Quality_R

WCui

9/26/2020

Object: Build a predictive model to forecast the wine quality.

1. Data preprocessing

First, we will have a peek at the dataset.

```
redwine = read.csv("winequality_red.csv")
redwine$good.wine = ifelse(redwine$quality > 6, 1, 0)
str(redwine)
```

```
## 'data.frame':    1599 obs. of  13 variables:
##  $ fixed.acidity       : num  7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
##  $ volatile.acidity    : num  0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...
##  $ citric.acid         : num  0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
##  $ residual.sugar      : num  1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...
##  $ chlorides           : num  0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.071 ...
##  $ free.sulfur.dioxide : num  11 25 15 17 11 13 15 15 9 17 ...
##  $ total.sulfur.dioxide: num  34 67 54 60 34 40 59 21 18 102 ...
##  $ density             : num  0.998 0.997 0.997 0.998 0.998 ...
##  $ pH                  : num  3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...
##  $ sulphates           : num  0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...
##  $ alcohol             : num  9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
##  $ quality             : int  5 5 5 6 5 5 5 7 7 5 ...
##  $ good.wine           : num  0 0 0 0 0 0 0 1 1 0 ...
```

```
summary(redwine)
```

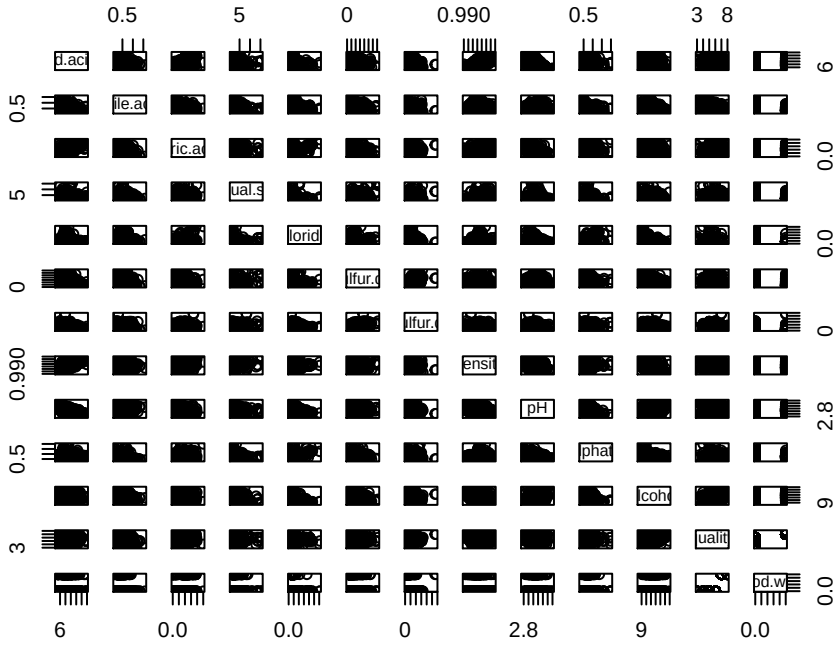
```
##  fixed.acidity   volatile.acidity  citric.acid    residual.sugar
##  Min.   : 4.60   Min.   :0.1200   Min.   :0.000   Min.   : 0.900
##  1st Qu.: 7.10   1st Qu.:0.3900   1st Qu.:0.090   1st Qu.: 1.900
##  Median : 7.90   Median :0.5200   Median :0.260   Median : 2.200
##  Mean   : 8.32   Mean   :0.5278   Mean   :0.271   Mean   : 2.539
##  3rd Qu.: 9.20   3rd Qu.:0.6400   3rd Qu.:0.420   3rd Qu.: 2.600
##  Max.   :15.90   Max.   :1.5800   Max.   :1.000   Max.   :15.500
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide    density
##  Min.   :0.01200   Min.   : 1.00       Min.   :  6.00       Min.   :0.9901
##  1st Qu.:0.07000   1st Qu.: 7.00       1st Qu.: 22.00       1st Qu.:0.9956
##  Median :0.07900   Median :14.00       Median : 38.00       Median :0.9968
##  Mean   :0.08747   Mean   :15.87       Mean   : 46.47       Mean   :0.9967
##  3rd Qu.:0.09000   3rd Qu.:21.00       3rd Qu.: 62.00       3rd Qu.:0.9978
```

```
##  Max.   :0.61100   Max.   :72.00       Max.   :289.00       Max.   :1.0037
##        pH          sulphates         alcohol         quality
##  Min.   :2.740   Min.   :0.3300   Min.   : 8.40   Min.   :3.000
##  1st Qu.:3.210   1st Qu.:0.5500   1st Qu.: 9.50   1st Qu.:5.000
##  Median :3.310   Median :0.6200   Median :10.20   Median :6.000
##  Mean   :3.311   Mean   :0.6581   Mean   :10.42   Mean   :5.636
##  3rd Qu.:3.400   3rd Qu.:0.7300   3rd Qu.:11.10   3rd Qu.:6.000
##  Max.   :4.010   Max.   :2.0000   Max.   :14.90   Max.   :8.000
##    good.wine
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.1357
##  3rd Qu.:0.0000
##  Max.   :1.0000
```

