HarvardX Data Science Capstone Final

Daniel Yoo

3/6/2020

Contents

1 Introduction

This is a part of the last course of HarvardX's Data Science Professional Certificate series. We used a publicly available dataset from UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/index.php. The selected dataset includes patient records collected from North East of Andhra Pradesh, India. Patients with liver disease have been continuously increasing because of excessive consumption of alcohol, inhale of harmful gases, intake of contaminated food, pickles and drugs. In this study, we aim to predict which patients have liver disease and which ones do not by using the patient records in an effort to reduce burden on doctors.

2 Population

First, we load required packages and the dataset for our analysis.

```r
# install and load packages
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(dplyr)) install.packages("dplyr", repos = "http://cran.us.r-project.org")
if(!require(ggplot2)) install.packages("ggplot2", repos = "http://cran.us.r-project.org")
if(!require(corrplot)) install.packages("corrplot", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(randomForest)) install.packages("randomForest", repos = "http://cran.us.r-project.org")
if(!require(xgboost)) install.packages("xgboost", repos = "http://cran.us.r-project.org")
if(!require(plyr)) install.packages("plyr", repos = "http://cran.us.r-project.org")
if(!require(klaR)) install.packages("klaR", repos = "http://cran.us.r-project.org")
if(!require(caretEnsemble)) install.packages("caretEnsemble", repos = "http://cran.us.r-project.org")
if(!require(kableExtra)) install.packages("kableExtra", repos = "http://cran.us.r-project.org")
if(!require(compareGroups)) install.packages("compareGroups", repos = "http://cran.us.r-project.org")
if(!require(gridExtra)) install.packages("gridExtra", repos = "http://cran.us.r-project.org")
if(!require(stepPlr)) install.packages("stepPlr", repos = "http://cran.us.r-project.org")
if(!require(yardstick)) install.packages("yardstick", repos = "http://cran.us.r-project.org")
```

```
# load the dataset from github repo
liver_data <- read.csv("https://raw.githubusercontent.com/udaniel/capstone_DS_liver/master/indian_liver_
```

We look at the basic structure of the given data.

```
# basic structure of the dataset
liver_data %>% dim()
```

```
## [1] 583  11
```

```
liver_data %>% head()
```

```
##   Age Gender Total_Bilirubin Direct_Bilirubin Alkaline_Phosphotase
## 1  65 Female             0.7              0.1                  187
## 2  62   Male            10.9              5.5                  699
## 3  62   Male             7.3              4.1                  490
## 4  58   Male             1.0              0.4                  182
## 5  72   Male             3.9              2.0                  195
## 6  46   Male             1.8              0.7                  208
##   Alamine_Aminotransferase Aspartate_Aminotransferase Total_Protiens
## 1                       16                         18            6.8
## 2                       64                        100            7.5
## 3                       60                         68            7.0
## 4                       14                         20            6.8
## 5                       27                         59            7.3
## 6                       19                         14            7.6
##   Albumin Albumin_and_Globulin_Ratio Dataset
## 1     3.3                       0.90       1
## 2     3.2                       0.74       1
## 3     3.3                       0.89       1
## 4     3.4                       1.00       1
## 5     2.4                       0.40       1
## 6     4.4                       1.30       1
```

```
str(liver_data)
```

```
## 'data.frame':    583 obs. of  11 variables:
##  $ Age                       : int  65 62 62 58 72 46 26 29 17 55 ...
##  $ Gender                    : Factor w/ 2 levels "Female","Male": 1 2 2 2 2 2 1 1 2 2 ...
##  $ Total_Bilirubin           : num  0.7 10.9 7.3 1 3.9 1.8 0.9 0.9 0.9 0.7 ...
##  $ Direct_Bilirubin          : num  0.1 5.5 4.1 0.4 2 0.7 0.2 0.3 0.3 0.2 ...
##  $ Alkaline_Phosphotase      : int  187 699 490 182 195 208 154 202 202 290 ...
##  $ Alamine_Aminotransferase  : int  16 64 60 14 27 19 16 14 22 53 ...
##  $ Aspartate_Aminotransferase: int  18 100 68 20 59 14 12 11 19 58 ...
##  $ Total_Protiens            : num  6.8 7.5 7 6.8 7.3 7.6 7 6.7 7.4 6.8 ...
##  $ Albumin                   : num  3.3 3.2 3.3 3.4 2.4 4.4 3.5 3.6 4.1 3.4 ...
##  $ Albumin_and_Globulin_Ratio: num  0.9 0.74 0.89 1 0.4 1.3 1 1.1 1.2 1 ...
##  $ Dataset                   : int  1 1 1 1 1 1 1 1 2 1 ...
```

We have 583 patients, 11 variables. Among them, 10 are independent variables and one variable is the outcome variable.

Columns include:

- Age of the patient
- Gender of the patient
- Total Bilirubin
- Direct Bilirubin
- Alkaline Phosphotase
- Alamine Aminotransferase
- Aspartate Aminotransferase
- Total Proteins
- Albumin
- Albumin and Globulin Ratio
- Dataset: field used to split the data into two sets (patient with liver disease, or no disease)

The final "Dataset" variable is our outcome variable. Since we only have 10 independent variables, the feature selection process would be redundant. From the dataset source https://www.kaggle.com/uciml/indian-liver-patient-records/data, we can acknowledge that any patient whose age exceeded 89 is listed as being of age 90.

3 Exploratory Data Analysis

Before proceed, we should clean the dataset little to make it more readable.

```
# properly factor the outcome variable
liver_data$Dataset <- factor(liver_data$Dataset, levels = c(1, 2),
                             labels = c("disease", "no_disease"))
# change outcome name
liver_data <-
    liver_data %>%
    dplyr::rename(outcome = Dataset)
# fix the misspelling
liver_data <-
    liver_data %>%
    dplyr::rename(Total_Proteins = Total_Protiens)
```

3.1 Univariate Analysis

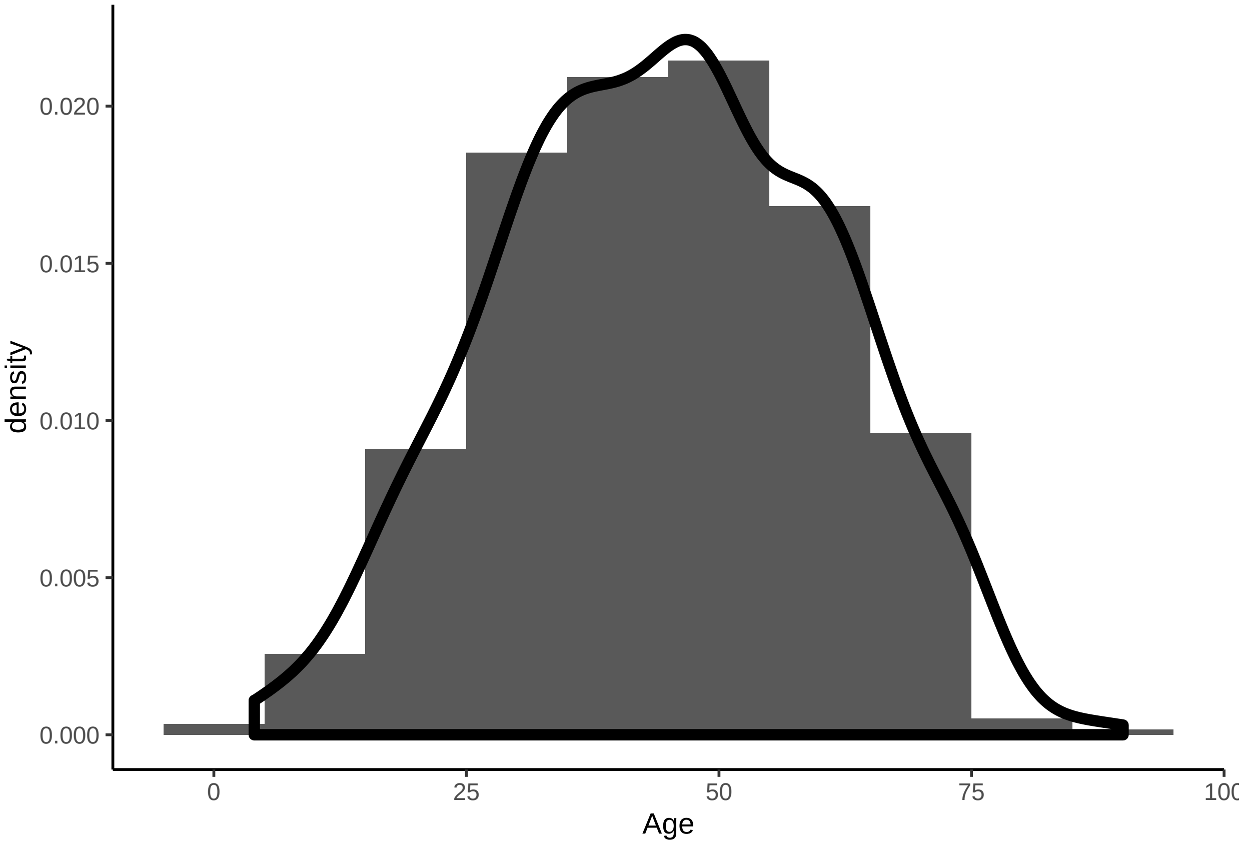
```
# summary
summary(liver_data)
```

```
##       Age           Gender    Total_Bilirubin  Direct_Bilirubin
##  Min.   : 4.00   Female:142   Min.   : 0.400   Min.   : 0.100
##  1st Qu.:33.00   Male  :441   1st Qu.: 0.800   1st Qu.: 0.200
##  Median :45.00                Median : 1.000   Median : 0.300
##  Mean   :44.75                Mean   : 3.299   Mean   : 1.486
##  3rd Qu.:58.00                3rd Qu.: 2.600   3rd Qu.: 1.300
##  Max.   :90.00                Max.   :75.000   Max.   :19.700
##
##  Alkaline_Phosphotase Alamine_Aminotransferase Aspartate_Aminotransferase
##  Min.   :  63.0       Min.   :  10.00          Min.   :  10.0
##  1st Qu.: 175.5       1st Qu.:  23.00          1st Qu.:  25.0
##  Median : 208.0      Median :  35.00         Median :  42.0  
##  Mean   : 290.6      Mean   :  80.71         Mean   : 109.9  
##  3rd Qu.: 298.0      3rd Qu.:  60.50         3rd Qu.:  87.0  
##  Max.   :2110.0      Max.   :2000.00         Max.   :4929.0  
##                                                              
##  Total_Proteins     Albumin      Albumin_and_Globulin_Ratio
##  Min.   :2.700   Min.   :0.900   Min.   :0.3000            
##  1st Qu.:5.800   1st Qu.:2.600   1st Qu.:0.7000            
##  Median :6.600   Median :3.100   Median :0.9300            
##  Mean   :6.483   Mean   :3.142   Mean   :0.9471            
##  3rd Qu.:7.200   3rd Qu.:3.800   3rd Qu.:1.1000            
##  Max.   :9.600   Max.   :5.500   Max.   :2.8000            
##                                  NA's   :4                 
##        outcome   
##  disease   :416  
##  no_disease:167  
##                  
##                  
##                  
##                  
## 
```

