Project 8 Group 6 Annette Gailliot

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

theme_set(theme_dag())
source("pretty_dag.R")
heart_disease <- read_csv(here('heart_disease_tmle.csv'))
```

```
## Rows: 10000 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (14): age, sex_at_birth, simplified_race, college_educ, income_thousands...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

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

```
# set up parallelization
# Parallel backend Mac/Linux
n_cores <- availableCores() - 1

plan(multicore,
     workers = n_cores)
```

```
## Warning in supportsMulticoreAndRStudio(...): [ONE-TIME WARNING] Forked
## processing ('multicore') is not supported when running R from RStudio because
## it is considered unstable. For more details, how to control forked processing
## or not, and how to silence this warning in future R sessions, see
## ?parallelly::supportsMulticore
```

```r
set.seed(44, "L'Ecuyer-CMRG")

## Train/Test split
# initial_split function from tidymodels/rsample
heart_disease_t1 <- heart_disease %>%
  select(-ends_with("_2"))

heart_split <- initial_split(heart_disease_t1, prop = 3/4)

# Declare the training set with rsample::training()
train <- training(heart_split)
# y_train is mortality
y_train <- train %>%
  pull(mortality)

# x_train is everything but the outcome
x_train <- train %>%
  select(-mortality)

# Do the same procedure with the test set
test <- testing(heart_split)

y_test <- test %>%
  pull(mortality)

x_test <- test %>%
  select(-mortality)

## sl lib
## What models do we have?
listWrappers()
```

```
## All prediction algorithm wrappers in SuperLearner:
```

```
##  [1] "SL.bartMachine"      "SL.bayesglm"         "SL.biglasso"
##  [4] "SL.caret"            "SL.caret.rpart"      "SL.cforest"
##  [7] "SL.earth"            "SL.extraTrees"       "SL.gam"
## [10] "SL.gbm"              "SL.glm"              "SL.glm.interaction"
## [13] "SL.glmnet"           "SL.ipredbagg"        "SL.kernelKnn"
## [16] "SL.knn"              "SL.ksvm"             "SL.lda"
## [19] "SL.leekasso"         "SL.lm"               "SL.loess"
## [22] "SL.logreg"           "SL.mean"             "SL.nnet"
## [25] "SL.nnls"             "SL.polymars"         "SL.qda"
## [28] "SL.randomForest"     "SL.ranger"           "SL.ridge"
## [31] "SL.rpart"            "SL.rpartPrune"       "SL.speedglm"
## [34] "SL.speedlm"          "SL.step"             "SL.step.forward"
## [37] "SL.step.interaction" "SL.stepAIC"          "SL.svm"
## [40] "SL.template"         "SL.xgboost"
```

```
##
## All screening algorithm wrappers in SuperLearner:
```

```
## [1] "All"
## [1] "screen.corP"           "screen.corRank"        "screen.glmnet"
## [4] "screen.randomForest"   "screen.SIS"            "screen.template"
```

```
## [7] "screen.ttest"          "write.screen.template"
```

```
# Fit SuperLearner Model
cv_sl <- mcSuperLearner(Y = y_train,
                        X = x_train,
                        family = binomial(),
                        SL.library = c("SL.glmnet", "SL.mean", "SL.ranger", "SL.knn", "SL.svm"))
```

```
## Loading required namespace: e1071
```

```
# SL.glmnet", "SL.mean", "SL.ranger", SL.svm", "SL.lda" , "SL.mean", "SL.ranger"

## Train SuperLearner
## Risk and Coefficient of each model
cv_sl
```

```
##
## Call:
## mcSuperLearner(Y = y_train, X = x_train, family = binomial(), SL.library = c("SL.glmnet",
##     "SL.mean", "SL.ranger", "SL.knn", "SL.svm"))
##
##
##                     Risk      Coef
## SL.glmnet_All 0.2348294 0.2952888
## SL.mean_All   0.2496221 0.0000000
## SL.ranger_All 0.2318644 0.5019503
## SL.knn_All    0.2721006 0.0000000
## SL.svm_All    0.2364607 0.2027609
```

```
## Discrete winner and superlearner ensemble performance
preds <- predict(cv_sl,
                 x_test,
                 onlySL = TRUE)
```

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

```
# start with y_test
validation <- y_test %>%
  # add our predictions
  bind_cols(preds$pred[,1]) %>%
  # rename columns
  rename(obs = `...1`,
         pred = `...2`) %>%
  mutate(pred = ifelse(pred >= .5,
                           1,
                           0))
```

```
## New names:
## * `` -> `...1`
## * `` -> `...2`
```

```
head(validation)
```

```
## # A tibble: 6 x 2
##     obs  pred
##   <dbl> <dbl>
## 1     1     1
## 2     1     1
```

```
## 3     0     1
## 4     1     0
## 5     0     1
## 6     0     0
```

```
## Confusion Matrix
caret::confusionMatrix(as.factor(validation$pred),
                       as.factor(validation$obs))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0  390  218
##          1  844 1048
##
##                Accuracy : 0.5752
##                  95% CI : (0.5555, 0.5947)
##     No Information Rate : 0.5064
##     P-Value [Acc > NIR] : 3.096e-12
##
##                   Kappa : 0.1448
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.3160
##             Specificity : 0.8278
##          Pos Pred Value : 0.6414
##          Neg Pred Value : 0.5539
##              Prevalence : 0.4936
##          Detection Rate : 0.1560
##    Detection Prevalence : 0.2432
##       Balanced Accuracy : 0.5719
##
##        'Positive' Class : 0
##
```

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: The SuperLearner ensemble is a method that combines multiple algorithms to make more accurate predictions than any single algorithm alone. The SuperLearner algorithm uses cross-validation to select the best combination of algorithms and their weights for a particular dataset. Doing this increases stability, improves performance, allows for more flexibility, and improves robustness.

Targeted Maximum Likelihood Estimation

Causal Diagram

TMLE requires estimating two models:

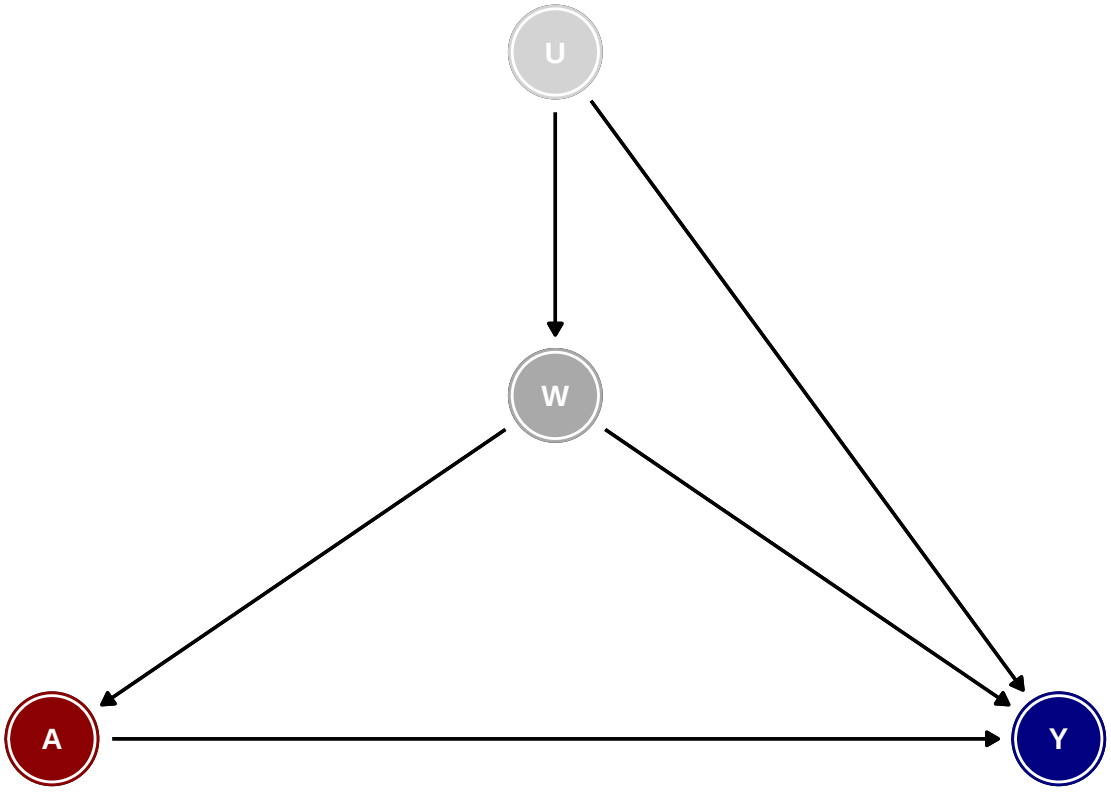
1. The outcome model, or the relationship between the outcome and the treatment/predictors, $P(Y|(A, W)$.

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. **Answer**: Y is the outcome variable, mortality. A is the treatment of blood pressure medication. W is the set of observed confounders including demographics (age, sex at birth, racial characteristics, college eduction, and income) and health variables (bmi, blood pressure, and cholesterol).

```
# DAG for TMLE

dagify(Y ~ A,
       Y ~ U,
       Y ~ W,
       W ~ U,
       A ~ W,
       exposure = "A",
       outcome = "Y") %>%
  tidy_dagitty() %>%
  pretty_dag() %>%
  ggdag() +
  geom_dag_edges() +
  geom_dag_node(aes(color = color)) +
  geom_dag_text(col = "white") +
  theme(legend.position = "none") +
  scale_color_manual(values=c("darkred", "lightgrey", "darkgrey", "navy"))
```



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

```
# TMLE SuperLearner Library same as before
sl_libs <- c("SL.glmnet", "SL.mean", "SL.ranger", "SL.knn", "SL.svm")


# prep data
#Q <- cv_sl

Y <- heart_disease_t1 %>%
  pull(mortality)

W <- heart_disease_t1 %>%
  select(-mortality) %>%
  select(-blood_pressure_medication)

A <- heart_disease_t1 %>%
  pull(blood_pressure_medication)


# Same outcome and propensity model from DAG.
tmle_fit <-
  tmle::tmle(Y = Y,
             A = A,
             W = W,
             Q.SL.library = sl_libs,
             g.SL.library = sl_libs)
```

```
## Error in (function (Y, X, newX, family, k = 10, ...)  :
##   SL.knn only available for family = binomial()
## Error in (function (Y, X, newX, family, k = 10, ...)  :
##   SL.knn only available for family = binomial()
## Error in (function (Y, X, newX, family, k = 10, ...)  :
##   SL.knn only available for family = binomial()
## Error in (function (Y, X, newX, family, k = 10, ...)  :
##   SL.knn only available for family = binomial()
## Error in (function (Y, X, newX, family, k = 10, ...)  :
##   SL.knn only available for family = binomial()
## Error in (function (Y, X, newX, family, k = 10, ...)  :
##   SL.knn only available for family = binomial()
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
```

```
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
```

```
tmle_fit
```

```
##  Additive Effect
##    Parameter Estimate:  -0.35466
##    Estimated Variance:  7.1114e-05
##               p-value:  <2e-16
##     95% Conf Interval: (-0.37118, -0.33813)
##
##  Additive Effect among the Treated
##    Parameter Estimate:  -0.32044
##    Estimated Variance:  0.00014646
##               p-value:  <2e-16
##     95% Conf Interval: (-0.34416, -0.29672)
##
##  Additive Effect among the Controls
##    Parameter Estimate:  -0.36972
##    Estimated Variance:  6.3806e-05
##               p-value:  <2e-16
##     95% Conf Interval: (-0.38538, -0.35407)
```

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 A “double robust” estimator is a statistical method used to estimate a treatment effect in an observational study. The term “double robust” refers to the fact that the estimator is designed to be robust to misspecification of either the outcome model or the propensity score model, as long as at least one of these models is correctly specified. A double robust estimator combines these two models in a way that allows for flexibility in either one of them. Specifically, the estimator uses a weighted combination of the outcome and propensity score models to estimate the treatment effect. This weighting ensures that the estimator is doubly robust, meaning that it is consistent if either the outcome model or the propensity score model is correctly specified, even if the other is misspecified. To see why this is the case, consider a scenario where the outcome model is correctly specified, but the propensity score model is misspecified. In this case, the matched sample may not be well balanced, but the outcome model can still provide an unbiased estimate of the treatment effect for the matched sample. The double robust estimator takes advantage of this by using the outcome model to weight the propensity score model, which helps to adjust for any remaining imbalances in the covariates. Similarly, if the outcome model is misspecified but the propensity score model is correctly specified, the double robust estimator can still provide a consistent estimate of the treatment effect by using the propensity score model to weight the outcome model.

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.