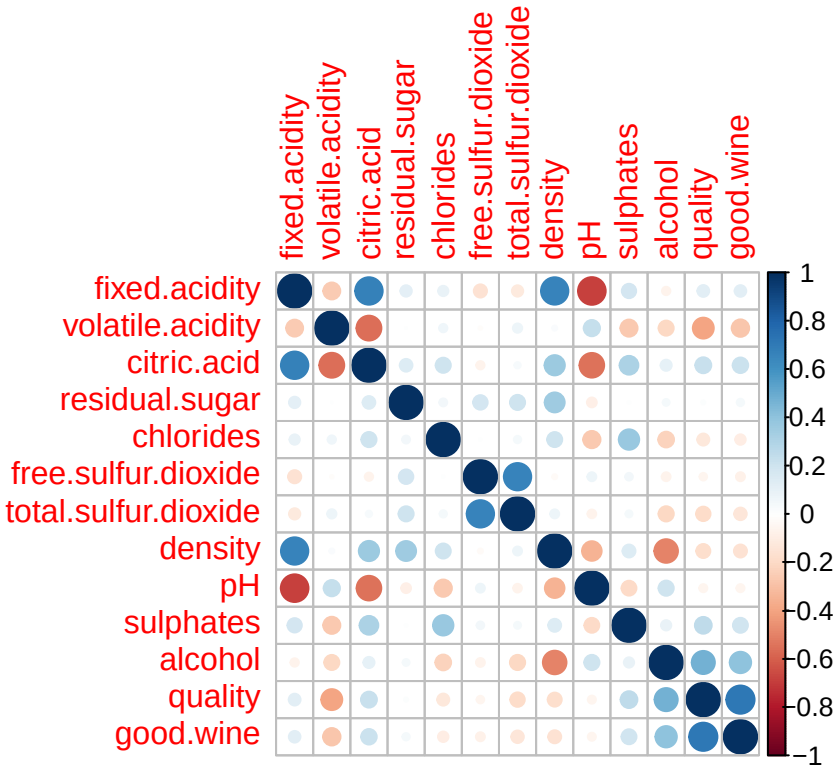
2. Data Visulization (ggplot)

Next, we will explore the correlation among the variables.

```
plot(redwine)
```