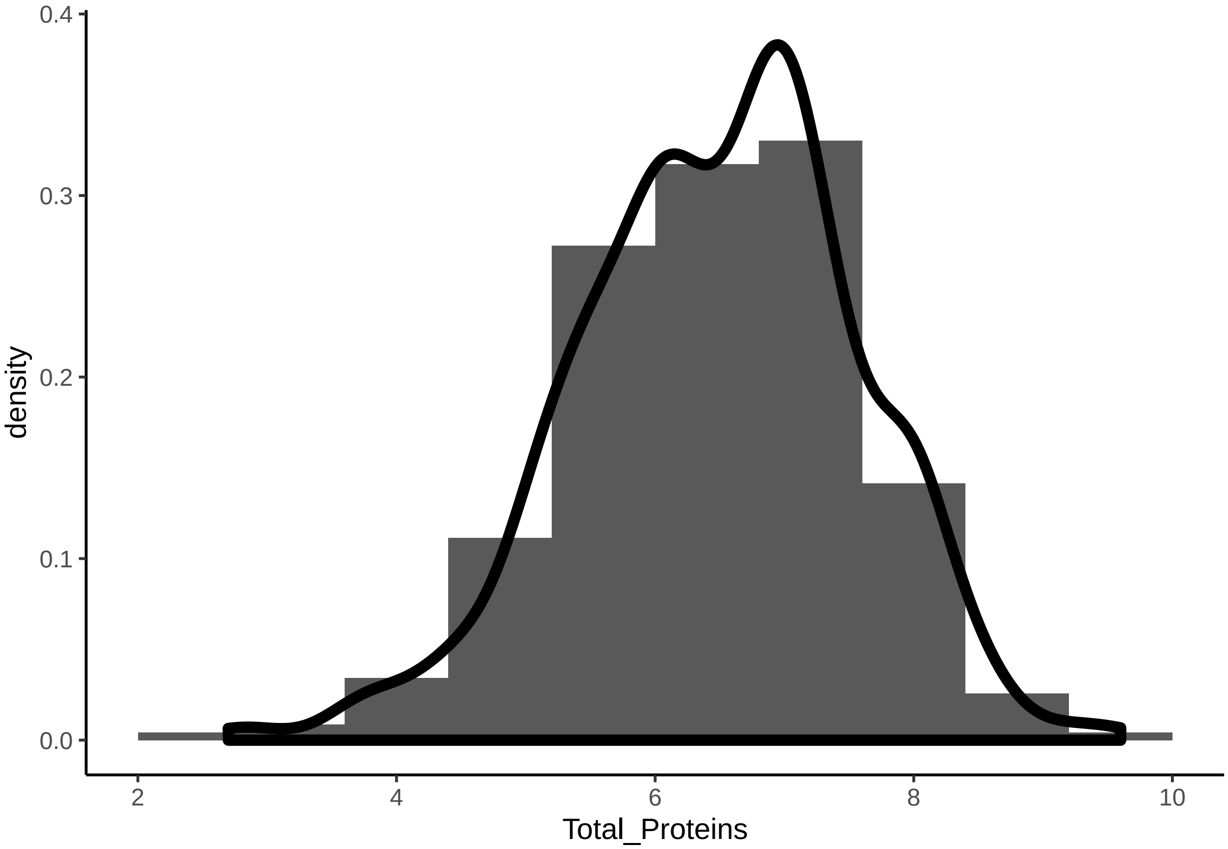
There are 142 female patient records and 441 male patient records. The minimum age of the recorded patient is 4 and 90 for the maximum since we considered all patients with older than 90 years old 90. Total bilirubin, direct bilirubin, alkaline phosphotase, alamine aminotransferase and aspartate aminotransferase look normally distributed until the 3rd quartiles. The maximum values of these variables are highly skewed; it is worth to examine those patient records. Total proteins and albumin variables look normally distributed. Albumin and globulin ratio variable has 4 NAs. We have 416 liver patient records and 167 non liver patient records.

Let's look at histograms and density plots for numerical variables. First, we observe presumaly normally distributed numerical variables.

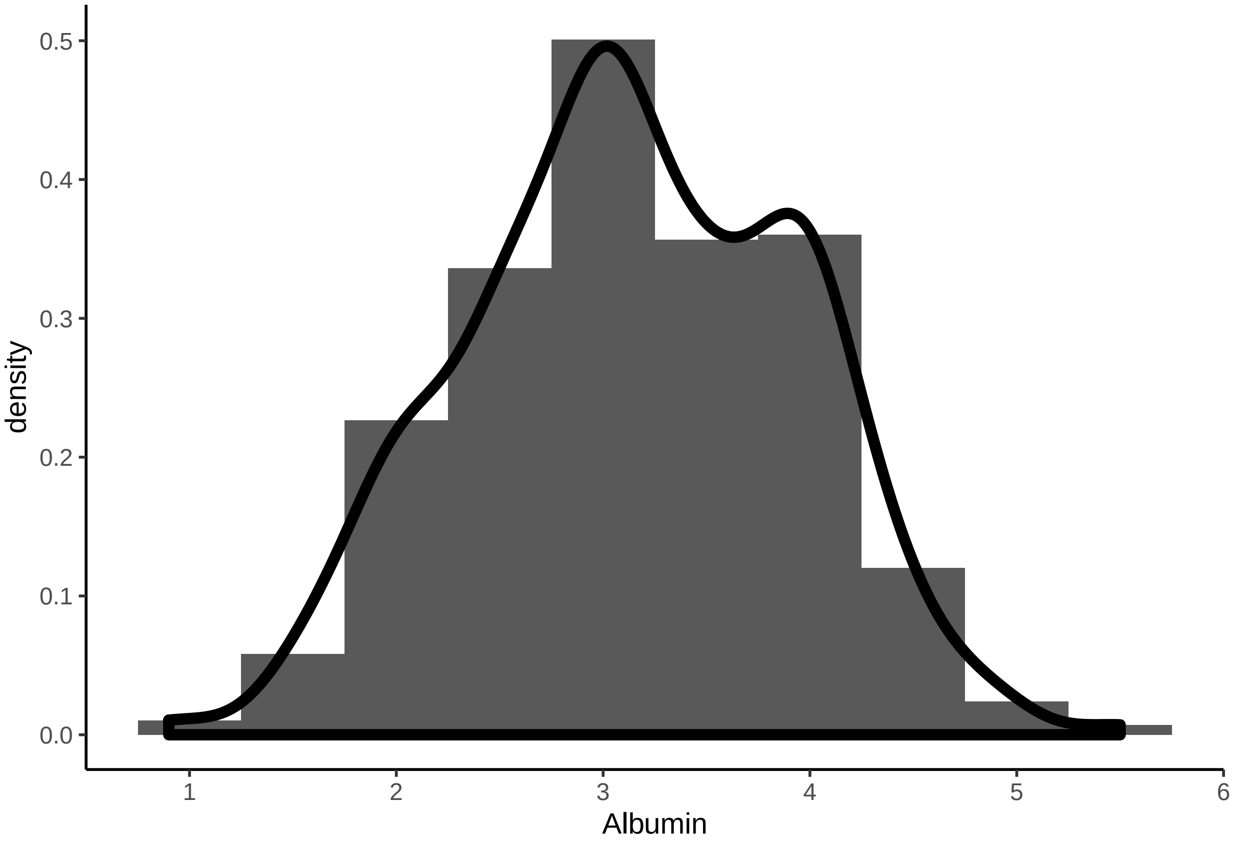
```
# age distribution
liver_data %>%
    ggplot(aes(x = Age)) +
    geom_histogram(aes(y = ..density..), binwidth = 10) +
    geom_density(lwd = 2) +
    theme_classic()
```



