Final Project

Wine Quality

Nam Jun Lee

12/10/2021

Contents

1 Dataset Information & Goal

The data set I chose is red wine quality. This dataset is from https://archive.ics.uci.edu/ml/index.php, which has **1599** instances and a total of **12** objects.
1. Fixed acidity
2. Volatile acidity
3. Citric acid
4. Residual sugar
5. Chloride
6. Free Sulfur dioxide
7. Total sulfur dioxide
8. Density
9. pH
10. Sulphates
11. Alcohol
12. Quality (0-10 points)

Check the individual objects, I decided to focus on statistically evaluating wine quality by classifying it into two types, and we think **binary classification analysis** is suitable because my goal is to test models that distinguish quality and bad by creating dummy variables rather than session analysis that predicts wine quality out of 0 to 10. To this end, I plan to evaluate which models perform well using **Logistic Regression analysis**, **LDA**, **QDA**, and **KNN**.

2 Data Collection

2.1 Load the dataset

```
# import csv file
rw <- read.csv("winequality-red.csv", stringsAsFactors = F, sep = ";")
# head for red wine data
head(rw)
```

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1           7.4             0.70        0.00            1.9     0.076
## 2           7.8             0.88        0.00            2.6     0.098
## 3           7.8             0.76        0.04            2.3     0.092
## 4          11.2             0.28        0.56            1.9     0.075
## 5           7.4             0.70        0.00            1.9     0.076
## 6           7.4             0.66        0.00            1.8     0.075
##   free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 1                  11                   34  0.9978 3.51      0.56     9.4
## 2                  25                   67  0.9968 3.20      0.68     9.8
## 3                  15                   54  0.9970 3.26      0.65     9.8
## 4                  17                   60  0.9980 3.16      0.58     9.8
## 5                  11                   34  0.9978 3.51      0.56     9.4
## 6                  13                   40  0.9978 3.51      0.56     9.4
##   quality
## 1       5
## 2       5
## 3       5
## 4       6
```

```
## 5       5
## 6       5
```

Load the red wine data needed for data analysis and obtain rows for the first six to see what is there.

2.2 Data information & Summary

```
# summart red wine dataset
summary(rw)
```

```
##  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
```

```
# convert the specified value into a red wine dataset
str(rw)
```

```
## 'data.frame':    1599 obs. of  12 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 ...
```

This dataset has a total of 1599 data, 12 variables, and contains data frame properties. Just quality variable is integer variable.
As a result of checking the summary, total.sulfur.dioxide, residual.sugar, and free.sulfur.dioxide, has see that the maximum is very high between the median. Through this, that there is an outlier.

3 Data Preprocessing & EDA

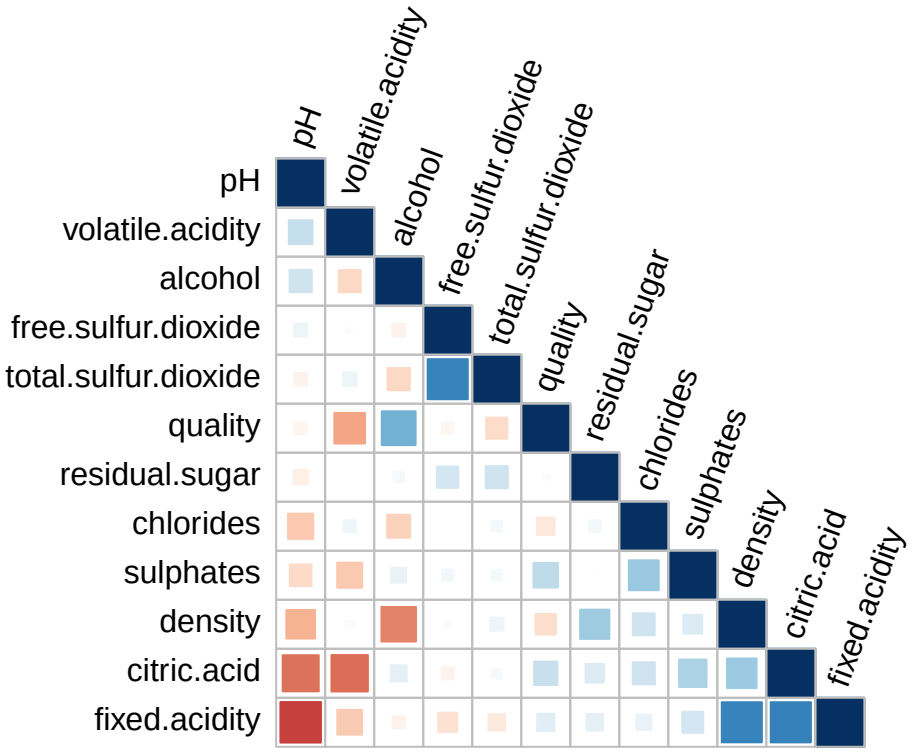
```
# find null
colSums(is.na(rw))
```

```
##        fixed.acidity     volatile.acidity          citric.acid
##                    0                    0                    0
##       residual.sugar            chlorides  free.sulfur.dioxide
##                    0                    0                    0
## total.sulfur.dioxide              density                   pH
##                    0                    0                    0
##            sulphates              alcohol              quality
##                    0                    0                    0
```

After checking the missing values in the data, it does not have any missing value in all variables.

3.1 Correlation of all data

