Project 8 Nereida Heller

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

#install.packages("xgboost")

library(xgboost)
```

```
## Warning: package 'xgboost' was built under R version 4.4.3

##
## Attaching package: 'xgboost'

## The following object is masked from 'package:dplyr':
##
##     slice
```

```r
library(readr)
library(caret)
library(ggdag)
library(dplyr)
library(ggplot2)


#Keeping original code in case you need to run it, Kasey
#heart_disease <- read_csv(here('heart_disease_tmle.csv'))
heart_disease <- read_csv(here("Project 8", "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

```r
# Subsetting to just necessary variables
heart_disease_SL <- heart_disease %>% select(-ends_with("_2"))
#glimpse(heart_disease_SL)

# Fit SuperLearner Model

## sl lib
#listWrappers()
# set seed
set.seed(987)

## Train/Test split

# initial split
# ----------
heart_split_SL <-
  initial_split(heart_disease_SL, prop = 3/4) # create initial split (tidymodels)

train <-
  # Declare the training set with rsample::training()
  training(heart_split_SL)

# y_train
y_train <-
  train %>%
  pull(mortality)

# x_train
x_train <-
  train %>%
  select(-mortality)

# Testing
# ----------
```

```r
test <-
  testing(heart_split_SL)

# y test
y_test <-
  test %>%
  pull(mortality)

# x test
x_test <-
  test %>%
  select(-mortality)
```

```r
# multiple models
## Train SuperLearner
# ----------
sl = SuperLearner(Y = y_train,
                  X = x_train,
                  family = binomial(),
                  SL.library = c('SL.mean',    # Baseline
                                 'SL.glm',     # Logistic regression
                                 'SL.glmnet',  # Lasso and ridge
                                 'SL.ranger',  # Random forest
                                 'SL.xgboost')) #Because I remember this from earlier in the semester
```

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

```r
## Risk and Coefficient of each model
sl
```

```
##
## Call:
## SuperLearner(Y = y_train, X = x_train, family = binomial(), SL.library = c("SL.mean",
##     "SL.glm", "SL.glmnet", "SL.ranger", "SL.xgboost"))
##
##
##                      Risk       Coef
## SL.mean_All    0.2497082 0.00000000
## SL.glm_All     0.2353773 0.00000000
## SL.glmnet_All  0.2352033 0.36620934
## SL.ranger_All  0.2310625 0.58237631
## SL.xgboost_All 0.2453601 0.05141435
```

```r
## Discrete winner and superlearner ensemble performance
sl$cvRisk[which.min(sl$cvRisk)]
```

```
## SL.ranger_All
##     0.2310625
```

```r
## Confusion Matrix
```

```r
# predictions
# ----------
preds <-
  predict(sl,             # use the superlearner not individual models
          x_test,         # prediction on test set
          onlySL = TRUE)  # use only models that were found to be useful (had weights)
glimpse(preds)
```

```
## List of 2
##  $ pred           : num [1:2500, 1] 0.627 0.568 0.672 0.586 0.655 ...
##  $ library.predict: num [1:2500, 1:5] 0 0 0 0 0 0 0 0 0 0 ...
```

```r
# start with y_test
validation <-
  y_test %>%
  # add our predictions - first column of predictions
  bind_cols(preds$pred[,1]) %>%
  # rename columns
  rename(obs = `...1`,      # actual observations --> r names them ...1 and ...2, we're renaming them s
         pred = `...2`) %>% # predicted prob
  # change pred column so that obs above .5 are 1, otherwise 0
  mutate(pred = ifelse(pred >= .5,
                       1,
                       0))
```

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

```r
# confusion matrix
# table(Predicted = validation$pred, Actual = validation$obs)
# confusionMatrix(factor(validation$pred), factor(validation$obs))
print(caret::confusionMatrix(as.factor(validation$pred),
                       as.factor(validation$obs)))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0  410  236
##          1  808 1046
##
##                Accuracy : 0.5824
##                  95% CI : (0.5628, 0.6018)
##     No Information Rate : 0.5128
##     P-Value [Acc > NIR] : 1.684e-12
##
##                   Kappa : 0.1543
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.3366
##             Specificity : 0.8159
##          Pos Pred Value : 0.6347
##          Neg Pred Value : 0.5642
##              Prevalence : 0.4872
##          Detection Rate : 0.1640
##    Detection Prevalence : 0.2584
##       Balanced Accuracy : 0.5763
##
##        'Positive' Class : 0
##
```

```
#caret
cm <- caret::confusionMatrix(as.factor(validation$pred),
                       as.factor(validation$obs))
cm_table <- cm$table
cm_percent <- cm_table / sum(cm_table) * 100
cm_percent_rounded <- round(cm_percent, 2)
print(cm_percent_rounded)
```

```
##           Reference
## Prediction     0     1
##          0 16.40  9.44
##          1 32.32 41.84
```

```
#base R
# cm_table <- table(Predicted = validation$pred, Actual = validation$obs)
#
# cm_table <- cm$table
# cm_percent <- cm$table / sum(cm$table) * 100
# cm_percent_rounded <- round(cm_percent, 2)
# print(cm_percent_rounded)

# heart_split_SL
#
# print(410+236+808+1046)
```

The random forest trainer “ranger” did the best. The model didn’t perform all that well in the confusion matrix though.

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)?