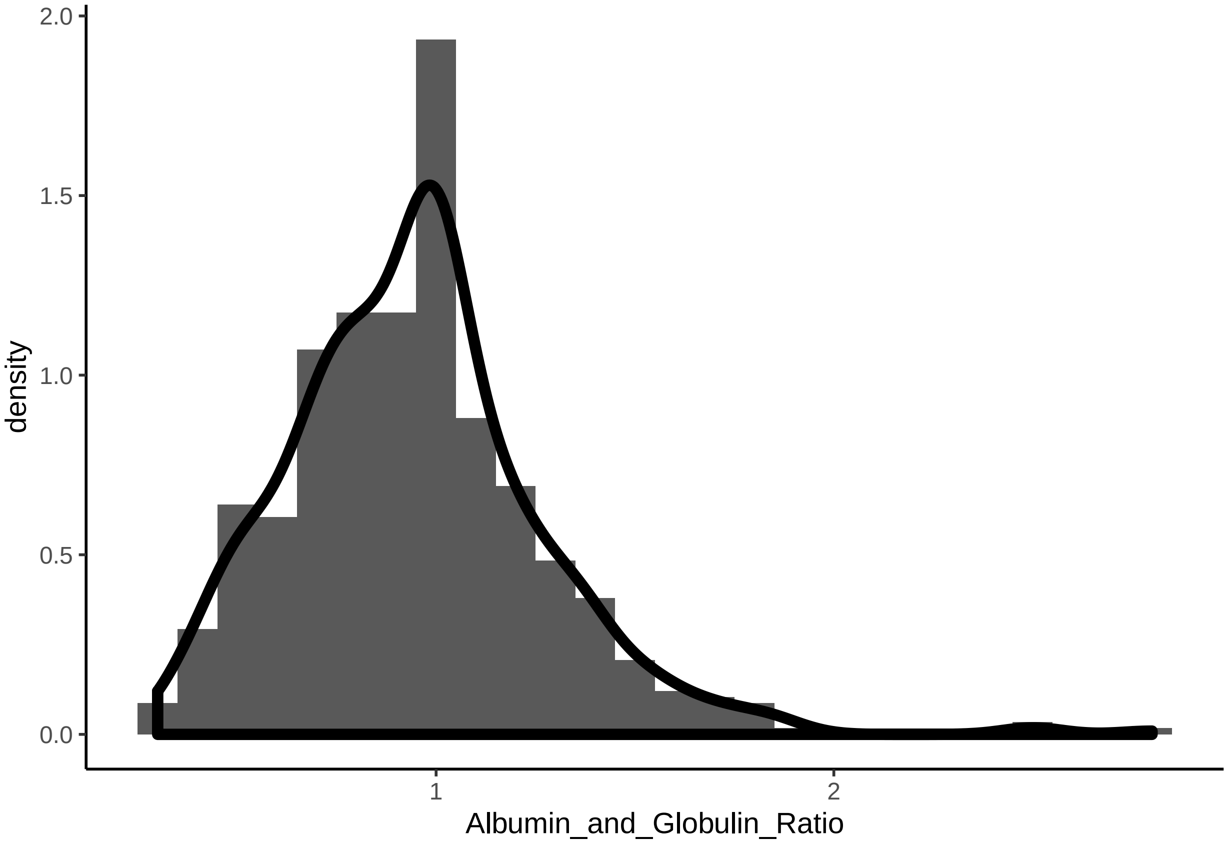
```r
# Total_Proteins distribution
liver_data %>%
    ggplot(aes(x = Total_Proteins)) +
    geom_histogram(aes(y = ..density..), binwidth = 0.8) +
    geom_density(lwd = 2) +
    theme_classic()
```



```
# Albumin distribution
liver_data %>%
    ggplot(aes(x = Albumin)) +
    geom_histogram(aes(y = ..density..), binwidth = 0.5) +
    geom_density(lwd = 2) +
    theme_classic()
```

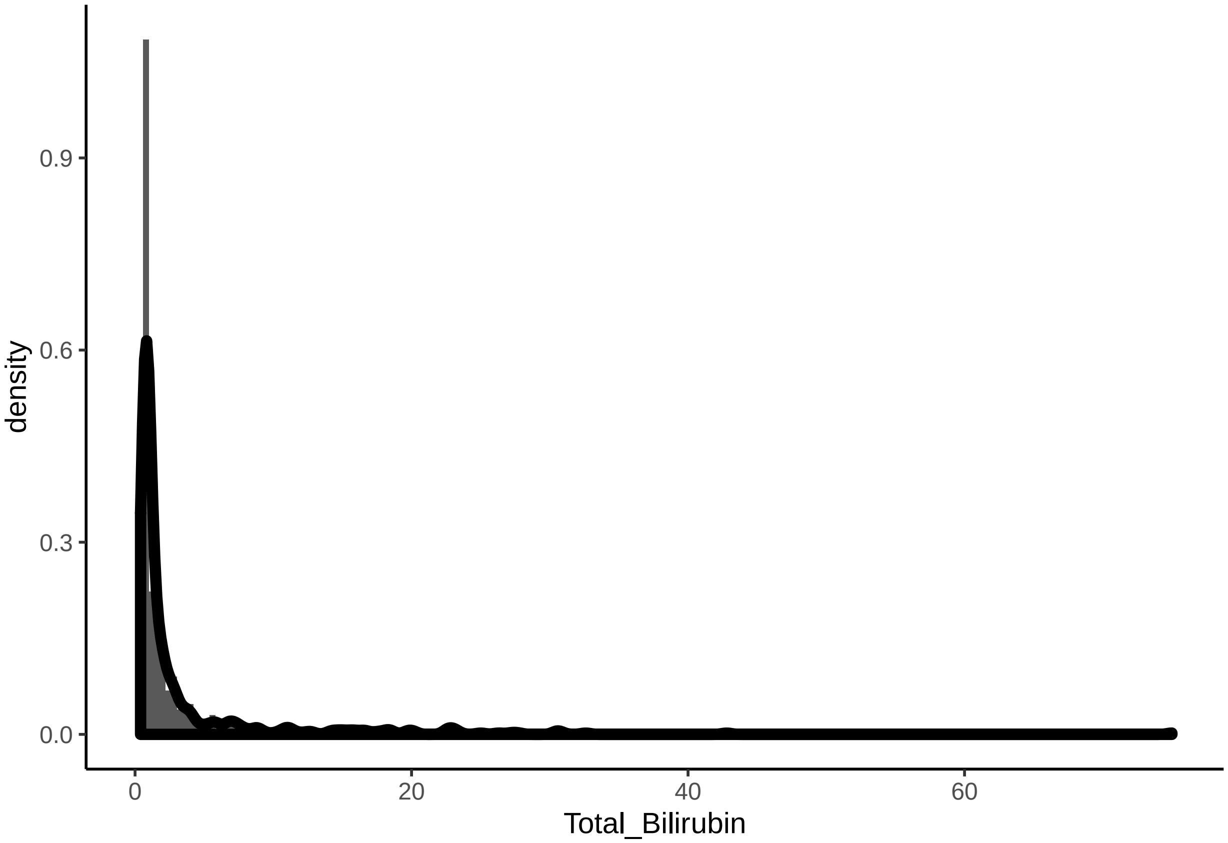


```
# Albumin_and_Globulin_Ratio distribution
liver_data %>%
    ggplot(aes(x = Albumin_and_Globulin_Ratio)) +
    geom_histogram(aes(y = ..density..), binwidth = 0.1) +
    geom_density(lwd = 2) +
    theme_classic()
```

