Project 8 Template

```
options(warn = -1)

# 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("C:/Users/MSalman/Desktop/courses/2024-01_Spring/CSS 11/Computational-Social-S
```

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.

```
#Clean out the "..._2" variables
library(dplyr)
heart_disease <- select(heart_disease, -c(bmi_2, blood_pressure_2, chol_2, blood_pressure_medication_2))
```

Explorative data analysis...

```
summary(heart_disease) #First, let's have a look at the data...
```

```
##       age           sex_at_birth    simplified_race  college_educ
##  Min.   :  1.972   Min.   :0.0000   Min.   :1.000   Min.   :1.00
##  1st Qu.: 40.655   1st Qu.:0.0000   1st Qu.:1.000   1st Qu.:1.00
##  Median : 50.066   Median :1.0000   Median :2.000   Median :2.00
##  Mean   : 50.086   Mean   :0.5099   Mean   :2.256   Mean   :1.67
##  3rd Qu.: 59.471   3rd Qu.:1.0000   3rd Qu.:3.000   3rd Qu.:2.00
##  Max.   :101.168   Max.   :1.0000   Max.   :5.000   Max.   :2.00
##  income_thousands      bmi        blood_pressure       chol
##  Min.   :-10.40   Min.   :14.81   Min.   :106.5   Min.   :151.8
##  1st Qu.: 60.03   1st Qu.:23.83   1st Qu.:125.1   1st Qu.:193.3
##  Median : 75.58   Median :25.78   Median :130.1   Median :203.2
##  Mean   : 75.25   Mean   :26.02   Mean   :132.1   Mean   :203.8
##  3rd Qu.: 90.57   3rd Qu.:27.94   3rd Qu.:136.2   3rd Qu.:213.4
##  Max.   :157.18   Max.   :42.30   Max.   :182.6   Max.   :272.9
##  blood_pressure_medication   mortality
##  Min.   :0.0000            Min.   :0.0000
##  1st Qu.:0.0000            1st Qu.:0.0000
##  Median :0.0000            Median :1.0000
##  Mean   :0.1551            Mean   :0.5168
##  3rd Qu.:0.0000            3rd Qu.:1.0000
##  Max.   :1.0000            Max.   :1.0000
```

There are no missing values in any of the columns:

```r
colSums(is.na(heart_disease)) #Great, there are no missing values in any of the columns!
```

```
##                        age              sex_at_birth           simplified_race
##                          0                         0                         0
##              college_educ          income_thousands                       bmi
##                          0                         0                         0
##            blood_pressure                      chol blood_pressure_medication
##                          0                         0                         0
##                 mortality
##                          0
```

```r
table(heart_disease$sex_at_birth)
```

```
##
##    0    1
## 4901 5099
```

```r
table(heart_disease$simplified_race)
```

```
##
##    1    2    3    4    5
## 4468 1559 1934 1027 1012
```

```r
table(heart_disease$college_educ)
```

```
##
##    1    2
## 3300 6700
```

```r
# table(heart_disease$income_thousands)
# table(heart_disease$bmi)
# table(heart_disease$blood_pressure)
# table(heart_disease$chol)
table(heart_disease$blood_pressure_medication)
```

```
##
##    0    1
## 8449 1551
```

```r
table(heart_disease$bmi_2)
```

```
## < table of extent 0 >
```

```r
table(heart_disease$blood_pressure_2)
```

```
## < table of extent 0 >
```

```r
table(heart_disease$chol_2)
```

```
## < table of extent 0 >
```

```r
table(heart_disease$blood_pressure_medication_2)
```

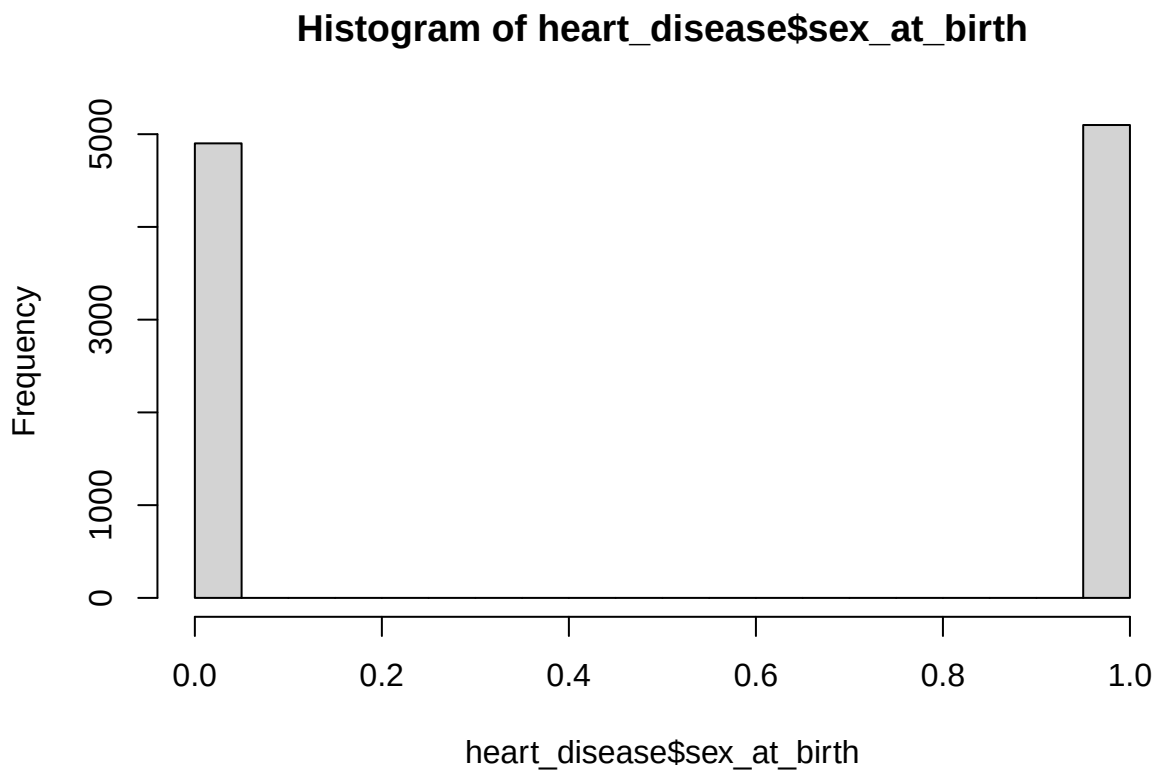
```
## < table of extent 0 >
```

```r
table(heart_disease$mortality)
```

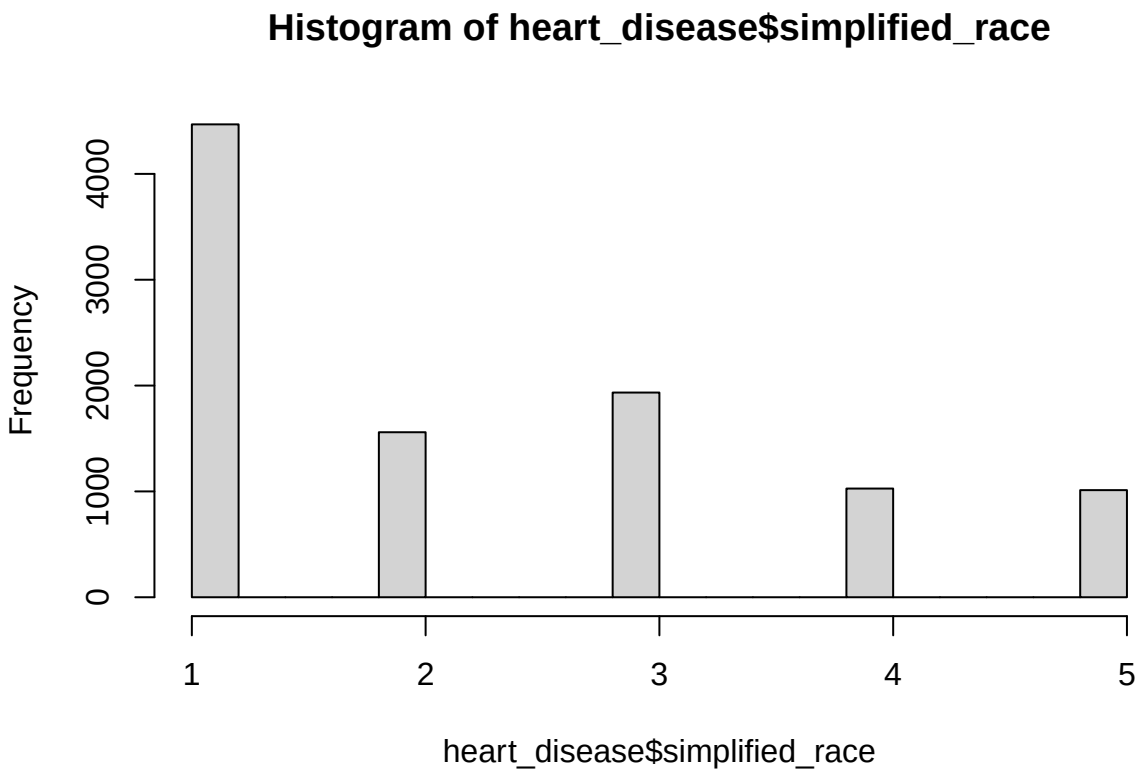
```
##
##    0    1
## 4832 5168
```

```r
# table(heart_disease$age)
```

