HW3_STAT515_Mykola_Signayevskyy

Mykola Signayesvkyy

2023-04-27

Problem 1: Logistic Regression

This question should be answered using the "Banknote Authentication" data set. Description about the data set can be found on the link provided. Objective of this question is to fit an logistic regression model to classify forged banknote from genuine banknotes. (Presumably 0 for genuine and 1 for forged bank notes)

```
banknote <- read.table("/Users/mykola/Desktop/STAT515/hw3/banknote_authentication(1).txt", header=TRUE,
head(banknote)
```

```
##   Variance skewness curtosis  entropy class
## 1  3.62160   8.6661  -2.8073 -0.44699     0
## 2  4.54590   8.1674  -2.4586 -1.46210     0
## 3  3.86600  -2.6383   1.9242  0.10645     0
## 4  3.45660   9.5228  -4.0112 -3.59440     0
## 5  0.32924  -4.4552   4.5718 -0.98880     0
## 6  4.36840   9.6718  -3.9606 -3.16250     0
```

```
any(is.na(banknote))
```

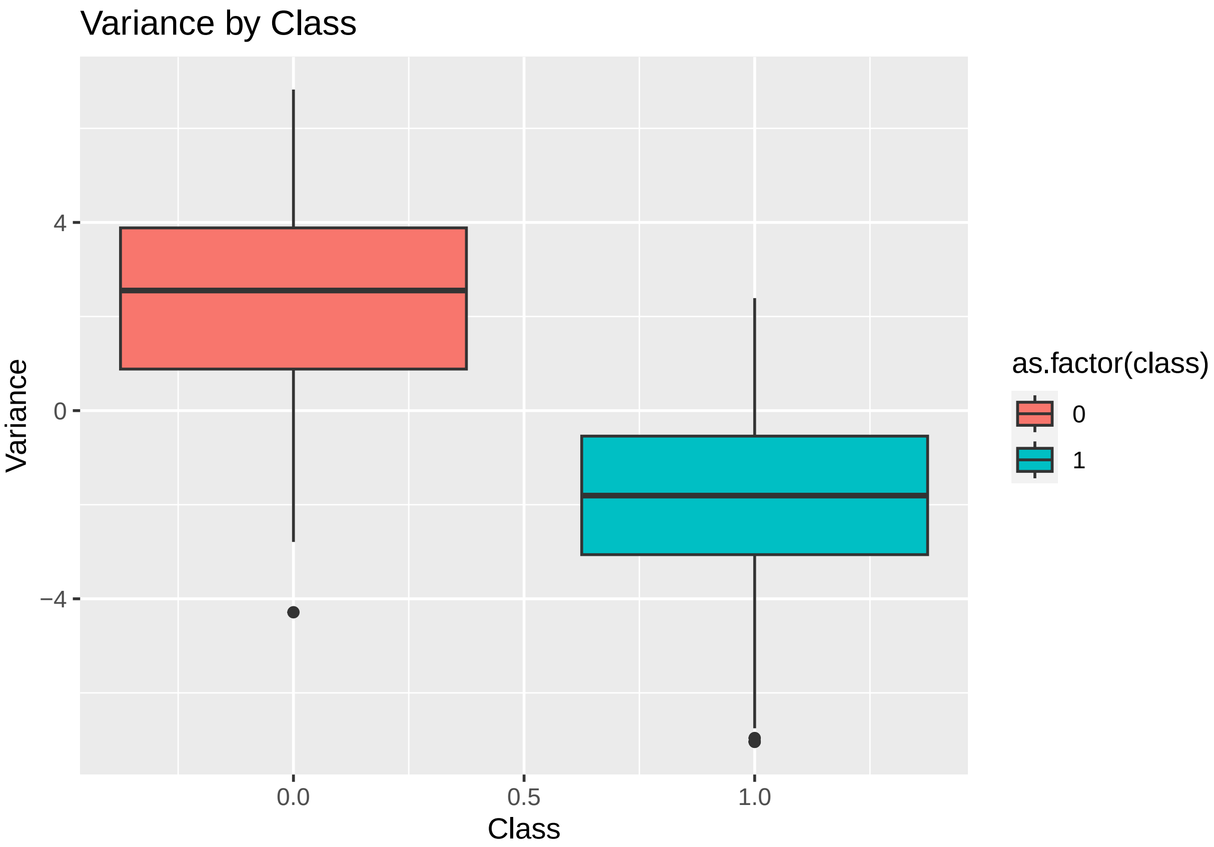
```
## [1] FALSE
```

Produce some numerical and graphical summaries of the data set. Explain the relationships.