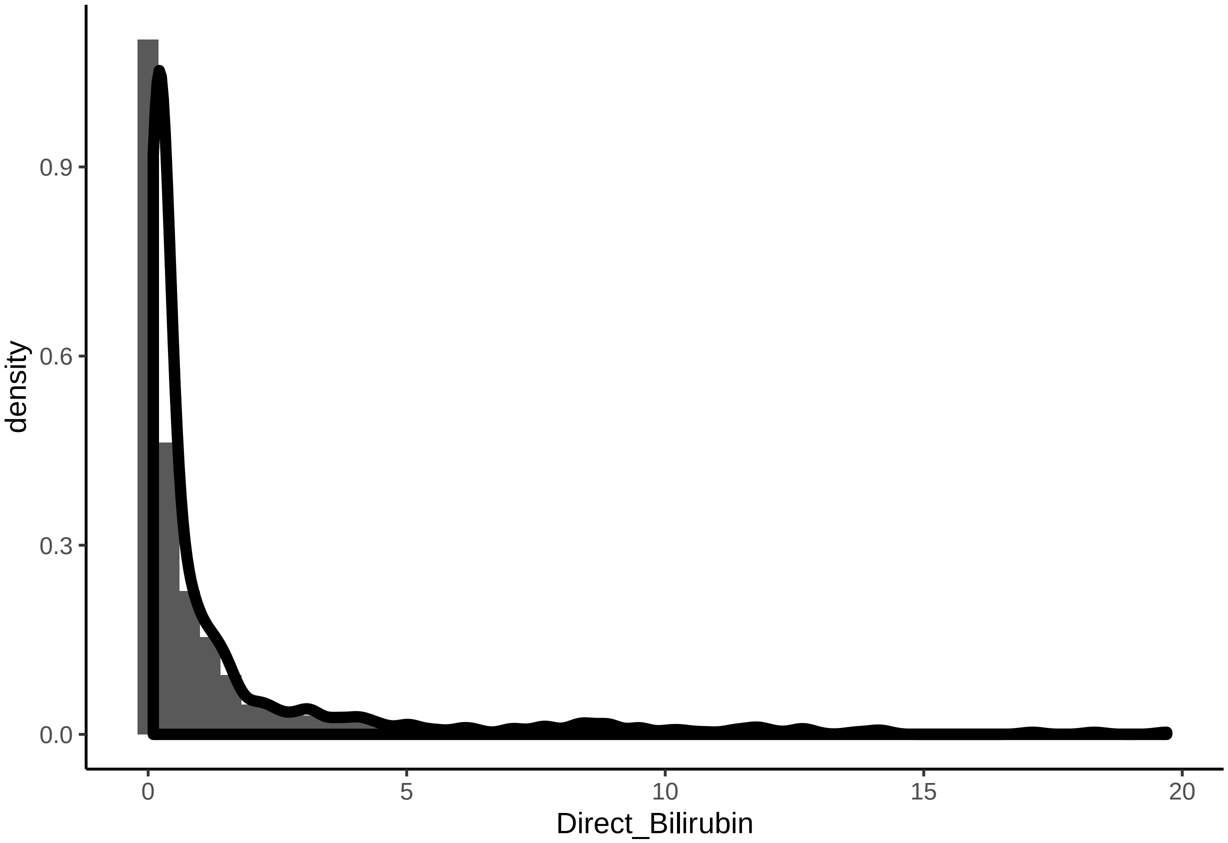


We can see that Albumin_and_Globulin_Ratio variable has some outliers.

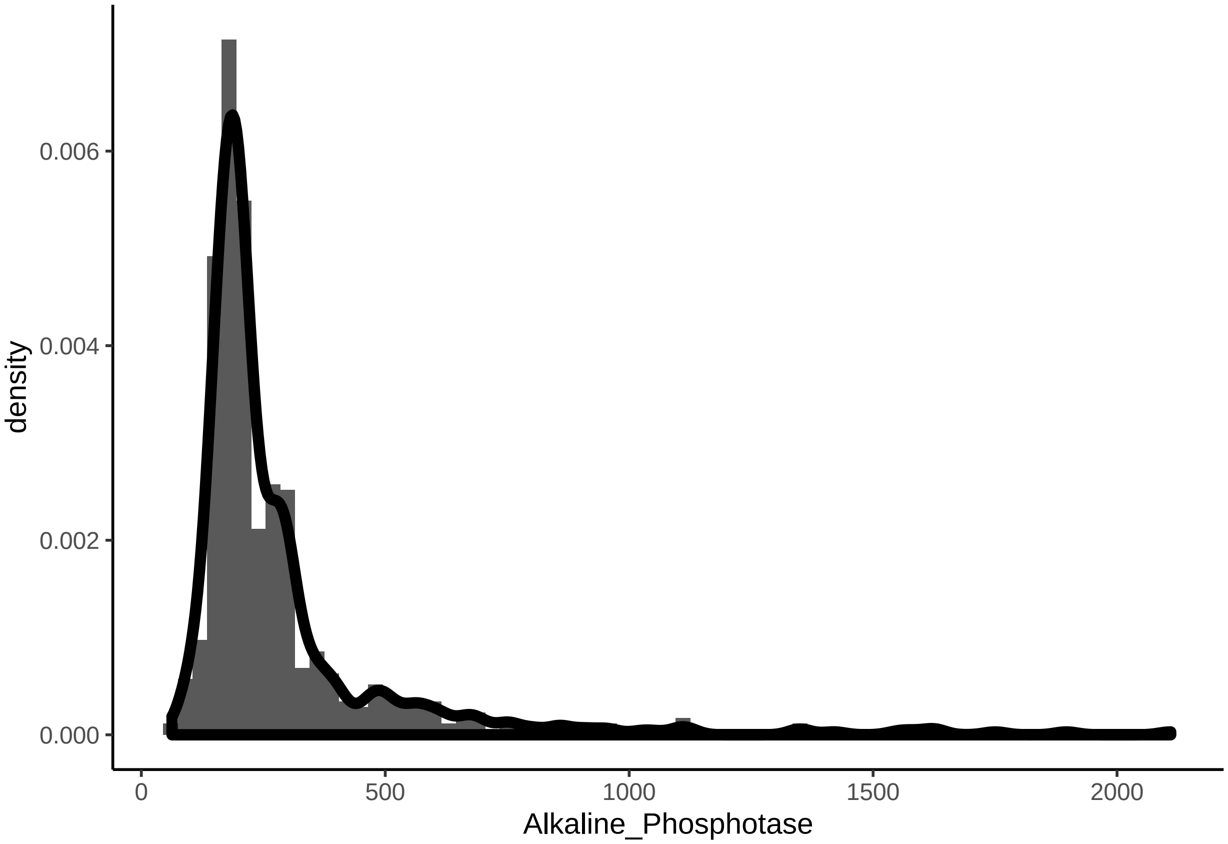
```
# Total_Bilirubin distribution
liver_data %>%
    ggplot(aes(x = Total_Bilirubin)) +
    geom_histogram(aes(y = ..density..), binwidth = 0.4) +
    geom_density(lwd = 2) +
    theme_classic()
```



