What characteristics does a good wine have?

```
knitr::include_graphics("wine.png")
```

The data

The dataset contains information about 1599 red wines from Portugal.

You can find the dataset here

1. Exploring the data

Describing the data

```
library(kableExtra)
wine <- read.csv("winequality-red.csv")
wine <- na.omit(wine)
kable(head(wine)) %>%
kable_styling(full_width = T)
```

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

```
desc_stats <- data.frame(
Min = apply(wine, 2, min),
Med = apply(wine, 2, median),
Mean = apply(wine, 2, mean),
SD = apply(wine, 2, sd),
Max = apply(wine, 2, max)
)
desc_stats <- round(desc_stats, 1)
kable(desc_stats) %>%
kable_styling(full_width = T)
```

	Min	Med	Mean	SD	Max
fixed.acidity	4.6	7.9	8.3	1.7	15.9
volatile.acidity	0.1	0.5	0.5	0.2	1.6
citric.acid	0.0	0.3	0.3	0.2	1.0
residual.sugar	0.9	2.2	2.5	1.4	15.5
chlorides	0.0	0.1	0.1	0.0	0.6
free.sulfur.dioxide	1.0	14.0	15.9	10.5	72.0
total.sulfur.dioxide	6.0	38.0	46.5	32.9	289.0
density	1.0	1.0	1.0	0.0	1.0
pH	2.7	3.3	3.3	0.2	4.0
sulphates	0.3	0.6	0.7	0.2	2.0
alcohol	8.4	10.2	10.4	1.1	14.9
quality	3.0	6.0	5.6	0.8	8.0

Analyzing the variables

Wine quality

Since we're interested in wine quality, we think it's appropriate to first take a look at the distribution of `quality`:

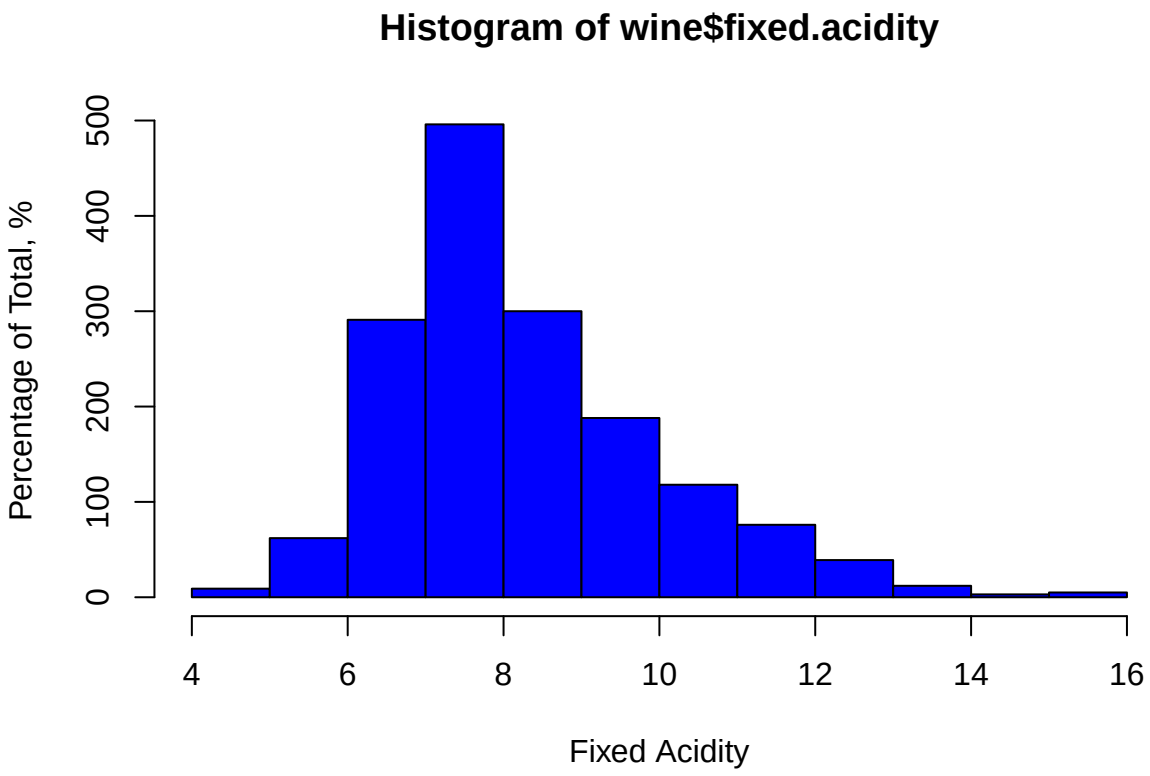
```
hist(wine$quality, col= "red", xlab = "Wine Quality", ylab = "Percentage of Total, %")
```

It's evident that the dataset has wines mainly in the categories rated 5 and 6.

Fixed acidity

The variable `fixed.acidity` charecterises the posession of wine acids and acidic salt in wines.

```
hist(wine$fixed.acidity, col= "blue", xlab = "Fixed Acidity", ylab = "Percentage of Total, %")
```

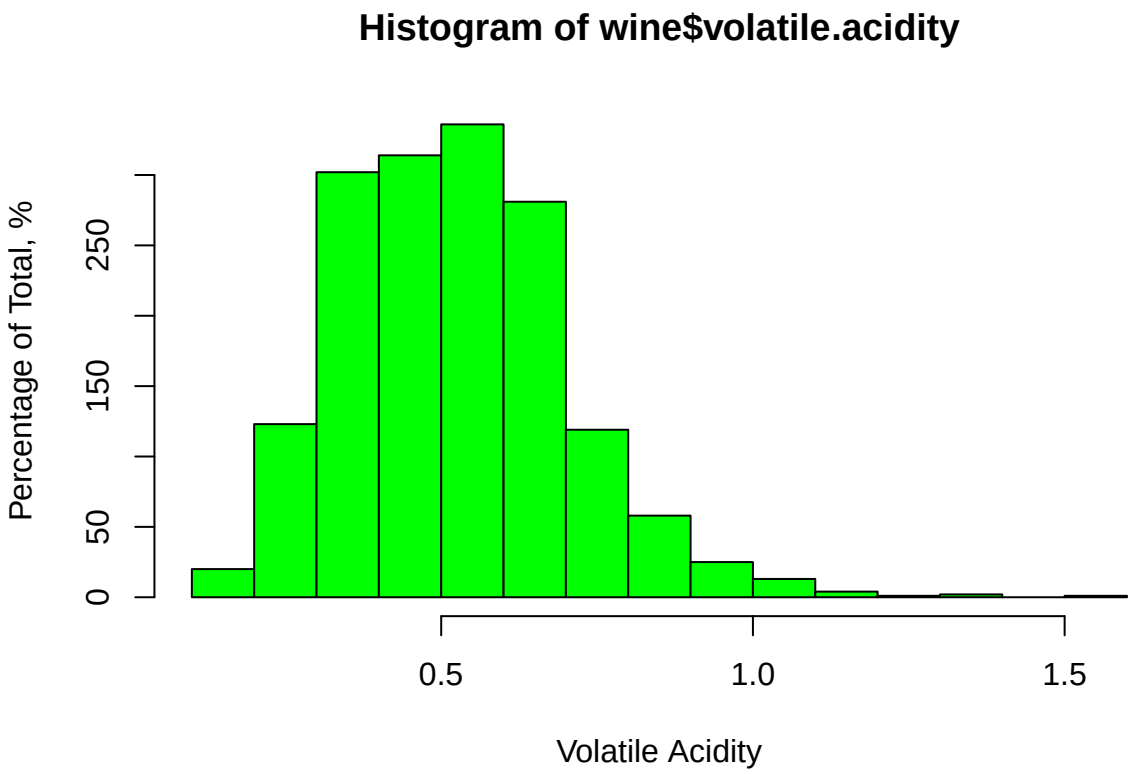


We can see that the average amount of fixed acidity in wines is around 7.5 grams per liter.