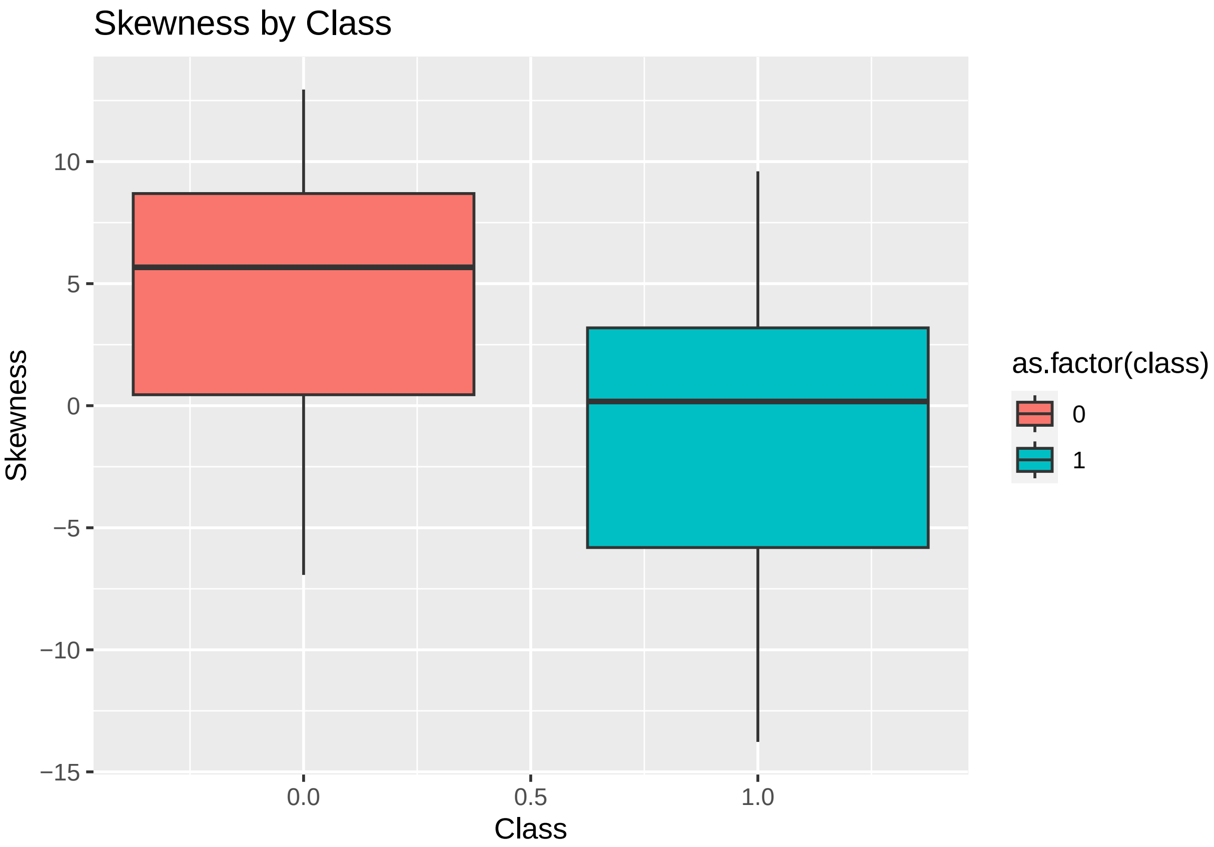
```
summary(banknote)
```

```
##     Variance          skewness          curtosis          entropy
##  Min.   :-7.0421   Min.   :-13.773   Min.   :-5.2861   Min.   :-8.5482
##  1st Qu.:-1.7730   1st Qu.: -1.708   1st Qu.:-1.5750   1st Qu.:-2.4135
##  Median : 0.4962   Median :  2.320   Median : 0.6166   Median :-0.5867
##  Mean   : 0.4337   Mean   :  1.922   Mean   : 1.3976   Mean   :-1.1917
##  3rd Qu.: 2.8215   3rd Qu.:  6.815   3rd Qu.: 3.1793   3rd Qu.: 0.3948
##  Max.   : 6.8248   Max.   : 12.952   Max.   :17.9274   Max.   : 2.4495
##      class
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.4446
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

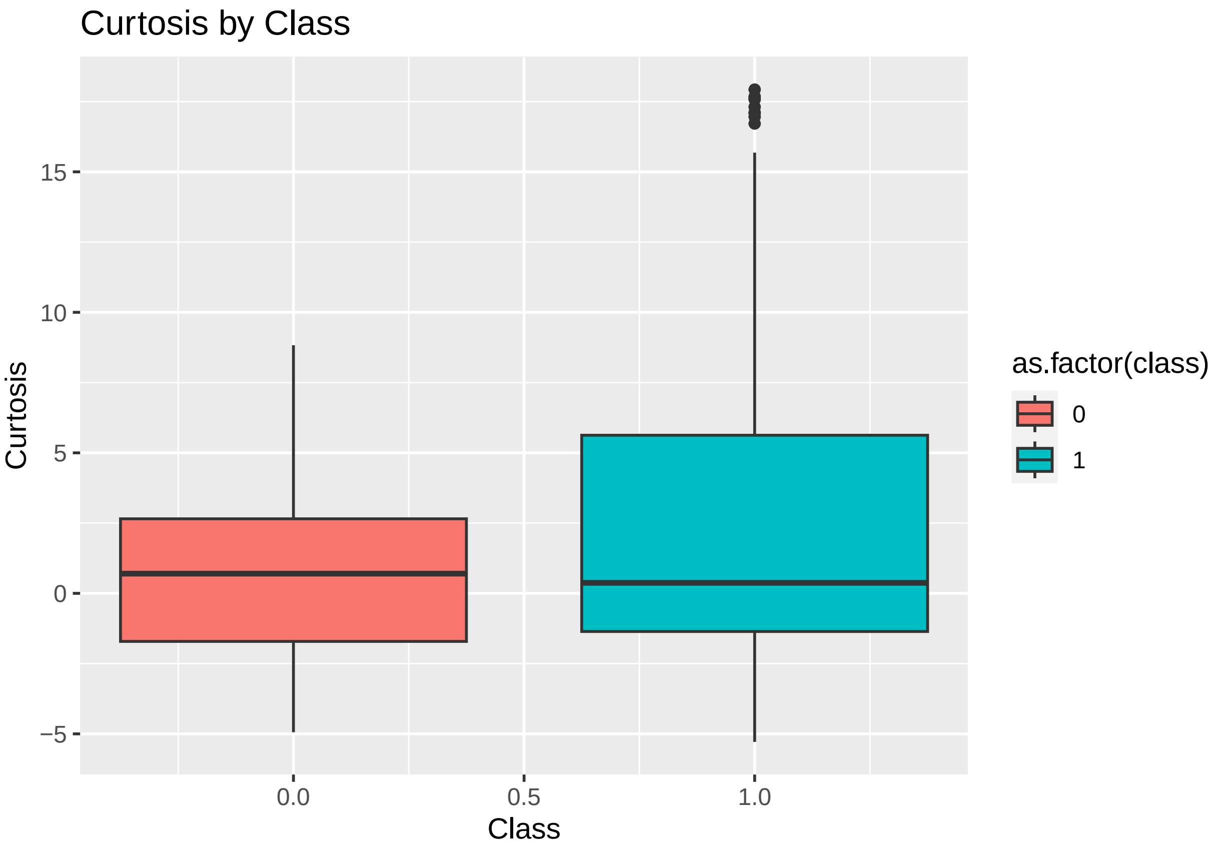
```
library(ggplot2)
ggplot(banknote, aes(x=class, y=Variance, fill=as.factor(class))) +
  geom_boxplot() +
  labs(title="Variance by Class", x="Class", y="Variance")
```



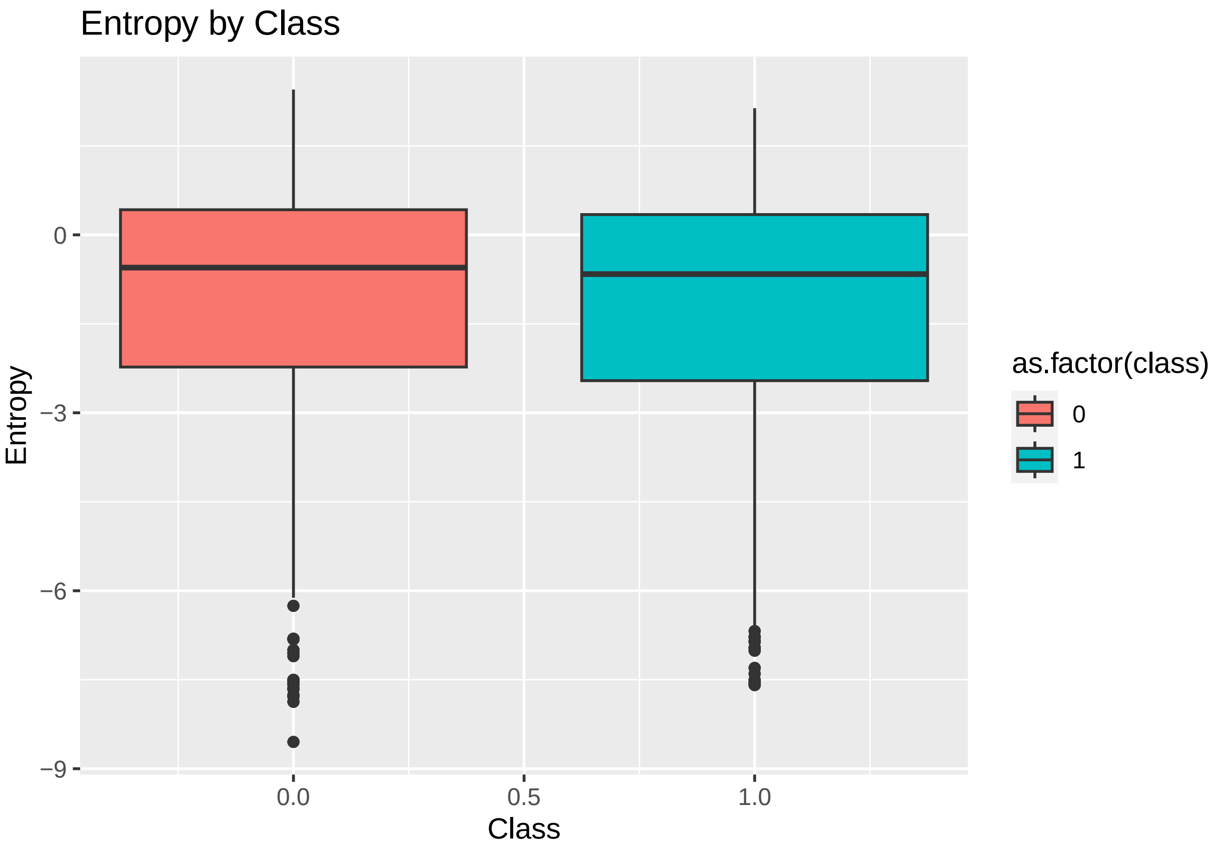
```
ggplot(banknote, aes(x=class, y=skewness, fill=as.factor(class))) +
  geom_boxplot() +
  labs(title = "Skewness by Class", x='Class', y='Skewness')
```