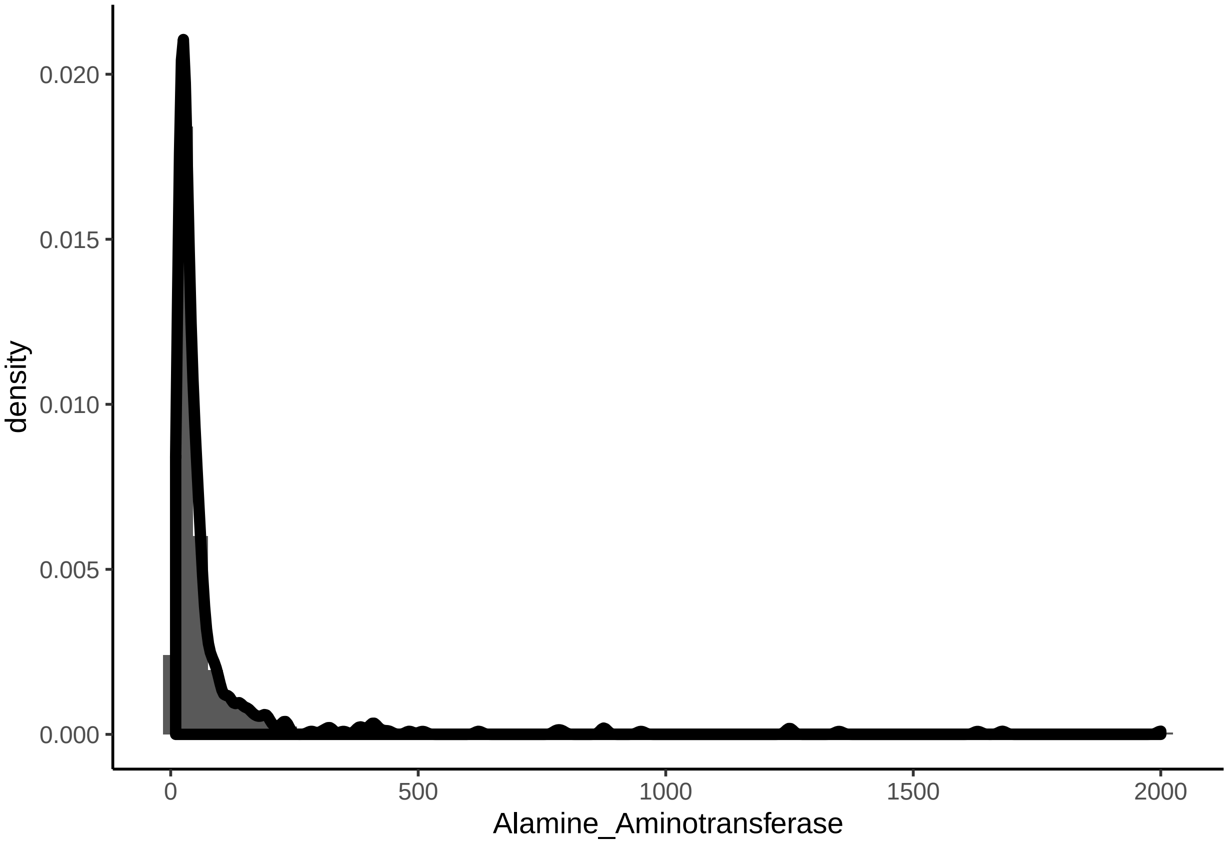
```r
# Direct_Bilirubin distribution
liver_data %>%
    ggplot(aes(x = Direct_Bilirubin)) +
    geom_histogram(aes(y = ..density..), binwidth = 0.4) +
    geom_density(lwd = 2) +
    theme_classic()
```



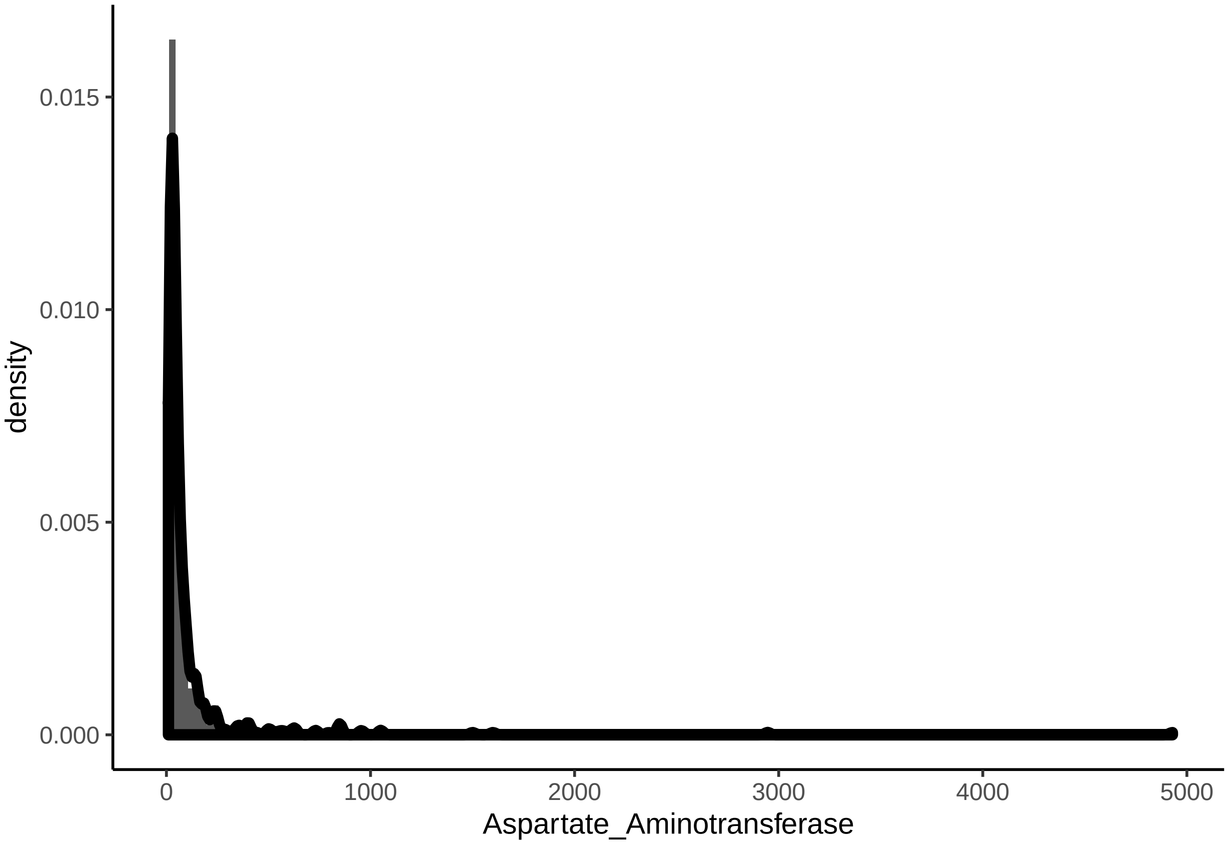
```r
# Alkaline_Phosphotase distribution
liver_data %>%
    ggplot(aes(x = Alkaline_Phosphotase)) +
    geom_histogram(aes(y = ..density..), binwidth = 30) +
    geom_density(lwd = 2) +
    theme_classic()
```



```
# Alamine_Aminotransferase distribution
liver_data %>%
    ggplot(aes(x = Alamine_Aminotransferase)) +
    geom_histogram(aes(y = ..density..), binwidth = 30) +
    geom_density(lwd = 2) +
    theme_classic()
```



```
# Aspartate_Aminotransferase distribution
liver_data %>%
    ggplot(aes(x = Aspartate_Aminotransferase)) +
    geom_histogram(aes(y = ..density..), binwidth = 30) +
    geom_density(lwd = 2) +
    theme_classic()
```



As we expected, we have very highly skewed distributions for these variables.