```
# correlation relationship plot
corr <- cor(rw)
corrplot(corr, method = "square", tl.co = "black", tl.srt = 70, cl.pos = "n", order = "FPC",
    type = "lower")
```

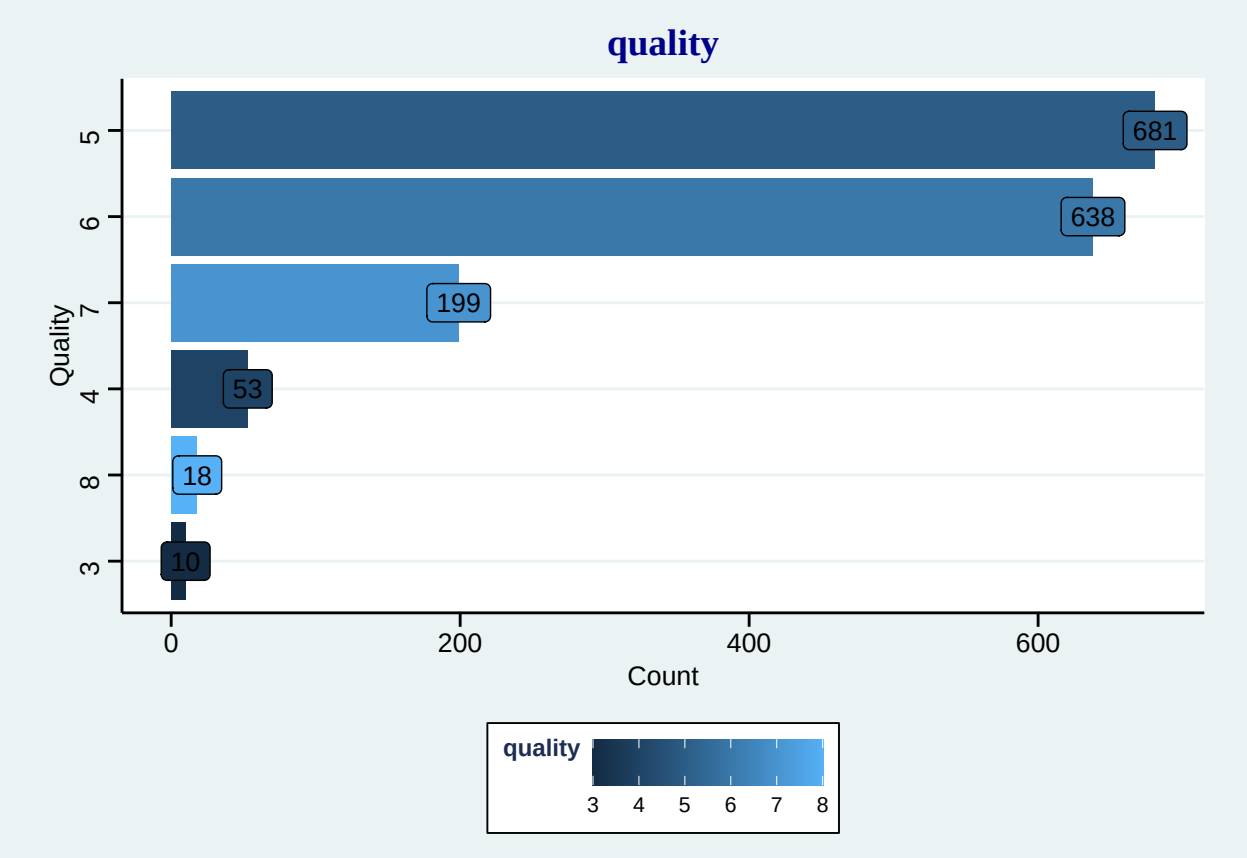


Through the correlation plot, it can be seen that alcohol has a high positive correlation with quality and

volatile acidity has a high negative correlation with quality. In addition, it can be seen that density and citric acid have a moderate positive correlation with quality. Through this, it can be seen that **alcohol is the most related to the quality of wine among all variables**.

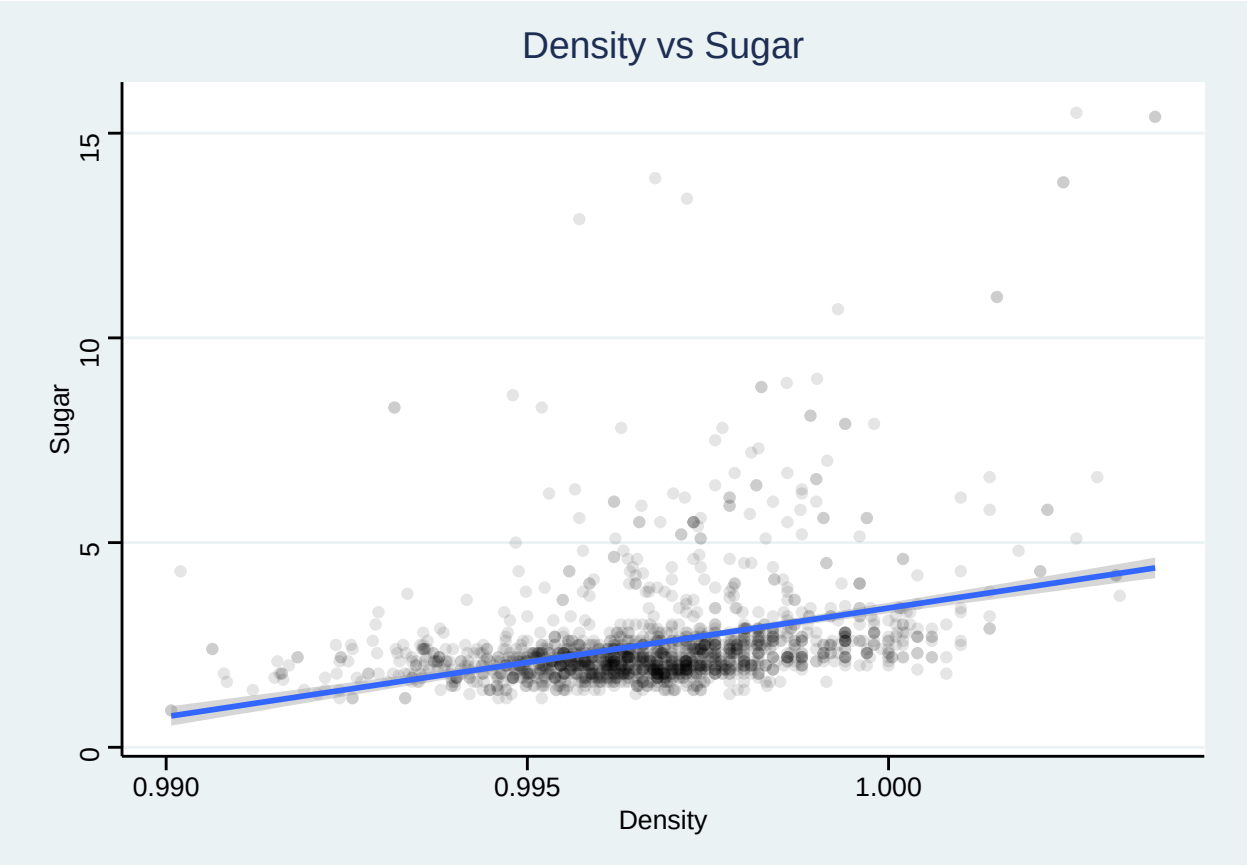
3.2 Visualization

```
# count the each quality
group_quality <- rw %>%
    group_by(quality) %>%
    dplyr::summarise(count = n()) %>%
    arrange(desc(count))
# show the count of quality plot
ggplot(group_quality, aes(x = reorder(quality, count), y = count, fill = quality)) +
    coord_flip() + geom_bar(stat = "identity", position = "dodge") + labs(title = "quality") +
    geom_label(aes(label = count), size = 3.4) + xlab("Quality") + ylab("Count") +
    theme_stata() + theme(plot.title = element_text(family = "serif", color = "darkblue",
    face = "bold")) + theme(legend.title = element_text(size = 9, face = "bold")) +
    theme(legend.text = element_text(size = 7.5))
```



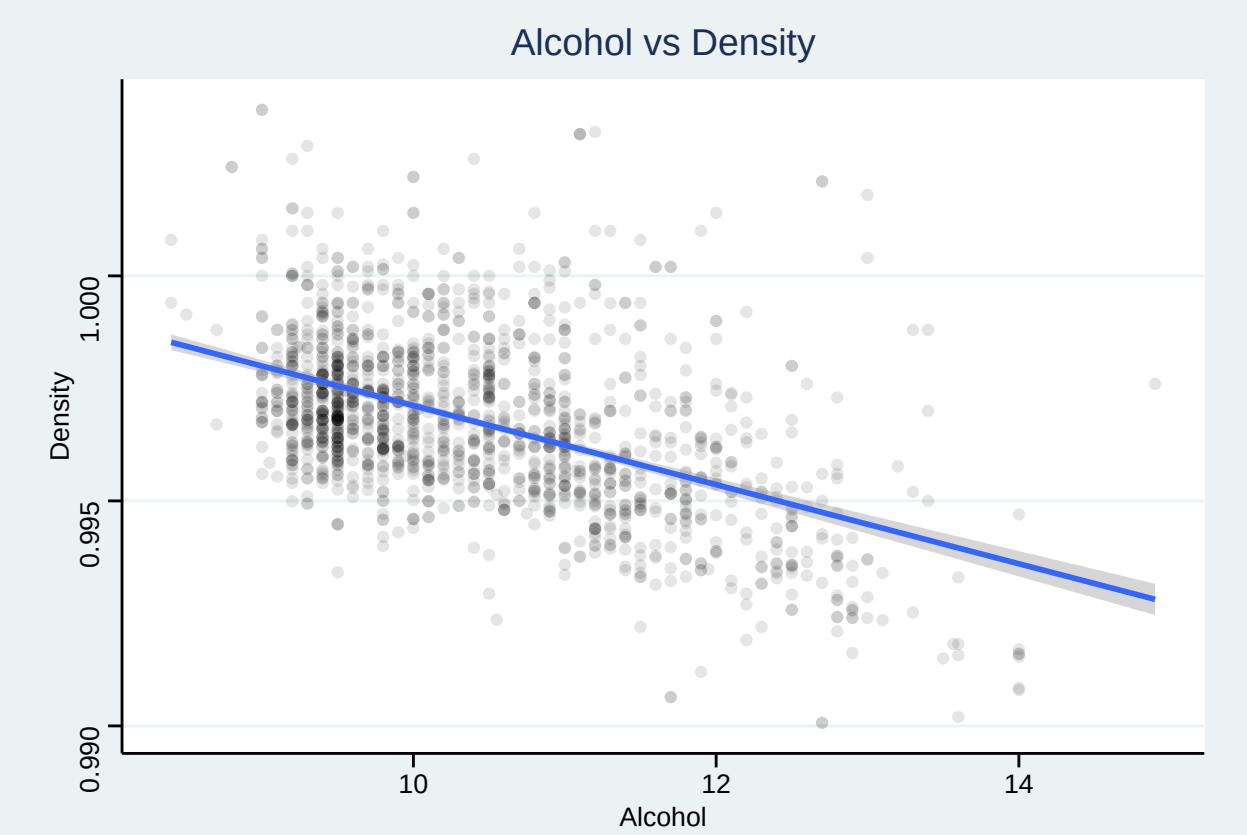
As a result of checking the quality, quality 3 for the smallest proportion, quality 5 for the largest proportion, and next quality 6 for the largest proportion. Based on this, I think it is better to cut bad wine into good wine based on 6.