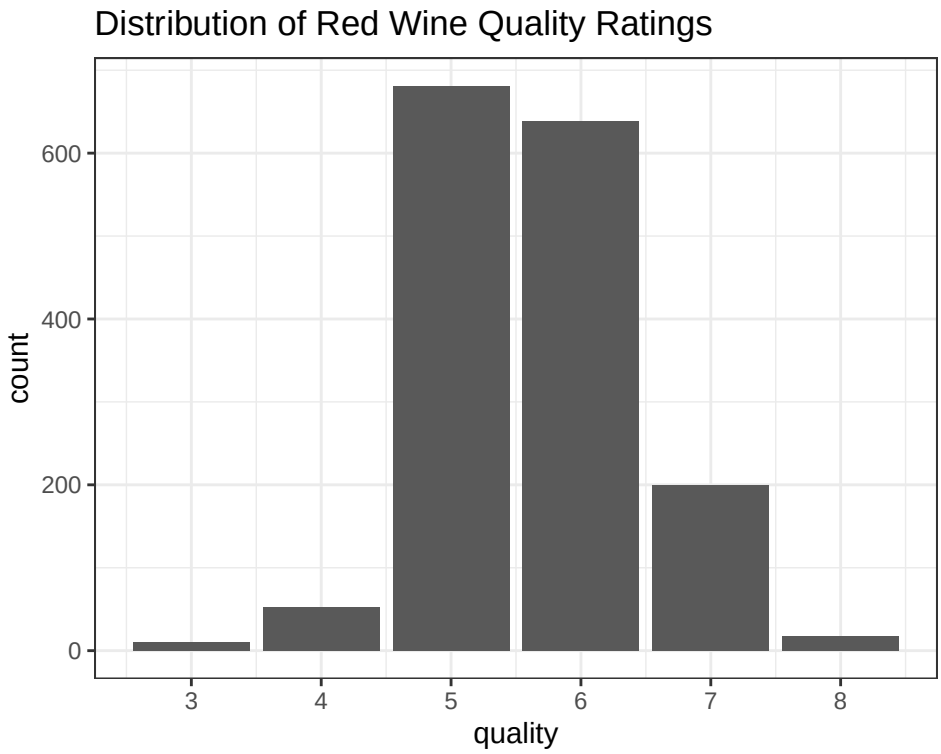



```
corrplot(cor(redwine))
```

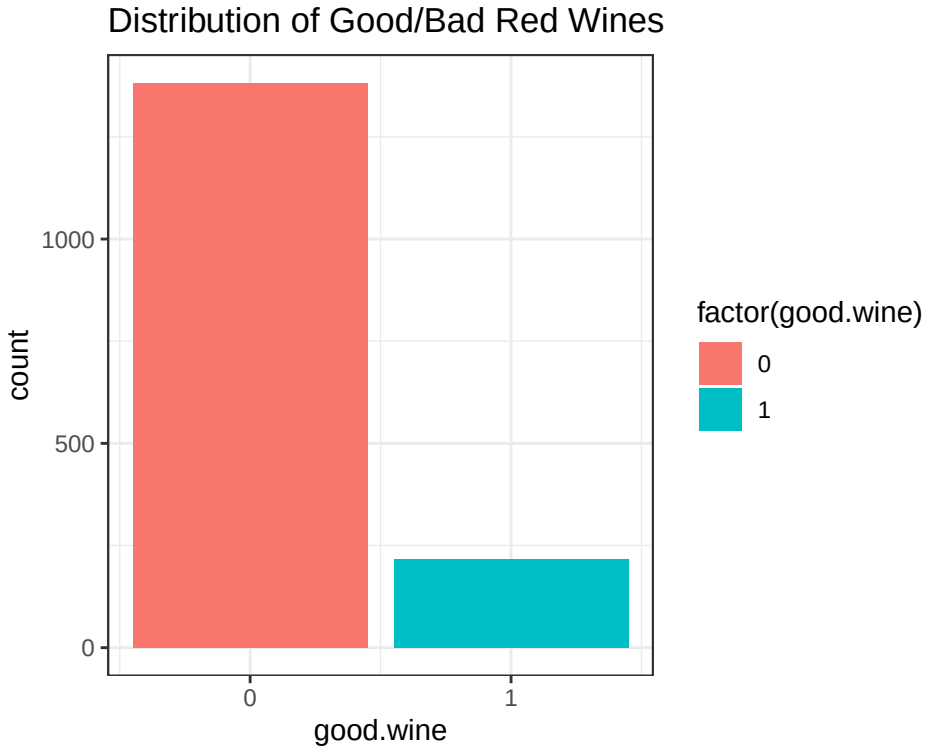


Let's see how the wine quality distributed.

```
ggplot(redwine, aes(x = quality)) + geom_bar(stat = "count",
    position = "dodge") + scale_x_continuous(breaks = seq(3,
    8, 1)) + ggtitle("Distribution of Red Wine Quality Ratings") +
    theme_bw()
```