Volatile acidity

The variable `volatile.acidity` describes the amount of acetic acid in wine, which at too high of levels can lead to an unpleasant, vinegar taste.

```
hist(wine$volatile.acidity, col= "green", xlab = "Volatile Acidity", ylab = "Percentage of Total, %")
```

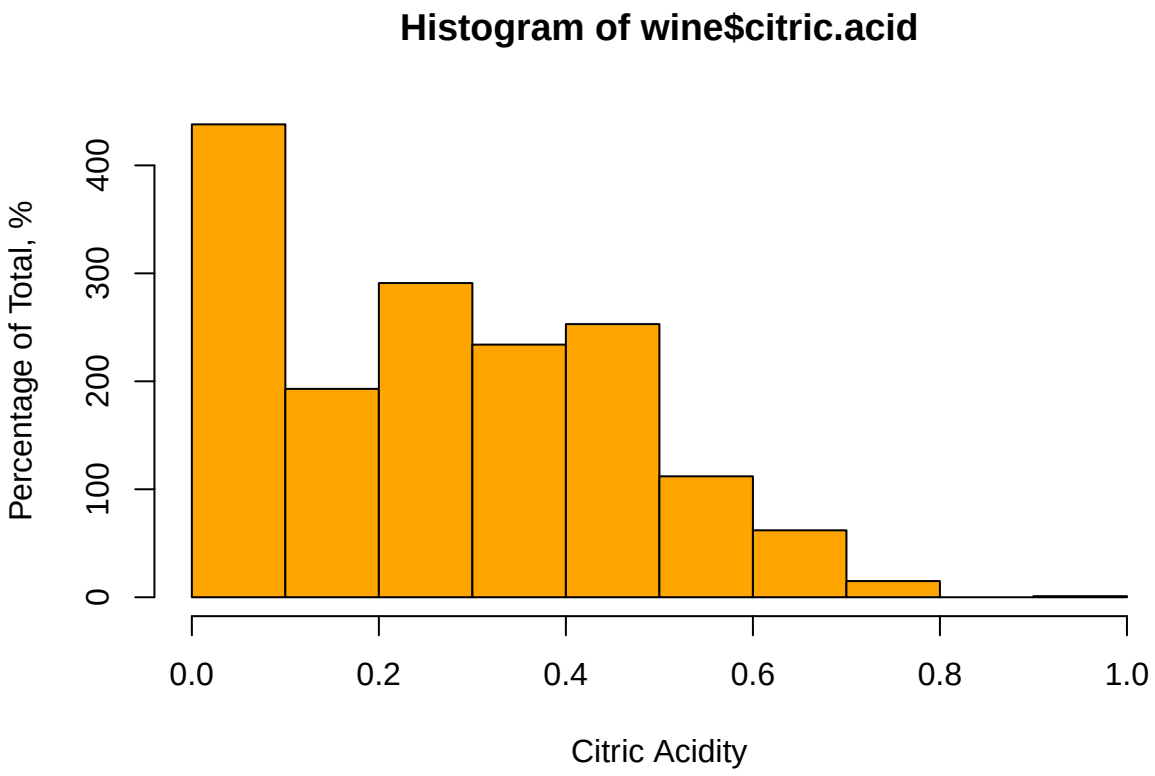


The distribution looks normal with average 0.5.

Citric acidity

Citric acid is found in small quantities, citric acid can add 'freshness' and flavor to wines.

```
hist(wine$citric.acid, col= "orange", xlab = "Citric Acidity", ylab = "Percentage of Total, %")
```

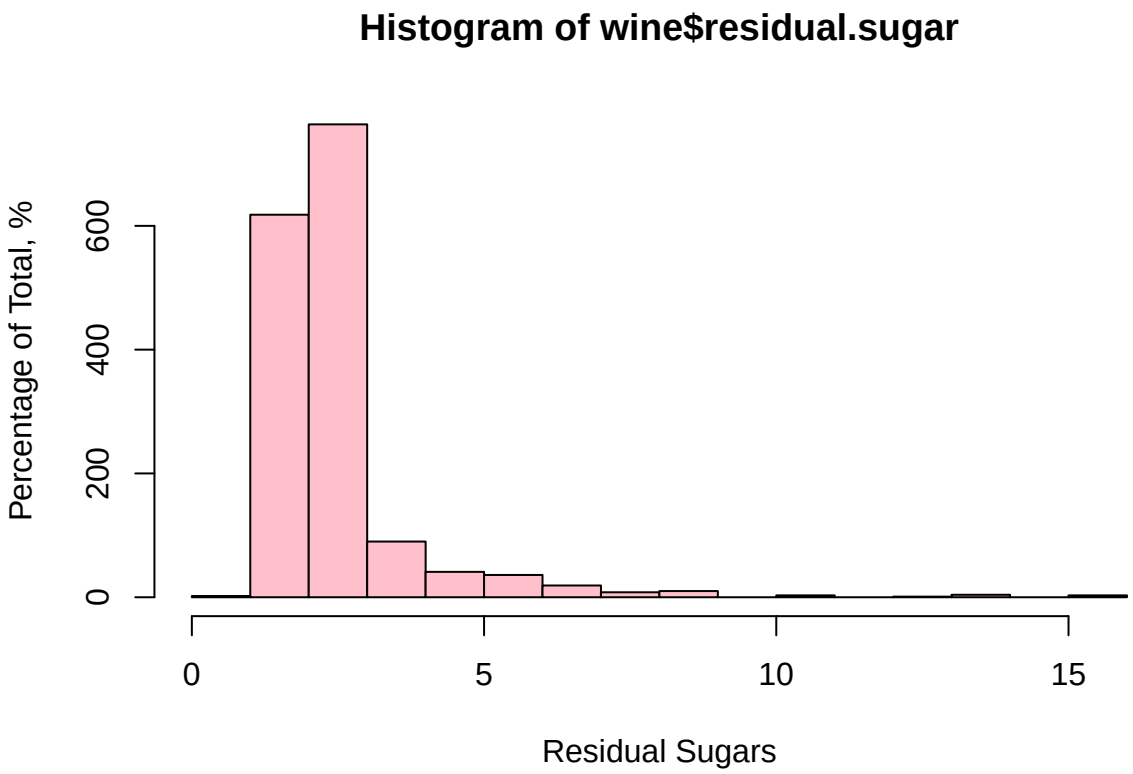


We can see that the prevalence of citric acidity is mainly in the first half of the [0,1] interval