```
ggplot(banknote, aes(x=class, y=curtosis, fill=as.factor(class))) +
  geom_boxplot() +
  labs(title = 'Curtosis by Class', x="Class", y="Curtosis")
```



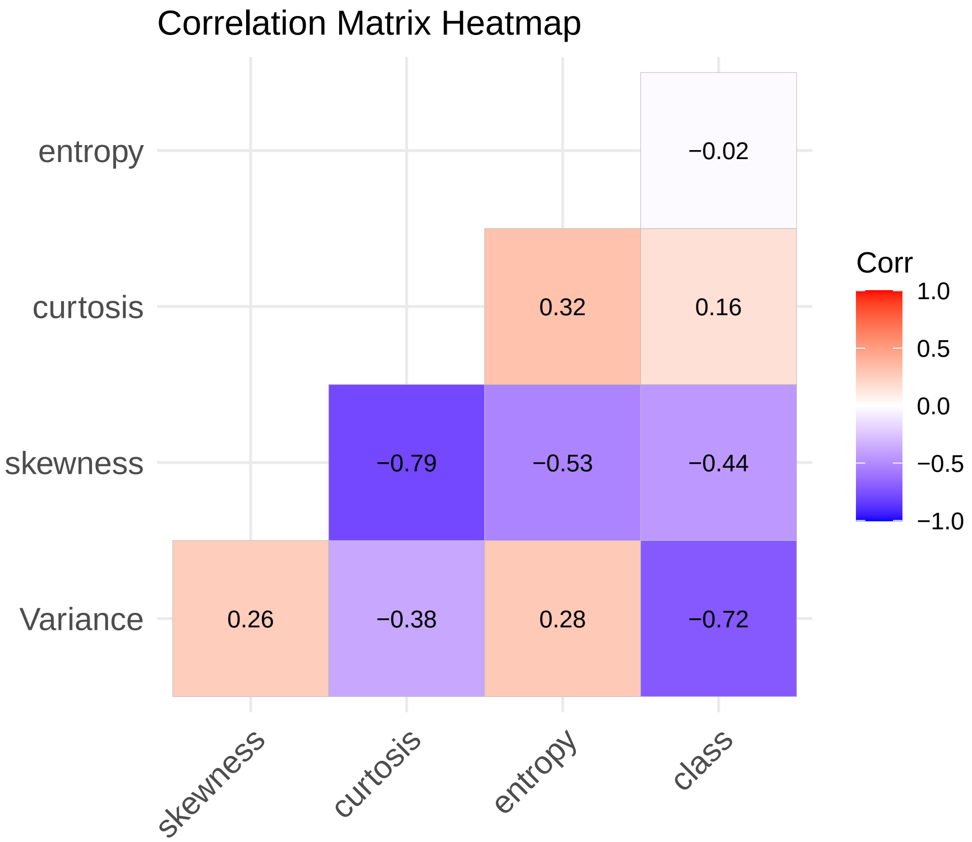
```
ggplot(banknote, aes(x=class, y=entropy, fill=as.factor(class))) +
  geom_boxplot() +
  labs(title = "Entropy by Class", x="Class", y="Entropy")
```



```r
library(ggcorrplot)

# I am computing the correlation matrix
corr_matrix <- cor(banknote[, 1:5])

ggcorrplot(corr_matrix,
           type = "lower",
           lab = TRUE,
           lab_size = 3,
           colors = c("BLUE", "WHITE", "RED"),
           title = "Correlation Matrix Heatmap")
```



Variance has the biggest negative relationship for forged banknotes. Also I can see that variance has positive correlations with skewness and entropy.

Skewness has negative relationship for forged banknotes, but not that significant as Variance. Skewness has a very big negative correlation with curtosis and smaller negative correlations with entropy.

Curtosis has a small positive correlations with class. Also, curtosis has a very big negative correlation with Skewness.

Entropy does not have significant relationship with class. It has some positive correlations with variance and curtosis, negative correlation with skewness.

Is this a balanced data set?.

```
table(banknote$class)
```

```
##
##   0   1
## 762 610
```

I see that there are more genuine banknotes than forged ones. I would say that the difference is not that huge, but the dataset is not balanced.

Use the full data set to perform a logistic regression with Class as the response variable. Do any of the predictors appear to be statistically significant? If so, which ones?

```
logit_model <- glm(class ~ ., data = banknote, family = binomial)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(logit_model)
```

```
##
## Call:
## glm(formula = class ~ ., family = binomial, data = banknote)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.70001   0.00000   0.00000   0.00029   2.24614
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   7.3218     1.5589   4.697 2.64e-06 ***
## Variance     -7.8593     1.7383  -4.521 6.15e-06 ***
## skewness     -4.1910     0.9041  -4.635 3.56e-06 ***
## curtosis     -5.2874     1.1612  -4.553 5.28e-06 ***
## entropy      -0.6053     0.3307  -1.830   0.0672 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1885.122  on 1371  degrees of freedom
## Residual deviance:   49.891  on 1367  degrees of freedom
## AIC: 59.891
##
## Number of Fisher Scoring iterations: 12
```

All of the variables are statistically significant except entropy. So Variance, skewness, and curtosis

Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix id telling you about the types of mistakes made by logistic regression.

```
#confusion matrix
predicted_class <- ifelse(predict(logit_model, type = "response") > 0.5, 1, 0)
confusion_matrix <- table(predicted_class, banknote$class)
accuracy <- sum(diag(confusion_matrix))/sum(confusion_matrix)
confusion_matrix
```

```
##
## predicted_class   0   1
##               0 757   6
##               1   5 604
```

```
accuracy
```

```
## [1] 0.9919825
```

I see that there are 757 True negatives and 5 False negatives. Meaning that out of 762 genuine banknotes, 757 were correctly classified while 5 are mistakenly were considered as fakes. Also, 604 banknotes were correctly classified as forged banknotes (True positives) and only 6 were mistakenly classified as genuine (False poisitives).

```
accuracy
```

```
## [1] 0.9919825
```

Accuracy is very high.

Create a training set with 80% of the observations, and a testing set containing the remaining 20%.Compute the confusion matrix and the overall fraction of correct prediction for the testing data set.

```
set.seed(123)
train_index <- sample(nrow(banknote), round(0.8 * nrow(banknote)))
train <- banknote[train_index, ]
test <- banknote[-train_index, ]
```

```
model <- glm(class ~ ., family = binomial, data = train)
```

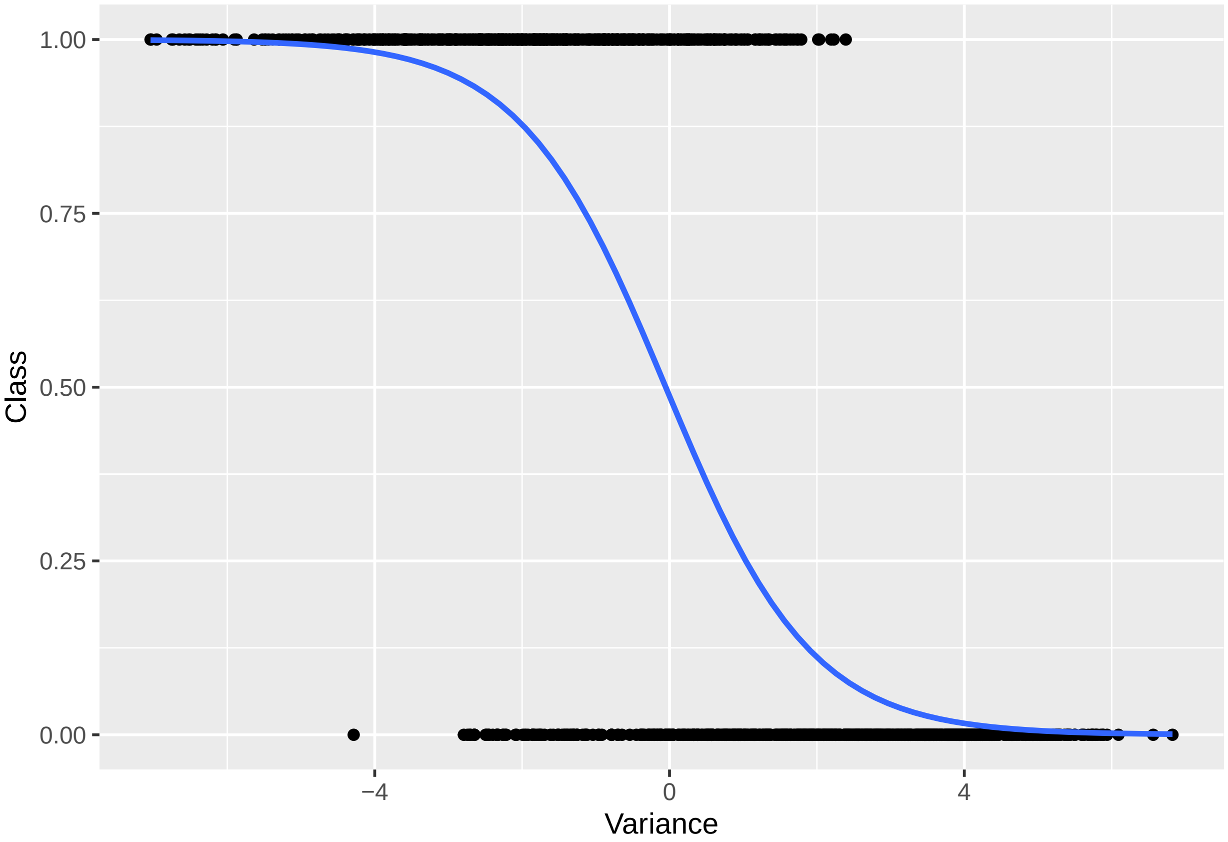
```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
test$predicted <- ifelse(predict(model, test, type = "response") > 0.5, "Real", "Fake")
table(test$class, test$predicted)
```