```
# show the density vs sugar plot
ggplot(rw, aes(x = density, y = residual.sugar)) + xlab("Density") + ylab("Sugar") +
    geom_point(alpha = 0.1) + geom_smooth(method = "lm", formula = y ~ x) + theme_stata() +
    ggtitle("Density vs Sugar")
```



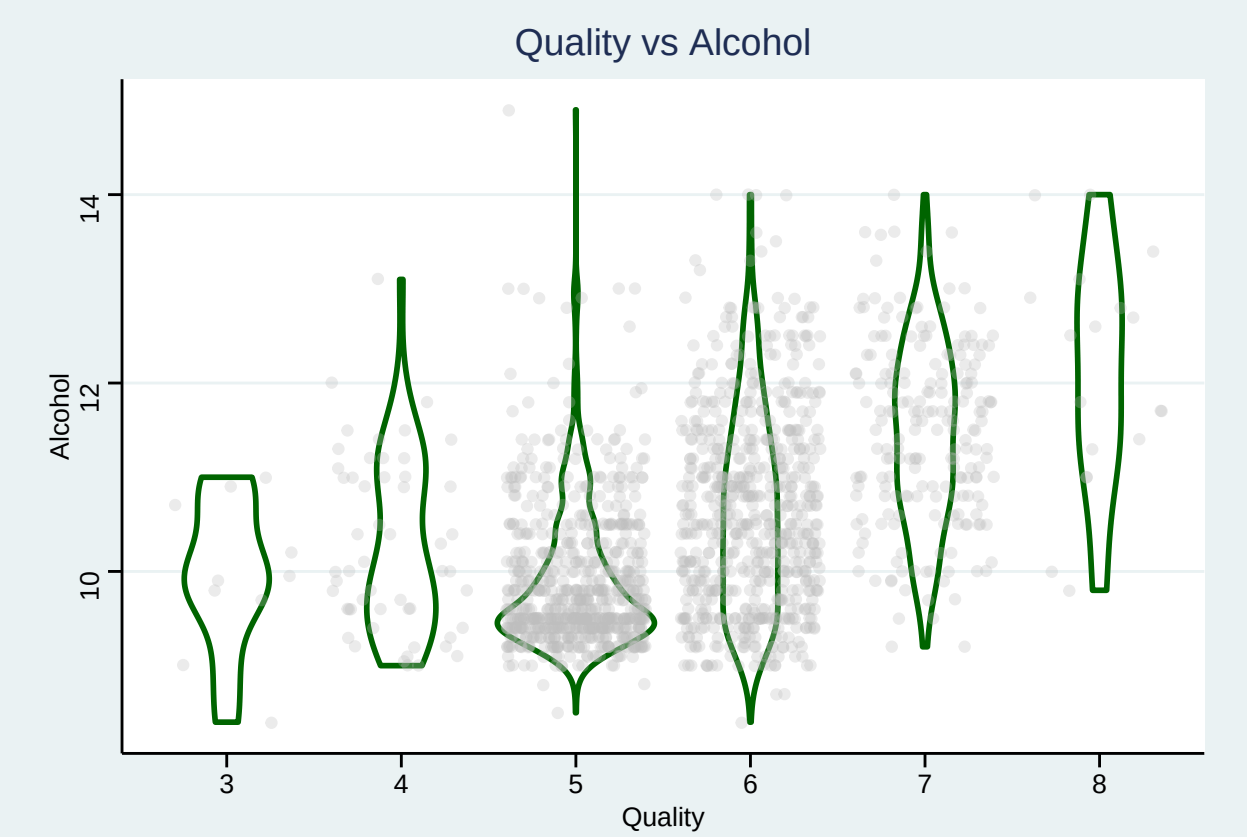
From the scatter plot, it can be seen that there is a weak positive correlation between sugar and density.

```
# show the alcohol vs density plot
ggplot(rw, aes(x = alcohol, y = density)) + geom_point(alpha = 0.1) + theme_stata() +
    geom_smooth(method = "lm", formula = y ~ x) + ggtitle("Alcohol vs Density") +
    xlab("Alcohol") + ylab("Density")
```



From the scatter plot, it can be seen that there is a negative correlation between alcohol and density.

```
# show the alcohol vs quality plot
ggplot(rw, aes(x = factor(quality), y = alcohol)) + geom_violin(color = "darkgreen",
    lwd = 1) + ggtitle("Quality vs Alcohol") + geom_jitter(color = "gray", alpha = 0.3) +
    xlab("Quality") + ylab("Alcohol") + theme_stata()
```



When viewed from the violin plot and the scatter plot above, it can be seen that alcohol and quality are somewhat related, and the two variables are nonlinear.

3.3 Dummy Variable

```
# create dummy variable based on 6 (quality)
rw$rating <- ifelse(as.numeric(rw$quality) > 6, 1, 0)
red.wine = rw
# change to factor data type
red.wine$rating <- as.factor(rw$rating)
# representation of data with variable name and the frequncy
table(red.wine$rating)
```