Residual sugars

`residual.sugar` is the amount of sugar remaining after fermentation stops, it's rare to find wines with less than 1 gram/liter and wines with greater than 45 grams/liter are considered sweet.

```
hist(wine$residual.sugar, col= "pink", xlab = "Residual Sugars", ylab = "Percentage of Total, %")
```

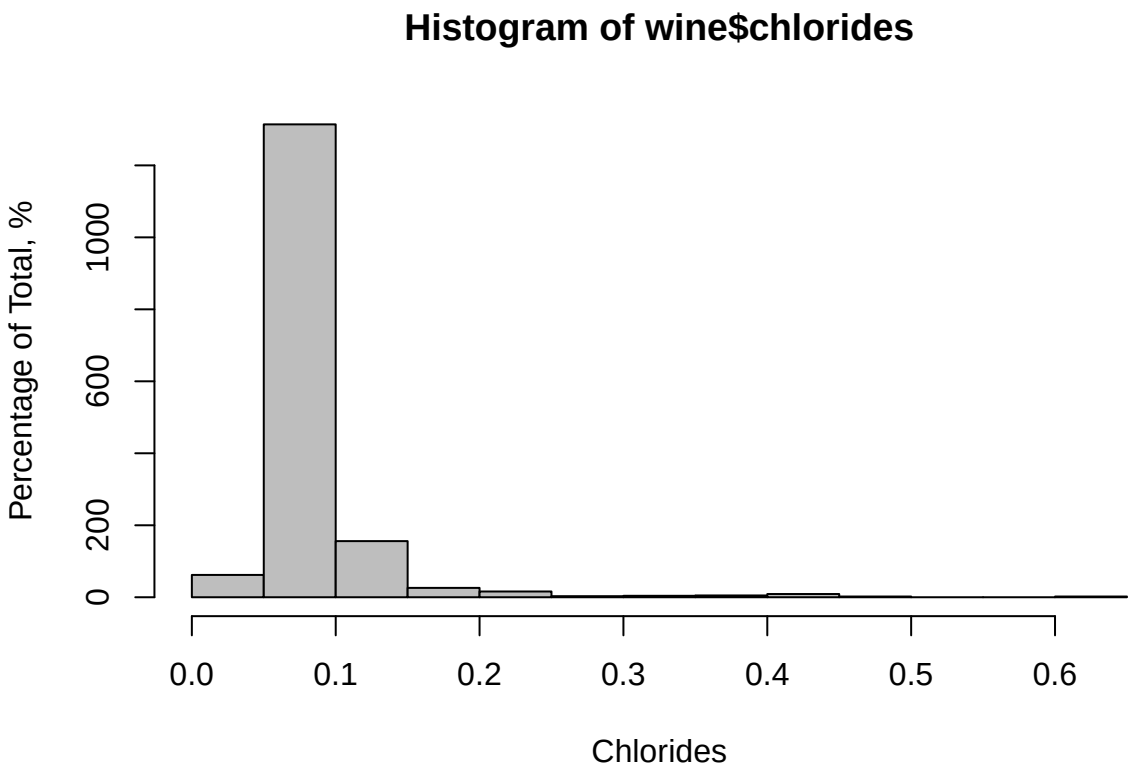


The majority of wines have 2.5 grams perliter of residual sugars.

Chlorides

The variable `chlorides` represents the amount of salt in wines.

```
hist(wine$chlorides, col= "grey", xlab = "Chlorides", ylab = "Percentage of Total, %")
```

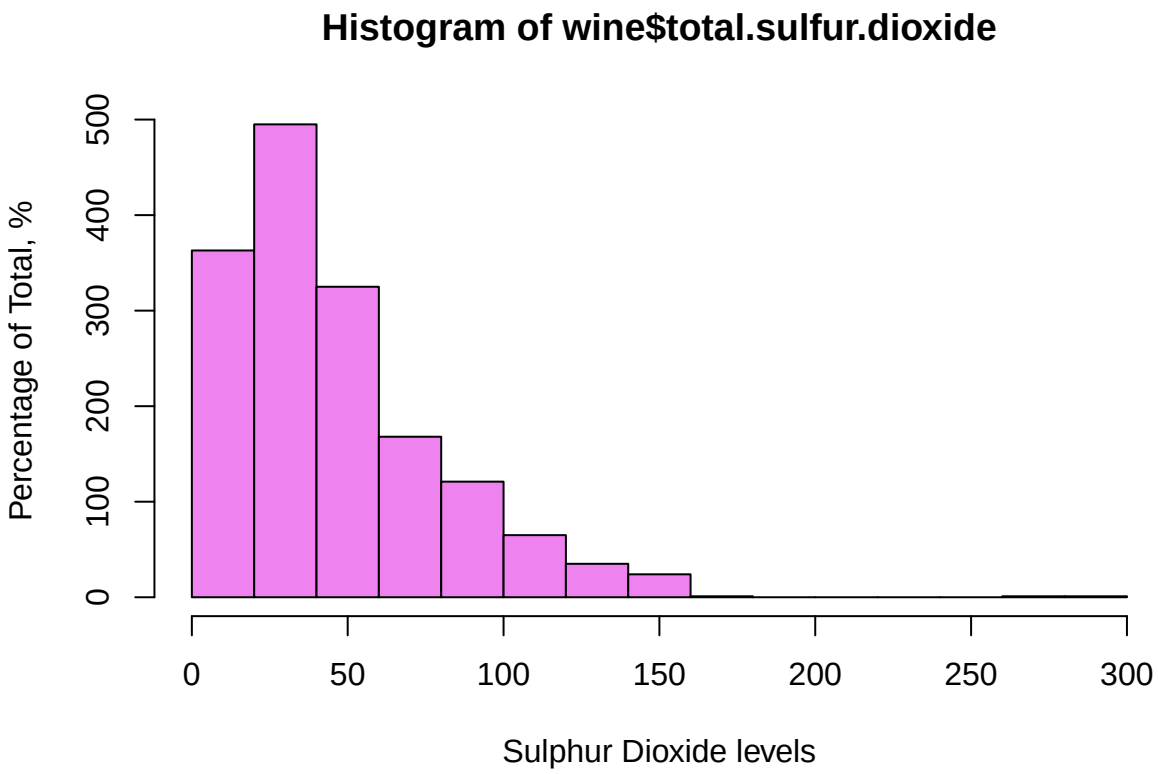


We can see that chlorides are mainly distributed in the 0.05-0.1 range.