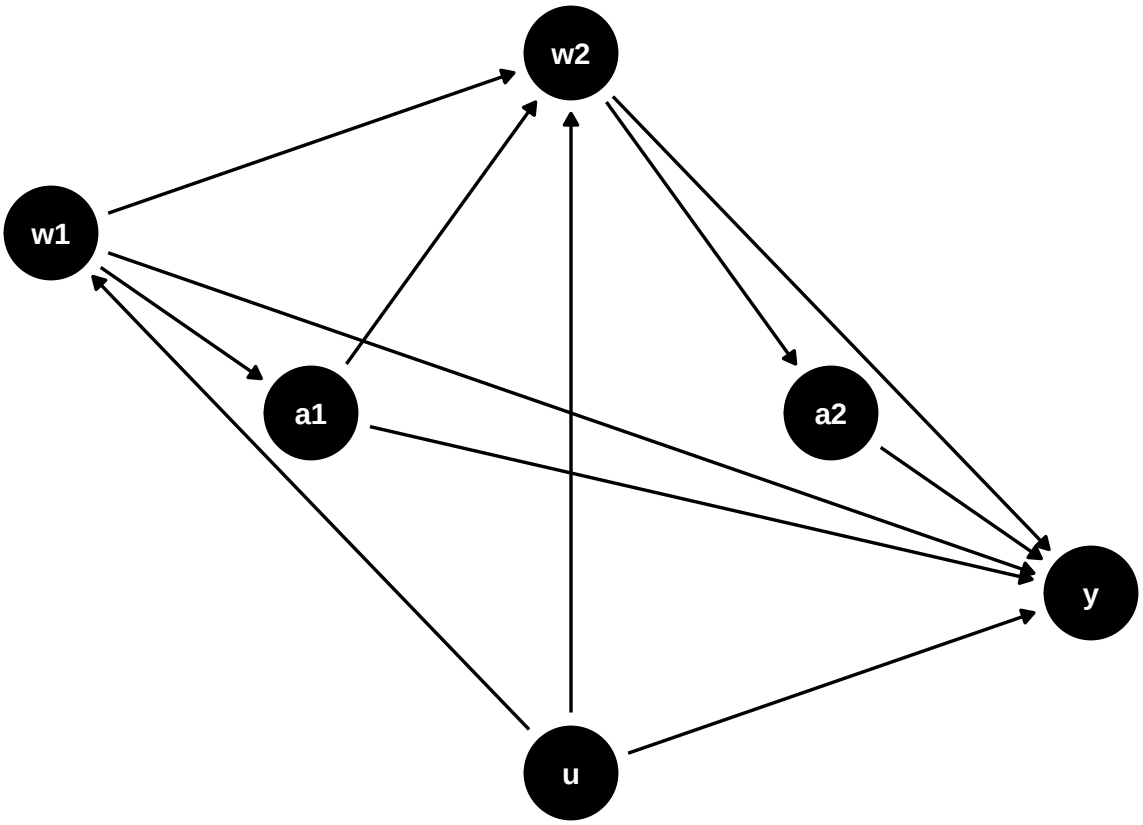
Hint: Keep in mind that any of the variables that end in "_2" are likely affected by both the previous covariates and the first treatment when drawing your DAG.

```
# DAG for TMLE

coord_dag <- list(
  x = c(w1 = 0, a1 = 1, w2 = 2, u = 2, a2 = 3, y= 4),
  y = c(w1 = 3, a1 = 2, w2 = 4, u = 0, a2 = 2, y= 1)
)

our_dag <- ggdag::dagify(y ~ a2 + u + w2 + a1 + w1,
                                a2 ~ w2,
                                w2 ~ u + a1 + w1,
                                a1 ~ w1,
                                w1 ~ u,
                         coords = coord_dag)

ggdag::ggdag(our_dag) + theme_void()
```