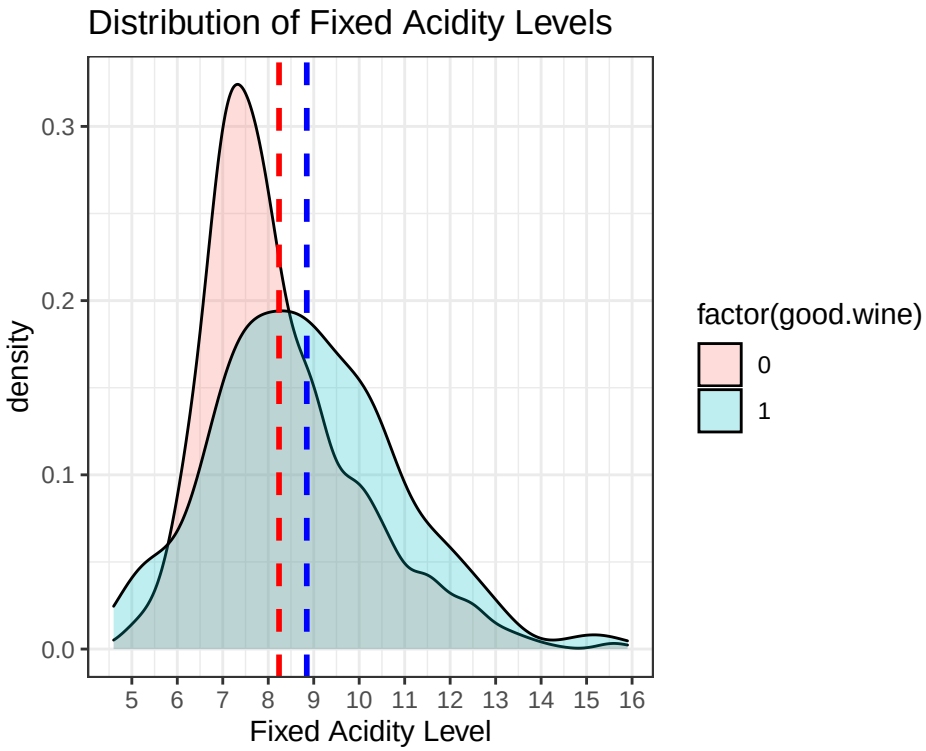


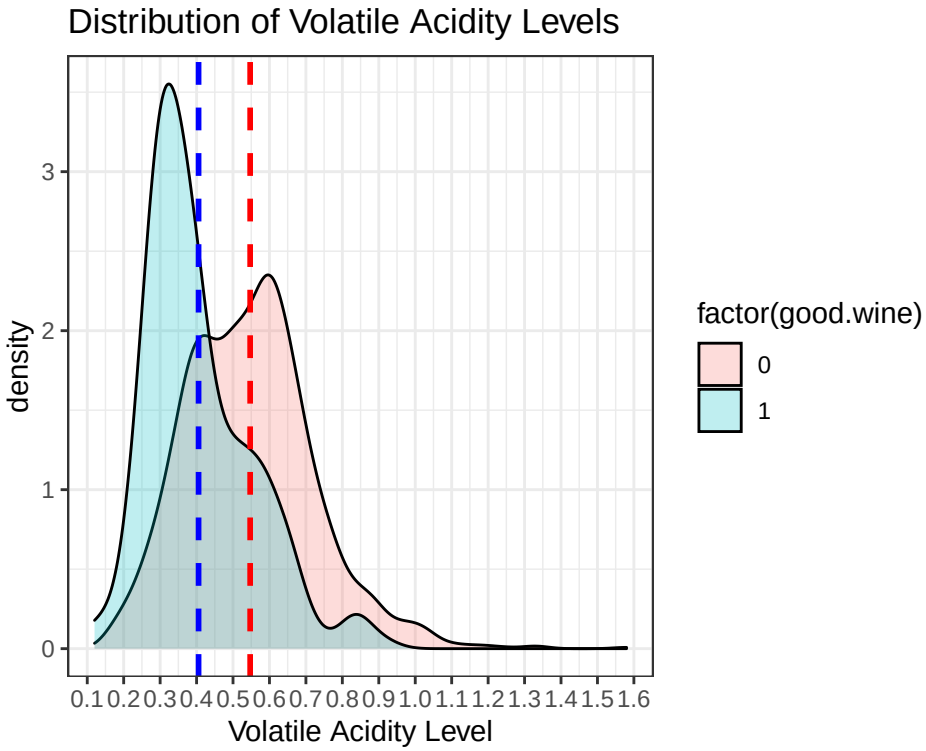

```
ggplot(redwine, aes(x = good.wine, fill = factor(good.wine))) +
    geom_bar(stat = "count", position = "dodge") + scale_x_continuous(breaks = seq(0,
    1, 1)) + ggtitle("Distribution of Good/Bad Red Wines") +
    theme_bw()
```