Sulphur dioxide levels

The free form of SO2 exists in equilibrium between molecular SO2 (as a dissolved gas) and bisulfite ion; it prevents microbial growth and the oxidation of wine.

```
hist(wine$total.sulfur.dioxide, col= "violet", xlab = "Sulphur Dioxide levels", ylab = "Percentage of To
```

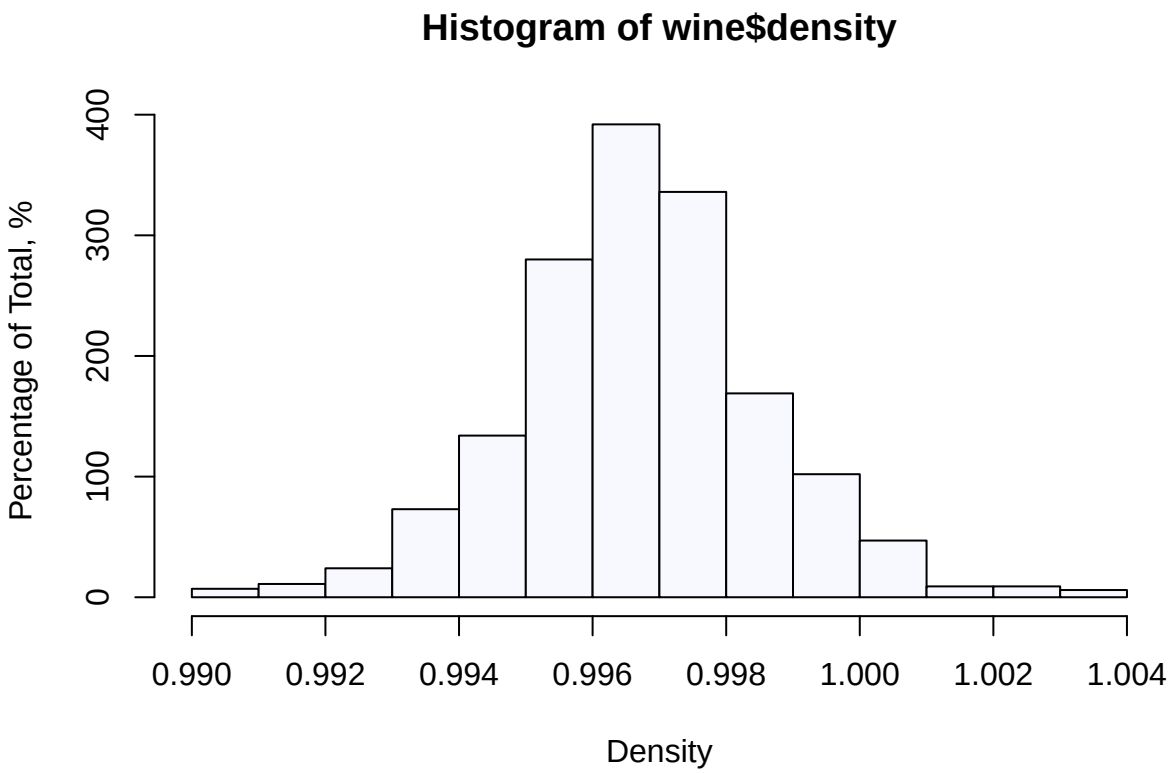


We can see that the sulphor dioxide levels are prevalent in the first half of the 0-300 interval.

Density

The density of water is close to that of water depending on the percent alcohol and sugar content.

```
hist(wine$density, col= "ghostwhite", xlab = "Density", ylab = "Percentage of Total, %")
```

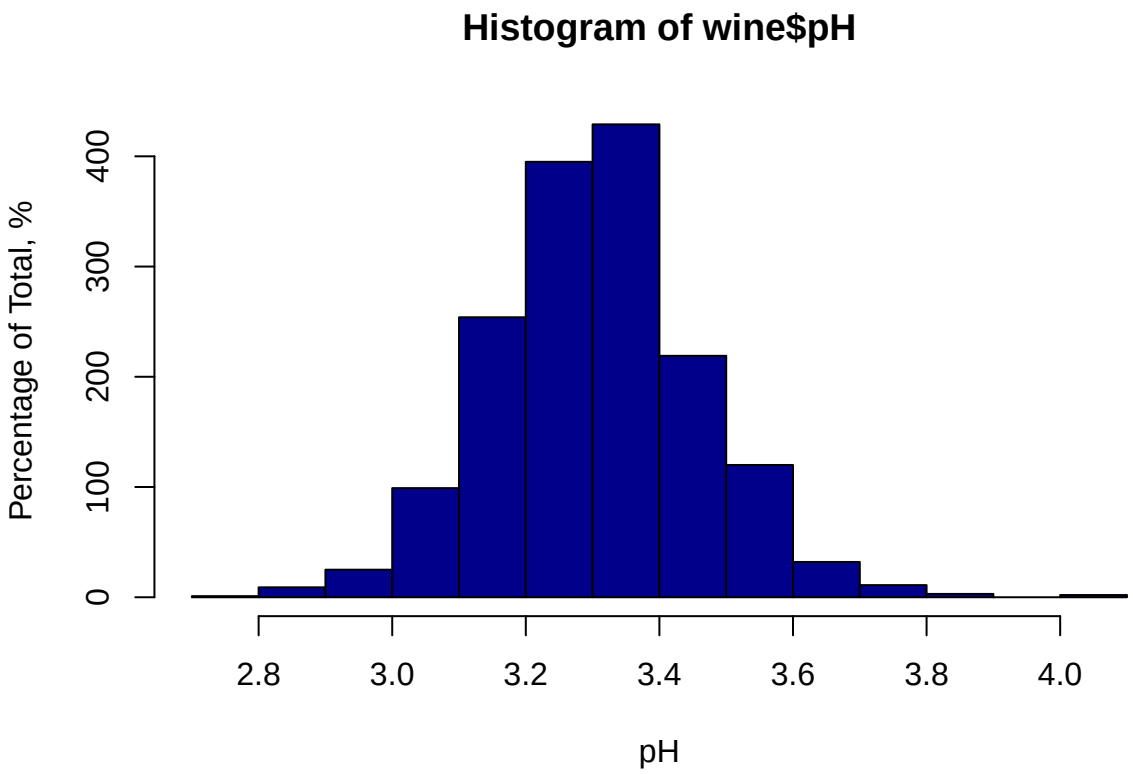


This variable looks like normaly distributed with mean 0.997.

pH

The variable `pH` describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic); most wines are between 3-4 on the pH scale.