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?

```
## Naive Model (no time-dependent confounding) estimate

Y <- heart_disease_t1 %>%
  pull(mortality)

L <- heart_disease_t1 %>%
  select(-mortality) %>%
  select(-blood_pressure_medication)

A1 <- heart_disease_t1 %>%
  pull(blood_pressure_medication)

## LTMLE estimate
tmle_fit <-
  tmle::tmle(Y = Y,
             A = A1,
             W = L,
             Q.SL.library = sl_libs,
             g.SL.library = sl_libs)
```

```
## Error in (function (Y, X, newX, family, k = 10, ...)  :
##   SL.knn only available for family = binomial()
```

```
## Error in (function (Y, X, newX, family, k = 10, ...)  :
##   SL.knn only available for family = binomial()
## Error in (function (Y, X, newX, family, k = 10, ...)  :
##   SL.knn only available for family = binomial()
## Error in (function (Y, X, newX, family, k = 10, ...)  :
##   SL.knn only available for family = binomial()
## Error in (function (Y, X, newX, family, k = 10, ...)  :
##   SL.knn only available for family = binomial()
## Error in (function (Y, X, newX, family, k = 10, ...)  :
##   SL.knn only available for family = binomial()
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
```

```
tmle_fit
```

```
##  Additive Effect
##    Parameter Estimate:  -0.35382
##    Estimated Variance:  7.0442e-05
##               p-value:  <2e-16
##     95% Conf Interval: (-0.37027, -0.33737)
##
##  Additive Effect among the Treated
##    Parameter Estimate:  -0.32011
##    Estimated Variance:  0.00014549
##               p-value:  <2e-16
##     95% Conf Interval: (-0.34375, -0.29647)
##
##  Additive Effect among the Controls
##    Parameter Estimate:  -0.36887
##    Estimated Variance:  6.2925e-05
##               p-value:  <2e-16
##     95% Conf Interval: (-0.38442, -0.35332)
```

```
## LTMLE Model ( time-dependent confounding) estimate

Y <- heart_disease %>%
  pull(mortality)

L <- heart_disease %>%
  select(bmi, bmi_2, blood_pressure, blood_pressure_2, chol, chol_2)

A1 <- heart_disease %>%
  pull(blood_pressure_medication)

A2 <- heart_disease %>%
```

```
  pull(blood_pressure_medication_2)

W <- heart_disease %>%
  select(age, sex_at_birth, simplified_race, college_educ, income_thousands)

data <- data.frame(W, A1, L, A2, Y)

## LTMLE estimate
ltmle(data, Anodes=c("A1", "A2"), Lnodes=c("bmi", "bmi_2", "blood_pressure", "blood_pressure_2", "chol"
```

```
## Qform not specified, using defaults:
## formula for bmi:
## Q.kplus1 ~ age + sex_at_birth + simplified_race + college_educ +     income_thousands + A1
## formula for Y:
## Q.kplus1 ~ age + sex_at_birth + simplified_race + college_educ +     income_thousands + A1 + bmi + b
##
## gform not specified, using defaults:
## formula for A1:
## A1 ~ age + sex_at_birth + simplified_race + college_educ + income_thousands
## formula for A2:
## A2 ~ age + sex_at_birth + simplified_race + college_educ + income_thousands +     A1 + bmi + bmi_2 +
##
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
```

```
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!

## Estimate of time to completion: 4 to 19 minutes

## Warning in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class, :
## Variable(s) 'Xx.1' constant. Cannot scale data.

## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
```

```
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm  on full data
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in SuperLearner::SuperLearner(Y = Y.subset, X = X.subset, SL.library =
## SL.library, : Coefficients already 0 for all failed algorithm(s)

## Warning in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class, :
## Variable(s) 'Xx.1' constant. Cannot scale data.

## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)
```

```
## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!
```

```
## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class,  :
##   specified nu is infeasible!

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm  on full data
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in SuperLearner::SuperLearner(Y = Y.subset, X = X.subset, SL.library =
## SL.library, : Coefficients already 0 for all failed algorithm(s)

## Warning in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class, :
## Variable(s) 'Xx.1' constant. Cannot scale data.

## Warning in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class, :
## Variable(s) 'Xx.1' constant. Cannot scale data.

## Warning in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class, :
## Variable(s) 'Xx.1' constant. Cannot scale data.

## Warning in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class, :
## Variable(s) 'Xx.1' constant. Cannot scale data.

## Warning in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class, :
## Variable(s) 'Xx.1' constant. Cannot scale data.

## Warning in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class, :
## Variable(s) 'Xx.1' constant. Cannot scale data.

## Warning in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class, :
## Variable(s) 'Xx.1' constant. Cannot scale data.

## Warning in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class, :
## Variable(s) 'Xx.1' constant. Cannot scale data.

## Warning in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class, :
## Variable(s) 'Xx.1' constant. Cannot scale data.

## Warning in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class, :
## Variable(s) 'Xx.1' constant. Cannot scale data.

## Warning in svm.default(y = as.factor(Y), x = X, nu = nu, type = type.class, :
## Variable(s) 'Xx.1' constant. Cannot scale data.

## Error in lognet(xd, is.sparse, ix, jx, y, weights, offset, alpha, nobs,  :
##   one multinomial or binomial class has 1 or 0 observations; not allowed
```

```
## Warning in FUN(X[[i]], ...): Error in algorithm SL.glmnet
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Growing trees.. Progress: 3%. Estimated remaining time: 14 minutes, 40 seconds.
## Growing trees.. Progress: 6%. Estimated remaining time: 15 minutes, 6 seconds.
## Growing trees.. Progress: 10%. Estimated remaining time: 14 minutes, 15 seconds.
## Growing trees.. Progress: 13%. Estimated remaining time: 14 minutes, 9 seconds.
## Growing trees.. Progress: 16%. Estimated remaining time: 13 minutes, 25 seconds.
## Growing trees.. Progress: 19%. Estimated remaining time: 13 minutes, 17 seconds.
## Growing trees.. Progress: 23%. Estimated remaining time: 12 minutes, 35 seconds.
## Growing trees.. Progress: 26%. Estimated remaining time: 12 minutes, 13 seconds.
## Growing trees.. Progress: 29%. Estimated remaining time: 11 minutes, 47 seconds.
## Growing trees.. Progress: 33%. Estimated remaining time: 10 minutes, 53 seconds.
## Growing trees.. Progress: 36%. Estimated remaining time: 10 minutes, 15 seconds.
## Growing trees.. Progress: 39%. Estimated remaining time: 9 minutes, 58 seconds.
## Growing trees.. Progress: 43%. Estimated remaining time: 9 minutes, 5 seconds.
## Growing trees.. Progress: 47%. Estimated remaining time: 8 minutes, 30 seconds.
## Growing trees.. Progress: 51%. Estimated remaining time: 7 minutes, 44 seconds.
## Growing trees.. Progress: 55%. Estimated remaining time: 7 minutes, 4 seconds.
## Growing trees.. Progress: 58%. Estimated remaining time: 6 minutes, 36 seconds.
## Growing trees.. Progress: 62%. Estimated remaining time: 5 minutes, 58 seconds.
## Growing trees.. Progress: 65%. Estimated remaining time: 5 minutes, 30 seconds.
## Growing trees.. Progress: 68%. Estimated remaining time: 5 minutes, 3 seconds.
## Growing trees.. Progress: 72%. Estimated remaining time: 4 minutes, 23 seconds.
## Growing trees.. Progress: 75%. Estimated remaining time: 3 minutes, 52 seconds.
## Growing trees.. Progress: 79%. Estimated remaining time: 3 minutes, 18 seconds.
## Growing trees.. Progress: 82%. Estimated remaining time: 2 minutes, 46 seconds.
## Growing trees.. Progress: 86%. Estimated remaining time: 2 minutes, 14 seconds.
## Growing trees.. Progress: 89%. Estimated remaining time: 1 minute, 44 seconds.
## Growing trees.. Progress: 91%. Estimated remaining time: 1 minute, 21 seconds.
## Growing trees.. Progress: 95%. Estimated remaining time: 49 seconds.
## Growing trees.. Progress: 98%. Estimated remaining time: 18 seconds.
## Error in pred[, "1"] : subscript out of bounds

## Warning in FUN(X[[i]], ...): Error in algorithm SL.ranger
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)


## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error : vector memory exhausted (limit reached?)

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Error in lognet(xd, is.sparse, ix, jx, y, weights, offset, alpha, nobs,  :
##   one multinomial or binomial class has 1 or 0 observations; not allowed

## Warning in FUN(X[[i]], ...): Error in algorithm SL.glmnet
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Growing trees.. Progress: 3%. Estimated remaining time: 15 minutes, 37 seconds.
## Growing trees.. Progress: 7%. Estimated remaining time: 13 minutes, 43 seconds.
## Growing trees.. Progress: 11%. Estimated remaining time: 12 minutes, 40 seconds.
## Growing trees.. Progress: 14%. Estimated remaining time: 12 minutes, 34 seconds.
## Growing trees.. Progress: 18%. Estimated remaining time: 12 minutes, 24 seconds.
## Growing trees.. Progress: 21%. Estimated remaining time: 12 minutes, 7 seconds.
## Growing trees.. Progress: 25%. Estimated remaining time: 11 minutes, 20 seconds.
```

```
## Growing trees.. Progress: 28%. Estimated remaining time: 10 minutes, 51 seconds.
## Growing trees.. Progress: 32%. Estimated remaining time: 10 minutes, 21 seconds.
## Growing trees.. Progress: 35%. Estimated remaining time: 9 minutes, 58 seconds.
## Growing trees.. Progress: 39%. Estimated remaining time: 9 minutes, 19 seconds.
## Growing trees.. Progress: 42%. Estimated remaining time: 8 minutes, 44 seconds.
## Growing trees.. Progress: 46%. Estimated remaining time: 8 minutes, 14 seconds.
## Growing trees.. Progress: 49%. Estimated remaining time: 7 minutes, 43 seconds.
## Growing trees.. Progress: 53%. Estimated remaining time: 7 minutes, 9 seconds.
## Growing trees.. Progress: 56%. Estimated remaining time: 6 minutes, 36 seconds.
## Growing trees.. Progress: 60%. Estimated remaining time: 6 minutes, 9 seconds.
## Growing trees.. Progress: 63%. Estimated remaining time: 5 minutes, 39 seconds.
## Growing trees.. Progress: 66%. Estimated remaining time: 5 minutes, 8 seconds.
## Growing trees.. Progress: 70%. Estimated remaining time: 4 minutes, 29 seconds.
## Growing trees.. Progress: 74%. Estimated remaining time: 3 minutes, 59 seconds.
## Growing trees.. Progress: 77%. Estimated remaining time: 3 minutes, 29 seconds.
## Growing trees.. Progress: 81%. Estimated remaining time: 2 minutes, 55 seconds.
## Growing trees.. Progress: 84%. Estimated remaining time: 2 minutes, 31 seconds.
## Growing trees.. Progress: 87%. Estimated remaining time: 1 minute, 58 seconds.
## Growing trees.. Progress: 90%. Estimated remaining time: 1 minute, 30 seconds.
## Growing trees.. Progress: 93%. Estimated remaining time: 1 minute, 1 seconds.
## Growing trees.. Progress: 97%. Estimated remaining time: 29 seconds.
## Error in pred[, "1"] : subscript out of bounds

## Warning in FUN(X[[i]], ...): Error in algorithm SL.ranger
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error : vector memory exhausted (limit reached?)

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Error in lognet(xd, is.sparse, ix, jx, y, weights, offset, alpha, nobs,  :
##   one multinomial or binomial class has 1 or 0 observations; not allowed

## Warning in FUN(X[[i]], ...): Error in algorithm SL.glmnet
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Growing trees.. Progress: 3%. Estimated remaining time: 16 minutes, 42 seconds.
## Growing trees.. Progress: 7%. Estimated remaining time: 14 minutes, 51 seconds.
## Growing trees.. Progress: 10%. Estimated remaining time: 14 minutes, 52 seconds.
## Growing trees.. Progress: 13%. Estimated remaining time: 14 minutes, 8 seconds.
## Growing trees.. Progress: 17%. Estimated remaining time: 13 minutes, 1 seconds.
## Growing trees.. Progress: 21%. Estimated remaining time: 12 minutes, 27 seconds.
## Growing trees.. Progress: 24%. Estimated remaining time: 11 minutes, 55 seconds.
## Growing trees.. Progress: 27%. Estimated remaining time: 11 minutes, 33 seconds.
## Growing trees.. Progress: 30%. Estimated remaining time: 11 minutes, 6 seconds.
## Growing trees.. Progress: 34%. Estimated remaining time: 10 minutes, 38 seconds.
## Growing trees.. Progress: 37%. Estimated remaining time: 9 minutes, 57 seconds.
## Growing trees.. Progress: 40%. Estimated remaining time: 9 minutes, 35 seconds.
## Growing trees.. Progress: 44%. Estimated remaining time: 8 minutes, 57 seconds.
## Growing trees.. Progress: 47%. Estimated remaining time: 8 minutes, 24 seconds.
## Growing trees.. Progress: 50%. Estimated remaining time: 7 minutes, 59 seconds.
## Growing trees.. Progress: 53%. Estimated remaining time: 7 minutes, 33 seconds.
## Growing trees.. Progress: 57%. Estimated remaining time: 6 minutes, 59 seconds.
## Growing trees.. Progress: 60%. Estimated remaining time: 6 minutes, 25 seconds.
```

```
## Growing trees.. Progress: 63%. Estimated remaining time: 5 minutes, 53 seconds.
## Growing trees.. Progress: 67%. Estimated remaining time: 5 minutes, 22 seconds.
## Growing trees.. Progress: 70%. Estimated remaining time: 4 minutes, 43 seconds.
## Growing trees.. Progress: 74%. Estimated remaining time: 4 minutes, 8 seconds.
## Growing trees.. Progress: 78%. Estimated remaining time: 3 minutes, 33 seconds.
## Growing trees.. Progress: 81%. Estimated remaining time: 3 minutes, 5 seconds.
## Growing trees.. Progress: 84%. Estimated remaining time: 2 minutes, 28 seconds.
## Growing trees.. Progress: 87%. Estimated remaining time: 2 minutes, 0 seconds.
## Growing trees.. Progress: 91%. Estimated remaining time: 1 minute, 27 seconds.
## Growing trees.. Progress: 94%. Estimated remaining time: 57 seconds.
## Growing trees.. Progress: 98%. Estimated remaining time: 22 seconds.
## Error in pred[, "1"] : subscript out of bounds

## Warning in FUN(X[[i]], ...): Error in algorithm SL.ranger
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error : vector memory exhausted (limit reached?)

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Error in lognet(xd, is.sparse, ix, jx, y, weights, offset, alpha, nobs,  :
##   one multinomial or binomial class has 1 or 0 observations; not allowed

## Warning in FUN(X[[i]], ...): Error in algorithm SL.glmnet
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Growing trees.. Progress: 4%. Estimated remaining time: 13 minutes, 50 seconds.