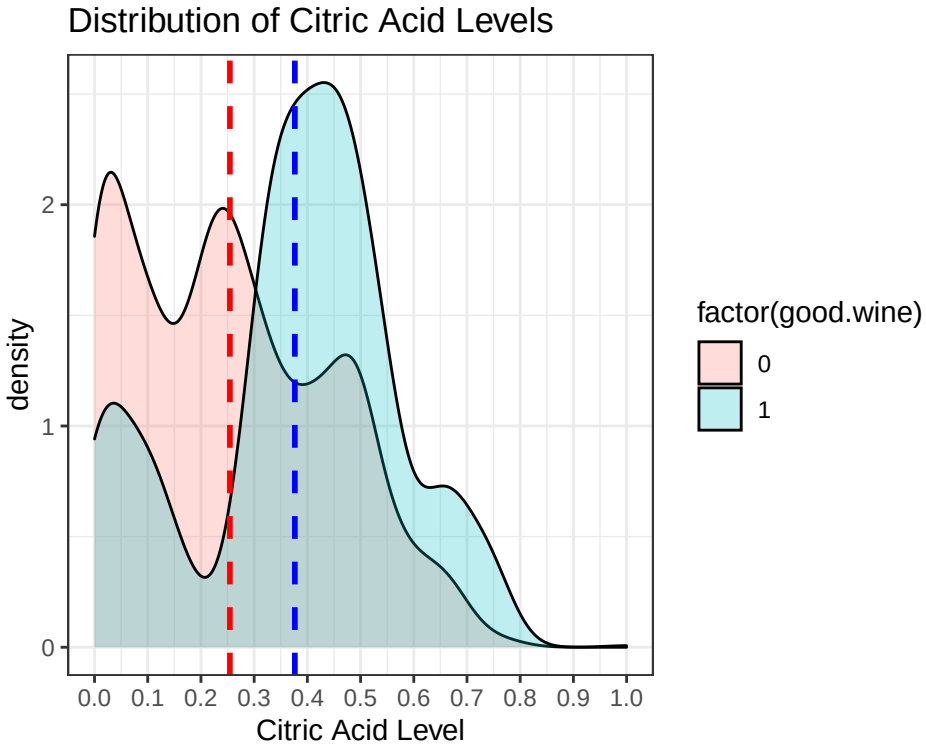


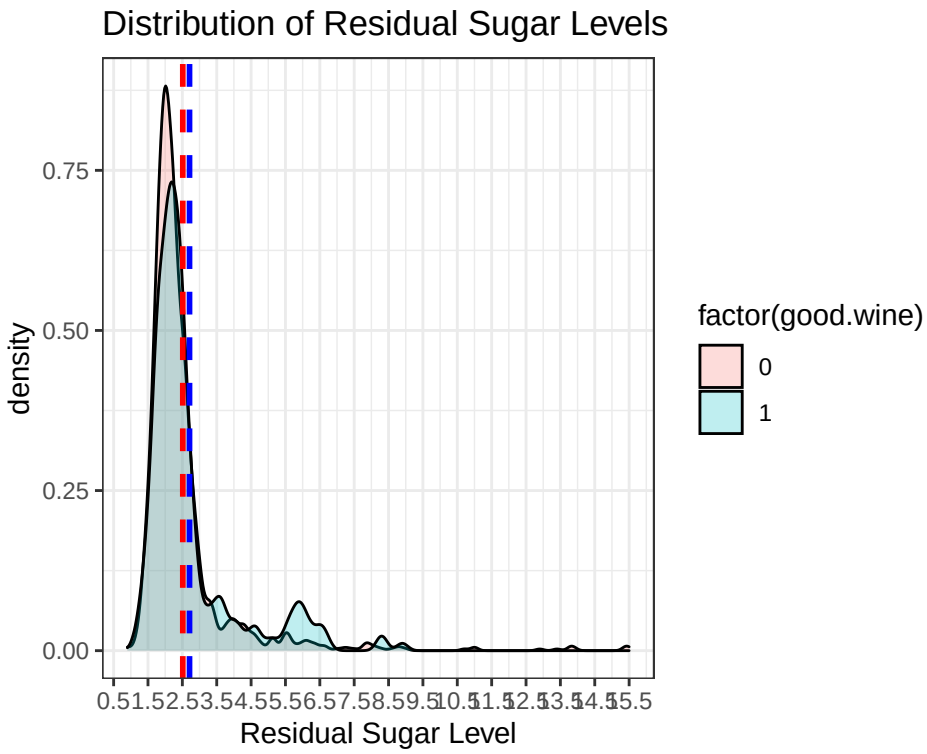
Effect of physiochemical properties on the wine quality