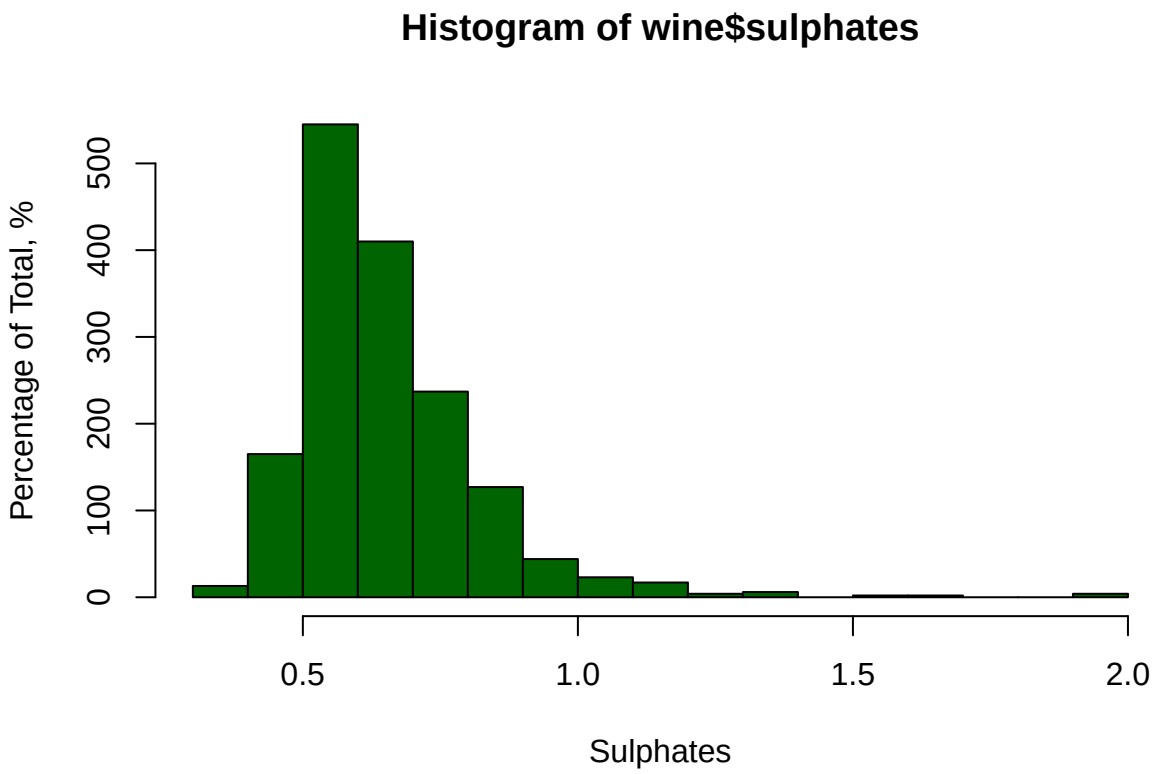
```
hist(wine$pH, col= "darkblue", xlab = "pH", ylab = "Percentage of Total, %")
```



Sulphates

The variable `sulphates` is a wine additive which can contribute to sulfur dioxide gas (S02) levels, wich acts as an antimicrobial and antioxidant.

```
hist(wine$sulphates, col= "darkgreen", xlab = "Sulphates", ylab = "Percentage of Total, %")
```



We can see that suplphates are prevalent in the 0.25-1 grams per liter interval.

Alcohol

The variable `alcohol` represents the percentage of alcohol in the wines.

```
hist(wine$alcohol, col= "lightgreen", xlab = "Alcohol", ylab = "Percentage of Total, %")
```

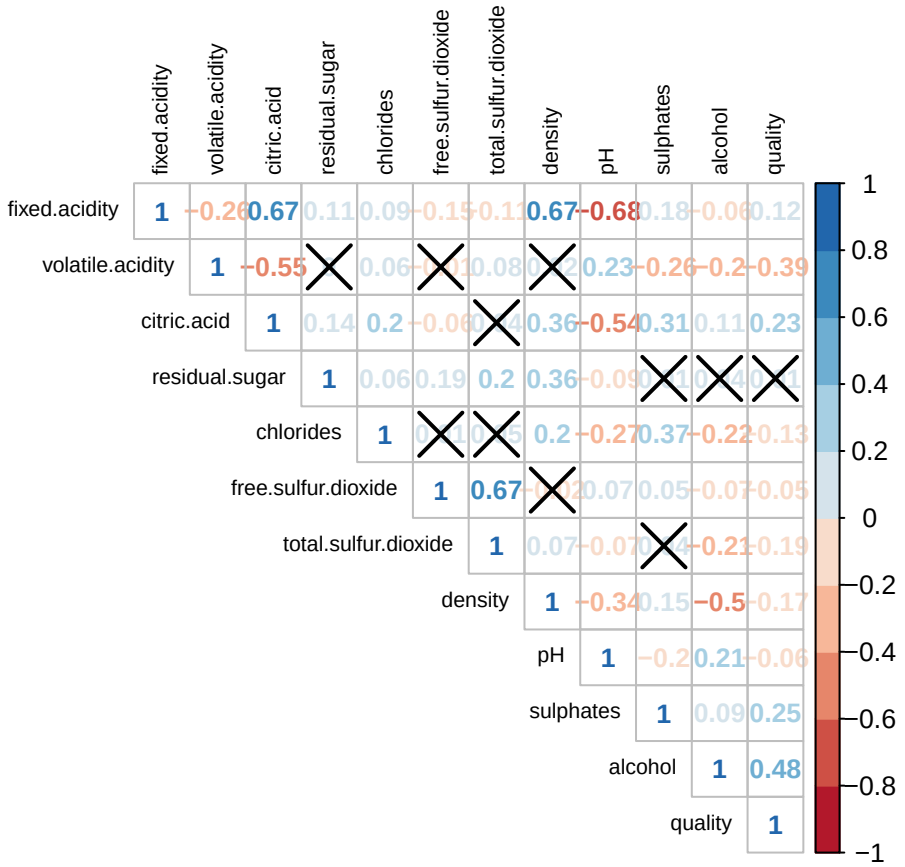
We can see that we are dealing with weak wines in this dataset.

Correlation analysis

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
wine$quality <- as.numeric(wine$quality)

col1 <- colorRampPalette(c("#B2182B", "#D6604D", "#F4A582", "#FDDBC7", "#D1E5F0", "#92C5DE", "#4393C3",

res1 <- cor.mtest(wine, conf.level = .95)

corrplot(cor(wine), method = "number", type = "upper", col = col1(10), number.cex = 0.8, tl.cex = 0.7,
p.mat = res1$p, sig.level = .05)
```



Let's recall that we're inerested in the quality of the wine. This means that we're going to focus on the correlation between `quality` and everything else. We can see that wine quality has a positive correlation with fixed acidity, citric acid, sulphates and alcohol.

We also observe that wine quality has negative correlation with volatile acidity, chlorides, sulfur dioxide, denisty,pH.Residual sugar doesn't correlate with wine quality at all, since it's a taste quality mainly.

3. Building a linear regression model

Now that we have saw the behaviour of all the variables and the correlation relationships, we can build a linear regression model. We will include only variables which have correlation coefficient higher than 0.15 or lower than -0.15 in the correlation matrix above.

Variables we will include:

- Alcohol
- Sulphates
- Total sulphur dioxide
- Citric acid
- Volatile acidity
- Density

```
library(stargazer)
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.
```

```
##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```
wine_lm <- lm(formula = wine$quality ~ wine$alcohol + wine$volatile.acidity +
wine$citric.acid + wine$density + wine$total.sulfur.dioxide +
wine$sulphates, data = wine)

stargazer(wine_lm, type = "text")
```

```
##
## ================================================
##                          Dependent variable:
##                      ---------------------------
##                                quality
## ------------------------------------------------
## alcohol                       0.305***
##                                (0.020)
##
## volatile.acidity             -1.247***
##                                (0.116)
##
## citric.acid                    -0.093
##                                (0.120)
##
## density                         9.820
##                               (11.931)
##
## total.sulfur.dioxide         -0.002***
##                                (0.001)
##
## sulphates                     0.710***
##                                (0.104)
##
## Constant                       -7.009
##                               (11.972)
##
## ------------------------------------------------
## Observations                    1,599
## R2                              0.344
## Adjusted R2                     0.342
## Residual Std. Error       0.655 (df = 1592)
## F Statistic           139.219*** (df = 6; 1592)
## ================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```

We can see that the variables with significant impact on the wine quality are alcohol, volatile acidity, total sulfur dioxide and total sulphates.