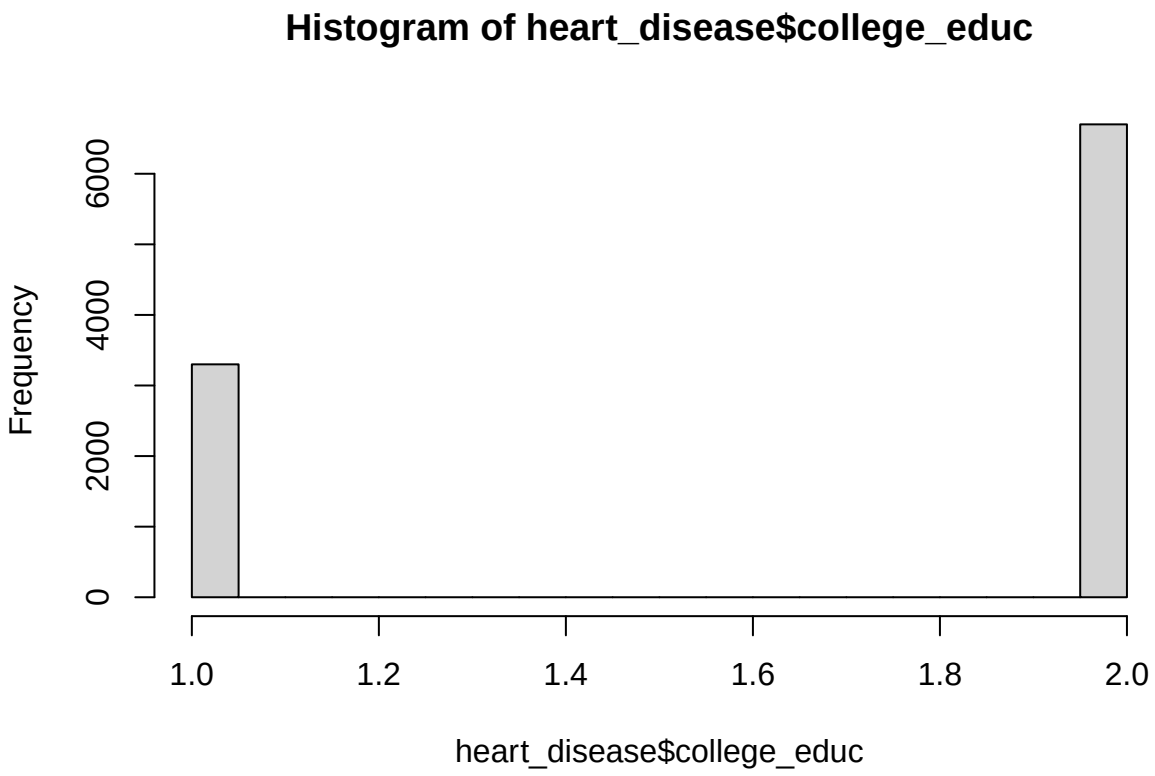
```r
hist(heart_disease$sex_at_birth)
```



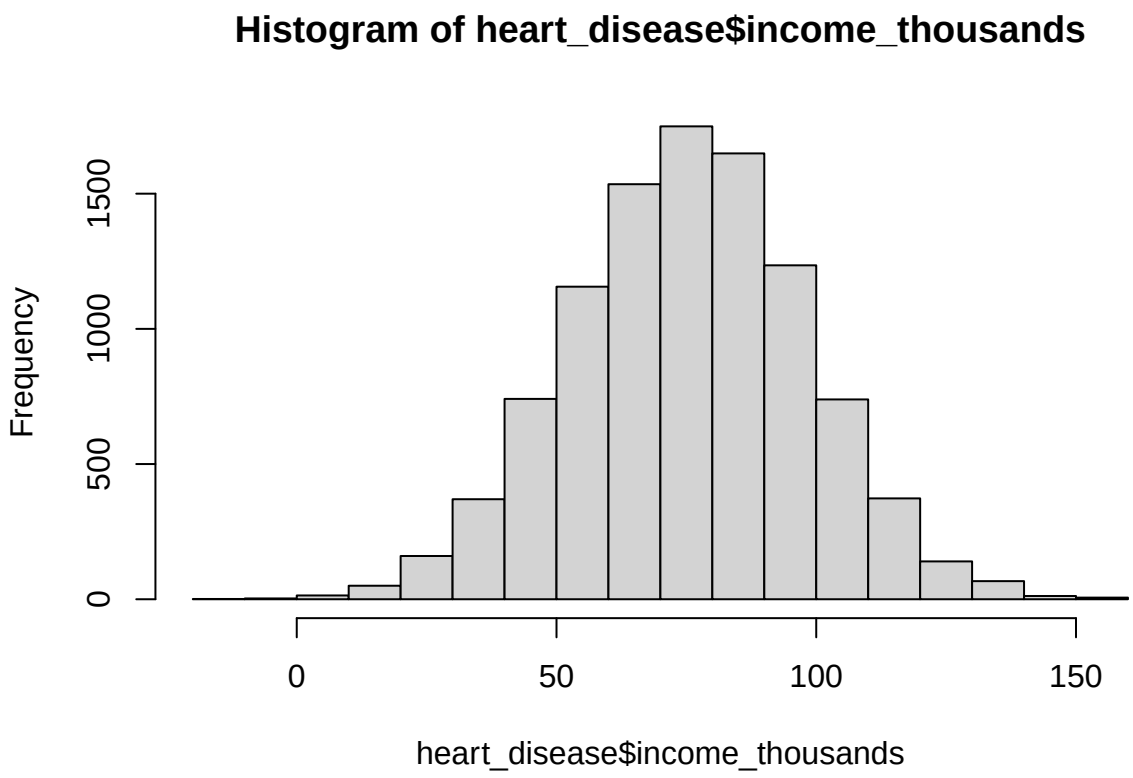

```r
hist(heart_disease$simplified_race)
```




```r
hist(heart_disease$college_educ)
```

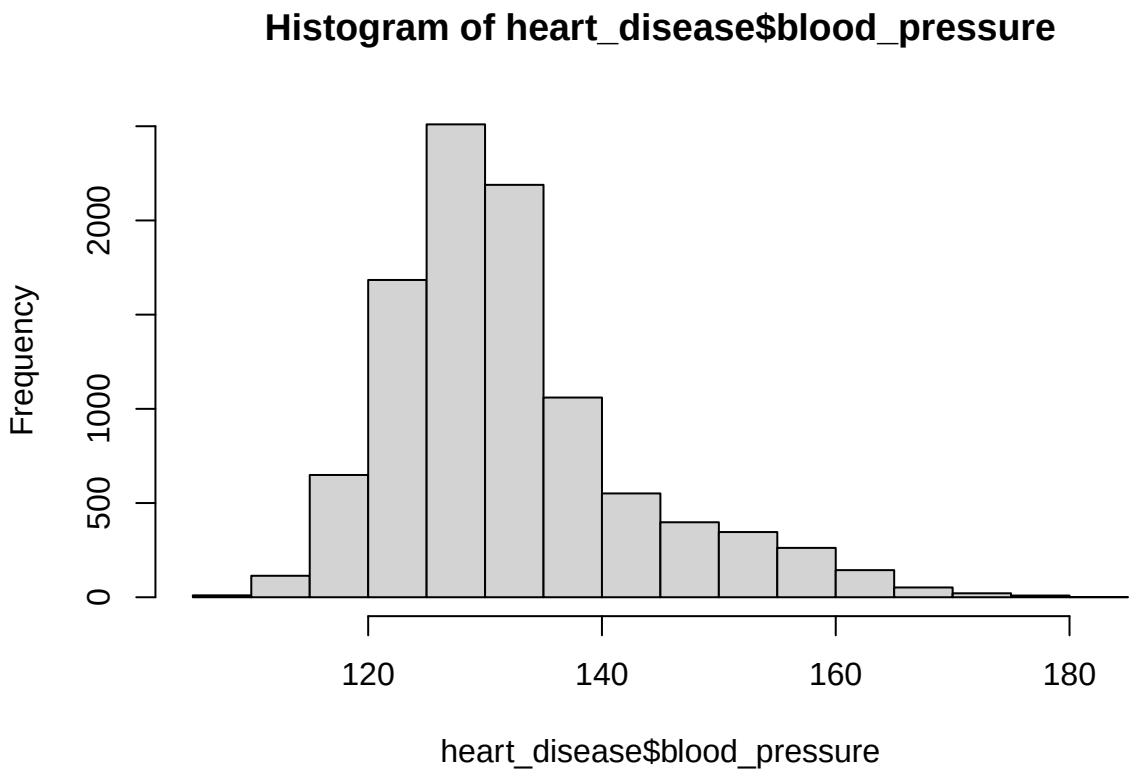



```
hist(heart_disease$income_thousands)
```