Blending multiple algorithms will be more robust bc you end up taking advantage of the strengths of various algorithms, reducing the chance of overfitting. Superlearner also assign weights to the models, which helps mitigate the weaknesses of any one model by balancing their contributions. All of these advantages just mean that SuperLearner will outperform any single model in terms of predictive accuracy and generalization.

Targeted Maximum Likelihood Estimation

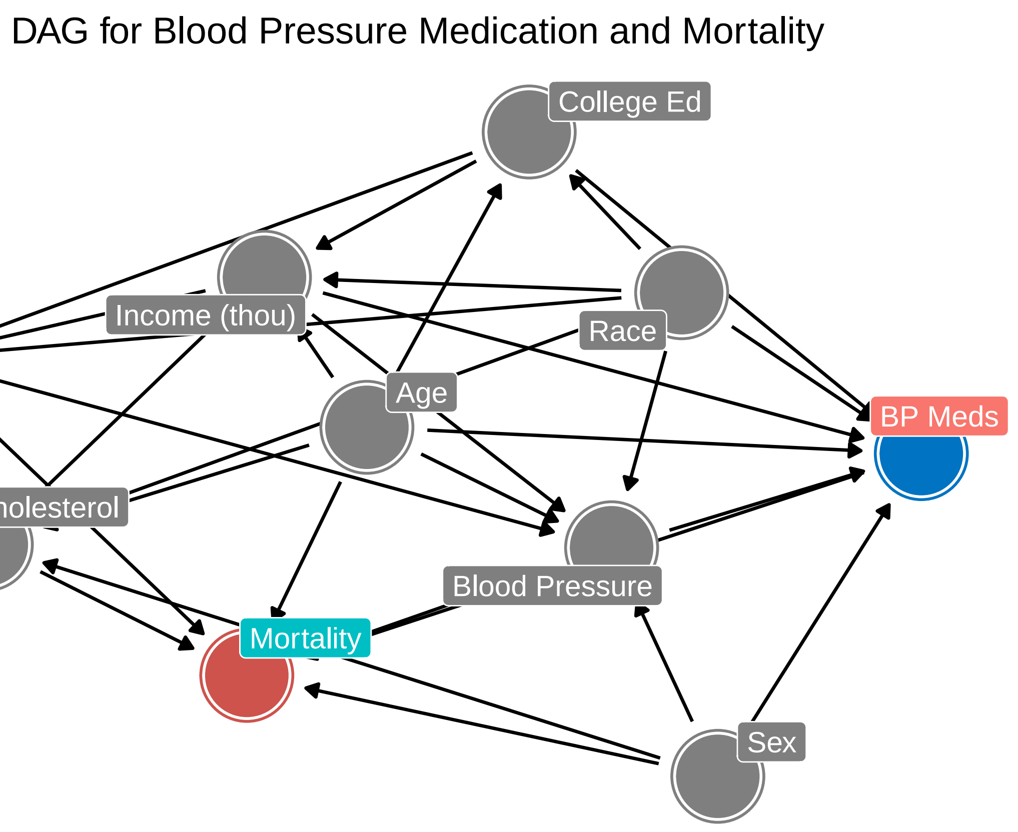
Causal Diagram

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.

```r
# DAG for TMLE
blood_pressure_dag <- dagify(
  # Outcome
  mortality ~ blood_pressure_medication + blood_pressure + chol + bmi + age + sex_at_birth,

  # Treatment assignment
  blood_pressure_medication ~ blood_pressure + age + sex_at_birth + simplified_race + college_educ + in

  # Other relationships
  blood_pressure ~ age + bmi + simplified_race + sex_at_birth + income_thousands,
  chol ~ age + bmi + simplified_race + sex_at_birth + income_thousands,
  bmi ~ simplified_race + income_thousands + college_educ,
  income_thousands ~ college_educ + simplified_race + age,
  college_educ ~ simplified_race + age,

  # Exposure and outcome
  exposure = "blood_pressure_medication",
  outcome = "mortality",

  # Variable labels
  labels = c(
    "mortality" = "Mortality",
    "blood_pressure_medication" = "BP Meds",
    "blood_pressure" = "Blood Pressure",
    "chol" = "Cholesterol",
    "bmi" = "BMI",
    "age" = "Age",
    "sex_at_birth" = "Sex",
    "simplified_race" = "Race",
    "college_educ" = "College Ed",
    "income_thousands" = "Income (thou)"
  )
)

# Plot
ggdag_status(blood_pressure_dag,
             use_labels = "label",
             text = FALSE,
             stylized = TRUE) +
  theme_dag() +
  scale_color_manual(values = c("exposure" = "#0073C2",
                                "outcome" = "#CD534C",
                                "latent" = "#grey80")) +
  guides(color = "none") +  # Remove the legend
  theme(plot.title = element_text(hjust = 0.5)) +
  ggtitle("DAG for Blood Pressure Medication and Mortality")
```



```
# # Identify adjustment sets
# adjustmentSets <- ggdag_adjustment_set(blood_pressure_dag)
# print(adjustmentSets)
#
# # Alternative view showing minimal adjustment set paths
# ggdag_paths(blood_pressure_dag) +
#   theme_dag() +
#   ggtitle("Causal Pathways in Blood Pressure Medication DAG")
```

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

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.

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
```

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

## LTMLE estimate
```

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?