R squared in this model is not very big. This means that 34% of the quality of wines are impacted by these significant variables states above.

Moving on to multivariate statistical analysis

Let's first split the dataset into training and test subsets with proportions 80 to 20 respectively.

```r
#Cross Validation Setup

library(caret)
```

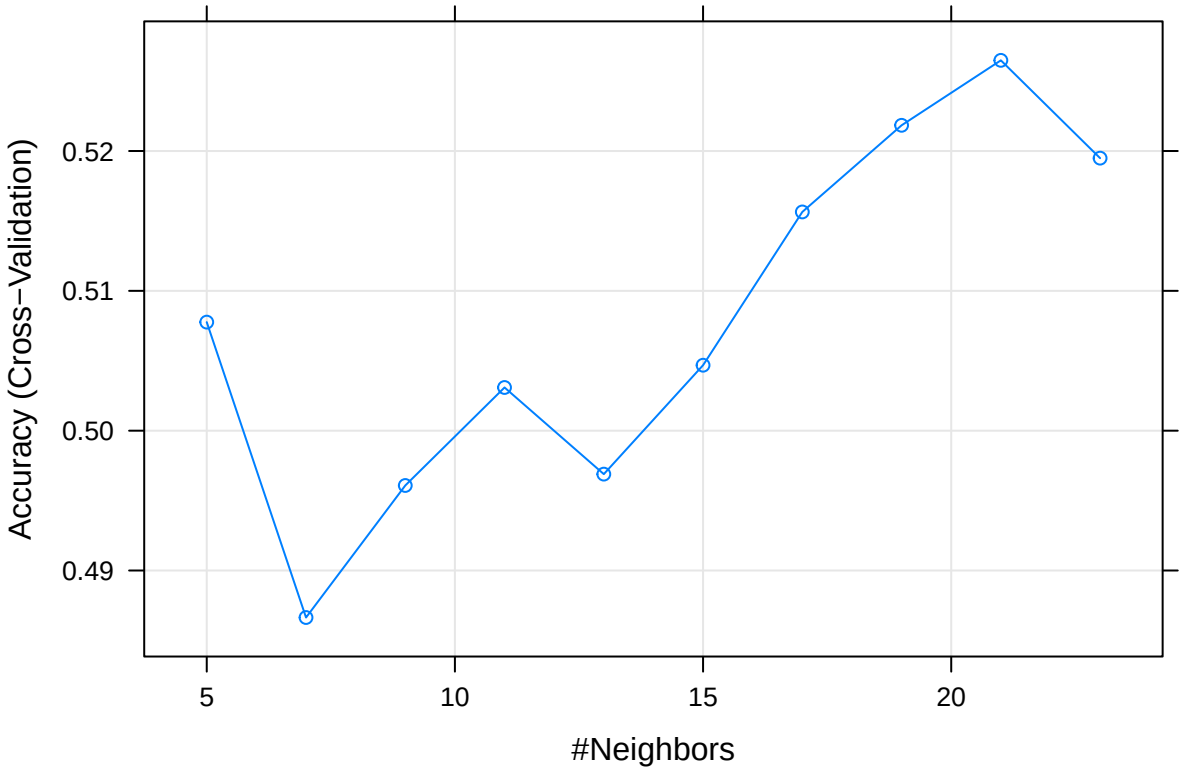
```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```r
set.seed(42)
wine$quality <- as.factor(wine$quality)

inTrain <- createDataPartition(wine$quality, p=.8, list = F)

train <- wine[inTrain,]
test <- wine[-inTrain,]


rm(inTrain)
```

4.1 k nearest neighbours (kNN)

```r
library(e1071)
k_model <- train(
quality ~ .,
tuneLength = 10,
data = train, method = "knn",
trControl = trainControl(method = "cv", number = 10, verboseIter = FALSE)
)

plot(k_model)
```




```
k_predict <- predict(k_model, test)
k_matrix <- confusionMatrix(k_predict, test$quality)
k_matrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  3  4  5  6  7  8
##          3  0  0  0  0  0  0
##          4  0  0  0  0  0  0
##          5  1  5 96 57  9  1
##          6  1  5 37 64 26  2
##          7  0  0  3  6  4  0
##          8  0  0  0  0  0  0
##
## Overall Statistics
##
##                Accuracy : 0.5174
##                  95% CI : (0.4608, 0.5735)
##     No Information Rate : 0.429
##     P-Value [Acc > NIR] : 0.0009526
##
##                   Kappa : 0.1897
##
##  Mcnemar's Test P-Value : NA
##
```

```
## Statistics by Class:
##
##                      Class: 3 Class: 4 Class: 5 Class: 6 Class: 7 Class: 8
## Sensitivity          0.000000  0.00000   0.7059   0.5039  0.10256 0.000000
## Specificity          1.000000  1.00000   0.5967   0.6263  0.96763 1.000000
## Pos Pred Value            NaN      NaN   0.5680   0.4741  0.30769      NaN
## Neg Pred Value       0.993691  0.96845   0.7297   0.6538  0.88487 0.990536
## Prevalence           0.006309  0.03155   0.4290   0.4006  0.12303 0.009464
## Detection Rate       0.000000  0.00000   0.3028   0.2019  0.01262 0.000000
## Detection Prevalence 0.000000  0.00000   0.5331   0.4259  0.04101 0.000000
## Balanced Accuracy    0.500000  0.50000   0.6513   0.5651  0.53510 0.500000
```

Result:

Accuracy=49%. We can conclude that kNN is as good as flipping a coin for predicting whether wine quality.

4.2 Decision tree

```
library("rpart.plot")
```

```
## Loading required package: rpart
```

```
library("visTree")
library("caret")
rpart_model <- rpart(quality ~ alcohol + volatile.acidity +
total.sulfur.dioxide + sulphates, train)

rpart_result <- predict(rpart_model, newdata = test[,!colnames(test) %in% c("quality")], type='class')

confusionMatrix(rpart_result, test$quality)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  3  4  5  6  7  8
##          3  0  0  0  0  0  0
##          4  0  0  0  0  0  0
##          5  1  6 87 25  2  0
##          6  1  4 47 89 24  3
##          7  0  0  2 13 13  0
##          8  0  0  0  0  0  0
##
## Overall Statistics
##
##                Accuracy : 0.5962
##                  95% CI : (0.5399, 0.6507)
##     No Information Rate : 0.429
##     P-Value [Acc > NIR] : 1.613e-09
##
##                   Kappa : 0.3414
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
```

```
##                      Class: 3 Class: 4 Class: 5 Class: 6 Class: 7 Class: 8
## Sensitivity          0.000000  0.00000   0.6397   0.7008  0.33333 0.000000
## Specificity          1.000000  1.00000   0.8122   0.5842  0.94604 1.000000
## Pos Pred Value            NaN      NaN   0.7190   0.5298  0.46429      NaN
## Neg Pred Value       0.993691  0.96845   0.7500   0.7450  0.91003 0.990536
## Prevalence           0.006309  0.03155   0.4290   0.4006  0.12303 0.009464
## Detection Rate       0.000000  0.00000   0.2744   0.2808  0.04101 0.000000
## Detection Prevalence 0.000000  0.00000   0.3817   0.5300  0.08833 0.000000
## Balanced Accuracy    0.500000  0.50000   0.7259   0.6425  0.63969 0.500000
```

Result:

We can again see that the accuracy of the model is not so good (57%), so we are not going to elaborate on anything else.