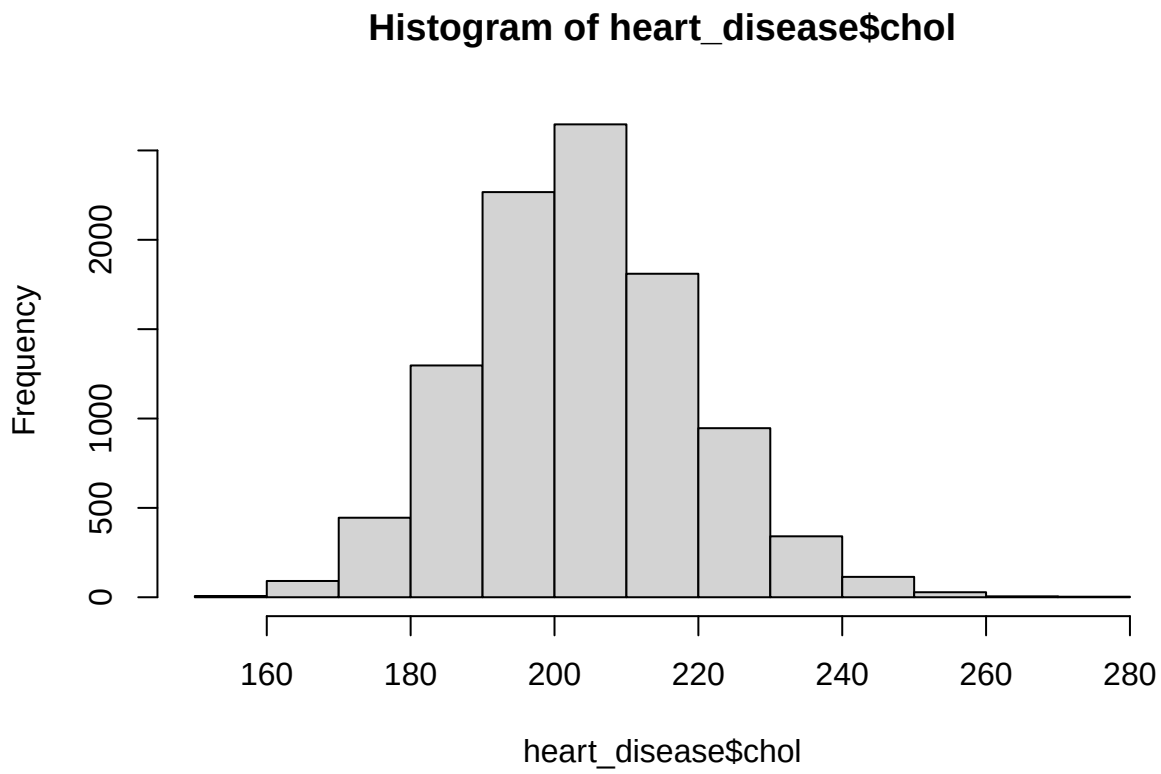



```
hist(heart_disease$bmi)
```

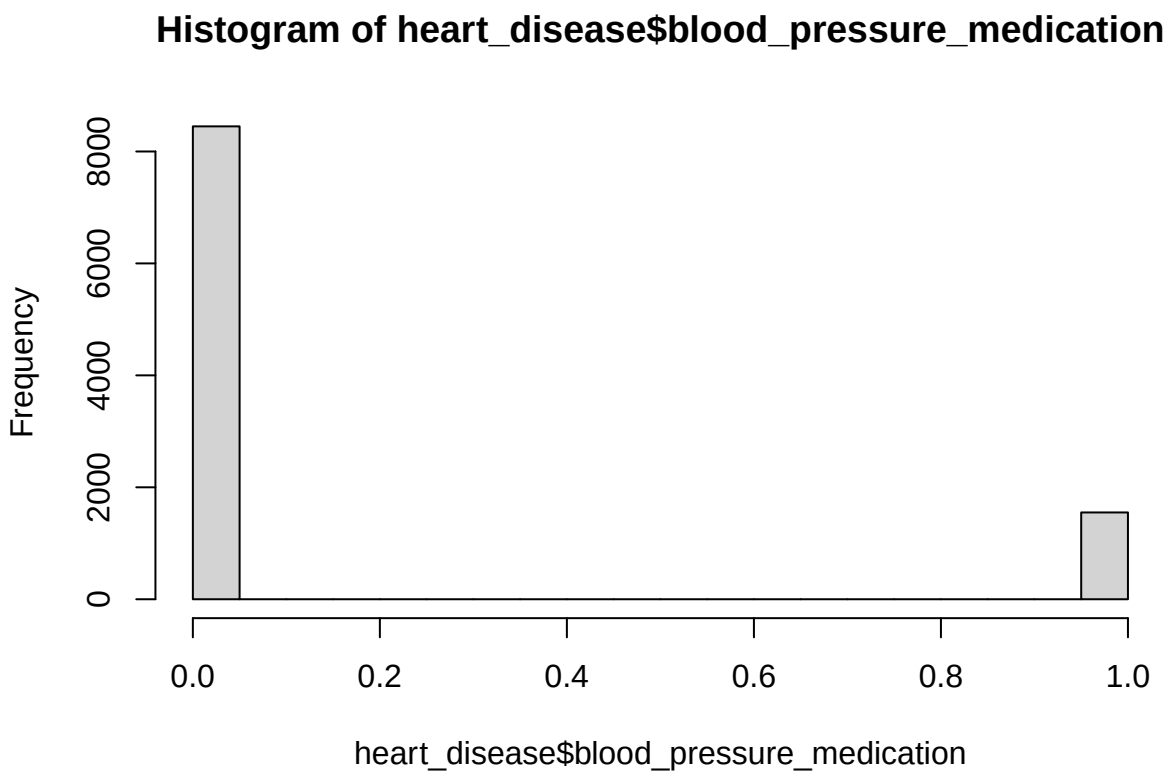

```r
hist(heart_disease$blood_pressure)
```