```
##
##     Fake Real
##   0  145    3
##   1    1  125
```

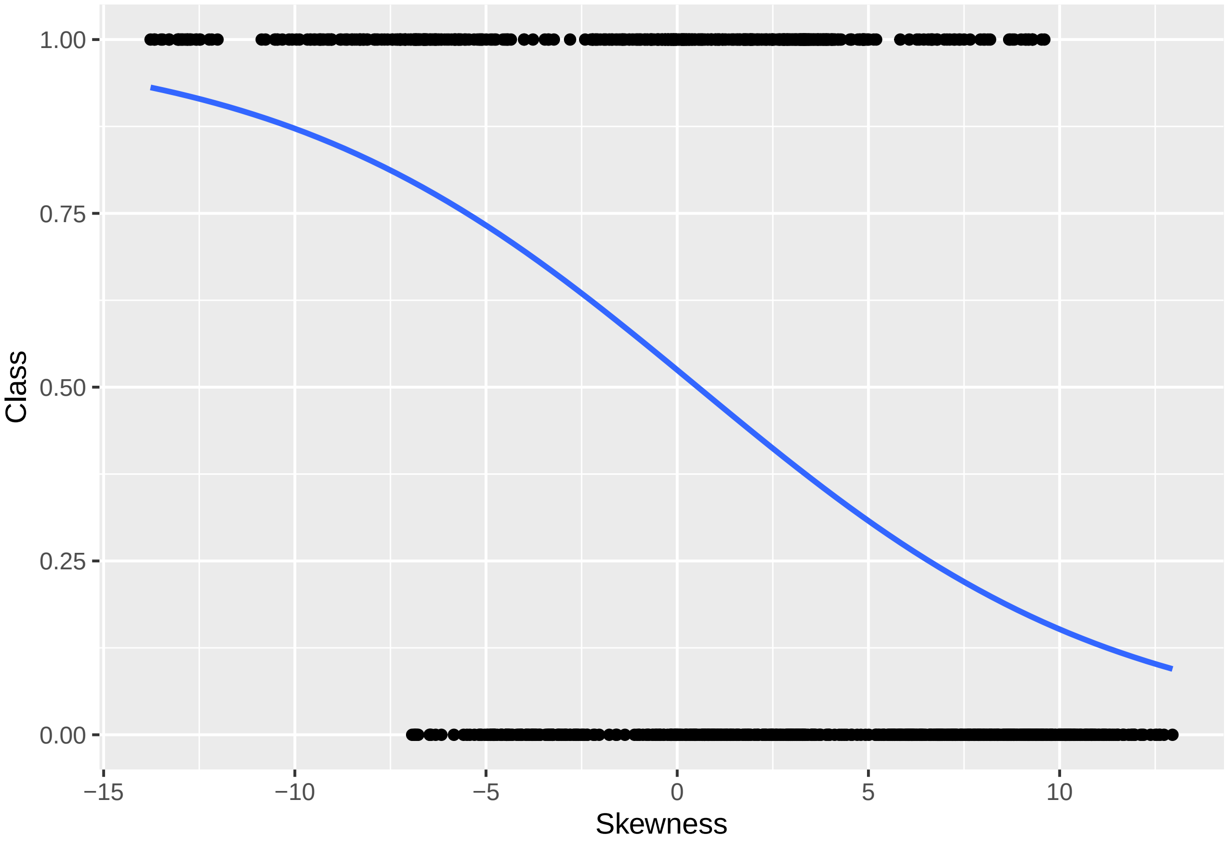
```
ggplot(banknote, aes(x = Variance, y = class)) +
  geom_point() +
  stat_smooth(method = "glm", method.args = list(family = "binomial"), se = FALSE) +
  xlab("Variance") +
  ylab("Class")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
ggplot(banknote, aes(x = skewness, y = class)) +
  geom_point() +
  stat_smooth(method = "glm", method.args = list(family = "binomial"), se = FALSE) +
  xlab("Skewness") +
  ylab("Class")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
sum(diag(table(test$class, test$predicted))) / nrow(test)
```

```
## [1] 0.9854015
```

Problem 2: Tree based models

This question should be answered using the "Wine Quality" data set. Description about the data set can be found on the link provided. Objective of this question is to fit an regression tree model to predict quality of wine.

```
wine <- read.csv("/Users/mykola/Desktop/STAT515/hw3/winequality(1).csv", header=TRUE, sep=";")
head(wine)
```

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1           7.0             0.27        0.36           20.7     0.045
## 2           6.3             0.30        0.34            1.6     0.049
## 3           8.1             0.28        0.40            6.9     0.050
## 4           7.2             0.23        0.32            8.5     0.058
## 5           7.2             0.23        0.32            8.5     0.058
## 6           8.1             0.28        0.40            6.9     0.050
##   free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 1                  45                  170  1.0010 3.00      0.45     8.8
## 2                  14                  132  0.9940 3.30      0.49     9.5
## 3                  30                   97  0.9951 3.26      0.44    10.1
## 4                  47                  186  0.9956 3.19      0.40     9.9
## 5                  47                  186  0.9956 3.19      0.40     9.9
## 6                  30                   97  0.9951 3.26      0.44    10.1
##   quality
## 1       6
## 2       6
## 3       6
## 4       6
## 5       6
## 6       6
```

```
any(is.na(wine))
```

```
## [1] FALSE
```

Produce some numerical and graphical summaries of the data set. Explain the relationships.