4.3 Random forest

```
library(randomForest)
```

```
## randomForest 4.6-14
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
rf_model <- randomForest(quality ~ alcohol + volatile.acidity +
total.sulfur.dioxide + sulphates, train)

rf_result <- predict(rf_model, newdata = test[,!colnames(test) %in% c("quality")])

confusionMatrix(rf_result, test$quality)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   3   4   5   6   7   8
##          3   0   0   0   0   0   0
##          4   0   0   0   1   0   0
##          5   2   8 113  24   5   0
##          6   0   2  23  93  16   1
##          7   0   0   0   8  18   2
##          8   0   0   0   1   0   0
##
## Overall Statistics
##
##                Accuracy : 0.7066
##                  95% CI : (0.6532, 0.7562)
##     No Information Rate : 0.429
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5212
```

```
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 3 Class: 4 Class: 5 Class: 6 Class: 7 Class: 8
## Sensitivity          0.000000 0.000000   0.8309   0.7323  0.46154 0.000000
## Specificity          1.000000 0.996743   0.7845   0.7789  0.96403 0.996815
## Pos Pred Value            NaN 0.000000   0.7434   0.6889  0.64286 0.000000
## Neg Pred Value       0.993691 0.968354   0.8606   0.8132  0.92734 0.990506
## Prevalence           0.006309 0.031546   0.4290   0.4006  0.12303 0.009464
## Detection Rate       0.000000 0.000000   0.3565   0.2934  0.05678 0.000000
## Detection Prevalence 0.000000 0.003155   0.4795   0.4259  0.08833 0.003155
## Balanced Accuracy    0.500000 0.498371   0.8077   0.7556  0.71278 0.498408
```

Result:

Random forest let us improve the accuracy to 71%, which is a significant improvement. This model also allows us to appreciate the impact of every variable on wine quality.

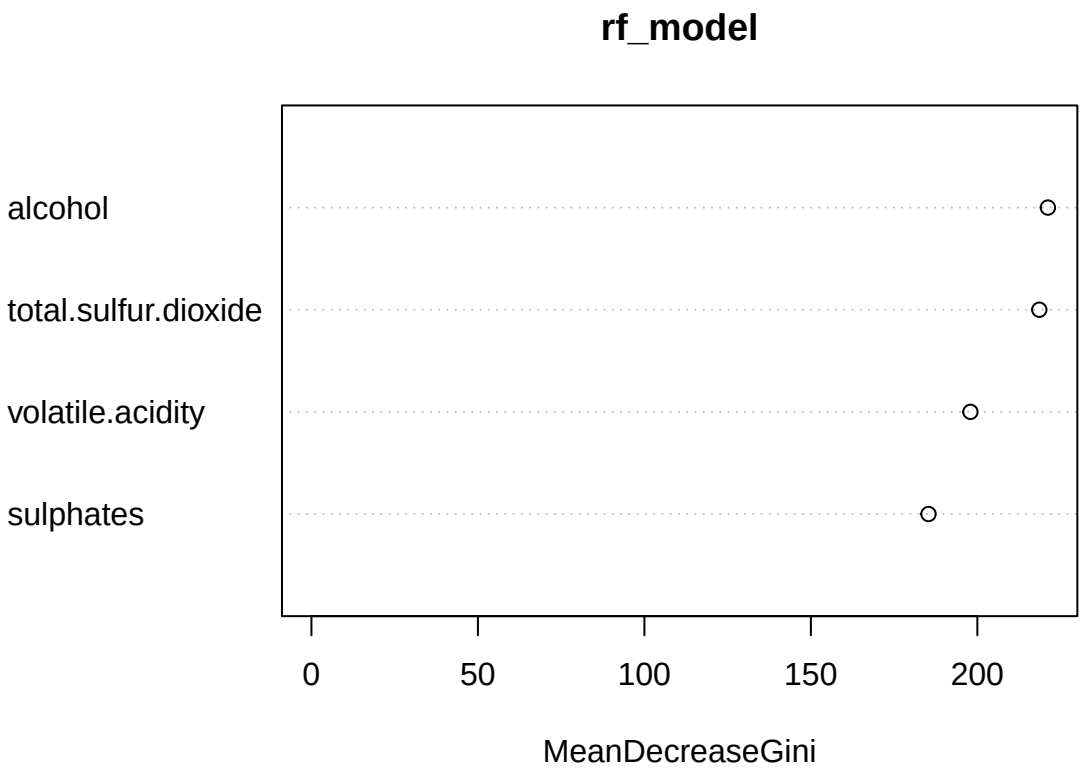
```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:randomForest':
##
##     combine

## The following object is masked from 'package:kableExtra':
##
##     group_rows

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(kableExtra)
varImp(rf_model) %>% kable()
```

	Overall
alcohol	221.1884
volatile.acidity	197.9186
total.sulfur.dioxide	218.6059
sulphates	185.3297

```
varImpPlot(rf_model)
```




```
rm(rf_model, rf_result)
```

We can confirm that the previously chosen variables indeed do have a great impact on wine quality.

SVM

```
library(e1071)
svm_model <- svm(quality ~ alcohol + volatile.acidity +
total.sulfur.dioxide + sulphates, train)

#plot(svm_model, data = train, quality ~., train)

svm_result <- predict(svm_model, newdata = test[,!colnames(test) %in% c("quality")])

confusionMatrix(svm_result, test$quality)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   3   4   5   6   7   8
##          3   0   0   0   0   0   0
##          4   0   0   0   0   0   0
##          5   2   8 104  34   4   0
##          6   0   2  32  90  27   2
##          7   0   0   0   3   8   1
##          8   0   0   0   0   0   0
```

```
##
## Overall Statistics
##
##                Accuracy : 0.6372
##                  95% CI : (0.5816, 0.6902)
##     No Information Rate : 0.429
##     P-Value [Acc > NIR] : 7.119e-14
##
##                   Kappa : 0.3916
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 3 Class: 4 Class: 5 Class: 6 Class: 7 Class: 8
## Sensitivity          0.000000  0.00000   0.7647   0.7087  0.20513 0.000000
## Specificity          1.000000  1.00000   0.7348   0.6684  0.98561 1.000000
## Pos Pred Value            NaN      NaN   0.6842   0.5882  0.66667      NaN
## Neg Pred Value       0.993691  0.96845   0.8061   0.7744  0.89836 0.990536
## Prevalence           0.006309  0.03155   0.4290   0.4006  0.12303 0.009464
## Detection Rate       0.000000  0.00000   0.3281   0.2839  0.02524 0.000000
## Detection Prevalence 0.000000  0.00000   0.4795   0.4826  0.03785 0.000000
## Balanced Accuracy    0.500000  0.50000   0.7498   0.6885  0.59537 0.500000
```

Result:

Accuracy=58%. No improvement from our random forest model, so we don't elaborate on its results.