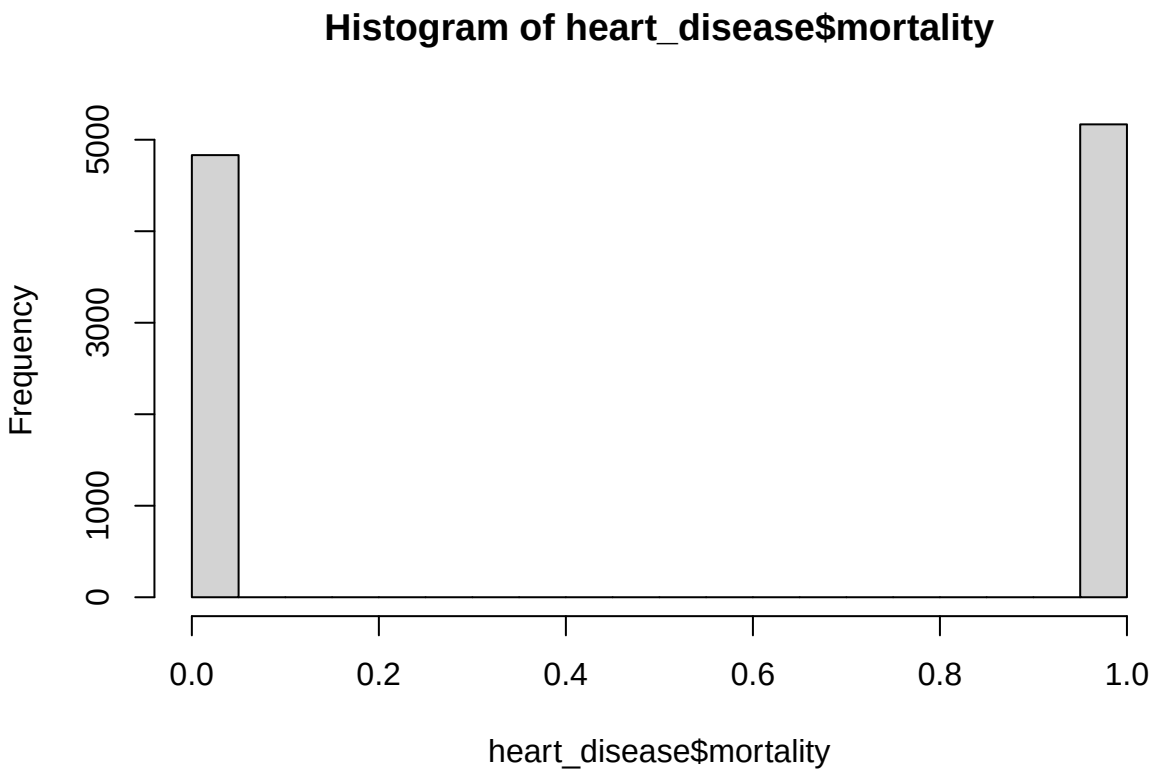

```r
hist(heart_disease$chol)
```



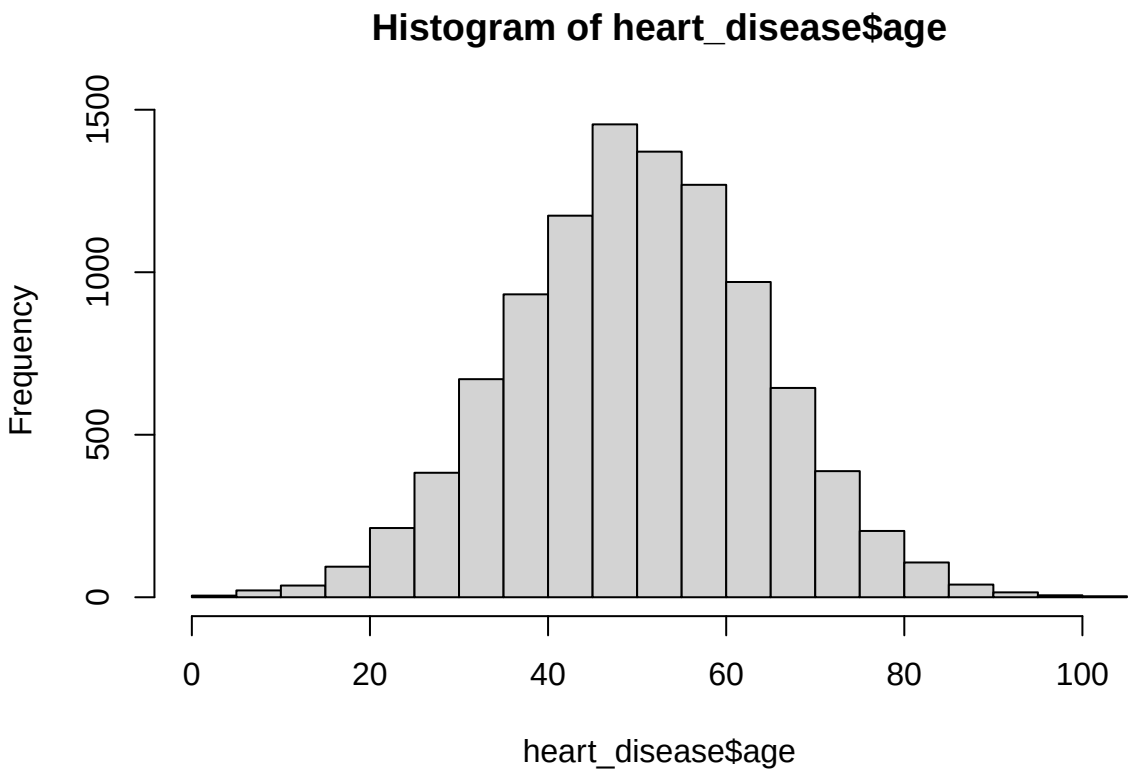

```r
hist(heart_disease$blood_pressure_medication)
```




```
hist(heart_disease$mortality)
```




```
hist(heart_disease$age)
```



The correlation table (below) shows a negative correlations between blood pressure medication and mortality (-0.23), indicating a reducing effect of medication on mortality. The table shows a relatively high correlation between BMI and blood pressure (0.35). This can potentially be explained with the fact that both BMI and blood pressure are associated with cardiovascular health and weight.

```
cor(heart_disease[, sapply(heart_disease, is.numeric)]) #Correlation matrix
```

```
##                                age sex_at_birth simplified_race
## age                    1.000000000 -0.011245344     0.006274486
## sex_at_birth          -0.011245344  1.000000000    -0.013117504
## simplified_race        0.006274486 -0.013117504     1.000000000
## college_educ          -0.006495024 -0.005245463    -0.024480307
## income_thousands       0.014136186 -0.016757659    -0.011382655
## bmi                    0.254699860 -0.007053100     0.009453725
## blood_pressure        -0.007011953 -0.003236226     0.011404186
## chol                  -0.002547905  0.007200745    -0.006538182
## blood_pressure_medication  0.045026519  0.005053532     0.001717970
## mortality              0.015868364 -0.001666548    -0.002607019
##                       college_educ income_thousands          bmi
## age                   -0.006495024      0.014136186  0.254699860
## sex_at_birth          -0.005245463     -0.016757659 -0.007053100
## simplified_race       -0.024480307     -0.011382655  0.009453725
## college_educ           1.000000000     -0.003069459 -0.005230619
## income_thousands      -0.003069459      1.000000000  0.007558260
## bmi                   -0.005230619      0.007558260  1.000000000
## blood_pressure        -0.006296430      0.003786941  0.354177236
```

```
## chol                       -0.011723063      0.004121498  0.141476572
## blood_pressure_medication   0.004600011     -0.005965218  0.196291270
## mortality                  -0.003642962     -0.001299843  0.008330067
##                           blood_pressure         chol blood_pressure_medication
## age                         -0.007011953 -0.002547905               0.045026519
## sex_at_birth                -0.003236226  0.007200745               0.005053532
## simplified_race              0.011404186 -0.006538182               0.001717970
## college_educ                -0.006296430 -0.011723063               0.004600011
## income_thousands             0.003786941  0.004121498              -0.005965218
## bmi                          0.354177236  0.141476572               0.196291270
## blood_pressure               1.000000000  0.179442766               0.064744818
## chol                         0.179442766  1.000000000               0.040164915
## blood_pressure_medication    0.064744818  0.040164915               1.000000000
## mortality                    0.025513241  0.052898204              -0.227507916
##                             mortality
## age                        0.015868364
## sex_at_birth              -0.001666548
## simplified_race           -0.002607019
## college_educ              -0.003642962
## income_thousands          -0.001299843
## bmi                        0.008330067
## blood_pressure             0.025513241
## chol                       0.052898204
## blood_pressure_medication -0.227507916
## mortality                  1.000000000
```

In the table below, the rows show whether someone took blood pressure medication or not (0/1–> no/yes).The rows portray mortality, whether someone survived or died (0/1–> O=survived; 1=died).

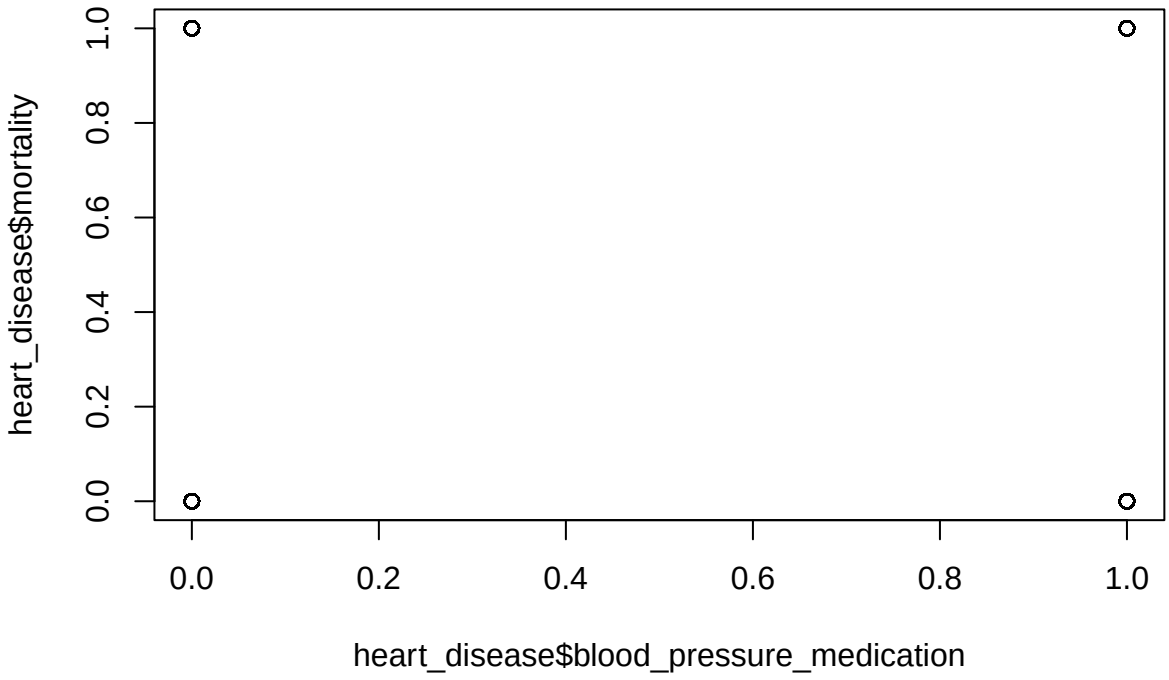
(0, 0): 3671 people didn't take blood pressure medication and didn't die. (0, 1): 4778 people didn't take blood pressure medication and died. (1, 0): 1161 people took blood pressure medication and didn't die. (1, 1): 390 people took blood pressure medication but still died.

```
table(heart_disease$blood_pressure_medication, heart_disease$mortality)
```

```
##
##        0    1
##   0 3671 4778
##   1 1161  390
```

The table shows that there were more deaths among individuals who did not take blood pressure medication (4778) compared to those who did (390), indicating a potential reducing effect of blood pressure medication on mortality.
Interestingly, the number of survivors is higher among those who did not take the medication (3671) compared to those who did (1161). This might be due to selection bias. Healthy individuals do not need to take any medication. Those with blood pressure issues already have a higher mortality. The medication might still have reduced their mortality, but those who didn't take the medication is not a good comparison or reference point to determine the effect of the medication, due to the selection bias explained.