```
##
##    0    1
## 1382  217
```

```
# show every column in a data frame
glimpse(red.wine)
```

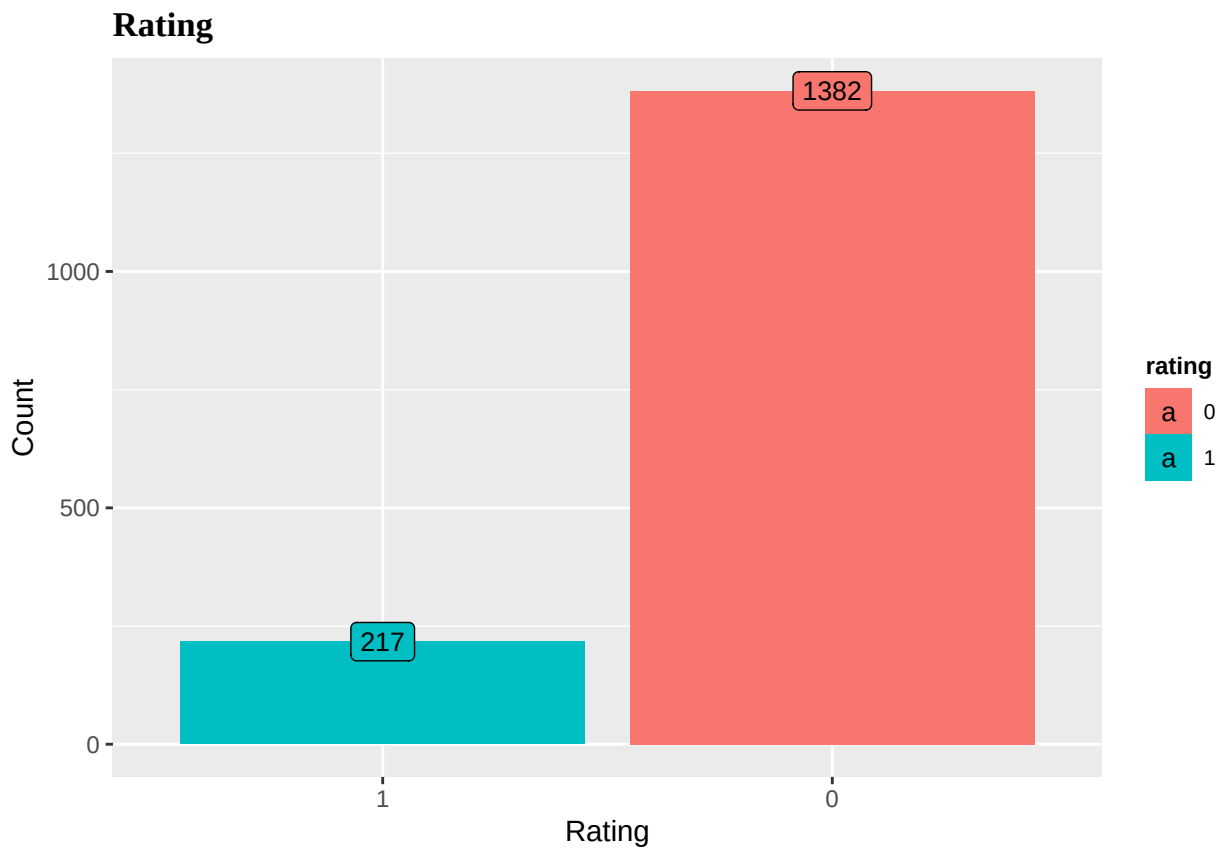
```
## Rows: 1,599
## Columns: 13
## $ fixed.acidity        <dbl> 7.4, 7.8, 7.8, 11.2, 7.4, 7.4, 7.9, 7.3, 7.8, 7.5~
## $ volatile.acidity     <dbl> 0.700, 0.880, 0.760, 0.280, 0.700, 0.660, 0.600, ~
```

```
## $ citric.acid          <dbl> 0.00, 0.00, 0.04, 0.56, 0.00, 0.00, 0.06, 0.00, 0~
## $ residual.sugar       <dbl> 1.9, 2.6, 2.3, 1.9, 1.9, 1.8, 1.6, 1.2, 2.0, 6.1,~
## $ chlorides            <dbl> 0.076, 0.098, 0.092, 0.075, 0.076, 0.075, 0.069, ~
## $ free.sulfur.dioxide  <dbl> 11, 25, 15, 17, 11, 13, 15, 15, 9, 17, 15, 17, 16~
## $ total.sulfur.dioxide <dbl> 34, 67, 54, 60, 34, 40, 59, 21, 18, 102, 65, 102,~
## $ density              <dbl> 0.9978, 0.9968, 0.9970, 0.9980, 0.9978, 0.9978, 0~
## $ pH                   <dbl> 3.51, 3.20, 3.26, 3.16, 3.51, 3.51, 3.30, 3.39, 3~
## $ sulphates            <dbl> 0.56, 0.68, 0.65, 0.58, 0.56, 0.56, 0.46, 0.47, 0~
## $ alcohol              <dbl> 9.4, 9.8, 9.8, 9.8, 9.4, 9.4, 9.4, 10.0, 9.5, 10.~
## $ quality              <int> 5, 5, 5, 6, 5, 5, 5, 7, 7, 5, 5, 5, 5, 5, 5, 5, 7~
## $ rating               <fct> 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1~
```

The rating variable was created, added to the data frame, and converted into factor data type.

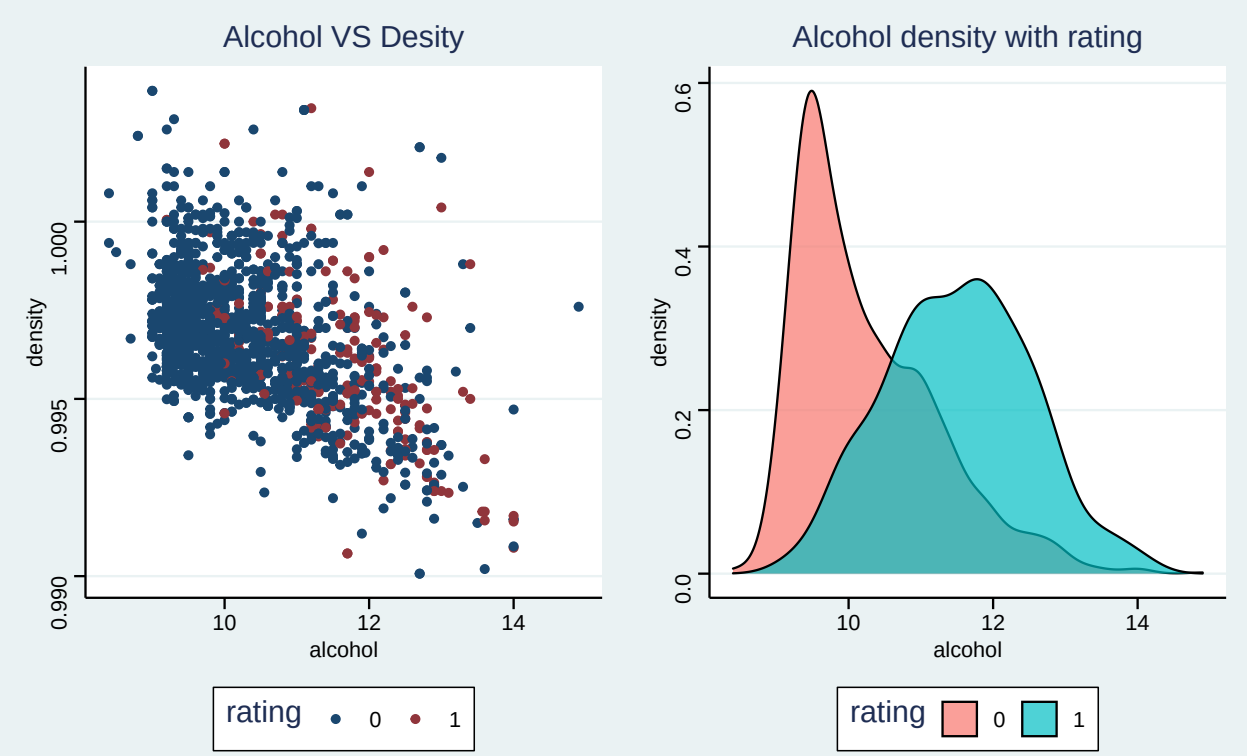
3.4 Visualization after add dummy variable