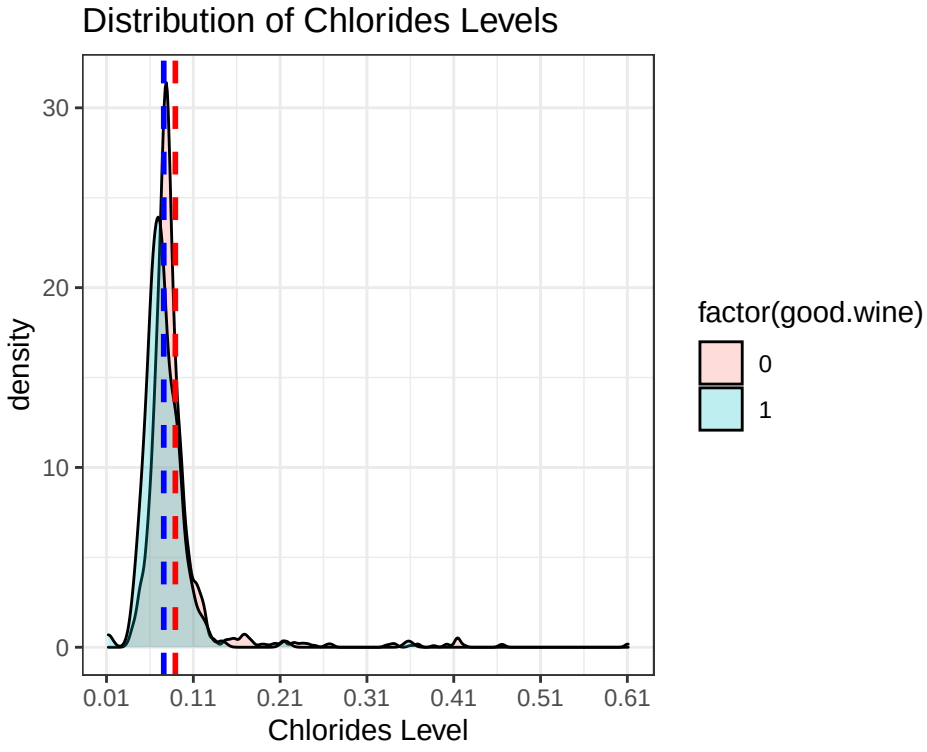
```
ggplot(redwine, aes(x = fixed.acidity, fill = factor(good.wine))) +
    geom_density(alpha = 0.25) + geom_vline(aes(xintercept = mean(fixed.acidity[good.wine ==
    0], na.rm = T)), color = "red", linetype = "dashed", lwd = 1) +
    geom_vline(aes(xintercept = mean(fixed.acidity[good.wine ==
        1], na.rm = T)), color = "blue", linetype = "dashed",
        lwd = 1) + scale_x_continuous(breaks = seq(4, 16, 1)) +
    xlab(label = "Fixed Acidity Level") + ggtitle("Distribution of Fixed Acidity Levels") +
    theme_bw()
```