## Growing trees.. Progress: 7%. Estimated remaining time: 13 minutes, 56 seconds.
## Growing trees.. Progress: 10%. Estimated remaining time: 13 minutes, 56 seconds.
## Growing trees.. Progress: 14%. Estimated remaining time: 13 minutes, 13 seconds.
## Growing trees.. Progress: 17%. Estimated remaining time: 12 minutes, 45 seconds.
## Growing trees.. Progress: 20%. Estimated remaining time: 12 minutes, 38 seconds.
## Growing trees.. Progress: 24%. Estimated remaining time: 11 minutes, 53 seconds.
## Growing trees.. Progress: 27%. Estimated remaining time: 11 minutes, 15 seconds.
## Growing trees.. Progress: 31%. Estimated remaining time: 10 minutes, 44 seconds.
## Growing trees.. Progress: 35%. Estimated remaining time: 9 minutes, 57 seconds.
## Growing trees.. Progress: 39%. Estimated remaining time: 9 minutes, 19 seconds.
## Growing trees.. Progress: 42%. Estimated remaining time: 8 minutes, 54 seconds.
## Growing trees.. Progress: 46%. Estimated remaining time: 8 minutes, 11 seconds.
## Growing trees.. Progress: 49%. Estimated remaining time: 7 minutes, 38 seconds.
## Growing trees.. Progress: 53%. Estimated remaining time: 7 minutes, 8 seconds.
## Growing trees.. Progress: 56%. Estimated remaining time: 6 minutes, 43 seconds.
## Growing trees.. Progress: 59%. Estimated remaining time: 6 minutes, 10 seconds.
## Growing trees.. Progress: 63%. Estimated remaining time: 5 minutes, 38 seconds.
## Growing trees.. Progress: 67%. Estimated remaining time: 5 minutes, 4 seconds.
## Growing trees.. Progress: 70%. Estimated remaining time: 4 minutes, 36 seconds.
## Growing trees.. Progress: 73%. Estimated remaining time: 4 minutes, 5 seconds.
## Growing trees.. Progress: 77%. Estimated remaining time: 3 minutes, 32 seconds.
## Growing trees.. Progress: 80%. Estimated remaining time: 2 minutes, 59 seconds.
## Growing trees.. Progress: 83%. Estimated remaining time: 2 minutes, 35 seconds.
## Growing trees.. Progress: 87%. Estimated remaining time: 2 minutes, 3 seconds.
## Growing trees.. Progress: 90%. Estimated remaining time: 1 minute, 34 seconds.
## Growing trees.. Progress: 94%. Estimated remaining time: 59 seconds.
## Growing trees.. Progress: 97%. Estimated remaining time: 27 seconds.
```

```
## Error in pred[, "1"] : subscript out of bounds

## Warning in FUN(X[[i]], ...): Error in algorithm SL.ranger
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error : vector memory exhausted (limit reached?)

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Error in lognet(xd, is.sparse, ix, jx, y, weights, offset, alpha, nobs,  :
##   one multinomial or binomial class has 1 or 0 observations; not allowed

## Warning in FUN(X[[i]], ...): Error in algorithm SL.glmnet
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Growing trees.. Progress: 4%. Estimated remaining time: 13 minutes, 4 seconds.
## Growing trees.. Progress: 7%. Estimated remaining time: 13 minutes, 32 seconds.
## Growing trees.. Progress: 10%. Estimated remaining time: 13 minutes, 56 seconds.
## Growing trees.. Progress: 14%. Estimated remaining time: 12 minutes, 53 seconds.
## Growing trees.. Progress: 17%. Estimated remaining time: 12 minutes, 34 seconds.
## Growing trees.. Progress: 21%. Estimated remaining time: 12 minutes, 20 seconds.
## Growing trees.. Progress: 24%. Estimated remaining time: 11 minutes, 47 seconds.
## Growing trees.. Progress: 28%. Estimated remaining time: 11 minutes, 3 seconds.
## Growing trees.. Progress: 32%. Estimated remaining time: 10 minutes, 21 seconds.
## Growing trees.. Progress: 35%. Estimated remaining time: 9 minutes, 51 seconds.
## Growing trees.. Progress: 39%. Estimated remaining time: 9 minutes, 16 seconds.
## Growing trees.. Progress: 42%. Estimated remaining time: 8 minutes, 52 seconds.
## Growing trees.. Progress: 46%. Estimated remaining time: 8 minutes, 13 seconds.
## Growing trees.. Progress: 49%. Estimated remaining time: 7 minutes, 46 seconds.
## Growing trees.. Progress: 53%. Estimated remaining time: 7 minutes, 5 seconds.
## Growing trees.. Progress: 57%. Estimated remaining time: 6 minutes, 35 seconds.
## Growing trees.. Progress: 61%. Estimated remaining time: 5 minutes, 56 seconds.
## Growing trees.. Progress: 65%. Estimated remaining time: 5 minutes, 18 seconds.
## Growing trees.. Progress: 68%. Estimated remaining time: 4 minutes, 45 seconds.
## Growing trees.. Progress: 72%. Estimated remaining time: 4 minutes, 7 seconds.
## Growing trees.. Progress: 76%. Estimated remaining time: 3 minutes, 31 seconds.
## Growing trees.. Progress: 80%. Estimated remaining time: 2 minutes, 54 seconds.
## Growing trees.. Progress: 84%. Estimated remaining time: 2 minutes, 20 seconds.
## Growing trees.. Progress: 88%. Estimated remaining time: 1 minute, 46 seconds.
## Growing trees.. Progress: 91%. Estimated remaining time: 1 minute, 15 seconds.
## Growing trees.. Progress: 94%. Estimated remaining time: 49 seconds.
## Growing trees.. Progress: 98%. Estimated remaining time: 15 seconds.
## Error in pred[, "1"] : subscript out of bounds

## Warning in FUN(X[[i]], ...): Error in algorithm SL.ranger
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error : vector memory exhausted (limit reached?)

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Error in lognet(xd, is.sparse, ix, jx, y, weights, offset, alpha, nobs,  :
##   one multinomial or binomial class has 1 or 0 observations; not allowed
```

```
## Warning in FUN(X[[i]], ...): Error in algorithm SL.glmnet
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Growing trees.. Progress: 3%. Estimated remaining time: 14 minutes, 40 seconds.
## Growing trees.. Progress: 8%. Estimated remaining time: 12 minutes, 58 seconds.
## Growing trees.. Progress: 11%. Estimated remaining time: 12 minutes, 33 seconds.
## Growing trees.. Progress: 15%. Estimated remaining time: 12 minutes, 22 seconds.
## Growing trees.. Progress: 18%. Estimated remaining time: 12 minutes, 14 seconds.
## Growing trees.. Progress: 21%. Estimated remaining time: 11 minutes, 46 seconds.
## Growing trees.. Progress: 25%. Estimated remaining time: 11 minutes, 6 seconds.
## Growing trees.. Progress: 29%. Estimated remaining time: 10 minutes, 21 seconds.
## Growing trees.. Progress: 33%. Estimated remaining time: 9 minutes, 38 seconds.
## Growing trees.. Progress: 37%. Estimated remaining time: 9 minutes, 9 seconds.
## Growing trees.. Progress: 40%. Estimated remaining time: 8 minutes, 37 seconds.
## Growing trees.. Progress: 44%. Estimated remaining time: 8 minutes, 12 seconds.
## Growing trees.. Progress: 47%. Estimated remaining time: 7 minutes, 37 seconds.
## Growing trees.. Progress: 52%. Estimated remaining time: 6 minutes, 56 seconds.
## Growing trees.. Progress: 55%. Estimated remaining time: 6 minutes, 28 seconds.
## Growing trees.. Progress: 58%. Estimated remaining time: 6 minutes, 0 seconds.
## Growing trees.. Progress: 62%. Estimated remaining time: 5 minutes, 29 seconds.
## Growing trees.. Progress: 66%. Estimated remaining time: 4 minutes, 56 seconds.
## Growing trees.. Progress: 69%. Estimated remaining time: 4 minutes, 25 seconds.
## Growing trees.. Progress: 73%. Estimated remaining time: 3 minutes, 54 seconds.
## Growing trees.. Progress: 77%. Estimated remaining time: 3 minutes, 19 seconds.
## Growing trees.. Progress: 81%. Estimated remaining time: 2 minutes, 48 seconds.
## Growing trees.. Progress: 84%. Estimated remaining time: 2 minutes, 15 seconds.
## Growing trees.. Progress: 88%. Estimated remaining time: 1 minute, 46 seconds.
## Growing trees.. Progress: 91%. Estimated remaining time: 1 minute, 16 seconds.
## Growing trees.. Progress: 95%. Estimated remaining time: 41 seconds.
## Growing trees.. Progress: 99%. Estimated remaining time: 12 seconds.
## Error in pred[, "1"] : subscript out of bounds

## Warning in FUN(X[[i]], ...): Error in algorithm SL.ranger
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error : vector memory exhausted (limit reached?)

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Error in lognet(xd, is.sparse, ix, jx, y, weights, offset, alpha, nobs,  :
##   one multinomial or binomial class has 1 or 0 observations; not allowed

## Warning in FUN(X[[i]], ...): Error in algorithm SL.glmnet
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Growing trees.. Progress: 4%. Estimated remaining time: 13 minutes, 50 seconds.