5. The same models but with a new variable

Since we're interested in the quality of wine, it makes sense to create a binary variable, with would split the wine quality into "good" and "other" (medium or bad).

We speculate that because of the fact that the dataset has prevalently wines of medium qualities, the models previously reviewed may be inaccurate in predictions of the best wines. So in order to get rid of that effect, we will split the dataset and add a new variable which will represent "good wine", which has ratings strictly bigger than 6.

With the following code, we will add a binary variable `good`, which will be equal to 1 if the rating of the wine is bigger than 6 and equal to 0 otherwise.

```
wine_g <- read.csv("winequality-red.csv")
wine_good <- wine_g %>%
mutate(good = ifelse(quality > 6, 1, 0))
```

Now we're going to do the same operations as we did in section 4.

Let's first split the dataset into training and test subsets with proportions 80 to 20 respectively.

```
#Cross Validation Setup
set.seed(42)

inTrain_good <- createDataPartition(wine_good$good, p = .8, list = FALSE)

train_good <- wine_good[ inTrain_good ,]
test_good <- wine_good[ - inTrain_good, ]
```

```r
ctrl <- trainControl(method = "repeatedcv", number = 5, repeats = 3)

carettrain <- train_good %>%
mutate(good = factor(good)) %>%
select(-quality)

carettest <- test_good %>%
mutate(good = factor(good)) %>%
select(-quality)
```

5.1 kNN

```r
knnfit <- train(good ~ alcohol + volatile.acidity +
total.sulfur.dioxide + sulphates, data = carettrain, method = "knn", trControl = ctrl)
knnpred <- predict(knnfit, newdata = carettest)
confusionMatrix(data = knnpred, carettest$good)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 259  45
##          1   9   6
##
##                Accuracy : 0.8307
##                  95% CI : (0.785, 0.8702)
##     No Information Rate : 0.8401
##     P-Value [Acc > NIR] : 0.7078
##
##                   Kappa : 0.1177
##
##  Mcnemar's Test P-Value : 1.908e-06
##
##             Sensitivity : 0.9664
##             Specificity : 0.1176
##          Pos Pred Value : 0.8520
##          Neg Pred Value : 0.4000
##              Prevalence : 0.8401
##          Detection Rate : 0.8119
##    Detection Prevalence : 0.9530
##       Balanced Accuracy : 0.5420
##
##        'Positive' Class : 0
##
```

Result:

We have a big improvement with accuracy equal to 83%!

5.2 Random forest

```
rffit <- train(good ~ alcohol + volatile.acidity +
total.sulfur.dioxide + sulphates, data = carettrain, method = "rf", trControl = ctrl)
rfpred <- predict(rffit, newdata = carettest)
confusionMatrix(data = rfpred, carettest$good)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 259  18
##          1   9  33
##
##                Accuracy : 0.9154
##                  95% CI : (0.8792, 0.9435)
##     No Information Rate : 0.8401
##     P-Value [Acc > NIR] : 5.733e-05
##
##                   Kappa : 0.6607
##
##  Mcnemar's Test P-Value : 0.1237
##
##             Sensitivity : 0.9664
##             Specificity : 0.6471
##          Pos Pred Value : 0.9350
##          Neg Pred Value : 0.7857
##              Prevalence : 0.8401
##          Detection Rate : 0.8119
##    Detection Prevalence : 0.8683
##       Balanced Accuracy : 0.8067
##
##        'Positive' Class : 0
##
```

Result:

Accuracy=90%!

5.4 SVM

```
svmfit <- train(good ~ alcohol + volatile.acidity +
total.sulfur.dioxide + sulphates, data = carettrain, method = "svmRadial", trControl = ctrl)
svmpred <- predict(svmfit, newdata = carettest)
confusionMatrix(data = svmpred, carettest$good)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 265  35
##          1   3  16
##
##                Accuracy : 0.8809
##                  95% CI : (0.8402, 0.9143)
##     No Information Rate : 0.8401
```

```
##     P-Value [Acc > NIR] : 0.02486
##
##                   Kappa : 0.4056
##
##  Mcnemar's Test P-Value : 4.934e-07
##
##             Sensitivity : 0.9888
##             Specificity : 0.3137
##          Pos Pred Value : 0.8833
##          Neg Pred Value : 0.8421
##              Prevalence : 0.8401
##          Detection Rate : 0.8307
##    Detection Prevalence : 0.9404
##       Balanced Accuracy : 0.6513
##
##        'Positive' Class : 0
##
```

Result:

Accuracy = 87%.

6 Conclusion

Since we're interested in the models which will predict whether wine qualities are good, we would recommend using the second approach which consists of having the binary variable.

6.1 Using the best model (Random forest) for predicting wine quality

Imagine we're in a wine shop, we see a wine with its characteristics on the side of the bottle. The characteristics are:

- citric acid = 1 gram per liter
- total sulphure dioxide = 50 grams per liter
- sulphates = 1 gram per liter
- alcohol = 11%

```
# creating an input vector with the characteristics mentioned above
q_0 <- t(as.data.frame(c(1, 50, 1, 11)))
colnames(q_0) <- colnames(wine[,c(2,7,10,11)])
q_0
```

```
##                 volatile.acidity total.sulfur.dioxide sulphates alcohol
## c(1, 50, 1, 11)                1                   50         1      11
```

```
# predicting the wine quality with those characteristics
pred <- predict(svmfit, as.data.frame(q_0), type="raw")
print(pred)
```

```
## [1] 0
## Levels: 0 1
```

That's not a wine we would recommend taking.

However, if the characteristics were of the following quality:

- citric acid = 0.6 gram per liter
- total sulphure dioxide = 45 grams per liter
- sulphates = 1 gram per liter
- alcohol = 13%

```
q_1 <- t(as.data.frame(c(0.6, 45, 1, 13)))
colnames(q_1) <- colnames(wine[,c(2,7,10,11)])
q_1
```

```
##                   volatile.acidity total.sulfur.dioxide sulphates alcohol
## c(0.6, 45, 1, 13)              0.6                   45         1      13
```

```
pred <- predict(svmfit, as.data.frame(q_1), type="raw")
print(pred)
```

```
## [1] 0
## Levels: 0 1
```

You would have a great evening with the company of this bottle, although we recommend inviting someone!

Final conclusion

We conclude that in order to pick a good wine, we recommend using the Random forest model with the binary variable mentioned before. This model predicts 9 out of 10 good wines (we don't recommend checking this yourself). So, if you're in Portugal, you can trust our model and you won't regret!

```
knitr::include_graphics("vine.png")
```