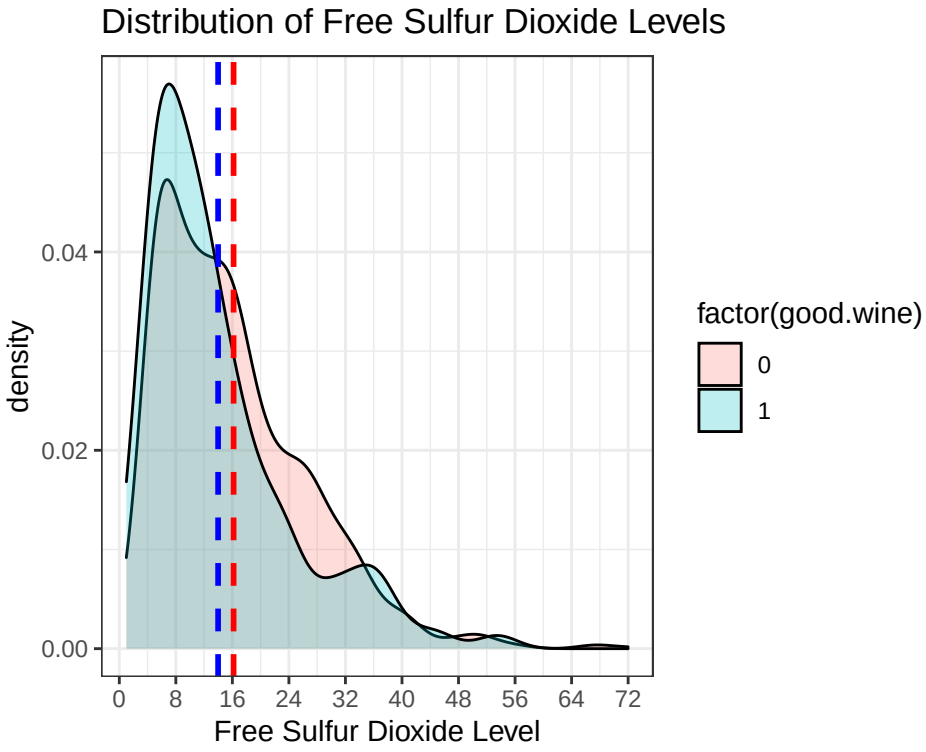


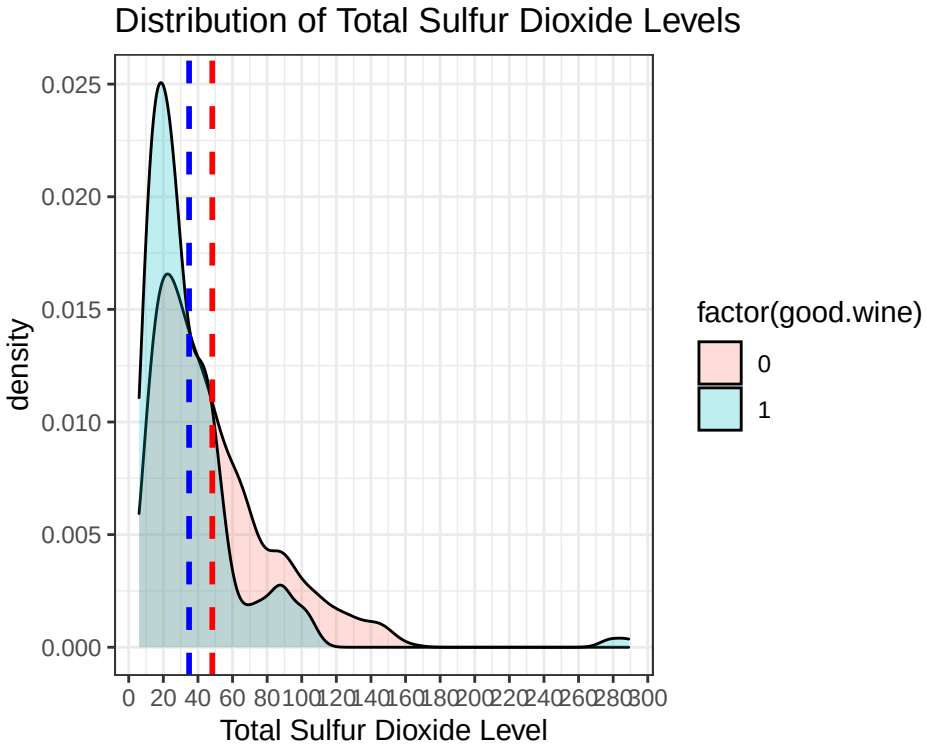

Distribution of Volatile Acidity Levels
density
0
1
2
3
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4 1.5 1.6
Volatile Acidity Level
factor(good.wine)
0
1

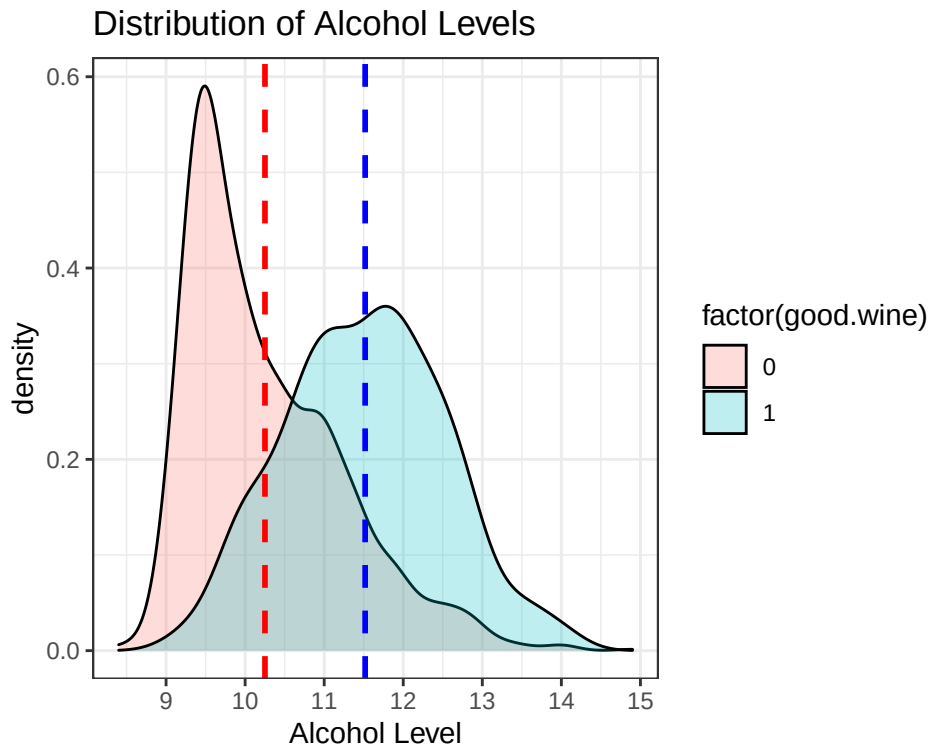

Distribution of Citric Acid Levels
density
0
1
2
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
Citric Acid Level
factor(good.wine)
0
1