```
# count each rating (bad, good)
group_rating <- red.wine %>%
    group_by(rating) %>%
    dplyr::summarise(count = n()) %>%
    arrange(desc(count))
# show count each rating plot
ggplot(group_rating, aes(x = reorder(rating, count), y = count, fill = rating)) +
    geom_bar(stat = "identity", position = "dodge") + ggtitle("Rating") + xlab("Rating") +
    ylab("Count") + theme(plot.title = element_text(family = "serif", color = "black",
    face = "bold")) + theme(legend.title = element_text(size = 9, face = "bold")) +
    theme(legend.text = element_text(size = 7.5)) + geom_label(aes(label = count),
    size = 3.4)
```



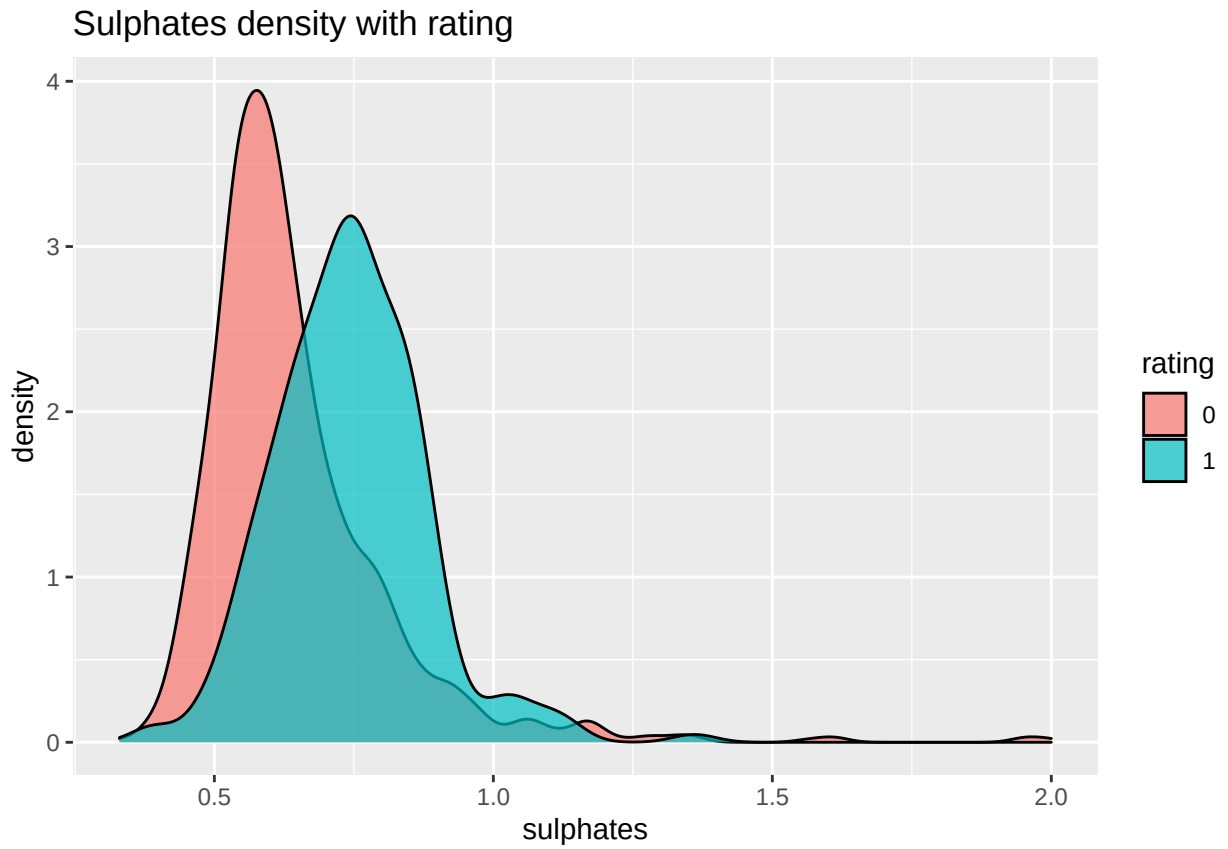
It can be seen that there are more high-quality red wines than poor-quality red wines.

```
# alcohol vs density scatter plot each rating
ad <- ggplot(red.wine, aes(alcohol, density)) + geom_point(aes(color = rating)) +
    theme_stata() + scale_color_stata() + labs(title = "Alcohol VS Desity")
# alcohol vs density plot each rating
ar <- ggplot(red.wine, aes(alcohol, fill = rating)) + geom_density(alpha = 0.7) +
    theme_stata() + scale_color_stata() + labs(title = "Alcohol density with rating")
# Show two graphs on one page
grid.arrange(ad, ar, ncol = 2)
```



The first graph shows that alcohol density has a **negative correlation regardless of quality**.
The second graph shows that the two variables have a **nonlinear relationship** and that the higher the alcohol concentration, the better the quality of wine.

```
# sulphates density plot each rating
ggplot(red.wine, aes(sulphates, fill = rating)) + ggtitle("Sulphates density with rating") +
    geom_density(alpha = 0.7)
```



When checking the density of sulphates, it can be seen that wine with low quality belongs to the lower density of sulfate than high quality red wine.

4 Split Data Set

```
# set test and train use to predictions
train <- 1:(dim(red.wine)[1]/2)
test <- (dim(red.wine)[1]/2 + 1):dim(red.wine)[1]
train_wine <- red.wine[train, ]
test_wine <- red.wine[test, ]
rating_wine <- red.wine$rating[test]
```

Data was divided prior to the creation of a learning model.

5 Create Model & Model Verification

5.1 Logistic Regression

```
set.seed(1)
# fit logistic model with all variables
glm.fit <- glm(rating ~ . - quality, data = red.wine, family = binomial, subset = train)
```

```r
# summary logiistic model
summary(glm.fit)
```

```
##
## Call:
## glm(formula = rating ~ . - quality, family = binomial, data = red.wine,
##     subset = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5052  -0.3618  -0.2218  -0.1274   2.7226
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           200.65698  167.99563   1.194 0.232315
## fixed.acidity           0.22517    0.19507   1.154 0.248382
## volatile.acidity       -0.54475    1.04183  -0.523 0.601060
## citric.acid             1.73114    1.25402   1.380 0.167440
## residual.sugar          0.17764    0.12890   1.378 0.168171
## chlorides              -7.81249    4.01743  -1.945 0.051817 .
## free.sulfur.dioxide     0.05549    0.02425   2.288 0.022148 *
## total.sulfur.dioxide   -0.03687    0.01006  -3.663 0.000249 ***
## density              -218.26142  171.20372  -1.275 0.202357
## pH                      1.12955    1.55284   0.727 0.466976
## sulphates               3.72222    0.75801   4.911 9.08e-07 ***
## alcohol                 0.67026    0.17613   3.806 0.000141 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 524.30  on 798  degrees of freedom
## Residual deviance: 361.68  on 787  degrees of freedom
## AIC: 385.68
##
## Number of Fisher Scoring iterations: 7
```

This summary shows that the **total.sulfur.dioxide**, **sulphates**, and **alcohol** p-value are 0.001, rejecting the null hypothesis, showing that the **free.sulfur.dioxide** p-value is 0.05, and that the remaining variables cannot reject the null hypothesis.