## Growing trees.. Progress: 7%. Estimated remaining time: 13 minutes, 32 seconds.
## Growing trees.. Progress: 10%. Estimated remaining time: 14 minutes, 33 seconds.
## Growing trees.. Progress: 13%. Estimated remaining time: 14 minutes, 0 seconds.
## Growing trees.. Progress: 18%. Estimated remaining time: 12 minutes, 23 seconds.
## Growing trees.. Progress: 22%. Estimated remaining time: 11 minutes, 32 seconds.
## Growing trees.. Progress: 25%. Estimated remaining time: 11 minutes, 4 seconds.
## Growing trees.. Progress: 29%. Estimated remaining time: 10 minutes, 39 seconds.
## Growing trees.. Progress: 33%. Estimated remaining time: 9 minutes, 55 seconds.
```

```
## Growing trees.. Progress: 36%. Estimated remaining time: 9 minutes, 28 seconds.
## Growing trees.. Progress: 39%. Estimated remaining time: 9 minutes, 1 seconds.
## Growing trees.. Progress: 43%. Estimated remaining time: 8 minutes, 33 seconds.
## Growing trees.. Progress: 47%. Estimated remaining time: 7 minutes, 57 seconds.
## Growing trees.. Progress: 50%. Estimated remaining time: 7 minutes, 25 seconds.
## Growing trees.. Progress: 54%. Estimated remaining time: 6 minutes, 47 seconds.
## Growing trees.. Progress: 57%. Estimated remaining time: 6 minutes, 22 seconds.
## Growing trees.. Progress: 61%. Estimated remaining time: 5 minutes, 43 seconds.
## Growing trees.. Progress: 65%. Estimated remaining time: 5 minutes, 16 seconds.
## Growing trees.. Progress: 68%. Estimated remaining time: 4 minutes, 47 seconds.
## Growing trees.. Progress: 72%. Estimated remaining time: 4 minutes, 12 seconds.
## Growing trees.. Progress: 75%. Estimated remaining time: 3 minutes, 39 seconds.
## Growing trees.. Progress: 79%. Estimated remaining time: 3 minutes, 3 seconds.
## Growing trees.. Progress: 83%. Estimated remaining time: 2 minutes, 27 seconds.
## Growing trees.. Progress: 87%. Estimated remaining time: 1 minute, 53 seconds.
## Growing trees.. Progress: 91%. Estimated remaining time: 1 minute, 17 seconds.
## Growing trees.. Progress: 95%. Estimated remaining time: 44 seconds.
## Growing trees.. Progress: 98%. Estimated remaining time: 14 seconds.
## Error in pred[, "1"] : subscript out of bounds

## Warning in FUN(X[[i]], ...): Error in algorithm SL.ranger
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error : vector memory exhausted (limit reached?)

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Error in lognet(xd, is.sparse, ix, jx, y, weights, offset, alpha, nobs,  :
##   one multinomial or binomial class has 1 or 0 observations; not allowed

## Warning in FUN(X[[i]], ...): Error in algorithm SL.glmnet
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Growing trees.. Progress: 4%. Estimated remaining time: 12 minutes, 24 seconds.
## Growing trees.. Progress: 8%. Estimated remaining time: 12 minutes, 4 seconds.
## Growing trees.. Progress: 12%. Estimated remaining time: 12 minutes, 11 seconds.
## Growing trees.. Progress: 15%. Estimated remaining time: 11 minutes, 54 seconds.
## Growing trees.. Progress: 19%. Estimated remaining time: 11 minutes, 13 seconds.
## Growing trees.. Progress: 23%. Estimated remaining time: 10 minutes, 46 seconds.
## Growing trees.. Progress: 27%. Estimated remaining time: 10 minutes, 11 seconds.
## Growing trees.. Progress: 31%. Estimated remaining time: 9 minutes, 40 seconds.
## Growing trees.. Progress: 35%. Estimated remaining time: 9 minutes, 4 seconds.
## Growing trees.. Progress: 38%. Estimated remaining time: 8 minutes, 31 seconds.
## Growing trees.. Progress: 42%. Estimated remaining time: 7 minutes, 58 seconds.
## Growing trees.. Progress: 46%. Estimated remaining time: 7 minutes, 39 seconds.
## Growing trees.. Progress: 49%. Estimated remaining time: 7 minutes, 14 seconds.
## Growing trees.. Progress: 53%. Estimated remaining time: 6 minutes, 44 seconds.
## Growing trees.. Progress: 56%. Estimated remaining time: 6 minutes, 15 seconds.
## Growing trees.. Progress: 60%. Estimated remaining time: 5 minutes, 48 seconds.
## Growing trees.. Progress: 63%. Estimated remaining time: 5 minutes, 15 seconds.
## Growing trees.. Progress: 67%. Estimated remaining time: 4 minutes, 42 seconds.
## Growing trees.. Progress: 72%. Estimated remaining time: 4 minutes, 2 seconds.
## Growing trees.. Progress: 75%. Estimated remaining time: 3 minutes, 29 seconds.
## Growing trees.. Progress: 79%. Estimated remaining time: 2 minutes, 59 seconds.
```

```
## Growing trees.. Progress: 83%. Estimated remaining time: 2 minutes, 22 seconds.
## Growing trees.. Progress: 87%. Estimated remaining time: 1 minute, 54 seconds.
## Growing trees.. Progress: 90%. Estimated remaining time: 1 minute, 21 seconds.
## Growing trees.. Progress: 94%. Estimated remaining time: 53 seconds.
## Growing trees.. Progress: 98%. Estimated remaining time: 20 seconds.
## Error in pred[, "1"] : subscript out of bounds

## Warning in FUN(X[[i]], ...): Error in algorithm SL.ranger
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error : vector memory exhausted (limit reached?)

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Error in lognet(xd, is.sparse, ix, jx, y, weights, offset, alpha, nobs,  :
##   one multinomial or binomial class has 1 or 0 observations; not allowed

## Warning in FUN(X[[i]], ...): Error in algorithm SL.glmnet
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Growing trees.. Progress: 4%. Estimated remaining time: 14 minutes, 16 seconds.
## Growing trees.. Progress: 7%. Estimated remaining time: 13 minutes, 44 seconds.
## Growing trees.. Progress: 11%. Estimated remaining time: 13 minutes, 4 seconds.
## Growing trees.. Progress: 15%. Estimated remaining time: 11 minutes, 43 seconds.
## Growing trees.. Progress: 19%. Estimated remaining time: 11 minutes, 22 seconds.
## Growing trees.. Progress: 23%. Estimated remaining time: 10 minutes, 53 seconds.
## Growing trees.. Progress: 26%. Estimated remaining time: 10 minutes, 27 seconds.
## Growing trees.. Progress: 30%. Estimated remaining time: 10 minutes, 11 seconds.
## Growing trees.. Progress: 33%. Estimated remaining time: 9 minutes, 36 seconds.
## Growing trees.. Progress: 37%. Estimated remaining time: 9 minutes, 11 seconds.
## Growing trees.. Progress: 40%. Estimated remaining time: 8 minutes, 40 seconds.
## Growing trees.. Progress: 43%. Estimated remaining time: 8 minutes, 23 seconds.
## Growing trees.. Progress: 47%. Estimated remaining time: 7 minutes, 43 seconds.
## Growing trees.. Progress: 52%. Estimated remaining time: 6 minutes, 59 seconds.
## Growing trees.. Progress: 55%. Estimated remaining time: 6 minutes, 28 seconds.
## Growing trees.. Progress: 60%. Estimated remaining time: 5 minutes, 48 seconds.
## Growing trees.. Progress: 63%. Estimated remaining time: 5 minutes, 15 seconds.
## Growing trees.. Progress: 68%. Estimated remaining time: 4 minutes, 37 seconds.
## Growing trees.. Progress: 71%. Estimated remaining time: 4 minutes, 12 seconds.
## Growing trees.. Progress: 74%. Estimated remaining time: 3 minutes, 43 seconds.
## Growing trees.. Progress: 78%. Estimated remaining time: 3 minutes, 15 seconds.
## Growing trees.. Progress: 81%. Estimated remaining time: 2 minutes, 46 seconds.
## Growing trees.. Progress: 85%. Estimated remaining time: 2 minutes, 8 seconds.
## Growing trees.. Progress: 89%. Estimated remaining time: 1 minute, 35 seconds.
## Growing trees.. Progress: 92%. Estimated remaining time: 1 minute, 9 seconds.
## Growing trees.. Progress: 96%. Estimated remaining time: 34 seconds.
## Growing trees.. Progress: 99%. Estimated remaining time: 6 seconds.
## Error in pred[, "1"] : subscript out of bounds

## Warning in FUN(X[[i]], ...): Error in algorithm SL.ranger
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error : vector memory exhausted (limit reached?)
```

```
## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Error in lognet(xd, is.sparse, ix, jx, y, weights, offset, alpha, nobs,  :