Examine the highly skewed data using 90% quantile.

```
# 90% quantile
# Albumin_and_Globulin_Ratio
liver_data %>%
    filter(Albumin_and_Globulin_Ratio > quantile(Albumin_and_Globulin_Ratio, 0.9, na.rm = T)) %>%
    summary()
```

```
##       Age           Gender   Total_Bilirubin  Direct_Bilirubin
##  Min.   :11.00   Female:13   Min.   : 0.400   Min.   : 0.1000
##  1st Qu.:29.25   Male  :45   1st Qu.: 0.725   1st Qu.: 0.2000
##  Median :38.50               Median : 0.900   Median : 0.2000
##  Mean   :41.91               Mean   : 1.960   Mean   : 0.8793
##  3rd Qu.:54.75               3rd Qu.: 1.350   3rd Qu.: 0.5000
##  Max.   :70.00               Max.   :25.000   Max.   :13.7000
##  Alkaline_Phosphotase Alamine_Aminotransferase Aspartate_Aminotransferase
##  Min.   : 63.0        Min.   : 12.00           Min.   : 14.00
##  1st Qu.:156.8        1st Qu.: 23.25           1st Qu.: 24.00
##  Median :185.5        Median : 36.00           Median : 34.00
##  Mean   :211.1        Mean   : 40.21           Mean   : 55.76
##  3rd Qu.:212.0        3rd Qu.: 52.75           3rd Qu.: 62.00
##  Max.   :768.0        Max.   :196.00           Max.   :401.00
##  Total_Proteins     Albumin      Albumin_and_Globulin_Ratio
##  Min.   :4.100   Min.   :2.100   Min.   :1.340
##  1st Qu.:6.225   1st Qu.:3.700   1st Qu.:1.400
##  Median :6.900   Median :4.000   Median :1.500
##  Mean   :6.726   Mean   :3.974   Mean   :1.579  
##  3rd Qu.:7.300   3rd Qu.:4.300   3rd Qu.:1.645  
##  Max.   :8.700   Max.   :5.500   Max.   :2.800  
##        outcome  
##  disease   :38  
##  no_disease:20  
##                 
##                 
##                 
##                 
```

```
# Total_Bilirubin distribution
liver_data %>%
    filter(Total_Bilirubin > quantile(Total_Bilirubin, 0.9)) %>% summary()
```

```
##       Age           Gender   Total_Bilirubin Direct_Bilirubin
##  Min.   : 7.00   Female: 9   Min.   : 7.90   Min.   : 3.600  
##  1st Qu.:32.00   Male  :50   1st Qu.:11.20   1st Qu.: 5.350  
##  Median :46.00               Median :16.40   Median : 8.400  
##  Mean   :46.51               Mean   :18.37   Mean   : 8.615  
##  3rd Qu.:58.00               3rd Qu.:22.65   3rd Qu.:11.050  
##  Max.   :75.00               Max.   :75.00   Max.   :19.700  
##  Alkaline_Phosphotase Alamine_Aminotransferase Aspartate_Aminotransferase
##  Min.   : 108.0       Min.   :  21.0           Min.   :  25.0            
##  1st Qu.: 242.0       1st Qu.:  40.5           1st Qu.:  64.0            
##  Median : 300.0       Median :  58.0           Median : 130.0            
##  Mean   : 465.6       Mean   : 205.5           Mean   : 361.3            
##  3rd Qu.: 559.0       3rd Qu.: 140.0           3rd Qu.: 311.5            
##  Max.   :1550.0       Max.   :2000.0           Max.   :4929.0            
##  Total_Proteins     Albumin      Albumin_and_Globulin_Ratio
##  Min.   :4.300   Min.   :1.600   Min.   :0.3000            
##  1st Qu.:5.600   1st Qu.:2.100   1st Qu.:0.5000            
##  Median :6.400   Median :2.600   Median :0.7000            
##  Mean   :6.422   Mean   :2.607   Mean   :0.7724            
##  3rd Qu.:7.050   3rd Qu.:3.000   3rd Qu.:0.9000            
##  Max.   :9.200   Max.   :4.100   Max.   :2.8000            
##        outcome  
##  disease   :59  
##  no_disease: 0  
##                 
##                 
##                 
##                 
```

```
# Direct_Bilirubin distribution
liver_data %>%
    filter(Direct_Bilirubin > quantile(Direct_Bilirubin, 0.9)) %>% summary()
```

```
##       Age           Gender   Total_Bilirubin Direct_Bilirubin
##  Min.   : 7.00   Female: 8   Min.   : 1.50   Min.   : 4.100  
##  1st Qu.:32.00   Male  :51   1st Qu.:11.05   1st Qu.: 5.800  
##  Median :45.00               Median :15.90   Median : 8.400  
##  Mean   :45.54               Mean   :16.95   Mean   : 8.768  
##  3rd Qu.:55.50               3rd Qu.:22.55   3rd Qu.:11.050  
##  Max.   :75.00               Max.   :42.80   Max.   :19.700  
##  Alkaline_Phosphotase Alamine_Aminotransferase Aspartate_Aminotransferase
##  Min.   : 108.0       Min.   :  21.0           Min.   :  25.0  
##  1st Qu.: 238.5       1st Qu.:  42.0           1st Qu.:  70.0  
##  Median : 298.0       Median :  60.0           Median : 130.0  
##  Mean   : 439.4       Mean   : 205.5           Mean   : 351.1  
##  3rd Qu.: 510.0       3rd Qu.: 154.0           3rd Qu.: 255.0  
##  Max.   :1550.0       Max.   :2000.0           Max.   :4929.0  
##  Total_Proteins     Albumin      Albumin_and_Globulin_Ratio
##  Min.   :4.300   Min.   :1.600   Min.   :0.3000            
##  1st Qu.:5.600   1st Qu.:2.100   1st Qu.:0.5000            
##  Median :6.400   Median :2.600   Median :0.7000            
##  Mean   :6.441   Mean   :2.659   Mean   :0.7942            
##  3rd Qu.:7.100   3rd Qu.:3.100   3rd Qu.:0.9000            
##  Max.   :9.200   Max.   :4.100   Max.   :2.8000            
##        outcome  
##  disease   :59  
##  no_disease: 0  
##                 
##                 
##                 
##                 
```

```r
# Alkaline_Phosphotase distribution
liver_data %>%
    filter(Alkaline_Phosphotase > quantile(Alkaline_Phosphotase, 0.9)) %>% summary()
```

```
##       Age           Gender   Total_Bilirubin  Direct_Bilirubin
##  Min.   : 7.00   Female:16   Min.   : 0.600   Min.   : 0.100  
##  1st Qu.:39.00   Male   :43   1st Qu.: 1.350   1st Qu.: 0.700  
##  Median :50.00                Median : 2.700   Median : 1.300  
##  Mean   :48.58                Mean   : 6.344   Mean   : 3.136  
##  3rd Qu.:60.00                3rd Qu.: 9.750   3rd Qu.: 4.450  
##  Max.   :75.00                Max.   :27.200   Max.   :13.700  
##  Alkaline_Phosphotase Alamine_Aminotransferase Aspartate_Aminotransferase
##  Min.   : 512.0       Min.   :  16.0           Min.   :  17.0  
##  1st Qu.: 590.0       1st Qu.:  41.5           1st Qu.:  44.0  
##  Median : 699.0       Median :  64.0           Median :  79.0  
##  Mean   : 868.5       Mean   : 115.4           Mean   : 199.7  
##  3rd Qu.: 991.0       3rd Qu.: 113.0           3rd Qu.: 141.0  
##  Max.   :2110.0       Max.   :1250.0           Max.   :4929.0  
##  Total_Proteins     Albumin      Albumin_and_Globulin_Ratio
##  Min.   :3.600   Min.   :1.500   Min.   :0.350             
##  1st Qu.:5.650   1st Qu.:2.150   1st Qu.:0.575             
##  Median :6.300   Median :2.700   Median :0.740             
##  Mean   :6.402   Mean   :2.788   Mean   :0.778             
##  3rd Qu.:7.250   3rd Qu.:3.300   3rd Qu.:0.900             
##  Max.   :8.000   Max.   :4.900   Max.   :2.500             
##        outcome  
##  disease   :55  
##  no_disease: 4  
##                 
##                 
##                 
##                 
```

```r
# Alamine_Aminotransferase distribution
liver_data %>%
    filter(Alamine_Aminotransferase > quantile(Alamine_Aminotransferase, 0.9)) %>% summary()
```

```
##       Age           Gender   Total_Bilirubin  Direct_Bilirubin
##  Min.   : 4.00   Female:12   Min.   : 0.500   Min.   : 0.100  
##  1st Qu.:32.00   Male  :46   1st Qu.: 1.250   1st Qu.: 0.450  
##  Median :38.50               Median : 3.800   Median : 2.000  
##  Mean   :39.47               Mean   : 6.671   Mean   : 3.284  
##  3rd Qu.:49.50               3rd Qu.: 7.550   3rd Qu.: 4.075  
##  Max.   :75.00               Max.   :27.200   Max.   :12.800  
##  Alkaline_Phosphotase Alamine_Aminotransferase Aspartate_Aminotransferase
##  Min.   : 108.0       Min.   : 141.0           Min.   :  17.0            
##  1st Qu.: 216.0       1st Qu.: 178.2           1st Qu.: 159.0            
##  Median : 298.0       Median : 232.5           Median : 390.5            
##  Mean   : 383.6       Mean   : 443.6           Mean   : 573.5            
##  3rd Qu.: 408.2       3rd Qu.: 436.2           3rd Qu.: 731.0            
##  Max.   :1550.0       Max.   :2000.0           Max.   :4929.0            
##  Total_Proteins     Albumin      Albumin_and_Globulin_Ratio
##  Min.   :3.600   Min.   :1.000   Min.   :0.3000            
##  1st Qu.:5.700   1st Qu.:2.700   1st Qu.:0.7000            
##  Median :6.800   Median :3.150   Median :0.9500            
##  Mean   :6.617   Mean   :3.129   Mean   :0.8998            
##  3rd Qu.:7.475   3rd Qu.:3.700   3rd Qu.:1.1000            
##  Max.   :9.200   Max.   :4.500   Max.   :1.5000            
##        outcome  
##  disease   :55  
##  no_disease: 3  
##                 
##                 
##                 
## 
```

```r
# Aspartate_Aminotransferase distribution
liver_data %>%
    filter(Aspartate_Aminotransferase > quantile(Aspartate_Aminotransferase, 0.9)) %>% summary()
```