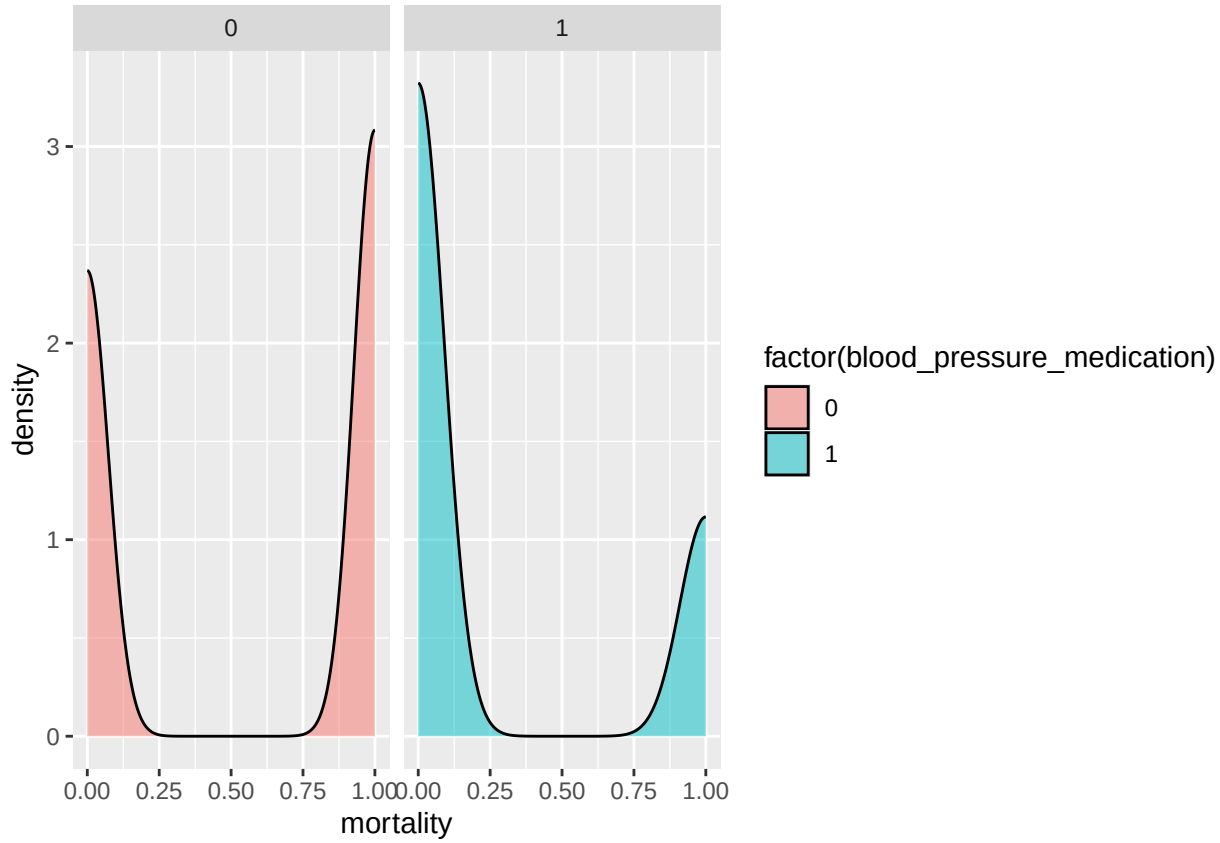
```
plot(heart_disease$blood_pressure_medication, heart_disease$mortality)
```




```
tapply(heart_disease$bmi, heart_disease$blood_pressure_medication, mean)
```

```
##        0        1
## 25.74328 27.53030
```

```
library(ggplot2)
ggplot(heart_disease, aes(x = mortality, fill = factor(blood_pressure_medication))) +
  geom_density(alpha = 0.5) +
  facet_wrap(~blood_pressure_medication)
```



I ran a regression model (glm) to analyze the relationship between the variables age, sex at birth, bmi, and cholesterol and mortality (1=died, 0=survived). The variable most significant in predicting mortality is cholesterol. The coefficient on cholesterol is 0.007031, with a p-value < 0.001, likely indicating that higher cholesterol levels increase the risk of mortality. Age, sex, and BMI are no statistically significant predictors of mortality. That they don't have a significant impact on mortality, all other variables controlled.

```
glm_model <- glm(mortality ~ age + sex_at_birth + bmi + chol, data = heart_disease, family = "binomial")
summary(glm_model)
```

```
##
## Call:
## glm(formula = mortality ~ age + sex_at_birth + bmi + chol, family = "binomial",
##     data = heart_disease)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.426987   0.299162  -4.770 1.84e-06 ***
## age           0.002427   0.001486   1.633    0.102
## sex_at_birth -0.007575   0.040096  -0.189    0.850
## bmi          -0.002151   0.006361  -0.338    0.735
## chol          0.007031   0.001332   5.281 1.29e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 13852  on 9999  degrees of freedom
## Residual deviance: 13821  on 9995  degrees of freedom
## AIC: 13831
##
## Number of Fisher Scoring iterations: 3
```

The second model (below) indicates that taking blood pressure medication significantly reduces the risk of mortality when cholesterol levels are accounted for. Higher cholesterol levels significantly increase the risk of mortality. The coefficient on blood pressure medication is -1.375064 with a statistically significant p-value of <2e-16. Taking blood pressure medication is associated with a decrease in mortality, when controlling for cholesterol levels.The coefficient on cholesterol is 0.008664 with a significantly high p-value of 1.83e-10. This indicates that higher cholesterol levels are associated with a higher risk of mortality.

```
glm_model <- glm(mortality ~ blood_pressure_medication + chol, data = heart_disease, family = "binomial
summary(glm_model)
```

```
##
## Call:
## glm(formula = mortality ~ blood_pressure_medication + chol, family = "binomial",
##     data = heart_disease)
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -1.498311   0.277049  -5.408 6.37e-08 ***
## blood_pressure_medication -1.375064   0.062763 -21.909  < 2e-16 ***
## chol                        0.008664   0.001359   6.375 1.83e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13852  on 9999  degrees of freedom
## Residual deviance: 13276  on 9997  degrees of freedom
## AIC: 13282
##
## Number of Fisher Scoring iterations: 4
```

Now that we assume that blodd pressure medication is associated with a reduction in mortality from heart disease, let's also look into which factors are associated with blood pressure medication intake. The results below suggest that neither college education (coeff. = 0.027, p-value = 0.64), income (coeff. = -0.0007, p-value = 0.55), nor simplified race (coeff. = 0.0035, p-value = 0.86) significantly influence the likelihood of using blood pressure medication. However, the this model does not show a lot of effectiveness in predicting blood pressure medication use, as the p-values are not very significant.

```
glm_model <- glm(blood_pressure_medication ~ college_educ + income_thousands + simplified_race, data = h
summary(glm_model)
```

```
##
## Call:
## glm(formula = blood_pressure_medication ~ college_educ + income_thousands +
##     simplified_race, family = "binomial", data = heart_disease)
##
```

```
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.6940049  0.1463043 -11.579   <2e-16 ***
## college_educ      0.0272647  0.0589544   0.462    0.644
## income_thousands -0.0007284  0.0012280  -0.593    0.553
## simplified_race   0.0035349  0.0200434   0.176    0.860
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8629.1  on 9999  degrees of freedom
## Residual deviance: 8628.5  on 9996  degrees of freedom
## AIC: 8636.5
##
## Number of Fisher Scoring iterations: 3
```