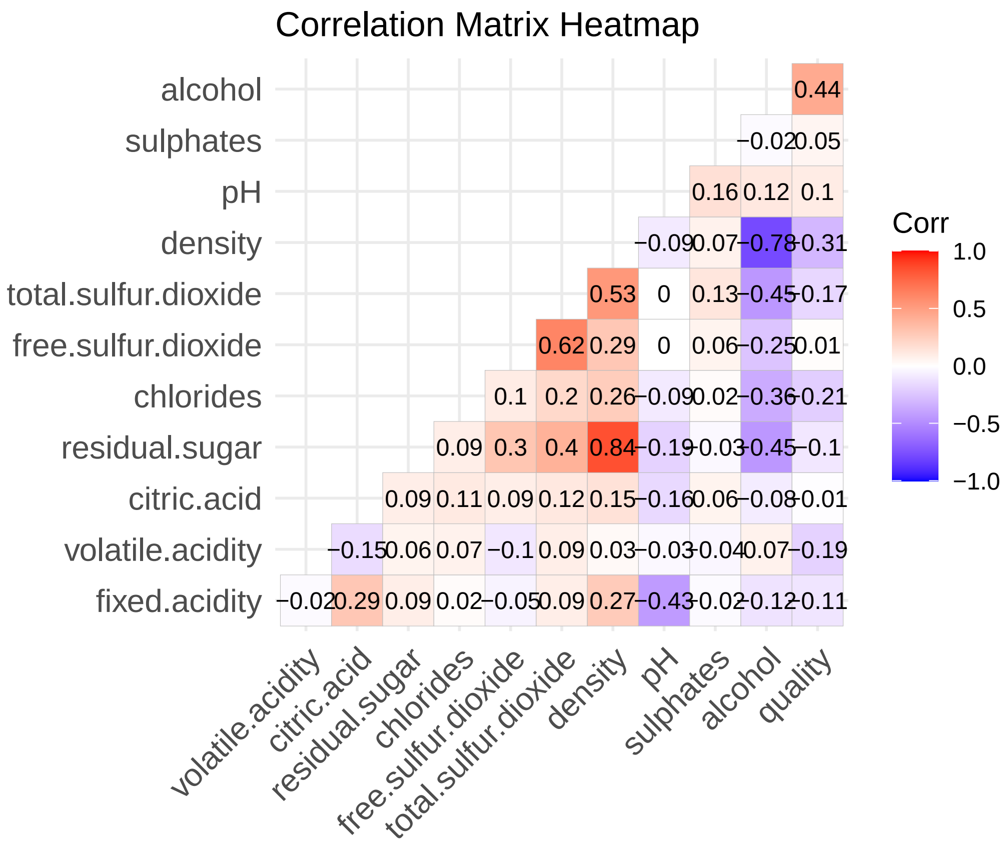
```
summary(wine)
```

```
##  fixed.acidity    volatile.acidity  citric.acid     residual.sugar
##  Min.   : 3.800   Min.   :0.0800   Min.   :0.0000   Min.   : 0.600
##  1st Qu.: 6.300   1st Qu.:0.2100   1st Qu.:0.2700   1st Qu.: 1.700
##  Median : 6.800   Median :0.2600   Median :0.3200   Median : 5.200
##  Mean   : 6.855   Mean   :0.2782   Mean   :0.3342   Mean   : 6.391
##  3rd Qu.: 7.300   3rd Qu.:0.3200   3rd Qu.:0.3900   3rd Qu.: 9.900
##  Max.   :14.200   Max.   :1.1000   Max.   :1.6600   Max.   :65.800
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide    density
##  Min.   :0.00900   Min.   :  2.00      Min.   :  9.0        Min.   :0.9871
##  1st Qu.:0.03600   1st Qu.: 23.00      1st Qu.:108.0        1st Qu.:0.9917
##  Median :0.04300   Median : 34.00      Median :134.0        Median :0.9937
##  Mean   :0.04577   Mean   : 35.31      Mean   :138.4        Mean   :0.9940
##  3rd Qu.:0.05000   3rd Qu.: 46.00      3rd Qu.:167.0        3rd Qu.:0.9961
##  Max.   :0.34600   Max.   :289.00      Max.   :440.0        Max.   :1.0390
##        pH          sulphates         alcohol          quality
##  Min.   :2.720   Min.   :0.2200   Min.   : 8.00   Min.   :3.000
##  1st Qu.:3.090   1st Qu.:0.4100   1st Qu.: 9.50   1st Qu.:5.000
##  Median :3.180   Median :0.4700   Median :10.40   Median :6.000
##  Mean   :3.188   Mean   :0.4898   Mean   :10.51   Mean   :5.878
##  3rd Qu.:3.280   3rd Qu.:0.5500   3rd Qu.:11.40   3rd Qu.:6.000
##  Max.   :3.820   Max.   :1.0800   Max.   :14.20   Max.   :9.000
```

I see that some variables are in different numerical scales, but as we use tree based model we can avoid normalizaton of the dataset.

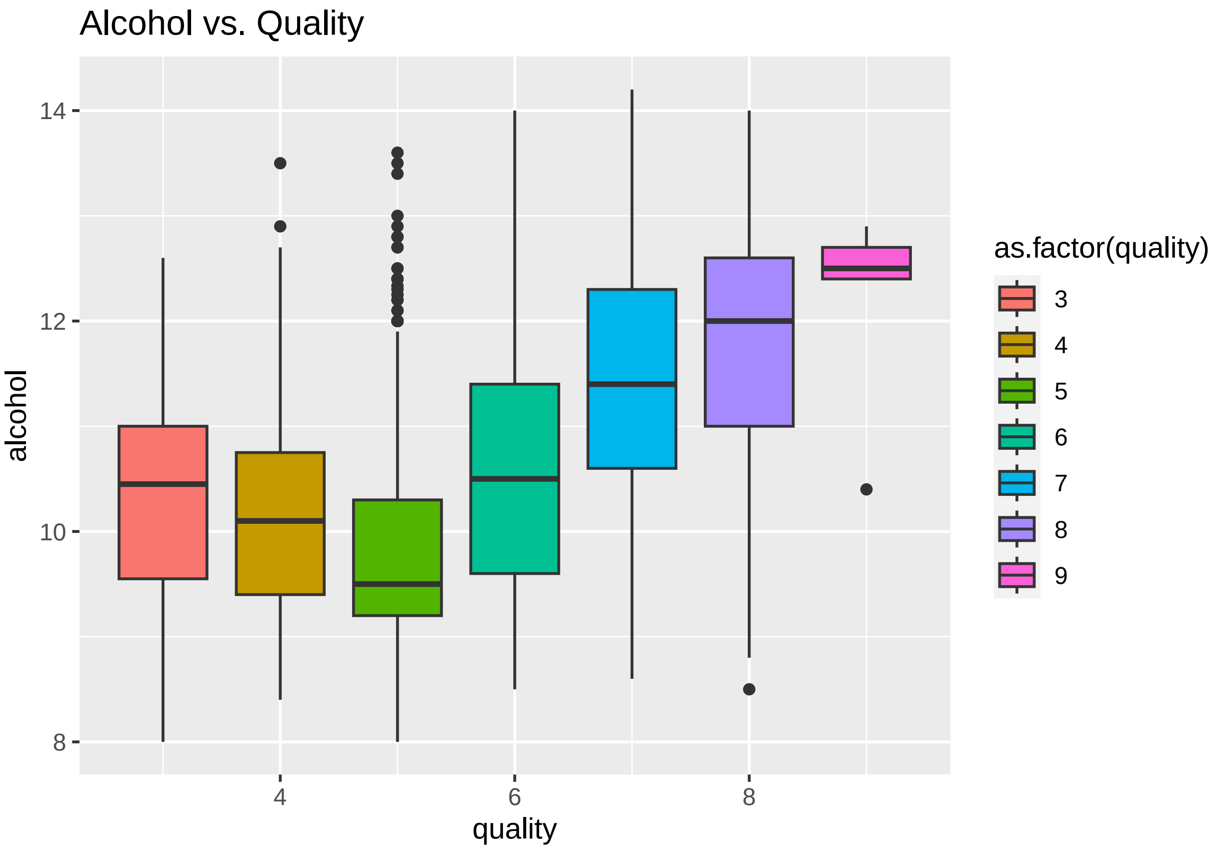
```
library(ggcorrplot)
# I am computing the correlation matrix
corr_matrix_w <- cor(wine[, 1:12])

ggcorrplot(corr_matrix_w,
           type = "lower",
           lab = TRUE,
           lab_size = 3,
           colors = c("BLUE", "#FFFFFF", "RED"),
           title = "Correlation Matrix Heatmap")
```



I see that alcohol has highest positive relationship with quality. Also, density has some negative relationship with quality. I can also see some interesting correlations between residual sugar and density (positive), alcohol and density (negative), free and total dioxide (positive).