```r
# test the anova
anova(glm.fit, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: rating
##
## Terms added sequentially (first to last)
##
```

```
##
##                      Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                  798     524.30
## fixed.acidity         1  28.8946       797     495.41 7.643e-08 ***
## volatile.acidity      1  15.0811       796     480.32 0.0001030 ***
## citric.acid           1   5.7897       795     474.53 0.0161206 *
## residual.sugar        1   2.8366       794     471.70 0.0921372 .
## chlorides             1   7.7818       793     463.92 0.0052775 **
## free.sulfur.dioxide   1   2.8095       792     461.11 0.0937055 .
## total.sulfur.dioxide  1  24.9256       791     436.18 5.958e-07 ***
## density               1  23.3174       790     412.86 1.373e-06 ***
## pH                    1   7.2552       789     405.61 0.0070695 **
## sulphates             1  29.0516       788     376.56 7.048e-08 ***
## alcohol               1  14.8786       787     361.68 0.0001147 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# check multicollinearity
vif(glm.fit)
```

```
##        fixed.acidity     volatile.acidity          citric.acid
##             9.381108             1.678724             3.626691
##       residual.sugar            chlorides  free.sulfur.dioxide
##             1.629977             1.358625             2.623342
## total.sulfur.dioxide              density                   pH
##             2.883413             5.952404             3.762087
##            sulphates              alcohol
##             1.357697             2.124702
```

The development table was analyzed using the anova() function. Here, depending on how large the difference between null development and residual development is, how far it is from the null model. The null model refers to a state in which there is only an intercept value without any input variables. It can be seen that all variables are suitable except for residual.sugar and free.sulfur.dioxide. In addition, when checking multicollinearity, it can be seen that there is no problem with multicollinearity because all variables are less than 10. Through this, it can be seen that all variables can be used to form a data model.

```
# prediction for all values (logistic)
glm.pred <- predict(glm.fit, test_wine, type = "response")
# create all values '0'
pred_glm <- rep(0, length(glm.pred))
# all of the elements for which predicted prob of a quality increase exceeds
# 0.5
pred_glm[glm.pred > 0.5] = 1
# confusion matrix
table(pred_glm, rating_wine)
```

```
##         rating_wine
## pred_glm   0   1
##        0 638  89
##        1  25  47
```

```r
# find accuracy
mean(pred_glm == rating_wine)
```

```
## [1] 0.8573217
```

```r
# find error rate
mean(pred_glm != rating_wine)
```

```
## [1] 0.1426783
```

```r
# find recall
recall <- 47/(47 + 25)
# find precision
precision <- 47/(47 + 89)
# find f1 score
f1.score <- 2 * ((precision * recall)/(precision + recall))
f1.score
```

```
## [1] 0.4519231
```

Percentage of current logistic predictions:
Accuracy: **85.73** %
Error rate: **14.27** %
F-1 Score: **45.19** %

5.2 LDA

```r
set.seed(1)
# fit LDA
lda.fit <- lda(rating ~ . - quality, data = red.wine, subset = train)
# predictions data (LDA)
lda.pred <- predict(lda.fit, test_wine)
# contains LDA predictions about the movement of the quality
lda.class = lda.pred$class
# confusing matrix
table(lda.class, rating_wine)
```

```
##          rating_wine
## lda.class   0   1
##         0 610  67
##         1  53  69
```

```r
# find accuracy
mean(lda.class == rating_wine)
```

```
## [1] 0.8498123
```

```r
# find error rate
mean(lda.class != rating_wine)
```

```
## [1] 0.1501877
```

```r
# find recall
recall <- 69/(69 + 53)
# find precision
precision <- 69/(69 + 67)
# find f1 score
f1.score <- 2 * ((precision * recall)/(precision + recall))
f1.score
```

```
## [1] 0.5348837
```

Percentage of LDA predictions:
Accuracy: **84.98** %
Error rate: **15.02** %
F-1 Score: **53.49** %

5.3 QDA

```r
set.seed(1)
# fit QDA
qda.fit <- qda(rating ~ . - quality, data = red.wine, subset = train)
# predictions data (QDA)
qda.pred <- predict(qda.fit, test_wine)
# contains QDA predictions about the movement of the quality
qda.class = qda.pred$class
# confusing matrix
table(qda.class, rating_wine)
```

```
##          rating_wine
## qda.class   0   1
##         0 549  50
##         1 114  86
```

```r
# find accuracy
mean(qda.class == rating_wine)
```

```
## [1] 0.7947434
```

```r
# find error rate
mean(qda.class != rating_wine)
```

```
## [1] 0.2052566
```

```r
# find recall
recall <- 86/(86 + 114)
# find precision
precision <- 86/(86 + 50)
# find f1 score
f1.score <- 2 * ((precision * recall)/(precision + recall))
f1.score
```

```
## [1] 0.5119048
```

Percentage of QDA predictions:
Accuracy: **79.47** %
Error rate: **20.53** %
F-1 Score: **51.19** %

5.4 KNN

```r
# create train data (KNN)
train_x_wine <- cbind(red.wine$fixed.acidity, red.wine$volatile.acidity, red.wine$citric.acid,
    red.wine$residual.sugar, red.wine$chlorides, red.wine$free.sulfur.dioxide, red.wine$pH,
    red.wine$density, red.wine$sulphates, red.wine$alcohol, red.wine$total.sulfur.dioxide)[train,
    ]
# create test data (KNN)
test_x_wine <- cbind(red.wine$fixed.acidity, red.wine$volatile.acidity, red.wine$citric.acid,
    red.wine$residual.sugar, red.wine$chlorides, red.wine$free.sulfur.dioxide, red.wine$pH,
    red.wine$density, red.wine$sulphates, red.wine$alcohol, red.wine$total.sulfur.dioxide)[test,
    ]
# vector containing the class labels for the training observations
train_rating_test <- rating_wine[train]
```

5.4.1 K = 1

```r
set.seed(1)
# predictions data in the KNN 1
knn.1 = knn(train_x_wine, test_x_wine, rating_wine, k = 1)
# confusing matrix
table(knn.1, train_rating_test)
```

```
##      train_rating_test
## knn.1   0   1
##     0 560 107
##     1 103  29
```

```r
# find accuracy
mean(knn.1 == train_rating_test)
```

```
## [1] 0.7371715
```

```
# find error rate
mean(knn.1 != train_rating_test)
```

```
## [1] 0.2628285
```

Percentage of KNN k = 1 predictions:
Accuracy: **73.72** %
Error rate: **26.28** %

5.4.2 K = 5

```
set.seed(1)
# predictions data in the KNN 5
knn.5 = knn(train_x_wine, test_x_wine, rating_wine, k = 5)
# confusing matrix
table(knn.5, train_rating_test)
```

```
##      train_rating_test
## knn.5   0   1
##     0 643 132
##     1  20   4
```