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

```
# Fit SuperLearner Model

set.seed(44)
library(SuperLearner)
if (!require(gbm)) install.packages("gbm")
```

```
## Loading required package: gbm
```

```
## Loaded gbm 2.1.9
```

```
## This version of gbm is no longer under development. Consider transitioning to gbm3, https://github.co
```

```
library(gbm)
if (!require(caret)) install.packages("caret", dependencies = TRUE)
library(caret)
if (!require(stats)) install.packages("stats")
library(stats)
if (!require(randomForest)) install.packages("randomForest")
```

```
## Loading required package: randomForest
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

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

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(randomForest)
if (!require(nnet)) install.packages("nnet")
```

```
## Loading required package: nnet
```

```
library(nnet)
if (!require(e1071)) install.packages("e1071")
```

```
## Loading required package: e1071
```

```
##
## Attaching package: 'e1071'
```

```
## The following object is masked from 'package:tune':
##
##     tune
```

```
## The following object is masked from 'package:rsample':
##
##     permutations
```

```
## The following object is masked from 'package:parsnip':
##
##     tune
```

```r
library(e1071)
```

```r
## Train/Test split
split <- createDataPartition(y = heart_disease$mortality, p = 0.75, list = FALSE)
train_set <- heart_disease[split, ]
test_set <- heart_disease[-split, ]
```

I use the parallelization method running codes parallel to save time

```r
# Load necessary libraries
library(parallel)
library(doParallel)
```

```
## Loading required package: iterators
```

```r
no_cores <- detectCores() - 1  # reserve one core for system processes

#creating a cluster here
cl <- makeCluster(no_cores)
registerDoParallel(cl)
```

```r
## sl lib AND ## Train SuperLearner

#listWrappers()
```

Looking at the potential models at our disposal (code commented out above, for convenience and computational efficiency), I chose 5 models to include in my SuperLearner library: Model 1: Logistic Regression (SL.glm) Model 2: Random Forest (SL.randomForest) Model 3: Gradient Boosting Machine (SL.gbm) Model 4: Support Vector Machines (SL.svm) Model 5: Neural Network (SL.nnet)

When running those it took a significant amount of time. Thus for the sake of getting through this exercise I continued with a reduced SuperLearner. Ideally I would have run all 5.

```r
# learners <- c("SL.glm", "SL.randomForest", "SL.gbm",
#               "SL.svm", "SL.nnet")

learners <- c("SL.glm", "SL.randomForest")

sl_model <- SuperLearner(Y = train_set$mortality,
                         X = train_set[, -which(names(train_set) == "mortality")],
                         SL.library = learners,
                         family = binomial(),
                         method = "method.NNLS")
summary(sl_model)
```

```
##                   Length Class  Mode
## call                   6 -none- call
## libraryNames           2 -none- character
## SL.library             2 -none- list
## SL.predict          7500 -none- numeric
## coef                   2 -none- numeric
```

```
## library.predict   15000  -none- numeric
## Z                 15000  -none- numeric
## cvRisk                2  -none- numeric
## family               13  family list
## fitLibrary            2  -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     2  -none- logical
## errorsInLibrary       2  -none- logical
## metaOptimizer         8  nnls   list
## env                  10  -none- environment
## times                 3  -none- list
```

The risk values below (GLM: 0.2347; Random Forest: 0.2307) indicate that the Random Forest model is slightly more accurate or this dataset than the GLM. The ensemble risk of 0.2322 is lower than the individual risks of each model. This is an expected outcome as ensemble learning methods balance the strengths of multiple models to account for the weaknesses, improving the overall performance of the prediction.

The weights (GLM: 0.37; Random Forest: 0.63) indicate that the Random Forest model can be considered more reliable for predicting the outcome based on the data used to train the models.

```
print(sl_model)
```

```
##
## Call:
## SuperLearner(Y = train_set$mortality, X = train_set[, -which(names(train_set) ==
##     "mortality")], family = binomial(), SL.library = learners, method = "method.NNLS")
##
##
##
##                           Risk      Coef
## SL.glm_All           0.2347497 0.3730675
## SL.randomForest_All 0.2307031 0.6269325
```

```
## Risk and Coefficient of each model

# Coefficients and Cross-Validated Risks
weights <- sl_model$coef
risks <- sl_model$cvRisk
discrete_winner <- names(which.min(risks))
#Ensemble performance
ensemble_risk <- sum(weights * risks)

print(paste("Weights: ", toString(weights)))
```

```
## [1] "Weights:  0.373067539918813, 0.626932460081186"
```

```r
print(paste("Risks: ", toString(risks)))
```

```
## [1] "Risks:  0.234749709967233, 0.2307031052"
```

```r
print(paste("Discrete Winner: ", discrete_winner))
```

```
## [1] "Discrete Winner:  SL.randomForest_All"
```

```r
print(paste("Ensemble Risk: ", ensemble_risk))
```

```
## [1] "Ensemble Risk:  0.232212762085535"
```

```r
## Confusion Matrix

library(caret)

predictions_prob <- predict(sl_model, newdata = test_set[, -which(names(test_set) == "mortality")], typ
```

```r
str(predictions_prob)
```

```
## List of 2
##  $ pred           : num [1:2500, 1] 0.538 0.577 0.6 0.596 0.535 ...
##  $ library.predict: num [1:2500, 1:2] 0.563 0.562 0.554 0.545 0.585 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:2] "SL.glm_All" "SL.randomForest_All"
```

```r
final_predictions <- predictions_prob$pred
binarized_predictions <- ifelse(final_predictions > 0.5, 1, 0)

library(caret)
conf_matrix <- confusionMatrix(as.factor(binarized_predictions), as.factor(test_set$mortality))

accuracy <- conf_matrix$overall['Accuracy']
recall <- conf_matrix$byClass['Sensitivity']
precision <- conf_matrix$byClass['Pos Pred Value']

print(conf_matrix$table)
```

```
##           Reference
## Prediction    0    1
##          0  369  216
##          1  874 1041
```

```r
cat(sprintf("\nAccuracy: %.2f%%\n", accuracy * 100))
```

```
##
## Accuracy: 56.40%
```

```r
cat(sprintf("Recall (Sensitivity): %.2f%%\n", recall * 100))
```

```
## Recall (Sensitivity): 29.69%
```

```r
cat(sprintf("Precision (Positive Predictive Value): %.2f%%\n", precision * 100))
```

```
## Precision (Positive Predictive Value): 63.08%
```

The low accuracy and recall suggest that the model potentially does not capture all of the relevant variables, or the variables included might not have a strong enough relationship with the treatment effectiveness. The results suggest that predicting treatment effectiveness with the current set of predictors is challenging and their effects on each other are not fully captured by the model.

```r
## 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)?

Ensemble learning methods, such as the SuperLearner model, ensure better generalizability by combining different models (Ganaie et al., 2022). Generally, ensemble methods reduce the variance of prediction errors made by any one of the models, leading to reduction in the spread in the avg. skill of a predictive model and improvement of the avg. prediction performance over any contributing member in the ensemble.

Ganaie, M. A., Hu, M., Malik, A. K., Tanveer, M., & Suganthan, P. N. (2022). Ensemble deep learning: A review. Engineering Applications of Artificial Intelligence, 115, 105151-. https://doi.org/10.1016/j.engappai.2022.105151

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
```