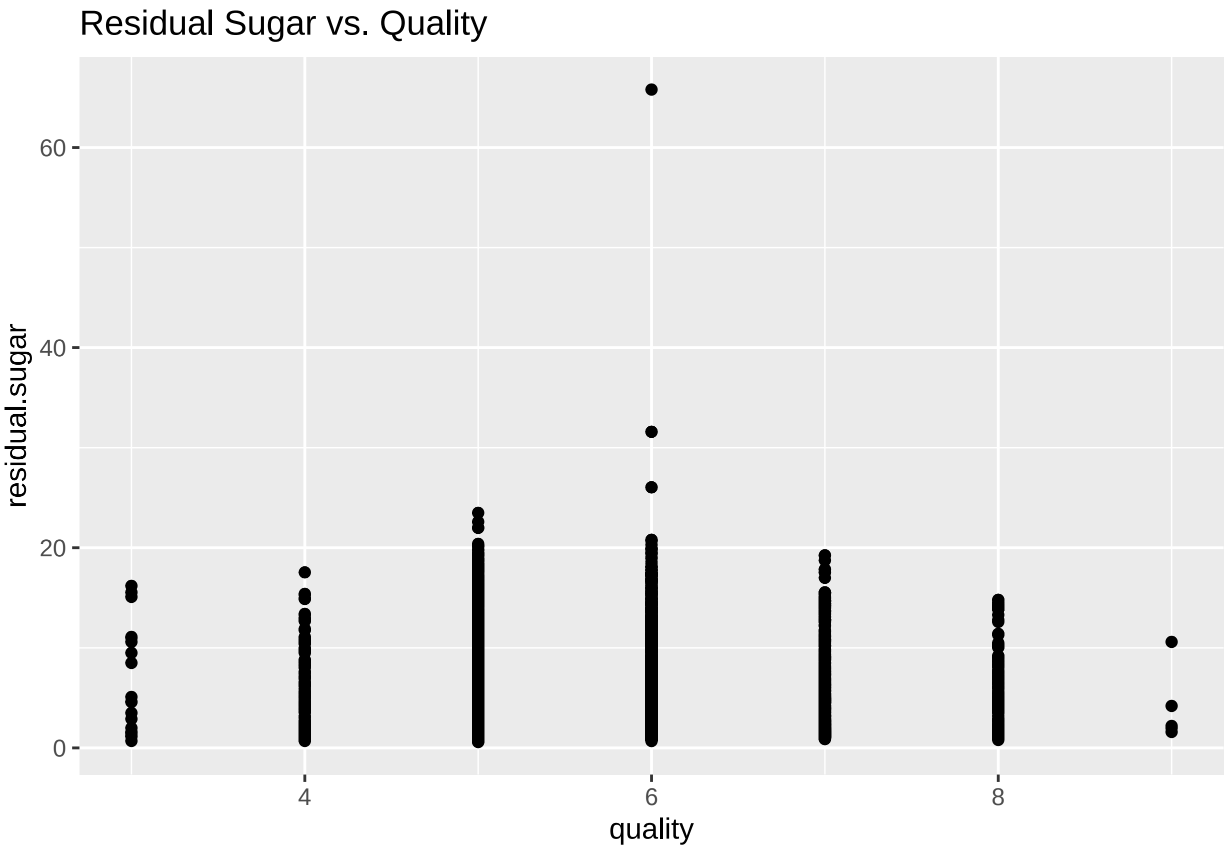
```
ggplot(wine, aes(x = quality, y = alcohol, fill=as.factor(quality))) +
  geom_boxplot() +
  labs(title = "Alcohol vs. Quality")
```



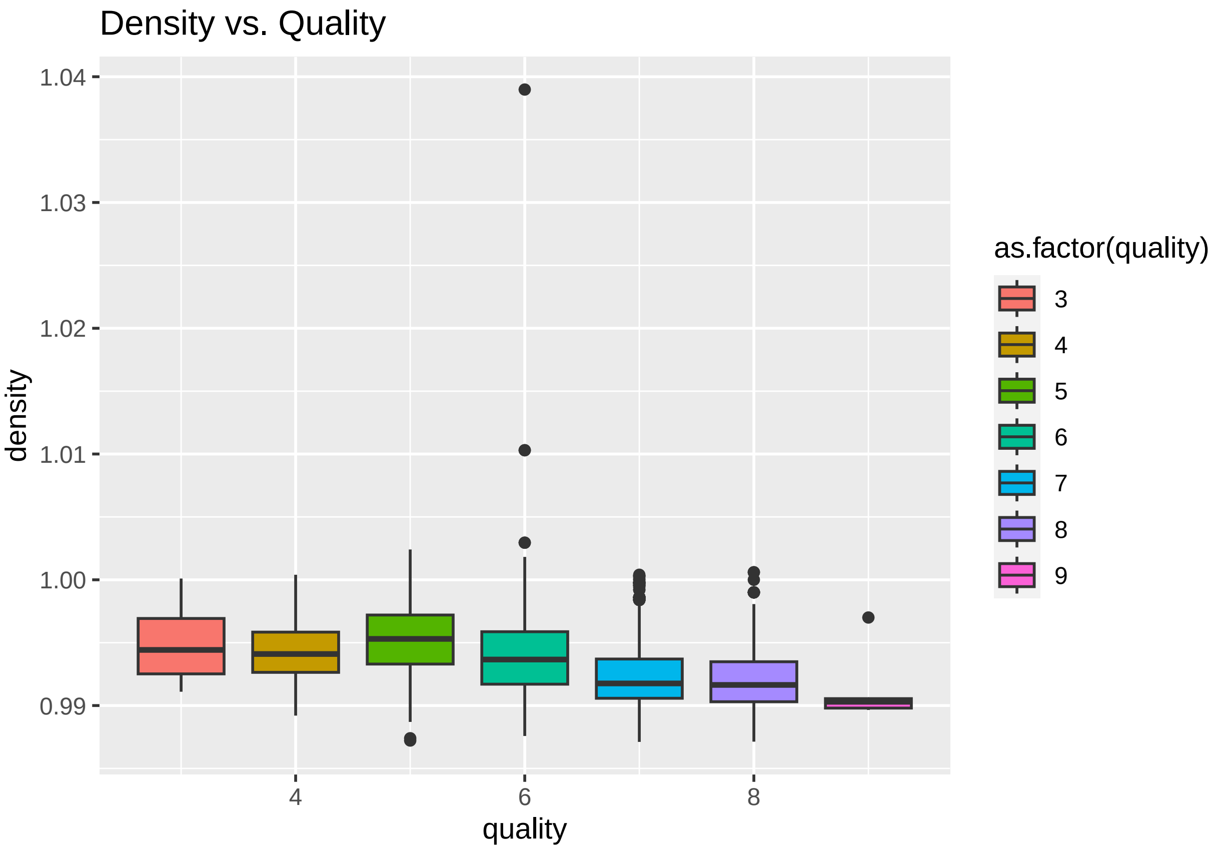
I

plotted alcohol to quality relationship to have a better obsirvation of this relathionship.

```
ggplot(wine, aes(x = quality, y = residual.sugar)) +
  geom_point() +
  labs(title = "Residual Sugar vs. Quality")
```



```
ggplot(wine, aes(x = quality, y = density, fill=as.factor(quality))) +
  geom_boxplot() +
  labs(title = "Density vs. Quality")
```



Create a training set with 80% of the observations, and a testing set containing the remaining 20%.

```
library(caret)
```

```
## Loading required package: lattice
```

```
set.seed(123)

index <- createDataPartition(wine$quality, p = 0.8, list = FALSE)
w_train <- wine[index, ]

w_test <- wine[-index, ]
```

Fit a regression tree with quality as the response variable using the training set. Plot the tree and interpret the results. What test MSE do you obtain?

```
library(rpart)
library(rpart.plot)
library(rattle)
```

```
## Loading required package: tibble
```

```
## Loading required package: bitops
```