```
# find accuracy
mean(knn.5 == train_rating_test)
```

```
## [1] 0.8097622
```

```
# find error rate
mean(knn.5 != train_rating_test)
```

```
## [1] 0.1902378
```

Percentage of KNN k = 5 predictions:
Accuracy: **80.98** %
Error rate: **19.02** %

5.4.3 K = 7

```
set.seed(1)
# predictions data in the KNN 7
knn.7 = knn(train_x_wine, test_x_wine, rating_wine, k = 7)
# confusing matrix
table(knn.7, train_rating_test)
```

```
##      train_rating_test
## knn.7   0   1
##     0 650 135
##     1  13   1
```

```r
# find accuracy
mean(knn.7 == train_rating_test)
```

```
## [1] 0.8147685
```

```r
# find error rate
mean(knn.7 != train_rating_test)
```

```
## [1] 0.1852315
```

Percentage of KNN k = 7 predictions:
Accuracy: **81.48** %
Error rate: **18.52** %

5.4.4 K = 9

```r
set.seed(1)
# predictions data in the KNN 9
knn.9 = knn(train_x_wine, test_x_wine, rating_wine, k = 9)
# confusing matrix
table(knn.9, train_rating_test)
```

```
##      train_rating_test
## knn.9   0   1
##     0 655 135
##     1   8   1
```

```r
# find accuracy
mean(knn.9 == train_rating_test)
```

```
## [1] 0.8210263
```

```r
# find error rate
mean(knn.9 != train_rating_test)
```

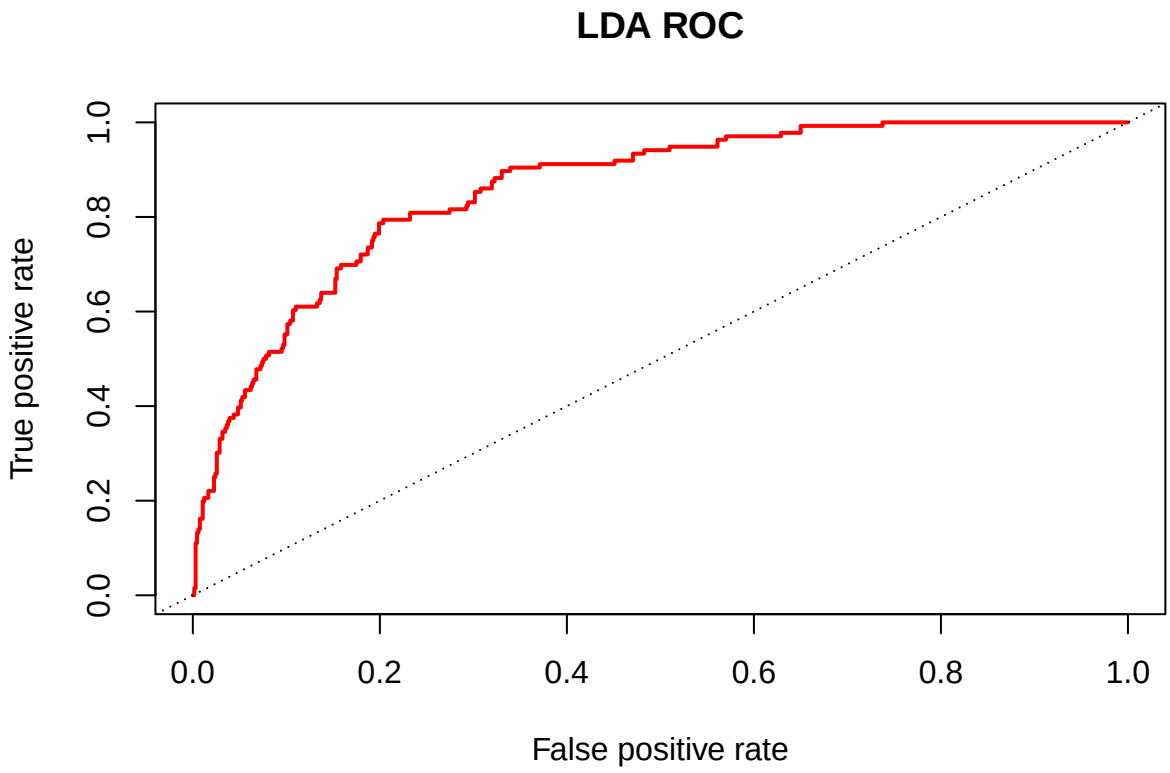
```
## [1] 0.1789737
```

Percentage of KNN k = 9 predictions:
Accuracy: **82.10** %
Error rate: **17.90** %

6 Compare Model

6.1 LDA ROC Curve

```r
# the classification rate is calculated by comparing the calculated probability
# p with the actual test data; LDA
pred.LDA <- prediction(lda.pred$posterior[, 2], rating_wine)
# calculate the sensitivity and 1-specificity to draw the ROC curve
pefLDA <- performance(pred.LDA, measure = "tpr", x.measure = "fpr")
# show LDA ROC plot
plot(pefLDA, col = "red", main = "LDA ROC", lwd = 2, lty = 1)
abline(0, 1, lty = 3)
```




```r
# AUC
aic <- performance(pred.LDA, measure = "auc")
aic <- aic@y.values[[1]]
aic
```

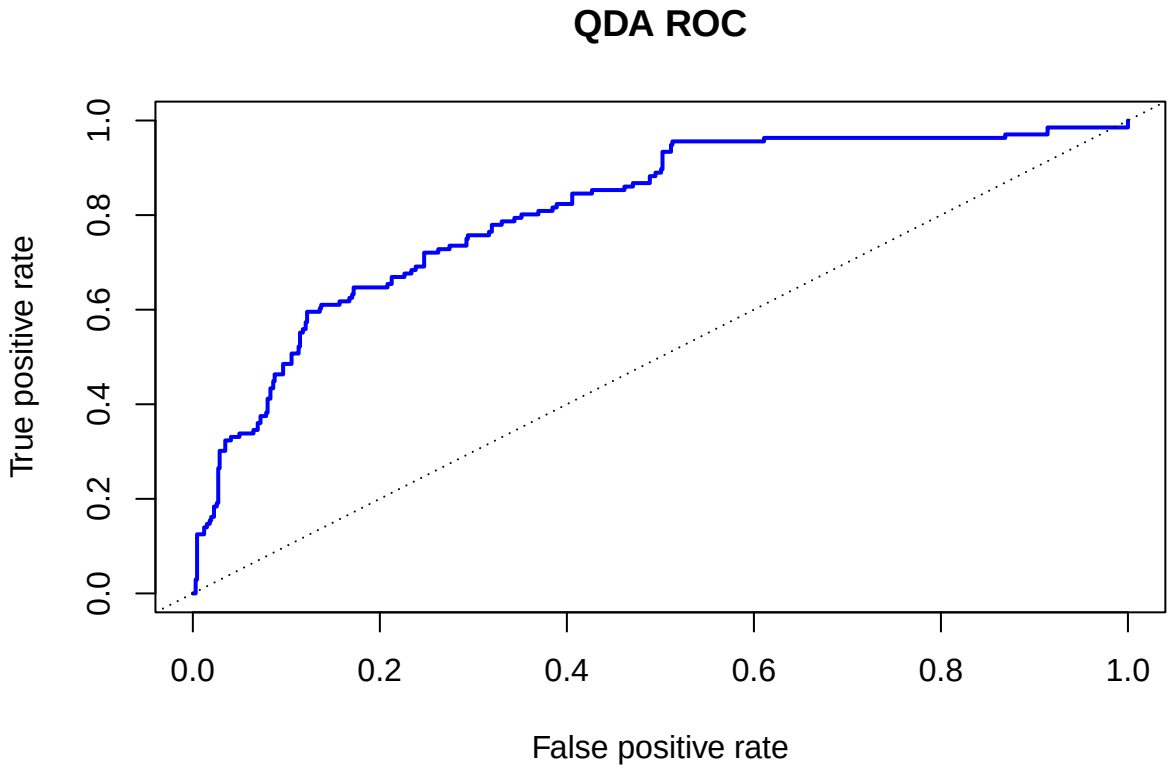
```
## [1] 0.8591851
```

LDA Model's AUC: **0.8591851**

6.2 QDA ROC Curve