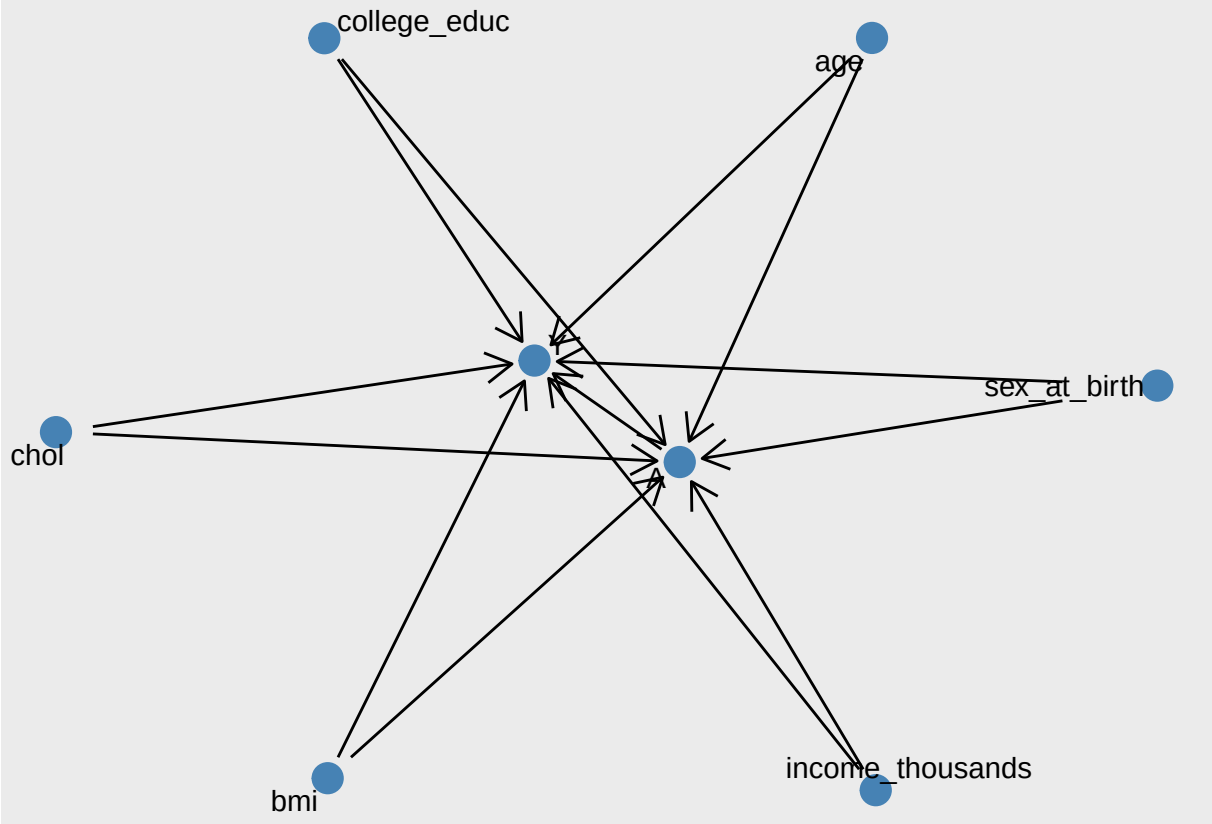
```r
if (!require(ggdag)) install.packages("ggdag")
if (!require(dagitty)) install.packages("dagitty")
if (!require(dagitty)) {
    install.packages("dagitty")
    library(dagitty)
}

library(dagitty)
library(ggplot2)
library(dplyr)
library(ggdag)
```

##Pretty DAG

```r
pretty_dag <- function(dag) {
  nodes <- unique(dag$data$name)
  old_x <- unique(dag$data$x)
  old_y <- unique(dag$data$y)
  num_nodes <- length(nodes)
  num_ws <- sum(str_detect(nodes, "^(age|sex|bmi|chol|income_thousands)$"))

  for(i in 1:num_nodes) {
    if (nodes[i] == "A") {
      new_y <- 0
      new_x <- 0
    } else if (nodes[i] == "Y") {
      new_y <- 0
      new_x <- 2 * (num_ws + 1)
    } else if (str_detect(nodes[i], "^W")) {
      w_num <- as.numeric(str_remove(nodes[i], "W"))
      new_y <- 2
      new_x <- 2 * (w_num + 1)
    } else {
      # Default case for any other nodes that might be unaccounted
      new_y <- 4
      new_x <- 2
    }

    # Apply the new coordinates
    dag$data <- dag$data %>%
      mutate(x = replace(x, x == old_x[i], new_x),
             y = replace(y, y == old_y[i], new_y),
             xend = replace(xend, xend == old_x[i], new_x),
             yend = replace(yend, yend == old_y[i], new_y))
  }

  # Recolor nodes based on type for better visualization
  dag <- dag %>%
    mutate(color = case_when(
      str_detect(name, "^[A|Y]$") ~ name,
      str_detect(name, "^(age|sex|bmi|chol|income_thousands)$") ~ "W",
      TRUE ~ "U"
    ), circular = TRUE)
```

```
  return(dag)
}
```

```
library(ggraph)

# DAG formula definition as preparation for the TMLE
dag_formula <- dagify(
  Y ~ A + age + sex_at_birth + bmi + chol + income_thousands + college_educ,
  A ~ age + sex_at_birth + bmi + chol + income_thousands + college_educ,
  labels = c(Y = "Mortality", A = "Blood Pressure Medication",
             age = "Age", sex_at_birth = "Sex at Birth", bmi = "BMI",
             chol = "Cholesterol Level", income_thousands = "Income in Thousands",
             college_educ = "College Education")
)

# convert DAG formula to object
dag_object <- tidy_dagitty(dag_formula)

ggraph(dag_object, layout = "stress") +  # You can try different layouts like "circle", "stress", "kk",
  geom_edge_link(aes(start_cap = label_rect(node1.name), end_cap = label_rect(node2.name)), arrow = arr
  geom_node_point(color = 'steelblue', size = 5) +
  geom_node_text(aes(label = name), repel = TRUE)  # Ensures text doesn't overlap
```



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 findings (below) show an estimated mean outcome under treatment at 0.23 with a confidence interval of (0.22, 0.25), suggesting high precision level in the estimate. Under No Treatment the mean outcome is significantly higher at 0.57, with a similarly tight confidence interval.

The overall treatment effect is -0.33, indicating that the treatment reduces the outcome measure by about 33.3% compared to no treatment. This effect is statistically significant with a narrow confidence interval of (-0.35, -0.31).

Without the treatment, the outcome is only 23% likely to occur.

Comparing the TMLW findings to the SuperLearner findings:

The results of the SuperLearner indicated that the Random Forest was slightly more effective. The coefficients in the TMLE model suggest that 'SL.rpart_All' has significant weight in predicting the outcome.

While SuperLearner focuses on prediction risk and model accuracy, TMLE provides a causal inference perspective, estimating the effect of treatment on outcomes, which is a more direct measure of treatment effectiveness.

TMLE provides a robust estimation of treatment effects with statistical significance. It complements the predictive insights from SuperLearner, which identifies the most effective predictive models.

The negative ATE indicates a beneficial effect of the treatment in reducing the probability of the adverse outcome. The high statistical significance and narrow confidence intervals across TMLE estimates provide strong support for the treatment's effectiveness.

Model Reliability: The consistency in statistical significance and effect sizes across different estimates and subgroups in the TMLE analysis underscores the reliability of the findings, while the SuperLearner's ensemble approach optimizes prediction accuracy. Combined, these models are highly effective for prediction as well as causal inference.

```
#1
library(SuperLearner)
library(tmle)

sl_libs <- c("SL.mean", "SL.glm", "SL.glmnet", "SL.rpart", "SL.randomForest")

sl <- SuperLearner(Y = heart_disease$mortality, X = heart_disease[, c("age", "sex_at_birth", "bmi", "cho
                   newX = NULL, family = binomial(), SL.library = sl_libs, method = "method.NNloglik")
```

```
#2
sl_simpler <- SuperLearner(Y = heart_disease$mortality, X = heart_disease[, c("age", "chol")],
                           newX = NULL, family = binomial(), SL.library = sl_libs, method = "method.NNlo
```

```r
#3
library(tmle)

Q.SL.library <- sl_libs
g.SL.library <- sl_libs
tmle_fit <- tmle(Y = heart_disease$mortality,
                 A = heart_disease$blood_pressure_medication,
                 W = heart_disease[, c("age", "chol")],
                 family = "binomial",
                 Q.SL.library = Q.SL.library,
                 g.SL.library = g.SL.library)
summary(tmle_fit)
```

```
##  Initial estimation of Q
##   Procedure: cv-SuperLearner, ensemble
##   Model:
##       Y ~  SL.mean_All + SL.glm_All + SL.glmnet_All + SL.rpart_All + SL.randomForest_All
##
##   Coefficients:
##       SL.mean_All    0
##        SL.glm_All    0.02938674
##     SL.glmnet_All    0
##      SL.rpart_All    0.9653669
##   SL.randomForest_All    0.005246311
##
##   Cross-validated pseudo R squared :  0.0684
##
##  Estimation of g (treatment mechanism)
##   Procedure: SuperLearner, ensemble
##   Model:
##       A ~  SL.mean_All + SL.glm_All + SL.glmnet_All + SL.rpart_All + SL.randomForest_All
##
##   Coefficients:
##       SL.mean_All    0
##        SL.glm_All    0.8982216
##     SL.glmnet_All    0
##      SL.rpart_All    0.05901825
##   SL.randomForest_All    0.04276014
##
##  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/ATC: (0.0054, 0.9946)
##
##  Marginal Mean under Treatment (EY1)
##    Parameter Estimate:  0.23368
##    Estimated Variance:  6.6069e-05
##               p-value:  <2e-16
```

```
##     95% Conf Interval:  (0.21775, 0.24961)
##
##  Marginal Mean under Comparator (EY0)
##    Parameter Estimate:  0.56573
##    Estimated Variance:  2.8805e-05
##               p-value:  <2e-16
##     95% Conf Interval:  (0.55522, 0.57625)
##
##  Additive Effect
##    Parameter Estimate:  -0.33206
##    Estimated Variance:  9.4702e-05
##               p-value:  <2e-16
##     95% Conf Interval:  (-0.35113, -0.31298)
##
##  Additive Effect among the Treated
##    Parameter Estimate:  -0.31739
##    Estimated Variance:  0.00014215
##               p-value:  <2e-16
##     95% Conf Interval:  (-0.34075, -0.29402)
##
##  Additive Effect among the Controls
##    Parameter Estimate:  -0.33519
##    Estimated Variance:  9.1369e-05
##               p-value:  <2e-16
##     95% Conf Interval:  (-0.35393, -0.31646)
##
##  Relative Risk
##    Parameter  Estimate:  0.41305
##    Variance(log scale):  0.0012986
##                p-value:  <2e-16
##      95% Conf Interval:  (0.38488, 0.44328)
##
##  Odds Ratio
##     Parameter  Estimate:  0.23407
##     Variance(log scale):  0.0025337
##                 p-value:  <2e-16
##       95% Conf Interval:  (0.21208, 0.25834)
```

```
cat("Estimated ATE:", tmle_fit$estimates$ATE$psi, "\n")
```

```
## Estimated ATE: -0.3320569
```

```
cat("Standard Error:", tmle_fit$estimates$ATE$se.psi, "\n")
```

```
## Standard Error:
```

```
cat("95% Confidence Interval: [", tmle_fit$estimates$ATE$CI.lower, ", ", tmle_fit$estimates$ATE$CI.upper
```

```
## 95% Confidence Interval: [ ,  ]
```

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.