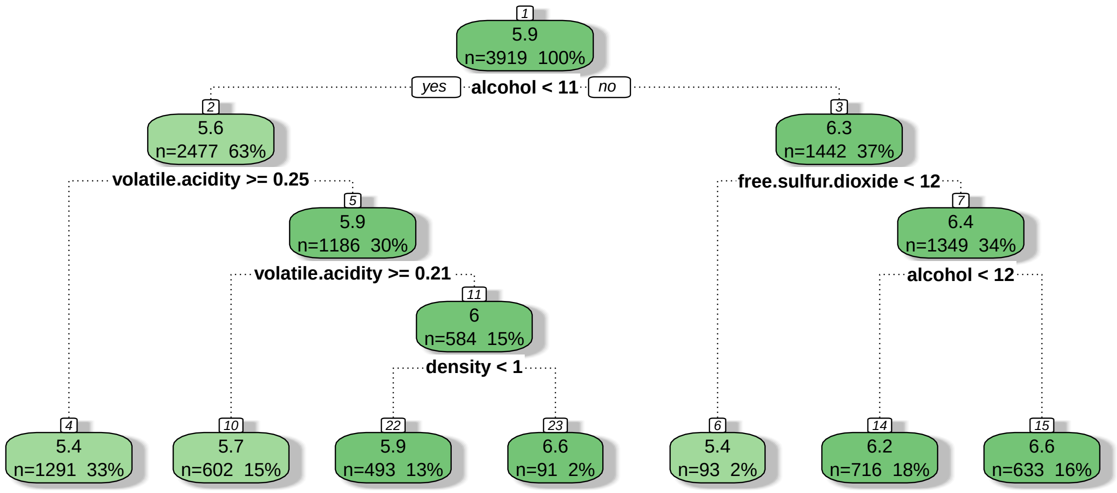
```
## Rattle: A free graphical interface for data science with R.
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
##
## Attaching package: 'rattle'

## The following object is masked _by_ '.GlobalEnv':
##
##     wine
```

```
wine_tree <- rpart(quality ~ ., data = w_train)
```

```
fancyRpartPlot(wine_tree)
```



Rattle 2023−Aug−01 03:31:16 mykola

```
#to get MSA I am going to apple model to test dataset to see actual error
wine_pred <- predict(wine_tree, newdata = w_test)

test_mse <- mean((w_test$quality - wine_pred)^2)
test_mse
```

```
## [1] 0.5924379
```

MSE is 0.5924379

Use cross-validation in order to determine the optimal level of tree complexity. Does pruning the tree improve the test MSE?

```
library(tree)

set.seed(123)
train_index <- sample(nrow(wine), nrow(wine)*0.8)
wine_train <- wine[train_index, ]
wine_test <- wine[-train_index, ]
wine_tree2 <- tree(quality ~ ., data = wine_train)
```

```
str(wine_test)
```

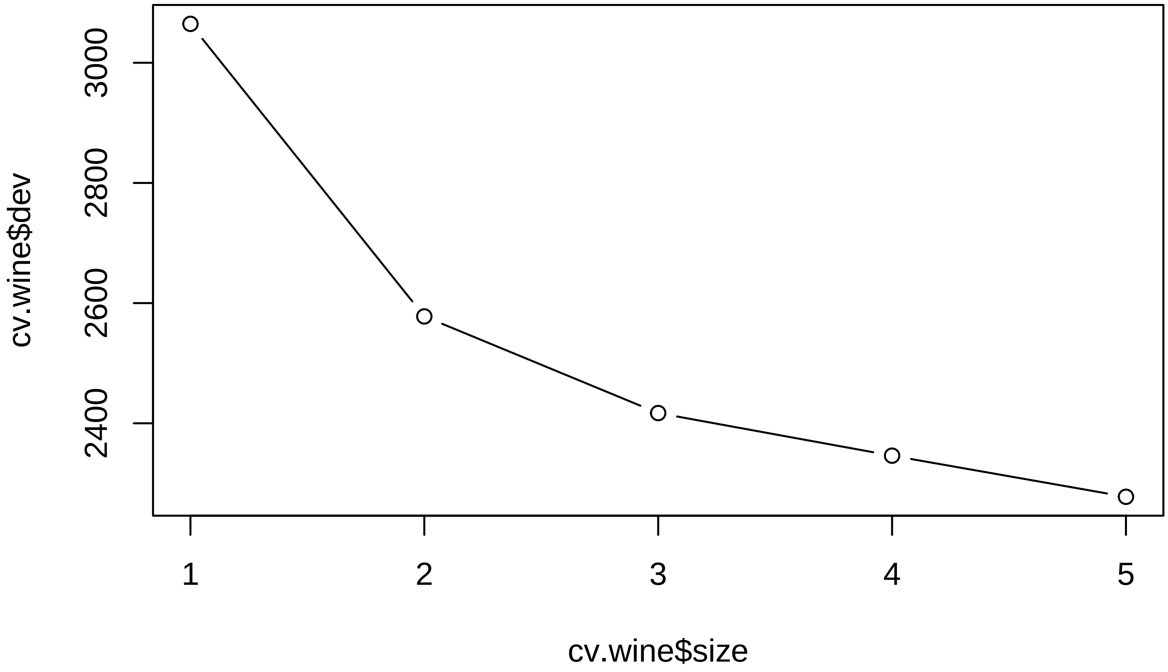
```
## 'data.frame':    980 obs. of  12 variables:
##  $ fixed.acidity       : num  8.1 7 6.8 7.6 7 6.2 6.9 6 6.6 6 ...
##  $ volatile.acidity    : num  0.28 0.27 0.26 0.67 0.33 0.46 0.19 0.19 0.38 0.27 ...
##  $ citric.acid         : num  0.4 0.36 0.42 0.14 0.32 0.25 0.35 0.26 0.15 0.28 ...
##  $ residual.sugar      : num  6.9 20.7 1.7 1.5 1.2 4.4 5 12.4 4.6 4.8 ...
##  $ chlorides           : num  0.05 0.045 0.049 0.074 0.053 0.066 0.067 0.048 0.044 0.063 ...
##  $ free.sulfur.dioxide : num  30 45 41 25 38 62 32 50 25 31 ...
##  $ total.sulfur.dioxide: num  97 170 122 168 138 207 150 147 78 201 ...
##  $ density             : num  0.995 1.001 0.993 0.994 0.991 ...
##  $ pH                  : num  3.26 3 3.47 3.05 3.13 3.25 3.36 3.3 3.11 3.69 ...
##  $ sulphates           : num  0.44 0.45 0.48 0.51 0.28 0.52 0.48 0.36 0.38 0.71 ...
##  $ alcohol             : num  10.1 8.8 10.5 9.3 11.2 9.8 9.8 8.9 10.2 10 ...
##  $ quality             : int  6 6 8 5 6 5 5 6 6 5 ...
```

```
summary(wine_tree2)
```

```
##
## Regression tree:
## tree(formula = quality ~ ., data = wine_train)
## Variables actually used in tree construction:
## [1] "alcohol"             "volatile.acidity"    "density"
## [4] "free.sulfur.dioxide"
## Number of terminal nodes:  7
## Residual mean deviance:  0.5602 = 2191 / 3911
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -3.6040 -0.3536 -0.2257  0.0000  0.6464  3.6460
```

```
wine_tree <- tree(quality ~ ., data = w_train)
cv.wine <- cv.tree(wine_tree)

plot(cv.wine$size,cv.wine$dev,type='b')
```



```
best.size <- cv.wine$size[which.min(cv.wine$dev)]
best.size
```

```
## [1] 5
```

```
pruned_wtree <- prune.tree(wine_tree, best=5)
str(pruned_wtree)
```

```
## List of 6
##  $ frame  :'data.frame': 9 obs. of  5 variables:
##   ..$ var   : Factor w/ 12 levels "<leaf>","fixed.acidity",..: 12 3 1 1 7 1 12 1 1
##   ..$ n     : num [1:9] 3919 2477 1186 1291 1442 ...
##   ..$ dev   : num [1:9] 3063 1468 682 623 1097 ...
##   ..$ yval  : num [1:9] 5.88 5.61 5.87 5.36 6.34 ...
##   ..$ splits: chr [1:9, 1:2] "<10.85" "<0.2525" "" "" ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : NULL
##   .. .. ..$ : chr [1:2] "cutleft" "cutright"
##  $ where  : Named int [1:3919] 4 4 3 3 4 4 4 8 6 3 ...
##   ..- attr(*, "names")= chr [1:3919] "1" "2" "4" "5" ...
##  $ terms  :Classes 'terms', 'formula'  language quality ~ fixed.acidity + volatile.acidity + citric.a
##   .. ..- attr(*, "variables")= language list(quality, fixed.acidity, volatile.acidity, citric.acid, r
##   .. ..- attr(*, "factors")= int [1:12, 1:11] 0 1 0 0 0 0 0 0 0 0 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:12] "quality" "fixed.acidity" "volatile.acidity" "citric.acid" ...
##   .. .. .. ..$ : chr [1:11] "fixed.acidity" "volatile.acidity" "citric.acid" "residual.sugar" ...
##   .. ..- attr(*, "term.labels")= chr [1:11] "fixed.acidity" "volatile.acidity" "citric.acid" "residua
##   .. ..- attr(*, "order")= int [1:11] 1 1 1 1 1 1 1 1 1 1 ...
##   .. ..- attr(*, "intercept")= int 1
##   .. ..- attr(*, "response")= int 1
##   .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##   .. ..- attr(*, "predvars")= language list(quality, fixed.acidity, volatile.acidity, citric.acid, re
##   .. ..- attr(*, "dataClasses")= Named chr [1:12] "numeric" "numeric" "numeric" "numeric" ...
##   .. .. ..- attr(*, "names")= chr [1:12] "quality" "fixed.acidity" "volatile.acidity" "citric.acid"
##  $ call   : language tree(formula = quality ~ ., data = w_train)
##  $ y      : Named int [1:3919] 6 6 6 6 6 6 6 6 5 5 ...
##   ..- attr(*, "names")= chr [1:3919] "1" "2" "4" "5" ...
##  $ weights: num [1:3919] 1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "class")= chr "tree"
##  - attr(*, "xlevels")=List of 11
##   ..$ fixed.acidity       : NULL
##   ..$ volatile.acidity    : NULL
##   ..$ citric.acid         : NULL
##   ..$ residual.sugar      : NULL
##   ..$ chlorides           : NULL
##   ..$ free.sulfur.dioxide : NULL
##   ..$ total.sulfur.dioxide: NULL
##   ..$ density             : NULL
##   ..$ pH                  : NULL
##   ..$ sulphates           : NULL
##   ..$ alcohol             : NULL
```

```
wine_test_pred3 <- predict(pruned_wtree, newdata=wine_test)
```

```
mse2 <- mean((wine_test$quality - wine_test_pred3)^2)
mse2
```

```
## [1] 0.6019167
```