Distribution of Residual Sugar Levels
density
0.75
0.50
0.25
0.00
0.51.52.53.54.55.56.57.58.59.510.511.512.513.514.515.5
Residual Sugar Level
factor(good.wine)
0
1


Distribution of Chlorides Levels
density
30
20
10
0
0.01
0.11
0.21
0.31
0.41
0.51
0.61
Chlorides Level
factor(good.wine)
0
1


Distribution of Free Sulfur Dioxide Levels
density
0.04
0.02
0.00
0 8 16 24 32 40 48 56 64 72
Free Sulfur Dioxide Level
factor(good.wine)
0
1


Distribution of Total Sulfur Dioxide Levels
density
0.025
0.020
0.015
0.010
0.005
0.000
0 20 40 60 80 100 120 140 160 180 200 220 240 260 280 300
Total Sulfur Dioxide Level
factor(good.wine)
0
1



3. Predictive Modeling by Random Forest

```
redwineRF = randomForest(factor(good.wine) ~ . - quality, redwine,
    ntree = 200)
redwineRF
```

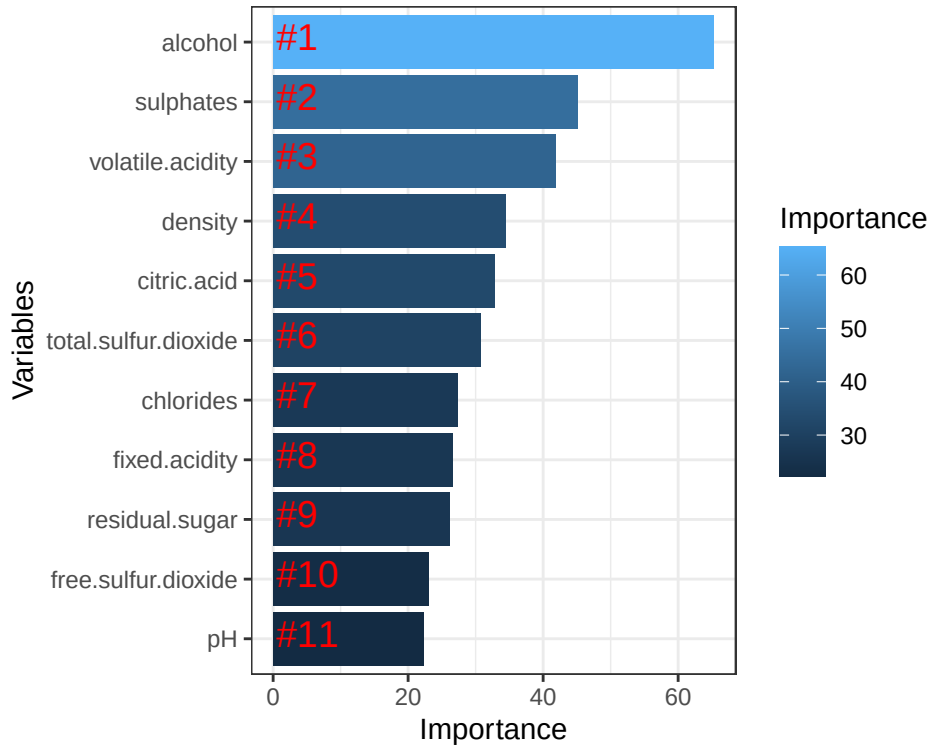
3.1 Model Fit

```
##
## Call:
##  randomForest(formula = factor(good.wine) ~ . - quality, data = redwine,      ntree = 200)
##                Type of random forest: classification
##                      Number of trees: 200
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 8.19%
## Confusion matrix:
##      0   1 class.error
## 0 1345  37  0.02677279
## 1   94 123  0.43317972
```

The classification accuracy reach to ~92%.

3.2 Variable Importance

```
importance = importance(redwineRF)
varImportance = data.frame(Variables = row.names(importance),
    Importance = round(importance[, "MeanDecreaseGini"], 2))
rankImportance = varImportance %>% mutate(Rank = paste0("#",
    dense_rank(desc(Importance))))
ggplot(rankImportance, aes(x = reorder(Variables, Importance),
    y = Importance, fill = Importance)) + geom_bar(stat = "identity") +
    geom_text(aes(x = Variables, y = 0.5, label = Rank), hjust = 0,
        vjust = 0.3, size = 5, colour = "red") + labs(x = "Variables") +
    coord_flip() + theme_bw()
```



The alcohol has the highest importance among the features, which is consistent with the density plot observation.