A double robust estimator provides consistent estimates of treatment effects as long as at least either the outcome model or the propensity score model is correctly specified, thus mitigating the risk of biased results from incorrect model assumptions. The process behind double robust estimators is: They use two models. One estimates the outcome conditioned on treatment and covariates. The other estimates the probability of receiving the treatment given the covariates (propensity score model). The models complement each other's weaknesses with their strengths. If one of the models fails, the other will provide the necessary results. This approach helps improve on confidence degrees.

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.

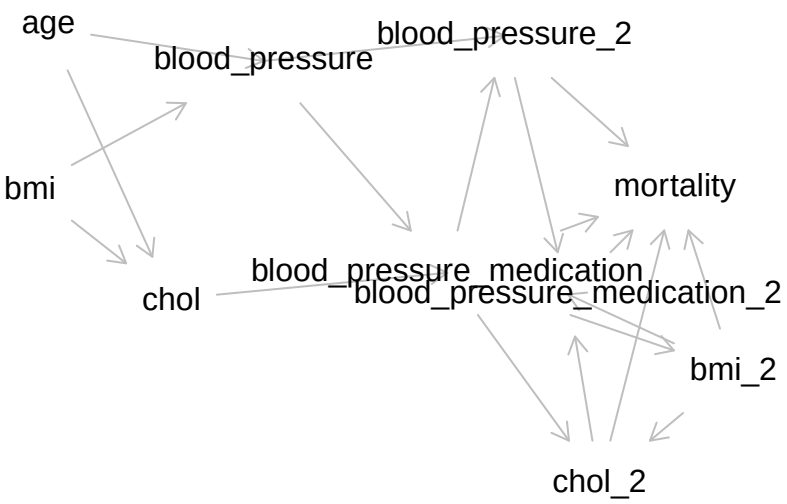
Hint: Check out slide 27 from Maya's lecture, or slides 15-17 from Dave's second slide deck in week 8 on matching.

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
dag <- dagitty('dag {
"age" -> "blood_pressure";
"age" -> "chol";

"sex_at_birth";

"bmi" -> "blood_pressure";
"bmi" -> "chol";

"bmi_2" -> "blood_pressure_medication_2";
"bmi_2" -> "mortality";
"bmi_2" -> "chol_2";

"blood_pressure" -> "blood_pressure_medication";
"blood_pressure" -> "blood_pressure_2";
```

```
"blood_pressure_2" -> "blood_pressure_medication_2";
"blood_pressure_2" -> "mortality";

"chol" -> "blood_pressure_medication";

"chol_2" -> "blood_pressure_medication_2"
"chol_2" -> "mortality"

"blood_pressure_medication" -> "bmi_2";
"blood_pressure_medication" -> "blood_pressure_2";
"blood_pressure_medication" -> "chol_2"
"blood_pressure_medication" -> "mortality"

"blood_pressure_medication_2" -> "mortality";
}')

plot(dag)
```

```
## Plot coordinates for graph not supplied! Generating coordinates, see ?coordinates for how to set your
```



sex_at_birth

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?

The 1. result shows an estimate of 0.2, indicating a moderate treatment effect.

The 2. result shows an estimate of approximately 0.16, suggesting a slightly smaller effect size than the first model. This may indicate less efficacy

The 3. result of the estimate of 1 could potentially indicate a directly causal relation. This is unusually high and might suggest an error.

Comparison with TMLE Results: TMLE generally provides a point estimate for the treatment effect at a single time, without accounting for changes over time. The LTMLE estimates vary more significantly across different models compared to TMLE estimates, which might suggest different dynamics captured in longitudinal settings. LTMLE's capacity to incorporate multiple treatments over time (as in the third call) offers a dynamic understanding of treatment effects which TMLE might not capture directly. The third LTMLE result shows the potential for treatment effects to change significantly over time. The third estimate suggests that initial treatment alone may not be as effective as subsequent or combined treatments, which is vital for treatment planning and patient management. Compared to standard TMLE results, LTMLE offers a more dynamic perspective relevant for clinical trials which are longitudinal.

```r
# Load the necessary library
library(ltmle)

data_obs_ltmle <- heart_disease %>%
rename(W1 = age, W2 = chol, W3 = bmi) %>%
select(W1, W2, W3, A = blood_pressure_medication, Y = mortality)
result1 <- ltmle(data = data_obs_ltmle,
                 Anodes = "A",
                 Ynodes = "Y",
                 abar = 1)
```

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

## formula for Y:

## Q.kplus1 ~ W1 + W2 + W3 + A

##

## gform not specified, using defaults:

## formula for A:

## A ~ W1 + W2 + W3

##

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

```r
rexpit <- function(x) rbinom(n=length(x), size=1, prob=plogis(x))
n <- 1000
W1 <- rnorm(n)
W2 <- rbinom(n, size=1, prob=0.3)
W3 <- rnorm(n)
A <- rexpit(0.01 * W1 - 0.05 * W2 + 0.01 * W3 - 2)
Y <- rexpit(-1 + 0.02 * W1 + 0.04 * W2 - 0.6 * A + 0.03 * W3 * A - 0.8 * W3)
data <- data.frame(W1, W2, W3, A, Y)
result2 <- ltmle(data, Anodes="A", Lnodes=NULL, Ynodes="Y", abar=1, SL.library=sl_libs)
```

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

## formula for Y:

## Q.kplus1 ~ W1 + W2 + W3 + A

##

## gform not specified, using defaults:

## formula for A:

## A ~ W1 + W2 + W3

##

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

```r
#Longitudinal data structure
n <- 1000
W <- rnorm(n, mean = 200, sd = 50)  # Cholesterol levels
A1 <- rexpit(0.01 * W - 3)
L <- 0.05 * W - 0.1 * A1 + rnorm(n, mean = 25, sd = 5)
A2 <- rexpit(-0.02 * W + 0.3 * A1 + 0.1 * L - 2)
Y <- rexpit(0.03 * W - 0.5 * A1 + 0.15 * L - 0.8 * A2 - 1)

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

result3 <- ltmle(data, Anodes=c("A1", "A2"), Lnodes="L", Ynodes="Y", abar=c(1, 1), SL.library = sl_libs
```

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

## formula for L:

## Q.kplus1 ~ W + A1

## formula for Y:

## Q.kplus1 ~ W + A1 + L + A2

##

## gform not specified, using defaults:

## formula for A1:

## A1 ~ W

## formula for A2:
```

```
## A2 ~ W + A1 + L

##

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

```
if (exists("summary.ltmle")) {
  print(summary(result1))
  print(summary(result2))
  print(summary(result3))
} else {
  print(result1)
  print(result2)
  print(result3)
}
```

```
## Call:
## ltmle(data = data_obs_ltmle, Anodes = "A", Ynodes = "Y", abar = 1)
##
## TMLE Estimate:  0.2039971
## Call:
## ltmle(data = data, Anodes = "A", Lnodes = NULL, Ynodes = "Y",
##     abar = 1, SL.library = sl_libs)
##
## TMLE Estimate:  0.1950373
## Call:
## ltmle(data = data, Anodes = c("A1", "A2"), Lnodes = "L", Ynodes = "Y",
##     abar = c(1, 1), SL.library = sl_libs)
##
## TMLE Estimate:  1
```

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?

Time-dependent confounding is a phenomenon (in longitudinal studies) where variables that change over time influence both the treatment and the outcome. This is particularly critical for variables like blood pressure, which can be affected by prior treatments and subsequently affect future treatment decisions and health outcomes. To the contrary, for example age while changing over time, typically does not act as a time-dependent confounder, as it is not influenced by the treatment. It is a fixed variable progressing uniformly.