##   one multinomial or binomial class has 1 or 0 observations; not allowed

## Warning in FUN(X[[i]], ...): Error in algorithm SL.glmnet
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Growing trees.. Progress: 4%. Estimated remaining time: 14 minutes, 16 seconds.
## Growing trees.. Progress: 7%. Estimated remaining time: 13 minutes, 33 seconds.
## Growing trees.. Progress: 11%. Estimated remaining time: 12 minutes, 56 seconds.
## Growing trees.. Progress: 14%. Estimated remaining time: 12 minutes, 40 seconds.
## Growing trees.. Progress: 18%. Estimated remaining time: 12 minutes, 24 seconds.
## Growing trees.. Progress: 22%. Estimated remaining time: 11 minutes, 21 seconds.
## Growing trees.. Progress: 25%. Estimated remaining time: 10 minutes, 54 seconds.
## Growing trees.. Progress: 30%. Estimated remaining time: 9 minutes, 58 seconds.
## Growing trees.. Progress: 34%. Estimated remaining time: 9 minutes, 17 seconds.
## Growing trees.. Progress: 38%. Estimated remaining time: 8 minutes, 49 seconds.
## Growing trees.. Progress: 42%. Estimated remaining time: 8 minutes, 14 seconds.
## Growing trees.. Progress: 46%. Estimated remaining time: 7 minutes, 35 seconds.
## Growing trees.. Progress: 49%. Estimated remaining time: 7 minutes, 9 seconds.
## Growing trees.. Progress: 53%. Estimated remaining time: 6 minutes, 43 seconds.
## Growing trees.. Progress: 56%. Estimated remaining time: 6 minutes, 10 seconds.
## Growing trees.. Progress: 60%. Estimated remaining time: 5 minutes, 38 seconds.
## Growing trees.. Progress: 64%. Estimated remaining time: 5 minutes, 6 seconds.
## Growing trees.. Progress: 67%. Estimated remaining time: 4 minutes, 38 seconds.
## Growing trees.. Progress: 71%. Estimated remaining time: 4 minutes, 7 seconds.
## Growing trees.. Progress: 74%. Estimated remaining time: 3 minutes, 41 seconds.
## Growing trees.. Progress: 78%. Estimated remaining time: 3 minutes, 8 seconds.
## Growing trees.. Progress: 82%. Estimated remaining time: 2 minutes, 38 seconds.
## Growing trees.. Progress: 85%. Estimated remaining time: 2 minutes, 11 seconds.
## Growing trees.. Progress: 88%. Estimated remaining time: 1 minute, 42 seconds.
## Growing trees.. Progress: 92%. Estimated remaining time: 1 minute, 13 seconds.
## Growing trees.. Progress: 95%. Estimated remaining time: 43 seconds.
## Growing trees.. Progress: 98%. Estimated remaining time: 15 seconds.
## Error in pred[, "1"] : subscript out of bounds

## Warning in FUN(X[[i]], ...): Error in algorithm SL.ranger
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)


## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error : vector memory exhausted (limit reached?)

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Error in lognet(xd, is.sparse, ix, jx, y, weights, offset, alpha, nobs,  :
##   one multinomial or binomial class has 1 or 0 observations; not allowed

## Warning in FUN(X[[i]], ...): Error in algorithm SL.glmnet  on full data
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Growing trees.. Progress: 3%. Estimated remaining time: 17 minutes, 47 seconds.
## Growing trees.. Progress: 6%. Estimated remaining time: 17 minutes, 14 seconds.
## Growing trees.. Progress: 9%. Estimated remaining time: 16 minutes, 20 seconds.
## Growing trees.. Progress: 12%. Estimated remaining time: 16 minutes, 4 seconds.
```

```
## Growing trees.. Progress: 15%. Estimated remaining time: 14 minutes, 58 seconds.
## Growing trees.. Progress: 18%. Estimated remaining time: 14 minutes, 46 seconds.
## Growing trees.. Progress: 20%. Estimated remaining time: 14 minutes, 34 seconds.
## Growing trees.. Progress: 23%. Estimated remaining time: 13 minutes, 54 seconds.
## Growing trees.. Progress: 26%. Estimated remaining time: 13 minutes, 28 seconds.
## Growing trees.. Progress: 29%. Estimated remaining time: 13 minutes, 11 seconds.
## Growing trees.. Progress: 32%. Estimated remaining time: 12 minutes, 32 seconds.
## Growing trees.. Progress: 35%. Estimated remaining time: 12 minutes, 7 seconds.
## Growing trees.. Progress: 38%. Estimated remaining time: 11 minutes, 20 seconds.
## Growing trees.. Progress: 41%. Estimated remaining time: 10 minutes, 40 seconds.
## Growing trees.. Progress: 45%. Estimated remaining time: 9 minutes, 57 seconds.
## Growing trees.. Progress: 47%. Estimated remaining time: 9 minutes, 34 seconds.
## Growing trees.. Progress: 50%. Estimated remaining time: 9 minutes, 9 seconds.
## Growing trees.. Progress: 53%. Estimated remaining time: 8 minutes, 35 seconds.
## Growing trees.. Progress: 56%. Estimated remaining time: 8 minutes, 3 seconds.
## Growing trees.. Progress: 59%. Estimated remaining time: 7 minutes, 29 seconds.
## Growing trees.. Progress: 62%. Estimated remaining time: 6 minutes, 49 seconds.
## Growing trees.. Progress: 65%. Estimated remaining time: 6 minutes, 14 seconds.
## Growing trees.. Progress: 69%. Estimated remaining time: 5 minutes, 38 seconds.
## Growing trees.. Progress: 72%. Estimated remaining time: 4 minutes, 57 seconds.
## Growing trees.. Progress: 75%. Estimated remaining time: 4 minutes, 30 seconds.
## Growing trees.. Progress: 77%. Estimated remaining time: 4 minutes, 6 seconds.
## Growing trees.. Progress: 80%. Estimated remaining time: 3 minutes, 37 seconds.
## Growing trees.. Progress: 83%. Estimated remaining time: 3 minutes, 4 seconds.
## Growing trees.. Progress: 86%. Estimated remaining time: 2 minutes, 36 seconds.
## Growing trees.. Progress: 88%. Estimated remaining time: 2 minutes, 8 seconds.
## Growing trees.. Progress: 92%. Estimated remaining time: 1 minute, 28 seconds.
## Growing trees.. Progress: 95%. Estimated remaining time: 56 seconds.
## Growing trees.. Progress: 98%. Estimated remaining time: 19 seconds.
## Error in pred[, "1"] : subscript out of bounds

## Warning in FUN(X[[i]], ...): Error in algorithm SL.ranger  on full data
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in FUN(X[[i]], ...): Variable(s) 'Xx.1' constant. Cannot scale data.

## Error : vector memory exhausted (limit reached?)

## Warning in FUN(X[[i]], ...): Error in algorithm SL.svm  on full data
##   The Algorithm will be removed from the Super Learner (i.e. given weight 0)

## Warning in SuperLearner::SuperLearner(Y = Y.subset, X = X.subset, SL.library =
## SL.library, : Coefficients already 0 for all failed algorithm(s)

## Call:
## ltmle(data = data, Anodes = c("A1", "A2"), Lnodes = c("bmi",
##     "bmi_2", "blood_pressure", "blood_pressure_2", "chol", "chol_2"),
##     Ynodes = "Y", abar = c(1, 1), SL.library = sl_libs)
##
## TMLE Estimate:  0.2507633
```

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 In the LTMLE model, time-dependent confounding can be a concern when estimating the causal effect of a time-varying exposure or treatment on an outcome. Time-dependent confounding occurs when an unmeasured or poorly measured variable that changes over time is associated with both the treatment and the outcome. There could be time-varying confounders (like diet and blood pressure), time-varying effect modification (condition severity changes), and time-dependent selection bias (population dropping from study). For specific examples, a running variable for age would not necessarily be a time-dependent confounder if it is not associated with the outcome or the treatment. However, if age is associated with both the treatment and outcome, and its relationship with the outcome changes over time, it could be a time-varying confounder. On the other hand, blood pressure measured at two different times could be a time-varying confounder if it is associated with both the treatment and outcome, and its relationship with the outcome changes over time. Blood pressure is more variable than age, capturing variability over time that is unpredictible compared to the predictible variation in age over time.