accuracy is almost same as it was before pruning, but now I have less fewer nodes so it is easier to explain how model works.

Use random forests to analyze this data. What test MSE do you obtain?

```
library(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:rattle':
##
##     importance

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

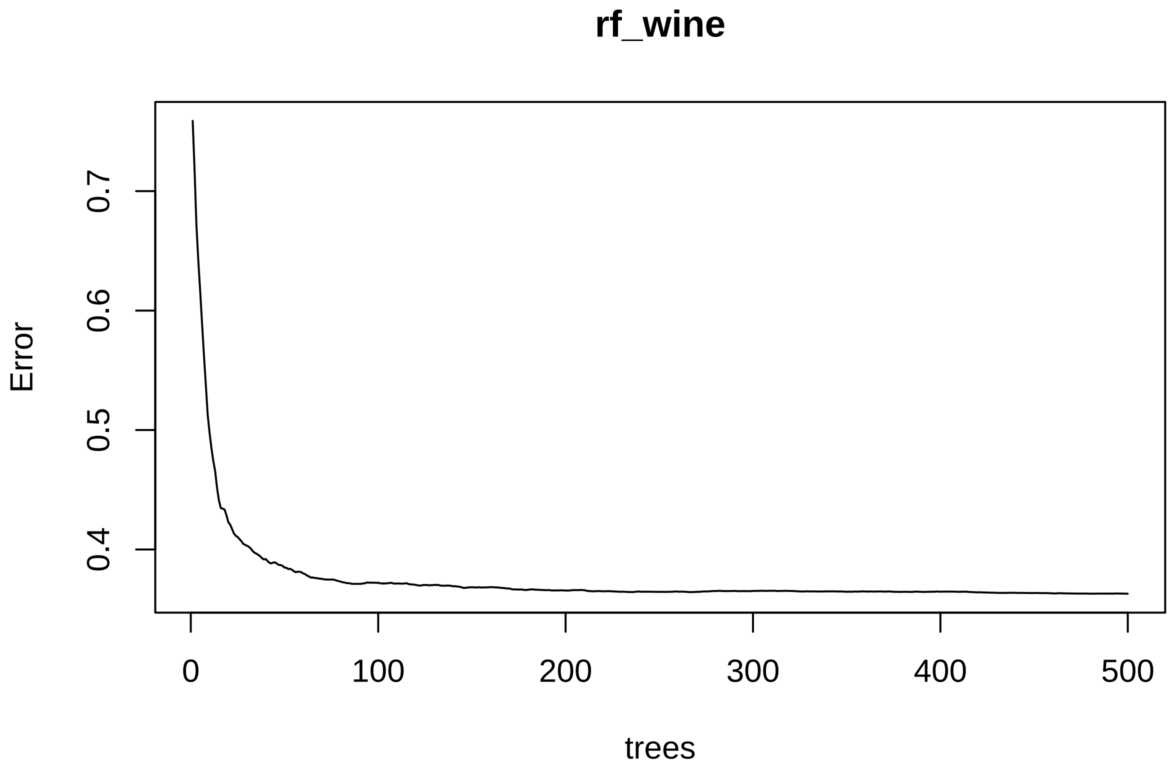
```
set.seed(123)

rf_train_indices <- sample(nrow(wine), 0.8 * nrow(wine))
rf_train_data <- wine[rf_train_indices, ]
rf_test_data <- wine[-rf_train_indices, ]
```

```
rf_wine <- randomForest(quality ~ ., data = rf_train_data)
```

```
rf_pred <- predict(rf_wine, newdata = rf_test_data)
rf_mse <- mean((rf_test_data$quality - rf_pred)^2)
```

```
plot(rf_wine)
```



```
summary(rf_wine)
```

```
##                 Length Class  Mode
## call               3   -none- call
## type               1   -none- character
## predicted       3918   -none- numeric
## mse              500   -none- numeric
## rsq              500   -none- numeric
## oob.times       3918   -none- numeric
## importance        11   -none- numeric
## importanceSD       0   -none- NULL
## localImportance    0   -none- NULL
## proximity          0   -none- NULL
## ntree              1   -none- numeric
## mtry               1   -none- numeric
## forest            11   -none- list
## coefs              0   -none- NULL
## y               3918   -none- numeric
## test               0   -none- NULL
## inbag              0   -none- NULL
## terms              3   terms  call
```

```
rf_mse
```

```
## [1] 0.3779944
```

New test MSE for random forest is much lower than I had with just a one tree model.

Use the importance() function to determine which variables are most important.

```
importance(rf_wine)
```

```
##                      IncNodePurity
## fixed.acidity             180.1598
## volatile.acidity          310.7747
## citric.acid               201.6107
## residual.sugar            220.5591
## chlorides                 252.3272
## free.sulfur.dioxide       314.1201
## total.sulfur.dioxide      228.8406
## density                   334.9593
## pH                        201.4173
## sulphates                 180.7872
## alcohol                   501.5684
```

The variable alcohol has the greatest relevance rating which is around 500 . Second highest relevant rating variable I can specify is the density. Also I can highlight volatile.acidity and free.sulfur.dioxide. The alcohol variable is the most significant one for my random forest model.