```r
# the classification rate is calculated by comparing the calculated probability
# p with the actual test data; QDA
```

```r
pred.QDA <- prediction(qda.pred$posterior[, 2], rating_wine)
# calculate the sensitivity and 1-specificity to draw the ROC curve
pefQDA <- performance(pred.QDA, measure = "tpr", x.measure = "fpr")
# show QDA ROC plot
plot(pefQDA, main = "QDA ROC", col = "blue", lty = 1, lwd = 2)
abline(0, 1, lty = 3)
```




```r
# AUC
auc.qda <- performance(pred.QDA, measure = "auc")
auc.qda <- auc.qda@y.values[[1]]
auc.qda
```

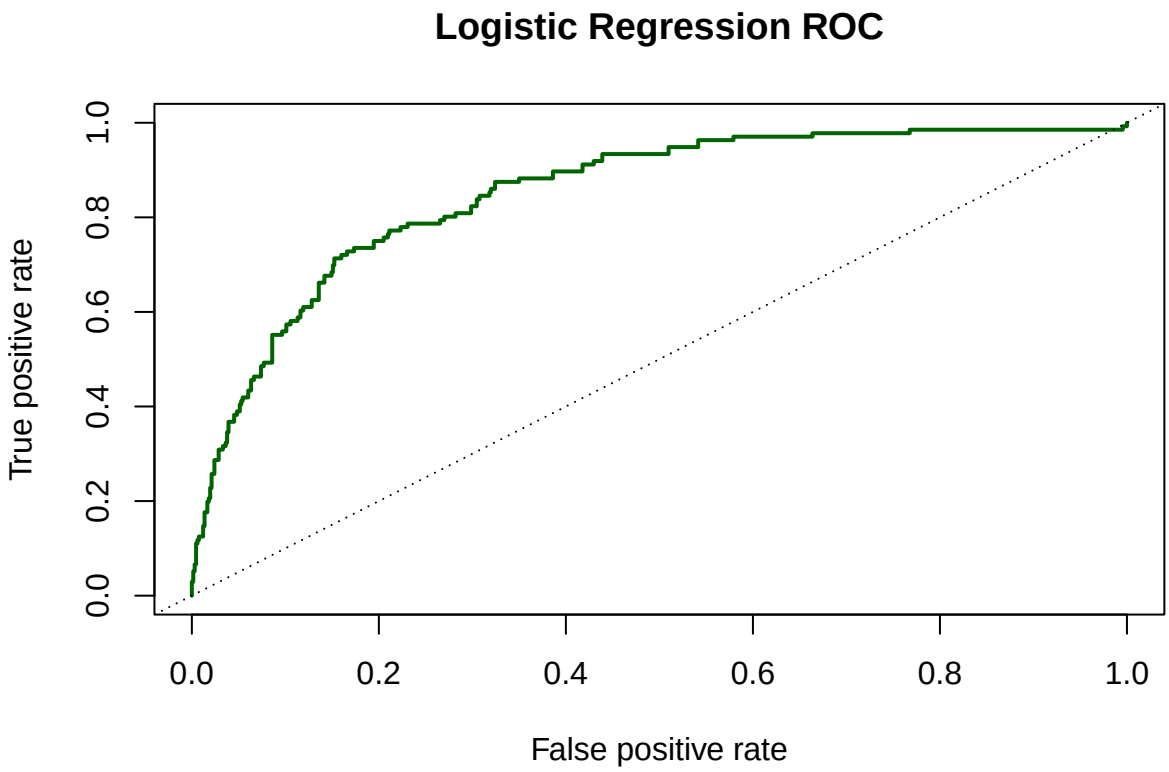
```
## [1] 0.8081803
```

QDA Model's AUC: **0.8081803**

6.3 Logistic Regression ROC Curve

```r
# the classification rate is calculated by comparing the calculated probability
# p with the actual test data; Logistic
predLR <- prediction(glm.pred, rating_wine)
# calculate the sensitivity and 1-specificity to draw the ROC curve
```

```r
pefLR <- performance(predLR, measure = "tpr", x.measure = "fpr")
# show Logistic ROC plot
plot(pefLR, main = "Logistic Regression ROC", col = "darkgreen", lty = 1, lwd = 2)
abline(0, 1, lty = 3)
```




```r
# AUC
auc <- performance(predLR, measure = "auc")
auc <- auc@y.values[[1]]
auc
```

```
## [1] 0.8503571
```

Logistic Regression Model's AUC: **0.8503571**

As a result, each accuracy of the prediction is as follows.
Logistic Regression: **85.73%**
LDA: **84.98%**
QDA: **79.47%**
KNN (k=1): **73.72 %**
KNN (k=5): **80.98 %**
KNN (k=7): **81.48 %**
KNN (k=9): **82.10 %**

As a result of checking with accuracy, it can be seen that all models have not bad accuracy. Among them, the accuracy of logistic regression analysis and lda analysis is judged to be very accurate.

Accuracy:

$$Logistic > LDA > KNN(k = 9) > KNN(k = 7) > KNN(k = 5) > QDA > KNN(k = 1)$$

When checking the KNN model, it can be seen that as the value of K increases, the accuracy increases and the error rate improves. The KNN model has different results as the value of k increases and excludes it from comparing the most appropriate models to predict through simple distances between observations rather than focusing on the importance of variables.

Therefore, the most suitable models can be considered are logistic regression models and LDA models. Although the logistic regression model has a slightly higher accuracy, it is not the most suitable model just because it is highly accurate, so we checked using f1 score because the data is enhanced.
Logistic regression model F-1 score: **0.4519231**
LDA F-1 score: **0.5348837**
Indicating that LDA is a more suitable model.

To compare more reliably, the ROC curve and AUC were used as above. (LDA, QDA, Logistic Regression) For a model with a perfect ROC curve graph, TPR is 1 and FPR is 0 for all data points. It also shows that the larger the value of AUC, the better the performance of the model.
As a result, considering that the AUC value of QDA is smaller than the other two models and that the QDA ROC curve TPR is not closer to 1 and FPR is not closer to 0 than the two models, QDA is not an optimal model. In addition, the ROC curve of LDA and ROC curve of the logistic regression model look similar, but when I look at the AUC value, I can see that the **LDA model is more suitable**, given that the *AUC of the LDA is 0.8591851* and the *AUC of the logistic regression model is 0.8503571.*

7 Conclusion

- Individuals of red wine quality were identified and classified into two types (bad, good) to statistically evaluate wine quality, **which showed 84.98% accuracy of the LDA model and 85.92% better and more suitable for the AUC value than other models.** As a result, the **LDA model is the most suitable model**.

- Additionally, in the red wine quality dataset, there is no problem with the multicollinearity line of all variables because the multicollinearity line of all variables is less than 10, and alcohol has the strongest correlation in wine quality.