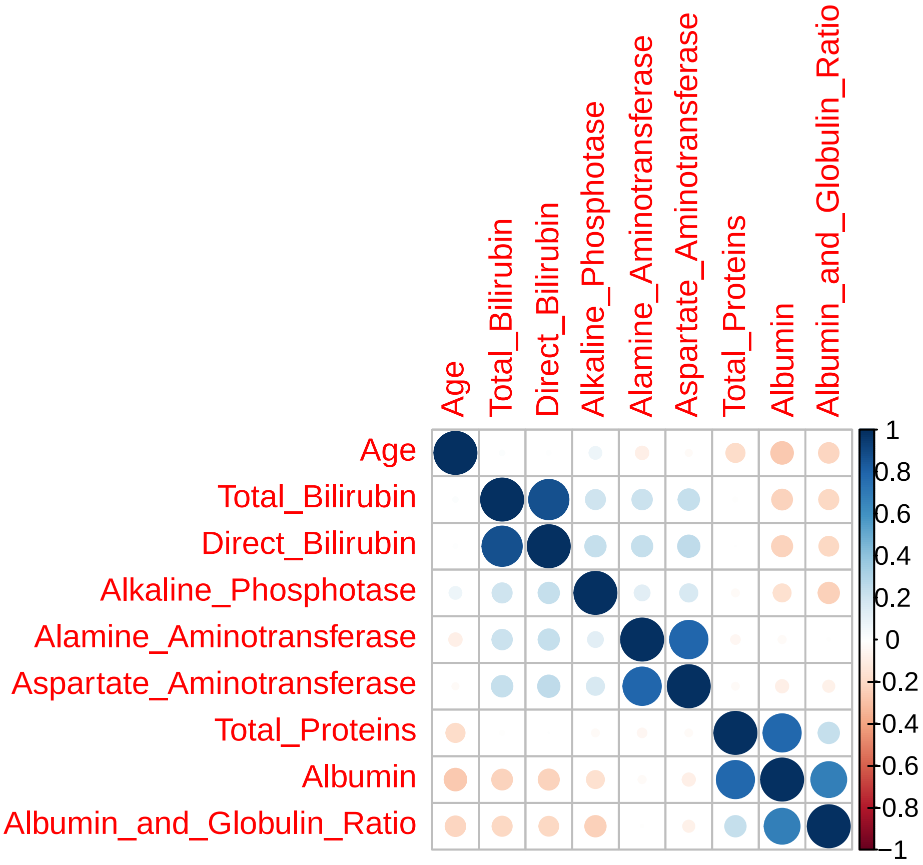
```
##       Age           Gender   Total_Bilirubin  Direct_Bilirubin
##  Min.   : 4.00   Female:10   Min.   : 0.700   Min.   : 0.100  
##  1st Qu.:33.25   Male  :48   1st Qu.: 2.150   1st Qu.: 1.050  
##  Median :40.00               Median : 5.750   Median : 2.600  
##  Mean   :42.38               Mean   : 8.769   Mean   : 4.119  
##  3rd Qu.:52.50               3rd Qu.:15.650   3rd Qu.: 7.150  
##  Max.   :66.00               Max.   :32.600   Max.   :14.100  
##  Alkaline_Phosphotase Alamine_Aminotransferase Aspartate_Aminotransferase
##  Min.   :  92.0       Min.   :  39.0           Min.   : 200.0            
##  1st Qu.: 221.0       1st Qu.: 134.0           1st Qu.: 248.5            
##  Median : 291.5       Median : 225.0           Median : 403.0            
##  Mean   : 358.9       Mean   : 412.5           Mean   : 632.2            
##  3rd Qu.: 354.5       3rd Qu.: 436.2           3rd Qu.: 731.0            
##  Max.   :1550.0       Max.   :2000.0           Max.   :4929.0            
##  Total_Proteins     Albumin      Albumin_and_Globulin_Ratio
##  Min.   :3.600   Min.   :1.000   Min.   :0.3000            
##  1st Qu.:5.700   1st Qu.:2.525   1st Qu.:0.6175            
##  Median :6.800   Median :3.000   Median :0.8000            
##  Mean   :6.552   Mean   :2.986   Mean   :0.8524
##  3rd Qu.:7.275   3rd Qu.:3.475   3rd Qu.:1.1000
##  Max.   :9.200   Max.   :4.500   Max.   :1.5000
##        outcome
##  disease   :56
##  no_disease: 2
##
##
##
##
```

Compare to the original data, it does not seem like we have enough evidences to call them outliers.

3.2 Bivariate analysis

Here, we analyze relationships among variables, especially with our outcome variable.

```
# correlation plot among numerical variables
liver_data %>%
    select_if(is.numeric) %>%
    filter(!is.na(Albumin_and_Globulin_Ratio)) %>%
    cor() %>%
    corrplot()
```



Here, we do not see very unexpected results. Two bilirubin variables are highly correlated. Two aminotransferase variables are highly correlated. Albumin is highly correlated with total proteins. This is easily understandable because albumin is a kind of protein. Albumin and globulin ratio is highly correlated with albumin as expected. One thing unexpected was the negative correlation between age and albumin.

Now, we search for statistical difference in independent variables by the outcome variable.

```
# calculate p-values
comp <- compareGroups(outcome ~ ., data = liver_data)
tab <- createTable(comp, extra.labels = c("", ""))
tab
```

```
## 
## --------Summary descriptives table by 'outcome'---------
## 
## -----------------------------------------------------------------------
##                                         disease   no_disease  p.overall
##                                          N=416       N=167             
## -----------------------------------------------------------------------
## Age, Mean (SD)                        46.2 (15.7) 41.2 (17.0)   0.001  
## Gender:                                                         0.060  
##     Female                            92 (22.1%)  50 (29.9%)           
##     Male                              324 (77.9%) 117 (70.1%)          
## Total_Bilirubin, Mean (SD)            4.16 (7.14) 1.14 (1.00)  <0.001  
## Direct_Bilirubin, Mean (SD)           1.92 (3.21) 0.40 (0.52)  <0.001  
## Alkaline_Phosphotase, Mean (SD)        319 (268)   220 (141)   <0.001  
## Alamine_Aminotransferase, Mean (SD)   99.6 (213)  33.7 (25.1)  <0.001  
## Aspartate_Aminotransferase, Mean (SD)  138 (337)  40.7 (36.4)  <0.001  
## Total_Proteins, Mean (SD)             6.46 (1.09) 6.54 (1.06)   0.393  
## Albumin, Mean (SD)                    3.06 (0.79) 3.34 (0.78)  <0.001  
## Albumin_and_Globulin_Ratio, Mean (SD) 0.91 (0.33) 1.03 (0.29)  <0.001  
## -----------------------------------------------------------------------
```

Student's t-test and Chi-square test were used. We found that many independent variables are statistically significant risk factors to the disease.

4 Modeling

Here, we perform several different algorithms. We first try familiar penalized logistic linear regression. Then, we try more sophisticated yet popular algorithms, naive bayes, random forest and extreme gradient boosting tree. We finally use linear regression to combine all these probabilistic classifiers to maximize our performance. We do not use grid search for tuning the hyperparameters for the sake of simplicity. Caret handles random search generally very well. For discrimination, we use the most popular metric, area under the ROC curve (AUC).

