_2,
         W1 = age, W2 = sex_at_birth,
         W3 = simplified_race,
         W4 = college_educ,
         W5 = income_thousands,
         W6 = bmi, W7 = blood_pressure,
         W8 = chol,
         L1 = bmi_2,
         L2 = blood_pressure_2,
         L3 = chol_2)

# Check if data_ is a data frame (needed from ltmle code)
print(is.data.frame(data))
```

```
## [1] TRUE
```

```
# kept getting an error, see below...
```

moving to text rather than code

I wrote this code, but keep getting the error below and am not sure how to fix it. . .

#Implement ltmle ltmle(data, Anodes = c("A1", "A2"), # Two treatment variables Lnodes = c("L1", "L2", "L3"), # L indicators Ynodes = "Y", # Outcome abar = c(1, 1), # Treatment indicator in Anodes vector SL.library = sl_lib) # Assuming sl_libs is defined elsewhere

Error in CheckInputs(data, all.nodes, survivalOutcome, Qform, gform, gbounds, : All nodes after the first A/C node must be in A-, C-, L-, or Ynodes

Although I couldn't get the time dependent model to run, I'm sure there would be a difference between the two estimates. How big of a difference depends on how much the covariates chagned beteween time 1 and time 2.

Discussion Questions

1. What sorts of time-dependent confounding should we be especially worried about? For instance, would we be concerned about a running variable for age the same way we might be concerned about blood pressure measured at two different times?

Answer: No, I think blood pressure would be more important. When blood pressure is measured at multiple time points, decisions about medication (treatment) are based on these measurements. Therefore, blood pressure both influences *and* is influenced by treatment. On the other hand, age changes naturally over time and is not influenced by past treatment. Age can be associated with both exposure and outcome but is not typically affected by previous exposures like blood pressure is.