4.1 Pre-processing

We remove 4 missing values then separate the data into a train and a test sets with 80:20 ratio due to the small number observations of dataset.

```
# remove 4 missing values
liver_data_naomit <- liver_data %>% na.omit()

# data split
set.seed(1)
ind <- createDataPartition(liver_data_naomit$outcome, times = 1, p = 0.8, list = F)
```

```
train_set <- liver_data_naomit[ind, ]
test_set <- liver_data_naomit[-ind, ]
```

```
# pre-process first step
# 10-folds cross-validation to tune the models
x <- trainControl(
    method = "cv",
    number = 10,
    classProbs = T,
    summaryFunction = twoClassSummary
)
```

4.2 Penalized Logistic Linear Regression

```
set.seed(1)
# train the model
logistic <- train(
    outcome ~ ., data = train_set, method = "plr",
    trControl = x,
    # standardize the data
    preProc = c("center", "scale"),
    # we use ROC AUC as a metric
    metric = "ROC"
)

logistic
```

```
## Penalized Logistic Regression
##
## 464 samples
##  10 predictor
##   2 classes: 'disease', 'no_disease'
##
## Pre-processing: centered (10), scaled (10)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 418, 418, 418, 417, 418, 418, ...
## Resampling results across tuning parameters:
##
##   lambda  ROC        Sens       Spec
##   0e+00   0.7447729  0.9005348  0.3032967
##   1e-04   0.7447729  0.9005348  0.3032967
##   1e-01   0.7455226  0.9065954  0.2890110
##
## Tuning parameter 'cp' was held constant at a value of bic
## ROC was used to select the optimal model using the largest value.
## The final values used for the model were lambda = 0.1 and cp = bic.
```

4.3 Naive Bayes

```
set.seed(1)
# train the model
```

```r
nb <- train(
    outcome ~ ., data = train_set, method = "nb",
    trControl = x,
    # standardize the data
    preProc = c("center", "scale"),
    # we use ROC AUC as a metric
    metric = "ROC"
)
nb
```

```
## Naive Bayes
##
## 464 samples
##  10 predictor
##   2 classes: 'disease', 'no_disease'
##
## Pre-processing: centered (10), scaled (10)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 418, 418, 418, 417, 418, 418, ...
## Resampling results across tuning parameters:
##
##   usekernel  ROC        Sens       Spec
##   FALSE      0.7145521  0.3733512  0.9472527
##    TRUE      0.7036640  0.5688057  0.7730769
##
## Tuning parameter 'fL' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## ROC was used to select the optimal model using the largest value.
## The final values used for the model were fL = 0, usekernel = FALSE
##  and adjust = 1.
```

We can repeat this training for random forest and extreme gradient boosting tree. However, caretEnsemble package has an efficient caretList function to keep all the models into one list. Additionally, with the list, we can create an ensemble model.

4.4 Ensemble

Here, we use caretList function to perform multiple models at one time with 10-folds cross-validation with random hyperparameter search.

```r
# train multiple models at once
set.seed(1)
models_list <-
    caretList(
        outcome ~ ., data = train_set,
        # standardize the data
        preProc = c("center", "scale"),
        trControl = x,
        # we use ROC AUC as a metric
        metric = "ROC",
        # list the base classifiers
        methodList = c("plr", "nb", "rf", "xgbTree")
    )
```

```r
set.seed(1)
model_ensemble <-
    caretEnsemble(models_list,
                  trControl = x,
                  metric = "ROC")
```

5 Results

Now, it is time to check our performances on the test set. Note that generating an ensemble model does not always better result than the base models.

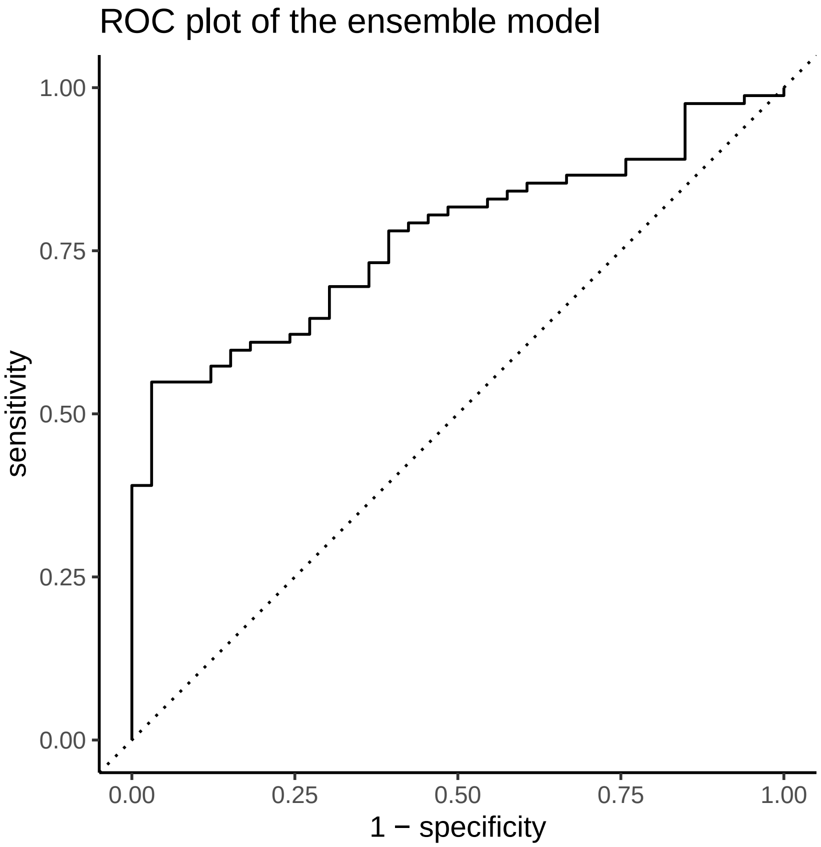
```r
# predict probabilities
pred_log <- predict(model_ensemble$models$plr, test_set, type = "prob")
pred_nb <- predict(model_ensemble$models$nb, test_set, type = "prob")
pred_rf <- predict(model_ensemble$models$rf, test_set, type = "prob")
pred_xgbTree <- predict(model_ensemble$models$xgbTree, test_set, type = "prob")
# final model prediction
pred_ensemble <- predict(model_ensemble, test_set, type = "prob")

# make temporary dataframes
logis_df_tmp <- tibble(truth = test_set$outcome) %>%
    bind_cols(pred_log)
nb_df_tmp <- tibble(truth = test_set$outcome) %>%
    bind_cols(pred_nb)
rf_df_tmp <- tibble(truth = test_set$outcome) %>%
    bind_cols(pred_rf)
xgbTree_df_tmp <- tibble(truth = test_set$outcome) %>%
    bind_cols(pred_xgbTree)
ensemble_df_tmp <- tibble(truth = test_set$outcome) %>%
    bind_cols(disease = 1 - pred_ensemble,
              no_disease = pred_ensemble)

# performance for each generated model
roc_logis <- roc_auc(data = logis_df_tmp, truth = truth, disease)$.estimate
roc_nb <- roc_auc(data = nb_df_tmp, truth = truth, disease)$.estimate
roc_rf <- roc_auc(data = rf_df_tmp, truth = truth, disease)$.estimate
roc_xgbTree <- roc_auc(data = xgbTree_df_tmp, truth = truth, disease)$.estimate
roc_ensemble <- roc_auc(data = ensemble_df_tmp, truth = truth, disease)$.estimate

# save the final performance into a tibble data frame
roc_results <-
    tibble(
        method = c("Penalized Logistic Regression", "Naive Bayes", "Random Forest",
                   "Extreme Gradient Boosting Tree", "Ensemble"),
        ROCAUC = c(roc_logis, roc_nb, roc_rf, roc_xgbTree, roc_ensemble)
    )

# plot a roc curve of the ensemble model
roc_curve(data = ensemble_df_tmp, truth = truth, disease) %>%
    ggplot(aes(x = 1 - specificity, y = sensitivity)) +
    geom_path() +
    geom_abline(lty = 3) +
    coord_equal() +
    ggtitle("ROC plot of the ensemble model") +
    theme_classic()
```



```
roc_results %>% kable()
```

method	ROCAUC
Penalized Logistic Regression	0.7424242
Naive Bayes	0.7250554
Random Forest	0.7793792
Extreme Gradient Boosting Tree	0.7198817
Ensemble	0.7690318

The final AUCs for the models are 0.74, 0.73, 0.78, 0.72, and 0.77 for logistic regression, naive bayes, random forest, extreme gradient boosting tree, and ensemble model, respectively. We did not succeed to make an improvement by aggregating the base four models.

6 Conclusion

In this project, we aimed to predict the liver disease status in Indian population using 10 predictors. We used the caret R package to perform multiple machine learning classifiers. We tuned penalized logistic regression, naive bayes, random forest, extreme gradient boosting tree with 10-folds cross-validation. Finally, we aggregated the four trained base models using linear regression to creat an ensemble model. Among them, the random forest classifier demonstrated the best performance. In our study, we have several drawbacks. First, we did not grid hyperparameter searches for our base classifiers. Spending more time on hyperparameters would significantly improve our models. Second, since the distribution of the outcome variable is unbalanced, we naturally have stronger power to predict the major class, in our case, having the disease. We might need to consider methods to overcome such situation such as up-sampling or down-sampling. Lastly, we used a curated dataset which is considerably rare in the real world problems. Further hyperparameter tuning would be recommended on this project and usage of other metrics might be useful as well such as Precision-Recall AUC.

7 Acknowledgements

This dataset was downloaded from the UCI ML Repository:

Lichman, M. (2013). UCI Machine Learning Repository https://archive.ics.uci.edu/ml/index.php. Irvine, CA: University of California, School of Information and Computer Science.